On Deep Machine Learning for Multi-view Object Detection and Neural Scene Rendering

ISAAC-MEDINA, BRIAN,KOSTADINOV,SHALON

On Deep Machine Learning for Multi-view Object Detection and Neural Scene Rendering

Brian Kostadinov Shalon Isaac Medina

A Thesis presented for the degree of
Doctor of Philosophy

Department of Computer Science
Durham University
United Kingdom
September 2023

Abstract

This thesis addresses two contemporary computer vision tasks using a set of multiple-view imagery, namely the joint use of multi-view images to improve object detection and neural scene rendering via a novel volumetric input encoding for Neural Radiance Fields (NeRF). While the former focuses on improving the accuracy of object detection, the latter contribution allows for better scene reconstruction, which ultimately can be exploited to generate novel views and perform multi-view object detection.

Notwithstanding the significant advances in automatic object detection in the last decade, multi-view object detection has received little attention. For this reason, two contributions regarding multi-view object detection in the absence of explicit camera pose information are presented in this thesis. First, a multi-view epipolar filtering technique is introduced, using the distance of the detected object centre to a corresponding epipolar line as an additional probabilistic confidence. This technique removes false positives without a corresponding detection in other views, giving greater confidence to consistent detections across the views. The second contribution adds an attention-based layer, called Multi-view Vision Transformer, to the backbone of a deep machine learning object detector, effectively aggregating features from different views and creating a multi-view aware representation.

The final contribution explores another application for multi-view imagery, namely novel volumetric input encoding of NeRF. The proposed method derives an analytical solution for the average value of a sinusoidal (inducing a high-frequency component) within a pyramidal frustum region, whereas previous state-of-the-art NeRF methods approximate this with a Gaussian distribution. This parameterisation obtains a better representation of regions where the Gaussian approximation is poor, allowing more accurate synthesis of distant areas and depth map estimation.

Experimental evaluation is carried out across multiple established benchmark datasets to compare the proposed methods against contemporary state-of-the-art architectures such that the efficacy of the proposed methods can be both quantitively and qualitatively illustrated.

Declaration

The work in this thesis is based on research carried out at the Department of Computer Science, Durham University, United Kingdom. No part of this thesis has been submitted elsewhere for any other degree or qualification and it is all my own work unless referenced to the contrary in the text.

Acknowledgements

First, I would like to express my deepest gratitude to my supervisors, Prof. Toby Breckon for his invaluable teaching and support during this four-and-a-half-year journey, and Dr Chris Willcocks for his key insights and discussions on each work. I am entirely grateful and completely sure that I would not have been able to finish this work without their help.

A mi mamá, por su inmenso cariño y comprensión día tras día, por estar ahí siempre, por ser la única que me ha visto crecer en lo personal y profesional. Por esos momentos de alegría y tristeza de estar lejos de casa persiguiendo mis metas. A mi papá, por ser el que más ha creído en mí. Por enseñarme lo que realmente valgo. Por ese pedacito de su cerebro que me ponía en mis licuados en la mañana antes de ir a la primaria. Estoy seguro de que no hay nadie más orgulloso que don Nacho al ver hasta dónde he llegado. Gracias papá. Gracias mamá.

A Diana, mi esposa, por su tremendo amor y apoyo durante esta aventura incluso cuando ella misma está realizando su PhD y al mismo tiempo planeó nuestra boda. Porque en los días más difíciles y en los días más felices, ella siempre ha estado a mi lado, siempre ha sido mi zona de confort, mi descanso. Por creer en mí e impulsarme en un principio a hacer el posgrado. Gracias Cori, este doctorado lo he logrado gracias a ti.

This endeavour would not have been possible without my amigos and colleagues from X-rayfess: Yona, Neel, Jack, Jia Lin, Zhongtian and Anoushka. Thanks for your support during these years, my experience at Durham would not be the same without you. For attending several Ustinov Lives, for welcoming me into their houses and for preparing amazing food. I think our friendship and the time here are extraordinary, I will always remember it.

A Nachito, por estar siempre al cuidado de nuestros padres mientras sigue construyendo su futuro. Estoy muy orgulloso de ti, sé que vas a ser alguien muy grande en el futuro. A mi hermano Hussein, por estar siempre pendiente de mis pasos y quien indirectamente ha sido parte de mi travesía. A mi tío Armín por todo el apoyo que nos ha dado desde pequeños. El Porsche que le prometí queda pendiente.

I am also thankful to the friends I have made along my journey at Durham from different parts of the world and different parts of Mexico. After finishing my PhD, I am happy to say that I have friends from the UK, India, China, Malaysia, Saudi Arabia, Portugal, Italy, USA, Colombia, Spain, Turkey, Iran, Sinaloa, Sonora, Chihuahua, Zacatecas, Nuevo León, Ciudad de México, Estado de México, Puebla, Tamaulipas and Veracruz (y a mis amigos yucatecos, Ba'ax ka wa'alik peel a na's). Special mention to the members of Los Patroncitos, The Observatories and Los Aluxes del Norte, because you gave me countless hours of fun. I loved to play with you guys, we will always be the best bands to perform at Ustinov.

A mis amigos en Mérida que cada vez que regreso me reciben con los brazos abiertos (y una caguama también), como si ningún día hubiese pasado desde que me fui. A Edgar, Sergio, Juan Pablo, Rodrigo, Víctor y Edwam, por su amistad de más de 15 años. A Héctor, Pedrito, (mi patrón) Gustavo, Cuy, Alonso, Poncho. Gracias por su amistad a pesar de que solo nos veamos una vez al año.

Finally, I would like to thank the Mexican Council of Science and Technology (CONACyT) for funding my PhD research.

Contents

List of Figures

List of Tables

Dedication

To my parents. To my wife.

CHAPTER 1

Introduction

This thesis comprises a two-fold research theme consisting of the use of multiple views to improve both object detection performance and novel-view synthesis. For the former, two strategies are investigated, namely the use of epipolar geometry constraints (*i.e.*, two-view geometric relations) and image feature fusion with contemporary deep learning architectures, while for the latter a reparameterisation of a highly influential neural scene rendering work is explored. While these research themes focus on different aspects of multi-view imagery, having better scene reconstructions using neural radiance fields might allow for the rendering of novel views which can be used to perform multi-view object detection. Similarly, multi-view object detection of 2D images can be exploited to localise objects in the 3D space of neural radiance fields.

Object detection is a classical computer vision task involving the location of an object within an image. The output of an object detector is usually a bounding box that surrounds the object of interest, along with a corresponding category. Since the general embracing of deep learning for computer vision in the early 2010s [1], several works have investigated different deep neural networks (DNN) for object detection [2–8]. The use of more extensive and increasingly diverse datasets has helped in the design of architectures that improve upon detection performance of challenging instances, such as small or occluded objects. Nonetheless, a particular scenario that has not been thoroughly investigated is the joint use of multiple views from the same scene for image recognition tasks, such as object detection. In this sense, this thesis explores the use of two techniques for multi-view object detection (Chapters 3 and 4) that have direct applications to both automated surveillance and image interpretation for X-ray security imagery.

Another task that is investigated in this work is novel view synthesis. The objective is to synthesize images at unseen views from input images and their respective camera poses. Different approaches have been taken for this goal, including the use of multi-view geometry constraints [9], numerical optimisation [10] and DNN [11]. A technique that has attracted significant recent attention is Neural Radiance Fields (NeRF) [12], which learns an implicit representation of the scene using a DNN to predict the colour and density of 3D points in space. In this thesis, a new formulation within the NeRF architecture to account for volumetric regions in space is presented (Chapter 5).

1.1 Motivation

The increasing advances in visual data acquisition devices, such as digital cameras, have brought cheaper and better quality devices that are accessible by virtually anyone. This has resulted in environments with multiple cameras looking at the same scene, such as in video surveillance or self-driving cars. Many industrial and medical applications require the capture of multi-view images, either because of their inherent acquisition process or the manipulation of the data. The importance of multiple views is that they provide different points of view where some regions or objects might be better visualized, either because they might be occluded or have unrecognizable shapes in some views. In general, multiple views usually mean more information about a scene that can be exploited in modern machine learning applications.

In computer vision, the geometry constraints and redundancy of multi-view images can be used for applications that are too complex (or impossible) with single-view images. Self-driving cars use stereo-vision by placing two cameras slightly separated and with the same field of view (FoV) in order to estimate the depth of the scene or to make choices based on the surrounding situation [13–15]. In the case of video surveillance, the occlusion of objects of interest can be reduced by looking at different views, resulting in techniques that track and identify people more accurately when compared with single-view tracking [16–19]. Many multi-view applications are also found in medical image analysis, such as for detection of anomalies, tissue classification and segmentation [20–23]. The diversity of applications and the gains in performance make the study of multi-view image analysis a subject worth dedicated research.

Despite the advances in multi-view image recognition, an area that remains under investigated is the simultaneous localization (and classification) of objects in different synchronized views, namely object detection. In some scenarios, such as medical and airport screening, a sparse collection of views is obtained for a single scene (*e.g.*, a patient or a passenger bag), with some instances having as little as two views. When object detection is a crucial task in the application context, the integration of the information from different views may boost the performance of both human operators and automated detection techniques. For instance, the visual inspection of X-ray cabin baggage by security operators at airports is significantly improved when two perpendicular views are used instead of one [24]. Multi-view detection can also improve tracking since many techniques are based on object detection followed by a matching algorithm across multiple sequential frames [25]. The difficulty of this task is that the geometry of the bounding boxes used to localize the objects might be complicated, especially if the relative position of the cameras is unknown. Furthermore, there are only a few public datasets with paired multi-view object-level annotations, making it difficult to assess the performance of such techniques for general applications. Some of these datasets, particularly the X-ray security imagery datasets, do not make the calibration publicly available. In addition, X-ray machines are not available to perform standard calibration. For this reason, multi-view object detection under these constraints is investigated in this work.

In addition to multi-view recognition, 3D reconstruction can be achieved if the relative position of the cameras is known or, equivalently, point correspondences are given or calculated. After the ground theory of the projective geometry of multiple views images was developed in the 1990s [26–29], the first applications for scene reconstruction from image sequences began to emerge [30, 31]. Furthermore, 3D scene reconstruction has benefited from the advances in deep learning, with more robust applications that handle occlusions [32] or complex scenes [33, 34]. Multiple views not necessarily synchronized or with unknown camera pose information can also be used to find point or line correspondences, which ultimately help in 3D reconstruction [35, 36].

A different perspective to 3D scene reconstruction is the generation of unseen views of the scene from a set of multi-view images. In this sense, the explicit computation of the 3D geometry is not necessary. Several deep learning techniques are able to learn the relations between views and aggregate them accordingly to generate new views [37–39]. An approach that has enjoyed substantial popularity recently is neural radiance fields (NeRF) [12], an image-based rendering technique which learns the visual field from multiple input images in order to render arbitrary novel views. This technique has proven to be very effective for many applications, such as city-level rendering [40] or medical 3D rendering [41]. A subsequent work related to NeRF, mip-NeRF [42], changes the original formulation of the input data to account for 3D volumetric regions, which has proven to be effective and is used in several variants. Although some follow-up works to this formulation have been proposed [43, 44], there is still notable room for improvement within the underlying applications.

Considering the recent advances and the potential applications, it is clear that multi-view imagery has the potential to boost image recognition and reconstruction performance, although their correct integration may be challenging. For this reason, this thesis is focused on the exploitation of multiple views in two tasks with promising applications, namely multi-view object detection and novel-view synthesis using NeRF.

1.2 Contributions

The main contributions of this thesis are the following:

- A multi-view detection approach that considers inter-view epipolar constraints as an additional measure of confidence within the non-maximum suppression post-processing step to cross-correlate detections from multiple views and improve detection performance by eliminating false positives. (Chapter 3)

- A novel Vision Transformed-based architecture [45] for multi-view object detection, called Multi-view Vision Transformer (MVViT), which aggregates the feature maps from different views to create a 3D geometry-aware feature representation. Under this framework, MVViT is integrated into three general object detection architectures, specifically, YOLOX [46], Deformable DETR [47] and Swin Transformers [7], demonstrating that MVViT is detector agnostic and it improves multi-view object detection performance. (Chapter 4)

- A novel formulation of the positional encoding of the mip-NeRF neural rendering architecture [42] based on a change from casting conical to pyramidal frustums within the underlying formulation. This reparameterisation thus facilitates solving the underpinning volumetric integral exactly, which is otherwise approximated with a multivariate Gaussian in mip-NeRF. This subsequently shows its potential to model distant objects in mip-NeRF 360 [43] with a comparably higher degree of accuracy. (Chapter 5)

1.2.1 Publications

The contributions in within this thesis have been published in the following peer-reviewed publications:

- Brian K.S. Isaac-Medina, Chris G. Willcocks and Toby P. Breckon. “Multi-view Object Detection Using Epipolar Constraints within Cluttered X-ray Security Imagery”. in *IEEE International Conference on Pattern Recognition (ICPR)*, pp. 9889-9896, 2020.

- Brian K.S. Isaac-Medina, Chris G. Willcocks and Toby P. Breckon. “Multi-view Vision Transformers for Object Detection”, in *IEEE International Conference on Pattern Recognition (ICPR)*, pp. 4678-4684, 2022.

- Brian K.S. Isaac-Medina, Chris G. Willcocks and Toby P. Breckon. “Exact-NeRF: An Exploration of a Precise Volumetric Parameterisation for Neural Radiance Fields”, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 66-75, 2023.

Additionally, two works related to Dual-energy X-ray imagery were published during the development of this thesis and are detailed in Appendices A and B:

- Brian K.S. Isaac-Medina, Neelanjan Bhowmik, Chris G. Willcocks and Toby P. Breckon. “Cross-modal Image Synthesis in Dual-Energy X-Ray Security Imagery”, in *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 332-340, 2022.

- Brian K.S. Isaac-Medina, Seyma Yucer, Neelanjan Bhowmik and Toby P. Breckon. “Seeing Through the Data: A Statistical Evaluation of Prohibited Item Detection Benchmark Datasets for X-Ray Security Screening”, in *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 524-533, 2023.

1.3 Scope

This thesis focuses on multi-view object detection and neural-based image rendering. However, multi-view image processing may have different meanings depending on the application. Yan *et al*. [48] define multi-view data as being captured in different modalities, spaces or sources. In this thesis, the term *multi-view* is used in the context of *multi-view stereo vision*: images of the same scene with different camera positions, not necessarily synchronized, albeit within the same or a similar modality.

The main goal of the first two contributions (Chapters 3 and 4) is to improve object detection performance, where the performance metric indicates the effectiveness of the proposed techniques to localize the objects in the image. This thesis deals with the case of multiple synchronized cameras with highly overlapping FoV. Henceforth, if an object is in one view, it will likely appear in another view and be constrained by the multi-view geometry. Although this thesis works with datasets with more than two views, the algorithms developed for object detection are based on epipolar geometry (*i.e.*, the intrinsic geometry of stereo vision, or two-view geometry) or the correlation between two feature maps. In traditional stereo vision, cameras are separated horizontally, as in Fig. 1.1a. In contrast, this work deals with views at arbitrarily oriented and placed cameras (although with some restrictions). Two different use cases for object detection are tested, namely X-ray security imagery and visual surveillance. In the former, the objects of interest always lie in the FoV of all the cameras (Fig. 1.1b), with the particular transparent nature of transmission imagery, where objects appear overlapped instead of occluded. In the latter, the different views share a smaller FoV, with objects not appearing in some of them (Fig. 1.1c). Although the evaluation is focused on these two use cases, the methods developed in Chapters 3 and 4 can be applied to any multi-view setting such as medical imaging, synthetic datasets or autonomous driving. In some instances, the methods developed in this thesis could also be applied to video datasets since they can be seen as multi-view data. While these applications are not explored, it is worth noting that there is little to no limitation to extrapolate the method to any multi-view data. As an additional consideration, the task of matching instances across all the views (re-identification) is not taken into account.

A further contribution related to the NeRF architecture is also considered (Chapter 5). In essence, NeRF uses a neural network that takes as input a 3D point in space and a viewing direction to predict the colour and density of that point. In order to account for high frequencies, NeRF uses a positional encoding $\gamma : \mathbb{R} \rightarrow \mathbb{R}^d$ on each 3D coordinate independently, consisting of a composition of sines and cosines. A follow-up work, mip-NeRF [42], encodes a volumetric region in the form of a cone frustum instead of using 3D points. However, a direct consequence is that a volumetric integral of the positional encoding is needed, which in turn has no closed-form solution. Instead of trying to calculate this integral, an approximation with a 3D Multivariate Gaussian is used in mip-NeRF and multiple follow-on

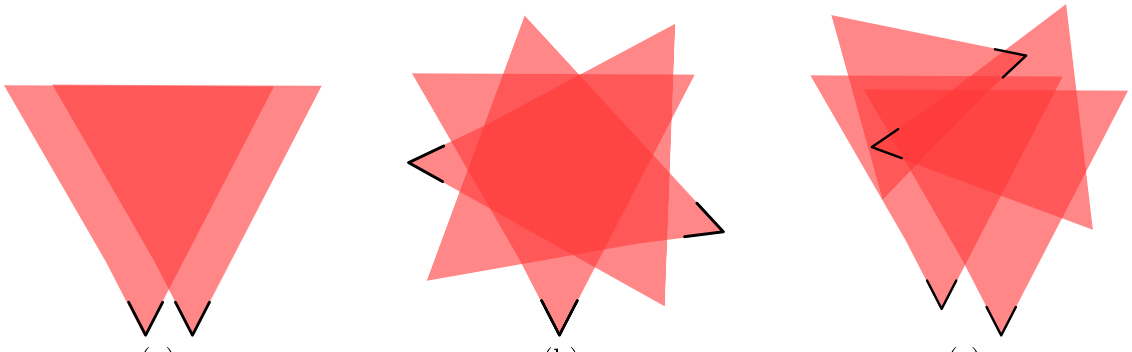

(a) (b) (c)

Figure 1.1: Different configurations for multi-view images. (a) Traditional stereo-vision, with cameras separated horizontally and with a similar FoV. (b) Multi-view stereo with a large overlapping FoV. This configuration is seen in X-ray screenings. (c) Multi-view configuration with different FoV as in outdoor surveillance systems.

contributions [40, 43, 44, 49, 50]. In this thesis, a change in the geometry of the volumetric regions is proposed, which instead allows for obtaining an exact solution for the integrated positional encoding in place of mip-NeRF approximation. We also demonstrate that it can be used without further modification in subsequent works such as mip-NeRF 360 [43], a more recent version of the NeRF architecture to account for unbounded 360° scenes.

1.4 Thesis Structure

The background theory and literature review are covered in Chapter 2. First, an introduction to multi-view geometry is presented in Section 2.1. Subsequently, Section 2.2 presents the general techniques for object detection, with a review of the state-of-the-art on multi-view object detection presented in Section 2.2.8. Finally, an overview of the theory, advances and applications of NeRF is given in Section 2.3.

The following two chapters present two methods for multi-view object detection. Chapter 3 addresses a post-processing non-maximum suppression technique that accounts for the epipolar geometry between two views in a multi-view array of cameras, aiming to reduce false positives by removing non-multi-view consistent detections. Chapter 4 presents the Multi-View Vision Transformer (MVViT), an approach for multi-view detection that combines the features from different views to create 3D-aware representations of the scenes. Both chapters have similar evaluation criteria, including evaluation among three different object detectors. Subsequently, Chapter 5 introduces Exact-NeRF, a reparameterisation of the position encoding of neural radiance fields. This chapter gives the mathematical formulation of an alternative parameterisation and compares them against mip-NeRF [42] and mip-NeRF 360 [43].

Since these techniques are integrated at different stages of a deep learning architecture (the method in Chapter 3 is a post-processing step, MVViT is a modification of a neural network for object detection and Exact-NeRF is a pre-processing encoding), each chapter presents its own implementation details. Additionally, each chapter gives an introduction to the problem and presents its own results and conclusions.

Finally, Chapter 6 gives a general overview of the contributions presented in this thesis, as well as their potential applications and directions for future work.

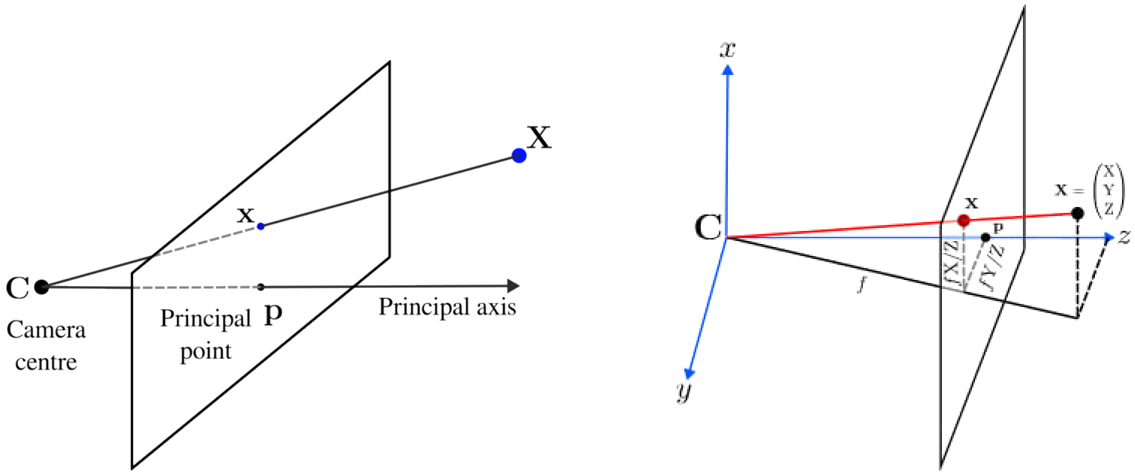
CHAPTER 2

Literature Review

In this chapter, the background theory and recent prior work related to this thesis are revisited. First, the theory for multi-view geometry is introduced in Section 2.1. Next, a review of the techniques for object detection is discussed in Section 2.2. An overview of the current approaches and datasets for multi-view object detection is explored in Section 2.2.8. Finally, Section 2.3 presents the formulation and applications of Neural Radiance Fields.

2.1 Multi-view Geometry

The contributions presented in this thesis rely on multiple views of the same scene, either for object detection or novel view synthesis. The projective nature of cameras imposes some constraints on the geometry of multiple views, giving way to special relations between image point correspondences. This section presents the mathematical framework that describes multi-view geometry. First, the most common camera models are presented in Section 2.1.1. Subsequently, the fundamental matrix, a mathematical object that describes the geometry of two views, and some techniques to estimate it are discussed in Sections 2.1.2 and 2.1.3. Although this is



(a)

(b)

Figure 2.1: Pinhole camera model.

enough for the methods described in Chapter 3, a brief review of the geometry of more than two views is given in Section 2.1.4. Most of the theory discussed in this section is described by Hartley and Zisserman [51].

2.1.1 Camera Models

Given a point in the space $\mathbf{X} = (\mathrm{X}, \mathrm{Y}, \mathrm{Z})^\top$, a camera is a mapping $g : \mathbb{R}^3 \rightarrow \mathbb{R}^2$ that takes $\mathbf{X}$ to an image plane $\mathbf{x} = (x, y)^\top$. We start by considering a central projection of the world to the image plane. The centre of projection $\mathbf{C}$ is called the camera centre, the line that passes through the camera centre and is perpendicular to the image plane is the principal ray and its intersection with the image plane is the principal point $\mathbf{p}$ (Fig. 2.1a). If the camera centre is at the origin and the plane of projection is $\mathrm{Z} = f$ (Fig. 2.1b), then the mapping can be written as

$$(\mathrm{X}, \mathrm{Y}, \mathrm{Z})^\top \rightarrow (f\mathrm{X}/\mathrm{Z}, f\mathrm{Y}/\mathrm{Z})^\top . \tag{2.1}$$

The value f is called the focal length. This is known as the pinhole camera model and can be used to model charge-coupled device (CCD) cameras or transmission imagery (such as X-ray).

Before continuing our analysis, the homogeneous coordinate system used in projective geometry is introduced. Homogeneous coordinates add an extra coordinate to the space and image points, having then $\mathbf{X} = (\mathrm{X}, \mathrm{Y}, \mathrm{Z}, \mathrm{W})^\top$ and $\mathbf{x} = (x, y, w)^\top$. This new representation forms a new space, called the Projective Space $\mathbb{P}^n$. A space $\mathbb{P}^n$ extends the Euclidean space $\mathbb{R}^n$ by adding an extra coordinate, meaning that $\mathbb{P}^2$ has three coordinates and $\mathbb{P}^3$ four. A point in homogeneous coordinates can be transformed back to Euclidean by dividing all the coordinates by the last value, such that $\tilde{\mathbf{x}} = (x/w, y/w)^\top$. In this sense, a point $\tilde{\mathbf{x}} = (x, y)^\top$ can be represented in homogeneous coordinates as $\mathbf{x} = (x, y, 1)^\top$ or $\mathbf{x} = (2x, 2y, 2)^\top$. In general, a point $\mathbf{x} = (kx, ky, k)^\top$, with $k \neq 0$, in homogeneous coordinates represents the same point $\tilde{\mathbf{x}} = (x, y)^\top$ in Euclidean coordinates. Homogeneous coordinates with $k = 0$ represent points at infinity, but their discussion is beyond the scope of this thesis[1]. For the rest of this chapter, Euclidean coordinates are also called inhomogeneous and are represented with a tilde, such as $\tilde{\mathbf{x}}$.

By using homogeneous coordinates, the camera becomes a mapping $g : \mathbb{P}^3 \to \mathbb{P}^2$. Therefore, Eq. (2.1) can be written as:

$$\mathbf{x} = \begin{pmatrix} f\mathrm{X} \\ f\mathrm{Y} \\ \mathrm{Z} \end{pmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} \mathrm{X} \\ \mathrm{Y} \\ \mathrm{Z} \\ 1 \end{pmatrix} = \mathrm{diag}(f, f, 1)\,[I|\mathbf{0}]\,\mathbf{X}\,. \tag{2.2}$$

Eq. (2.1) assumes that the origin of the image plane coincides with the principal point, which may not be the case. Considering an arbitrary principal point $\mathbf{p} = (p_x, p_y)^\top$, Eq. (2.2) becomes:

$$\mathbf{x} = \begin{pmatrix} f\mathrm{X} + \mathrm{Z}p_x \\ f\mathrm{Y} + \mathrm{Z}p_y \\ \mathrm{Z} \end{pmatrix} = \begin{bmatrix} f & 0 & p_x & 0 \\ 0 & f & p_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} \mathrm{X} \\ \mathrm{Y} \\ \mathrm{Z} \end{pmatrix} = K\,[I|\mathbf{0}]\,\mathbf{X}\,. \tag{2.3}$$

The matrix K is called the camera calibration matrix or the intrinsic parameters and the process to obtain K is called calibration.

[1] For a more extensive introduction to projective geometry, refer to Chapters 2 and 3 of Hartley and Zisserman [51].

The pinhole model described in Eq. (2.3) assumes that the image coordinates are Euclidean. However, CCD cameras might not have square pixels, meaning that the focal distance is different for the x and y directions. Considering these focal lengths f_x and f_y, and an additional parameter s called the skew (which is normally 0 except for some unusual cameras), the general CCD camera model is given by:

$$K = \begin{pmatrix} f_x & s & p_x \\ 0 & f_y & p_y \\ 0 & 0 & 1\,. \end{pmatrix} \tag{2.4}$$

The camera matrix in Eq. (2.4) is known as a finite projective camera.

So far, our analysis places the principal axis at the z axis and the camera centre at the origin. If the principal axis is rotated by a rotation matrix R and the camera centre is not at the origin, we can express a space point in inhomogeneous coordinates in the camera axis frame as $\tilde{\mathbf{X}}' = R(\tilde{\mathbf{X}} - \tilde{\mathbf{C}})$. This can be expressed in homogeneous coordinates as:

$$\mathbf{X}' = \begin{pmatrix} R & -R\tilde{C} \\ \mathbf{0}^\top & 1 \end{pmatrix} \mathbf{X}\,. \tag{2.5}$$

Substituting Eq. (2.5) in Eq. (2.3) gives:

$$\mathbf{x} = K\,[R|\mathbf{t}]\,\mathbf{X} = P\mathbf{X}\,, \tag{2.6}$$

where $\mathbf{t} = -R\tilde{\mathbf{C}}$ is the camera centre position vector in the world coordinates frame. The 3×4 matrix P is known as the camera projection matrix. It is divided into the intrinsic parameters K and the extrinsic parameters $[R|\mathbf{t}]$, which define the position of the camera. A further generalization can be made by considering an arbitrary 3×4 projection matrix P, called a general projective camera, but this is not considered in this thesis.

2.1.2 Epipolar Geometry and the Fundamental Matrix

In this section, we describe the geometry of two views, namely the epipolar geometry, and some entities that arise in this context. Consider two cameras with centres at

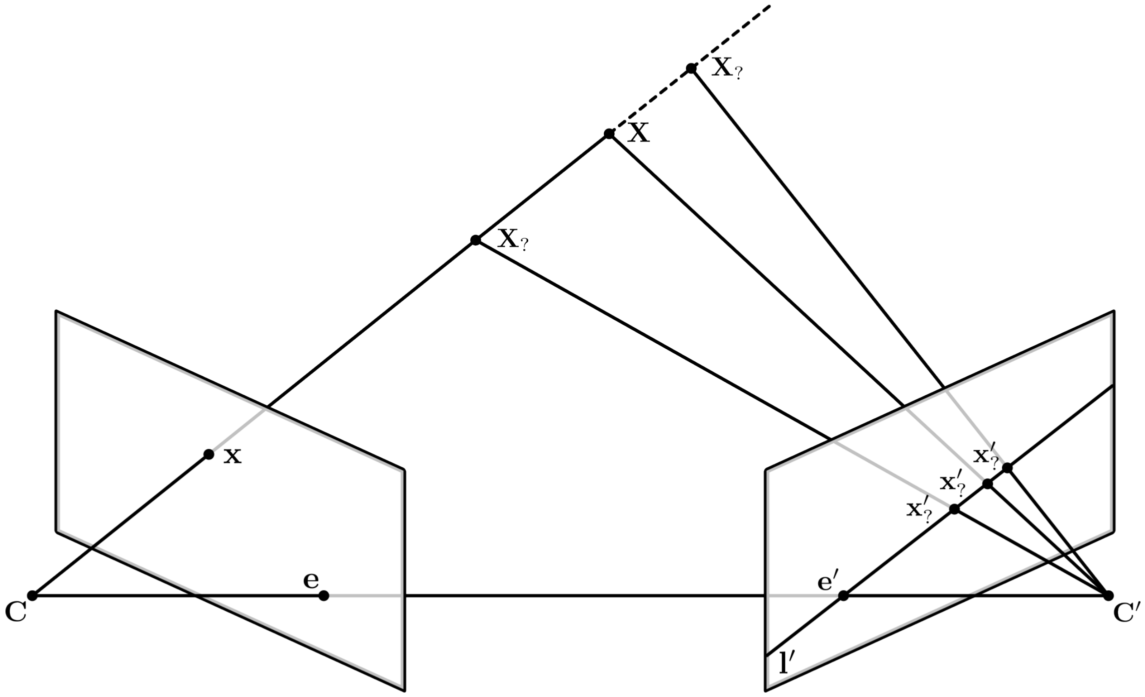


Figure 2.2: Epipolar ambiguity. A point $\mathbf{x}$ maps to an epipolar line $\mathbf{l}'$ in another view, where all possible mappings $(\mathbf{X}_? \rightarrow \mathbf{x}'_?)$ of the true spatial point $\mathbf{X}$ lie.

$\mathbf{C}$, $\mathbf{C}'$, projection matrices P, P' and image planes I, I$'$. If a point in space $\mathbf{X}$ is mapped to two points $\mathbf{x}$ and $\mathbf{x}'$ in both views, it is said that these are corresponding points and it is expressed as $\mathbf{x} \leftrightarrow \mathbf{x}'$. However, if only the point in the first image $\mathbf{x}$ is known, then a map from this point to a line[2] $\mathbf{x} \mapsto \mathbf{l}'$ in the second view, known as the epipolar line, can be obtained. The unknown point $\mathbf{x}'$ lies on the epipolar line. This is a property of the epipolar geometry because of the ambiguity that any point lying in the ray described by the camera centre and an image point will be mapped to the same point in the image plane, as seen in Fig. 2.2. The mapping of $\mathbf{C}$ in the second image plane is known as the epipole $\mathbf{e}'$ and is contained in all the epipolar lines from any point in the first view. Similarly, the camera centre $\mathbf{C}'$ is mapped to the first view as the epipole $\mathbf{e}$.

In order to derive a formula to obtain the epipolar line $\mathbf{l}'$, we need two points lying in the line. We already know that the epipole $\mathbf{e}'$ lies on $\mathbf{l}'$. Now we need to find another line in the ray defined by C and $\mathbf{x}$ and map it to the second image I$'$.

[2] For this analysis, a line is represented with a vector $\mathbf{l} = (a, b, c)^\top$ such that a point in homogeneous coordinates $\mathbf{x} = (x, y, 1)^\top$ lying in the line satisfies $\mathbf{l}^\top \mathbf{x} = ax + by + c = 0$

Given that $\mathbf{x} = P\mathbf{X}$, the point $P^{+}\mathbf{x}$ lies on the ray, where $P^{+} = P^{\top}(PP^{\top})^{-1}$ is the Moore-Penrose inverse, such that $PP^{+} = I$. This is easily verified by noting that $P^{+}\mathbf{x}$ projects back to $\mathbf{x}$, since $P(P^{+}\mathbf{x}) = (PP^{+})\mathbf{x} = I\mathbf{x} = \mathbf{x}$. Projecting this point in the second image, we obtain the second point in the epipolar line $P'P^{+}\mathbf{x}$. The epipolar line is then defined as the cross-product of this point and the epipole $\mathbf{e}'$:

$$\mathbf{l}' = \mathbf{e}' \times (P'P^{+}\mathbf{x}) = [\mathbf{e}']_{\times} P'P^{+}\mathbf{x} = F\mathbf{x}\,, \tag{2.7}$$

where $[\cdot]_{\times}$ is the skew-symmetric representation of a vector used to write cross products as matrix multiplications, such that if $\mathbf{a} = (a_1, a_2, a_3)^{\top}$, then

$$[\mathbf{a}]_{\times} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} . \tag{2.8}$$

The matrix F in Eq. (2.7) is a rank 2 matrix that defines the geometry of two views and is known as the fundamental matrix. If the camera projection matrices are known, F can be directly obtained. However, if the cameras are uncalibrated (meaning that P and P' are not known), then it can be derived using point correspondences. This is explored in Section 2.1.3.

If F is the fundamental matrix of two cameras P, P', then $F^{\top}$ is the fundamental matrix in the opposite direction P', P. Similarly, a point in the second view defines an epipolar line in the first view as $\mathbf{l} = F^{\top}\mathbf{x}'$. Finally, the fundamental matrix satisfies:

$$\mathbf{x}'^{\top} F\mathbf{x} = 0\,, \tag{2.9}$$

for any point correspondence $\mathbf{x} \leftrightarrow \mathbf{x}'$. Eq. (2.9) holds because $\mathbf{x}'$ lies on $\mathbf{l}'$, resulting in $\mathbf{x}'^{\top} F\mathbf{x} = \mathbf{x}'^{\top}\mathbf{l}' = 0$. It is also observed that since the epipoles $\mathbf{e}$' and $\mathbf{e}$ lie in their corresponding epipolar lines $\mathbf{l}' = F\mathbf{x}$ and $\mathbf{l} = F^{\top}\mathbf{x}'$, then $(\mathbf{e}'^{\top} F)\mathbf{x} = \mathbf{x}'^{\top}(F\mathbf{e}) = 0$. As the previous relationship holds for any points $\mathbf{x}$ and $\mathbf{x}'$, then $\mathbf{e}'^{\top} F = F\mathbf{e} = 0$, meaning that $\mathbf{e}'$ and $\mathbf{e}$ are the left and right null spaces of F.

2.1.3 Fundamental Matrix Estimation

It is seen in Eq. (2.7) that the fundamental matrix F can be obtained if the cameras are calibrated (*i.e.*, the projection matrices are known). If this is not the case, the fundamental matrix can be computed with pure point correspondences $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$. This section details the normalized 8-point algorithm for computing the fundamental matrix, including the enforcement of its rank 2 property. A more precise iterative solution is also described.

The expansion of the relation in Eq. (2.9) gives the equation:

$$x'xf_{11} + x'yf_{12} + x'f_{13} + y'xf_{21} + y'yf_{22} + y'f_{23} + xf_{31} + yf_{32} + f_{33} = 0 \qquad (2.10)$$

where f_{ij} are the entries of the fundamental matrix. For n correspondences, the following system of equations is obtained:

$$A\mathbf{f} = \begin{bmatrix} x'_1x_1 & x'_1y_1 & x'_1 & y'_1x_1 & y'_1y_1 & y'_1 & x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \\ x'_ix_i & x'_iy_i & x'_i & y'_ix_i & y'_iy_i & y'_i & x_i & y_i & 1 \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{12} \\ \vdots \\ f_{33} \end{bmatrix} = 0\,. \qquad (2.11)$$

Eq. (2.11) is a homogeneous system of equations. F can only be determined up to scale (because of the nature of the projective geometry), so we enforce $\|\mathbf{f}\| = 1$ to get a unique solution. In this sense, Eq. (2.11) is only solved for 8 values of $\mathbf{f}$, and the remaining value is found with the unit norm condition. As a result, a non-trivial solution exists for $\mathbf{f}$ if and only if A is at most rank 8. If the rank of A is 9, because of noise correspondences, then a least-squares solution is found by minimizing $\|A\mathbf{f}\|$ subject to $\|\mathbf{f}\| = 1$. Solving a system defined only up to scale, such as the computation of the fundamental matrix, is known as Direct Linear Transformation (DLT).

An important characteristic of the fundamental matrix is its rank 2 property. Generally, the fundamental matrix $\bar{F}$ obtained from Eq. (2.11) results in a non-singular matrix. The rank 2 matrix F most similar to $\bar{F}$ is found by using the singular value decomposition (SVD) of $\bar{F}$ and replacing the smallest singular value of the diagonal matrix with 0, *i.e.*, if the SVD of $\bar{F}$ is $U\text{diag}(s_1, s_2, s_3)V^\top$, $s_1 \geq s_2 \geq s_3$, then $F = U\text{diag}(s_1, s_2, 0)V^\top$.

The normalized 8-point algorithm

The terms of the matrix A in Eq. (2.11) are quadratic, linear or 1. Since the coordinates of the images are in the order of 10^2, a slight variation in one of the coordinates will greatly affect the quadratic terms, while the linear and constant terms will not see a big variation. To avoid this issue in the computation of $\mathbf{f}$, a normalization transformation is carried out. This transformation T scales the coordinates such that their distance to the centre of the image is $\sqrt{2}$ in average, meaning that the average point is $\tilde{\mathbf{x}} = (1, 1)^\top$. Additionally, the points are translated such that the origin of the coordinate system is at the centre of the image.

The normalized 8-point algorithm consists on getting the normalizing transformations $\hat{\mathbf{x}} = T\mathbf{x}$ and $\hat{\mathbf{x}}' = T'\mathbf{x}'$ for each view independently. Subsequently, a fundamental matrix $\hat{F}$ is obtained using the correspondences $\hat{\mathbf{x}}_i \leftrightarrow \hat{\mathbf{x}}'_i$ via DLT and enforcing the rank-2 constraint. Substituting these transforms in Eq. (2.9), the resulting fundamental matrix F from the original point correspondences is found as follows:

$$\begin{aligned} \hat{\mathbf{x}}'^\top \hat{F} \hat{\mathbf{x}} &= 0 \\ (T'\mathbf{x}')^\top \hat{F}(T\mathbf{x}) &= 0 \\ (\mathbf{x}'^\top T'^\top)\hat{F}(T\mathbf{x}) &= 0 \\ \mathbf{x}'^\top (T'^\top \hat{F} T)\mathbf{x} &= 0\,. \end{aligned} \tag{2.12}$$

It is observed from Eq. (2.12) that $F = T'^\top \hat{F} T$ for the original point correspondences $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$.

Minimization of the algebraic error

An alternative approach is to directly solve Eq. (2.11) subject to $\|\mathbf{f}\| = 1$ for a singular matrix F. This cannot be done linearly since $\det F = 0$ is a cubic equation. However, an iterative linear approach can be used based on minimizing the algebraic error $\|A\mathbf{f}\|$ for a singular matrix $\mathbf{f}$.

Given that $\mathbf{e}$ is the right null space of F, the fundamental matrix can be decomposed as $F = M[\mathbf{e}]_\times$ for an arbitrary non-singular matrix M (noting that $F\mathbf{e} = M[\mathbf{e}]_\times\mathbf{e} = M(\mathbf{e} \times \mathbf{e}) = 0$). By writing the F and M matrices as vectors $\mathbf{f}$ and $\mathbf{m}$, we can write $\mathbf{f} = E\mathbf{m}$, where:

$$E = \begin{bmatrix} [\mathbf{e}]_\times & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & [\mathbf{e}]_\times & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & [\mathbf{e}]_\times \end{bmatrix} . \tag{2.13}$$

Subsequently, the fundamental matrix estimation becomes to minimize $\|AE\mathbf{m}\|$ subject to $\|E\mathbf{m}\| = 1$.

The minimization of the algebraic error for the singular matrix F is an iterative approach that is based on a previous value for the epipole $\mathbf{e}$, which itself comes from a previous fundamental matrix. Once a new fundamental matrix F_i is calculated by minimizing $\|AE_{i-1}\mathbf{m}_i\|$, the new epipole $\mathbf{e}_i$ can be used to estimate a new fundamental matrix F_{i+1}. An initial estimate of the fundamental matrix F_0 can be obtained by using the normalized 8-point algorithm. The full process of estimating the fundamental matrix is summarized in Algorithm 1.

Other methods to estimate the fundamental matrix include the minimization of the reprojection error or the first-order geometric error, which minimizes the distance of a point to the reprojected epipolar line. In this thesis, the fundamental matrix is estimated using the algebraic error in Chapter 3, since it is found to be accurate for the data.

2.1.4 Multi-view geometry for more than two views

When more than two views are present for a single scene, there are other techniques and mathematical entities that describe the underlying geometry. For instance, the trifocal tensor [52, 53] describes the underlying three-view geometry, analogous to the fundamental matrix for two views. The trifocal tensor enables point transferring (*i.e.*, the exact location of a point in another view) from point correspondences in

Algorithm 1: Fundamental matrix estimation by minimizing the algebraic error

```
Data: Point Correspondences $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i, i = 1, \dots, n$ in the matrix format in
      Eq. (2.11) and the number $N$ of iterations for the minimization of
      the algebraic error.
Result: The fundamental matrix $F$
begin
    Point Normalization
    /* Normalize point correspondences to have their centre at
       the centroid of the image and average distance to the
       centre $\sqrt{2}$                                          */
    $\hat{\mathbf{x}} \leftarrow T\mathbf{x}$
    $\hat{\mathbf{x}}' \leftarrow T'\mathbf{x}'$

    DLT
    $\hat{\mathbf{f}}_0 \leftarrow \arg\min_{\hat{\mathbf{f}}} \|A\hat{\mathbf{f}}\|, \text{ s.t. } \|\hat{\mathbf{f}}\| = 1$
    $\hat{F}_0 \leftarrow \hat{\mathbf{f}}_0$
    $\bar{F}_0 \leftarrow T'^{\top}\hat{F}_0 T$

    Singularity enforcement
    $U\text{diag}(s_1, s_2, s_3)V^{\top} \leftarrow \bar{F}_0, \; s_1 \geq s_2 \geq s_3$
    $F_0 \leftarrow U\text{diag}(s_1, s_2, 0)V^{\top}$

    Minimization of the algebraic error
    $\mathbf{e} \leftarrow \text{RightNullVector}(F_0)$
    for $i \leftarrow 1$ to $N$ do
        $E \leftarrow \mathbf{e}$ /* From Eq. (2.13)                      */
        $\mathbf{m} \leftarrow \arg\min_{\mathbf{m}} \|AE\mathbf{m}\|, \text{ s.t. } \|E\mathbf{m}\| = 1$
        $M \leftarrow \mathbf{m}$ /* From vector to matrix representation */
        $F_i \leftarrow M[\mathbf{e}]_{\times}$
        $\mathbf{e} \leftarrow \text{RightNullVector}(F_i)$
Output: $F_N$
```

two views to the third view. Additionally, lines can also be transferred similarly to points, which cannot be done with only two views. A quadrifocal tensor [54] can also be formed for a four-views geometry with similar linear relationships. No more tensor representations have been found for more than four views [51], although Vidal and Abretske [55] have found that multi-linear relations hold for non-rigid shapes for an arbitrary number of views. Tomasi and Kanade [56] describe the factorization algorithm, which allows 3D reconstruction for affine cameras for four or more points correspondences over m views. Although the methods described in Chapters 3 and 4 are based on two-view correspondences, an extension could be implemented considering the relations for more than two views introduced in this section.

2.2 Object Detection

This section gives an introduction to object detection in computer vision. Current trends in object detection architectures and datasets are reviewed. The literature presented in this section is the base of the methods for Chapters 3 and 4.

2.2.1 Introduction

Automatic object detection is a contemporary task in computer vision. It aims at the localization of objects of interest within an image as it is usually presented by (axis-aligned) bounding boxes. If more than one type of object is aimed to be detected, a class is also associated with each bounding box.

Formally, given a set $\mathcal{C}$ with M categories, object detection can be defined as the task of finding a set $B = \{\mathbf{b}_i\}_1^N$ of N bounding boxes $\mathbf{b}_i \in \mathbb{R}^4 \times \mathcal{C}$ surrounding the objects of interest within an image $I \in \mathbb{R}^{W \times H \times d}$, where W and H are the width and height of the image and d is the number of channels (*e.g.*, 3 for RGB images). Each bounding box $\mathbf{b}_i$ is an ordered pair of a 4D vector with the dimension values (two values for the coordinates of its centre and two for its dimensions or, equivalently, four values specifying two opposite corners) and one category $c_j \in \mathcal{C}$. In this sense, an object detector is a function $\phi : \mathbb{R}^{W \times H \times d} \to \mathbb{R}^{N' \times 4} \times \mathcal{C}^{N'}$ that predicts a set $\hat{B}$ of N' bounding boxes with corresponding predicted categories $\hat{\mathbf{c}}$ within the image I. Modern detectors also include a confidence score $s_j \in [0, 1]$ of belonging to each category c_j, such that $\sum_{j=1}^{M} s_j = 1$, where closer values of s_j to 1 indicates greater confidence of belonging to the j-th class. This results in bounding boxes being represented by $\mathbf{b}_i \in \mathbb{R}^4 \times [0, 1]^M$, including the confidence to belong to each class. In practice, the quality of the predicted bounding boxes $\hat{B}$ is measured according to a metric based on the overlapping with the ground truth set B. A discussion of such metrics is given in Section 2.2.3.

Traditional object detectors are based on the matching of highly engineered features, such as SIFT [57], SURF [58] and Histogram of Gradients (HoG) [59]. These detectors reached a peak in performance with the Deformable Part-Based Model (DPM) [60], which is based on the learning (using a Support Vector Machine) of deformable parts of objects described by a pyramid of HoG features. However, given the higher availability and the reduction of costs of graphics processing units (GPUs) in the early 2010s, computer vision entered a new era of deep learning-based architectures motivated by AlexNet [1], a convolutional neural network (CNN) used for image classification. Since then, numerous milestones have been achieved in deep learning methods for computer vision, such as better CNN architectures [45, 61–64], optimization techniques [65], neural activation functions [66–68] and loss functions [6, 69, 70]. This advent of deep learning methods was first brought to the object detection task by Girshick *et al.* [71] with the introduction of Regions with CNN features (R-CNN). Subsequent improvements in accuracy [2, 72–74] and efficiency [3–5, 75, 76] have resulted in real-time high fidelity object detectors. The general architecture and types of deep learning-based object detectors are discussed in Sections 2.2.4 to 2.2.7. A comprehensive review of the evolution of object detection is given by Zou *et al.* [77].

2.2.2 Detection Datasets

In contrast with image classification, detection datasets require object-level annotations. Depending on the learning approach, different levels of annotations can be given. For supervised learning algorithms, the location and class of the ground truth bounding boxes are provided for each image. In crowded instances with substantial objects per scene, the annotation process may take a considerable amount of time and human resources. Considering this, many research groups have made a great effort to create these datasets. A different approach to overcome this issue is to use unsupervised and weakly supervised learning models which do not use object location. Given that all the contributions reported in this thesis are based on supervised techniques, only datasets that are fully annotated are reviewed in this section. The review is divided into general and task-specific datasets.

General Large-Scale Object Detection Datasets

General detection datasets are designed to test the capabilities of detectors under no specific constraint. These datasets need a large body of images and annotations to allow the detector to learn their general features. Before the deep learning era, the PASCAL Visual Object Classes (VOC) Challenge [78] published a series of datasets for object detection from 2005 to 2012. In their latest edition, the PASCAL VOC 2012 detection dataset consisted of 11,540 images with 27,450 annotated instances among 20 different classes, including persons, animals, automobiles and a variety of objects. Following the PASCAL VOC challenge, the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [79] ran from 2010 to 2017, publishing more extensive datasets for the classification, single-instance localisation and object detection tasks. For the classification task, a large dataset consisting of more than 1.2 million images and 1,000 classes was released. To this date, the classification dataset (referred to as simply the ImageNet dataset) is still used to evaluate modern image classification models [7, 80–83], and it is commonly used for several visual recognition tasks [84–87]. The ILSVRC detection dataset consists of more than 470,000 images with approximately 530,000 annotated objects spanning 200 classes. The Microsoft Common Objects in Context (MS-COCO) dataset [88] is arguably the most used dataset for general object detection, consisting of 2.5 million annotated instances, 328,000 images and 91 classes. This dataset aims to cover natural images following the real-world distribution with the bounding box, mask and id annotations for each object in the image. At present, it has become the *de facto* dataset for state-of-the-art detectors [7, 8, 81, 89–91]. It is also used for pre-training detectors for different domains with smaller datasets. The MS-COCO dataset also introduced a new set of evaluation metrics, which has also become the standard way of reporting the performance of detectors. This is further discussed in Section 2.2.3. In recent years, larger and more sophisticated datasets for general and natural object detection have been released. For instance, the LVIS dataset [92] uses the Zipfian distribution (*i.e.*, an inverse law for the appearance of categories in natural images) to create a new annotation dataset for the MS-COCO dataset, with a current size of approximately 160,000 images and 2 million instance annotations (1,203 classes).

The Objects365 dataset [93] is another large-scale dataset that contains more than 10 million bounding boxes over 365 categories and more than 600,000 images for training. To date, the largest (original) dataset with object localisation level annotations is Open Images v7 [94], with 16 million bounding boxes, 600 classes and 1.9 million images. The BigDetection dataset [95] comprises an ensemble of the LVIS, Objects365 and Open Images datasets with careful design principles in order to homogenize the data categories and annotations. In total, the BigDetection dataset consists of 3.4 million training images, 36 million instance annotations and 600 classes. The release of these datasets opens new challenges for modern detectors to account for multi-instance object detection or long-tail categorical distributions. In addition, these datasets may be used for domain adaption in detection tasks with fewer annotations.

Application Specific Detection Datasets

Several detection datasets are used for training object detectors based on the application, including autonomous driving, pedestrian detection or medical analysis. A brief review of such datasets is presented next.

Autonomous Driving Datasets. With the recent technological advances that have allowed the rapid development of self-driving cars, several related detection datasets have been published [96–98]. These datasets often include visual annotations for different sensors (*e.g.*, infrared or LiDAR) for classes that are relevant to this context, such as pedestrians, cars, bicycles, or signs. More challenging datasets for low-level vision (dark environments or problematic weather conditions) are also available [99–101]. Also, some datasets include paired multi-view annotations for stereo-vision applications, such as the KITTI dataset [96].

Pedestrian detection. These datasets have mainly one class only, with crowded instances being common. Some examples include the ETH Pedestrian [102], CrowdHuman [103], WiderPerson [104], and EuroCity [105]. A pedestrian dataset that is of particular interest in this thesis is the Wildtrack dataset [106], having 7 different viewpoints. This and similar datasets are reviewed in Section 2.2.8.

Aerial Detection. Another application of object detection is the identification of objects from aerial images, usually taken from drones [107, 108] or satellites [109–111]. The objects within the dataset images are small and usually crowded, imposing a challenging task. In some instances, the bounding boxes are exceptionally tiny such that tailored detectors have to be created.

Medical Image Analysis. Detection datasets for medical applications are usually very difficult to release because of privacy reasons. Moreover, as the annotations need to be performed by medical specialists, these datasets are usually small with not many instances per class. This, however, can be alleviated by prior knowledge of the human anatomy. Some of the few public datasets are for polyp detection [112], glaucoma assessment [113], lesion detection [114] and chest diseases detection [115].

X-ray Security Imagery. An application that is considered in this thesis is the recognition of objects within X-ray security images. These images are generally formed by dual-energy scanners, which use two energy bands to get the material composition of the scanned item. The combination of these two energies produces a pseudo-colour image that assigns a colour to each material category. In this regard, Appendix A gives an insight into the formation of these images, along with an architecture for cross-modality generation. SIXRay [116] is the largest publicly available dataset, with over 1 million images from subway stations spanning six categories of prohibited items. The GDXRay+ [117] baggage dataset contains more than 8,000 multi-view bag images for security purposes. However, these dataset images are not heavily cluttered, making them difficult to use for real-life applications. Similarly, the COMPASS-XP [118] dataset has X-ray images of prohibited items in uncluttered scenes. OPIXray [119] is a single-view dataset of prohibited items identified by professional inspectors from airports, including more than 8,800 annotated images. Finally, the Durham Bag Full Image Dataset (DBF6) [120] comprises a four-view dataset of 6 classes of manually annotated objects from dual-energy X-ray airport security screenings. The DBF6 is further explored in Section 2.2.8. These datasets exhibit a different object distribution than objects in natural images of general object detection datasets. A comprehensive statistical analysis of these differences is discussed in Appendix B.

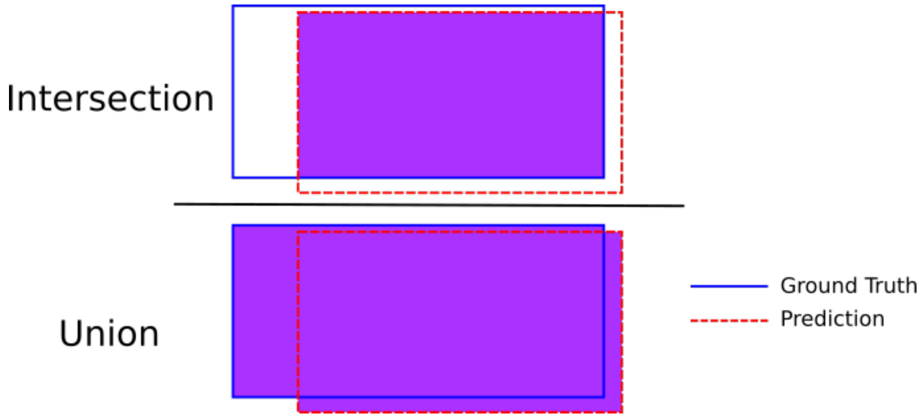


Figure 2.3: Intersection over Union.

2.2.3 Evaluation Metrics

In order to measure the performance of a detector, a definition of the matching of a predicted bounding box $\hat{\mathbf{b}}$ and a ground truth bounding box $\mathbf{b}$ is needed. In practice, the Intersection over Union (IoU) has been found to be a reliable matching measure. The IoU is defined as the area of the intersection of the prediction and the ground truth divided by the area of their union. This can be written as:

$$\mathrm{IoU}(\mathbf{b}, \hat{\mathbf{b}}) = \frac{\mathrm{area}(\mathbf{b} \cap \hat{\mathbf{b}})}{\mathrm{area}(\mathbf{b} \cup \hat{\mathbf{b}})}\,. \tag{2.14}$$

The illustration of the IoU is shown in Fig. 2.3. If the IoU is greater than a threshold value, it is said that the prediction is correct. A common threshold value is 0.5, where such a small value accounts for the noise during the annotation process [78]. Now that a method for defining matching pairs is defined, similar metrics as in binary classification can be used. These include the true positives (the number of correct predictions), false positives (predictions that have no matching ground truth) and false negatives (missed ground truth bounding boxes). Notice that in this framework, true negatives have no meaning. In this context, the precision P is defined as the ratio of true positives from all the predicted bounding boxes, whilst the recall R is the ratio of true positives and ground truth boxes.

Since the publication of the PASCAL VOC challenge [78], the measurement of performance for object detection has been based on the average precision (AP) calculated from the precision/recall (PR) curve. Each bounding box $\hat{\mathbf{b}}_i$ is predicted

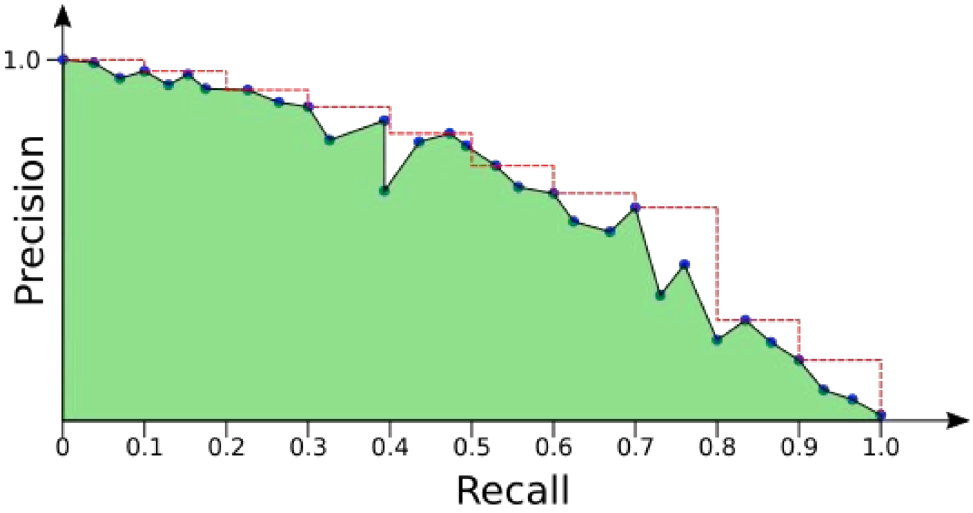


Figure 2.4: PR curve used for AP calculation. The red dotted lines represent the interpolated version of the PR-Curve, whilst the area under the curve (in green) is the AP from the 2010 PASCAL VOC challenge [121].

with a confidence score $s_i \in [0, 1]$, where $s = 1$ indicates maximum confidence. A bounding box is considered a prediction if its confidence score is greater than a threshold value s_t. By varying s_t from 0 to 1, different pairs of precision and recall (p, r) are obtained. Subsequently, the PR curve is constructed with the recall on the horizontal axis and their corresponding precision on the vertical one. Starting from the 2010 PASCAL VOC challenge [121], the AP is defined as the exact area under the PR curve:

$$\mathrm{AP} = \int_0^1 p(r)dr\,, \tag{2.15}$$

as seen in Fig. 2.4. If more than one class is present in the detection task, then the final metric is the average of the AP from each class. This value is known as the mean average precision (mAP) and for K classes is given by:

$$\mathrm{mAP} = \frac{1}{K}\sum_{k=1}^{K} \mathrm{AP}_k\,, \tag{2.16}$$

where AP_k is the AP for the k-th class. For PASCAL VOC, these metrics are calculated using an IoU threshold of 0.5.

The MS-COCO challenge [88] introduced a new set of metrics for object detection. These are based on the mAP metric in PASCAL VOC and are summarized in Table 2.1. The mAP in the MS-COCO metrics is simply called AP and there is no difference between AP and mAP (*i.e.*, the AP is always averaged among the classes). The main metric for the MS-COCO dataset is an average of APs at different IoU thresholds, specifically, 10 values from 0.5 to 0.95 in increments of 0.05. This metric is simply called AP in the MS-COCO framework. The AP with a fixed IoU at 0.5, called $\text{AP}_{0.5}$ is also used, which is identical to the PASCAL VOC AP. Similarly, a stricter metric $\text{AP}_{0.75}$ with an IoU threshold of 0.75 is also reported. Further AP metrics (averaged over the 10 IoU values as the main metric) are reported for small (area $\leq 32^2$ pixels), medium (32^2 pixels $<$ area $\leq 96^2$ pixels) and large (area $> 96^2$ pixels) objects, namely AP_S, AP_M and AP_L. In addition, the MS-COCO challenge includes the average recall (AR) between 0.5 and 1 IoU thresholds. This can be calculated as [122]:

$$\text{AR} = 2\int_{0.5}^{1} \text{Recall}(\tau)d\tau\,, \tag{2.17}$$

where $\text{Recall}(\tau)$ is the recall at an IoU threshold τ. Similarly, the mean AR considering all the classes is simply reported as the AR. The AR is reported for a maximum of 1, 10 and 100 detections per image (AR_1, AR_{10} and AR_{100}) and for small, medium and large objects (AR_S, AR_M and AR_L).

Although new metrics have been introduced, such as the variation of the AP in PASCAL VOC by the Open Images challenge [94], the MS-COCO metrics are still the most used and accepted for object detection and hence they are the ones used in this thesis.

2.2.4 Deep Learning-based Object Detection

This section revisits the general architecture, data augmentation, post-processing techniques and loss functions for deep learning-based detectors. Depending on their processing pipeline, modern object detectors can be classified into two-stage or one-stage detectors. Two-stage detectors first obtain a set of proposal object candidates which are then classified as belonging to one of the training classes or background. On the other hand, one-stage detectors are end-to-end networks that locate objects in a forward pass. In general, two-stage detectors are more accurate while one-stage detectors are faster and useful for real-time applications. Two-stage and one-stage

Table 2.1: MS-COCO [88] metrics.

Metric	Description
AP	Average precision at the IoU threshold values from 0.5 to 0.95 in increments of 0.05. This is the main metric.
$\mathrm{AP}_{0.5}$	Average precision for an IoU of 0.5. This is equivalent to the 2010 PASCAL VOC [121] AP.
$\mathrm{AP}_{0.75}$	Average precision for an IoU of 0.75.
AP_{S}	Similar to AP but for small objects (area $\leq 32^2$ pixels).
AP_{M}	Similar to AP but for medium-sized objects (32^2 pixels $<$ area $\leq 96^2$ pixels).
AP_{L}	Similar to AP but for large objects (area $> 96^2$ pixels).
AR_1	Average recall over the range of IoU from 0.5 to 1 [122], for a maximum 1 object per image.
AR_{10}	Average recall for a maximum 10 objects per image
AR_{100}	Average recall for a maximum 100 objects per image
AR_{S}	Average recall for small objects (area $\leq 32^2$ pixels).
AR_{M}	Average recall for medium-sized objects (32^2 pixels $<$ area $\leq 96^2$ pixels).
AR_{L}	Average recall for large objects (area $> 96^2$ pixels).

detectors are discussed in Sections 2.2.5 and 2.2.6.

Detectors Architecture

Modern deep learning-based object detectors share a similar architecture comprising a backbone, a neck and a head (Fig. 2.5). The backbone of an object detector is a feature extractor that is usually a classification network without the last layer (which performs logistic regression). Common choices of backbone include VGG [63], ResNet [61], EfficientNet [4], DenseNet [123], amongst others. The reasons to use a particular backbone are the same as in classification: some networks are more accurate while others are faster without compromising too much upon precision. Generally, these backbones are loaded with pre-trained weights in a large classification dataset, such as ImageNet [79]. Specialized backbones that consider the spatial nature of the detection task have been investigated, such as DetNet [124] or DetNASNet [125]. It is worth noting that having a good backbone may boost the performance of the object detector without further modification of the remaining

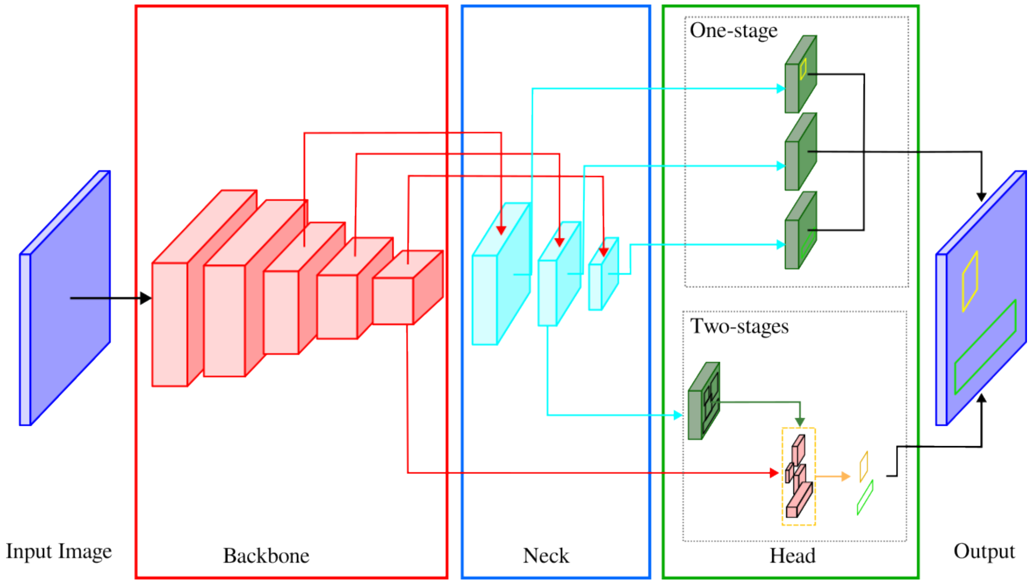


Figure 2.5: Modern deep learning-based object detectors. These include a feature extractor (the backbone), a network to aggregate different spatial resolution layers (the neck) and another network that predicts the bounding boxes (the head). The head can be either two-stage or one-stage.

architecture [7, 125].

The neck of a detector comprises a sub-network of extra layers that aggregate the information from different backbone layers at different spatial resolutions. This started with the development of one-stage detectors such as SSD [5] or YOLO [3], albeit not all detectors include a neck. The neck helps in the prediction of bounding boxes at different scales since very deep layers tend to have a big receptive field that makes it hard to detect small objects. A popular neck architecture is the use of Feature Pyramid Networks (FPN) [74]. FPN creates a feature pyramid that combines upsampled deeper layers with shallower layers with better inherent spatial resolution. Various neck architectures with similar concepts have also been developed [3, 126, 127].

Finally, the head of the detector is a network that performs the detection task with the features (or feature pyramids if using a neck network) to get the bounding boxes. Contrary to the backbone and the neck, the head significantly varies among detectors. Two-stage detectors first predict a set of proposal bounding boxes that are subsequently classified into one of $N+1$ classes (including a no-object background class) and whose geometric parameters are further refined. On the other hand, one-stage detectors directly get a bounding box with an associated class at each spatial location of the feature maps. Regarding the type of head, it may be implemented using anchor boxes, which are predefined rectangles that are used as a reference to regress the final predictions by means of calculating offsets in their location and dimensions. A different family of detectors use a head based on the Transformer architecture [128], which is reviewed in Section 2.2.7.

Some object detectors need a pre-processing step. For instance, YOLO [3] and SSD [5] detectors work with a fixed input size, while some backbones work with image patches, such as Swin Transformers [7].

Although there are some techniques for object detection that do not follow the previous approach, such as neural architecture search [125, 129], or multi-stage detectors [130–133], these are not explored in this work. For a general overview of object detection, the reader is directed to the survey of Zou *et al.* [77].

Data Augmentation for Object Detection

One of the most effective steps to improve detection performance (or in general, any task performance) and reduce overfitting, without negatively impacting the inference processing time, is data augmentation. In terms of image processing, data augmentation is a series of techniques that randomly transform the input images in order to add variability to the input data, virtually increasing the size of the dataset. Data augmentation includes geometric transforms, such as random flipping, rotations or affine transformations. It is important to notice that in this type of transformation, consistent operations have to be applied to the bounding boxes. Other techniques used by early detectors [1, 63] are based on intensity changes such as variation in brightness, contrast or noise addition. Recently, more sophisticated data augmentation techniques have been introduced, further improving object detectors. Random Erasing [134], Cutout [135] and GridMask [136] add random patches to avoid memorization and to improve generalization. Cutmix [137] is similar to patch-based augmentations but the patches are obtained by cutting random sections from the input data, avoiding the loss of information in techniques such as Cutout. Mixup [138] combines a pair of images to encourage a linear behaviour between instances. The Mosaic augmentation [139] mixes four images, allowing the detection of objects outside their normal context. Comprehensive reviews of data augmentation techniques for computer vision and object detection are given by Shorten and Khoshgoftaar [140] and Kaur *et al.* [141].

Post-processing

A common issue for object detectors is that they usually predict many similar overlapping boxes that refer to the same ground truth object. This is caused because many detectors predict the bounding boxes densely or with a sliding window (implemented via a convolution), and hence have independent predictions for each spatial location within the feature maps. The main post-processing technique to overcome this issue is non-maximum suppression [71] (NMS), which involves the removal of bounding boxes overlapping a same-class box with a better confidence score. Given a set of same-class bounding boxes $B = \{\mathbf{b}_i\}$ with associated confidence score $S = \{s_i\}$, and a bounding box $\mathbf{b}_{\max}$ with confidence $s_{\max}$ such that $\forall s_i \in S, s_{\max} \geq s_i$, then NMS assigns new confidence scores s'_i as follows:

$$s'_i = \begin{cases} s_i, & \mathrm{IoU}(\mathbf{b}_i, \mathbf{b}_{\max}) < \tau \\ 0, & \mathrm{IoU}(\mathbf{b}_i, \mathbf{b}_{\max}) \geq \tau \end{cases}, \tag{2.18}$$

where τ is an IoU threshold (usually set to 0.5). NMS, depicted in Fig. 2.6a, is effective at eliminating duplicated detections but it also eliminates valid high-confidence detections in crowded instances. With the intention of alleviating this, Bodla *et al.* [142] introduced Soft-NMS: instead of removing overlapping detections, the confidence score is reduced proportionally to the IoU with the closest highest confidence detection. This process is illustrated in Fig. 2.6b. Mathematically, the new confidence score using Soft-NMS is given by:

$$s'_i = \begin{cases} s_i, & \mathrm{IoU}(\mathbf{b}_i, \mathbf{b}_{\max}) < \tau \\ s_i \left(1 - \mathrm{IoU}(\mathbf{b}_i, \mathbf{b}_{\max})\right), & \mathrm{IoU}(\mathbf{b}_i, \mathbf{b}_{\max}) \geq \tau \end{cases}. \tag{2.19}$$

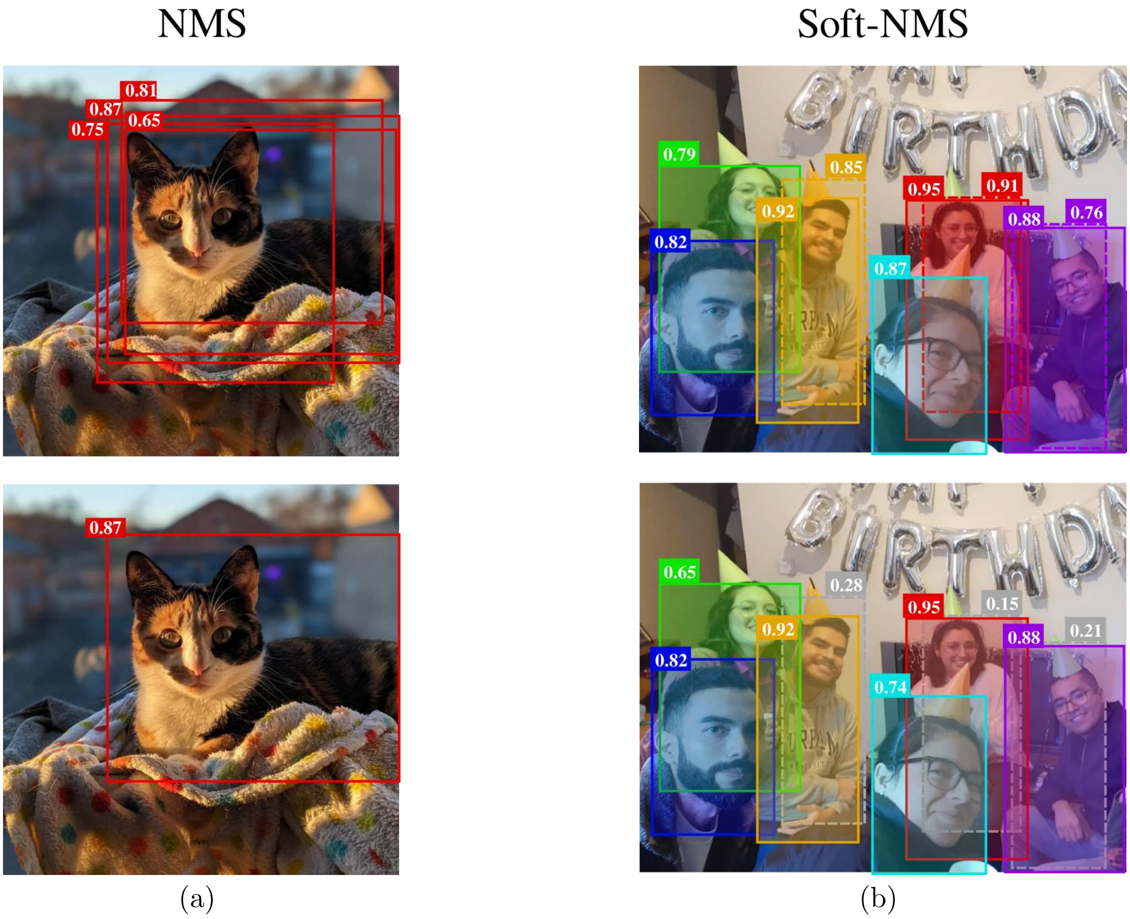


Figure 2.6: NMS for object detection. (a) NMS removes bounding boxes that overlap a box with maximum confidence. (b) Soft-NMS [142] reduces the confidence score proportionally to their IoU with the maximum confidence box. Grey boxes would be eliminated by rejecting detections with lower confidence than 30%.

This simple technique is able to increase the COCO AP metric by 1.1% using the Faster R-CNN [2] detector. Other alternative approaches to NMS exist, such as the positive sample selector [143], a convolutional layer that classifies if a predicted bounding box belongs to the predicted class.

In addition to NMS, other post-processing techniques may be applied if some priors are known [144, 145]. The first contribution discussed in this thesis (Chapter 3) is in fact a post-processing technique that considers the epipolar geometry of corresponding views.

Loss Functions

Object detection is a joint task involving the regression problem of finding the coordinates of the bounding boxes of the objects in the image and the classification problem of assigning a category to each of these bounding boxes. Each predicted bounding box is matched to a ground truth if the IoU is greater than a threshold value; otherwise, it is considered that its truth class is background. In general, the loss for the object category is a cross-entropy function, while the dimensions of the bounding box are usually regressed with an L_n norm. However, some works [146–148] have shown that using IoU-based loss functions allows for faster convergence time and improved performance. The reasoning behind this is that a small change in a bounding box corner may not represent an important error in the coordinates but it might significantly affect the IoU, especially for elongated objects. The details of the implementation of the loss functions depend on the architecture of the detector. Specific functions are reviewed in Sections 2.2.5 to 2.2.7.

2.2.5 Two-Stage Detectors

After the success of CNN for image classification demonstrated by the AlexNet [1], some works started to investigate their use for object detection [149]. The first CNN architecture that was able to substantially improve object detection performance is R-CNN [71] and its further improvements [2, 72]. The R-CNN family of detectors works by predicting a set of candidate bounding boxes and then classifying them as an object category or background. In the case of it being an object, the bounding box parameters are further regressed. This two-stage detection strategy, which can be extended to multi-stages, showed to be an attractive approach and forms the basis of many modern detector architectures.

The first version of R-CNN [71] uses a Selective Search [150] approach to predict a set of 2,000 bounding box candidates. Each bounding box wraps a region that is cropped and passed to an AlexNet backbone that is trained to classify among $N + 1$ classes (with the extra background class). A proposed region P is labelled as a class $c = 1, \ldots, N + 1$ object if it has an IoU > 0.5 with a ground truth bounding box of that class. The backbone outputs a vector $\phi(P) \in \mathbb{R}^{N+1}$ that is transformed into a class probability distribution $\mathbf{p} \in [0,1]^{N+1}$ using the softmax operator:

$$\mathbf{p}_c = \mathrm{softmax}(\phi(P), c) = \frac{\exp \phi(P)_c}{\sum_{k=1}^{N+1} \exp \phi(P)_k} \,. \tag{2.20}$$

Subsequently, the backbone is trained using a cross-entropy cost function

$$\mathcal{L}_{class}(\mathbf{p}, c) = -\log(\mathbf{p}_c) \,. \tag{2.21}$$

After the backbone is trained, the feature vectors are used to train a set of N linear support vector machines (SVM) to give a class confidence score to the wrapped region. Then, the class with the largest score is the predicted class. A further bounding box regressor is also used to improve localisation performance. Given the proposed region P with normalized centre $(P_x, P_y) \in [0,1]^2$, width P_w and height P_h, it is paired to the ground truth $G = (G_x, G_y, G_w, G_h)$, using the same subscript notation, with maximum overlap. If the IoU between P and G is less than 0.6, it is disregarded. A new bounding box $\hat{P}$ is parameterized by the predicted values $T = (t_x, t_y, t_w, t_h)$ such that

$$\begin{aligned} \hat{P}_{\{x,y\}} &= P_{\{w,h\}} t_{\{x,y\}} + P_{\{x,y\}} \,, \\ \hat{P}_{\{w,h\}} &= P_{\{w,h\}} \exp(t_{\{w,h\}}) \,, \end{aligned} \tag{2.22}$$

where $t_{\{x,y,w,h\}}$ is estimated from a regularized linear regression function $\zeta_{\{x,y,w,h\}}(P)$ that is trained with the paired tuples $(\hat{P}, G)$. R-CNN achieves an mAP of 53.7% on the VOC 2012 challenge (more than a 13% mAP increment when compared to the previous best result [151]).

Despite its success, R-CNN has several drawbacks. The most notable is that R-CNN is sufficiently slow given its multi-stage nature and the fact that features are extracted for each region proposal. To overcome this, Spatial Pyramid Pooling Network (SPPNet) [152] introduces the spatial pooling operation to allow feature extraction from any region in a feature map, meaning that the features were calculated only once per image. This pooling layer divides a region in the feature map into n bins and a max-pooling operation is performed spatially among all the feature vectors covered in that region. SPPNet performs this in a pyramid fashion with different numbers of bins per pyramid level. The features are then concatenated and trained in the same way as in R-CNN. This approach also removes the need of training fixed-size images. Fast-RCNN [72] uses the same pooling strategy as in SPPNet but uses only one resolution level, renaming it as Region of Interest (RoI)-pooling. To further improve detection speed, Fast-RCNN eliminates the multiple SVM in R-CNN and SPPNet. Instead, Fast-RCNN uses RoI-pooling to extract the features from each object and feed it to two separate heads for classification and bounding box localisation. The localisation head predicts bounding boxes $T_k = (t_{k,x}, t_{k,y}, t_{k,w}, t_{k,h})$, parameterized as in Eq. (2.22), for each class k. Fast R-CNN uses the multi-task loss function:

$$\mathcal{L}(\mathbf{p}, c, T_c, G^*) = \mathcal{L}_{class}(\mathbf{p}, c) + \mathbb{1}_{\{c \geq 1\}} \lambda \mathcal{L}_{loc}(T_c, G^*) \,, \tag{2.23}$$

where $\mathcal{L}_{class}(\mathbf{p}, c)$ is the same cross-entropy loss as in Eq. (2.21), the ground truth box $G^* = (g_x, g_y, g_w, g_h)$ is parameterized using the proposed region P as:

$$\begin{aligned} g_{\{x,y\}} &= (G_{\{x,y\}} - P_{\{x,y\}}) / P_{\{w,h\}} \,, \\ g_{\{w,h\}} &= \ln(G_{\{w,h\}} / P_{\{w,h\}}) \,, \end{aligned} \tag{2.24}$$

and the localisation loss is given by:

$$\mathcal{L}_{loc}(T_c, G^*) = \sum_{i \in \{x,y,w,h\}} \text{smooth}_{L_1}(t_{c,i} - g_i) \,, \tag{2.25}$$

where the smooth L_1 function is defined as:

$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 \,, & |x| < 1 \\ |x| - 0.5 \,, & \text{otherwise} \end{cases} . \tag{2.26}$$

The symbol $\mathbb{1}_{\{c \geq 1\}}$ indicates 1 when $c \geq 1$ and 0 otherwise, and the λ hyper-parameter controls the balance between the classification and the localisation tasks.

Additionally, Fast R-CNN replaces the AlexNet backbone with a VGG16 [63]. Fast R-CNN outperforms the previous R-CNN in the VOC 2007, 2010 and 2012 challenges with an increase in training speed up to 18.3 times and up to 213 times faster inference time.

The major bottleneck of the previous methods is the use of Selective Search for region proposals, which does not allow for real-time performance. To overcome this, Faster R-CNN [2] introduces the Region Proposal Network (RPN). This network is used to create the region proposals that are used in the Fast R-CNN architecture. The RPN takes a $W' \times H'$ feature map and predicts a set of region proposals. The RPN takes a 3×3 window for each feature map and predicts a foreground score (called objectness) and bounding box parameters with respect to k anchor boxes. These anchor boxes are predefined boxes at different aspect ratios and scales and form the basis of many detectors (although they are similar to the proposals used in R-CNN and Fast R-CNN). In total, it produces a set of $K = W'H'k$ anchor boxes for a total of $4K$ box parameters and $2K$ objectness scores. The RPN is trained with the same loss function as in Eq. (2.23) with only two classes (foreground and background). Once the regions are proposed, the rest of the architecture is similar to Fast R-CNN. Faster R-CNN is trained in a multi-step manner and the final model backbone shares the weights for both the RPN and the detection head. The integration of the RPN in Faster R-CNN improves the detection performance compared with Fast R-CNN by 2.6% in the COCO test set whilst achieving a 5 fps performance using a VGG backbone (compared to the 0.5 fps performance of Fast R-CNN) and 17 fps with a lighter backbone. The architecture of Faster R-CNN forms the basis of many object detectors, with modifications in the backbone, the RoI pooling function, the detection heads or the loss function [6,7,61,74,76,81,153–157]. Further stages can be added for refinement and improved accuracy [157].

2.2.6 One-Stage Detectors

Although Faster R-CNN [2] based architectures have great detection performance, their relatively slow processing time makes them difficult to implement in real-time applications. Considering this, Redmon *et al.* [3] proposed the You Only Look Once

(YOLO) detection architecture, comprising a CNN (based on GoogleLeNet [62]) followed by a feed-forward network. YOLO performs object detection within a single forward step, making it the first one-stage detector. Its output is a fixed-size grid $S \times S$ with B predicted bounding boxes per grid location. YOLO predicts the x, y, w and h of each bounding box, an objectness score $P(\text{object})$ and a probability $P(c|\text{object})$ of belonging to a class c. This is encoded in an $S \times S \times (5B + C)$ output tensor which predicts a class per grid location. YOLO is trained using the mean squared error (MSE) loss function among its different parameters:

$$\begin{aligned}\mathcal{L}_{YOLO} =& \sum_{i=1}^{S^2} \sum_{j=1}^{B} \mathbb{1}_{ij}^{obj} \left(\mathbf{b}_i^* - \hat{\mathbf{b}}_{ij}^*\right)^2 + \left(\mathbb{1}_{ij}^{obj} + \lambda_{noobj}(1 - \mathbb{1}_{ij}^{obj})\right) \left(C_{ij} - \hat{C}_{ij}\right)^2 \\ &+ \sum_{i=1}^{S^2} \sum_{c \in \text{Classes}} \mathbb{1}_i^{obj} \left(p(c)_i - \hat{p}(c)_i\right)^2 ,\end{aligned} \tag{2.27}$$

where $\mathbf{b}^* = (x, y, \sqrt{w}, \sqrt{h})$, C is the confidence score defined as the product of $P(\text{object})$ with the IoU with the ground truth bounding box, $p(c)$ is the class-probability, the λ_{coord} and λ_{noobj} coefficients are hyper-parameters and the symbol $\mathbb{1}^{obj}$ is 1 when an object is present. The width and height of the bounding box are root squared in the loss function to partially address the issue of bigger objects contributing more to the final loss. The subscripts refer to the i-th grid cell and the j-th bounding box. While YOLO has worse detection performance compared to Fast R-CNN [72] (a drop of more than 10% in mAP on the PASCAL VOC 2012 challenge [121]), it is extremely fast, having a processing time of 45 fps and 150 fps with a shallower backbone. YOLO needs a fixed input image size, using 448 × 448 images in the original work.

Another one-stage detector released soon after YOLO is the Single Shot Detector (SSD) [5]. SSD consists of a CNN with no feed-forward layers. Its backbone is a VGG16 with extra convolutional layers at the end (this can be seen as a neck architecture, Section 2.2.4). It incorporates a set of different scale and aspect ratio anchor boxes per grid location, similar to Faster R-CNN [2], and uses a slightly modified version of the loss function in Eq. (2.23). Another contribution from SSD is the use of a feature map pyramid to account for multi-scale predictions, assigning a set of anchor boxes to each grid cell from each feature map, but not combining higher resolution maps into deeper layers, as in FPN [74]. Apart from being a real-time detector (with up to 59 fps with a low-resolution input image), SSD outperforms Faster R-CNN and YOLO in the VOC 2007 and COCO 2015 challenges.

A subsequent variant of YOLO, YOLO9000 [158] was released following the improvements in accuracy and speed from SSD. YOLO9000 incorporates significant changes in the YOLO detector, such as the use of batch normalization [159], anchor boxes and multi-scale training. YOLO9000 follows a similar parameterisation as in Eq. (2.22), with the difference that it predicts anchor boxes offsets bounded to $[0, 1]$ in the feature grid space using a sigmoid function $\sigma(x)$. Hence, having an anchor box with dimensions p_w and p_h and the evaluated cell has an offset c_x, c_y from the top-left corner of the image, YOLO9000 predicts 5 values t_x, t_y, t_w, t_h and t_o per anchor box such that the final prediction $\mathbf{b} = (b_x, b_y, b_w, b_h)^\top$ is given by:

$$\begin{aligned} b_{\{x,y\}} &= \sigma(t_{\{x,y\}}) + c_{\{x,y\}}\,, \\ b_{\{w,h\}} &= p_{\{w,h\}} \exp(t_{\{w,h\}})\,, \\ P(\text{Object}) * \text{IoU}(\mathbf{b}, \text{object}) &= \sigma(t_o)\,. \end{aligned} \tag{2.28}$$

A class is predicted per bounding box, instead of the grid cell-wise strategy in YOLO. The dimensions of the anchor boxes are learnt using a k-means algorithm over the dimensions of the bounding boxes in the training set. To account equally for large and small boxes, the distance metric for the clustering is chosen to be:

$$d(\text{box}, \text{centroid}) = 1 - \text{IoU}(\text{box}, \text{centroid})\,. \tag{2.29}$$

YOLO9000 also introduces a new backbone called Darknet-19, which is adapted to run faster for object detection. YOLO9000 achieves real-time performance with 78.6% mAP in the PASCAL VOC 2007 challenge, running at 40 fps using 544×544 input images and 69.0% mAP at 91 fps using 288×288 images. A further iteration for the YOLO series is released with YOLOv3 [75] that considers a deeper backbone with residual connections (inspired by ResNets [61]), multi-scale prediction and a cross-entropy loss function for class prediction. With these changes, YOLOv3 performs significantly better at the COCO dataset with 57.9% $\mathrm{AP}_{0.5}$ (compared to 44.0% in YOLO9000) and it still achieves real-time performance running at 19 fps with a 608 × 608 input image and at 45 fps for 320 × 320 input images (51.5% $\mathrm{AP}_{0.5}$).

Other one-stage detectors have a similar architecture and training strategy as the YOLO series and SSD architecture. RetinaNet [6] includes an FPN and two separated classification and regression heads. It also introduces the focal loss

$$\mathrm{FL}(\mathbf{p}_c) = -(1 - \mathbf{p}_c)^{\gamma} \log(\mathbf{p}_c)\,, \tag{2.30}$$

a variation of the cross-entropy loss to account for class imbalance. Subsequent detectors inspired by the YOLO detector [46, 139, 160–162] have come out in the last years from different research groups incorporating recent advances in backbone architectures, data augmentation, decoupled heads or training strategies. Although is generally accepted that multi-stage detectors perform better but slower than one-stage detectors, YOLOv7 [162] achieves up to 56.8% COCO AP (at 36 fps), surpassing several two-stage detectors. Some one-stage detectors have opted for dropping the anchor boxes. The Fully Convolutional One-Stage (FCOS) detector [163] achieves a comparable performance to anchor-based detectors by adding a *centerness* loss, which is a measure of how the prediction is far from the centre of an object. Other detectors have instead focused on predicting the corners of the bounding box, translating the object detection to a keypoint detection problem [164, 165].

2.2.7 Transformers for Computer Vision

The success of the Transformer architecture [128] for natural language processing (NLP) modelling brought a significant interest to adapting it for computer vision. This section gives a brief introduction to the Transformer architecture and how it is adapted for object detection and computer vision applications.

The Transformer Architecture

Given ordered d-dimensional input values $\mathbf{x}_1, \ldots, \mathbf{x}_n$, the Transformer, depicted in Fig. 2.7, predicts an ordered output $\mathbf{y}_1, \ldots, \mathbf{y}_m$. It consists of a series of stacked

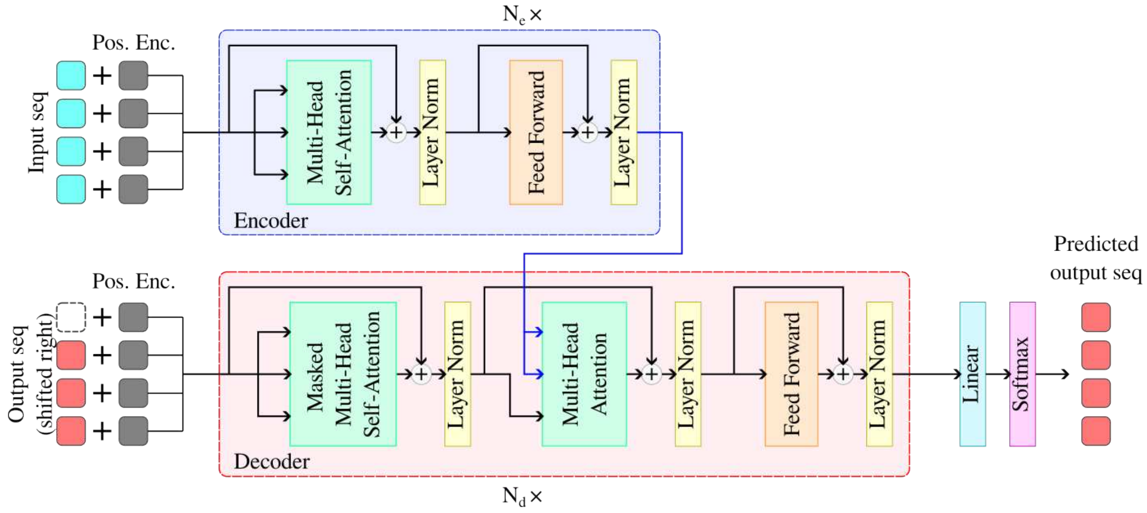


Figure 2.7: The Transformer [128] architecture.

encoder and decoder networks. If the inputs are contained in a matrix $X \in \mathbb{R}^{n \times d}$ such that $X = [\mathbf{x}_1 \ldots \mathbf{x}_n]^\top$, the encoder network is a function $f : \mathbb{R}^{n \times d} \to \mathbb{R}^{n \times d}$ that produces encoded values. The first layer of the encoder is based on the attention mechanism: given a set of m d_k-dimensional queries contained in a matrix $Q \in \mathbb{R}^{m \times d_k}$, the attention takes the weighted sum over m values $v_i \in \mathbb{R}^{d_v}$ based on the similarity of each query to another set of m key values. The keys and values are encoded in the matrices $K \in \mathbb{R}^{n \times d_k}$ and $V \in \mathbb{R}^{n \times d_v}$. If the similarity is calculated using the dot product, the attention mechanism is obtained by:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right) V \, . \tag{2.31}$$

The denominator inside the softmax function is used to stabilize gradients during training. In the Transformer architecture, the queries, keys and values are linearly transformed into h heads, defining a multi-head attention (MHA) as:

$$\text{MHA}(Q, K, V) = \text{concat}\left(H_1, \ldots, H_h\right) W^O \, , \tag{2.32}$$

$$H_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \, , \tag{2.33}$$

where $W^Q \in \mathbb{R}^{d_k \times d'_k}$, $W^K \in \mathbb{R}^{d_k \times d'_k}$, $W^V \in \mathbb{R}^{d_v \times d'_v}$ and $W^Q \in \mathbb{R}^{hd_v \times d_{model}}$ are learnable parameters. The encoder of the Transformer first uses a multi-head self-attention (MHSA) layer, given by $\text{MHSA}(X) = \text{MHA}(X, X, X)$, with $X \in \mathbb{R}^{n \times d}$

being the input (or encoded) sequence. Following the MHSA layer, the result is passed through a feed-forward layer. Both layers are residual (*i.e.*, added with the input of the layer) and normalized. The Transformer stacks a series of N_e encoders, where the output of the k-th encoder is the input for the $(k+1)$-th encoder. On the other hand, the Transformer decoder takes the output sequence shifted to the right $\varnothing, \mathbf{y}_1, \ldots, \mathbf{y}_m$, where $\varnothing$ is a special value indicating the beginning of the sequence. This is passed to a masked MHSA layer, which is used to only consider attention with respect to previous values and preserve causality, then to an MHA layer where the Q and K values are the output of the encoder, and finally to a feed-forward layer. Similar to the encoder, these layers are residual and normalized. The Transformer also stacks N_d decoders. Finally, a linear layer followed by a softmax operation is performed in order to predict the next element of the sequence. In order to preserve the order of the sequence, a positional encoding (PE) is added to each input and output sequence. The PE used in the Transformer is a vector with the same dimensions as the input, and comprises a combination of sines and cosines:

$$\begin{aligned} \mathrm{PE}_{pos,2i} &= \sin(pos/10,000^{2i/d})\,, \\ \mathrm{PE}_{pos,2i+1} &= \cos(pos/10,000^{2i/d})\,. \end{aligned} \tag{2.34}$$

The Transformer architecture is currently the *de facto* architecture for NLP tasks, being the basis for many state-of-the-art models [166–168]. Next, its use for object detection is discussed.

Detection Transformer

The detection Transformer (DETR) [8] is a detection head that uses the Transformer architecture to output a fixed-size set $\hat{B}$ of predictions from the feature map of an image. DETR takes the detection task as a direct set prediction. In this sense, for an image with M ground truth bounding boxes $\mathbf{b}_i$, DETR predicts a set of $N \gg M$ bounding boxes $\hat{\mathbf{b}}_i$. Each box is associated with a class c_i, including the background category. To allow a direct set comparison, DETR adds $N-M$ background instances to the ground truth set. Subsequently, given the set of N permutations $\mathcal{G}_N$, the permutation with the minimum loss with respect to the ground truth set,

$$\hat{\sigma} = \underset{\sigma \in \mathcal{G}_N}{\arg\min} \sum_{i}^{N} \mathcal{L}_{match}(\mathbf{b}_i, c_i, \hat{\mathbf{b}}_{\sigma_i}, \hat{c}_{\sigma_i}) \,, \tag{2.35}$$

is found via the Hungarian algorithm. The match loss $\mathcal{L}_{match}$ is a combination of a cross-entropy loss for the classification task and a bounding box function, which itself is a combination of the generalized IoU loss [146] and a L_1 loss, thus given by

$$\mathcal{L}_{match} = -\log \hat{p}_{\sigma_i}(c_i) + \mathbb{1}_{\{c_i \neq \varnothing\}} \left(\lambda_{IoU} \mathcal{L}_{IoU}(\mathbf{b}_i, \hat{\mathbf{b}}_{\sigma_i}) + \lambda_{L_1} \|\mathbf{b}_i - \hat{\mathbf{b}}_{\sigma_i}\|_1 \right) \,, \tag{2.36}$$

where $\hat{p}_{\sigma_i}(c_i)$ is the cross-entropy of the σ_i-th predicted class with respect to the c_i class, $\mathbb{1}_{\{c_i \neq \varnothing\}}$ indicates 1 if the ground truth class is an object (0 otherwise) and λ_{IoU} and λ_{L_1} are hyper-parameters. The final DETR loss is the sum of Eq. (2.36) for all predicted bounding boxes.

The general DETR architecture is similar to the original Transformer architecture. It takes a feature map $f \in \mathbb{R}^{d \times H \times W}$ (which is the output of a backbone with a reduced feature dimension d via a 1×1 convolutional layer) as the input sequence of a Transformer. In order to maintain the inherent structure of the 2D feature map, a positional encoding is also added. The encoder follows the same architecture as in the Transformer, while the decoder adds some modifications: the input of the decoder is a sequence of N learned embeddings, named object queries, which act as positional encodings, and the self-attention layer of the decoder is unmasked, meaning that all predictions are performed at the same time, in contrast to the autoregressive nature of the original Transformer architecture. Finally, each output of the encoder is passed to a feed-forward network that gives the final predictions. It is worth noting that DETR removes the need for heuristic post-processing steps such as NMS. DETR gets comparable detection performance when compared to a similar Faster R-CNN architecture in the COCO validation set, although with fewer frames per second. DETR can also be implemented in a multi-stage framework, where the output of the encoder serves as proposal inputs for the second stage.

A subsequent work, Deformable DETR [47] addresses some of the issues with the original implementation of DETR. It uses a pyramid of features (as in FPN [74])

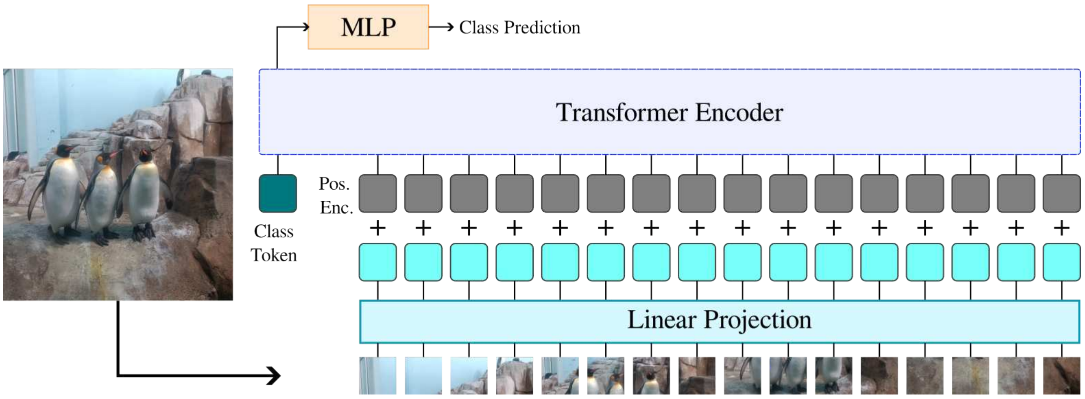


Figure 2.8: The Vision Transformer (ViT) [45] architecture.

and a deformable attention mechanism. Deformable DETR improves the COCO AP metric in the COCO 2017 validation set with 10× fewer epochs. These works show that Transformers are also suitable for vision tasks such as object detection. A different approach to using the Transformer architecture as a backbone for general-purpose computer vision tasks is discussed in the next section.

Vision Transformers

Following the growing success of the Transformer, Dosovitskiy *et al.* [45] presented the Vision Transformer (ViT), a Transformer-Encoder model that is used for image classification. The general architecture is shown in Fig. 2.8. ViT divides an image into a fixed number of patches and transforms them using a linear projection. These reprojected patches, along with positional encoding embeddings, are then fed to a standard Transformer encoder. In addition to the image patches, a special learnable class embedding is added at the start of the sequence. The output in the position of the class embedding is then passed through a multi-layer perceptron that performs the classification task.

The ViT has shown competitive performance for object classification and it has been studied as a backbone for object detection [169]. The idea to keep it similar to the original Transformer is that further improvements in the original architecture can be implemented in the ViT as well. Currently, ViT is the basis for state-of-the-art backbones for object detection [7, 81, 170] and it inspires the contribution presented in Chapter 4.

2.2.8 Multi-View Object Detection

Multi-view object detection (also referred to as *cross-view* object detection) refers to the location of objects that appear simultaneously in different views. It can be addressed using single-view object detection, but the multi-view geometry imposes some constraints that can be exploited to improve detection accuracy. In contrast with re-identification and tracking, multi-view object detection is not interested in identifying what objects represent the same instance. Additionally, objects may or not appear across different views depending on the shared FoV. In the literature, the term *multi-view* is sometimes a synonym for multi-modal learning (*e.g.*, Deng *et al.* [171]). However, this thesis uses multi-view in the geometric sense of Section 2.1, meaning that a camera representation of all the views must be possible. Although it is not strictly necessary, it is assumed that all views come from the same modality.

Recent work explicitly addressing multi-view object detection using contemporary detection architectures is limited. Nassar *et al.* [172] apply a convolutional neural network that takes multi-view images and corresponding geolocation information as inputs and uses a joint loss function considering all views, resulting in an increase of the detection mAP by up to 27.8%. Hou *et al.* [173] create a multi-view pedestrian detection by using projective transformations from each image to a floor plane. In their work, a head and foot detector is trained for all the views. The features learned from this intermediate task are passed to a projective transformation to take the feature to the same plane. Once in the same plane, this combined feature map is trained through another network to detect the location of the pedestrians. Despite using multi-view information to improve person localisation, they do not perform object detection. A similar approach is used in a subsequent work by Hou and Zheng [174], where the projective transformations are used to train a multi-view object detector. Their detector is similar to Deformable DETR [47] but with an extra consideration of view-wise attention normalization. A multi-view consistent augmentation technique is also proposed, demonstrating its ability to help in the generalization of their method. A significant limitation of these works [173, 174] is that they rely on the pedestrian lying on the same plane (the floor plane), to the point that people lying at different floor levels are not considered. In addition, the intrinsic and extrinsic camera parameters are also needed in order to get the projective transformations. This restriction does not hold for the methods presented in Chapters 3 and 4. Liu *et al.* [175] use different cross-sectional views of mammograms for breast cancer detection. They construct a bipartite graph node based on a k nearest neighbours (kNN) clustering of the feature map of each view. A graph convolution followed by an inverse kNN mapping is performed to create a cross-view representation. This is then aggregated to the previous feature maps and the final representation is fed to a detection head. Their technique shows better detection performance when compared to single-view detection baselines and previous multi-view techniques for breast cancer detection. Multi-view images have also been used to train 3D bounding box detectors [176, 177] and epipolar consistent keypoint detectors [178, 179].

Multi-view Datasets

Since the ground truth objects in multi-view data must be carefully annotated to comply with multi-view consistency, there is a limited collection of multi-view detection datasets within different application contexts. The details of these datasets are presented in Table 2.2. One of the first datasets with multi-view annotated objects is the Durham Baggage 6-classes (DB6) dataset [120], consisting of a four-view (fully overlapped) X-ray security imagery from a dual-energy Smiths 6040i X-ray scanner. One drawback of this dataset is that it is a private dataset for security reasons, as in most airport security imagery.

The use of such datasets for X-ray multi-view detection is discussed in the next section. In a different application context, the Wildtrack [106] dataset is a 7-view dataset including the annotation of over 300 people from an outdoor video sequence, with not all cameras sharing a similar FoV. WiseNET [180] is an indoor surveillance camera dataset, where only a few cameras share a similar FoV. The Home Action Genome (HOMAGE) dataset [181] includes annotations at different levels: it is a multi-view multi-modality dataset with hierarchical activity and atomic action annotations. Its third-person data branch includes corresponding annotations of people and objects in a room. More recent datasets include MultiviewC [182], a synthetic dataset for cattle tracking and detection that includes 2D and 3D bounding boxes; and the DOLPHINS dataset [183], an autonomous driving large-scale dataset with different scenes and modalities, including multi-view object annotations of cars on the road. Although these datasets have been used for a variety of applications, they usually rely on extra information or strong priors, such as calibrated cameras, known camera poses or objects sharing a common plane. Many of these priors cannot be extended to other domain applications, such as X-ray threat item detection. The methods of this thesis are developed to work with only the ground truth bounding boxes and no extra information is assumed.

Table 2.2: Multi-view detection datasets.

Dataset	Views	Data	Comments
Durham Baggage [120]	4	11,627 images with 494 cameras, 1,596 ceramic knives, 3,208 knives, 3,192 firearms, 1,203 firearm parts and 2,390 laptops.	X-ray private dataset. Objects appear overlapped.
Wildtrack [106]	7	2800 images and 42,533 objects (one class: person).	Outdoors pedestrian dataset. Annotated objects lie on the same plane.
WiseNET [180]	6	122,021 images and 111,913 objects (one class: person).	Indoors pedestrian dataset. Not all cameras share an FoV.
HOMAGE [181]	2	1,725 synchronized sequences, a total of 5,900 videos, and 497,534 annotated objects (across 86 classes).	Dataset for indoors action recognition. One ego-view is provided and from one to four extra views at different locations of the room.
MultiviewC [182]	7	3,920 images (560 for each view). 15 annotated instances (2D and 3D) on each image (one class: cow).	Synthetic cow detection dataset. 4 cameras on each corner and 3 on the top.
DOLPHINS [183]	3	42,276 images and 292,549 objects (two classes: pedestrian and car.	Autonomous driving detection dataset. 3D annotations, geo-positions and calibrations are provided.

Multi-view detection of X-ray imagery

An application context for multi-view object detection that is explored in this thesis is the identification of threat items in X-ray security imagery. X-ray cabin baggage scanners usually present multiple viewpoints of the bag since it allows screeners for better detection performance [24]. One of the earliest works to use multiple views from X-ray imagery is presented by Mery [184]. In this work, objects of interest such as razor blades and pencil tips are segmented using classical feature descriptors and are matched across different views if they lie near a region defined by the epipolar geometry. Fundamental matrix estimation is carried out using point correspondences generated by feature descriptors. Although this method shows a recall of 94.3% and a false positive rate of 5.6%, the test data set is small and samples are not highly cluttered in contrast to the consideration of operational conditions in the X-ray threat object detection work of [185]. A later work from Mery et al. [186] proposes the spatial reconstruction of matched keypoints. Subsequently, these points are clustered and projected back to the 2D domain only if they are large enough. The fundamental matrix is estimated as in [184] and matching keypoints are obtained through a heuristic process. More recent work from the same team on multi-view object detection includes a three-step process with deep learning approaches [187]. In the first step, threat objects are detected using the similarity of features and spatial distribution. Subsequently, reinforcement learning is used to predict the next view given the object in one source view. Finally, predictions are constrained using the epipolar geometry and the process described in [186]. This method increases the precision of handgun detection from 33% to 84% and the recall from 18% to 66%. Nevertheless, deep CNN object detectors outperform these approaches using single view imagery [185, 188, 189].

In the same context of classical techniques for object detection, Bastan et al. [190] proposed a simple method to search for objects in a spatial domain from 2D raw features. They noticed that in X-ray scanner imagery, bounding boxes of the same object at different views have approximately the same height and the same y coordinate (this is a consequence of the multi-view sensors lying in the same plane). They take advantage of this constraint but do not fully exploit the fact that these conditions are an effect of the epipolar geometry (i.e., epipolar lines being almost vertical).

With a more similar approach to the multi-view detection techniques presented in this thesis, Steitz *et al.* [191] add a 3D region of interest pooling layer to the Faster R-CNN architecture [191] for multi-view object detection in X-ray imagery. This work assumes that the relative position of the viewpoints is known, so scene reconstruction is possible [51]. This method pools deep features of each view into a spatial feature tensor to regress a 3D bounding box. Ground truth 3D bounding boxes are constructed by wrapping the polyhedron formed from the intersection of the rays of projection of 2D bounding boxes. Standard metrics are calculated by re-projecting back the detected 3D bounding box to the 2D domain. They were able to increase the average precision for firearm detection from 85.56% to 92.29%. This work, however, has the limitation that the relative position of the cameras has to be known. In the contribution presented in Chapters 3 and 4, this is not necessary.

2.3 Neural Radiance Fields

This section examines the theory and applications of Neural Radiance fields (NeRF), an image-based novel view synthesis proposed by Mildenhall *et al.* [12]. This technique has become increasingly popular in recent years due to its applications in virtual reality, video games and autonomous driving. An introduction and problem statement of NeRF is given in Section 2.3.1. Following, the detailed description of the formulation is presented in Section 2.3.2. Further recent NeRF research is discussed in Section 2.3.3. Finally, a review of NeRF applications is detailed in Section 2.3.4.

2.3.1 Introduction

Novel view synthesis is a classical and long-standing task in computer vision. An approach for this task is image-based rendering, which synthesizes novel views from a set of given input images. In general, in traditional image-based rendering, the denser the image representation, the less knowledge of the internal geometry is needed. A review of non-deep-learning image-based rendering is presented by Shum and Kang [192]. More recent techniques for image-based rendering use deep neural networks to create a volumetric representation from sampled input images [193–195] or to predict a set of weights for image blending (mosaicking) [196]. Novel view synthesis has been thoroughly re-investigated after the introduction of Neural Radiance Fields [12]. NeRF learns an implicit representation of a 3D scene from a set of 2D images via a Multi-Layer Perceptron (MLP) that predicts the visual properties of 3D points uniformly sampled along the viewing ray given the spatial coordinates and viewing direction. This parameterisation gives NeRF the dual ability to both represent 3D scenes and synthesize unseen views. Nonetheless, the underlying sparse representation of 3D points learnt by the MLP may cause ambiguities that can lead to aliasing and blurring.

To overcome these issues, Barron *et al*. proposed mip-NeRF [42], an architecture that uses cone tracing instead of rays. This architecture encodes conical frustums as the inputs of the MLP by approximating the frustum regions in the space with a multivariate Gaussian. This re-parameterisation notably increases the reconstruction quality of multi-scale datasets. The next section overviews the formulation of NeRF and mip-NeRF, which form the basis for the Exact-NeRF contribution, presented in Chapter 5.

2.3.2 Formulation

A NeRF is a function f with parameters $\boldsymbol{\Theta}$ (normally an MLP) that maps a 3D point $\mathbf{x} \in \mathbb{R}^3$ and a viewing direction $\hat{\mathbf{d}} \in S^2$, where S^2 is the unit sphere in $\mathbb{R}^3$, to a colour $\mathbf{c} \in \mathbb{R}^3$ and a density $\sigma \in [0, +\infty)$, such that:

$$(\mathbf{c}, \sigma) = f(\mathbf{x}, \hat{\mathbf{d}}; \boldsymbol{\Theta}) \,. \tag{2.37}$$

In this context, f defines a field[3] which is used to render an image by compositing points lying on the ray defined from the centre of the camera and the midpoint of a

[3] This term is coined from physics, where a *field* is a physical quantity defined over the space.

pixel. This ray is represented by $\mathbf{r}(t) = t\mathbf{d} + \mathbf{o}$, where $\mathbf{o}$ is the camera position and $\mathbf{d}$ is the vector that goes from the camera centre to the pixel in the image plane. Under the NeRF framework, the ray is divided into N intervals. A set of points $\mathbf{r}(t_i)$ are drawn from a uniform distribution over each interval, such that:

$$t_i \sim \mathcal{U}\left[t_n + \frac{i-1}{N}(t_f - t_n), t_N + \frac{i}{N}(t_f - t_n)\right] , \tag{2.38}$$

where t_n and t_f are the near and far planes. In this sense, the colour and density of each point over the ray are obtained by $(\mathbf{c}_i, \sigma_i) = f(\mathbf{r}(t_i), \mathbf{d}/\|\mathbf{d}\|; \mathbf{\Theta})$.

The predicted pixel colour $\hat{C}(\mathbf{r})$ is obtained using numerical quadrature,

$$\begin{aligned} \hat{C}(\mathbf{r}) &= \sum_{i=1}^{N} T_i(1 - \exp(-\sigma_i \delta_i)) \\ T_i &= \exp\left(-\sum_{j=1}^{i-1} \sigma_j \delta_j\right) , \end{aligned} \tag{2.39}$$

where $\delta_i = t_{i+1} - t_i$. This process is carried out hierarchically by using a coarse sampling of N_c points using the uniform distribution in Eq. (2.38), and a fine sampling of N_f points, where the 3D points are drawn from the PDF formed by the weights of the density values of the coarse sampling. Different MLP with parameters $\mathbf{\Theta}_{\text{coarse}}$ and $\mathbf{\Theta}_{\text{fine}}$ are used for each sampling level, *i.e.*:

$$(\mathbf{c}_i^c, \sigma_i^c) = f(\mathbf{r}(t_i^{\text{uniform}}), \mathbf{d}/\|\mathbf{d}\|; \mathbf{\Theta}_{\text{coarse}}) \,, i = 1, \ldots, N_c \,, \tag{2.40}$$

$$t_j^f \sim \text{histogram}(\{t_i^{\text{uniform}}, \sigma_i^c\}_{i=1}^{N_c}) \,, \tag{2.41}$$

$$\left(\mathbf{c}_j^f, \sigma_j^f\right) = f(\mathbf{r}(t_j^f), \mathbf{d}/\|\mathbf{d}\|; \mathbf{\Theta}_{\text{fine}}) \,, j = 1, \ldots, N_f \,. \tag{2.42}$$

A predicted colour $\hat{C}^c$ is obtained using Eq. (2.39) over the set of coarse values in Eq. (2.40); a fine colour $\hat{C}^f$ is also obtained but using the $N_c + N_f$ points from Eqs. (2.40) and (2.42). NeRF is trained by minimizing the combination of the mean-squared error of the coarse and fine renderings for all rays in a dataset $\mathbf{r} \in \mathcal{R}$:

$$\mathcal{L} = \sum_{\mathbf{r} \in \mathcal{R}} \left[\|\hat{C}^c(\mathbf{r}) - C(\mathbf{r})\|_2^2 + \|\hat{C}^f(\mathbf{r}) - C(\mathbf{r})\|_2^2\right] . \tag{2.43}$$

NeRF uses a positional encoding (PE) on the input point raw coordinates to induce the network to learn higher-frequency features [197]. This PE is a high-frequency function $\gamma : \mathbb{R} \to \mathbb{R}^d$ that is applied to each coordinate individually. NeRF uses the function:

$$\gamma(x) = \left[\sin(2^0 x), \cos(2^0 x), \sin(2^1 x), \cos(2^1 x), \ldots, \sin(2^{L-1} x), \cos(2^{L-1} x)\right]^\top . \quad (2.44)$$

Although γ is similar to the Transformer PE in Eq. (2.34), it serves a different purpose: while the Transformer uses the PE to introduce a natural sense of order within a sequence, NeRF uses the PE to induce a bias to learn high-frequency functions, which is desired in image rendering. In this sense, Eq. (2.37) becomes:

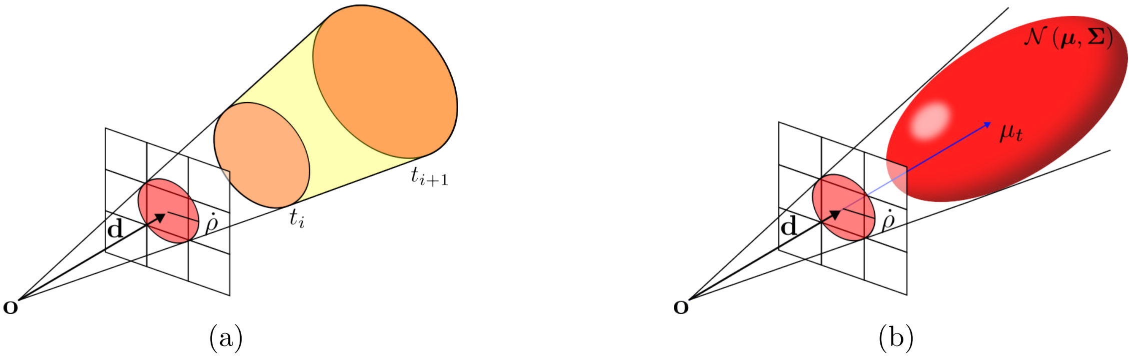
$$(\mathbf{c}, \sigma) = (f \circ \gamma)(\mathbf{x}, \hat{\mathbf{d}}; \boldsymbol{\Theta}) = f(\gamma(\mathbf{x}), \gamma(\hat{\mathbf{d}}); \boldsymbol{\Theta}) . \quad (2.45)$$

Mip-NeRF and Cone Tracing

The sampled points in NeRF are intended to represent a region in the volumetric space. This can lead to ambiguities that may cause aliasing. In this sense, mip-NeRF [42] proposes to use a volumetric rendering by casting cones instead of rays, changing the input of the MLP from points to cone frustums. This change has the direct consequence of replacing ray intervals by conical frustums $F(\mathbf{d}, \mathbf{o}, \dot{\rho}, t_i, t_{i+1})$, where $\dot{\rho}$ is the radius of the circular section of the cone at the image plane (Fig. 2.9). This leads to the need for a new positional encoding that summarizes the function in Eq. (2.44) over the region defined by the frustum. The proposed integrated positional encoding (IPE) is thus given by:

$$\gamma_I(\mathbf{d}, \mathbf{o}, \dot{\rho}, t_i, t_{i+1}) = \frac{\iiint_F \gamma(\mathbf{x}) dV}{\iiint_F dV} . \quad (2.46)$$

Since the integral in the numerator of Eq. (2.46) has no closed-form solution, mip-NeRF approximates it by considering multivariate Gaussians fitted to the cone



(a) (b)

Figure 2.9: Mip-NeRF [42] cone parameterisation. (a) Mip-NeRF cone tracing. (b) In practice, a Gaussian is fitted to the cone representation to approximate the positional encoding of the cone.

frustums. Subsequently, the approximated IPE γ^* is given by:

$$\begin{aligned}\gamma^*(\boldsymbol{\mu}, \boldsymbol{\Sigma}) &= \mathbb{E}_{x\sim\mathcal{N}(\mathbf{P}\boldsymbol{\mu}, \mathbf{P}\boldsymbol{\Sigma}\mathbf{P}^\top)}\left[\gamma(\mathbf{x})\right] \\ &= \begin{bmatrix} \sin(\mathbf{P}\boldsymbol{\mu}) \circ \exp(-(1/2)\mathrm{diag}(\mathbf{P}\boldsymbol{\Sigma}\mathbf{P}^\top)) \\ \cos(\mathbf{P}\boldsymbol{\mu}) \circ \exp(-(1/2)\mathrm{diag}(\mathbf{P}\boldsymbol{\Sigma}\mathbf{P}^\top)) \end{bmatrix} ,\end{aligned} \tag{2.47}$$

where $\boldsymbol{\mu} = \mathbf{o} + \mu_t \mathbf{d}$ is the centre of the Gaussian for a frustum with mean distance along the ray μ_t, $\boldsymbol{\Sigma}$ is the covariance matrix, $\circ$ denotes element-wise product and:

$$\mathbf{P} = \left[I_{3\times3}, 2I_{3\times3}, 4I_{3\times3}, \ldots, 2^{L-1}I_{3\times3}\right]^\top . \tag{2.48}$$

Mip-NeRF shows superior performance in scenes with samples at different resolutions or with variable distances to the object centre by reducing blurring and aliasing. It is also the base formulation for subsequent architectures [40, 43, 50, 198–200].

2.3.3 Recent Advances in Neural Radiance Fields

Since its original conception, recent changes in formulations have been developed within the NeRF architecture, attending to challenging setups, shortening the training time or general performance improvements (*e.g.*, Mip-NeRF [42]).

Changes in Parameterisation

A careful analysis of NeRF and Mip-NeRF parameterisation may lead to an overall improvement in the reconstruction performance. For instance, Ref-NeRF [201] changes the parameterisation to use the normal surface vectors and the reflection angles, also including diffuse colour and roughness variables to the MLP. Ref-NeRF introduces an integrated directional encoding (similar to the IPE of mip-NeRF) in order to model the rates of change on the appearance of different materials when the angle of reflection changes. Additionally, Ref-NeRF uses a regularization term to concentrate the volume density around surfaces and predict better normal vectors (having a direct impact on foggy regions). Some works have investigated the use of NeRF with unconventional input images. RawNeRF [199] uses raw input images to train a NeRF, *i.e.*, images that have not gone through a preprocessing pipeline such as distortion removal or low dynamic range tone mapping. This allows the synthesis of raw renderings that can be post-processed in order to have different properties, such as exposure, focus or tone mapping. NeRFReN [129] proposes to divide the scene between transmittance and reflectance to model reflections in NeRF. This formulation, along with some geometric constraints, outperforms the modelling of reflecting surfaces (such as mirrors or windows), which is best noticed when looking at the depth maps.

Sparse views and Multi-view Consistency

Some techniques focus on improving the novel view synthesis of NeRF for a limited number of input views. Reg-NeRF [50] introduces appearance and geometry regularizers that enable NeRF to generalize with sparser views (as low as three views), whilst pixelNeRF [202] uses convolutional features from the input images to train an architecture that learns scene priors, enabling novel view synthesis from even one single view. RapNeRF [203] uses a multi-view consistent ray sampling strategy that enables view extrapolation, where the original NeRF formulation fails. Aug-NeRF [204] incorporates strong augmentation techniques by augmenting perturbations to the training process, aiming for an improved generalization ability. BARF [205] uses bundle adjustment, an optimisation technique that corrects the position of cameras and triangulated 3D points, to account for inaccurate camera poses in the input training set. Dynamic scenes can also be modelled with NeRF as HyperNeRF [198], which uses a hyper-space that models each 5D radiance field as a slice of a higher dimensional space.

Unbounded Scenes

Recent works have focused on improving NeRF on unbounded scenes. In the original work, NeRF uses Normalized Device Coordinates (NDC) to model forward-facing unbounded scenes, which maps the viewing frustum to the cube $[-1, 1]^3$ [12]. On the other hand, NeRF++ [44] addresses a different parameterisation to allow the representation of 360° unbounded scenes. This parameterisation divides the scene volume into two regions, depending if they lie inside an inner sphere. These regions are processed using different networks, where the final rendering is a combination of foreground and background renderings. Mip-NeRF 360 [43] extends this concept and contracts the outer region to be used within the mip-NeRF parameterisation. It also includes an efficient network architecture and a regularizer that penalizes floaters (unconnected non-void regions that arise to try to explain multi-view inconsistencies).

Training Speed

Some efforts have been carried out to improve the training speed of NeRF. Most notably, Müller *et al*. [206] use a multi-resolution hash table of trainable feature vectors to train an optimized neural network for rendering. This formulation, along with an efficient GPU implementation, allows almost instant training of NeRF. Plenoxels [207] models instead a sparse voxel grid, where each voxel vertex stores a scalar opacity and a learned vector of (colour-wise) spherical harmonics, which are used to interpolate the colour and opacity of the voxel. These learned spherical harmonics are used instead of a trained neural network. Plenoxels achieve improved performance in a matter of minutes when compared to NeRF (which usually takes more than one day of training). Point-NeRF [208] trains NeRF 30× faster by predicting a point cloud from the input images using a pre-trained convolutional neural network and then building a radiance field upon this cloud.

2.3.4 Applications

NeRF applications are similar to those of traditional image-based rendering. However, the recent advances in NeRF (Section 2.3.3) and their powerful ability to represent scenes with high fidelity have made NeRF a great solution for new applications. Block-NeRF [40] composes several independently trained NeRF to model very large scenes, such as city blocks. It also uses an environment embedding to account for scenes that are obtained under different lighting or climate conditions. Similarly, Turki *et al.* [209] divide a large scene into different regions that are processed with different NeRF sub-modules. Given its ability to model 3D scenes, NeRF may have important medical applications: MedNeRF [41] learns a continuous representation of computed tomography scans and then creates a NeRF representation of novel scans using few input images (including one-image renderings), while Li *et al.* [210] use NeRF for 3D spine reconstruction from ultrasound images. 3D object detection can also be applied to NeRF scenes. Given a viewing direction and a pre-trained NeRF scene, NeRF-Loc [211] uses a Transformer on top of the output of the MLP of a NeRF to predict the coordinates of the projected 3D bounding box in the image. On the other hand, NeRF-RPN [212] uses 3D CNN using the predicted colour and density of the NeRF scene in a similar manner as Faster R-CNN [2] to directly output the 3D coordinates of the object. Other applications include AD-NeRF [213], which uses audio features as inputs to a NeRF to create dynamic scenes in the form of dynamic heads, and scene editing, including relighting [214, 215] and geometry based edition [216]. Although the research on NeRF is still incipient, these potential applications show the importance of the theoretical and practical understanding of NeRF.

2.4 Summary

This chapter presents a general overview of multi-view geometry, object detection and neural radiance fields. Section 2.1 presents the mathematical formulation of epipolar geometry and the fundamental matrix, whilst Section 2.2 discusses the general and multi-view object detection using modern deep learning architectures. Multi-view object detection presents additional challenges compared to traditional single-view detection, such as limited datasets, unknown camera poses and the need for multi-view consistent detections. While only a few works have focused on these problems, they do not demonstrate that their method can be applied to different scenarios or with further constraints. In this regard, Chapters 3 and 4 address these challenges, with a focus on multi-view X-ray security imagery and pedestrian datasets. In addition, Section 2.3 introduces NeRF, a recent state-of-the-art image-based rendering technique. NeRF formulation is discussed in Section 2.3.2, identifying the importance of volumetric parameterisations for reducing blurring and aliasing. Nonetheless, the spatial volumetric encodings introduced in this section are an approximation of an underlying integral with no analytical solution. Whilst some works have focused on improving NeRF for unbounded scenes, they rely on the basic parameterization of NeRF or mip-NeRF. For this reason, Chapter 5 explores an alternative volumetric parameterisation with an exact solution for NeRF encoding, showing better reconstructions of background regions. This improved quality of reconstructions might help to render better novel views which can be used in conjunction with multi-view object detection, aiding in producing multi-view consistent detections.

CHAPTER 3

Multi-view Object Detection Using Epipolar Constraints within Cluttered X-ray Security Imagery

Automatic detection for threat object items is an increasingly emerging area of future applications in X-ray security imagery. Although modern X-ray security scanners provide two or more views, the integration of automated object detection across the views has not been widely explored with rigour.

Therefore, this chapter investigates the application of geometric constraints using the epipolar nature of multi-view imaging to improve object detection performance. Furthermore, it assumes that images come from uncalibrated views, as this is true for most publicly available datasets. In this sense, a method to estimate the fundamental matrix using ground truth bounding box centroids from multiple view instance annotations is proposed. In addition, detections are given an epipolar confidence probability based on their distance distribution to the epipolar line. This probability is used as a confidence metric for merging duplicated predictions across multiple views using soft non-maximum suppression (NMS). While evaluation is carried out on X-ray security imagery given the increasing need to improve object detection, the methods developed in this chapter can be applied to any multi-view data with object-level annotations.

A subset of the contributions of this chapter has been published in the following peer-reviewed publication:

> Brian K.S. Isaac-Medina, Chris G. Willcocks and Toby P. Breckon. "Multi-view Object Detection Using Epipolar Constraints within Cluttered X-ray Security Imagery", in *IEEE International Conference on Pattern Recognition (ICPR)*, pp. 9889-9896, 2020.

3.1 Introduction

The screening of passenger baggage is an essential task for airport security to avoid threat items entering secure zones. In this regard, the efficiency and aptitude of screening operators are crucial in order to meet the required security standards. Due to the complex and cluttered nature of X-ray security screening imagery, operators must be assessed constantly in order to monitor their performance. Additionally, the ever-increasing use of air travel by the public puts increasing pressure on security screening efficiencies. The International Air Transport Association forecasts that the number of air transport passengers could double with up to 8.7 billion passengers globally by 2037 [217]. As a result, the introduction of assistive and automated technologies to aid in the security screening process is a major interest for future security needs [218].

Deep CNN architectures for object detection (Section 2.2.4) have shown to be effective for recognizing threat items in X-ray cabin baggage images [189, 219–221]. Different architectures have been tested in X-ray images for threat identification [188], validating their use in this domain. Motivated by the limited availability of X-ray cabin baggage images, transfer learning is used as an initialization step before training [188, 219]. As a result, in this chapter, CNN-based architectures for single-view object detection are used as the basis for extension into multiple-view object detection.

Contemporary X-ray scanners used for aviation security screening provide two or more views of the baggage content (Fig. 3.1). The geometry of two views of the same scene is related by epipolar geometry (Section 2.1.2). As seen in Sections 2.1.2

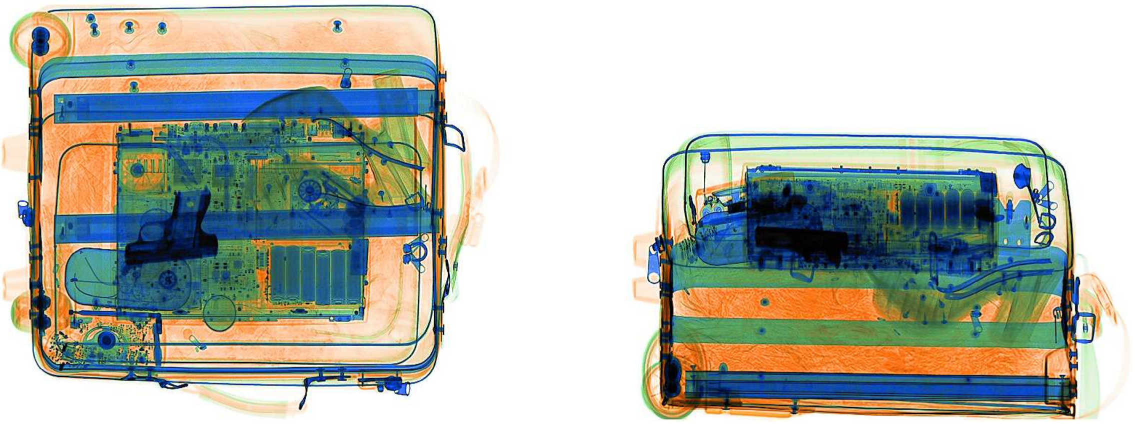


Figure 3.1: Exemplar of multi-view X-ray security imagery (bottom/side view).

and 2.1.3, the fundamental matrix can be constructed using the internal parameters of the cameras and their relative position (calibrated cameras), or estimated if a set of point correspondences $\{x_i \leftrightarrow x'_i\}$ is given. When the geometry is unknown (uncalibrated cameras) and point correspondences are not provided, the common methodology is to use feature detectors and descriptors to find matches between the different image views and then proceed to solve for F via least-squares minimization of the geometric inter-image feature projection error [222]. However, prior work from Kluppel *et al.* [223] demonstrates that conventional feature detection and matching is not suitable for transmission imagery (such as X-ray) due to the transparent nature of the object projections which vary with perspective view. Moreover, prior object detection work using multiple-view X-ray imagery, with consideration for epipolar constraints, is limited and primarily focuses on 3D bounding box reconstruction [191], where three views are needed [51]. A review of the techniques for multi-view object detection applied to X-ray security imagery is discussed in Section 2.2.8.

By contrast, this chapter addresses the use of epipolar geometry as a constraint to improve the performance of object detection in X-ray security imagery, where perspective viewpoints are uncalibrated and point correspondences are unknown. Our approach leverages the centres of ground truth bounding boxes used for training modern object detectors as an approximation of point correspondences to estimate the fundamental matrix. Subsequently, the distance of a given bounding box detection from an epipolar line projected from another view is modelled as a random variable with a normal distribution. Finally, the inter-view projection distance of

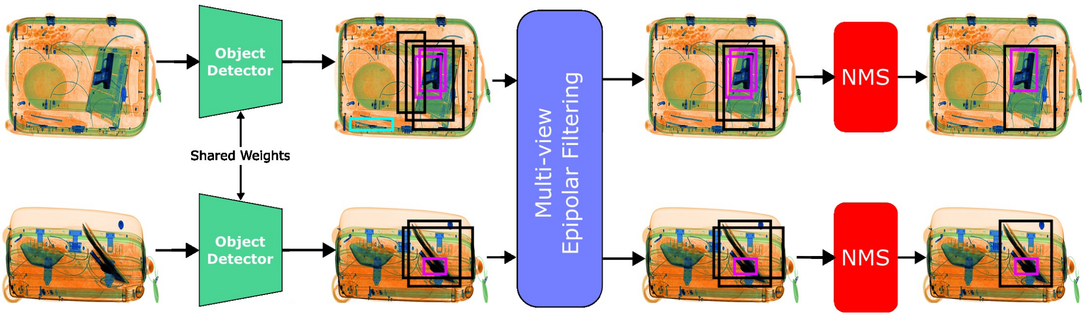


Figure 3.2: Overview of the epipolar filtering for multi-view object detection. In this approach, the raw predictions of an object detector are filtered to provide multi-view consistent predictions before the NMS step.

the epipolar line is used to get a multi-view correspondence probability which is jointly used with class and objectness probabilities for subsequent Soft-NMS post-processing. A general overview of the proposed architecture is shown in Fig. 3.2.

The key contributions of this chapter are as follows:

- A novel approach to recover the fundamental matrix from uncalibrated views based on the use of readily available ground truth object-level annotations for camera calibration, thus without needing to annotate a high volume of corresponding keypoints manually, applied to transmission (X-ray) imagery where conventional feature point matching fails [223].
- Formulation of a multi-view detection approach that cross correlates detections from multiple views by considering the inter-view epipolar constraint as an additional measure of confidence with Soft-NMS post-processing.
- Improved benchmark performance for the detection of representative threat objects within X-ray security imagery, based on the correlation of detections across multiple views, outperforming the prior work of Akcay *et al.* [188].

3.2 Method

The aim of the approach of this chapter is to exploit the constraints imposed by the epipolar geometry among the multiple X-ray views in order to improve detection performance. Specifically, we are interested in increasing detection performance whilst reducing false positive detection by correlating across multiple X-ray views and simultaneously improving object localization using the geometric distance of the bounding box centroid to the associated inter-image epipolar lines. In this way, we deal with uncalibrated image viewpoints such that object annotations, available from detector training, are used to estimate the fundamental matrix between these views. The resulting epipolar constraints between views are used to form the basis for subsequent multi-view object detection and filtering.

3.2.1 Fundamental Matrix Estimation from Object Level Annotations

In Section 2.1.3, it is demonstrated that the fundamental matrix can be estimated with at least 8 corresponding points between two views for uncalibrated cameras. When these point correspondences are unknown, one can rely on traditional feature matching among views. This technique, combined with RANSAC sampling, generally results in a good approximation of F [51]. However, as discussed before, this method is unreliable for X-ray imagery [223]. Under these conditions, the only available information is the instance-level annotations, *i.e.*, the ground truth bounding boxes of the threat items used for training an object detector. Although there are no explicit correspondences, the bounding box centroids can be used as approximations of the projection of the object centre (*i.e.*, the geometric centre defined as the arithmetic mean position of all points comprising the object), hence considering them as point correspondences. Nonetheless, it is worth noting that the centre of the bounding box $\hat{\mathbf{x}}_i$ does not necessarily coincide with the projection of the object centre $\bar{\mathbf{x}}_i$ (*e.g.*, Fig. 3.3). This difference can be modelled as a function of the relative position and orientation of the object with respect to the camera. Hence, the centre of a bounding box $\hat{\mathbf{x}}_i$ is modelled as:

$$\hat{\mathbf{x}}_i = \bar{\mathbf{x}}_i + \Psi_{obj} + \Delta\mathbf{x}\,, \tag{3.1}$$

where $\Psi_{obj} : \mathbb{P}^2 \rightarrow \mathbb{R}^2$ is a function that maps the offset between the projected centre of the object in the projective space $\mathbb{P}^2$ and the image to the centroid of the corresponding bounding box, and $\Delta\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_{err})$ is the associated annotation error. Since the position and orientation of the objects can be described as random events, we can model Ψ_{obj} as a random variable with a normal distribution $\mathcal{N}(\boldsymbol{\mu}_{obj}, \mathbf{\Sigma}_{obj})$. While in a completely random system, Ψ_{obj} should be unbiased (*i.e.*, there is not a *more probable* direction of the error) this work considers the biased distribution given the priors imposed by the dataset, such as all bags always lying in a belt of the X-ray scanner and being in a similar position towards the X-ray generators. Finally, Eq. (3.1) can be written as:

$$\hat{\mathbf{x}}_i = \bar{\mathbf{x}}_i + \Delta\hat{\mathbf{x}}\,, \tag{3.2}$$

with $\Delta\hat{\mathbf{x}} \sim \mathcal{N}(\boldsymbol{\mu}_{obj}, \mathbf{\Sigma})$. In this sense, the fundamental matrices estimated using the bounding box centres will carry an error that is modelled by $\Delta\hat{\mathbf{x}}$. This use of an error to create an epipolar confidence score is addressed in Section 3.2.2.

Since $\Delta\hat{\mathbf{x}}$ is a function of the object, using $\hat{\mathbf{x}}_i \leftrightarrow \hat{\mathbf{x}}'_i$ as point correspondences for the fundamental matrix estimation will carry different error modes for each object category. For this reason, this work gets a fundamental matrix for each object category as there are enough annotations for each class. An alternative parameterisation, which is not explored in this work and remains an area of future analysis, is the computation of a single fundamental matrix with per-class associated errors.

3.2.2 Epipolar Detection Confidence

The proposed method aims to use epipolar geometry as a constraint for post-processing object detection across multiple views in order to improve global detection performance. The proposed method relies on introducing a new object detection confidence based on epipolar constraints. Since NMS is based on confidence scores, the epipolar detection confidence has to be applied prior to NMS (since it modifies the prediction scores). In this sense, although this approach is detector agnostic, it cannot be applied to detectors that do not perform NMS, such as DETR [8, 47] or CenterNet [165]. In this context, four different detectors that define a class probability for a detected object, sometimes called confidence score, are tested. This

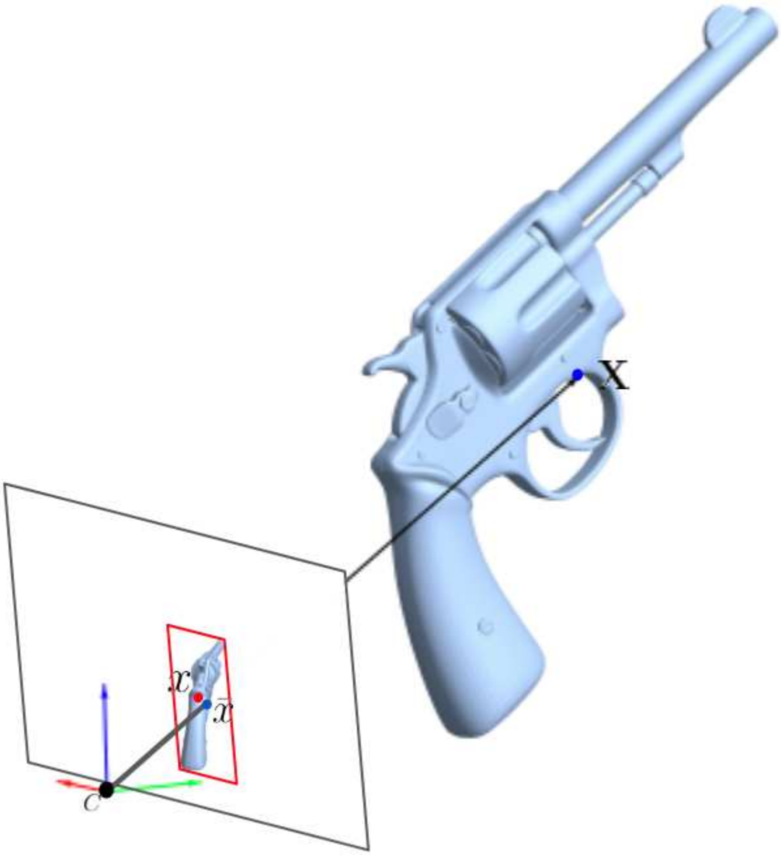


Figure 3.3: Comparison of the centre of a bounding box $\mathbf{x}$ with the projection of the real centre of the object $\bar{\mathbf{x}}$ in the plane defined by the camera $\mathbf{C}$.

confidence score is defined as the joint probability of being an object and belonging to a class, such that:

$$P(C = c, O) = P(C = c|O)P(O)\,, \tag{3.3}$$

where $P(O)$ is the objectness probability, *i.e.* the probability of the object being an occurrence of one of the object class types considered at training time, and $P(C = c|O)$ is the conditional probability of an object belonging to category c given that it is a valid object. This probability can be calculated either explicitly (*e.g.*, YOLOv3 [75]) or implicitly (*e.g.*, Faster R-CNN [2]) and it is used as a weight value for the Soft-NMS post-processing (Section 2.2.4).

We are now interested in extending the probability associated with each detection to take into consideration concurrent detections from other views. Recalling Section 2.1.2, a point $\mathbf{x}_i$ is projected to an epipolar line $\mathbf{l}'$ in another view as $\mathbf{l}' = F\mathbf{x}_i$.

The distance of a noisy corresponding point $\tilde{\mathbf{x}}'_i$ within this secondary view to the projected epipolar line is:

$$d(\tilde{\mathbf{x}}'_i, \mathbf{l}') = \frac{\tilde{\mathbf{x}}'^{\intercal}_i \mathbf{l}'}{\sqrt{l'^2_1 + l'^2_2}} = \frac{1}{c}\tilde{\mathbf{x}}'^{\intercal}_i \mathbf{l}' \, , \tag{3.4}$$

where l'^2_1 and l'^2_2 are the first two components of the epipolar line vector and $c = \sqrt{l^2_1 + l^2_2}$. The sign in Eq. (3.4) indicates the half-plane (defined by $\mathbf{l}'$) where $\tilde{\mathbf{x}}'_i$ lies. Substituting the point coordinates using the relation in Eq. (3.2) into Eq. (3.4) gives:

$$d(\tilde{\mathbf{x}}'_i, \mathbf{l}') = \frac{1}{c}\bar{\mathbf{x}}'^{\intercal}_i \mathbf{l}' + \frac{1}{c}\mathbf{l}' \cdot \Delta\hat{\mathbf{x}}'_i \, . \tag{3.5}$$

In this analysis, it is assumed that the epipolar line $\mathbf{l}'$ comes from the actual object centre mapped in the first view [1]. Therefore, the first element of the right side of the previous equation vanishes as the true correspondence point $\bar{\mathbf{x}}_i$ lies in $\mathbf{l}'$. Since $\mathbf{l}' \cdot \Delta\hat{\mathbf{x}}_i$ is a linear combination of the error in both coordinates of $\hat{\mathbf{x}}'_i$, and considering that these are linearly independent, we conclude that $d(\tilde{\mathbf{x}}'_i, \mathbf{l}') \sim \mathcal{N}(\mu_d, \sigma^2_d)$.

Next, we obtain the probability of a bounding box B' in one view belonging to the same object instance as a bounding box B in another view based on the distance of the centroid of B' to the epipolar line defined by the projection of the centroid of B via the corresponding fundamental matrix F. If $\mathrm{D} \sim \mathcal{N}(\mu_d, \sigma^2_d)$ is the random variable describing the distance of the centroid of B' to the epipolar line given by the centroid of B from the corresponding view, the probability of B and B' belonging to the same instance if D is at least d is given by $P(|\mathrm{D}| \geq d|B)$, which is the sum of the tails of the probability distribution of D. For a normal distribution, the probability is thus given by:

$$P(|\mathrm{D}| \geq d|B) = \operatorname{erfc}\left(\frac{d - \mu_d}{\sqrt{2}\sigma_d}\right) , \tag{3.6}$$

where erfc() is the complement of the error function. Eq. (3.6) can also be seen as the p-value under the hypothesis that B' is a match of B (under the assumption that the occurrence of threat objects is sparse within the imagery, giving rise to a

[1] This might not the case because the point in the first view is also an approximation of the object centre. The consideration of this source of error is left for future work.

simplified one-to-one / one-to-few matching problem). We refer to Eq. (3.6) as the multi-view epipolar confidence.

Eq. (3.6) can be used to get an interval of confidence of valid bounding boxes based on their distance to the epipolar line. Riffo *et al.* [187] explore a heuristic approach for choosing the size of this region. Another option is to combine Eqs. (3.3) and (3.6) to get a new extended confidence probability based on the original probabilities from the object detection model and the epipolar constraints between that view and the view containing B. This new confidence probability, which we call multi-view epipolar class confidence, is expressed as:

$$P(C = c, O, |\mathrm{D}| \geq d|B) = P(C = c, O)P(|\mathrm{D}| \geq d|B, C = c, O)\,, \tag{3.7}$$

where $P(|\mathrm{D}| \geq d|B, C = c, O)$ indicates that the distance-based probability is only considered for detected objects and it is given by Eq. (3.6). Eq. (3.7) explicitly states that the multi-view epipolar class confidence is not independent of the detected class c, thus indicating that the parameters of the normal distribution μ_d and σ_d^2 are class dependent. The option of the multi-view epipolar confidence being independent of the class, effectively converting Eq. (3.7) into

$$P(C = c, O, |\mathrm{D}| \geq d|B) = P(C = c, O)P(|\mathrm{D}| \geq d|B)\,, \tag{3.8}$$

is explored in the ablation studies (Section 3.4.3).

3.2.3 Multi-view Filtering

In single-view detection, the output of the model is filtered by its confidence score and redundant boxes are removed using NMS (or Soft-NMS). As an extension, we propose a post-processing algorithm that uses the epipolar constraints described in previous sections as an extra step before NMS. We refer to this algorithm as multi-view filtering and the general outline is presented in Fig. 3.4.

First, single view bounding box predictions $B_m = \{b_{m,i}\}$ with a confidence score greater than a threshold value t_s are obtained for a view m. For each $b_{m,i}$ with

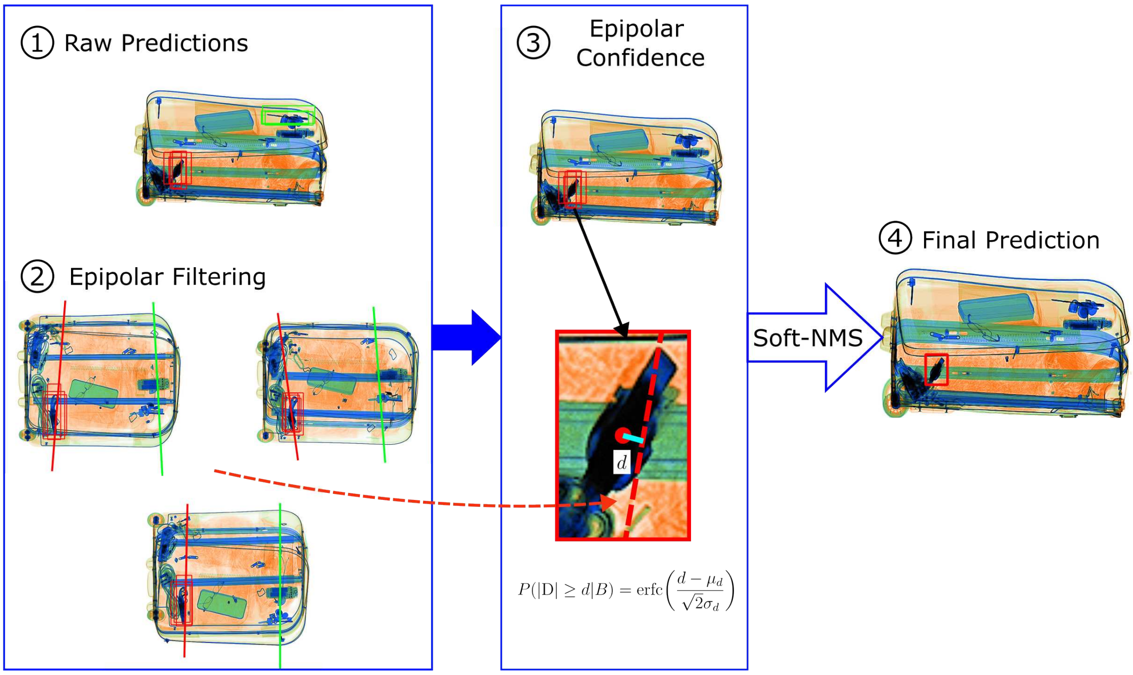

1 Raw Predictions
2 Epipolar Filtering
3 Epipolar Confidence
d
$P(|\mathrm{D}| \geq d|B) = \mathrm{erfc}\left(\frac{d-\mu_d}{\sqrt{2}\sigma_d}\right)$
Soft-NMS
4 Final Prediction

Figure 3.4: Multi-view epipolar filtering. This algorithm uses epipolar constraints for filtering invalid matches and as a confidence measure for Soft-NMS. *1:* Raw predictions (*i.e.*, before NMS) are obtained from an object detector. *2:* (a) An epipolar filtering step is used to remove invalid detections that do not have a same-class detection near the epipolar line in other views. *3:* From those valid bounding boxes, an epipolar confidence score is assigned, which depends on the minimum distance to the epipolar line from other views. *4:* Finally, NMS is applied using the epipolar and class confidences.

category c, we find a set of bounding boxes $B_{(m,i)\to n} = \{b_{(m,i)\to n,j}\}$ in a different view n with a multi-view epipolar confidence, defined in Eq. (3.7), satisfying:

$$P(C = c, O, |\mathrm{D}| \geq c) > rt_s\,, \tag{3.9}$$

where r is the minimum p-value of $b_{(m,i)\to n,j}$ as being a correspondence of $b_{m,i}$. These boxes are combined using NMS and the resulting bounding box $b_{(m,i)\to n,j}$ with the greatest multi-view epipolar class confidence is considered as the match of $b_{m,i}$ in the view n. If $B_{(m,i)\to n}$ is empty for all $n \neq m$, $b_{m,i}$ is disregarded, forming then the set $\hat{B}_m$ whose members have at least a matching bounding box in another view. Finally, for a dataset with N views, we combine the filtered single-view predictions $\hat{B}_m$ and the best match from other views into a single set of bounding boxes for each view m:

$$B^*_m = \hat{B}^m \cup \bigcup_{\substack{n=1 \\ n \neq m}}^{N} \bigcup_{b_{n,i} \in B_n} b_{(n,i) \to m} \,. \tag{3.10}$$

Redundancies in B^*_m are removed by Soft-NMS using their multi-view epipolar class confidence scores. While this multi-view filtering will mostly help in removing false positives, the use of two thresholds inEq. (3.9) (r for the p-value and t_s for confidence threshold) give a greater score to low confidence predictions closer to the epipolar line (that are sometimes removed after NMS) than higher confidence predictions that are not multi-view consistent, therefore aiding in the recovery of missed objects.

As an alternative, we can first filter the bounding boxes within the interval of confidence r (*i.e.*, removing the bounding boxes without a match in another view) and then filter them by their class probabilities (or vice-versa), applying NMS with class probabilities as weights. The only difference with the proposed approach in this work is that the confidence scores are not multiplied by their epipolar confidence (Eq. (3.6)), making $r = 1$ in Eq. (3.9). This technique is similar to the work of Riffo *et al*. [187] and it is subsequently explored in the ablation studies, showing that the proposed algorithm in this chapter yields superior overall detection performance.

3.3 Experimental Setup

In this section, the details about the dataset and the training of the object detection architectures used in this work are described.

3.3.1 Dataset

The dataset used in this chapter consists of conventional false-coloured X-ray security imagery from a Smith Detection dual-energy scanner with four views (three below and one at a side), as depicted in Fig. 3.5. An analysis of the composition from the dual energy modalities and the statistical distribution of the objects are presented in Appendices A and B. In this context, a *sample* refers to the set of all views of one bag. A total of 2,528 baggage items (10,112 images) were scanned and four object categories were identified. In total, there are samples of 1,090 firearm,

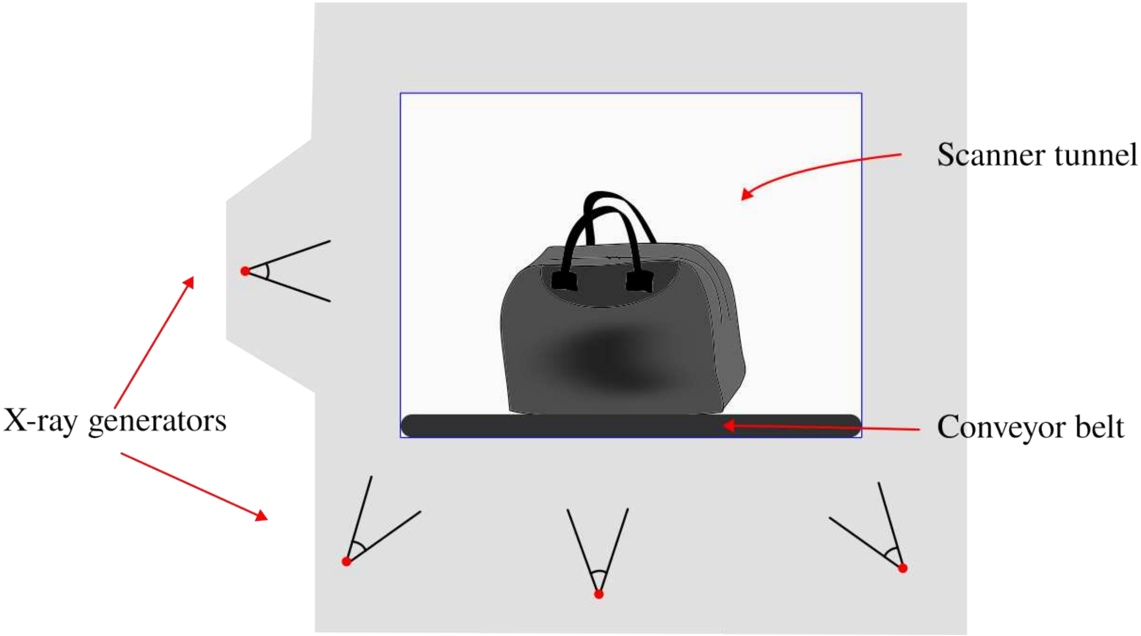


Figure 3.5: Diagram of the four-view X-ray scanner camera positions of the dataset used in this chapter (precise positions are unknown).

594 laptop, 1,184 knife and 166 camera items across all the scans. A split of 80% of the samples was used for model training and fundamental matrix estimation. These objects were manually annotated with bounding boxes across all views and a local index was assigned to identify the same object instance across all views. The dataset includes images with only one object and more challenging samples with two or more objects.

3.3.2 Object detection training details

Four object detectors using NMS or Soft-NMS during the post-processing step are trained for single-view object detection. All models are pre-trained on the COCO dataset and the detection performance is evaluated using the COCO detection metrics [88].

YOLOv3 [75]. The YOLOv3 is used because it is a fast detector that has shown superior performance in prior work on threat object detection in X-ray images [188]. The input images are square padded with a white background and resized to 544 × 544. The model is trained using Adam optimization [65] with a learning rate of 1×10^{-4}, weight decay of 0.0005, batch size of 8 and for 50 epochs. The learning rate is reduced by a factor of 10 after 15 and 30 epochs.

YOLOX-S [46]. A more recent version of the YOLO family of detectors, YOLOX, is also tested to verify the validity of the epipolar filtering in single-stage detectors. YOLOX is anchorless and uses a decoupled head for bounding box regression and class prediction. The small version, YOLOX-S, is trained using the original implementation but for 30 epochs (instead of 300) and with an initial learning rate of 1×10^{-3}.

Swin Transformer [7]. A Faster R-CNN [2] with a Swin Transformer backbone is used to validate our method against two-stage detectors. This architecture was trained for 12 epochs with an initial learning rate of 1×10^{-4}, decreasing it by a factor of 10 at epochs 9 and 11; apart from this, all other training details remain unchanged. We refer to this model simply as Swin Transformer.

FCOS [163]. Finally, the proposed multi-view filtering is validated against the FCOS detector, a fully convolutional one-stage anchorless detector that uses NMS as post-processing. FCOS is trained for 12 epochs with an initial learning rate of 1×10^{-3}, which is reduced by a factor of 10 during epochs 8 and 11. Similarly to YOLOX and Swin Transformer, all training remains unchanged except for the previous details.

During multi-view filtering, the confidence score threshold is set to 0.5, while the minimum p-value for epipolar filtering is 0.05. All models were trained using the MMDetection [224] framework and with an Nvidia GeForce 2080Ti.

3.4 Results

In this section, the results of the proposed methods for fundamental matrix estimation and multi-view filtering of predictions are reviewed. Ablation studies are carried out for object detection, modifying some parts of the multi-view filtering algorithm.

3.4.1 Fundamental Matrix Estimation

The performance of the proposed fundamental matrix estimation method, *i.e.*, using the centre of bounding boxes as correspondences, is reported in Table 3.1 as mean/standard deviation pairs of the distance (in pixels) from a corresponding centre to the epipolar line. This metric is chosen because the ground truth fundamental matrices between the views are unknown. The rows in Table 3.1 represent the *source* (or *from*) view whilst the columns are the *target* (or *to*) view, such that a value in the A row and C column represents the mean and standard deviation of the distance of a correspondence point in view C to the epipolar line that is obtained from the fundamental matrix and the corresponding point in view A, recalling that corresponding points are the bounding box centres of corresponding bounding boxes. It is observed that using bounding box centres as correspondences allows for an accurate fundamental matrix estimation since the mean distance of the centres to the epipolar line is less or slightly above a pixel, with standard deviations of around 5 pixels in most of the view pairs. A large standard deviation (∼70 pixels) is observed for firearms from views {A, D} to {B, C}, and vice-versa. The reason for this being the only case of a large standard deviation could be explained as the actual centre of the firearms not being too close to the bounding box centre as with the other classes, given that the shape of the firearm cannot be accommodated inside a bounding box (on the other hand, laptops, cameras and knives are better accommodated with a rectangular bounding box). Additionally, views A and D are similar to each other, while views B and C are closer between them. Despite this, the use of multi-view epipolar filtering improves detection accuracy for all classes, as will be seen in the ablation studies (Section 3.4.3).

Qualitative results of the fundamental matrix estimation are shown in Fig. 3.6. The left images show a ground truth bounding box while the right images show the p-value as a function of the distance to the epipolar line defined by the source images, given by Eq. (3.6). Values for μ_d and σ_d used for getting the p-value come from Table 3.1. It is seen that some objects such as knives have slightly wider dispersion. This can be explained by the greater variability of the position of knives in the baggage as compared with bigger objects. Also, the error associated with the mea-

Table 3.1: Fundamental Matrix Estimation Performance (μ_d/σ_d in pixels).

	Firearm				**Knife**			
	A	B	C	D	A	B	C	D
A		-0.241/70.15	0.273/71.10	-0.035/3.47		-1.072/3.23	0.559/3.63	-0.247/3.54
B	-0.175/70.17		0.242/3.98	-0.324/70.51	-0.166/3.63		-0.771/2.64	0.276/3.22
C	-0.338/71.72	-0.369/3.41		-0.277/71.27	0.549/3.27	1.244/4.99		-0.048/3.32
D	0.490/2.97	-0.487/71.25	0.509/71.7		0.244/4.99	-0.586/3.85	-0.063/3.28	

	Laptop				**Camera**			
	A	B	C	D	A	B	C	D
A		-0.294/5.69	0.648/5.72	-0.619/4.75		-1.612/4.874	0.146/4.72	-0.192/3.99
B	0.190/5.65		-0.918/5.58	0.280/5.07	1.474/6.00		1.966/8.16	2.989/9.24
C	1.271/6.26	-0.060/6.29		-0.797/6.54	0.667/5.02	0.273/6.87		-1.430/5.98
D	-0.922/4.67	0.406/5.14	0.479/6.53		0.585/4.60	-0.261/4.91	1.762/5.15	

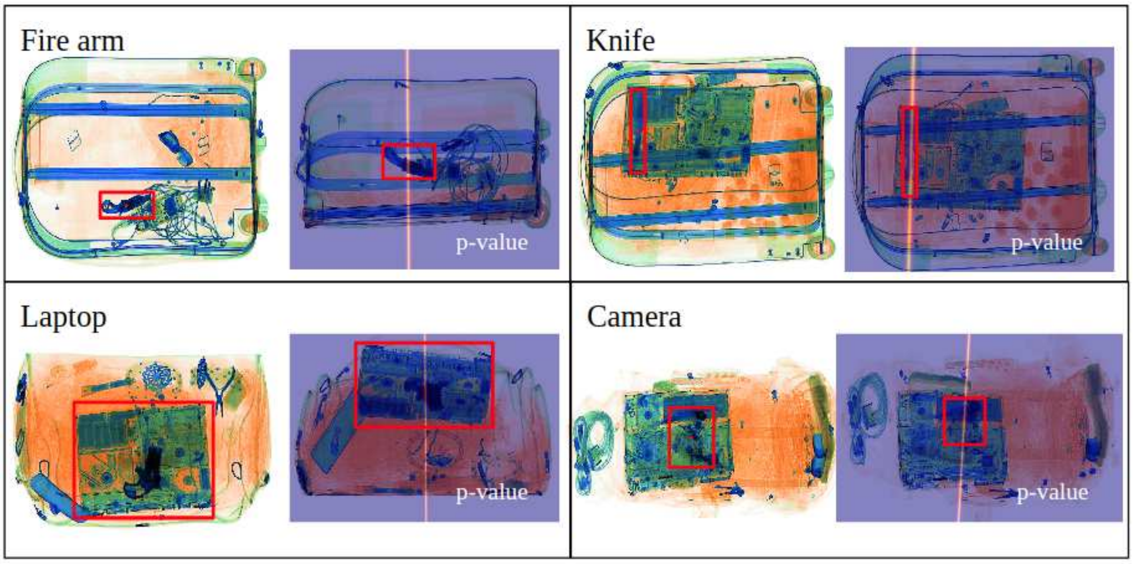


Figure 3.6: Results of fundamental matrix estimation per class. The right images for each category show the p-value of the position of candidate bounding box centroids with respect to the epipolar line defined by the left images in another view.

surement process (*i.e.*, the manual annotation process) is bigger for smaller objects. These results further validate the use of bounding box centres as approximations for inter-view correspondences.

3.4.2 Object Detection Performance

Object detection using multi-view filtering is compared against standard single-view detection. Table 3.2 shows the performance evaluation using COCO metrics for each class as well as metrics for all classes (AP_{100} is not included as our dataset only has up to three objects per image), with the best models, based on the AP metric, highlighted in yellow for each class and in red for the overall best model

Table 3.2: Multi-view Object Detection Results

Detector	Category	Method	AP	$AP_{0.5}$	$AP_{0.75}$	AP_S	AP_M	AP_L	AR_1	AR_{10}	AR_S	AR_M	AR_L
YOLOv3	Firearm	SV	0.670	0.983	0.816	-	0.681	0.630	0.743	0.747	-	0.744	**0.776**
		MV	**0.691**	**0.988**	**0.848**	-	**0.702**	**0.679**	**0.746**	**0.749**	-	**0.747**	0.775
	Laptop	SV	**0.705**	0.972	**0.886**	-	-	**0.705**	**0.770**	**0.772**	-	-	**0.772**
		MV	0.697	**0.973**	0.872	-	-	0.697	0.764	0.766	-	-	0.766
	Knife	SV	0.320	0.726	0.236	0.083	0.349	**0.175**	0.440	0.447	0.112	0.464	0.263
		MV	**0.382**	**0.800**	**0.322**	**0.125**	**0.412**	0.138	**0.455**	**0.463**	**0.154**	**0.478**	**0.287**
	Camera	SV	0.530	0.848	0.621	-	**0.700**	0.530	**0.605**	**0.605**	-	**0.700**	**0.605**
		MV	**0.546**	**0.881**	**0.633**	-	**0.700**	**0.546**	0.603	0.603	-	**0.700**	0.602
	All	SV	0.557	0.882	0.640	0.083	0.577	0.510	0.640	0.643	0.112	0.636	0.604
		MV	**0.579**	**0.910**	**0.669**	**0.125**	**0.605**	**0.515**	**0.642**	**0.645**	**0.154**	**0.641**	**0.608**
YOLOX	Firearm	SV	**0.785**	0.970	**0.924**	-	**0.797**	**0.843**	**0.808**	0.826	-	0.820	**0.891**
		MV	**0.785**	**0.971**	**0.924**	-	0.796	0.842	0.806	**0.827**	-	**0.821**	0.889
	Laptop	SV	**0.877**	**0.988**	**0.967**	-	-	**0.877**	**0.912**	**0.912**	-	-	**0.912**
		MV	0.875	**0.988**	**0.967**	-	-	0.875	0.911	0.911	-	-	0.911
	Knife	SV	0.471	0.774	0.504	0.201	0.501	0.156	0.512	0.518	0.229	0.530	0.600
		MV	**0.492**	**0.813**	**0.521**	**0.225**	**0.525**	**0.174**	**0.532**	**0.555**	**0.239**	**0.568**	**0.613**
	Camera	SV	0.624	0.856	0.671	-	**0.800**	0.624	0.659	0.659	-	**0.800**	0.658
		MV	**0.642**	**0.900**	**0.687**	-	**0.800**	**0.644**	**0.683**	**0.683**	-	**0.800**	**0.683**
	All	SV	0.689	0.897	0.766	0.201	0.699	0.625	0.723	0.729	0.229	0.717	0.765
		MV	**0.698**	**0.916**	**0.775**	**0.225**	**0.706**	**0.634**	**0.732**	**0.743**	**0.239**	**0.729**	**0.774**
Swin Transformer	Firearm	SV	**0.776**	0.988	0.936	-	**0.784**	**0.791**	**0.808**	**0.816**	-	**0.810**	**0.879**
		MV	**0.776**	**0.989**	**0.944**	-	**0.784**	0.789	**0.808**	**0.816**	-	**0.810**	0.876
	Laptop	SV	0.873	**0.992**	**0.982**	-	-	0.873	0.907	0.907	-	-	0.907
		MV	**0.876**	**0.992**	**0.982**	-	-	**0.876**	**0.909**	**0.909**	-	-	**0.909**
	Knife	SV	0.467	0.835	0.467	0.221	0.502	0.325	0.514	0.534	0.290	0.543	0.750
		MV	**0.483**	**0.864**	**0.476**	**0.255**	**0.518**	**0.341**	**0.527**	**0.559**	**0.317**	**0.568**	**0.775**
	Camera	SV	0.672	0.915	**0.846**	-	**0.800**	0.673	0.717	0.717	-	**0.800**	0.717
		MV	**0.681**	**0.925**	0.841	-	**0.800**	**0.681**	**0.730**	**0.730**	-	**0.800**	**0.730**
	All	SV	0.697	0.933	0.808	0.221	0.695	0.665	0.737	0.744	0.290	0.718	0.813
		MV	**0.704**	**0.942**	**0.811**	**0.255**	**0.701**	**0.672**	**0.743**	**0.753**	**0.317**	**0.726**	**0.822**
FCOS	Firearm	SV	0.776	0.969	0.936	-	0.789	0.789	0.806	0.810	-	0.805	**0.871**
		MV	**0.783**	**0.979**	**0.945**	-	**0.796**	**0.793**	**0.811**	**0.817**	-	**0.811**	0.879
	Laptop	SV	0.895	0.980	0.970	-	-	0.895	0.930	0.930	-	-	0.930
		MV	**0.900**	**0.990**	**0.980**	-	-	**0.900**	**0.933**	**0.937**	-	-	**0.937**
	Knife	SV	0.425	0.734	0.445	0.175	0.461	0.060	0.477	0.481	0.188	0.494	0.463
		MV	**0.458**	**0.811**	**0.466**	**0.235**	**0.492**	**0.076**	**0.508**	**0.522**	**0.261**	**0.532**	**0.625**
	Camera	SV	0.670	0.860	0.813	-	**0.800**	0.670	0.714	0.714	-	**0.800**	0.714
		MV	**0.710**	**0.910**	**0.866**	-	**0.800**	**0.710**	**0.756**	**0.777**	-	**0.800**	**0.777**
	All	SV	0.691	0.886	0.791	0.175	0.683	0.604	0.732	0.734	0.188	0.700	0.744
		MV	**0.707**	**0.915**	**0.808**	**0.235**	**0.690**	**0.619**	**0.746**	**0.758**	**0.261**	**0.709**	**0.803**

(FCOS with multi-view epipolar filtering). SV refers to single-view detection and MV to detection processed with multi-view filtering. Multi-view epipolar filtering increases the detection performance across all detectors, with a maximum increase of 2.2% of the average precision metric (YOLOv3), a maximum increase of 2.9% of average precision with a fixed IoU of 0.5 and a maximum increase of 2.4% on the recall (FCOS) with all possible detections (AR_{10}). Although the multi-view epipolar filtering increases the overall performance, its impact differs. For instance, the detection performance on firearms is not always improved, as the AP metric for the YOLOX and Swin Transformer detectors does not change, which can be partially because of the high variance in the fundamental matrix computation of firearms (Section 3.4.1). Additionally, the performance on laptop detection does not exhibit a great improvement when using our multi-view epipolar filtering, with slight decreases in the YOLOv3 and YOLOX detectors (both being anchor-based one-stage detectors). In this regard, the larger size of laptops makes that a small variation in the predicted bounding boxes leads to a greater decrease of the IoU; it can also be noted that laptop detection is almost saturated, with $AP_{0.5} > 97\%$ in all detectors. The performance of knife and camera detection increases in all cases with the use of epipolar filtering. Finally, it is seen that the FCOS detector with multi-view epipolar filtering achieves the best performance by means of the AP metric. However, the Swin Transformer detector achieves a better $AP_{0.5}$, which can be more useful in the application context. The YOLOv3 detector performs the poorest, showing its unreliability for extensive threat item detection.

A comparison between single and multi-view detections using the proposed method for epipolar filtering is shown in Figs. 3.7 to 3.10. The improvement of the precision metrics is associated with the elimination of false positives that do not fulfil the epipolar constraints. Some examples of this elimination are shown in Figs. 3.7 to 3.9, where incorrectly identified knives and firearms are eliminated after multi-view filtering. On the other hand, some instances are difficult to detect given that other objects, such as laptops or tablets overlap them. In this context, multi-view epipolar filtering allows for the identification of lower-confidence objects as long as a confident enough object is found in another view. For instance, Fig. 3.9 shows a scanned suitcase with a laptop overlapping a firearm, which is intentionally hidden. It is seen that using multi-view epipolar filtering allows for the identification of a firearm in other views, that are not identified in the single-view detection. However, it is also noted that our algorithm exhibits a greater amount of redundant detections around a correctly identified object. Although all redundant bounding boxes in Fig. 3.9 are near the epipolar line of the firearm, and include the firearm within their boundaries, they are different enough to not be eliminated by the NMS algo-

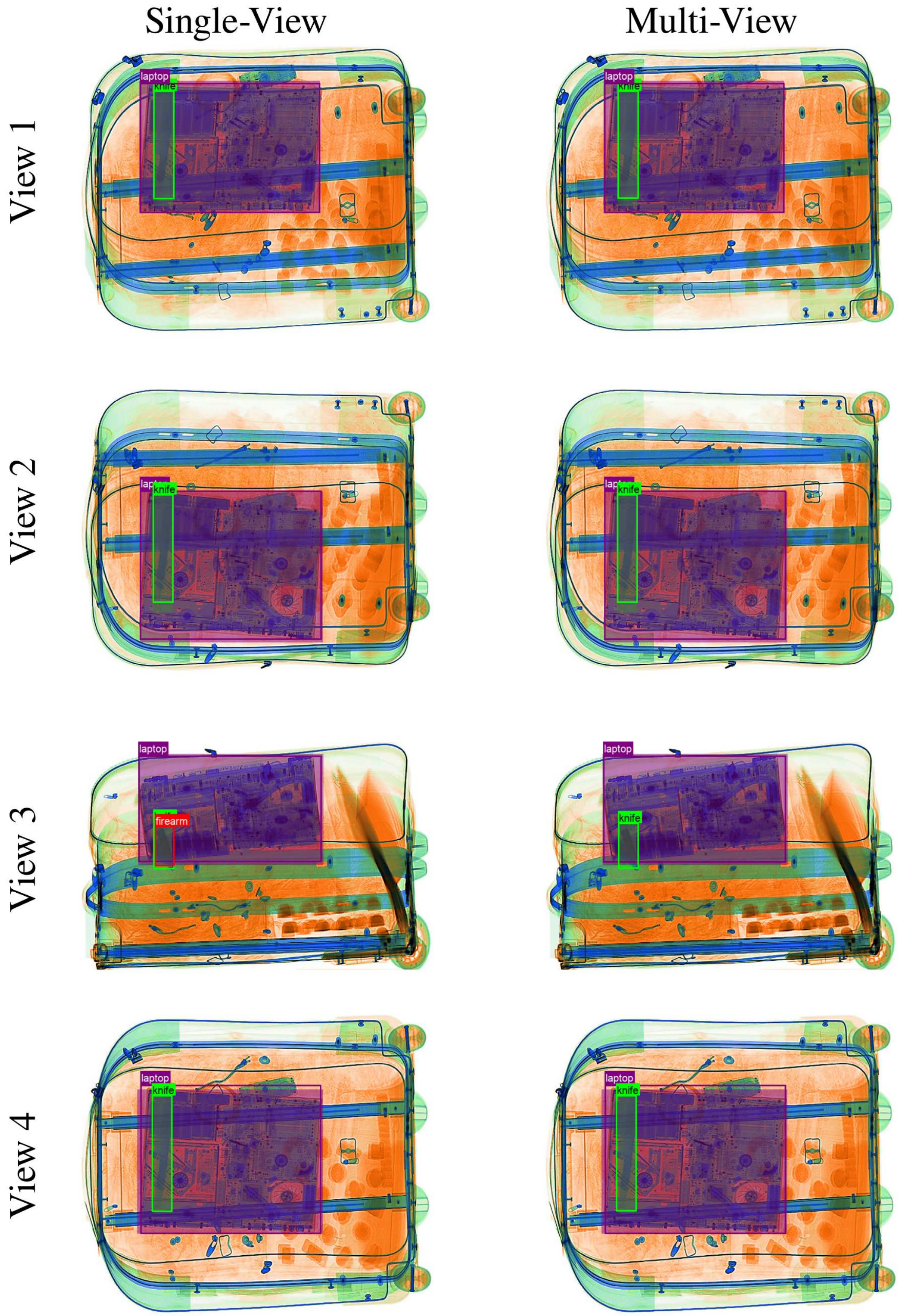


Figure 3.7: Comparison between single view and multi-view detection. Example of a removed firearm false positive in single-view detection.

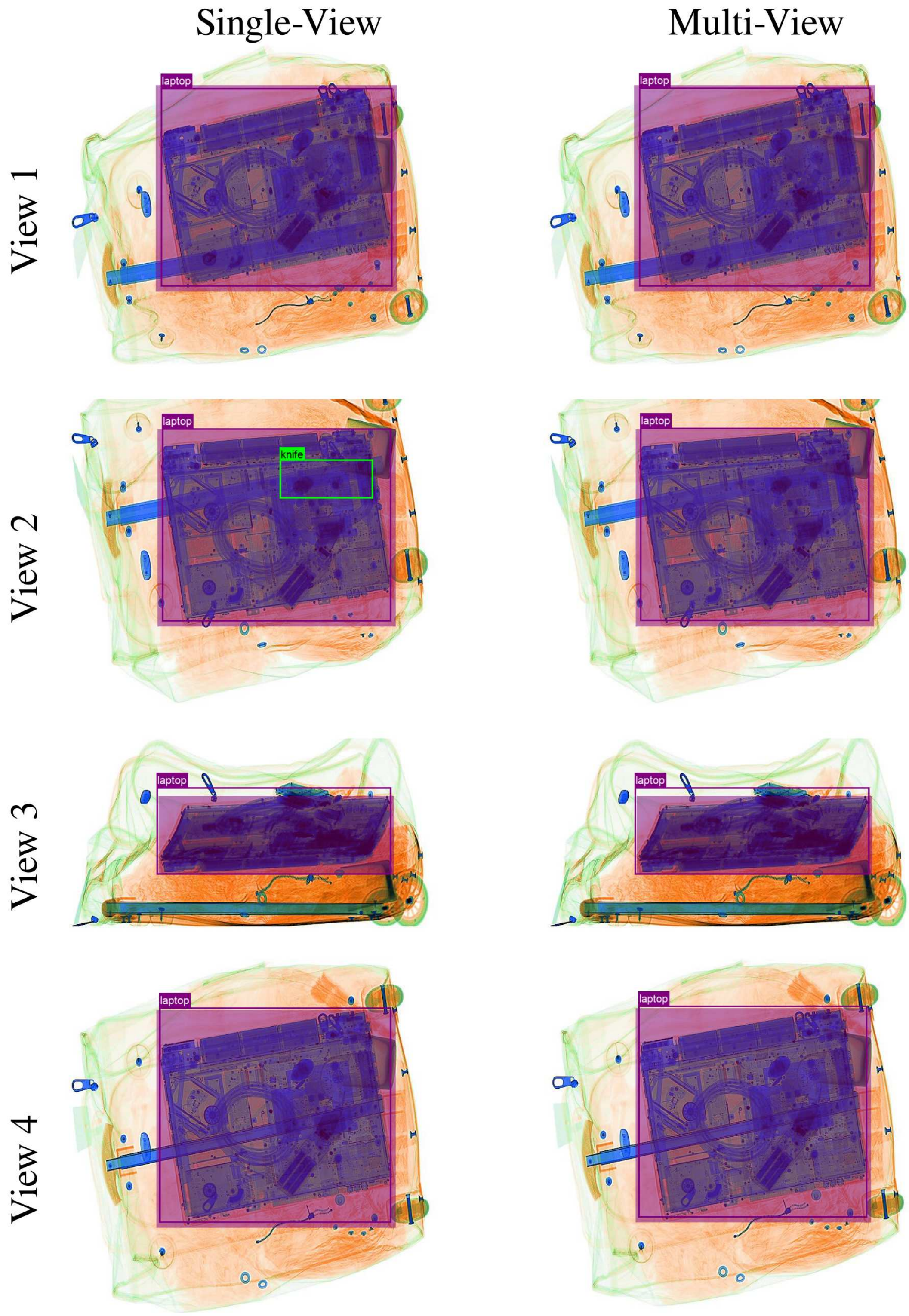


Figure 3.8: Comparison between single view and multi-view detection. Example of a removed knife false positive in single-view detection.

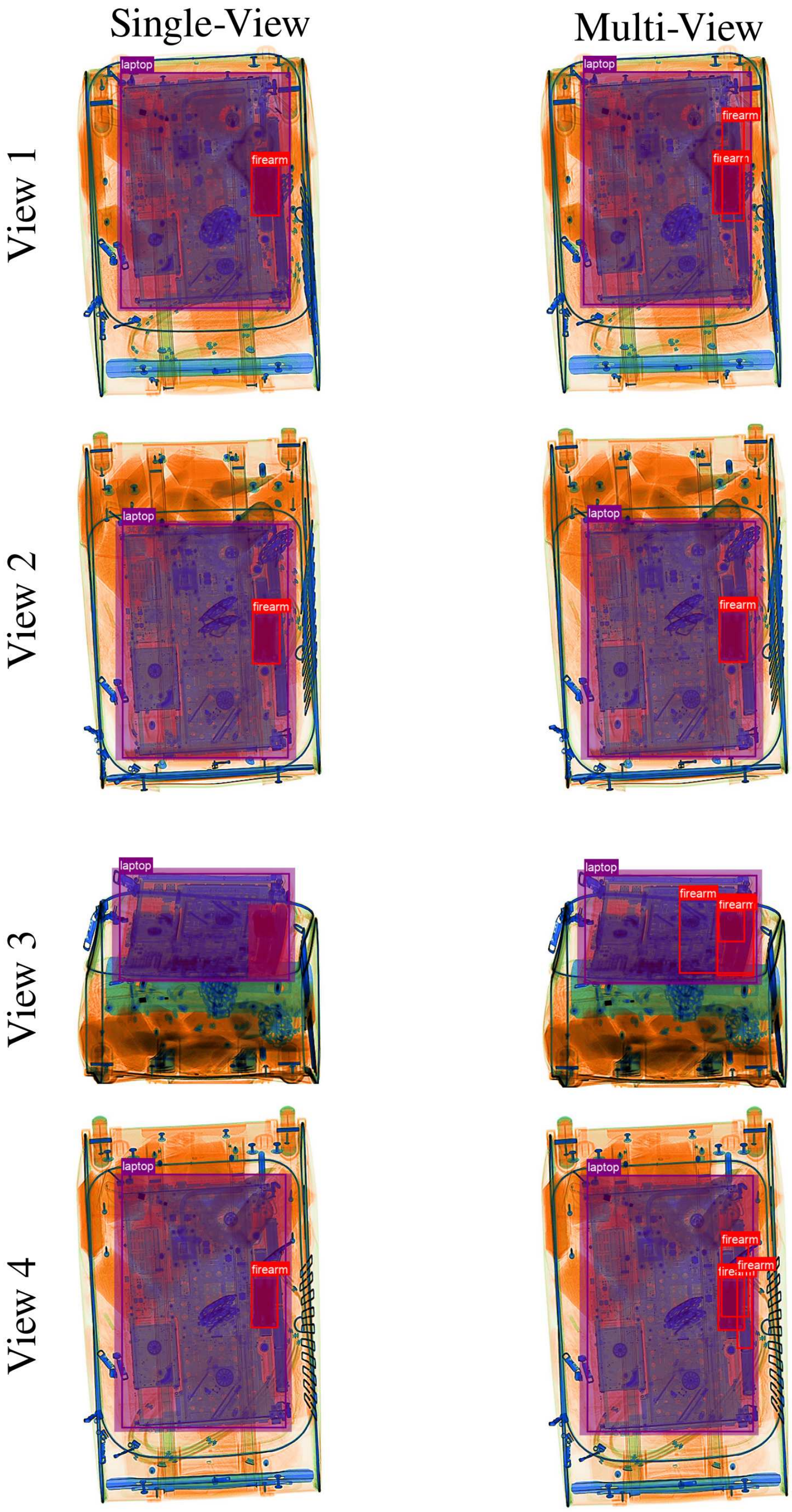


Figure 3.9: Comparison between single view and multi-view detection. Example of a missed firearm in single-view detection.

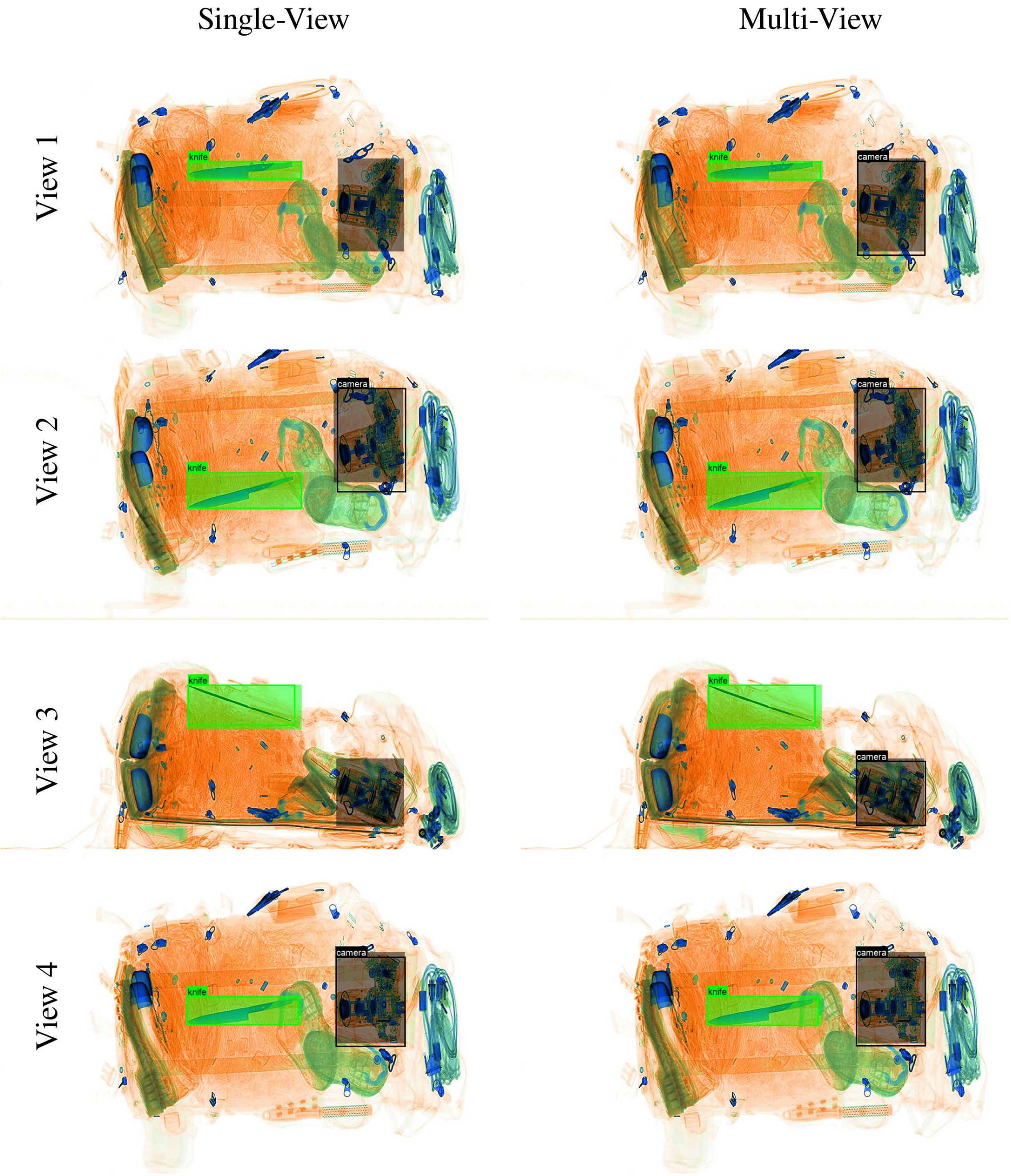


Figure 3.10: Comparison between single view and multi-view detection. Example of a missed camera in single-view detection.

rithm. Henceforth, this suggests the addition of further algorithms for keeping the same number of detected objects across views while maximising the detection confidence, which is a matter of future work. Finally, in not highly cluttered scenes, our epipolar filtering method may correctly identify missed objects, such as the camera detection in the third view of Fig. 3.10.

3.4.3 Ablation Studies

In this section, the performance of our model under our methodological choices is assessed experimentally. We focus on three main parts of the method: modelling the distance between a bounding box and an epipolar line as a normal distribution with a non-zero mean μ_d (Section 3.2.2), the use of the multi-view epipolar confidence in Eq. (3.7) for Soft-NMS (*i.e.*, using the distance to the epipolar line as a measure of confidence) and getting different fundamental matrices for each category independently (Eq. (3.7) vs. Eq. (3.8)). The results of these ablations for all detectors are shown in Table 3.3. In all cases, multi-view filtering is performed.

First, the choice of modelling our distance of the bounding box to the epipolar line with a biased estimator, *i.e.*, $\mathrm{D} \sim \mathcal{N}(\mu_d, \sigma_d^2)$, with μ_d not necessarily 0, is validated. To do so, a test is performed assuming the distance follows a normal distribution with a 0 mean, suggesting that the centre of the object is in fact the centre of the bounding box. As can be seen in the first and third rows of each detector in Table 3.3, using an unbiased estimation of the distance, multi-view filtering without epipolar filtering gives a slightly worse performance in most metrics. Such minor changes in performance are a consequence that the mean distance given by our fundamental matrices is very close to zero, as seen in Table 3.1, with only the standard deviation varying significantly across classes. The reason behind using a biased estimator for the distance is that the mean μ_d serves as a correction of the unknown distance Ψ_{obj} in Eq. (3.1) of the centres of the bounding box with the actual projection of the object centre in the image plane. Subsequently, if this bias is not induced, the multi-view filtering algorithm searches for matches in a region further away from the actual match.

Secondly, the multi-view filtering algorithm by epipolar confidence (Section 3.2.3) is compared against simple class confidence filtering. In this case, instead of using the relation in Eq. (3.9) for filtering and performing NMS, we test a model that looks in the second view for matching bounding boxes within an interval, disregarding those without a match (*i.e.*, epipolar filtering), but only using the class confidence as weights for Soft-NMS, as in single view detection. Riffo *et al.* [187] partially address this method but choosing the interval of confidence heuristically. The results are

Table 3.3: Results of ablation studies testing our methodological choices

Detector	p-value	$\mu_d \neq 0$	w/class	**AP**	$\mathbf{AP_{0.5}}$	$\mathbf{AP_{0.75}}$	$\mathbf{AP_S}$	$\mathbf{AP_M}$	$\mathbf{AP_L}$	$\mathbf{AR_1}$	$\mathbf{AR_{10}}$	$\mathbf{AR_S}$	$\mathbf{AR_M}$	$\mathbf{AR_L}$
YOLOv3			✓	0.577	0.904	**0.671**	0.085	0.601	0.514	0.641	0.643	0.107	0.636	0.604
		✓	✓	0.577	0.904	0.669	0.094	0.601	**0.515**	0.641	0.643	0.110	0.637	0.607
	✓		✓	0.576	0.909	0.666	0.099	0.569	0.512	0.640	0.644	0.132	0.606	0.606
	✓	✓		0.576	0.907	0.670	0.105	0.569	0.512	0.641	0.644	0.137	0.606	0.605
	✓	✓	✓	**0.579**	**0.910**	0.669	**0.125**	**0.605**	**0.515**	**0.642**	**0.645**	**0.154**	**0.641**	**0.608**
YOLOX			✓	0.695	0.912	0.770	0.216	0.703	0.628	0.730	0.738	0.256	0.724	0.769
		✓	✓	0.695	0.912	0.770	0.216	0.703	**0.628**	0.730	0.738	0.256	0.724	0.769
	✓		✓	**0.698**	**0.916**	**0.775**	**0.205**	**0.707**	**0.634**	**0.732**	**0.743**	**0.215**	**0.729**	**0.774**
	✓	✓		0.696	0.917	0.772	0.196	0.704	0.628	0.731	0.740	0.220	0.723	0.769
	✓	✓	✓	**0.698**	**0.916**	**0.775**	0.203	0.706	**0.634**	**0.732**	**0.743**	0.212	**0.729**	**0.774**
Swin Transformer			✓	0.699	0.938	0.811	0.235	0.698	0.666	0.739	0.749	0.317	0.722	0.808
		✓	✓	0.699	0.938	0.811	0.235	0.698	0.666	0.739	0.749	0.317	0.722	0.808
	✓		✓	**0.704**	0.941	**0.812**	0.254	**0.701**	**0.672**	**0.743**	**0.753**	**0.317**	**0.726**	**0.826**
	✓	✓		0.699	0.939	0.805	0.240	0.698	0.666	0.739	0.748	0.315	0.721	0.811
	✓	✓	✓	**0.704**	**0.942**	0.811	**0.255**	**0.701**	**0.672**	**0.743**	**0.753**	**0.317**	**0.726**	0.822
FCOS			✓	0.684	0.893	0.776	0.224	**0.691**	0.592	0.723	0.731	0.263	0.710	0.747
		✓	✓	0.684	0.893	0.776	0.224	**0.691**	0.592	0.723	0.731	0.263	**0.710**	0.747
	✓		✓	0.706	0.913	0.801	**0.239**	0.688	**0.619**	0.745	0.756	**0.263**	0.707	**0.803**
	✓	✓		0.669	0.875	0.758	0.233	0.683	0.580	0.709	0.714	0.273	0.702	0.733
	✓	✓	✓	**0.707**	**0.915**	**0.808**	0.235	0.690	**0.619**	**0.746**	**0.758**	0.261	0.709	**0.803**

shown in the first and second rows of Table 3.3 for each detector, with both biased and unbiased estimators of the distance. Again, this method performs poorly against the superior performance offered by the approach presented in this chapter. This is explained by noting that the use of the multi-view epipolar confidence defined in Eq. (3.7) gives greater weights to bounding box detections that are closer to the epipolar line, resulting in higher quality bounding boxes after NMS.

Finally, the use of a single fundamental matrix for all objects without considering the categories is also tested. This approach is an implementation of the independence of distance of the bounding box centre to the epipolar line and the class of the object, as in Eq. (3.8). The performance of the estimation of such matrix is presented in Table 3.4. It can be observed that although the μ_d values are also close to zero, the standard deviation is larger for all view pairs. The performance of the proposed multi-view epipolar filtering using this matrix is presented in the fourth row for each detector in Table 3.3. In all cases, the performance drops in all metrics, with a maximum decrement of 3.8% for the FCOS detector. As with the decision of using a $\mu_d \neq 0$, the motivation of using different fundamental matrices for each class relies

Table 3.4: Fundamental Matrix Estimation Performance, no Categories (μ_d/σ_d).

	A	B	C	D
A		-0.342/43.15	0.595/66.52	-0.314/51.04
B	-0.555/42.46		-0.144/51.53	0.197/66.90
C	0.612/66.80	-0.039/51.34		0.304/43.11
D	-0.662/51.22	-0.056/66.71	0.127/43.11	

on the unknown nature of how the actual object centre differs from the bounding box centre, which is modelled by the different means and standard deviations of each categorical fundamental matrix.

3.5 Conclusion

In this chapter, a new multi-view filtering approach using epipolar constraints as an additional confidence probability for Soft-NMS has been developed. The distance of bounding box centroids from corresponding epipolar lines is modelled as a random variable with a normal distribution and non-zero mean. The p-value of the distance with respect to that distribution is used as a new confidence probability for NMS post-processing. Furthermore, the estimation of the fundamental matrix by making use of ground truth object annotations available from object detector model training instead of actual correspondences is explored.

It is shown that using bounding box centroids as point correspondences across different views allows for high-quality estimation of the fundamental matrix, which is validated by measuring the distance of the bounding box centre to its corresponding epipolar line. The proposed approach increases the average precision of the MS-COCO metric by 2.2%, 0.9%, 0.7% and 1.6% for the YOLOv3, YOLOX, Faster R-CNN with a Swin Transformer backbone and FCOS detectors, respectively, and by 2.8%, 1.9%, 0.9% and 2.9% when using a fixed IoU of 0.5 ($\text{AP}_{0.5}$) for the same detectors, without affecting the recall. Additionally, it is found that the proposed method outperforms the approach of simply constraining the bounding boxes to a maximum distance to the epipolar line. These results show that the use of epipolar constraints for multi-view object detection is a key contribution to decreasing false positives and improving detection performance in the context of cluttered X-ray security imagery. Multi-view epipolar filtering is the first method to estimate the epipolar geometry without camera poses or prior knowledge of point correspondences and use it to get multi-view consistent object detections.

CHAPTER 4

Multi-view Vision Transformers for Object Detection

Object detection has been thoroughly investigated during the last decade using deep neural networks [2, 3, 8, 46, 72, 75, 139, 158, 163, 165, 225]. However, the fusion of multiple concurrent views of the same scene to improve detection performance has not received much attention (see discussion in Section 2.2.8). As established in Chapter 3, in scenarios where objects may appear in obscure or very intricate poses from certain viewpoints, the use of differing simultaneous views can improve object detection.

Therefore, in this chapter, a multi-view fusion network to enrich the backbone features of standard object detection architectures across multiple sources and target viewpoints is proposed. The presented method, named Multi-View Vision Transformers (MVViT), consists of a Transformer decoder for a target view that combines the remaining source view feature maps. In this way, the feature representation of a target view can be aggregated by the features from the remaining source views through an attention mechanism. The MVViT architecture is an *add-on* sub-network that is included in the detector backbone, meaning that it is detector-agnostic. The performance of MVViT across leading contemporary object detectors, namely YOLOX [46], Deformable DETR [47] and Swin Transformer [7], is assessed, comparing standard single view performance against the addition of the proposed multi-view Transformer architecture. The addition of an MVViT layer achieves a 3% increase of the COCO AP over a four-view X-ray security dataset (Section 3.3.1) and a slight 0.7% increase on a seven-view pedestrian dataset [106] presenting several occlusions and different fields of view. It is demonstrated that integrating different views using attention-based networks improves the detection performance of multi-view datasets. The work of this chapter has been published in the following peer-reviewed publication:

> Brian K.S. Isaac-Medina, Chris G. Willcocks and Toby P. Breckon. "Multi-view Vision Transformers for Object Detection", in *IEEE International Conference on Pattern Recognition (ICPR)*, pp. 4678-4684, 2022.

4.1 Introduction

Multi-view object detection refers to localising objects of interest given multiple images of the same scene where the viewpoints of each image may be either fully or partially overlapping. In this context, these views can be used in conjunction to improve detection performance but the investigation of deep neural network architectures that specifically exploit this condition remains limited.

As seen in Section 2.2.4, contemporary detectors consist of three subnetworks: backbone, neck and head. The backbone is responsible for extracting the feature maps which are usually taken from high-accuracy image classification networks, such as VGG [63], ResNet [61] and Darknet [158]. Some detectors include a subnetwork, sometimes called the neck, that is used to aggregate features from different layers of the backbone. The head of the detector localises the objects based on the feature maps from the backbone (or neck). A trend that is arising in the computer vision context is the implementation of the Transformer architecture [128]. The basic building block of the Transformer encoder is a self-attention layer followed by a feed-forward layer. Similarly, the Transformer decoder has a self-attention layer and a feed-forward layer, but it also includes an additional attention layer where the source sequence is the encoder output. Carion *et al*. [8] proposed the Detection Transformer

(DETR), the first architecture that implements a Transformer for object detection, using it as the head of the detector. Zhu *et al.* [47] further improved DETR by using deformable attention. A recent successful implementation of the Transformer for image classification is the Vision Transformer (ViT) by Dosovitskiy *et al.* [45]. This architecture is further discussed in Section 2.2.7.

In certain circumstances, object detection using multiple concurrent views of the same scene is possible. In this context, detection accuracy is evaluated on each view independently, although objects can be predicted using the views jointly. This is of interest in scenarios where objects can be highly occluded in one view, but are clearer in another view, such as in multi-camera visual surveillance, autonomous vehicle sensing solutions and multi-view X-ray security screening. Although some works have addressed multi-view object detection [173, 174, 191], including the use of epipolar geometry in Chapter 3, the detailed consideration of this task remains fairly limited. Furthermore, the use of recent deep learning architectures based on attention has not been investigated thoroughly, leading to the proposal of a novel architecture based on a Transformer decoder that uses the feature representations across multiple concurrent views to improve detection accuracy.

In this chapter, multi-view object detection is addressed by using such a Transformer-based architecture to combine the intermediate features from multiple concurrent views using the backbone of a standard object detection architecture. The fusion of these features is carried out by a Transformer decoder using the target view feature vectors as queries attending to the features from a source view. In this sense, the feature representation is aggregated with information from other views, making it aware of the 3D scene geometry. The Transformer decoder is applied to each view, so all views are target and source at the same time. For scenarios with more than two views, we propose to account for the feature maps of each of the source views via concatenation. This architecture is named Multi-view Vision Transformers (MVViT) and the general outline is shown in Figure 4.1. The key contributions of this chapter are as follows:

– A novel Transformed-based architecture for multi-view object detection. The proposed architecture aggregates the feature representation of each view hence

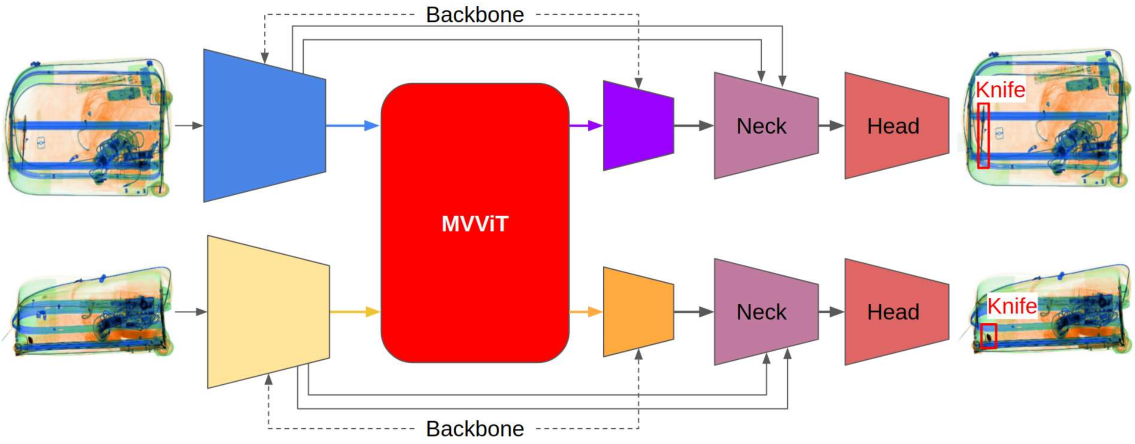


Figure 4.1: Illustrating the Multi-view Vision Transformer to create 3D scene geometry aware feature representations.

constructing a joint feature representation with awareness of the underlying 3D scene geometry.

- Consideration of three modern object detection architectures, namely YOLOX [46], Deformable DETR [47] and Swin Transformers [7], where MVViT is integrated. It is shown that MVViT can improve multi-view object detection for all detectors, demonstrating to be detector agnostic.

- Improved multi-view object detection performance compared to a single view baseline, for both a multi-camera surveillance dataset (+0.7% COCO AP, +0.7% COCO $\mathrm{AP}_{0.5}$) and an X-ray security imagery dataset (+3.0% COCO AP, +1.9% COCO $\mathrm{AP}_{0.5}$).

4.2 Multi-view Vision Transformer

This chapter implements the Transformer decoder architecture to leverage multiple viewpoint feature maps to create a feature representation with implicit awareness of the underlying 3D scene geometry. The proposed method, the Multi-view Vision Transformer (MVViT), acts as an extra layer within the backbone of the existing baseline detection architecture, and it is detailed in Figure 4.2.

Since the aim of MVViT is to create stronger feature representations that integrate multi-view feature maps, it is added to the backbone of the detector. While

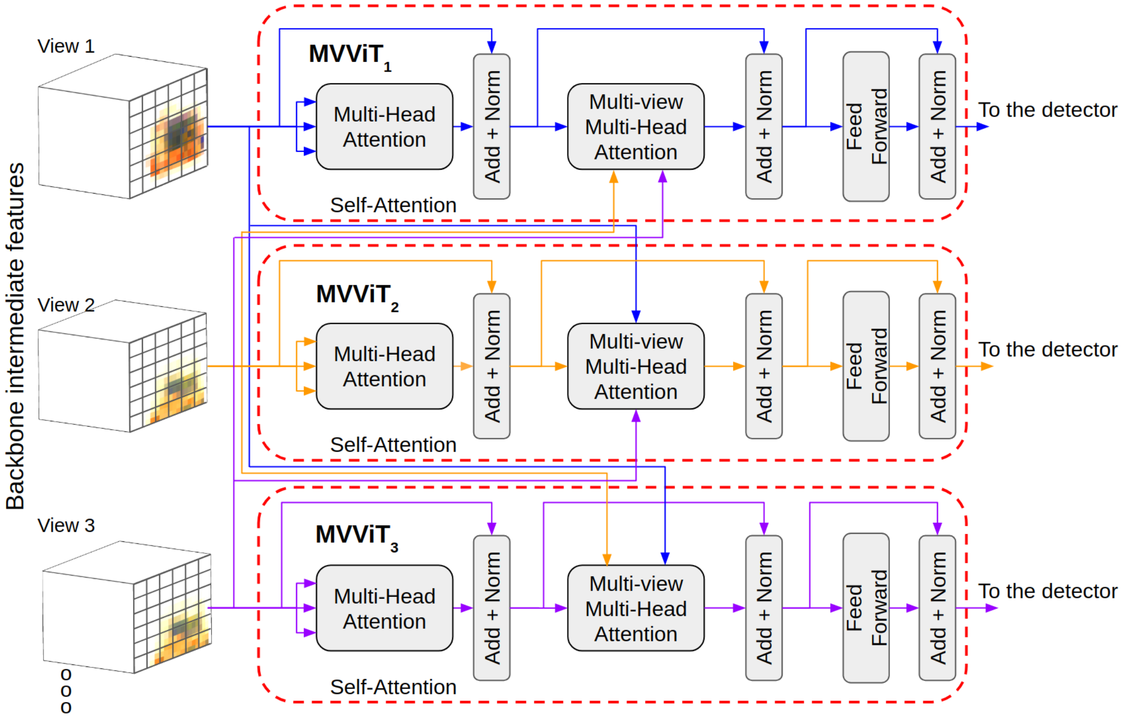


Figure 4.2: MVViT for Object Detection: architectural design and overview.

deeper layers in the backbone encode high-level features, a relatively large spatial resolution is necessary. For this reason, MVViT is added in an intermediate layer of the backbone. For each view $i = 1, ..., v$, MVViT applies a Transformer decoder taking the intermediate feature map $z_i^l \in \mathbb{R}^{W' \times H' \times C}$ as input and the remaining views feature maps z_j^l, $j \neq i$ as source views for the attention layer. Following the ViT architecture, each decoder comprising MVViT is composed of a multi-head self-attention layer, a multi-head attention module and a feed-forward network consisting of two linear layers with internal dimension d_f. All of the sub-modules use residual connection followed by layer normalisation [226].

The attention mechanism, which is the basic building block of Transformers, can be described as a weighted sum based on a similarity function. Given N query d_k dimensional vectors embedded in the matrix $Q \in \mathbb{R}^{N \times d_k}$ and M pairs of key and value matrices $K \in \mathbb{R}^{M \times d_k}$ and $V \in \mathbb{R}^{M \times d_v}$ of d_k and d_v dimensional vectors, the attention mechanism is described as:

$$\text{Attention}(Q, K, V) = \text{sim}\,(Q, K)\,V\,, \tag{4.1}$$

where $\mathrm{sim}(\cdot, \cdot)$ is a similarity function. A popular choice for the similarity function is the scaled dot product followed by a $softmax$ operation, that is:

$$\mathrm{sim}(Q, K) = \mathrm{softmax}\left(\frac{QK^\intercal}{\sqrt{d_k}}\right) . \tag{4.2}$$

Transformers define a multi-head attention (MHA) mechanism, where the attention inputs are linearly projected h times and attention is applied on each projection. This can be written as:

$$\begin{aligned} \mathrm{MHA}(Q, K, V) &= \mathrm{concat}\left(head_1, ..., head_h\right) \mathbf{W}^O , \\ head_i &= \mathrm{Attention}\left(Q\mathbf{W}_i^Q, K\mathbf{W}_i^K, V\mathbf{W}_i^V\right) , \ i = 1, ..., h , \end{aligned} \tag{4.3}$$

where $W_i^Q \in \mathbb{R}^{d_k \times d_m}$, $W_i^K \in \mathbb{R}^{d_k \times d_m}$, $W_i^V \in \mathbb{R}^{d_v \times d_m}$ and $W^O \in \mathbb{R}^{hd_m \times d_k}$ are learnable linear projections.

The feature map z_i^l can be seen as a $W' \times H'$ grid of feature vectors that serve as a sequence input for the decoder. These feature vectors are the object queries in the attention functions and an attention map of the source view is obtained for each of them. To achieve this, the original implementation of the Transformer is modified to use batched matrix multiplications in Equation (4.3) instead of being flattened to a 1D sequence.

In order to account for cases with more than one source view (*i.e.*, multi-view datasets with at least 3 concurrent views), the source views are concatenated in the feature dimension such that $\mathcal{V}_i = \mathrm{concat}(\{z_j^l\}_{j \neq i}) \in \mathbb{R}^{W' \times H' \times (v-1)C}$; then, the multi-view MHA (MVMHA) is given by:

$$\mathrm{MVMHA}(z_i^l, \{z_j^l\}_{j \neq i}) = \mathrm{MHA}\left(z_i, \mathcal{V}_i, \mathcal{V}_i\right) . \tag{4.4}$$

In this context, the target view attends to the source views at the same time, making it possible to dampen the attention from views where object instances do not appear in a source view with an overlapping field of view. Regarding the increased complexity of using multiple views, it is worth noting that the only impact would be in the size of the learnable linear projections of the key and value matrices in

Eq. (4.3). Therefore, the size of the model increases linearly with the number of views, such that for a dataset with $v > 2$ views, the model parameters increase by $2(v-2)Cd_m$, while the only extra computational burden is found in the matrix multiplication of $\mathcal{V}_i$ with $\mathbf{W}_i^K$ and $\mathbf{W}_i^V$. While this is not significant for the evaluated datasets, it might become a problem for denser views. In that case, a method to reduce the dimensionality of the concatenated feature maps could be used, such as 1×1 convolutions. Finally, no additional formulation is used for handling non-overlapping fields of view.

4.3 Experimental Setup

The evaluation of the performance of MVViT is based on two different multi-view datasets (Section 4.3.1), with implementation details presented for repeatability (Section 4.3.2) and measured using the MS-COCO detection metrics [88].

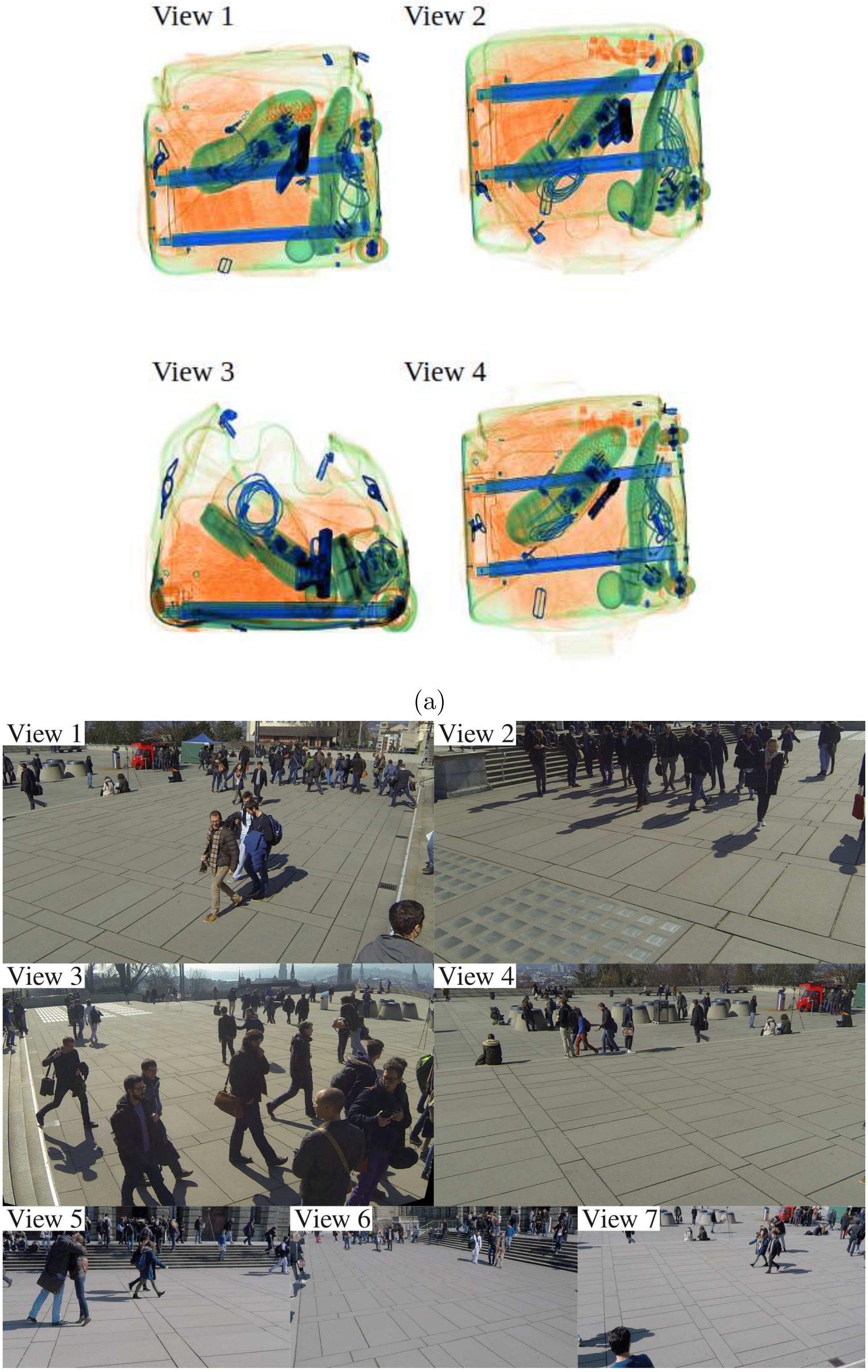
4.3.1 Datasets

X-ray-Quad: same X-ray security dataset used in Chapter 3, it consists of false-coloured cabin baggage security imagery from a Smith Detection security scanner with four views (Fig. 4.3a). A total of 10,112 images were scanned and four object categories were identified (4,260 firearms, 2,376 laptops, 4,736 knives and 664 cameras). A split of 80% for training and 20% for testing is used. To assess the impact of the number of viewpoints, a partition X-ray-Dual is also assessed, with only two perpendicular views (views 1 and 3, Fig. 4.3a).

Wildtrack: the Wildtrack seven-camera HD dataset [106] comprises a set of 7 outdoors concurrent videos from different points of view with only one class, namely the person category (Fig. 4.3b). This dataset includes scenarios where instances may appear in one view but not in the other. A total of 2,240 images and 33,962 object instances accounting for all views were used for training and 560 images and 8,571 object instances were used for validation. Originally, this dataset was created to use the common floor plane among the views as a geometry constraint to improve detection accuracy. For this reason, only people lying in the same floor plane are annotated.

4.3.2 Implementation details

In order to assess the performance of MVViT in different detectors, YOLOX-S [46], Deformable DETR [47] and Swin Transformer [7] architectures are used as baselines. The Swin Transformer backbone is used in conjunction with a Faster-RCNN architecture [2], similar to the original work. MixUp, Mosaic and Random Affine augmentations were removed in the YOLOX-S implementation since they are not multi-view consistent. In order to avoid an increased performance due to having larger datasets in the implementation of the MVViT, the same datasets were used when comparing to the single-view (sv) baselines, with the difference that different views from the same when are used to create the 3D aware features in MVViT layers. Input images for YOLOX-S are square padded (with a white background for X-ray datasets and a grey background for the Wildtrack dataset) and resized to 640 × 640, while the input images for Deformable DETR and Swin Transformer are kept to a maximum size of 1333 for the X-ray-Dual dataset and 800 for X-ray-Quad and Wildtrack datasets. MVViT is applied before the fourth CSP block of the YOLOX-S backbone (Modified CSPNet v5 [73]), after the fourth convolutional block (*conv4*) of the Deformable DETR backbone (ResNet-50 [61]) and before the fourth stage swin block of the Swin Transformer [7]. We use 8 heads for the MHA modules, internal decoder dimension $d_k = 512$ and feed-forward dimension $d_f = 2048$. ReLU activations are used and a dropout with a rate of 0.1 is applied after each MVViT layer. The model is trained using Stochastic Gradient Descent for YOLOX-S and AdamW optimization [227] for Deformable DETR and Swin Transformer. A batch size of 6 images per view is used to train YOLOX-S for both X-ray datasets and 2 images per view for the Wildtrack dataset. On the other hand, a batch size of 2 images per view is used to train Deformable DETR for the X-ray-Dual dataset and 1 image per view for both X-ray-Quad and Wildtrack datasets. Finally, a batch size of 4 is used for both X-ray datasets and 3 for the Wildtrack dataset. MMDetection [224] framework is used with the original training and optimisation settings for the three detectors. Models were trained using an NVIDIA Tesla V100.



(a)

(b)

Figure 4.3: Multi-view Detection Datasets used in this chapter. (a) X-ray-Quad dataset (4 views, X-ray security imagery). (b) Wildtrack [106] dataset (7 views, outdoors).

Table 4.1: Single View (SV) *vs.* MVViT Detection - X-ray-Dual - Statistical Performance

Architecture	**Category**	**Method**	**AP**	**AP$_{0.5}$**	**AP$_{0.75}$**	**AP$_{S}$**	**AP$_{M}$**	**AP$_{L}$**	**AR**	**AR$_{S}$**	**AR$_{M}$**	**AR$_{L}$**
YOLOX-S	Firearm	SV	0.624	0.939	0.730	-	0.633	0.709	0.674	-	0.666	0.796
		MVViT	**0.695**	**0.972**	**0.830**	-	**0.704**	**0.747**	**0.735**	-	**0.729**	**0.832**
	Knife	SV	0.242	0.540	0.169	0.093	0.280	**0.048**	0.349	0.118	0.366	**0.425**
		MVViT	**0.285**	**0.619**	**0.229**	**0.098**	**0.325**	0.033	**0.383**	**0.147**	**0.402**	0.350
	Laptop	SV	0.710	0.981	**0.869**	-	-	0.710	0.762	-	-	0.762
		MVViT	**0.723**	**0.990**	0.868	-	-	**0.723**	**0.771**	-	-	**0.771**
	Camera	SV	0.566	0.867	0.672	-	**0.800**	0.562	0.624	-	**0.800**	0.622
		MVViT	**0.632**	**0.896**	**0.811**	-	**0.800**	**0.630**	**0.688**	-	**0.800**	**0.686**
	All	SV	0.535	0.832	0.610	0.093	0.571	0.507	0.602	0.118	0.611	0.651
		MVViT	**0.583**	**0.869**	**0.685**	**0.098**	**0.610**	**0.533**	**0.644**	**0.147**	**0.644**	**0.660**
Deformable DETR	Firearm	SV	0.674	**0.968**	0.816	-	0.685	0.667	0.741	-	0.735	**0.832**
		MVViT	**0.680**	0.960	**0.817**	-	**0.689**	**0.679**	**0.743**	-	**0.738**	0.818
	Knife	SV	**0.251**	**0.626**	**0.142**	**0.139**	**0.285**	0.033	**0.428**	**0.162**	**0.450**	0.325
		MVViT	0.237	0.598	0.116	0.112	0.269	**0.159**	0.423	0.135	0.444	**0.425**
	Laptop	SV	0.803	0.990	0.947	-	-	0.803	0.855	-	-	0.855
		MVViT	**0.839**	**0.995**	**0.953**	-	-	**0.839**	**0.879**	-	-	**0.879**
	Camera	SV	**0.646**	**0.918**	**0.837**	-	0.700	**0.647**	**0.738**	-	0.700	**0.738**
		MVViT	0.601	0.847	0.739	-	**0.800**	0.600	0.723	-	**0.800**	0.722
	All	SV	**0.593**	**0.876**	**0.686**	**0.139**	0.557	0.537	0.691	**0.162**	0.628	0.688
		MVViT	0.589	0.850	0.656	0.112	**0.586**	**0.569**	**0.692**	0.135	**0.661**	**0.711**
Swin Transformer	Firearm	SV	0.698	**0.989**	0.873	-	0.705	**0.747**	0.741	-	0.735	**0.821**
		MVViT	**0.702**	**0.989**	**0.898**	-	**0.711**	0.718	**0.746**	-	**0.741**	0.818
	Knife	SV	0.419	0.821	0.370	0.189	0.449	0.311	0.493	0.279	0.507	0.675
		MVViT	**0.428**	**0.847**	**0.381**	**0.219**	**0.458**	**0.317**	**0.499**	**0.315**	**0.512**	**0.700**
	Laptop	SV	**0.833**	**0.991**	**0.976**	-	-	**0.833**	**0.876**	-	-	**0.876**
		MVViT	0.820	0.987	**0.976**	-	-	0.820	0.864	-	-	0.864
	Camera	SV	**0.680**	0.967	**0.836**	-	**0.700**	**0.681**	0.721	-	**0.700**	0.722
		MVViT	0.668	**0.976**	0.806	-	**0.700**	0.669	**0.723**	-	**0.700**	**0.723**
	All	SV	**0.657**	0.942	0.764	0.189	0.618	**0.643**	**0.708**	0.279	0.648	0.773
		MVViT	0.655	**0.950**	**0.765**	**0.219**	**0.623**	0.631	**0.708**	**0.315**	**0.651**	**0.776**

4.4 Results

The statistical performance of MVViT compared with single view detection is presented in Tables 4.1 to 4.3. For the X-ray-Dual (Table 4.1) and X-ray-Quad (Table 4.2) datasets, results for each class, as well as the overall performance are presented. The results for the X-ray-Dual dataset show an improvement when training with YOLOX-S, with an increase of 4.8% on the AP metric and 6.7% on the AP$_{0.5}$ metric. The performance slightly worsens when training Deformable DETR and Swin Transformer with MVViT on the X-ray-Dual dataset. This effect may be caused by the fact that these architectures obtain high precision for almost all

Table 4.2: Single View (SV) *vs.* MVViT Detection - X-ray-Quad - Statistical Performance

Architecture	**Category**	**Method**	**AP**	**AP$_{0.5}$**	**AP$_{0.75}$**	**AP$_S$**	**AP$_M$**	**AP$_L$**	**AR**	**AR$_S$**	**AR$_M$**	**AR$_L$**
YOLOX-S	Firearm	SV	0.734	0.973	0.884	-	0.742	0.787	0.767	-	0.759	**0.845**
		MVViT	**0.748**	**0.979**	**0.907**	-	**0.760**	**0.790**	**0.779**	-	**0.774**	0.838
	Knife	SV	0.353	0.693	0.331	0.150	0.392	0.022	0.447	**0.188**	0.459	**0.325**
		MVViT	**0.379**	**0.732**	**0.346**	**0.152**	**0.414**	**0.051**	**0.461**	0.178	**0.475**	**0.325**
	Laptop	SV	0.765	0.987	0.909	-	-	0.765	0.812	-	-	0.812
		MVViT	**0.806**	**0.992**	**0.946**	-	-	**0.806**	**0.844**	-	-	**0.844**
	Camera	SV	0.639	0.899	0.778	-	**0.800**	0.639	0.688	-	**0.800**	0.687
		MVViT	**0.678**	**0.926**	**0.840**	-	0.700	**0.678**	**0.726**	-	0.700	**0.726**
	All	SV	0.623	0.888	0.726	0.150	**0.644**	0.553	0.678	**0.188**	**0.673**	0.667
		MVViT	**0.653**	**0.907**	**0.760**	**0.152**	0.625	**0.581**	**0.703**	0.178	0.650	**0.683**
Deformable DETR	Firearm	SV	**0.726**	0.978	**0.885**	-	**0.740**	0.711	0.784	-	0.779	0.832
		MVViT	0.724	**0.997**	0.884	-	0.735	**0.720**	**0.788**	-	**0.782**	**0.854**
	Knife	SV	**0.352**	0.751	**0.286**	0.123	**0.390**	**0.143**	0.501	**0.163**	0.517	0.375
		MVViT	0.347	**0.760**	0.261	**0.140**	0.386	0.085	**0.506**	0.161	**0.521**	**0.438**
	Laptop	SV	0.847	0.984	0.970	-	-	0.847	0.896	-	-	0.896
		MVViT	**0.859**	**0.993**	**0.977**	-	-	**0.859**	**0.912**	-	-	**0.912**
	Camera	SV	0.646	0.896	0.772	-	**0.800**	0.647	**0.773**	-	**0.800**	**0.773**
		MVViT	**0.674**	**0.909**	**0.836**	-	0.700	**0.674**	0.772	-	0.700	**0.773**
	All	SV	0.643	0.902	0.728	0.123	**0.643**	**0.587**	0.738	**0.163**	**0.699**	0.719
		MVViT	**0.651**	**0.910**	**0.739**	**0.140**	0.607	0.585	**0.745**	0.161	0.668	**0.744**
Swin Transformer	Firearm	SV	**0.742**	**0.990**	0.932	-	**0.755**	**0.770**	**0.780**	-	**0.774**	**0.846**
		MVViT	0.738	**0.990**	**0.934**	-	0.751	0.758	0.779	-	**0.774**	0.836
	Knife	SV	0.503	**0.904**	0.515	**0.288**	0.537	0.290	0.566	**0.359**	0.574	**0.738**
		MVViT	**0.508**	0.901	**0.531**	0.254	**0.539**	**0.305**	**0.569**	0.302	**0.580**	0.713
	Laptop	SV	0.863	0.990	0.977	-	-	0.863	0.903	-	-	0.903
		MVViT	**0.873**	**0.992**	**0.982**	-	-	**0.873**	**0.908**	-	-	**0.908**
	Camera	SV	0.669	0.918	**0.854**	-	**0.800**	0.669	**0.720**	-	**0.800**	**0.720**
		MVViT	**0.671**	**0.927**	0.814	-	**0.800**	**0.670**	0.709	-	**0.800**	0.708
	All	SV	0.694	0.950	**0.819**	**0.288**	**0.697**	0.648	**0.742**	**0.359**	0.716	**0.802**
		MVViT	**0.698**	**0.952**	0.815	0.254	**0.697**	**0.651**	0.741	0.302	**0.718**	0.791

Table 4.3: Single View (SV) *vs.* MVViT Detection - Wildtrack - Statistical Performance

Architecture	**Method**	**AP**	**AP$_{0.5}$**	**AP$_{0.75}$**	**AP$_S$**	**AP$_M$**	**AP$_L$**	**AR**	**AR$_S$**	**AR$_M$**	**AR$_L$**
YOLOX-S	SV	**0.383**	**0.773**	**0.334**	-	**0.299**	**0.412**	**0.492**	-	**0.453**	**0.514**
	MVViT	0.370	0.764	0.301	-	0.274	0.409	0.471	-	0.368	0.511
Deformable DETR	SV	**0.417**	**0.772**	**0.388**	-	**0.335**	**0.450**	**0.587**	-	**0.515**	**0.613**
	MVViT	0.401	0.761	0.368	-	0.318	0.432	0.577	-	0.513	0.601
Swin Transformer	SV	0.367	0.780	0.267	-	0.266	0.408	0.489	-	0.449	0.508
	MVViT	**0.374**	**0.784**	**0.300**	-	**0.274**	**0.419**	**0.503**	-	**0.463**	**0.522**

classes (except for knives). The remaining not-detected objects present a significant detection challenge against which further advancement may adversely impact overall network performance across other classes. For the X-ray-Quad dataset, the precision of the three detectors improves when using MVViT, with an increase of 3% AP, 1.9% AP$_{0.5}$ with the YOLOX-S architecture, and small increments of 0.8% and

0.4% on the AP when using the Deformable DETR and Swin Transformer architectures. These results indicate that the performance can be increased if the model integrates features from different views, having a better performance when more views are used. However, as seen in the performance on the Wildtrack dataset, it is sensitive to highly occluded data. On the other hand, MVViT outperforms single view detection in the Wildtrack dataset (Table 4.3) only for the Swin Transformer architecture, with a slight increment of 0.7% in the COCO AP, while small decrements are seen in the YOLOX-S and Deformable DETR architectures. As seen in Figure Fig. 4.3b, this dataset has many occlusions across the different views, which imposes an additional challenge for MVViT.

Figs. 4.4 to 4.6 show detection examples for the X-ray-Quad dataset across the different detectors. For instance, Fig. 4.4 shows an example of a missed highly occluded knife in views 1 and 3 of the single view detection with a Swin Transformer backbone that is correctly detected when adding an MVViT layer. It can be seen that in the third view, the knife is pointing towards the camera centre and in a cluttered area, making it a challenging single-view detection instance. However, aggregating the features from other views allows for a 3D awareness that is translated to the knife being detected. It is also observed that a false positive detection for a knife in view 2 could not be removed, which is also attributed to being too close to the ground truth knife and integrating a similar set of features from the other views, indicating that further work to avoid this behaviour is needed. Fig. 4.5 shows another example of multi-view detection using a YOLOX detector that is improved by the use of an MVViT layer. In this example, a camera that cannot be detected in any of the views independently is correctly located when aggregating the features from all the views. Additionally, a firearm in view 3 is also detected, which is missed in its single-view counterpart. Finally, Fig. 4.6 shows a firearm overlapped with a laptop in view 2 that is detected with MVViT in a Deformable DETR detector, which is missed in single-view detection, with an additional false-positive knife being removed from the same view. It is hypothesised that the overlapping nature of transmission images, such as X-ray, puts an additional difficulty since feature vectors may contain information from more than one class, which can be alleviated with an MVViT layer.

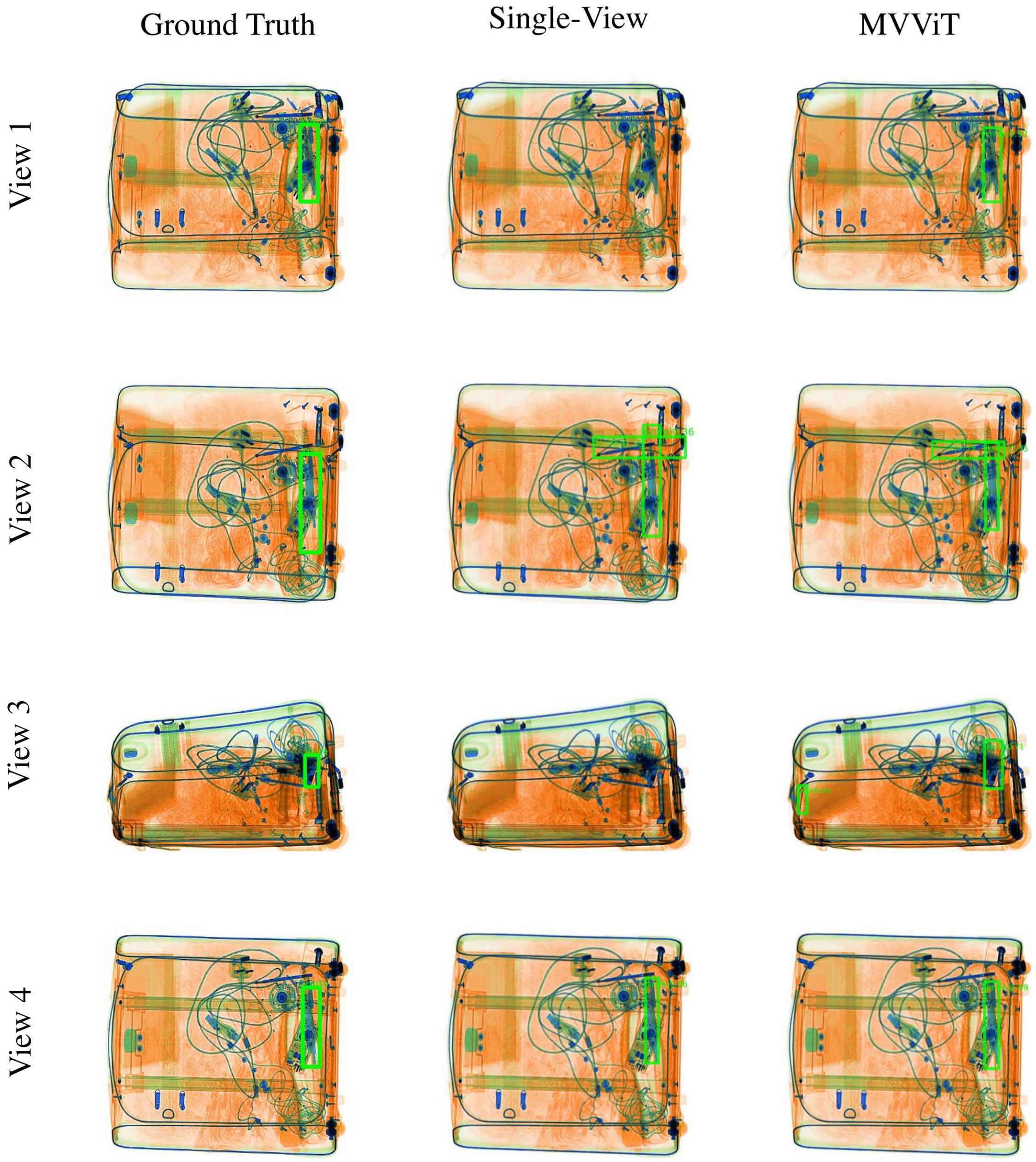


Figure 4.4: Single-view *vs.* Multi-view detection with MVViT on the X-ray-Quad dataset. Detector: Faster RCNN with a Swin Transformer backbone.

Figs. 4.7 to 4.9 show a comparison between single-view detection and multi-view detection for the Wildtrack dataset across the different detectors. This dataset presents several difficulties for multi-view object detection. For instance, only people standing on the same ground plane are annotated. Fig. 4.7 shows the results when using a Faster-RCNN with a Swin Transformer backbone. Some redundant detections are removed when using an MVViT layer, such as the person in the left of view

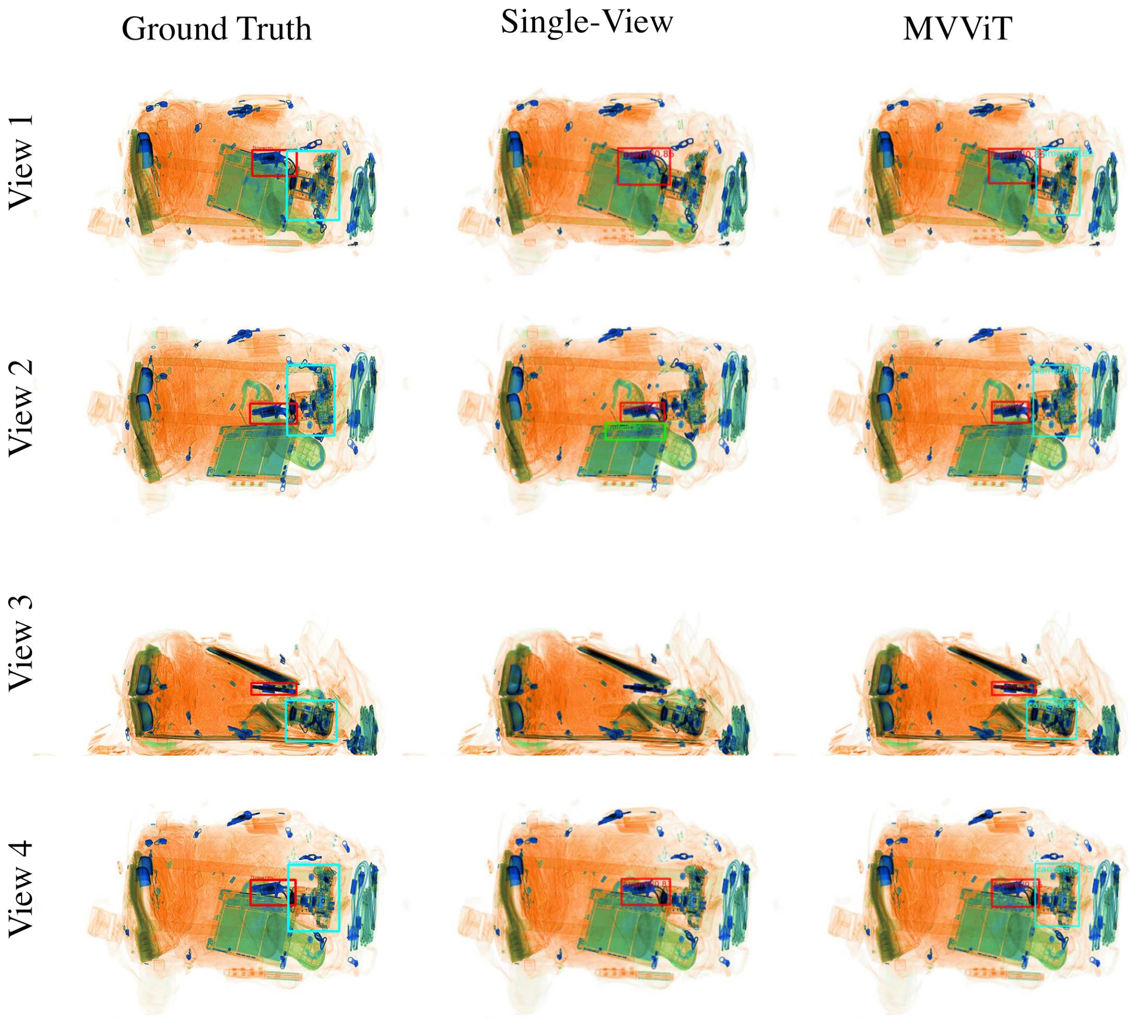


Figure 4.5: Single-view *vs.* Multi-view detection with MVViT on the X-ray-Quad dataset. Detector: YOLOX-S.

1 of Fig. 4.7. Also, people instances detected in single-view detection that are not part of the ground truth objects are removed from the multi-view detected objects, such as the people in the back (top-centre) of view 3, the person walking in the background in view 6 and the person with a red jacket in the back of view 1. However, some duplicates are also introduced, such as the right-centre people instances in view 7. Additionally, MVViT within the Swin Transformer backbone cannot improve on the crowd instance detections that are already incorrectly detected in single-view detection (*e.g.*, view 2 and view 7). Further examples of MVViT in YOLOX-S and Deformable DETR are seen in Figs. 4.8 and 4.9. Some redundant detections are again removed when using MVViT, such as view 7 in Fig. 4.8. Furthermore, in some instances where there is a small group of people, MVViT can detect the

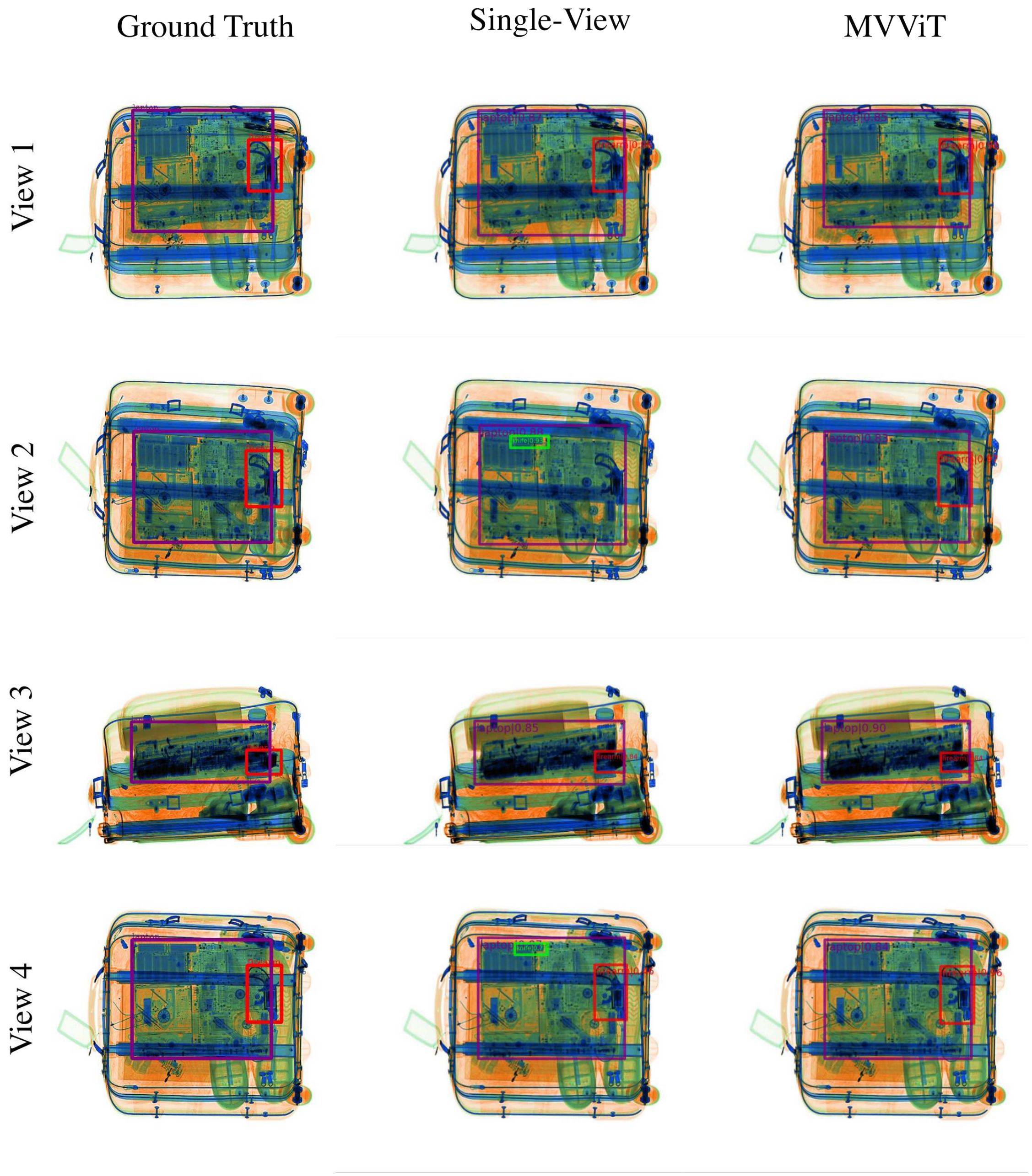


Figure 4.6: Single-view *vs.* Multi-view detection with MVViT on the X-ray-Quad dataset. Detector: Deformable DETR.

correct number of people with the aggregation of multi-view information (*e.g.*, the left group of people in the left part of view 2 of Fig. 4.9). These results indicate that although our network is 3D-aware and integrates the corresponding features correctly, it is difficult to cope with highly occluded and crowded scenarios.

Finally, a qualitative analysis of the attention map in the source view given a feature vector in the target view is performed on the X-ray-Quad dataset. Fig. 4.10

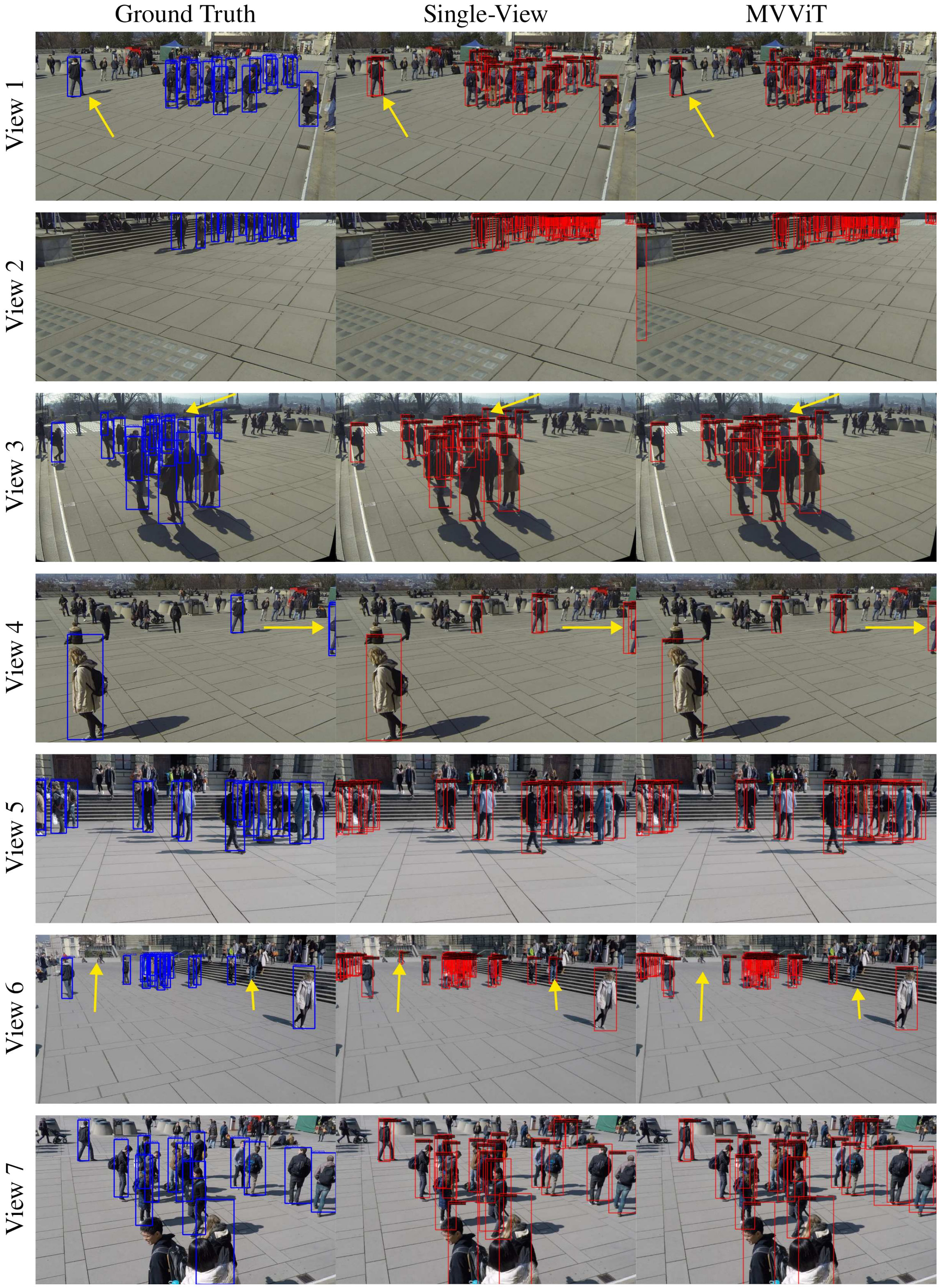


Figure 4.7: Single-view *vs.* Multi-view detection with MVViT on the Wildtrack dataset. Yellow arrows indicate notable differences. Detector: Faster RCNN with a Swin Transformer backbone.

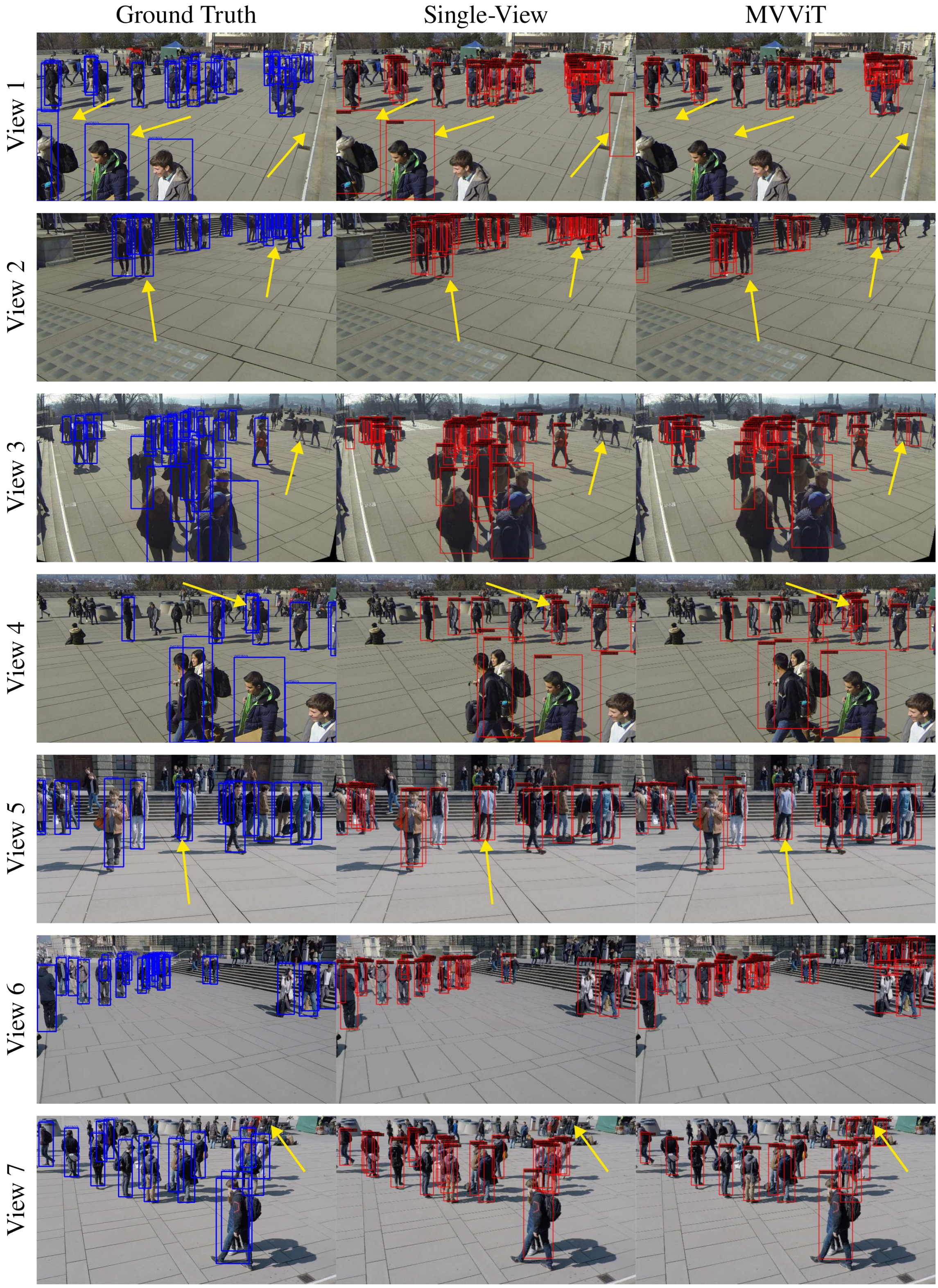


Figure 4.8: Single-view *vs.* Multi-view detection with MVViT on the Wildtrack dataset. Yellow arrows indicate notable differences. Detector: YOLOX-S.

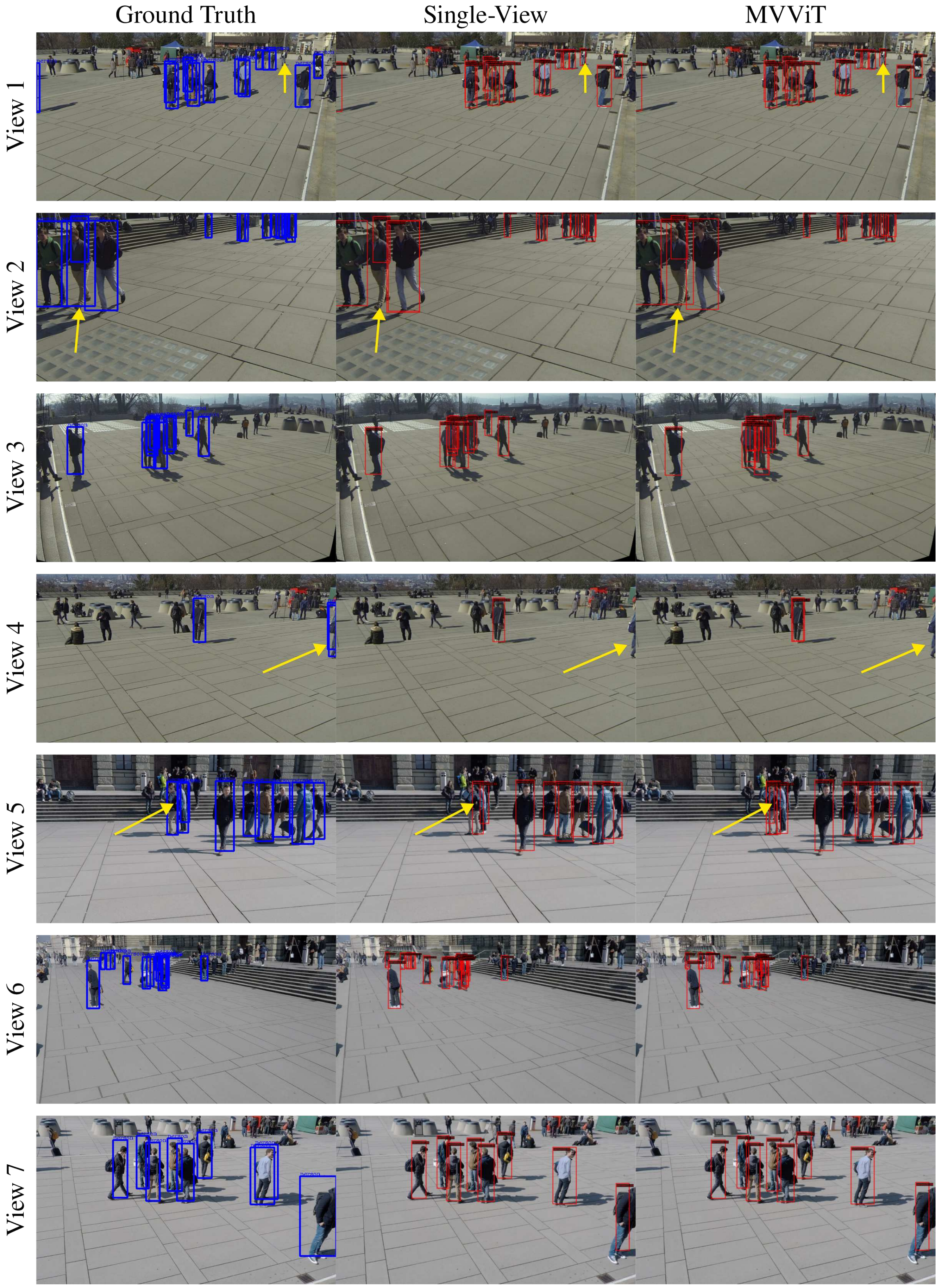
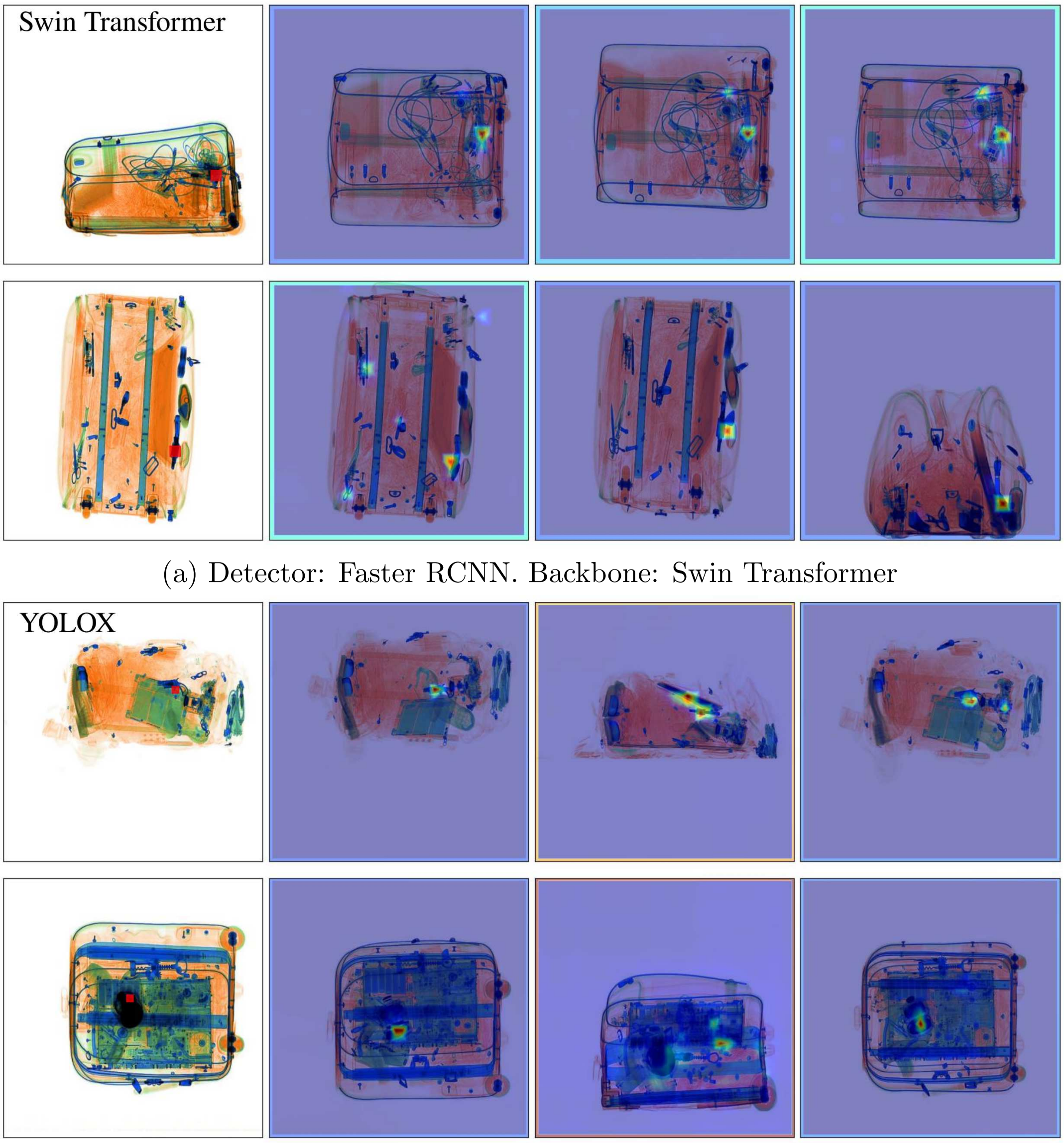


Figure 4.9: Single-view *vs.* Multi-view detection with MVViT on the Wildtrack dataset. Yellow arrows indicate notable differences. Detector: Deformable DETR.



(a) Detector: Faster RCNN. Backbone: Swin Transformer

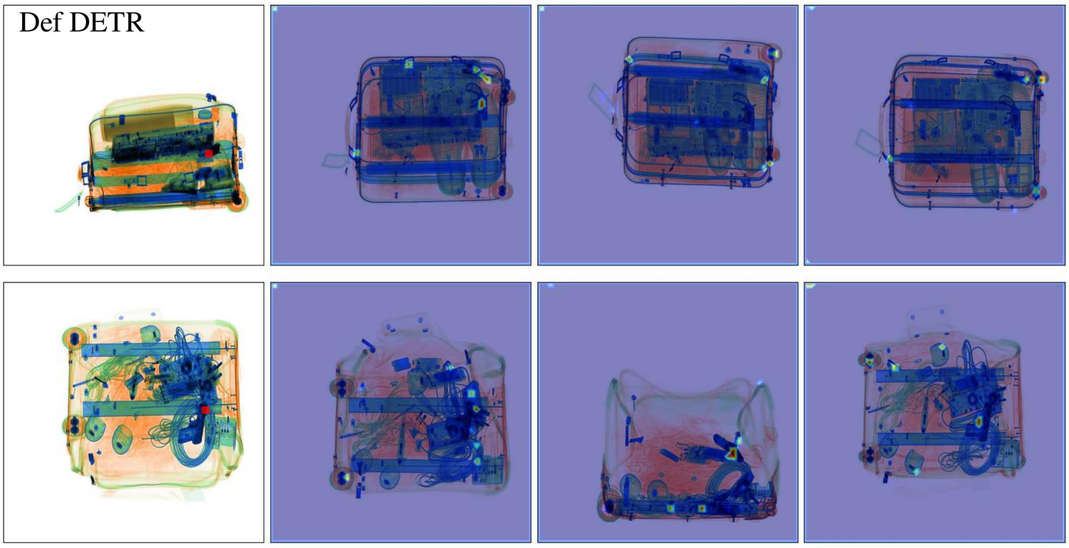
(b) Detector: YOLOX-S. Backbone: Darknet.

Figure 4.10: Attention mechanism in MVViT: the left image is the target view where the red square represents the location of the reference feature vector whilst the right three images are the source views with a colour map representing the attention weights, with red being the maximum weight and blue the minimum.

shows the attention mechanism in the source view (right three images) given a feature vector from the target view (left image) whose spatial location is represented by a red square. A colour map of the attention weights is drawn over the source views, normalized such that the minimum and maximum values of the colour map correspond to the minimum and maximum weights across the three target views. In Fig. 4.10a, the target feature vector of the top example is located at a knife, corresponding to the same example of Fig. 4.4. It is seen that the attention mechanism gives a larger weight to the feature vectors of the same instance in other views. However, it is also observed that the false-positive knife detected in Fig. 4.4 is also highlighted, indicating a correlation in the feature space. The bottom example of Fig. 4.10a shows the attention mechanism when the target feature vector belongs to a firearm. In this case, the attention mechanism has significant weights in the firearms detected in the source views. Fig. 4.10b top example also shows the attention over a firearm that is partially overlapped with a tablet. In this case, all source views are attending to the firearm; nevertheless, the middle source view also has a high attention weight on the tablet, which appears as a high-density object given it is transversal with respect to the camera centre. This indicates that although MVViT is able to get a correlation between corresponding feature vectors, it might be lacking a broader consideration of the object instance instead of just a particular feature vector. This is further observed in the bottom example of Fig. 4.10b, where the target feature vector lies on a camera (more specifically, on a camera part) and the corresponding feature vector with the highest attention in the (transversal) middle source viewpoints to another high-density area. Similarly, Fig. 4.10c shows the same behaviour with the Deformable DETR detector. It is worth noting that in this detector, MVViT is added into a shallower layer of the backbone (before the *conv4* block of ResNet). In this case, the attention is almost always being concentrated in small regions around the corresponding source features, suggesting that MVViT is not attending to the entire object instance. A model that captures the shape of the object instances in the source view through attention could improve object detection performance. This has been recently explored by Liao *et al.* [228] with a shape-guided feature enhancement module. The implementation of such models for multi-view object detection remains an area for future work.

4.5 Conclusions

In this chapter, the Multi-View Vision Transformer (MVViT) is presented, a novel architecture that uses attention to aggregate the feature maps across multiple con-



(c) Detector: Deformable DETR. Backbone: ResNet-50.

Figure 4.10: Attention mechanism in MVViT: the left image is the target view where the red square represents the location of the reference feature vector whilst the right three images are the source views with a colour map representing the attention weights, with red being the maximum weight and blue the minimum. (Cont.)

current views within a standard detection architecture. MVViT takes as input the feature maps of a target view and applies attention to the feature maps of the other concurrent source views to create 3D scene geometry-aware feature representations.

An investigation of the performance of MVViT for a quad-view X-ray security scanner imagery dataset is carried out, obtaining an overall COCO AP increase of 4.8% for two views and 3% for four views using the YOLOX-S detector. Additionally, a slight increase in the performance is also observed with four views and using the Deformable DETR and Swin Transformer architectures. A decrease in the performance is observed when using a Deformable DETR and Swin Transformer detectors for the two views X-ray dataset, apparently caused by the detectors already reaching the best performance. It is also observed that MVViT increases the AP of a seven-view pedestrian dataset by 0.7% with the Swin Transformer architecture, but it fails with YOLOX-S and Deformable DETR. This indicates that the highly occluded nature of the Wildtrack dataset imposes a greater challenge for MVViT. Additionally, an analysis of the attention maps in the source views with respect to a feature vector in the target view is conducted. It is further observed that the attention in the source view matches the corresponding feature vector from the target view, although it does not capture the whole region of interest. Our MVViT architecture is able to aggregate feature vectors through attention without any explicit knowledge about the relative position of the cameras, as in previous works. This enables its use in other applications where the relative position of the cameras changes dynamically, as in medical imaging.

CHAPTER 5

Exact-NeRF: An Exploration of a Precise Volumetric Parameterisation for Neural Radiance Fields

Further to the consideration of multi-view imagery for the purposes of detection in Chapters 3 and 4, this chapter alternatively considers the problem of multiple view-based scene rendering, *i.e.*, the generation of novel views by using only a sparse set of 2D images of the scene. Neural Radiance Fields (NeRF) [12] have attracted significant attention due to their ability to synthesize novel scene views with great accuracy. However, inherent to their underlying formulation, the sampling of points along a ray with zero width may result in ambiguous representations that lead to further rendering artifacts, such as blurring and aliasing. To address this issue, the recent seminal variant mip-NeRF [42] proposes an Integrated Positional Encoding (IPE) based on casting conical frustums instead of rays. Although this is expressed with an integral formulation, mip-NeRF approximates this integral as the expected value of a multivariate Gaussian distribution. This approximation is reliable for short frustums but degrades with highly elongated regions, which arise when dealing with distant scene objects under a larger depth range.

In this chapter, the use of an exact approach for calculating the IPE by using a pyramid-based integral formulation instead of an approximated conical-based one is explored. This formulation is denoted as *Exact-NeRF* and contributes the first approach to offer a precise analytical solution to the IPE within the NeRF domain. This exploratory work illustrates that such an exact formulation (Exact-NeRF) matches the accuracy of mip-NeRF and furthermore provides a natural extension to more challenging scenarios without further modification, such as in the case of unbounded scenes of mip-NeRF 360 [43]. The contribution within this chapter aims to both address the hitherto unexplored issues of frustum approximation in earlier NeRF work and additionally provide insight into the potential future consideration of analytical solutions in future NeRF extensions.

The contributions of this chapter appear in the following peer-reviewed publication:

> Brian K. S. Isaac-Medina, Chris G. Willcocks and Toby P. Breckon. "Exact-NeRF: An Exploration of a Precise Volumetric Parameterisation for Neural Radiance Fields", in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 66-75, 2023.

5.1 Introduction

An introduction to the NeRF architecture is already presented in Section 2.3. Recalling this, NeRF uses an MLP to predict the colour and density of a scene given the location of a 3D point in space and a viewing direction. With this approach, NeRF learns the implicit geometry of a scene, having the inherent ability to synthesize novel views. In its original formulation, NeRF illustrates strong reconstruction performance for synthetic datasets comprising object-centric scenes and no background (bounded) and forward-facing real-world scenes. A review of the applications of NeRF is presented in Section 2.3.4.

Barron *et al.* propose mip-NeRF [42], an architecture similar to NeRF but that casts cones instead of rays to prevent aliasing and blurring. Mip-NeRF encodes cone frustums representing different regions of the field to predict their colour and density in a similar manner as NeRF. The cone frustums are approximated using 3D Gaussians with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. The actual formulation of mip-NeRF is presented in Section 2.3.2. This approximation, however, is only really valid for bounded scenes, where the conic frustums do not suffer from large elongations attributable to a large depth of field within the scene.

The NeRF concept has been extended to represent increasingly difficult scenes. For instance, mip-NeRF 360 [43] learns a representation of unbounded scenes with a central object by giving more capacity to points that are near the camera, modifying the network architecture and introducing a regularizer that penalizes ‘floaters’ (unconnected depth regions in free space) and other small unconnected regions. In order to model distant regions, mip-NeRF 360 transforms the multivariate Gaussians with a function that contracts the space beyond the unit sphere, namely:

$$f(\mathbf{x}) = \begin{cases} \mathbf{x} & \|\mathbf{x}\| \leq 1 \\ \left(2 - \frac{1}{\|\mathbf{x}\|}\right) \frac{\mathbf{x}}{\|\mathbf{x}\|} & \|\mathbf{x}\| > 1 \end{cases} . \tag{5.1}$$

Subsequently, the new $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ values are given by $f(\boldsymbol{\mu})$ and $\mathbf{J}_f(\boldsymbol{\mu})\boldsymbol{\Sigma}\mathbf{J}_f(\boldsymbol{\mu})^\top$, where $\mathbf{J}_f$ is the Jacobian matrix of f. Empirically, this re-parameterisation now allows learning the representation of scenes with distant backgrounds (*i.e.*, over a longer depth of field). This modification allows a better representation and outperforms standard mip-NeRF for an unbounded scenes dataset. However, the modification of the Gaussians requires attentive analysis to encode the correct information in the contracted space, which includes the linearization of the contraction function to accommodate the Gaussian approximations. This leads to a degraded performance of mip-NeRF 360 when the camera is far from the object. Additionally, mip-NeRF 360 struggles to render thin structures such as tree branches or bicycle rays.

NeRF uses a positional encoding (PE) on the raw coordinates of the input points in order to induce the network to learn higher-frequency features [197]. In the mip-NeRF context, the conical frustums are similarly encoded using an integrated positional encoding (IPE), which aims to integrate the PE over the cone frustums. Given that the associated integral has no closed-form solution, they formulate the IPE as the expected value of the positional encoding in a 3D Gaussian distribution centred in the frustum (Section 2.3.2). The IPE reduces aliasing by reducing the ambiguity of single-point encoding. For the contracted space of mip-NeRF 360 [43],

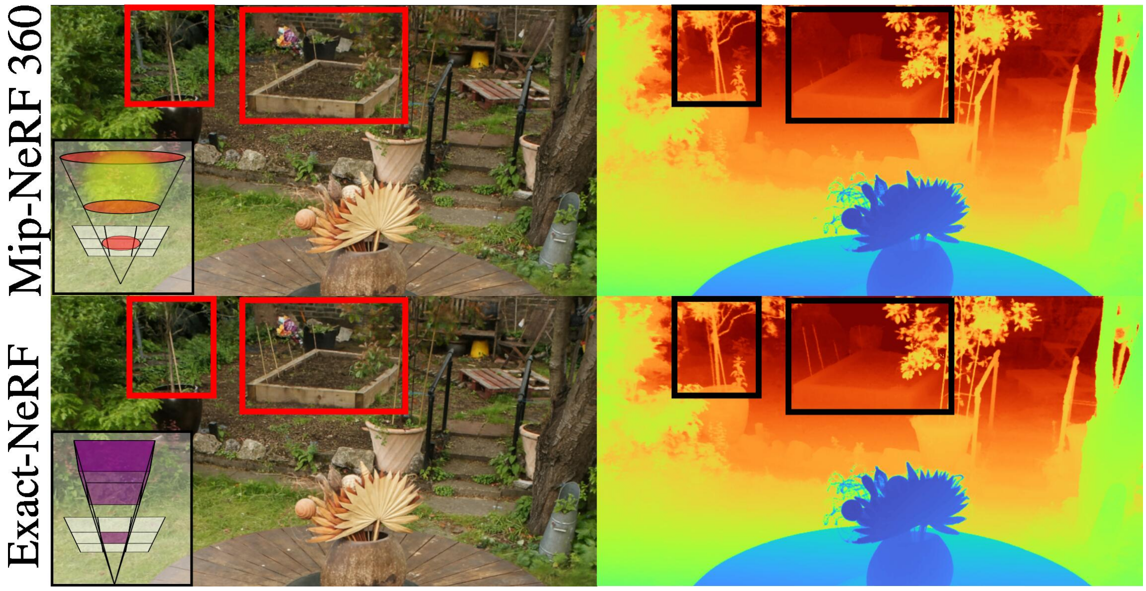


Figure 5.1: Comparison of Exact-NeRF (the proposed architecture of this chapter) with mip-NeRF 360 [43]. Exact-NeRF is able to both match the performance and obtain superior depth estimation over a larger depth of field.

mip-NeRF 360 applies the same IPE strategy considering the contraction function in Eq. (5.1). Mip-NeRF 360 samples the intervals of the volumetric regions using the inverse of the distance in order to assign a bigger capacity to nearer objects, similar to DONeRF [49]. In addition, other works consider the idea of volumetric sampling without using IPE. For instance, ZipNeRF [229] allows for volumetric sampling in the Instant NGP [206] framework by considering the weighted sum of points in an ordered structure resembling the volumetric region.

Motivated by this, this chapter introduces *Exact-NeRF* as an exploration of an alternative exact parameterisation of underlying volumetric regions that are used in the context of mip-NeRF and mip-NeRF 360 (Fig. 5.1). Exact-NeRF uses pyramid-based frustums in order to enable an exact integration of the IPE formula in Eq. (2.46), finding a closed-form volumetric positional encoding formulation (Section 5.2) instead of the multivariate Gaussian approximation used by mip-NeRF. Exact-NeRF matches the performance of mip-NeRF on a synthetic dataset but gets a sharper reconstruction around edges. This approach can be applied without further modification to the contracted space of mip-NeRF 360. The naive implementation of Exact-NeRF for the unbounded scenes of mip-NeRF 360 has a small decrease in performance, but it gets cleaner reconstructions of the background given its more accurate representation of volumetric regions. Additionally, the depth map estimations obtained by Exact-NeRF are less noisy than mip-NeRF 360. The key contribution within this chapter is the formulation of a general and exact IPE that can be applied to any shape that can be broken into triangles (*i.e.*, a polyhedron). This chapter is intended to serve as a motivation to investigate different shapes and analytical solutions of volumetric positional encoding.

5.2 Methodology: Exact-NeRF

In this chapter, Exact-NeRF is presented as an exploration of how the IPE approximations of earlier work [42, 43] based on a conic parameterisation can be replaced with a square pyramid-based formulation in order to obtain an exact IPE γ_E, as shown in Fig. 5.2. The motivation behind this formulation is to match the volumetric rendering with the pixel footprint, which in turn is a rectangle. Recalling Eq. (2.46), the definition of IPE is:

$$\gamma_I(\mathbf{d}, \mathbf{o}, \dot{\rho}, t_i, t_{i+1}) = \frac{\iiint_F \gamma(\mathbf{x}) dV}{\iiint_F dV} \,.$$

Section 5.2.1 deals with the integration of the denominator of Eq. (2.46) (the volume of the frustum) while Section 5.2.2 obtains the solution of the integration of the numerator. The handling of some indeterminate cases is discussed in Section 5.2.3.

5.2.1 Volume of Pyramidal Frustums

A pyramidal frustum can be defined by a set of 8 vertices $\mathcal{V} = \{\mathbf{v}_i\}_{i=1}^{8}$ and 6 quadrilateral faces $\mathcal{F} = \{f_j\}_{j=1}^{6}$. In order to get the volume in the denominator of Eq. (2.46), we use the divergence theorem:

$$\iiint \nabla \cdot F dV = \oiint_{\partial S} F \cdot d\mathbf{S} \,, \tag{5.2}$$

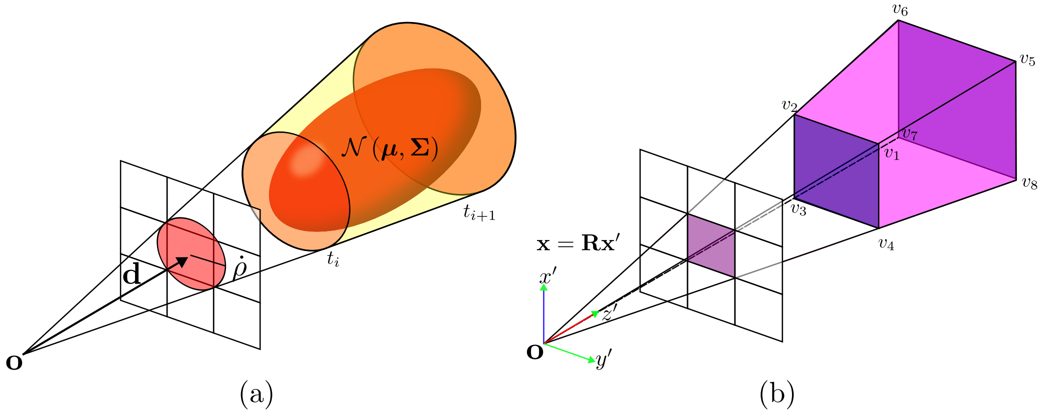


Figure 5.2: Cone and pyramid tracing for volumetric NeRF parameterisations. (a) Mip-NeRF [42] uses cone frustums to parameterize a 3D region. Since the IPE of these frustums does not have a closed-form solution, it is approximated by modelling the frustum as a multivariate Gaussian. (b) Exact-NeRF casts a square pyramid instead of a cone, allowing for an exact parameterisation of the IPE by using the vertices v_i of the frustum and the pose parameters $\mathbf{o}$ and $\mathbf{R}$.

with $F = \frac{1}{3}[x, y, z]^\top$, yielding to the solution for the volume as:

$$V = \iiint \nabla \cdot F dV = \iiint dV = \frac{1}{3} \oiint_{\partial S} [x, y, z] \, d\mathbf{S} \,. \tag{5.3}$$

Without losing generality, we divide each face into triangles, giving a set of triangular faces $\mathcal{T}$ such that the polyhedra formed by faces $\mathcal{F}$ and $\mathcal{T}$ are the same. Each triangle τ is defined by three points $\mathbf{P}_{\tau,0}, \mathbf{P}_{\tau,1}$ and $\mathbf{P}_{\tau,2}$, with $\mathbf{P}_{\tau,i} \in \mathcal{V}$, such that the cross product of the edges $\mathbf{E}_{\tau,1} = \mathbf{P}_{\tau,1} - \mathbf{P}_{\tau,0}$ and $\mathbf{E}_{\tau,2} = \mathbf{P}_{\tau,2} - \mathbf{P}_{\tau,0}$ points outside the frustum (Fig. 5.3). As a result, Eq. (5.3) equates to the sum of the surface integral for each triangle $\tau \in \mathcal{T}$,

$$V = \frac{1}{3} \sum_{\tau \in \mathcal{T}} \iint_{\tau} [x, y, z] \, d\mathbf{S} \,. \tag{5.4}$$

The points lying in the triangle $\triangle \mathbf{P}_{\tau,0}\mathbf{P}_{\tau,1}\mathbf{P}_{\tau,2}$ can hence be parameterized as:

$$\mathbf{P}_{\tau}(u, v) = \mathbf{P}_{\tau,0} + u\mathbf{E}_{\tau,1} + v\mathbf{E}_{\tau,2} \,, \tag{5.5}$$

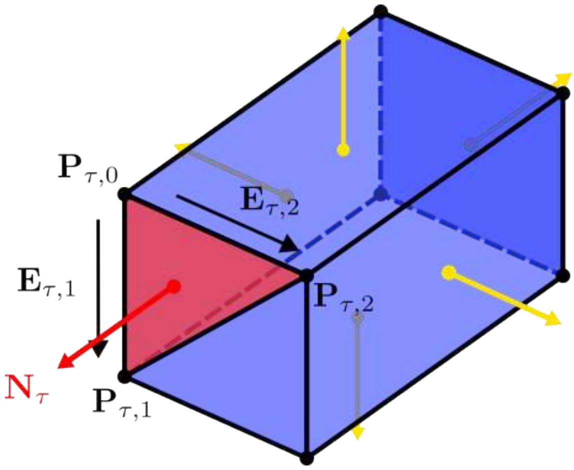


Figure 5.3: Parameterisation of triangular faces. The vertices are sorted counterclockwise, so the normal vector to their plane points outside the frustum.

such that $0 \leq u \leq 1, 0 \leq v \leq 1$ and $u + v \leq 1$. The differential term of Eq. (5.4) is then:

$$d\mathbf{S} = \left(\frac{\partial \mathbf{P}_\tau}{\partial u} \times \frac{\partial \mathbf{P}_\tau}{\partial v} \right) dudv \tag{5.6}$$

$$d\mathbf{S} = (\mathbf{E}_{\tau,1} \times \mathbf{E}_{\tau,2})\, dudv \triangleq \mathbf{N}_\tau dudv\,. \tag{5.7}$$

By substituting Eq. (5.7) into Eq. (5.4), and noting that $[x, y, z] = \mathbf{P}_\tau(u, v)$, we obtain:

$$V = \frac{1}{3} \sum_{\tau \in \mathcal{T}} \int_0^1 \int_0^{1-v} \mathbf{P}_\tau(u, v)^\top \mathbf{N}_\tau dudv\,. \tag{5.8}$$

Since the dot product of any point $\mathbf{P}_\tau$ in a face τ with a vector $\mathbf{N}_\tau$ normal to τ is constant, the product inside the integral of Eq. (5.8) is constant. Subsequently, $\mathbf{P}_\tau(u, v)$ can be replaced with any point, such as $\mathbf{P}_{\tau,0}$. Finally, the required volume is obtained as:

$$V = \frac{1}{3} \sum_{\tau \in \mathcal{T}} \mathbf{P}_{\tau,0}^\top \mathbf{N}_\tau \int_0^1 \int_0^{1-v} dudv = \frac{1}{6} \sum_{\tau \in \mathcal{T}} \mathbf{P}_{\tau,0}^\top \mathbf{N}_\tau\,. \tag{5.9}$$

5.2.2 Integration Over the Positional Encoding Function

Following from earlier, we can obtain the numerator of the IPE in Eq. (2.46) using the divergence theorem. We will base our analysis on the sine function and the x

coordinate, *i.e.*, $\gamma(x) = \sin(2^l x)$. Substituting $F = \left[-\frac{1}{2^l}\cos(2^l x), 0, 0\right]^\top$ in Eq. (5.2) we obtain:

$$\iiint \sin(2^l x) dV = \oiint_{\partial S} \left[-\frac{1}{2^l}\cos(2^l x), 0, 0\right] d\mathbf{S}\,. \tag{5.10}$$

Following the same strategy of dividing the surface into triangular faces as in the earlier volume calculation, Eq. (5.10) can be written as:

$$\iiint \sin(2^l x) dV = \sum_{\tau \in \mathcal{T}} \frac{1}{2^l} \sigma_{x,\tau} \mathbf{N}_\tau \cdot \hat{\mathbf{i}}\,, \tag{5.11}$$

where $\hat{\mathbf{i}}$ is the unit vector in the x direction and:

$$\sigma_{x,\tau} = \int_0^1 \int_0^{1-v} -\cos(2^l x_\tau(u,v)) du dv\,. \tag{5.12}$$

From Eq. (5.5), the x coordinate can be parameterized as:

$$x_\tau(u,v) = x_{\tau,0} + u(x_{\tau,1} - x_{\tau,0}) + v(x_{\tau,2} - x_{\tau,0})\,. \tag{5.13}$$

Substituting Eq. (5.13) in Eq. (5.12) and solving the integral, we obtain:

$$\begin{split}\sigma_{x,\tau} = \frac{1}{2^{2l}} \Bigg(\frac{\cos(2^l x_{\tau,0})}{(x_{\tau,0} - x_{\tau,1})(x_{\tau,0} - x_{\tau,2})} + \frac{\cos(2^l x_{\tau,1})}{(x_{\tau,1} - x_{\tau,0})(x_{\tau,1} - x_{\tau,2})} \\ + \frac{\cos(2^l x_{\tau,2})}{(x_{\tau,2} - x_{\tau,0})(x_{\tau,2} - x_{\tau,1})} \Bigg)\,.\end{split} \tag{5.14}$$

Furthermore, Eq. (5.14) can be written as:

$$\sigma_{x,\tau} = \frac{1}{2^{2l}} \frac{\det\left(\begin{bmatrix} \mathbf{1} & \boldsymbol{x}_\tau & \cos(2^l \boldsymbol{x}_\tau) \end{bmatrix}\right)}{\det\left(\begin{bmatrix} \mathbf{1} & \boldsymbol{x}_\tau & \boldsymbol{x}_\tau^{\circ 2} \end{bmatrix}\right)}\,, \tag{5.15}$$

where $\mathbf{1} = [1,1,1]^\top$, $\boldsymbol{x}_\tau = [x_{\tau,0}, x_{\tau,1}, x_{\tau,2}]^\top$ and $(\cdot)^{\circ n}$ is the element-wise power.

In general, we can also obtain the expression in Eq. (5.11) for the k-th coordinate of $\mathbf{x}$ as:

$$\iiint \sin(2^l \mathbf{x}_k) dV = \frac{1}{2^{3l}} \sum_{\tau \in \mathcal{T}} \sigma_{k,\tau} \mathbf{N}_\tau \cdot \mathbf{e}_k\,, \tag{5.16}$$

$$\sigma_{k,\tau} = \frac{\det\left(\begin{bmatrix}\mathbf{1} & \mathbf{X}_\tau^\top \mathbf{e}_k & \cos(2^l \mathbf{X}_\tau^\top \mathbf{e}_k)\end{bmatrix}\right)}{\det\left(\begin{bmatrix}\mathbf{1} & \mathbf{X}_\tau^\top \mathbf{e}_k & (\mathbf{X}_\tau^\top \mathbf{e}_k)^{\circ 2}\end{bmatrix}\right)}\,, \tag{5.17}$$

where $\mathbf{X}_\tau = \begin{bmatrix}\mathbf{P}_{\tau,0} & \mathbf{P}_{\tau,1} & \mathbf{P}_{\tau,2}\end{bmatrix}$ and $\mathbf{e}_k$ are the vectors that form the canonical basis in $\mathbb{R}^3$. Similarly, the integral over the cosine function is defined as:

$$\iiint \cos(2^l \mathbf{x}_k) dV = \frac{1}{2^{3l}} \sum_{\tau \in \mathcal{T}} \xi_{k,\tau} \mathbf{N}_\tau \cdot \mathbf{e}_k\,, \tag{5.18}$$

where:

$$\xi_{k,\tau} = -\frac{\det\left(\begin{bmatrix}\mathbf{1} & \mathbf{X}_\tau^\top \mathbf{e}_k & \sin(2^l \mathbf{X}_\tau^\top \mathbf{e}_k)\end{bmatrix}\right)}{\det\left(\begin{bmatrix}\mathbf{1} & \mathbf{X}_\tau^\top \mathbf{e}_k & (\mathbf{X}_\tau^\top \mathbf{e}_k)^{\circ 2}\end{bmatrix}\right)}\,. \tag{5.19}$$

Finally, we get the exact IPE (EIPE) of the frustum used by Exact-NeRF approach by dividing Eqs. (5.16) and (5.18) by Eq. (5.9) as follows:

$$\gamma_E(\mathbf{x}, l; \mathcal{V}) = \frac{6}{2^{3l}} \begin{bmatrix} \frac{\sum_{\tau \in \mathcal{T}} \boldsymbol{\sigma}_\tau \circ \mathbf{N}_\tau}{\sum_{\tau \in \mathcal{T}} \mathbf{P}_{\tau,0}^\top \mathbf{N}_\tau} \\ \frac{\sum_{\tau \in \mathcal{T}} \boldsymbol{\xi}_\tau \circ \mathbf{N}_\tau}{\sum_{\tau \in \mathcal{T}} \mathbf{P}_{\tau,0}^\top \mathbf{N}_\tau} \end{bmatrix}\,, \tag{5.20}$$

where $\boldsymbol{\sigma}_\tau = \begin{bmatrix}\sigma_{1,\tau} & \sigma_{2,\tau} & \sigma_{3,\tau}\end{bmatrix}^\top$ and $\boldsymbol{\xi}_\tau = \begin{bmatrix}\xi_{1,\tau} & \xi_{2,\tau} & \xi_{3,\tau}\end{bmatrix}^\top$. It is worth mentioning that Eq. (5.20) fails when a coordinate value repeats in any of the points of a triangle (*i.e.*, there is a triangle τ such that $\mathbf{P}_{\tau,i} = \mathbf{P}_{\tau,j}$ for a $i \neq j$). For these cases, *l'Hopital's rule* can be used to evaluate this limit (see Section 5.2.3).

Despite starting our analysis with squared pyramids, it can be noted that Eq. (5.20) is true for any set of vertices $\mathcal{V}$, meaning that this parameterisation can be applied for any shape with known vertices. This is particularly useful for scenarios where the space may be deformed and frustums may not be perfect pyramids, such as in mip-Nerf 360 [43]. Additionally, it can be noted that the proposed EIPE is multiplied by a factor of 2^{-3l}, meaning that $\gamma_E \to 0$ when $L \to \infty$, which hence makes our implementation robust to large values of L. This property of the Exact-NeRF formulation is consistent with that of the original mip-NeRF [42]. While Exact-NeRF does not affect the computational complexity of the MLP in the NeRF framework, the EIPE comprises more operations than the IPE in mip-NeRF (and therefore, more than NeRF) since Eq. (5.20) indicates that four different computations have to be performed for each triangular face. However, given that the computational burden of NeRF is in the neural network, this added complexity is not significant when compared to mip-NeRF during training and inference time.

5.2.3 Indeterminate Cases of the EIPE

By simplifying Eq. (5.14) we obtain:

$$\sigma_{x,\tau} = \frac{(x_{\tau,2} - x_{\tau,1})\cos(2^l x_{\tau,0}) + (x_{\tau,0} - x_{\tau,2})\cos(2^l x_{\tau,1}) + (x_{\tau,1} - x_{\tau,0})\cos(2^l x_{\tau,2})}{2^{2l}(x_{\tau,1} - x_{\tau,0})(x_{\tau,2} - x_{\tau,0})(x_{\tau,2} - x_{\tau,1})}\,. \tag{5.21}$$

From Eq. (5.21) we observe that an indetermination occurs for the case of two points in the triangle τ sharing the same coordinate, such that $x_{\tau,i} = x_{\tau,j}, i \neq j$. In order to get a valid value for these cases, we get the limit when those two coordinates approach. We can write Eq. (5.21) as:

$$\sigma_{x,\tau} = \frac{f(x_{\tau,0}, x_{\tau,1}, x_{\tau,2})}{2^{2l} g(x_{\tau,0}, x_{\tau,1}, x_{\tau,2})}\,. \tag{5.22}$$

Subsequently, we obtain the value for the case of $x_{\tau,0} = x_{\tau,1}$ using *l'Hopital's rule*:

$$\lim_{x_{\tau,0} \to x_{\tau,1}} \frac{f(x_{\tau,0}, x_{\tau,1}, x_{\tau,2})}{2^{2l} g(x_{\tau,0}, x_{\tau,1}, x_{\tau,2})} = \tag{5.23}$$

$$\lim_{x_{\tau,0} \to x_{\tau,1}} \frac{\frac{\partial}{\partial x_{\tau,0}} f(x_{\tau,0}, x_{\tau,1}, x_{\tau,2})}{2^{2l} \frac{\partial}{\partial x_{\tau,0}} g(x_{\tau,0}, x_{\tau,1}, x_{\tau,2})} = \tag{5.24}$$

$$\lim_{x_{\tau,0} \to x_{\tau,1}} \frac{-2^l (x_{\tau,2} - x_{\tau,1})\sin(2^l x_{\tau,0}) + \cos(2^l x_{\tau,1}) - \cos(2^l x_{\tau,2})}{2^{2l}(x_{\tau,2} - x_{\tau,1})(2x_{\tau,0} - x_{\tau,1} - x_{\tau,2})} = \tag{5.25}$$

$$\frac{2^l (x_{\tau,2} - x_{\tau,1})\sin(2^l x_{\tau,1}) - \cos(2^l x_{\tau,1}) + \cos(2^l x_{\tau,2})}{2^{2l}(x_{\tau,2} - x_{\tau,1})^2}\,. \tag{5.26}$$

Similarly, from Eq. (5.25), we evaluate the case $x_{\tau,0} = x_{\tau,2}$:

$$\frac{-2^l (x_{\tau,2} - x_{\tau,1})\sin(2^l x_{\tau,2}) + \cos(2^l x_{\tau,1}) - \cos(2^l x_{\tau,2})}{2^{2l}(x_{\tau,2} - x_{\tau,1})^2}\,. \tag{5.27}$$

For the case when $x_{\tau,1} = x_{\tau,2}$, we differentiate with respect to $x_{\tau,1}$ to obtain the corresponding value:

$$\lim_{x_{\tau,1}\to x_{\tau,2}} \frac{f(x_{\tau,0}, x_{\tau,1}, x_{\tau,2})}{2^{2l} g(x_{\tau,0}, x_{\tau,1}, x_{\tau,2})} = \tag{5.28}$$

$$\lim_{x_{\tau,1}\to x_{\tau,2}} \frac{-\cos(2^l x_{\tau,0}) + \cos(2^l x_{\tau,2}) + 2^l (x_{\tau,2} - x_{\tau,0}) \sin(2^l x_{\tau,1})}{2^{2l}(x_{\tau,2} - x_{\tau,0})(x_{\tau,0} + x_{\tau,2} - 2x_{\tau,1})} = \tag{5.29}$$

$$\frac{-2^l (x_{\tau,2} - x_{\tau,0}) \sin(2^l x_{\tau,1}) + \cos(2^l x_{\tau,0}) - \cos(2^l x_{\tau,2})}{2^{2l}(x_{\tau,2} - x_{\tau,0})^2} . \tag{5.30}$$

Finally, when $x_{\tau,0} = x_{\tau,1} = x_{\tau,2}$, we use again the *l'Hopital's rule* on Eq. (5.25) and differentiate again with respect to $x_{\tau,0}$ to obtain:

$$\lim_{x_{\tau,0}\to x_{\tau,1}\to x_{\tau,2}} \sigma_{x,\tau} = -\frac{1}{2} \cos(2^l x_{\tau,0}) . \tag{5.31}$$

Using the same approach, we can find the following expressions for $\xi_{x,\tau}$ (Eq. (5.19)):

$$\lim_{x_{\tau,0}\to x_{\tau,1}} \xi_{x,\tau} = \frac{2^l (x_{\tau,2} - x_{\tau,1}) \cos(2^l x_{\tau,1}) + \sin(2^l x_{\tau,1}) - \sin(2^l x_{\tau,2})}{2^{2l}(x_{\tau,2} - x_{\tau,1})^2} \tag{5.32}$$

$$\lim_{x_{\tau,0}\to x_{\tau,2}} \xi_{x,\tau} = \frac{-2^l (x_{\tau,2} - x_{\tau,1}) \cos(2^l x_{\tau,2}) - \sin(2^l x_{\tau,1}) + \sin(2^l x_{\tau,2})}{2^{2l}(x_{\tau,2} - x_{\tau,1})^2} \tag{5.33}$$

$$\lim_{x_{\tau,1}\to x_{\tau,2}} \xi_{x,\tau} = \frac{-2^l (x_{\tau,2} - x_{\tau,0}) \cos(2^l x_{\tau,2}) - \sin(2^l x_{\tau,0}) + \sin(2^l x_{\tau,2})}{2^{2l}(x_{\tau,2} - x_{\tau,0})^2} \tag{5.34}$$

$$\lim_{x_{\tau,0}\to x_{\tau,1}\to x_{\tau,2}} \xi_{x,\tau} = \frac{1}{2} \sin(2^l x_{\tau,0}) . \tag{5.35}$$

Similar expressions can be obtained for the y and z coordinates.

5.3 Implementation Details

Exact-NeRF is implemented using the original code of mip-NeRF, which is based on JAXNeRF [230]. Apart from the change of the positional encoding, no further modification is made. The same sampling strategy of ray intervals defined in Eq. (2.38) is used, but sampling $N + 1$ points to define N intervals. In order to obtain the vertices of the pyramid frustums, the coordinates of the corners of each pixel are used, multiplying them by the t_i values to get the front and back faces of the frustums. Double precision (64-bit float) is used for calculating the EIPE itself, as it relies upon arithmetic over very low numerical decimals that are otherwise prone to numerical precision error (see Eq. (5.14)). After calculation, the EIPE result is transformed back to single precision (32-bit float).

A comparison is carried out between the implementation of Exact-NeRF against the original mip-NeRF baseline on the benchmark Blender dataset [12], downsampled by a factor of 2. A similar training strategy is followed as in mip-NeRF: training both models for 800k iterations (instead of 1 million, as convergence at this point was observed) with a batch size of 4096 using Adam optimization [65] with a logarithmically annealed learning rate, $5 \times 10^{-4} \rightarrow 5 \times 10^{-6}$. All training is carried out using 2 × NVIDIA Tesla V100 GPU per scene.

Additionally, the use of the EIPE against mip-NeRF 360 on the dataset of Barron *et al.* [43] is compared. Similarly, the reference code from MultiNeRF [231] is used, which contains an implementation of mip-NeRF 360 [43]. Pyramidal frustum vertices are contracted using Eq. (5.1) and the EIPE is obtained using the Eq. (5.20) with the mapped vertices. It is noted that the contracting space of mip-NeRF 360 also contracts the polyhedra faces of the pyramid frustums. This means that the normal vectors obtained in Eq. (5.6) are not constant. While it would be needed to compute a new formulation to consider this effect, this work simplifies the problem by considering only the planar faces formed by the contracted vertices. Training is carried out using a batch size of 8192 for 500k iterations using 4 × NVIDIA Tesla V100 GPU per scene. Aside from the use of the EIPE, all other settings remained unchanged from mip-NeRF 360 [43].

5.4 Results

Mean PSNR, SSIM and LPIPS [232] metrics are reported for the Exact-NeRF approach, mip-NeRF [42] and mip-NeRF 360 [43]. Additionally, the DISTS [233] metric is reported since it provides another perceptual quality measurement. Similar to mip-NeRF, an average metric is also reported: the geometric mean of the

Model	PSNR ↑	SSIM ↑	LPIPS ↓	DISTS ↓	Avg ↓
Mip-NeRF	**34.766**	**0.9706**	0.0675	0.0878	**0.0242**
Exact-NeRF (ours)	34.707	0.9705	**0.0667**	**0.0822**	**0.0242**

Table 5.1: Quantitative results comparing mip-NeRF and Exact-NeRF performance on the Blender dataset.

Model	PSNR ↑	SSIM ↑	LPIPS ↓	DISTS ↓	Avg ↓
Mip-NeRF 360	**27.325**	**0.7942**	**0.6559**	**0.2438**	**0.1077**
Exact-NeRF (ours)	27.230	0.7881	0.6569	0.2452	0.1088

Table 5.2: Comparison of the performance of Exact-NeRF with mip-NeRF 360 on the unbounded dataset of Barron *et al.* [43].

MSE $= 10^{-\text{PSNR}/10}$, $\sqrt{1 - \text{SSIM}}$, the LPIPS and the DISTS.

Blender dataset: Table 5.1 presents a quantitative comparison between Exact-NeRF and mip-NeRF. It can be observed that the proposed Exact-NeRF architecture matches the reconstruction performance of mip-NeRF, with a marginal decrease of the PSNR and SSIM and an increment in the LPIPS and DISTS metrics, but with identical average performance. This small decrement in the PSNR and SSIM metrics can be explained by the loss of precision in the calculation of small quantities involved in the EIPE. Alternative formulations that avoid this issue could be used (see Section 5.5), but the intention of Exact-NeRF is to create a general approach for any volumetric positional encoding using the vertices of the volumetric region. Fig. 5.4 shows a qualitative comparison between mip-NeRF and Exact-NeRF. It can be observed that Exact-NeRF is able to match the reconstruction performance of mip-NeRF. A closer examination reveals that Exact-NeRF creates sharper reconstructions in some regions, such as the holes in the Lego scene or the water in the ship scene, which is explained by mip-NeRF approximating the conical frustums as Gaussians. This is consistent with the increase in the LPIPS and DISTS, which are perceptual similarity metrics.

Mip-NeRF 360 dataset: Table 5.2 shows the results for the unbounded mip-NeRF 360 dataset. Despite Exact-NeRF having marginally weaker reconstruction metrics, it shows a competitive performance without any changes to the implementation of the EIPE used earlier with the bounded blender dataset, *i.e.*, the contracted

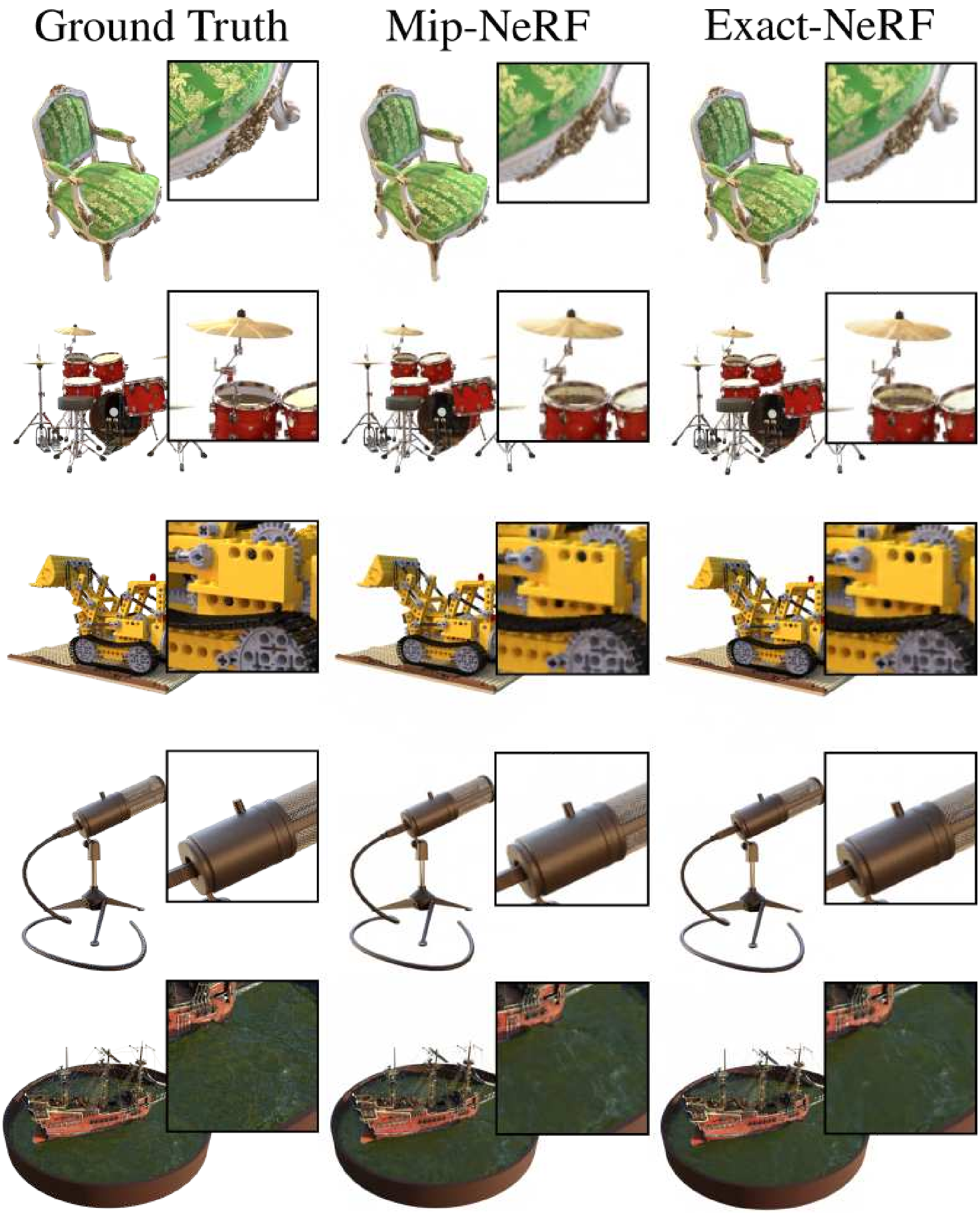


Figure 5.4: Qualitative comparison between mip-NeRF and Exact-NeRF for the blender dataset. The proposed architecture matches the mip-NeRF rendering capability but also produces slightly sharper renderings.

vertices were directly used without any further simplification or linearization, as in mip-NeRF 360 [43]. Similar to the Blender dataset results, this decrement can be explained with the loss of precision, which suggests that an alternative implementation of Eq. (5.20) may be needed. A qualitative comparison is shown in Fig. 5.5. It can

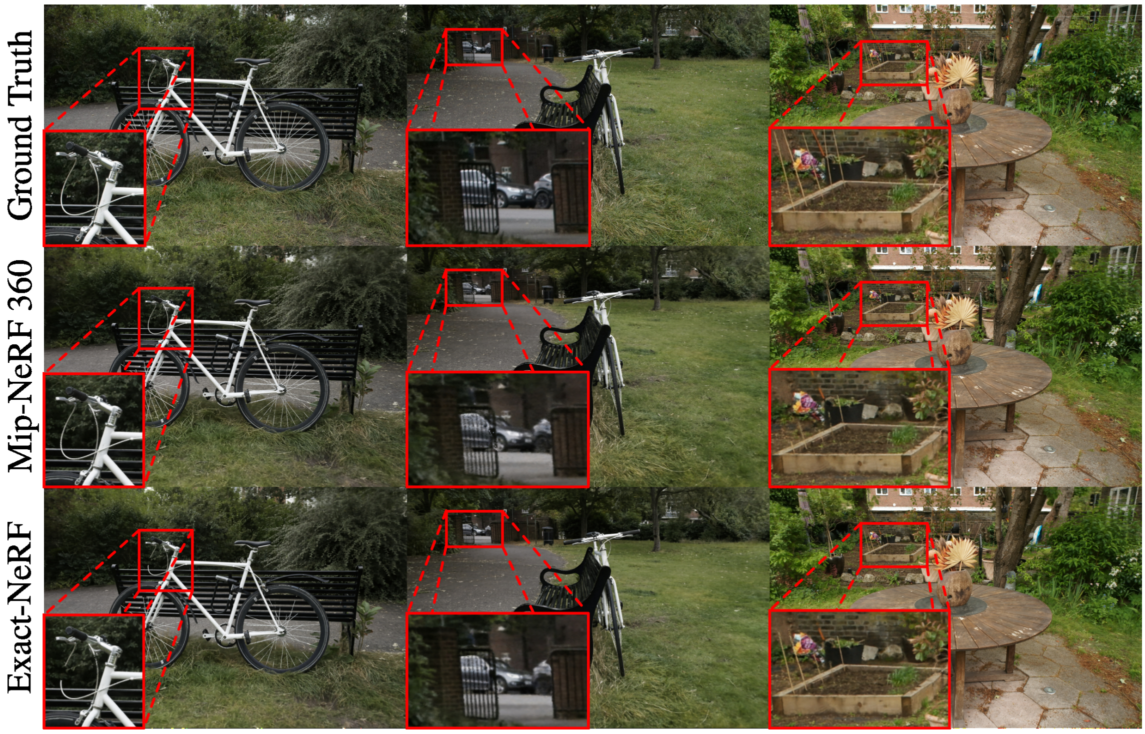


Figure 5.5: Qualitative comparison between mip-NeRF 360 and Exact-NeRF. *Left-column*: the proposed model, similar to mip-NeRF, struggles with tiny vessels. *Middle-column*: Exact-NeRF shows cleaner renderings. *Right-column*: Exact-NeRF has higher quality background reconstruction.

be observed that tiny vessels are more problematic for Exact-NeRF (Fig. 5.5, *left-column*), which can be explained again by the loss of precision. However, it is noted in Fig. 5.5, *middle-column*, that the reconstruction of far regions in mip-NeRF 360 is noisier than Exact-NeRF (see Fig. 5.5, *middle-column*, grill and the car), which is a consequence of the poor approximation of the Gaussian region for far depth of field objects in the scene. Fig. 5.5, *right-column*, reveals another example of a clearer region in the Exact-NeRF reconstruction for the background detail. Fig. 5.6 shows snapshots of the depth estimation for the bicycle, bonsai and garden scenes. Consistent with the colour reconstructions, some background regions have a more detailed estimation. It is also noticed (not shown) that despite Exact-NeRF having a smoother depth estimation, it may show some artifacts in the form of straight lines, which may be caused by the shape of the pyramidal frustums. It is worth reminding that the implementation of the EIPE in mip-NeRF 360 is identical to the EIPE in mip-NeRF.

Impact of Numerical Underflow As seen in Section 5.2, Exact-NeRF may suffer

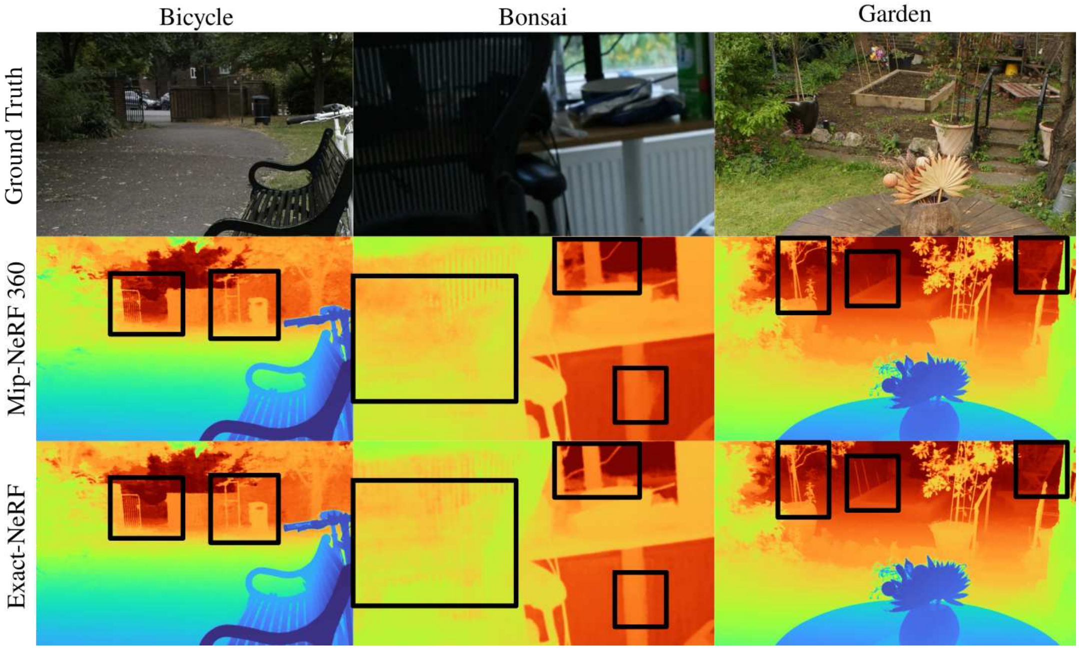
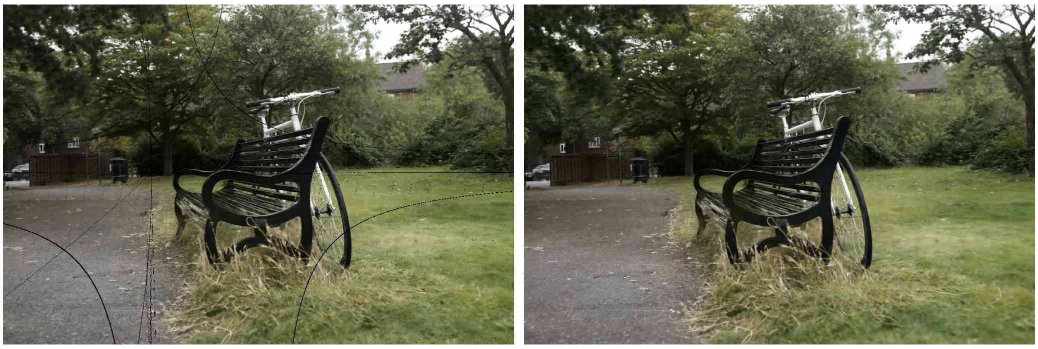


Figure 5.6: Depth estimation for mip-NerF 360 and Exact-NeRF. The Exact-NeRF approach shows better depth estimations for background regions (highlighted in the black boxes), although some artifacts in form of straight lines may appear, which is inherent in the pyramidal shapes.

from numerical underflow when the difference of a component of two points $\Delta = x_{\tau,i} - x_{\tau,j}$ is too close to zero $(\Delta \to 0)$. In the case of this difference being precisely zero, the limit can be found using *l'Hopital's rule*, as it is further developed in Section 5.2.3. However, if this value is not zero but approximately zero, numerical underflow could lead to exploding values in Eq. (5.14). This error hinders the training of the MLP since the IPE is bounded to the interval $[-1, 1]$ by definition (Eq. (2.46)). An example of the effect of numerical underflow in the EIPE applied under the mip-NeRF 360 framework is shown in Fig. 5.7. The black lines are the location of such instances where underflow occurs. The curvature of these lines is a direct consequence of the contracted space used in mip-NeRF 360. In order to eliminate this effect, double precision is used for the calculation of the EIPE. Additionally, all differences of a coordinate which are less than 1×10^{-6} are set to zero and reformulated using *l'Hopital's rule*.



(a) Single Precision (b) Double Precision

Figure 5.7: Numerical underflow artifacts in Exact-NeRF.

5.5 Alternative Parameterisations

As mentioned earlier, the EIPE in Eq. (5.20) can be used for any shape whose vertices are known. However, the computational cost increases if the 3D shape is complex since a larger number of triangular faces will need to be processed. For more efficient methods, we can focus our analysis on specific shapes. Particular to our scenario, we can obtain an alternative EIPE exclusively for a square pyramid (or a parallelepiped, note that this will not be the case for the contraction function in mip-NeRF 360) with a known camera pose $[\mathbf{R}|\mathbf{o}]$ and pixel width ω (similar to $\dot{r}$ in mip-NeRF). From Fig. 5.8, we calculate the volume of the frustum as:

$$V = \int_{t_i}^{t_{i+1}} \int_{-\omega z/2}^{\omega z/2} \int_{-\omega z/2}^{\omega z/2} dx' dy' dz' \tag{5.36}$$

$$V = \frac{\omega^2}{3} \left(t_{i+1}^3 - t_i^3 \right) \; . \tag{5.37}$$

The numerator in Eq. (2.46) for the x coordinate can be obtained in the same way:

$$I_x = \int_{t_i}^{t_{i+1}} \int_{-\omega z/2}^{\omega z/2} \int_{-\omega z/2}^{\omega z/2} \sin(2^l x) dx' dy' dz' \; . \tag{5.38}$$

Since the camera pose is known, we can express x as

$$x = r_{11}x' + r_{12}y' + r_{13}z' + o_1 \, , \tag{5.39}$$

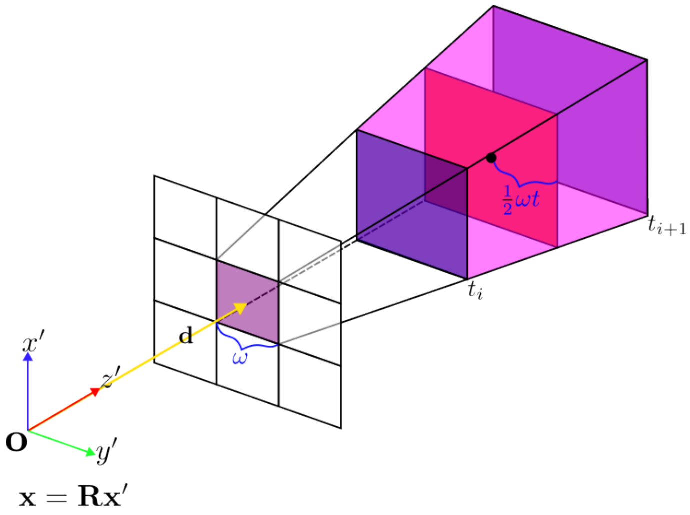


Figure 5.8: Parameterisation of the square pyramid using the pixel width ω.

where r_{ij} is an element of the rotation matrix $\mathbf{R}$ and o_1 is the first element of $\mathbf{o}$. Substituting Eq. (5.39) in Eq. (5.38) (and omitting the integration limits for clarity):

$$I_x = \iiint \sin(2^l(r_{11}x' + r_{12}y' + r_{13}z' + o_1))dx'dy'dz' \,. \tag{5.40}$$

The solution to the integral in Eq. (5.40) is then:

$$I_x = \frac{1}{2^{3l}r_{11}r_{12}} \left[\frac{C_1}{\zeta_1} - \frac{C_2}{\zeta_2} - \frac{C_3}{\zeta_3} + \frac{C_4}{\zeta_4} \right] \,, \tag{5.41}$$

$$C_j = \cos\left(2^l(t_{i+1}\zeta_j + o_1)\right) - \cos\left(2^l(t_i\zeta_j + o_1)\right) \,, \tag{5.42}$$

$$\zeta_j = \boldsymbol{\eta}_j^\top \begin{bmatrix} r_{11} \\ r_{12} \\ r_{13} \end{bmatrix} \,, \tag{5.43}$$

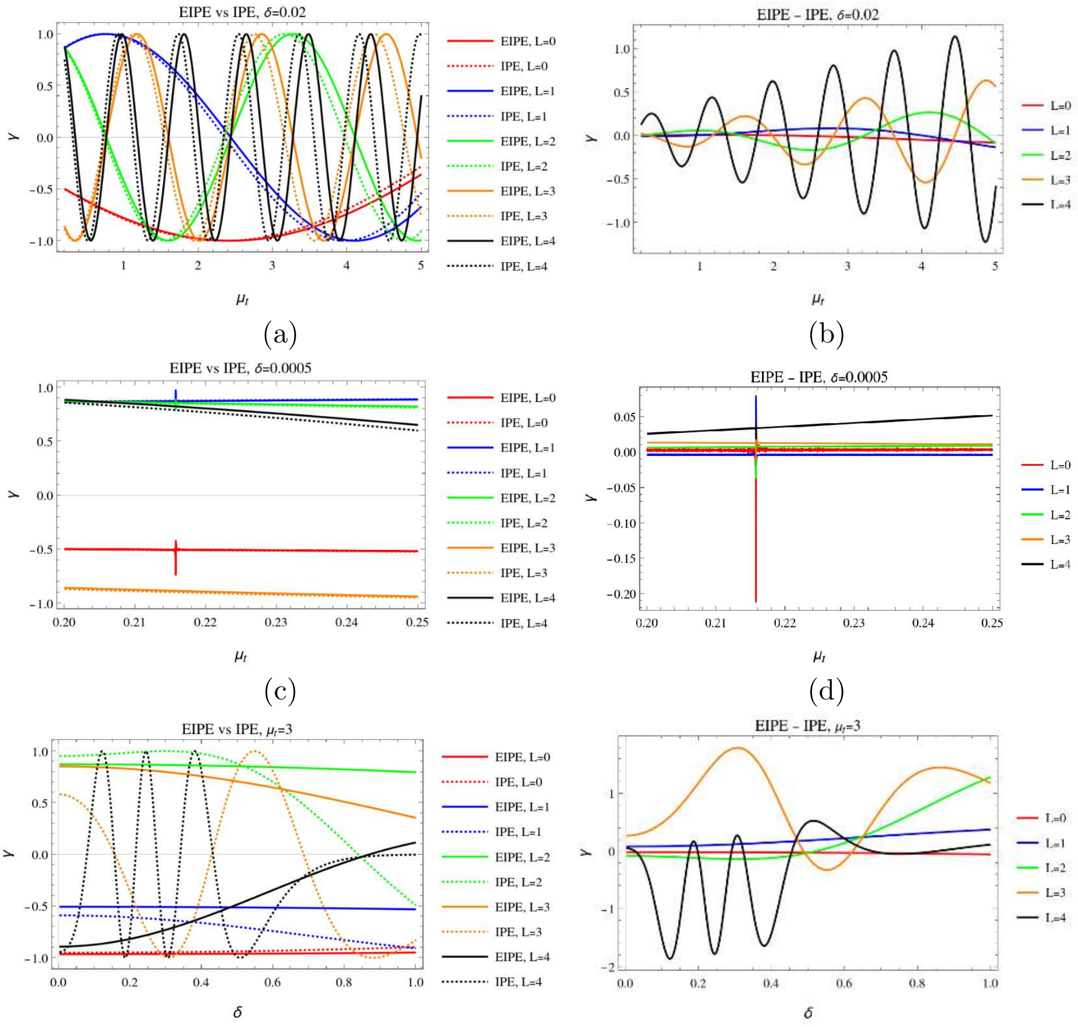
$$\boldsymbol{\eta}_1 = \begin{bmatrix} \frac{\omega}{2} \\ \frac{\omega}{2} \\ 1 \end{bmatrix}, \boldsymbol{\eta}_2 = \begin{bmatrix} -\frac{\omega}{2} \\ \frac{\omega}{2} \\ 1 \end{bmatrix}, \boldsymbol{\eta}_3 = \begin{bmatrix} \frac{\omega}{2} \\ -\frac{\omega}{2} \\ 1 \end{bmatrix}, \boldsymbol{\eta}_4 = \begin{bmatrix} -\frac{\omega}{2} \\ -\frac{\omega}{2} \\ 1 \end{bmatrix} \,. \tag{5.44}$$

Similarly to the EIPE in Eq. (5.20), an indeterminate value arises in Eq. (5.41) for $r_{11} = 0$ and $r_{12} = 0$. For these cases, *l'Hopital's rule* can be used as in Section 5.2.3 or Eq. (5.40) can be solved by substituting $r_{11} = 0$ and $r_{12} = 0$. These calculations are omitted for brevity.

5.6 Numerical Analysis between the IPE and EIPE

The exact value of the EIPE with the approximation in Eq. (2.47) used by mip-NeRF [42] is compared. In Fig. 5.9a the value of the EIPE vs the IPE is contrasted for frustums of length $\delta_i = 0.02$ at different positions along the ray $\mathbf{d}$ and at different positional encoding frequencies L. The values of $\mathbf{d}$, $\mathbf{o}$ and $\mathbf{R}$ correspond to a random pixel of a random image of the blender dataset. It is seen that the approximation is precise for frustums that are near the camera (small μ_t), but it degrades the further it gets. It is also observed that this effect grows faster for larger values of L. This trend is more noticeable in the plot of the error between the EIPE and IPE (Fig. 5.9b), where the magnitude of the error is a periodic function approximately bounded by two lines whose slope seems to grow proportional with L. Furthermore, it is observed that the frequency of the error is also proportional to L. Figs. 5.9c and 5.9d show a similar analysis for small values of μ_t and $\delta_i = 5 \times 10^{-4}$, which correspond to small frustums referring to objects near the camera (mostly associated with foreground objects). In these instances, it is observed that numerical errors occur (seen as jumps around $\mu_t = 0.216$), which is consistent with the analysis of the *Impact of Numerical Underflow* in Section 5.4. Therefore, removing or decreasing this effect could improve foreground reconstruction. A similar analysis for a fixed value of $\mu_t = 3$ and varying δ_i is shown in Figs. 5.9e and 5.9f. Here, a greater error is seen when δ_i increases, which is consistent with the observation made in [43] that the IPE does not approximate well for very elongated Gaussians. Additionally, rapid changes in the IPE are observed for small variations in the length of the frustum (see Fig. 5.9e, IPE $L = 3$ and IPE $L = 4$), which might not be desired. On the other hand, EIPE is more robust to these elongations, meaning that it could be a more reliable parameterisation for distant objects.

Despite the increasing error in the approximation of the IPE for larger values of L, this effect gets mitigated by the nature of the IPE itself, which gives more importance to the components of the positional encoding with smaller frequencies. However, in scenarios with distant backgrounds where more elongated frustum arises, such as in the bicycle scene, Exact-NeRF seems to perform better (Section 5.4). Given that the scenes in the blender and mip-NeRF 360 datasets are composed of one central object



(a) (b)

(c) (d)

(e) (f)

Figure 5.9: Numerical comparison between the IPE and our EIPE. (a) EIPE vs IPE for different values of μ_t and (b) their difference. (c) EIPE vs IPE with respect to the length of the frustum δ_i and (d) their difference.

only, it is difficult to evaluate the performance of the IPE and EIPE formulations for distant objects or scenarios with several objects.

5.7 Conclusion

In this chapter, Exact-NeRF, a novel precise volumetric parameterisation for neural radiance fields (NeRF), is presented. In contrast to conical frustum approximation via a multivariate Gaussian in mip-NeRF [42], Exact-NeRF uses a novel pyramidal parameterisation to encode 3D regions using an Exact Integrated Positional Encoding (EIPE). The EIPE applies the divergence theorem to compute the exact value of the positional encoding (an array of sine and cosines) in a pyramidal frustum using the coordinates of the vertices that define the region. The proposed EIPE methodology can be applied to any such architecture that performs volumetric positional encoding from simple knowledge of the pyramidal frustum vertices without the need for further processing.

Exact-NeRF is compared against mip-NeRF on the blender dataset, showing a matching performance with a marginal decrease in PSNR and SSIM but an overall improvement in the perceptual metrics LPIPS [232] and DISTS [233]. Qualitatively our approach exhibits slightly cleaner and sharper reconstructions of edges than mip-NeRF [42].

Similarly, Exact-NeRF is compared with mip-NeRF 360 [43]. Despite Exact-NeRF showing a marginal decrease in the reconstruction performance metrics, it illustrates the capability of the EIPE on a different architecture without further modification. Exact-NeRF obtains sharper renderings of distant (far depth of field) regions and areas where mip-NeRF 360 presents some noise, but it fails to reconstruct tiny vessels in near regions. The qualitative depth estimations maps also confirm these results. The marginal decrease in performance of Exact-NeRF can be attributed to numerical underflow and some artifacts caused by the choice of a step-function-based square pyramidal parameterisation. In addition, the results suggest using a combined encoding such that the EIPE is used for distance objects, where it is more stable and accurate. Although alternative solutions can be obtained by restricting the analysis to rectangular pyramids, the aim of this chapter is to introduce a general framework that can be applied to any representation of a 3D region with known vertices.

CHAPTER 6

Conclusion

The increasing availability of visual sensing devices, including modalities beyond the visible spectrum, has allowed scenarios with multiple views of the same scene where modern computer vision methods can be introduced for a better scene understanding. Nevertheless, the joint use of multi-view 2D imagery is not a trivial task considering the recent developments in image recognition, which are vastly dominated by deep learning architectures. Multi-view image processing presents additional challenges, such as multi-view consistency, proper feature fusion and accurate 3D representation and encoding. This thesis addresses these problems and proposes novel methods that account for multi-view filtering and better 3D awareness when only 2D images are given, showing that significant gains are obtained with careful consideration of the representation of objects in a scene. The methods presented in this work show that traditional geometry approaches along with deep feature representations can be used to improve the performance of single-view tasks. Despite the advantages shown by the proposed methods, the complex nature of the scenes, such as occlusions or unusual poses of objects, is still a challenge with a direct impact on the performance of our models.

This thesis explores two different tasks, namely multi-view object detection and volumetric representations of neural radiance fields. In the former, two techniques are proposed: the imposition of multi-view geometry constraints to reduce the solution space and the use of the attention mechanism to create 3D-aware features. In the latter, a theoretical formulation for encoding volumetric regions for neural radiance fields is developed, improving the reconstruction of under-represented areas. In this chapter, a summary of the contributions is presented in Section 6.1, while the limitations and future work are explored in Section 6.2.

6.1 Contributions

Multi-view object detection is an important and challenging task in the detection of threat items in X-ray security imagery since objects of interest may appear in unrecognizable poses in some views. In this regard, Chapter 3 explores a multi-view post-processing technique that considers an epipolar (two-views) constraint to penalize detections that are not found in other views. Specifically, given the epipolar line in a target view obtained from the centre of the bounding box of a source view, a multi-view epipolar confidence is assigned to the detections of the target view depending on the distance of their centre to the epipolar line, which is in turn used during non-maximum suppression (NMS). The filtering is carried out such that all pairs of views are constrained simultaneously, yielding a set of detected bounding boxes that are multi-view consistent. Since the proposed approach is detector-agnostic, four different architectures are considered, viz. YOLOv3 [75], YOLOX [46], Faster R-CNN [2] with a Swin Transformer backbone [7] and FCOS [163], and the results are compared against the single-view implementation of the detectors. In all cases, multi-view epipolar filtering increases the MS-COCO [88] average precision (AP) metric without affecting recall, with a maximum increment of 2.2% for the YOLOv3 detector. Similarly, the AP for a single intersection-over-union value of 0.5 is increased for the four detectors, with a maximal increment of 2.9% on the FCOS detector, when compared with its single-view counterpart. Qualitatively, it is seen that the multi-view epipolar confidence improves AP by removing false positive detections and allowing the identification of low-confidence objects that lie near the epipolar line.

In addition to the epipolar confidence, the computation of the fundamental matrix using the bounding box centres as correspondences is tested, given that finding true correspondences among two transmission images using classical approaches is challenging and usually inaccurate [223]. This approach is shown to be reliable in the context of the multi-view epipolar confidence, giving a maximum absolute error of less than three pixels from the centre of a bounding box to the epipolar line with respect to the centre of a corresponding bounding box in another view, although a high variance is observed among some views for the firearm class. These errors and variances are used to model the distance distributions to the epipolar lines, which are used for the multi-view epipolar confidence.

Ablation studies are carried out to assess the multi-view epipolar confidence. Particularly, it is shown that the most critical element is the integration of the epipolar confidence to the score (which is used for NMS), instead of simply using it to disregard detections not close enough to the epipolar line. Furthermore, since the distance to the epipolar line is modelled as a normal distribution, the use of a zero mean against a non-zero mean is also tested, resulting in a small variation favouring the non-zero case. Finally, it is demonstrated that the modelling of these normal distributions for each class performs better than using a class-agnostic distribution, with an AP increase of up to 0.8% on the FCOS detector.

Although the multi-view epipolar confidence filtering technique significantly reduces the number of false positives, it does not improve upon the detection of hard or hidden objects in a scene. For this reason, Chapter 4 introduces the Multi-View Vision Transformer (MVViT), which is based on the Transformer architecture [128]. MVViT is a multi-view layer that uses the attention mechanism to aggregate intermediate features in the backbone of the detector across different views in order to create multi-view aware feature representations. Three modern detectors comprising different architectural paradigms are tested, namely, YOLOX [46], Deformable DETR [47] and Faster R-CNN [2] with a Swin Transformer [7] backbone. Two multi-view datasets with different application contexts are tested: a four-view X-ray security imagery (the same used in Chapter 3) and Wildtrack [106], a seven-view pedestrian surveillance dataset with no fully overlapping field-of-view (FoV). In general, the addition of the MVViT layer increases the detection performance across all detectors in the four-view X-ray dataset, with a maximum AP increment of 3% for the YOLOX detector and 4.8% in a subset of the dataset that considers only two orthogonal views. MVViT increases the performance on the Wildtrack dataset more discreetly, with an increment of only 0.7% on the Swin Transformer detector, while showing small decrements for the YOLOX and Deformable DETR detectors. In contrast with the X-ray dataset, where images appear *overlapped*, the occluded nature of objects in the Wildtrack dataset makes it difficult to get a meaningful aggregated feature vector.

Qualitatively, MVViT detects objects that are missed by single-view detectors, whilst also removing false positives since their aggregated feature vectors do not contain supporting information across the different views. However, in some instances, MVViT may produce some redundancies or still produce some false positives with the same class of another correctly detected object. In addition, an analysis of the attention map given a reference feature vector in one view shows that the attention mechanism focuses on the relevant areas in the other views, confirming that the aggregated features are semantically similar. On the other hand, the attention maps exhibit that attention happens only in small regions, not being able to capture the overall structure of an object.

Multi-view imagery can also be used for image-based rendering, a 3D reconstruction technique that only uses a set of 2D images of the scene and their camera pose. In this regard, Neural Radiance Fields (NeRF) [12] have shown significant advances towards creating high-fidelity 3D reconstructions. In the NeRF method, a high-frequency function (a composition of sine and cosines at different frequencies) is applied to the input coordinates. A subsequent work, mip-NeRF [42], shows that aliasing and blurring are reduced by encoding volumetric regions, defined by conical frustums, instead of point-samples as in the original NeRF implementation. Consequently, the high-frequency encoding function must be applied to the volumetric region through integration over the volume. However, since the integration has no closed-form solution, mip-NeRF approximates it by using the expected value of a multivariate Gaussian that fits the conical frustums. In Chapter 5, a novel parameterisation of the encoded volumetric regions is introduced. This method, named Exact-NeRF, replaces the conical frustums with pyramidal frustums and, by using the divergence theorem, an exact value of the encoded volumetric region is found in terms of the vertices of the pyramid or, more generally, of any polyhedron. The proposed method matches the performance of mip-NeRF and can be extended without further changes into the unbounded parameterisation of mip-NeRF 360 [43]. Despite presenting a slight decrease in performance compared with mip-NeRF 360, Exact-NeRF obtains superior reconstructions of the background since these regions get affected by the multivariate Gaussian parameterisation in the mip-NeRF framework. Furthermore, Exact-NeRF exhibits better depth estimations in the mip-NeRF 360 unbounded dataset.

Despite Exact-NeRF giving an analytical solution to the high-frequency encoding of the volumetric region, the parameterisation in terms of the vertices may lead to singularities when the difference between two vertices' coordinates is close to zero. This numerical underflow may explain the slight decrease in performance when compared to mip-NeRF 360. In order to reduce this effect, *l'Hopital's* rule is used to derive a solution. Additionally, an alternative parameterisation of the strict case of a pyramidal is developed (which is valid for mip-NeRF but not for mip-NeRF 360). Finally, a numerical analysis comparing the encoding of Exact-NeRF with the one of mip-NeRF is presented, showing that the proposed approach is more stable to elongated frustums which occur in far regions of unbounded scenes, explaining why Exact-NeRF is superior for background reconstruction.

6.2 Limitations and Future Work

The methods proposed in Chapters 3 to 5, although exhibiting good performance, present some limitations. In this section, these limitations, along with identified future research directions, are discussed.

6.2.1 Evaluation Datasets

Although there are some multi-view datasets for object detection (see Section 2.2.8), the methods developed in Chapters 3 and 4 require synchronized views that have a common FoV. X-ray security imagery intrinsically has the same FoV for all its views since they observe the same object under the same imaging projection. Along with the restricted access nature of X-ray imagery, most of the X-ray datasets are single-view, making the evaluation of multi-view object detection methods fairly limited. Additionally, the camera parameters (both intrinsic and extrinsic) are not provided, impeding the implementation of any multi-view constraints or relationships. In order to develop more accurate and powerful models, more datasets with broader object categories and complete pose information are needed.

Regarding natural multi-view datasets, such as pedestrian or autonomous vehicle datasets, the fact that not all views share a similar FoV makes multi-view detection difficult. Other anomalies, such as inconsistent annotations, also affect the performance of the proposed methods. For that reason, handling these constraints remains part of future work. Specifically, the masking of shared FoV areas or the re-identification of instances in the same scene are possible extensions to the proposed methods that may directly improve the performance.

6.2.2 Extension to N-view models

Equivalent constraints to the epipolar line for three views are described via the trifocal tensor [51]. This tensor has the property of finding a corresponding point in one view given the correspondences in the other two views. This can be exploited similarly to the epipolar constraints in Chapter 3 to correlate detected bounding boxes in the three views. There are further multi-linear relations found for four or more views [234, 235], which may be used for constraining the solution space of the detections.

No theoretical research has been carried out regarding the multi-view relations of corresponding bounding boxes in a similar way as in points or lines. The exploration and development of such a theory may bring novel constraints and optimisation techniques for multi-view object detection.

6.2.3 Improved multi-view feature fusion

As discussed in Chapter 4, the multi-view attention mechanism in MVViT seems to focus only on highly localized regions in other views. While this still improves the detection accuracy, this limitation impacts the performance of bigger objects, such as laptops in X-ray imagery. Therefore, an improved feature fusion method is needed in order to capture and aggregate the information of the entire object. Moreover, MVViT does not exhibit learning multi-view constraints. Future research on multi-view attention might incorporate these constraints to modify the attention function to give greater importance to feature vectors closer to the epipolar line (or closer to a point correspondence if N-view models are explored).

Another improvement to MVViT feature fusion could be explored by penalizing multi-view inconsistent feature maps, similar to the epipolar filtering in Chapter 3. In this sense, the attention mechanism of MVViT may include information from the epipolar geometry, increasing the attention of features closer to the epipolar line (similar to the work of He *et al*. [179]). In the same context, explicit handling of non-overlapping fields of view could be added by using masked attention layers.

6.2.4 Stable volumetric parameterisations for modern Neural Radiance Fields

One of the main limitations of the pyramidal parameterisation proposed in the Exact-NeRF method of Chapter 5 is that it suffers from numerical underflow when two coordinate components are numerically close. Although this can be alleviated through the *l'Hopital's* rule, it still yields some problems in the foreground reconstruction. Since the mip-NeRF (and mip-NeRF 360) framework is used in most of the recent NeRF models, it is important to further explore and develop more stable volumetric representations of the space.

Alternative parameterisations of the encoding of the spatial coordinates have been explored. Notably, Instant-NGP [206] replaces the positional encoding by using an interpolation function over a multi-resolution grid of learned vectors, achieving accurate NeRF models being trained in seconds. Since the Instant-NGP formulation holds only for point encoding, as in the original implementation of NeRF, volumetric parameterisation in this context is still an open area of research. A first approach has been recently proposed by Zip-NeRF [229], which samples six points that resemble the shape of a cone frustum and performs a weighted average of the encoded vectors to represent the volumetric region. Zip-NeRF provides the anti-aliasing and anti-blurring characteristics of mip-NeRF in the Instant-NGP framework, thus further emphasising the importance of the investigation of such volumetric parameterisations.

6.2.5 Multi-view object detection in Neural Radiance Fields

Finally, the conjunction of both parts of this thesis, *i.e.*, multi-view object detection in NeRF, is a potential area for new research. Hu *et al.* [212] explored object detection in NeRF by implementing a 3D region proposal network (as in Faster R-CNN) using the predicted NeRF voxels. However, modern techniques in object detection may not be easily transferred to this method by the curse of dimensionality. Furthermore, there is a considerably larger body of image datasets than pre-trained NeRF models, making it difficult to scale up towards generalisation. Given that multi-view images with pose information are needed for NeRF, multi-view object detection could be performed in frameworks as in Chapters 3 and 4. In this sense, multi-view 2D detections could be combined into a 3D bounding box via geometrical relationships, as carried out by Rubino *et al.* [236]. In addition, by taking this approach, all the capabilities of state-of-the-art 2D detectors would be available for NeRF detection. Therefore, multi-view object detection in this context is an exciting and under-explored area of future work with direct implications for NeRF research.

Bibliography

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 25, p. 25, 2012. 1, 2.2.1, 2.2.4, 2.2.5

[2] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-time Object Detection with Region Proposal Networks," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 28, 2015. 1, 2.2.1, 2.2.4, 2.2.5, 2.2.5, 2.2.6, 2.2.6, 2.3.4, 3.2.2, 3.3.2, 4, 4.3.2, 6.1, B.1, B.2.2

[3] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-time Object Detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788, 2016. 1, 2.2.1, 2.2.4, 2.2.4, 2.2.6, 4

[4] M. Tan and Q. Le, "Efficientnet: Rethinking Model Scaling for Convolutional Neural Networks," in *International Conference on Machine Learning (ICML)*, pp. 6105–6114, 2019. 1, 2.2.1, 2.2.4

[5] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single Shot Multibox Detector," in *European Conference on Computer Vision (ECCV)*, pp. 21–37, 2016. 1, 2.2.1, 2.2.4, 2.2.4, 2.2.6

[6] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal Loss for Dense Object Detection," in *IEEE International Conference on Computer Vision (ICCV)*, pp. 2980–2988, 2017. 1, 2.2.1, 2.2.5, 2.2.6

[7] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows," in *IEEE International Conference on Computer Vision (ICCV)*, pp. 10012–10022, 2021. 1, 1.2, 2.2.2, 2.2.4, 2.2.4, 2.2.5, 2.2.7, 3.3.2, 4, 4.1, 4.3.2, 6.1, B.1, B.2.2, B.1

[8] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end Object Detection with Transformers," in *European Conference on Computer Vision (ECCV)*, pp. 213–229, 2020. 1, 2.2.2, 2.2.7, 3.2.2, 4, 4.1

[9] C. Weigel and L. L. Kreibich, “Advanced 3D Video Object Synthesis Based on Trilinear Tensors,” in *IEEE International Symposium on Consumer Electronics (ISCE)*, pp. 1–5, 2006. 1

[10] O. J. Woodford, I. Reid, P. Torr, and A. W. Fitzgibbon, “On New View Synthesis Using Multiview Stereo,” in *British Machine Vision Conference (BMVC)*, pp. 110.1–110.10, 2007. 1

[11] Minyoung, L. Yuan-Hong, Z. Ning, L. J. J. S. Shao-Hua, and Huh, “Multi-view to Novel View: Synthesizing Novel Views with Self-learned Confidence,” in *European Conference on Computer Vision (ECCV)*, pp. 162–178, 2018. 1

[12] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, “NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis,” *Communications of the ACM*, vol. 65, pp. 99–106, 2021. 1, 1.1, 2.3, 2.3.1, 2.3.3, 5, 5.3, 6.1

[13] C. J. Holder and T. P. Breckon, “Encoding Stereoscopic Depth Features for Scene Understanding in Off-Road Environments,” in *International Conference on Image Analysis and Recognition (ICIAR)*, pp. 427–434, Springer, 2018. 1.1

[14] N. Smolyanskiy, A. Kamenev, and S. Birchfield, “On the Importance of Stereo for Accurate Depth Estimation: An Efficient Semi-Supervised Deep Neural Network Approach,” in *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1120–1128, 2018. 1.1

[15] Y. Wang, P. Wang, Z. Yang, C. Luo, Y. Yang, and W. Xu, “UnOS: Unified Unsupervised Optical-Flow and Stereo-Depth Estimation by Watching Videos,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8071–8081, 2019. 1.1

[16] D. Baltieri, R. Vezzani, R. Cucchiara, Ákos Utasi, C. Benedek, and T. Szirányi, “Multi-view People Surveillance using 3D Information,” in *IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pp. 1817–1824, 2011. 1.1

[17] R. Han, Y. Wang, H. Yan, W. Feng, and S. Wang, “Multi-View Multi-Human Association With Deep Assignment Network,” *IEEE Transactions on Image Processing*, vol. 31, pp. 1830–1840, 2022. 1.1

[18] Y. Gan, R. Han, L. Yin, W. Feng, and S. Wang, “Self-Supervised Multi-View Multi-Human Association and Tracking,” in *ACM International Conference on Multimedia*, pp. 282–290, 2021. 1.1

[19] Y. Xu, X. Liu, Y. Liu, and S.-C. Zhu, “Multi-view People Tracking via Hierarchical Trajectory Composition,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4256–4265, 2016. 1.1

[20] Y. Xia, D. Yang, Z. Yu, F. Liu, J. Cai, L. Yu, Z. Zhu, D. Xu, A. Yuille, and H. Roth, “Uncertainty-aware Multi-view Co-training for Semi-supervised

Medical Image Segmentation and Domain Adaptation," *Medical Image Analysis*, vol. 65, p. 101766, 2020. 1.1

[21] D. Liu, Y. Gao, Q. Zhangli, L. Han, X. He, Z. Xia, S. Wen, Q. Chang, Z. Yan, M. Zhou, *et al.*, "Transfusion: Multi-view Divergent Fusion for Medical Image Segmentation with Rransformers," in *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pp. 485–495, 2022. 1.1

[22] Z. Li, S. Zhang, J. Zhang, K. Huang, Y. Wang, and Y. Yu, "MVP-Net: Multi-view FPN with Position-aware Attention for Deep Universal Lesion Detection," in *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pp. 13–21, 2019. 1.1

[23] G. Carneiro, J. Nascimento, and A. P. Bradley, "Unregistered Multiview Mammogram Analysis with Pre-trained Deep Learning Models," in *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pp. 652–660, 2015. 1.1

[24] C. von Bastian, A. Schwaninger, and S. Michel, "Do Multi-view X-ray Systems Improve X-ray Image Interpretation in Airport Security Screening?," *Zeitschrift für Arbeitswissenschaft*, vol. 62, pp. 165–173, 2008. 1.1, 2.2.8

[25] W. Luo, J. Xing, A. Milan, X. Zhang, W. Liu, and T.-K. Kim, "Multiple Object Tracking: A Literature Review," *Artificial Intelligence*, vol. 293, p. 103448, 2021. 1.1

[26] O. D. Faugeras, "What Can Be Seen in Three Dimensions with an Uncalibrated Stereo Rig?," in *European Conference on Computer Vision (ECCV)* (G. Sandini, ed.), pp. 563–578, Springer Berlin Heidelberg, 1992. 1.1

[27] R. Hartley, R. Gupta, and T. Chang, "Stereo from Uncalibrated Cameras," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 761–764, 1992. 1.1

[28] R. I. Hartley, "Estimation of Relative Camera Positions for Uncalibrated Cameras," in *European Conference on Computer Vision (ECCV)*, pp. 579–587, 1992. 1.1

[29] Andrew, H. R. A. Martin, and Zisserman, "Self-calibration from Image Triplets," in *European Conference on Computer Vision*, pp. 1–16, 1996. 1.1

[30] C. Baillard, C. Schmid, A. Zisserman, and A. Fitzgibbon, "Automatic Line Matching and 3D Reconstruction of Buildings from Multiple Views," in *ISPRS Conference on Automatic Extraction of GIS Objects from Digital Imagery*, vol. 32, Part 3-2W5, pp. 69–80, 1999. 1.1

[31] C. R. Dyer, *Volumetric Scene Reconstruction from Multiple Views*, pp. 469–489. Springer US, 2001. 1.1

[32] Y. Li, Z. Zhao, J. Fan, and W. Li, "ADR-MVSNet: A Cascade Network for 3D Point Cloud Reconstruction with Pixel Occlusion," *Pattern Recognition*, vol. 125, p. 108516, 2022. 1.1

[33] J. Liu, P. Ji, N. Bansal, C. Cai, Q. Yan, X. Huang, and Y. Xu, "PlaneMVS: 3D Plane Reconstruction From Multi-View Stereo," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8665–8675, 2022. 1.1

[34] H. Xie, H. Yao, X. Sun, S. Zhou, and S. Zhang, "Pix2Vox: Context-Aware 3D Reconstruction From Single and Multi-View Images," in *IEEE International Conference on Computer Vision (ICCV)*, pp. 2690–2698, 2019. 1.1

[35] P. Truong, M. Danelljan, L. V. Gool, and R. Timofte, "Learning Accurate Dense Correspondences and When To Trust Them," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5714–5724, 2021. 1.1

[36] W. Jiang, E. Trulls, J. Hosang, A. Tagliasacchi, and K. M. Yi, "Cotr: Correspondence Transformer for Matching Across Images," in *IEEE International Conference on Computer Vision (ICCV)*, pp. 6207–6217, 2021. 1.1

[37] Z. Xu, S. Bi, K. Sunkavalli, S. Hadap, H. Su, and R. Ramamoorthi, "Deep View Synthesis from Sparse Photometric Images," *ACM Transactions on Graphics (ToG)*, vol. 38, pp. 76:1–76:14, 2019. 1.1

[38] V. R. Gernot and Koltun, "Free View Synthesis," in *European Conference on Computer Vision (ECCV)*, pp. 623–640, 2020. 1.1

[39] G. Riegler and V. Koltun, "Stable View Synthesis," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12216–12225, 2021. 1.1

[40] M. Tancik, V. Casser, X. Yan, S. Pradhan, B. Mildenhall, P. Srinivasan, J. T. Barron, and H. Kretzschmar, "Block-NeRF: Scalable Large Scene Neural View Synthesis," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8248–8258, 2022. 1.1, 1.3, 2.3.2, 2.3.4

[41] A. Corona-Figueroa, J. Frawley, S. Bond-Taylor, S. Bethapudi, H. P. H. Shum, and C. G. Willcocks, "MedNeRF: Medical Neural Radiance Fields for Reconstructing 3D-aware CT-Projections from a Single X-ray," in *Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pp. 3843–3848, 2022. 1.1, 2.3.4

[42] J. T. Barron, B. Mildenhall, M. Tancik, P. Hedman, R. Martin-Brualla, and P. P. Srinivasan, "Mip-NeRF: A Multiscale Representation for anti-aliasing neural radiance fields," in *IEEE International Conference on Computer Vision (ICCV)*, pp. 5855–5864, 2021. 1.1, 1.2, 1.3, 1.4, 2.3.1, 2.3.2, 2.9, 2.3.3, 5, 5.1, 5.2, 5.2, 5.2.2, 5.4, 5.6, 5.7, 6.1

[43] J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. Hedman, "Mip-NeRF 360: Unbounded Anti-Aliased Neural Radiance Fields," in *IEEE*

Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5470–5479, 2022. 1.1, 1.2, 1.3, 1.4, 2.3.2, 2.3.3, 5, 5.1, 5.1, 5.1, 5.2, 5.2.2, 5.3, 5.4, 5.2, 5.4, 5.6, 5.7, 6.1

[44] K. Zhang, G. Riegler, N. Snavely, and V. Koltun, "Nerf++: Analyzing and Improving Neural Radiance Fields," *arXiv preprint arXiv:2010.07492*, 2020. 1.1, 1.3, 2.3.3

[45] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," in *International Conference on Learning Representations (ICLR)*, 2020. 1.2, 2.2.1, 2.8, 2.2.7, 4.1

[46] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "YOLOX: Exceeding YOLO Series in 2021," *arXiv preprint arXiv:2107.08430*, 2021. 1.2, 2.2.6, 3.3.2, 4, 4.1, 4.3.2, 6.1, B.1, B.2.2, B.1

[47] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable DETR: Deformable Transformers for End-to-End Object Detection," in *International Conference on Learning Representations (ICLR)*, 2020. 1.2, 2.2.7, 2.2.8, 3.2.2, 4, 4.1, 4.3.2, 6.1

[48] X. Yan, S. Hu, Y. Mao, Y. Ye, and H. Yu, "Deep Multi-view Learning Methods: A Review," *Neurocomputing*, vol. 448, pp. 106–129, 2021. 1.3

[49] T. Neff, P. Stadlbauer, M. Parger, A. Kurz, J. H. Mueller, C. R. A. Chaitanya, A. S. Kaplanyan, and M. Steinberger, "DONeRF: Towards Real-Time Rendering of Compact Neural Radiance Fields using Depth Oracle Networks," *Computer Graphics Forum*, vol. 40, 2021. 1.3, 5.1

[50] M. Niemeyer, J. T. Barron, B. Mildenhall, M. S. M. Sajjadi, A. Geiger, and N. Radwan, "RegNeRF: Regularizing Neural Radiance Fields for View Synthesis From Sparse Inputs," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5480–5490, 2022. 1.3, 2.3.2, 2.3.3

[51] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second ed., 2004. 2.1, 1, 2.1.4, 2.2.8, 3.1, 3.2.1, 6.2.2

[52] M. Spetsakis and J. Y. Aloimonos, "A Multi-frame Approach to Visual Motion Perception," *International Journal of Computer Vision*, vol. 6, pp. 245–255, 1991. 2.1.4

[53] R. I. Hartley *et al.*, "Projective Reconstruction from Line Correspondences," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 903–907, 1994. 2.1.4

[54] B. Triggs, "Matching Constraints and the Joint Image," in *IEEE International Conference on Computer Vision (ICCV)*, pp. 338–343, 1995. 2.1.4

[55] R. Vidal and D. Abretske, "Nonrigid Shape and Motion from Multiple Perspective Views," in *European Conference on Computer Vision (ECCV)*, pp. 205–218, 2006. 2.1.4

[56] C. Tomasi and T. Kanade, "Shape and Motion from Image Streams: A Factorization Method.," *Proceedings of the National Academy of Sciences*, vol. 90, pp. 9795–9802, 1993. 2.1.4

[57] D. G. Lowe, "Distinctive Image Features from Scale-invariant Keypoints," *International Journal of Computer Vision*, vol. 60, pp. 91–110, 2004. 2.2.1

[58] Tinne, V. G. L. B. Herbert, and Tuytelaars, "SURF: Speeded Up Robust Features," in *European Conference on Computer Vision (ECCV)*, pp. 404–417, 2006. 2.2.1

[59] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. 886–893, 2005. 2.2.1

[60] P. Felzenszwalb, D. McAllester, and D. Ramanan, "A Discriminatively Trained, Multiscale, Deformable Part Model," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–8, 2008. 2.2.1

[61] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016. 2.2.1, 2.2.4, 2.2.5, 2.2.6, 4.1, 4.3.2

[62] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going Deeper with Convolutions," in *IEEE Conference on computer Vision and Pattern Recognition (CVPR)*, pp. 1–9, 2015. 2.2.1, 2.2.6

[63] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in *International Conference on Learning Representations (ICLR)*, 2015. 2.2.1, 2.2.4, 2.2.4, 2.2.5, 4.1

[64] I. O. Tolstikhin, N. Houlsby, A. Kolesnikov, L. Beyer, X. Zhai, T. Unterthiner, J. Yung, A. Steiner, D. Keysers, J. Uszkoreit, *et al.*, "MLP-mixer: An All-MLP Architecture for Vision," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 34, pp. 24261–24272, 2021. 2.2.1

[65] D. P. Kingma and J. Ba, "ADAM: A Method for Stochastic Optimization," in *International Conference on Learning Representations (ICLR)*, 2015. 2.2.1, 3.3.2, 5.3, A.5.3, B.1

[66] G. Klambauer, T. Unterthiner, A. Mayr, and S. Hochreiter, "Self-normalizing Neural Networks," in *Advances in Neural Information Processing Systems (NeurIPS)*, p. 30, 2017. 2.2.1

[67] D. Hendrycks and K. Gimpel, "Gaussian Error Linear Units (GELUs)," *arXiv preprint arXiv:1606.08415*, 2016. 2.2.1

[68] V. Sitzmann, J. Martel, A. Bergman, D. Lindell, and G. Wetzstein, "Implicit Neural Representations with Periodic Activation Functions," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, pp. 7462–7473, 2020. 2.2.1

[69] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual Losses for Real-time Style Transfer and Super-resolution," in *European Conference on Computer Vision (ECCV)*, pp. 694–711, 2016. 2.2.1, A.4.3

[70] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A Unified Embedding for Face Recognition and Clustering," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 815–823, 2015. 2.2.1

[71] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580–587, 2014. 2.2.1, 2.2.4, 2.2.5, B.2.2

[72] R. Girshick, "Fast R-CNN," in *IEEE International Conference on Computer Vision (ICCV)*, pp. 1440–1448, 2015. 2.2.1, 2.2.5, 2.2.5, 2.2.6, 4

[73] C.-Y. Wang, H.-Y. M. Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, and I.-H. Yeh, "CSPNet: A New Backbone that Can Enhance Learning Capability of CNN," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 390–391, 2020. 2.2.1, 4.3.2

[74] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature Pyramid Networks for Object Detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2117–2125, 2017. 2.2.1, 2.2.4, 2.2.5, 2.2.6, 2.2.7

[75] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," *arXiv preprint arXiv:1804.02767*, 2018. 2.2.1, 2.2.6, 3.2.2, 3.3.2, 4, 6.1, B.2.2

[76] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4510–4520, 2018. 2.2.1, 2.2.5

[77] Z. Zou, K. Chen, Z. Shi, Y. Guo, and J. Ye, "Object Detection in 20 Years: A Survey," *Proceedings of the IEEE*, vol. 111, pp. 257–276, 2023. 2.2.1, 2.2.4

[78] M. Everingham, S. M. A. Eslami, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal Visual Object Classes Challenge: A Retrospective," *International Journal of Computer Vision*, vol. 111, pp. 98–136, 2015. 2.2.2, 2.2.3, 2.2.3, B.1

[79] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision*, vol. 115, pp. 211–252, 2015. 2.2.2, 2.2.4, B.1

[80] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A ConvNet for the 2020s," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11976–11986, 2022. 2.2.2

[81] Z. Liu, H. Hu, Y. Lin, Z. Yao, Z. Xie, Y. Wei, J. Ning, Y. Cao, Z. Zhang, L. Dong, F. Wei, and B. Guo, "Swin Transformer V2: Scaling Up Capacity and Resolution," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12009–12019, 2022. 2.2.2, 2.2.5, 2.2.7

[82] M. Wortsman, G. Ilharco, S. Y. Gadre, R. Roelofs, R. Gontijo-Lopes, A. S. Morcos, H. Namkoong, A. Farhadi, Y. Carmon, S. Kornblith, *et al.*, "Model Soups: Averaging Weights of Multiple Fine-tuned Models Improves Accuracy without Increasing Inference Time," in *International Conference on Machine Learning (ICML)*, pp. 23965–23998, 2022. 2.2.2

[83] A. Brock, S. De, S. L. Smith, and K. Simonyan, "High-performance Large-scale Image Recognition Wwthout Normalization," in *International Conference on Machine Learning (ICML)*, pp. 1059–1071, 2021. 2.2.2

[84] W. Chen, X. Du, F. Yang, L. Beyer, X. Zhai, T.-Y. Lin, H. Chen, J. Li, X. Song, Z. Wang, *et al.*, "A Simple Single-Scale Vision Transformer for Object Detection and Instance Segmentation," in *European Conference on Computer Vision (ECCV)*, pp. 711–727, 2022. 2.2.2

[85] P. Sermanet, C. Lynch, Y. Chebotar, J. Hsu, E. Jang, S. Schaal, S. Levine, and G. Brain, "Time-contrastive Networks: Self-supervised Learning from Video," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1134–1141, 2018. 2.2.2

[86] R. Tolosana, R. Vera-Rodriguez, J. Fierrez, A. Morales, and J. Ortega-Garcia, "Deepfakes and Beyond: A Survey of Face Manipulation and Fake Detection," *Information Fusion*, vol. 64, pp. 131–148, 2020. 2.2.2

[87] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-Resolution Image Synthesis with Latent Diffusion Models," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10684–10695, 2022. 2.2.2

[88] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common Objects in Context," in *European Conference on Computer Vision (ECCV)*, pp. 740–755, 2014. 2.2.2, 2.2.3, 2.1, 3.3.2, 4.3, 6.1, A.5.2, B.1, B.3, B.2.3, B.2.4

[89] J. Yang, C. Li, X. Dai, and J. Gao, "Focal Modulation Networks," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 35, pp. 4203–4217, 2022. 2.2.2

[90] M. Xu, Z. Zhang, H. Hu, J. Wang, L. Wang, F. Wei, X. Bai, and Z. Liu, "End-to-End Semi-Supervised Object Detection with Soft Teacher," in *IEEE International Conference on Computer Vision (ICCV)*, pp. 3060–3069, 2021. 2.2.2

[91] X. Dai, Y. Chen, B. Xiao, D. Chen, M. Liu, L. Yuan, and L. Zhang, "Dynamic Head: Unifying Object Detection Heads With Attentions," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7373–7382, 2021. 2.2.2

[92] A. Gupta, P. Dollar, and R. Girshick, "LVIS: A Dataset for Large Vocabulary Instance Segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5356–5364, 2019. 2.2.2

[93] S. Shao, Z. Li, T. Zhang, C. Peng, G. Yu, X. Zhang, J. Li, and J. Sun, "Objects365: A Large-scale, High-quality Dataset for Object Detection," in *IEEE International Conference on Computer Vision (ICCV)*, pp. 8430–8439, 2019. 2.2.2

[94] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, A. Kolesnikov, T. Duerig, and V. Ferrari, "The Open Images Dataset V4: Unified Image Classification, Object Detection, and Visual Relationship Detection at Scale," *International Journal of Computer Vision*, vol. 128, pp. 1956–1981, 2020. 2.2.2, 2.2.3

[95] L. Cai, Z. Zhang, Y. Zhu, L. Zhang, M. Li, and X. Xue, "BigDetection: A Large-scale Benchmark for Improved Object Detector Pre-training," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4777–4787, 2022. 2.2.2

[96] A. Geiger, P. Lenz, and R. Urtasun, "Are We Ready for Autonomous Driving? The KITTI Vision Benchmark Suite," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3354–3361, 2012. 2.2.2

[97] X. Huang, P. Wang, X. Cheng, D. Zhou, Q. Geng, and R. Yang, "The Apolloscape Open Dataset for Autonomous Driving and its Application," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, pp. 2702–2719, 2019. 2.2.2

[98] M. Li, Y.-X. Wang, and D. Ramanan, "Towards Streaming Perception," in *European Conference on Computer Vision (ECCV)*, pp. 473–488, 2020. 2.2.2

[99] Q.-H. Pham, P. Sevestre, R. S. Pahwa, H. Zhan, C. H. Pang, Y. Chen, A. Mustafa, V. Chandrasekhar, and J. Lin, "A 3D Dataset: Towards Autonomous Driving in Challenging Environments," in *International Conference in Robotics and Automation (ICRA)*, 2020. 2.2.2

[100] X. Jia, C. Zhu, M. Li, W. Tang, and W. Zhou, "LLVIP: A Visible-infrared Paired Dataset for Low-light Vision," in *IEEE International Conference on Computer Vision (ICCV)*, pp. 3496–3504, 2021. 2.2.2

[101] Y. P. Loh and C. S. Chan, "Getting to Know Low-light Images with The Exclusively Dark Dataset," *Computer Vision and Image Understanding*, vol. 178, pp. 30–42, 2019. 2.2.2

[102] S. Zhang, R. Benenson, and B. Schiele, "CityPersons: A Diverse Dataset for Pedestrian Detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3213–3221, 2017. 2.2.2

[103] S. Shao, Z. Zhao, B. Li, T. Xiao, G. Yu, X. Zhang, and J. Sun, "Crowd-Human: A Benchmark for Detecting Human in a Crowd," *arXiv preprint arXiv:1805.00123*, 2018. 2.2.2

[104] S. Zhang, Y. Xie, J. Wan, H. Xia, S. Z. Li, and G. Guo, "Widerperson: A Diverse Dataset for Dense Pedestrian Detection in the Wild," *IEEE Transactions on Multimedia*, vol. 22, pp. 380–393, 2019. 2.2.2

[105] M. Braun, S. Krebs, F. B. Flohr, and D. M. Gavrila, "EuroCity Persons: A Novel Benchmark for Person Detection in Traffic Scenes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, pp. 1844–1861, 2019. 2.2.2

[106] T. Chavdarova, P. Baqué, S. Bouquet, A. Maksai, C. Jose, T. Bagautdinov, L. Lettry, P. Fua, L. V. Gool, and F. Fleuret, "WILDTRACK: A Multi-camera HD Dataset for Dense Unscripted Pedestrian Detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5030–5039, 2018. 2.2.2, 2.2.8, 2.2, 4, 4.3.1, 4.3, 6.1

[107] P. Zhu, L. Wen, D. Du, X. Bian, H. Ling, Q. Hu, H. Wu, Q. Nie, H. Cheng, C. Liu, *et al.*, "VisDrone-VDT2018: The Vision Meets Drone Video Detection and Tracking Challenge Results," in *European Conference on Computer Vision (ECCV) Workshops*, 2018. 2.2.2

[108] X. Yu, Y. Gong, N. Jiang, Q. Ye, and Z. Han, "Scale Match for Tiny Person Detection," in *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1257–1265, 2020. 2.2.2

[109] G.-S. Xia, X. Bai, J. Ding, Z. Zhu, S. Belongie, J. Luo, M. Datcu, M. Pelillo, and L. Zhang, "DOTA: A Large-Scale Dataset for Object Detection in Aerial Images," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3974–3983, 2018. 2.2.2

[110] D. Lam, R. Kuzma, K. McGee, S. Dooley, M. Laielli, M. Klaric, Y. Bulatov, and B. McCord, "xView: Objects in Context in Overhead Imagery," *arXiv preprint arXiv:1802.07856*, 2018. 2.2.2

[111] T. N. Mundhenk, G. Konjevod, W. A. Sakla, and K. Boakye, "A Large Contextual Dataset for Classification, Detection and Counting of Cars with Deep Learning," in *European Conference on Computer Vision (ECCV)*, pp. 785–800, 2016. 2.2.2

[112] Debesh, T. Vajira, H. Pål, H. H. L., R. M. A. H. S. A., and Jha, "The EndoTect 2020 Challenge: Evaluation and Comparison of Classification, Segmentation and Inference Time for Endoscopy," in *Internation Conference on Pattern Recognition (ICPR) Workshops and Challenges*, pp. 263–274, 2021. 2.2.2

[113] J. I. Orlando, H. Fu, J. B. Breda, K. van Keer, D. R. Bathula, A. Diaz-Pinto, R. Fang, P.-A. Heng, J. Kim, J. Lee, *et al.*, "Refuge Challenge: A Unified Framework for Evaluating Automated Methods for Glaucoma Assessment from Fundus Photographs," *Medical Image Analysis*, vol. 59, p. 101570, 2020. 2.2.2

[114] K. Yan, X. Wang, L. Lu, and R. M. Summers, "DeepLesion: Automated Mining of Large-scale Lesion Annotations and Universal Lesion Detection with Deep Learning," *Journal of Medical Imaging*, vol. 5, p. 36501, 2018. 2.2.2

[115] J. Liu, J. Lian, and Y. Yu, "ChestX-Det10: Chest X-ray Dataset on Detection of Thoracic Abnormalities," *arXiv preprint arXiv:2006.10550v3*, 2020. 2.2.2

[116] C. Miao, L. Xie, F. Wan, chi Su, H. Liu, jianbin Jiao, and Q. Ye, "SIXray: A Large-scale Security Inspection X-ray Benchmark for Prohibited Item Discovery in Overlapping Images," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2119–2128, 2019. 2.2.2, A.1, B.1, B.2.1, B.3, B.4

[117] D. Mery, V. Riffo, U. Zscherpel, G. Mondragón, I. Lillo, I. Zuccar, H. Lobel, and M. Carrasco, "GDXray: The Database of X-ray Images for Nondestructive Testing," *Journal of Nondestructive Evaluation*, vol. 34, pp. 1–12, 2015. 2.2.2

[118] M. Caldwell and L. D. Griffin, "Limits on Transfer Learning from Photographic Image Data to X-ray Threat Detection," *Journal of X-ray Science and Technology*, vol. 27, pp. 1007–1020, 2019. 2.2.2

[119] Y. Wei, R. Tao, Z. Wu, Y. Ma, L. Zhang, and X. Liu, "Occluded Prohibited Items Detection: An X-ray Security Inspection Benchmark and De-occlusion Attention Module," in *ACM International Conference on Multimedia*, pp. 138–146, 2020. 2.2.2, A.1, B.1, B.2.1, B.3, B.4

[120] S. Akcay, M. E. Kundegorski, C. G. Willcocks, and T. P. Breckon, "Using Deep Convolutional Neural Network Architectures for Object Classification and Detection within X-ray Baggage Security Imagery," *IEEE Transactions on Information Forensics and Security*, vol. 13, pp. 2203–2215, 2018. 2.2.2, 2.2.8, 2.2

[121] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes (VOC) Challenge," *International Journal of Computer Vision*, vol. 88, pp. 303–338, 2010. 2.4, 2.2.3, 2.1, 2.2.6

[122] J. Hosang, R. Benenson, P. Dollár, and B. Schiele, "What Makes For Effective Detection Proposals?," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, pp. 814–830, 2015. 2.2.3, 2.1

[123] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely Connected Convolutional Networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2261–2269, 2017. 2.2.4

[124] Z. Li, C. Peng, G. Yu, X. Zhang, Y. Deng, and J. Sun, "Detnet: Design Backbone for Object Detection," in *European Conference on Computer Vision (ECCV)*, pp. 334–350, 2018. 2.2.4

[125] Y. Chen, T. Yang, X. Zhang, G. MENG, X. Xiao, and J. Sun, "DetNAS: Backbone Search for Object Detection," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 32, 2019. 2.2.4, 2.2.4

[126] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path Aggregation Network for Instance Segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8759–8768, 2018. 2.2.4

[127] S. Liu, D. Huang, *et al.*, "Receptive Field Block Net for Accurate and Fast Object Detection," in *European Conference on Computer Vision (ECCV)*, pp. 385–400, 2018. 2.2.4

[128] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Łukasz Kaiser, and I. Polosukhin, "Attention is All You Need," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, 2017. 2.2.4, 2.2.7, 2.7, 4.1, 6.1

[129] J. Guo, K. Han, Y. Wang, C. Zhang, Z. Yang, H. Wu, X. Chen, and C. Xu, "Hit-detector: Hierarchical Trinity Architecture Search for Object Detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11405–11414, 2020. 2.2.4, 2.3.3

[130] K. Chen, J. Pang, J. Wang, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Shi, W. Ouyang, *et al.*, "Hybrid Task Cascade for Instance Segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4974–4983, 2019. 2.2.4

[131] B. Jiang, R. Luo, J. Mao, T. Xiao, and Y. Jiang, "Acquisition of Localization Confidence for Accurate Object Detection," in *European Conference on Computer Vision (ECCV)*, pp. 784–799, 2018. 2.2.4

[132] S. Gidaris and N. Komodakis, "Attend Refine Repeat: Active Box Proposal Generation via In-Out Localization," in *British Machine Vision Conference (BMVC)*, pp. 90.1–90.13, 2016. 2.2.4

[133] B. Yang, J. Yan, Z. Lei, and S. Z. Li, "Craft Objects from Images," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6043–6051, 2016. 2.2.4

[134] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, "Random Erasing Data Augmentation," *AAAI Conference on Artificial Intelligence*, vol. 34, pp. 13001–13008, 2020. 2.2.4

[135] T. DeVries and G. W. Taylor, "Improved Regularization of Convolutional Neural Networks with Cutout," *arXiv preprint arXiv:1708.04552*, 2017. 2.2.4

[136] P. Chen, S. Liu, H. Zhao, and J. Jia, “GridMask Data Augmentation,” *arXiv preprint arXiv:2001.04086*, 2020. 2.2.4

[137] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo, “CutMix: Regularization Strategy to Train Strong Classifiers With Localizable Features,” in *IEEE International Conference on Computer Vision (ICCV)*, pp. 6023–6032, 2019. 2.2.4

[138] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, “mixup: Beyond Empirical Risk Minimization,” in *International Conference on Learning Representations (ICLR)*, 2018. 2.2.4

[139] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “YOLOv4: Optimal Speed And Accuracy of Object Detection,” *arXiv preprint arXiv:2004.10934*, 2020. 2.2.4, 2.2.6, 4

[140] C. Shorten and T. M. Khoshgoftaar, “A Survey on Image Data Augmentation for Deep Learning,” *Journal of Big Data*, vol. 6, pp. 1–48, 2019. 2.2.4

[141] P. Kaur, B. S. Khehra, and E. B. S. Mavi, “Data Augmentation for Object Detection: A Review,” in *IEEE International Midwest Symposium on Circuits and Systems (MWSCAS)*, pp. 537–543, 2021. 2.2.4

[142] N. Bodla, B. Singh, R. Chellappa, and L. S. Davis, “Soft-NMS Improving Object Detection with One Line of Code,” in *IEEE International Conference on Computer Vision (ICCV)*, pp. 5561–5569, 2017. 2.2.4, 2.6

[143] Q. Zhou, C. Yu, C. Shen, Z. Wang, and H. Li, “Object Detection Made Simpler by Eliminating Heuristic NMS,” *IEEE Transactions on Multimedia*, pp. 1–10, 2023. 2.2.4

[144] D. Park, Y. Seo, D. Shin, J. Choi, and S. Y. Chun, “A Single Multi-task Deep Neural Network with Post-processing for Object Detection with Reasoning and Robotic Grasp Detection,” in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7300–7306, 2020. 2.2.4

[145] A. Sabater, L. Montesano, and A. C. Murillo, “Robust and Efficient Post-processing for Video Object Detection,” in *IEEE International Conference on Intelligent Robots and Systems (IROS)*, pp. 10536–10542, 2020. 2.2.4

[146] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, “Generalized Intersection over Union: A Metric and A Loss for Bounding Box Regression,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 658–666, 2019. 2.2.4, 2.2.7

[147] J. Yu, Y. Jiang, Z. Wang, Z. Cao, and T. Huang, “Unitbox: An Advanced Object Detection Network,” in *ACM International Conference on Multimedia*, pp. 516–520, 2016. 2.2.4

[148] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, “Distance-IoU loss: Faster and Better Learning for Bounding Box Regression,” in *AAAI Conference on Artificial Intelligence*, vol. 34, pp. 12993–13000, 2020. 2.2.4

[149] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, “Overfeat: Integrated Recognition, Localization and Detection Using Convolutional Networks,” in *International Conference on Learning Representations (ICLR)*, 2014. 2.2.5

[150] J. R. R. Uijlings, K. E. A. V. D. Sande, T. Gevers, and A. W. M. Smeulders, “Selective Search for Object Recognition,” *International Journal of Computer Vision*, vol. 104, pp. 154–171, 2013. 2.2.5

[151] S. Fidler, R. Mottaghi, A. Yuille, and R. Urtasun, “Bottom-Up Segmentation for Top-Down Detection,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3294–3301, 2013. 2.2.5

[152] K. He, X. Zhang, S. Ren, and J. Sun, “Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, pp. 1904–1916, 2015. 2.2.5

[153] Hong, M. Bingpeng, W. Naiyan, C. X. Z. Hongkai, and Chang, “Dynamic R-CNN: Towards High Quality Object Detection via Dynamic Training,” in *European Conference on Computer Vision (ECCV)*, pp. 260–275, 2020. 2.2.5

[154] W. Chu and D. Cai, “Deep Feature Based Contextual Model for Object Detection,” *Neurocomputing*, vol. 275, pp. 1035–1042, 2018. 2.2.5

[155] J. Dai, Y. Li, K. He, and J. Sun, “R-FCN: Object Detection Via Region-based Fully Convolutional Networks,” in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 29, p. 29, 2016. 2.2.5

[156] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask R-CNN,” in *IEEE International Conference on Computer Vision (ICCV)*, pp. 2961–2969, 2017. 2.2.5

[157] Z. Cai and N. Vasconcelos, “Cascade R-CNN: Delving into High Quality Object Detection,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6154–6162, 2018. 2.2.5, A.5.2, A.4, B.1, B.2.2, B.1

[158] J. Redmon and A. Farhadi, “YOLO9000: Better, Faster, Stronger,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7263–7271, 2017. 2.2.6, 4, 4.1

[159] S. Ioffe and C. Szegedy, “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift,” in *International Conference on Machine Learning (ICML)*, pp. 448–456, 2015. 2.2.6

[160] C. Li, L. Li, H. Jiang, K. Weng, Y. Geng, L. Li, Z. Ke, Q. Li, M. Cheng, W. Nie, *et al.*, “YOLOv6: A Single-stage Object Detection Framework for Industrial Applications,” *arXiv preprint arXiv:2209.02976*, 2022. 2.2.6

[161] C.-Y. Wang, I.-H. Yeh, and H.-Y. M. Liao, “You Only Learn One Representation: Unified Network for Multiple Tasks,” *Journal of Information Science and Engineering*, vol. 39, pp. 691–709, 2023. 2.2.6

[162] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, “YOLOv7: Trainable Bag-of-freebies Sets New State-of-the-art for Real-time Object Detectors,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7464–7475, 2023. 2.2.6

[163] Z. Tian, C. Shen, H. Chen, and T. He, “Fcos: Fully Convolutional One-stage Object Detection,” in *IEEE International Conference on Computer Vision (ICCV)*, pp. 9627–9636, 2019. 2.2.6, 3.3.2, 4, 6.1

[164] H. Law and J. Deng, “CornerNet: Detecting Objects as Paired Keypoints,” in *European Conference on Computer Vision (ECCV)*, 2018. 2.2.6

[165] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, “Centernet: Keypoint Triplets for Object Detection,” in *IEEE International Conference on Computer Vision (ICCV)*, pp. 6569–6578, 2019. 2.2.6, 3.2.2, 4

[166] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” in *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pp. 4171–4186, 2019. 2.2.7

[167] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, *et al.*, “Language Models Are Few-shot Learners,” *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, pp. 1877–1901, 2020. 2.2.7

[168] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann, *et al.*, “Palm: Scaling Language Modeling with Pathways,” *arXiv preprint arXiv:2204.02311*, 2022. 2.2.7

[169] Y. Li, H. Mao, R. Girshick, and K. He, “Exploring Plain Vision Transformer Backbones for Object Detection,” in *European Conference on Computer Vision (ECCV)*, pp. 280–296, 2022. 2.2.7

[170] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, “Training Data-efficient Image Transformers & Distillation Through Attention,” in *International Conference on Machine Learning (ICML)*, pp. 10347–10357, 2021. 2.2.7

[171] J. Deng and K. Czarnecki, “MLOD: A Multi-view 3D Object Detection Based on Robust Feature Fusion Method,” in *IEEE Intelligent Transportation Systems Conference (ITSC)*, pp. 279–284, 2019. 2.2.8

[172] A. S. Nassar, S. Lefèvre, and J. D. Wegner, "Simultaneous Multi-view Instance Detection with Learned Geometric Soft-constraints," in *IEEE International Conference on Computer Vision (ICCV)*, pp. 6559–6568, 2019. 2.2.8

[173] Y. Hou, L. Zheng, and S. Gould, "Multiview Detection with Feature Perspective Transformation," in *European Conference on Computer Vision (ECCV)*, pp. 1–18, 2020. 2.2.8, 4.1

[174] Y. Hou and L. Zheng, "Multiview Detection with Shadow Transformer (and View-coherent Data Augmentation)," in *ACM International Conference on Multimedia*, pp. 1673–1682, 2021. 2.2.8, 4.1

[175] Y. Liu, F. Zhang, Q. Zhang, S. Wang, Y. Wang, and Y. Yu, "Cross-View Correspondence Reasoning Based on Bipartite Graph Convolutional Network for Mammogram Mass Detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3812–3822, 2020. 2.2.8

[176] C.-Y. Tseng, Y.-R. Chen, H.-Y. Lee, T.-H. Wu, W.-C. Chen, and W. Hsu, "CrossDTR: Cross-view and Depth-guided Transformers for 3D Object Detection," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4850–4857, 2023. 2.2.8

[177] Y. Wang, V. C. Guizilini, T. Zhang, Y. Wang, H. Zhao, and J. Solomon, "DERTR3D: 3D Object Detection from Multi-view Images Via 3D-to-2D Queries," in *Conference on Robot Learning (CoRL)*, pp. 180–191, 2022. 2.2.8

[178] Y. Yao, Y. Jafarian, and H. S. Park, "Monet: Multiview Semi-supervised Keypoint Detection Via Epipolar Divergence," in *IEEE International Conference on Computer Vision (ICCV)*, pp. 753–762, 2019. 2.2.8

[179] Y. He, R. Yan, K. Fragkiadaki, and S.-I. Yu, "Epipolar Transformers," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7779–7788, 2020. 2.2.8, 6.2.3

[180] R. Marroquin, J. Dubois, and C. Nicolle, "WiseNET: An Indoor Multi-camera Multi-space Dataset with Contextual Information and Annotations for People Detection and Tracking," *Data in Brief*, vol. 27, p. 104654, 2019. 2.2.8, 2.2

[181] N. Rai, H. Chen, J. Ji, R. Desai, K. Kozuka, S. Ishizaka, E. Adeli, and J. C. Niebles, "Home Action Genome: Cooperative Compositional Action Understanding," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11184–11193, 2021. 2.2.8, 2.2

[182] J. Ma, J. Tong, S. Wang, W. Zhao, L. Zheng, and C. Nguyen, "Voxelized 3D Feature Aggregation for Multiview Detection," *arXiv preprint arXiv:2112.03471*, 2021. 2.2.8, 2.2

[183] R. Mao, J. Guo, Y. Jia, Y. Sun, S. Zhou, and Z. Niu, "DOLPHINS: Dataset for Collaborative Perception enabled Harmonious and Interconnected Self-driving," in *Asian Conference on Computer Vision (ACCV)*, pp. 4361–4377, 2022. 2.2.8, 2.2

[184] D. Mery, “Automated Detection in Complex Objects Using a Tracking Algorithm in Multiple X-ray Views,” in *Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 41–48, 2011. 2.2.8

[185] N. Bhowmik, W. Q., Y. F. A. Gaus, M. Szarek, and T. P. Breckon, “The Good, the Bad and the Ugly: Evaluating Convolutional Neural Networks for Prohibited Item Detection Using Real and Synthetically Composite X-ray Imagery,” in *British Machine Vision Conference (BMVC) Workshops*, pp. 1–8, 2019. 2.2.8

[186] D. Mery, V. Riffo, I. Zuccar, and C. Pieringer, “Automated X-ray Object Recognition Using an Efficient Search Algorithm in Multiple Views,” in *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 368–374, 2013. 2.2.8

[187] V. Riffo, S. Flores, and D. Mery, “Threat Objects Detection in X-ray Images Using an Active Vision Approach,” *Journal of Nondestructive Evaluation*, vol. 36, p. 44, 2017. 2.2.8, 3.2.2, 3.2.3, 3.4.3

[188] S. Akcay, M. E. Kundegorski, C. G. Willcocks, and T. P. Breckon, “On Using Deep Convolutional Neural Network Architectures for Automated Object Detection and Classification within X-ray Baggage Security Imagery,” *IEEE Transactions on Information Forensics & Security*, vol. 13, pp. 2203–2215, 2018. 2.2.8, 3.1, 3.1, 3.3.2, B.1

[189] S. Akcay and T. Breckon, “Towards Automatic Threat Detection: A Survey of Advances of Deep Learning within X-ray Security Imaging,” *Pattern Recognition*, vol. 122, p. 108245, 2020. 2.2.8, 3.1, A.1, B.1, B.1

[190] M. Bastan, W. Byeon, and T. M. Breuel, “Object Recognition in Multi-view Dual Energy X-ray Images,” in *British Machine Vision Conference (BMVC)*, vol. 1, pp. 1045–1060, 2013. 2.2.8

[191] Faraz, R. S. S. J.-M. O., and Saeedan, “Multi-view X-ray R-CNN,” in *Pattern Recognition*, pp. 153–168, 2019. 2.2.8, 3.1, 4.1

[192] H. Shum and S. B. Kang, “Review of Image-based Rendering Techniques,” in *Visual Communications and Image Processing*, vol. 4067, pp. 2–13, 2000. 2.3.1

[193] P. P. Srinivasan, R. Tucker, J. T. Barron, R. Ramamoorthi, R. Ng, and N. Snavely, “Pushing the Boundaries of View Extrapolation with Multiplane Images,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 175–184, 2019. 2.3.1

[194] V. Sitzmann, J. Thies, F. Heide, M. Nießner, G. Wetzstein, and M. Zollhofer, “Deepvoxels: Learning Persistent 3D Feature Embeddings,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2437–2446, 2019. 2.3.1

[195] B. Mildenhall, P. P. Srinivasan, R. Ortiz-Cayon, N. K. Kalantari, R. Ramamoorthi, R. Ng, and A. Kar, "Local Light Field Fusion: Practical View Synthesis with Prescriptive Sampling Guidelines," *ACM Transactions on Graphics (ToG)*, vol. 38, pp. 1–14, 2019. 2.3.1

[196] P. Hedman, J. Philip, T. Price, J.-M. Frahm, G. Drettakis, and G. Brostow, "Deep Blending for Free-viewpoint Image-based Rendering," *ACM Transactions on Graphics (ToG)*, vol. 37, pp. 257:1–257:15, 2018. 2.3.1

[197] N. Rahaman, A. Baratin, D. Arpit, F. Draxler, M. Lin, F. Hamprecht, Y. Bengio, and A. Courville, "On the Spectral Bias of Neural Networks," in *International Conference on Machine Learning (ICML)*, pp. 5301–5310, 2019. 2.3.2, 5.1

[198] K. Park, U. Sinha, P. Hedman, J. T. Barron, S. Bouaziz, D. B. Goldman, R. Martin-Brualla, and S. M. Seitz, "HyperNeRF: A Higher-dimensional Representation for Topologically Varying Neural Radiance Fields," *ACM Transactions on Graphics (ToG)*, vol. 40, pp. 1–12, 2021. 2.3.2, 2.3.3

[199] B. Mildenhall, P. Hedman, R. Martin-Brualla, P. P. Srinivasan, and J. T. Barron, "NeRF in the Dark: High Dynamic Range View Synthesis From Noisy Raw Images," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16190–16199, 2022. 2.3.2, 2.3.3

[200] A. Jain, B. Mildenhall, J. T. Barron, P. Abbeel, and B. Poole, "Zero-Shot Text-Guided Object Generation With Dream Fields," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 867–876, 2022. 2.3.2

[201] D. Verbin, P. Hedman, B. Mildenhall, T. Zickler, J. T. Barron, and P. P. Srinivasan, "Ref-NeRF: Structured View-dependent Appearance for Neural Radiance Fields," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5481–5490, 2022. 2.3.3

[202] A. Yu, V. Ye, M. Tancik, and A. Kanazawa, "pixelNeRF: Neural Radiance Fields From One or Few Images," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4578–4587, 2021. 2.3.3

[203] J. Zhang, Y. Zhang, H. Fu, X. Zhou, B. Cai, J. Huang, R. Jia, B. Zhao, and X. Tang, "Ray Priors Through Reprojection: Improving Neural Radiance Fields for Novel View Extrapolation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18376–18386, 2022. 2.3.3

[204] T. Chen, P. Wang, Z. Fan, and Z. Wang, "Aug-NeRF: Training Stronger Neural Radiance Fields With Triple-Level Physically-Grounded Augmentations," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15191–15202, 2022. 2.3.3

[205] C.-H. Lin, W.-C. Ma, A. Torralba, and S. Lucey, "BARF: Bundle-Adjusting Neural Radiance Fields," in *IEEE International Conference on Computer Vision (ICCV)*, pp. 5741–5751, 2021. 2.3.3

[206] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant Neural Graphics Primitives with a Multiresolution Hash Encoding," *ACM Transactions on Graphics (ToG)*, vol. 41, pp. 102:1–102:15, 2022. 2.3.3, 5.1, 6.2.4

[207] S. Fridovich-Keil, A. Yu, M. Tancik, Q. Chen, B. Recht, and A. Kanazawa, "Plenoxels: Radiance Fields without Neural Networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5501–5510, 2022. 2.3.3

[208] Q. Xu, Z. Xu, J. Philip, S. Bi, Z. Shu, K. Sunkavalli, and U. Neumann, "Point-NeRF: Point-based Neural Radiance Fields," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5438–5448, 2022. 2.3.3

[209] H. Turki, D. Ramanan, and M. Satyanarayanan, "Mega-NERF: Scalable Construction of Large-Scale NeRFs for Virtual Fly-Throughs," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12922–12931, 2022. 2.3.4

[210] H. Li, H. Chen, W. Jing, Y. Li, and R. Zheng, "3D Ultrasound Spine Imaging with Application of Neural Radiance Field Method," in *IEEE International Ultrasonics Symposium (IUS)*, pp. 1–4, 2021. 2.3.4

[211] J. Sun, Y. Xu, M. Ding, H. Yi, J. Wang, L. Zhang, and M. Schwager, "NeRF-Loc: Transformer-Based Object Localization Within Neural Radiance Fields," *IEEE Robotics and Automation Letters*, vol. 8, pp. 5244 – 5250, 2022. 2.3.4

[212] B. Hu, J. Huang, Y. Liu, Y.-W. Tai, and C.-K. Tang, "NeRF-RPN: A General Framework for Object Detection in NeRFs," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 23528–23538, 2023. 2.3.4, 6.2.5

[213] Y. Guo, K. Chen, S. Liang, Y. Liu, H. Bao, and J. Zhang, "AD-NeRF: Audio Driven Neural Radiance Fields for Talking Head Synthesis," in *IEEE International Conference on Computer Vision (ICCV)*, pp. 5784–5794, 2021. 2.3.4

[214] V. Rudnev, M. Elgharib, W. Smith, L. Liu, V. Golyanik, and C. Theobalt, "NeRF for Outdoor Scene Relighting," in *European Conference on Computer Vision (ECCV)*, 2022. 2.3.4

[215] P. P. Srinivasan, B. Deng, X. Zhang, M. Tancik, B. Mildenhall, and J. T. Barron, "NeRV: Neural Reflectance and Visibility Fields for Relighting and View Synthesis," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7491–7500, 2021. 2.3.4

[216] Y.-J. Yuan, Y.-T. Sun, Y.-K. Lai, Y. Ma, R. Jia, and L. Gao, "NeRF-editing: Geometry Editing of Neural Radiance Fields," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18353–18364, 2022. 2.3.4

[217] I. A. T. Association, "IATA Forecast Predicts 8.2 billion Air Travelers in 2037," 2018. 3.1

[218] O. E. Wetter, "Imaging in Airport Security: Past, Present, Future, and the Link to Forensic and Clinical Radiology," *Journal of Forensic Radiology and Imaging*, vol. 1, pp. 152 – 160, 2013. 3.1

[219] S. Akcay, M. E. Kundegorski, M. Devereux, and T. P. Breckon, "Transfer Learning Using Convolutional Neural Networks for Object Classification within X-ray Baggage Security Imagery," in *International Conference on Imafe Processing (ICIP)*, pp. 1057 –1061, IEEE, 2016. 3.1, B.1

[220] S. Akcay and T. P. Breckon, "An Evaluation of Region Based Object Detection Strategies within X-ray Baggage Security Imagery," in *IEEE International Conference on Image Processing (ICIP)*, pp. 1337–1341, IEEE, 2017. 3.1

[221] K. J. Liang, J. B. Sigman, G. P. Spell, D. Strellis, W. Chang, F. Liu, T. Mehta, and L. Carin, "Toward Automatic Threat Recognition for Airport X-ray Baggage Screening with Deep Convolutional Object Detection," *arXiv preprint arXiv:1912.06329*, 2019. 3.1

[222] L. Wang, Z. Liu, and Z. Zhang, "Efficient Image Features Selection and Weighting for Fundamental Matrix Estimation," *IET Computer Vision*, vol. 10, pp. 67–78, 2016. 3.1

[223] M. Klüppel, J. Wang, D. Bernecker, P. W. Fischer, and J. Hornegger, "On Feature Tracking in X-ray Images," in *Bildverarbeitung für die Medizin1*, pp. 132–137, 2014. 3.1, 3.1, 3.2.1, 6.1

[224] K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Xu, Z. Zhang, D. Cheng, C. Zhu, T. Cheng, Q. Zhao, B. Li, X. Lu, R. Zhu, Y. Wu, J. Dai, J. W. Jingdong, Shi, W. Ouyang, C. C. Loy, and D. Lin, "MMDetection: Open MMLab Detection Toolbox and Benchmark," *arXiv preprint arXiv:1906.07155*, 2019. 3.3.2, 4.3.2, B.2.4

[225] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable Convolutional Networks," in *IEEE International Conference on Computer Vision (ICCV)*, pp. 764–773, 2017. 4, B.1, B.2.2, B.1

[226] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer Normalization," *arXiv preprint arXiv:1607.06450*, 2016. 4.2

[227] I. Loshchilov and F. Hutter, "Decoupled Weight Decay Regularization," in *International Conference on Learning Representations (ICLR)*, 2018. 4.3.2, B.1

[228] H. Liao, B. Huang, and H. Gao, "Feature-Aware Prohibited Items Detection for X-ray Images," in *IEEE International Conference on Image Processing (ICIP)*, pp. 1040–1044, 2023. 4.4

[229] J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. Hedman, "Zip-NeRF: Anti-Aliased Grid-Based Neural Radiance Fields," in *IEEE International Conference on Computer Vision (ICCV)*, 2023. 5.1, 6.2.4

[230] B. Deng, J. T. Barron, and P. P. Srinivasan, "JaxNeRF: An Efficient JAX Implementation of NeRF," *GitHub repository*, 2020. 5.3

[231] B. Mildenhall, D. Verbin, P. P. Srinivasan, P. Hedman, R. Martin-Brualla, and J. T. Barron, "MultiNeRF: A Code Release for Mip-NeRF 360, Ref-NeRF, and RawNeRF." `https://github.com/google-research/multinerf`, 2022. 5.3

[232] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The Unreasonable Effectiveness of Deep Features as a Perceptual Metric," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 586–595, 2018. 5.4, 5.7

[233] K. Ding, K. Ma, S. Wang, and E. P. Simoncelli, "Image Quality Assessment: Unifying Structure and Texture Similarity," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, pp. 2567 – 2581, 2020. 5.4, 5.7, A.5.2

[234] O. Faugeras and B. Mourrain, "On the Geometry and Algebra of the Point and Line Correspondences Between N Images," in *IEEE International Conference on Computer Vision (ICCV)*, pp. 951–956, 1995. 6.2.2

[235] Y. Ma, K. Huang, R. Vidal, J. Košecká, and S. Sastry, "Rank Conditions on the Multiple-view Matrix," *International Journal of Computer Vision*, vol. 59, pp. 115–137, 2004. 6.2.2

[236] C. Rubino, M. Crocco, and A. D. Bue, "3D Object Localisation from Multi-View Image Detections," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, pp. 1281–1294, 2018. 6.2.5

APPENDIX A

Cross-modal Image Synthesis within Dual-Energy X-ray Security Imagery

Dual-energy X-ray scanners are used for aviation security screening given their capability to discriminate materials inside passenger baggage. To facilitate manual operator inspection, a pseudo-colouring is assigned to the effective composition of the material. Recently, paired image-to-image translation models based on conditional Generative Adversarial Networks (cGAN) have been shown to be effective for image colourisation. In this appendix, we investigate the use of such a model to translate from the raw X-ray energy responses (high, low, effective-Z) to the pseudo-coloured images and vice versa. Specifically, given N X-ray modalities, we train a cGAN conditioned in $N - m$ domains to generate the remaining m representation. Our method achieves a mean squared error (MSE) of 16.5 and a structural similarity index (SSIM) of 0.9815 when using the raw modalities to generate the pseudo-colour representation. Additionally, raw X-ray high energy, low energy and effective-Z projections were generated given the pseudo-colour image with minimum MSE of 2.57, 5.63 and 1.43, and maximum SSIM of 0.9953, 0.9901 and 0.9921. Furthermore, we assess the quality of our synthesised pseudo-colour reconstructions by measuring the performance of two object detection models originally trained on real X-ray pseudo-colour images over our generated pseudo-colour images. Interestingly, our generated pseudo-colour images obtain marginally improved detection performance than the corresponding real X-ray pseudo-colour images, showing that meaningful representations are synthesized and that these reconstructions are applicable for differing aviation security tasks.

A.1 Introduction

Identification of material composition plays an important role in baggage security screening as it facilitates the material-based detection of prohibited items [A3, A4]. A material can be characterized by a mass attenuation coefficient which describes how beams at different energy levels are able to penetrate the material. In this sense, multiple-energy X-rays can be used to identify the composition of a scanned object. Particularly, dual-energy X-ray imaging has shown to be an effective technique for this task [A5]. The effective atomic number, effective-Z or Z_{eff}, can be approximated given two different energy projections between 20 and 200 keV [A6]. Subsequently, a look-up table is usually used to assign a material profile and hence corresponding pseudo-colour/RGB to a value of Z_{eff}, identifying between organic (orange), metallic (blue) and inorganic (green) [A7]. An example of such X-ray sub-modalities is shown in Fig. A.1 where the high and low energy images can be further processed, via the use of effective-Z, to create a corresponding pseudo-coloured image [A8].

The task of assigning an RGB colour to a greyscale (intensity) value is known as image colourisation. It is an ill-posed problem since the mapping from colour to greyscale $f : \mathbb{R}^3 \rightarrow \mathbb{R}$ is not injective; *i.e.* different RGB values may have the same grey value. It has been shown that deep neural networks have good performance for image colourisation [A9]. When paired data is available, a popular supervised architecture for this task is the *pix2pix* architecture proposed by Isola *et al.* [A10]. They use a conditional Generative Adversarial Network (cGAN) to generate an image in a different domain than the input image. In this sense, image colourisation is an image-to-image translation task where the greyscale and the coloured representations of the images are considered to belong to different domains. Since high and

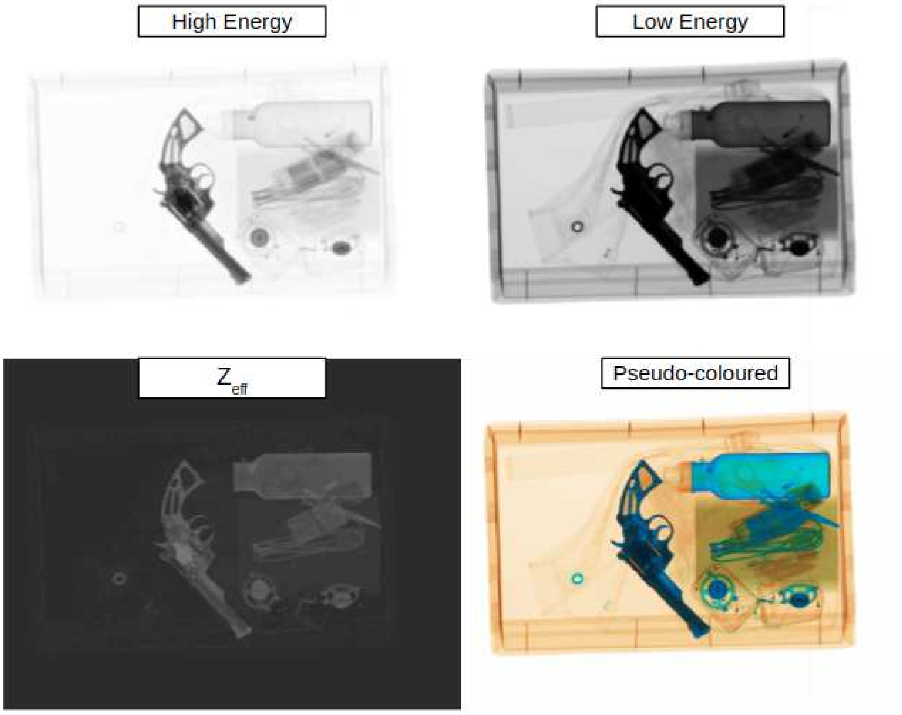


Figure A.1: Exemplar multi-modal X-ray screening imagery.

low energy responses can be seen as greyscale intensity images, cGAN can be used to translate between energy and coloured images.

Image pseudo-colourisation of dual-energy raw projections has been performed in recent years to aid the visual inspection of security imagery. However, recent works focusing on the automatic detection of threat items [189] have brought the question as to whether the raw energy images encode additional information that can be used for this purpose. Bhowmik *et al.* [A11] used the raw responses to train different object detection algorithms. They found that the energy responses can be used independently to detect objects of interest, but the best results are obtained when detectors are trained using the pseudo-coloured images, the energy responses and the Z_{eff} mapping in conjunction. Furthermore, they demonstrate that such models are transferable across differing X-ray scanners [A11]. Although several large-scale X-ray baggage imagery datasets exist [116, 119] [A12], raw X-ray projections are not usually provided as it is not archived by default in standard operational use.

In this context, this appendix investigates both the generation of pseudo-colour images from dual-energy X-ray security raw modalities (high energy, low energy and Z_{eff}) and the decomposition of these energy images from pseudo-coloured images. Our contributions are as follows:

- use of a GAN-based image to image translation architecture [A10] applied to the context of dual-energy X-ray security imagery for the generation of high energy, low energy and Z_{eff} modalities from pseudo-colour X-ray imagery and vice versa.
- the proposed use of two GAN generators for cross-modality synthesis with multiple paired input and output variants, namely, via input concatenation and Siamese network output for each input modality. Maximal quality is obtained with the Siamese version of the generator, with a mean squared error of 16.5 and a structural similarity index measure of 0.9815 for the generation of pseudo-coloured images from the raw X-ray energy modalities.
- assessment of the performance of two object detection models trained on real X-ray imagery when tested on the GAN-generated images. Interestingly, the performance of the generated pseudo-colour images outperforms the real X-ray images, showing that meaningful representations are learned with applications in downstream aviation security tasks.

A.2 Related Work

Earlier image colourisation techniques based on deep learning used plain convolutional neural networks in a supervised fashion [A13,A14]. Isola *et al.* [A10] proposed the *pix2pix* architecture which uses a cGAN for general paired image-to-image translation tasks. It is demonstrated that cGANs can be used for image colourisation, where the original image and its greyscale version are considered paired samples. A tailored version of *pix2pix* for image colourisation is explored by Nazeri *et al.* [A15]. Image colourisation has also been used to translate from a single-valued domain, such as infrared [A16] and radar [A17], to a coloured domain. For a comprehensive review on image colourisation, see Anwar *et al.* [A9].

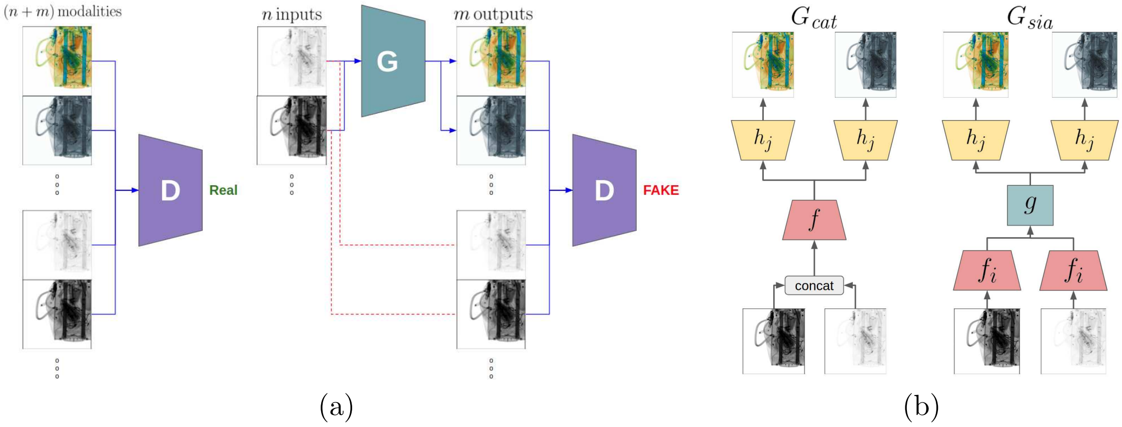


Figure A.2: Cross-modal image translation of dual-energy X-ray imagery with a cGAN. (a) Modified *pix2pix* architecture to account for multiple input and outputs. (b) Two generators are proposed: G_{cat} concatenates channel-wise the inputs while G_{sia} has a sub-network for each input modality. Both generators implement different output networks for each output modality.

Colourisation and enhancement of dual-energy X-ray imagery have been investigated in order to improve the detection of threat items [A18, A19]. However, to the best of our knowledge, this is the first work that aims to reconstruct the pseudo-colouring image from the raw X-ray projections and to recover the energy responses from the pseudo-colour image.

A.3 Dual-energy X-ray Imaging

X-ray images are formed by measuring the transmitted irradiance I of a beam with energy E through a material with thickness T and atomic number Z. This resulting irradiance I is given by the Beer's law:

$$I = I_0 e^{-\mu(E,Z)T}\,, \tag{A.1}$$

where μ is the attenuation coefficient which depends on the material and the energy of the beam. It is noted from Eq. (A.1) that the transmitted irradiance I is always less or equal to I_0, meaning that thicker objects appear darker in the resulting image, as seen in Fig. A.1. Since I_0 and I are known, we can obtain the expression:

$$\mu(E,Z)T = \ln\left(\frac{I}{I_0}\right)\,. \tag{A.2}$$

For energies less than 200 keV, μ can be decomposed into the attenuation coefficients μ_p and μ_c dominated by the photoelectric and the Compton scattering effects [A6], *i.e.*,

$$\mu(E,Z) = \mu_p(E,Z) + \mu_c(E,Z)\,. \tag{A.3}$$

Alvarez and Macovski [A20] empirically found that:

$$\mu_p(E,Z) \approx \frac{1}{E^3} K_p \frac{\rho}{A} Z^m \tag{A.4}$$

$$\mu_z(E,Z) \approx f_{KN}(E) K_c \frac{\rho}{A} Z\,, \tag{A.5}$$

where f_{KN} is the Klein-Nishina function, A is the atomic weight and K_p, K_c and m are constants. An approximation of the atomic number Z can be obtained by using low and high energies E_l and E_h where the response I is dominated by μ_p and μ_c, respectively. Since the response of both energies are measured with respect to the same object, and thus the same thickness, the ratio $\mu_p(E_l,Z)/\mu_c(E_h,Z)$ can be calculated using Eq. (A.2). From Eqs. (A.4) and (A.5), we can express this ratio as:

$$\frac{\mu_p}{\mu_c} \approx \frac{1}{E_l^3 f_{KN}(E_h)} Z^{m-1}\,. \tag{A.6}$$

The atomic number Z is then approximated by:

$$Z \approx K\left(\frac{\mu_p}{\mu_c}\right)^{\frac{1}{n}}\,, \tag{A.7}$$

where K is a value depending on the high and low energies and $n = m - 1$ is a constant. Finally, the thickness of a material can be obtained from Eq. (A.2).

Since an X-ray beam may penetrate different objects, instead of calculating the Z for each of them, we simplify our analysis by considering that the beam went through a homogeneous material. The resulting atomic number of this hypothetical material is known as the effective atomic number, Z_{eff}. Dual-energy pseudo-coloured images are coloured by assigning a colour depending on the Z_{eff} and the thickness, Eq. (A.2) [A6].

A.4 Methodology

In this appendix, we utilise an approach based on the *pix2pix* architecture, modified to account for multiple input and output images, for cross-modality translation of dual-energy X-ray imagery.

A.4.1 Problem Formulation

The *pix2pix* architecture is a cGAN consisting on a generator $G : \mathbb{R}^{C_{in} \times H \times W} \to \mathbb{R}^{C_{out} \times H \times W}$ that maps an image with C_{in} input channels and $H \times W$ spatial size from a domain to another domain with the same spatial size and C_{out} output channels, and a discriminator $D : \mathbb{R}^{(C_{in}+C_{out}) \times H \times W} \to (0, 1)$ that classifies if the image from the target domain is real or fake given the image from the source domain. Given two paired images $\{x_A, x_B\}$ from domains A and B, *pix2pix* uses the adversarial loss function:

$$\begin{aligned} \mathcal{L}_{cGAN}(G, D) = & \mathbb{E}_{x_A, x_B} \left[\log D(x_A, x_B)\right] + \\ & \mathbb{E}_{x_A} \left[\log \left(1 - D(x_A, G(x_A))\right)\right] . \end{aligned} \tag{A.8}$$

and additionally, an $\mathcal{L}_{L1}$ reconstruction loss is added as the final image reconstruction objective:

$$G^* = \arg\min_G \max_D \mathcal{L}_{cGAN}(G, D) + \lambda \mathcal{L}_{L1}(G) . \tag{A.9}$$

Conventionally, pseudo-coloured X-ray images (e.g. Fig. A.1) are formed from the effective atomic number Z_{eff} and the material thickness/density, which is obtained from the high and low energy responses (see Appendix A.3). Consequently, we extend the *pix2pix* architecture to accept multiple input and output images in order to allow us to work across the joint set of $\{$*pseudo-colour, high, low,* $Z_{\text{eff}}\}$ X-ray modalities (as shown in Fig. A.1).

A.4.2 Proposed Variants

Our proposed extended architecture is shown in Fig. A.2a. Given n paired inputs $\mathbf{x} = \{x_1, \ldots, x_n\}$ with $\{u_1, \ldots, u_n\}$ channels and m paired outputs $\mathbf{y} = \{y_1, \ldots, y_m\}$ with $\{v_1, \ldots, v_m\}$ channels, we define a multi-domain generator $G : \mathbb{R}^{\sum_i^n u_i \times H \times W} \rightarrow \mathbb{R}^{\sum_i^m v_i \times H \times W}$. Two methods of combining multiple domains are explored in this appendix: via channel concatenation and via a Siamese network sub-architecture. In the former, the generator G_{cat} takes the input images concatenated channel-wise as a single input for a network f, while in the latter, the generator G_{sia} process each input x_i in a sub-network f_i, where the resulting representations are concatenated channel-wise and combined fed into a common network g. Each domain output y_j is generated from a common feature representation of the input images using a different network h_j for each output modality. A diagram with these approaches is shown in Fig. A.2b. The generators G_{cat} and G_{sia} define the generation processes:

$$\begin{aligned}(\mathbf{y}_{\text{cat}})_j &= G_{\text{cat}}(\mathbf{x})_j \\ &= (h_j \circ f)\left([x_1, \ldots, x_n]\right)\end{aligned} \tag{A.10}$$

and:

$$\begin{aligned}(\mathbf{y}_{\text{sia}})_j &= G_{\text{sia}}(\mathbf{x})_j \\ &= (h_j \circ g)\left([f_1(x_1), \ldots, f_n(x_n)]\right) \ ,\end{aligned} \tag{A.11}$$

where $[\ldots]$ means concatenation. Similarly to the discriminator in the *pix2pix* architecture, our multi-domain discriminator $D : \mathbb{R}^{(\sum_i u_i + \sum_j v_j) \times H \times W} \rightarrow (0, 1)$ takes all inputs and outputs to classify them as real or fake.

The multi-domain adversarial and reconstruction losses are then:

$$\begin{aligned}\mathcal{L}_{mcGAN}(G, D) = &\mathbb{E}_{\mathbf{x},\mathbf{y}}\left[\log D(\mathbf{x}, \mathbf{y})\right] + \\ &\mathbb{E}_{\mathbf{x}}\left[\log\left(1 - D(\mathbf{x}, G(\mathbf{x}))\right)\right]\end{aligned} \tag{A.12}$$

Network	Architecture
f_i	Conv 7×7 Conv 3×3, stride $= 2$ Conv 3×3, stride $= 2$ $L\times$ Residual
g	$M\times$ Residual
h_j	$N\times$ Residual Transp Conv 3×3, stride $= 2$ Transp Conv 3×3, stride $= 2$ Conv, 7×7

Table A.1: Architecture of the generator sub-networks.

and:

$$\mathcal{L}_{mL1}(G) = \sum_{j}^{m} \mathbb{E}_{\mathbf{x},y_j}\left[\|y_j - G(\mathbf{x})_j\|_1\right]. \tag{A.13}$$

Furthermore, Jiang *et al.* [A21] introduced the frequency focal loss (FFL), which aims to reduce the gap in the frequency response of the synthesized images. We investigate if our multi-domain image translation can be improved using the FFL. Finally, our objective function is then:

$$\begin{aligned} G^* = \arg\min_G \max_D \mathcal{L}_{mcGAN}(G, D) + \\ \lambda_{mL1}\mathcal{L}_{mL1}(G) + \lambda_{\text{FFL}}\mathcal{L}_{\text{FFL}}(G). \end{aligned} \tag{A.14}$$

A.4.3 Network Architecture

The original *pix2pix* model uses a UNet [A22] with skip connections as the generator. However, following the approach of CycleGAN [A23], we implement the architecture described by Johnson *et al.* [69]. This network consists on three convolutional layers, a series of stacked residual blocks, two transposed convolutional layers and an output convolutional layer. Following this architecture, Table A.1 describes the f_i, g and h_j networks used for the generators in Eqs. (A.10) and (A.11). The f_i networks consist on three convolutional layers and L residual blocks, the g network is composed of M residual blocks and the h_j networks have N residual blocks, two transposed

Experimental label	Reconstruction Type	Description
$\{h, l, z\} \rightarrow rgb$	one-to-one	High energy, low energy or Z_{eff} to pseudo colour.
$hl_{\text{sia}} \rightarrow rgb$	many-to-one	High and low energy to pseudo colour (G_{sia}).
$hlz_{\text{sia}} \rightarrow rgb$	many-to-one	High energy, low energy and Z_{eff} to pseudo colour (G_{sia}).
$hlz_{\text{cat}} \rightarrow rgb$	many-to-one	High energy, low energy and Z_{eff} to pseudo colour (G_{cat}).
$rgb \rightarrow \{h, l, z\}$	one-to-one	Pseudo colour to high energy, low energy or Z_{eff}
$rgb \rightarrow hlz$	one-to-many	Pseudo colour to high energy, low energy and Z_{eff}

Table A.2: Experimental labels and descriptions for the experiments carried out in this work.

convolutions and a final convolutional layer. All layers use instance normalisation [A24] and ReLU activation except for the last convolutional layer in h_j, that does not use normalisation and has a Tanh function as activation. The network f in Eq. (A.10) is defined as $f = g \circ f_i$. In this appendix, we have three different cases of cross-modal synthesis: one-to-one mode, multi-to-one mode and one-to-multi modes. For one-to-one and one-to-many task we use $L = 4$, $M = 5$ and $N = 0$ while for many-to-one we use $L = M = N = 3$. Finally, the discriminator follows the PatchGAN network used by Isola *et al.* [A10].

A.5 Evaluation

We evaluate our multi-modal cross-modal translation architecture for pseudo-coloured and raw X-ray energy response images (as shown in Fig. A.1). We use the labels rgb, h, l and z for the pseudo-colour, high energy, low energy and Z_{eff} imagery subsets, respectively. The experiments performed in this appendix are described in Table A.2.

A.5.1 Dataset

We train our models in the *deei6* dataset [A11]. This dataset consists on 7,022 quadruplets (h, l, z and rgb) of bags scanned in a dual-energy Gilardoni FEP ME 640 AMX scanner [A25] (see Fig. A.1). Bounding box and instance mask annotations are given for six classes: bottle, hairdryer, iron, toaster, phone-tablet and laptop. The dataset is split in 4,909 quadruplets for training and 2,113 for testing.

A.5.2 Performance Metrics

Two image quality metrics are used in this appendix: mean squared error (MSE) and the structural similarity index measure (SSIM) [233]. Additionally, two detection networks, CARAFE [A2] and Cascade Mask RCNN [157], are trained on the real X-ray image datasets using the same settings as Bhowmik *et al*. [A11] and tested on the synthesized images generated from the same X-ray dataset under the experimental conditions set out in Table A.2. We report instance segmentation results using the MS COCO mean Average Precision (mAP) performance metric [88] (intersection over union of 0.50:.05:0.95), using Average Precision (AP) for class-wise and mAP for overall performance.

A.5.3 Implementation Details

Input images are resized to 600 × 600 pixels and random cropped to have a final size of 512 × 512. Differently from *pix2pix*, we do not use dropout. The model is trained using Adam optimization [65] with a learning rate of 2×10^{-4} for 100 epochs, linearly decaying to 0 for another 100 epochs. We choose $\lambda_{L1} = 100$ for the objective function defined in Eq. (A.14) and $\lambda_{\text{FFL}} = 10$ when the FFL is used. A batch size of 6 n-tuples of image modalities is used to train our models.

A.6 Results

In this section we review the results for image synthesis quality and detection performance. We evaluate the G_{sia} and G_{cat} generators and the impact of the FFL during training.

A.6.1 Reconstruction Quality

Cross-modality image synthesis performance is shown in Table A.3. MSE and SSIM metrics are reported, comparing the synthesis quality with the real images. The impact of using the FFL is also reported. It can be observed that in general, the best reconstructions are obtained when using the focal frequency loss, although the

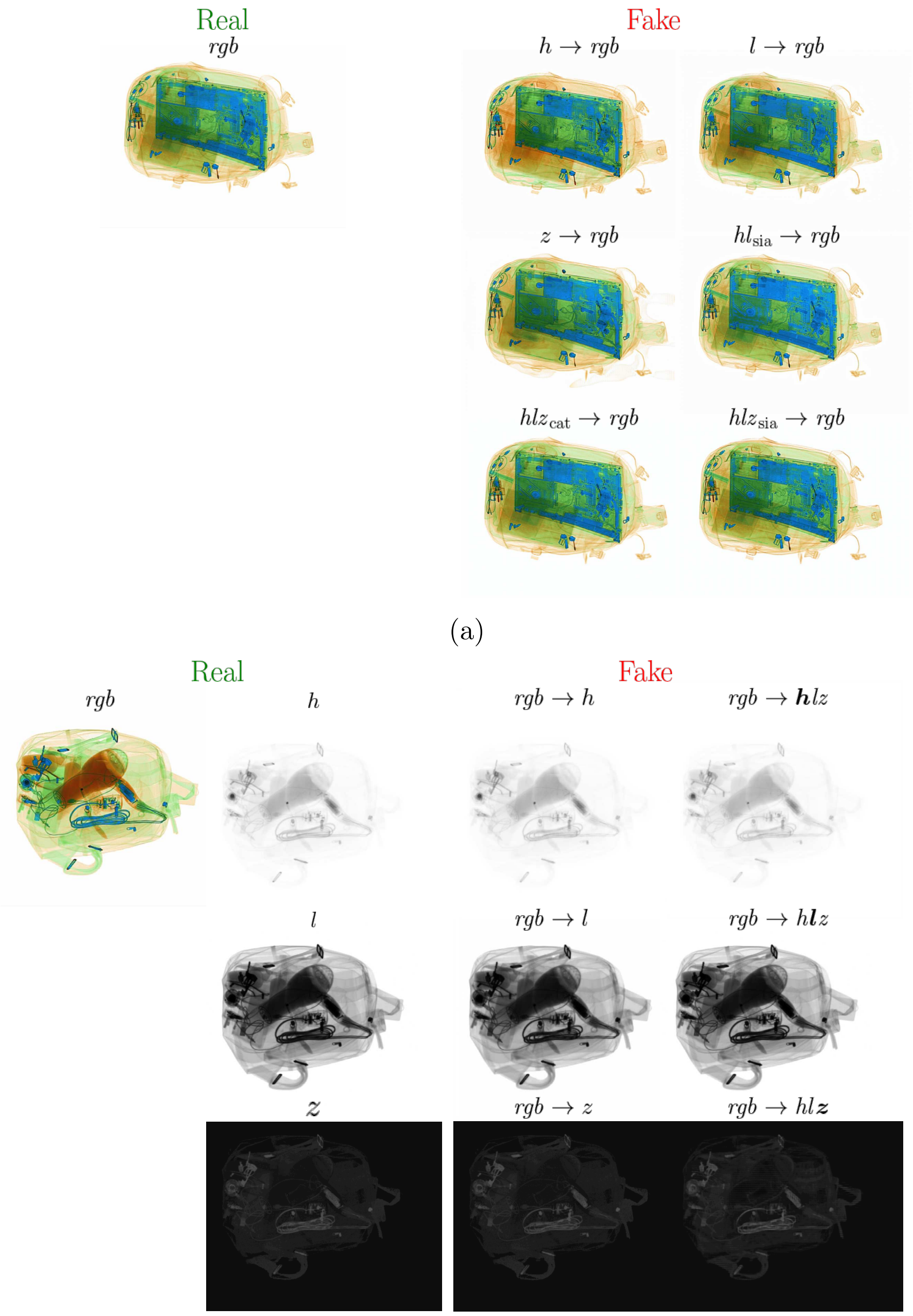
Model	w/o FFL		w/ FFL	
	MSE ↓	SSIM ↑	MSE ↓	SSIM ↑
$h \rightarrow rgb$	182.3	0.9229	185.6	0.9216
$l \rightarrow rgb$	183.6	0.9296	168.4	0.9297
$z \rightarrow rgb$	125.0	0.9041	121.5	0.9049
$hl_{\text{sia}} \rightarrow rgb$	465.1	0.9411	101.6	0.9600
$hlz_{\text{cat}} \rightarrow rgb$	20.1	0.9753	18.9	0.9766
$hlz_{\text{sia}} \rightarrow rgb$	**16.5**	**0.9815**	17.3	0.9808
$rgb \rightarrow h$	2.57	**0.9953**	**2.28**	0.9948
$rgb \rightarrow \boldsymbol{h}lz$	18.3	0.9910	12.3	0.9915
$rgb \rightarrow l$	5.63	**0.9901**	**5.43**	0.9888
$rgb \rightarrow h\boldsymbol{l}z$	7.55	0.9847	6.13	0.9885
$rgb \rightarrow z$	38.0	0.9823	4.61	0.9830
$rgb \rightarrow hl\boldsymbol{z}$	1.43	0.9794	**0.76**	**0.9921**

Table A.3: Cross-modality reconstruction performance.

improvement is minor and does not always lead to the best results.

The best pseudo-coloured reconstructions are obtained by using the three modalities h, l and z and the G_{sia} generator from Eq. (A.11), obtaining an MSE of 16.5 and SSIM of 0.9815. We also confirm that pseudo-coloured image reconstruction gets degraded when only using one energy level. Although the use of Z_{eff} individually significantly improves the MSE, the structural similarity gets worse because the thickness information is lost (Appendix A.3). Fig. A.3a shows an example of the pseudo-colour reconstructions. It can be seen that when only using the high or low energy images, the reconstructed image tends to get confused around the organic (orange) regions, getting materials mixed up. Although the material information can be matched better using only the Z_{eff} modality, the shape is not always obtained correctly (see for example the top right corner of the laptop). It is also observed that using more than just one modality creates very accurate reconstructions.

As seen in Table A.3, the energy modalities can be recovered with high SSIM from the pseudo-colour images. The best results for high and low energies are obtained when they are generated using separate models. This could be explained by earlier layers learning specific features that capture the effect from each energy modality. However, the Z_{eff} modality is better recovered when predicting the three raw modalities at the same time, meaning that the learned features guided from the other modalities help in the identification of the atomic number. Fig. A.3b shows


Real
rgb
Fake
$h \rightarrow rgb$
$l \rightarrow rgb$
$z \rightarrow rgb$
$hl_{\text{sia}} \rightarrow rgb$
$hlz_{\text{cat}} \rightarrow rgb$
$hlz_{\text{sia}} \rightarrow rgb$
Real
Fake
rgb
h
$rgb \rightarrow h$
$rgb \rightarrow \boldsymbol{h}lz$
l
$rgb \rightarrow l$
$rgb \rightarrow h\boldsymbol{l}z$
$\boldsymbol{z}$
$rgb \rightarrow z$
$rgb \rightarrow hl\boldsymbol{z}$

(a)

(b)

Figure A.3: Exemplar of cross-modality synthesis. (a) Raw modalities to pseudo-colour. (b) pseudo-colour to raw modalities.

an example of the high energy, low energy and Z_{eff} modalities synthesized from the pseudo-colour image. Some small blurring effects can be seen in the high and low energy generations for the $rgb \rightarrow hlz$ model. Nevertheless, it is seen that regardless the model, the generated images exhibit high fidelity.

Dataset		Bottle	Hairdryer	Iron	Toaster	P-tablet	Laptop	mAP
High Energy	Real	**0.641/0.628**	**0.640/0.657**	**0.675/0.689**	**0.787/0.793**	**0.516/0.533**	**0.771/0.776**	**0.672/0.679**
	$rgb \rightarrow h$	0.597/0.593	0.579/0.594	0.642/0.656	0.740/0.760	0.496/0.498	0.754/0.751	0.635/0.642
	$rgb \rightarrow h$ (FFL)	0.596/0.591	0.584/0.596	0.632/0.655	0.738/0.756	0.469/0.481	0.741/0.747	0.627/0.638
	$rgb \rightarrow \boldsymbol{h}lz$	0.578/0.571	0.548/0.552	0.613/0.638	0.728/0.745	0.476/0.469	0.733/0.744	0.613/0.620
	$rgb \rightarrow \boldsymbol{h}lz$ (FFL)	0.590/0.584	0.553/0.563	0.618/0.641	0.715/0.724	0.480/0.474	0.737/0.747	0.615/0.622
Low Energy	Real	**0.615/0.620**	**0.609/0.629**	**0.657/0.682**	0.751/**0.776**	**0.508/0.526**	**0.760/0.765**	**0.650/0.666**
	$rgb \rightarrow l$	0.585/0.606	0.552/0.569	0.626/0.649	0.747/0.762	0.498/0.490	0.731/0.743	0.623/0.637
	$rgb \rightarrow l$ (FFL)	0.584/0.607	0.559/0.574	0.630/0.651	**0.759**/0.771	0.507/0.500	0.739/0.748	0.630/0.642
	$rgb \rightarrow h\boldsymbol{l}z$	0.559/0.563	0.524/0.545	0.605/0.637	0.740/0.751	0.471/0.462	0.706/0.716	0.601/0.612
	$rgb \rightarrow h\boldsymbol{l}z$ (FFL)	0.578/0.593	0.544/0.561	0.623/0.649	0.740/0.758	0.494/0.493	0.727/0.733	0.618/0.631
Z_{FFL}	Real	0.534/**0.548**	**0.460/0.490**	**0.606**/0.634	**0.783/0.793**	**0.490/0.488**	**0.718**/0.732	**0.598/0.614**
	$rgb \rightarrow z$	0.533/0.540	0.355/0.386	0.604/**0.635**	0.779/0.786	0.485/0.483	**0.718**/0.732	0.579/0.593
	$rgb \rightarrow z$ (FFL)	**0.535**/0.543	0.442/0.471	0.603/0.634	0.776/0.787	0.483/0.480	0.715/**0.736**	0.592/0.609
	$rgb \rightarrow hl\boldsymbol{z}$	0.472/0.494	0.290/0.304	0.544/0.560	0.745/0.756	0.403/0.395	0.642/0.666	0.516/0.529
	$rgb \rightarrow hl\boldsymbol{z}$ (FFL)	0.460/0.492	0.241/0.271	0.551/0.576	0.766/0.767	0.387/0.391	0.611/0.616	0.502/0.519
Pseudo Colour	Real	0.638/**0.635**	0.609/0.638	0.662/0.694	0.788/0.790	**0.536/0.552**	0.754/0.776	0.665/0.681
	$h \rightarrow rgb$	0.575/0.573	0.517/0.528	0.557/0.576	0.718/0.729	0.419/0.441	0.703/0.722	0.581/0.595
	$h \rightarrow rgb$ (FFL)	0.567/0.567	0.512/0.532	0.557/0.573	0.715/0.730	0.424/0.445	0.716/0.730	0.582/0.596
	$l \rightarrow rgb$	0.525/0.534	0.290/0.315	0.423/0.432	0.704/0.719	0.388/0.398	0.520/0.577	0.475/0.496
	$l \rightarrow rgb$ (FFL)	0.556/0.569	0.396/0.400	0.503/0.494	0.734/0.747	0.430/0.435	0.615/0.671	0.539/0.553
	$z \rightarrow rgb$	0.560/0.554	0.476/0.478	0.571/0.577	0.777/0.784	0.480/0.482	0.748/0.756	0.602/0.605
	$z \rightarrow rgb$ (FFL)	0.568/0.566	0.489/0.487	0.572/0.578	0.779/0.790	0.484/0.482	0.743/0.754	0.606/0.609
	$hl_{\mathrm{sia}} \rightarrow rgb$	0.513/0.514	0.454/0.456	0.583/0.539	0.726/0.732	0.420/0.425	0.478/0.476	0.529/0.524
	$hl_{\mathrm{sia}} \rightarrow rgb$ (FFL)	0.615/0.615	0.531/0.531	0.660/0.642	0.783/0.791	0.479/0.485	0.727/0.738	0.632/0.634
	$hlz_{\mathrm{cat}} \rightarrow rgb$	0.634/0.627	0.628/0.639	0.678/0.697	0.792/0.799	0.517/0.532	0.771/0.773	0.670/0.678
	$hlz_{\mathrm{cat}} \rightarrow rgb$ (FFL)	0.635/0.628	0.621/0.636	0.683/0.700	0.792/0.795	0.531/0.544	0.769/0.772	0.672/0.679
	$hlz_{\mathrm{sia}} \rightarrow rgb$	0.637/0.631	**0.637/0.649**	**0.688/0.704**	**0.793/0.802**	0.524/0.537	**0.773/0.777**	**0.675/0.683**
	$hlz_{\mathrm{sia}} \rightarrow rgb$ (FFL)	**0.641/0.635**	0.628/0.644	0.685/0.701	**0.793/0.802**	0.528/0.536	0.768/**0.777**	0.674/0.682

Table A.4: Object detection results using different modalities of X-ray imagery from the *deei6* dataset. The two reported values are for the CARAFE [A2] and Cascade Mask RCNN [157] architectures.

A.6.2 Detection Performance

Detection performance for real and synthesized images is presented in Table A.4. Results are for instance segmentation predictions. They are presented with two values, each corresponding to the CARAFE and Cascade Mask RCNN models. Per-class AP and total mAP results are shown.

Synthesized raw modalities show a better detection performance when they are generated with individual models, which is consistent with the quality of the reconstructions in Table A.3. Compared to the real images, the detection performance in the synthesized raw modalities gets reduced. This means that although the generated images may seem very similar, the reconstructions do not perfectly match the energy projections. It is worth noticing that while the generated Z_{eff} from the $rgb \rightarrow hlz$ shows a good SSIM, its detection performance is reduced significantly while compared to the original Z_{eff} response. This shows that detection models are very sensitive to small variations in the input images. On the other hand, the mAP of the generated pseudo-colour rgb images gets improved by a 1% for CARAFE detection model when using the three raw modalities and the G_{sia} generator. This slight improvement over the detection performance may indicate that our model is learning to generate pseudo-coloured images more effectively than the standard formulation in terms of information retention in the resulting pseudo-coloured visualisation. These results illustrate that our proposed approach can be used to learn meaning from representations across differing X-ray modalities such that they can be used to effectively train a secondary deep neural network for subsequent downstream tasks.

A.7 Conclusions

In this appendix, we investigate the use of a conditional generative adversarial network for image to image translation of dual-energy X-ray security imagery. We perform image colourisation from high energy, low energy and effective atomic number Z_{eff} modalities and vice versa. Two novel generator architectures are proposed for the combination of multiple modalities as inputs and outputs. The first generator, G_{sia}, takes each input into a sub-network and then concatenates the resulting features. Our second proposed generator, G_{cat}, concatenates channel-wise the input images and process it as a single image multi-channel input. In both cases, multiple outputs are generated by having a sub-network to generate each modality. The use of the focal frequency loss (FFL) is also investigated.

It is observed that the best results for image colourisation are obtained when using the three modalities (high energy, low energy and Z_{eff}) and the G_{cat} generator, achieving a SSIM of 0.9766. In general, the FFL improved image colourisation. The best results for the extraction of the high and low energy modalities are obtained when having a separate model for each, having SSIMs of 0.9953 and 0.9901 (without FFL). On the other hand, the Z_{eff} gets a better reconstruction when using a model that predicts the three raw modalities at the same time, achieving a SSIM of 0.9921. A qualitative assessment shows that the differences are barely noticeable and reconstruction exhibit a good similarity when compared to the original X-ray modality imagery.

Detection performance results were obtained for two different architectures trained on the real images and tested on the synthesized images. For the raw X-ray energy response imagery, performance is worse on the generated images and compared to the original imagery. However, the pseudo-coloured images generated using the three raw modalities and the G_{sia} generator show a better detection performance than that obtained for the real images. On this basis, we hypothesize that the model learnt for raw X-ray energy response to pseudo-colour image translation offers a superior mapping in terms of information retention than the original raw X-ray imagery.

Future work will investigate the use of modern architectures for higher definition image-to-image translation and the transferability of these models to images obtained from different scanners that have no raw X-ray energy data availability.

APPENDIX B

Seeing Through the Data: A Statistical Evaluation of Prohibited Item Detection Benchmark Datasets for X-ray Security Screening

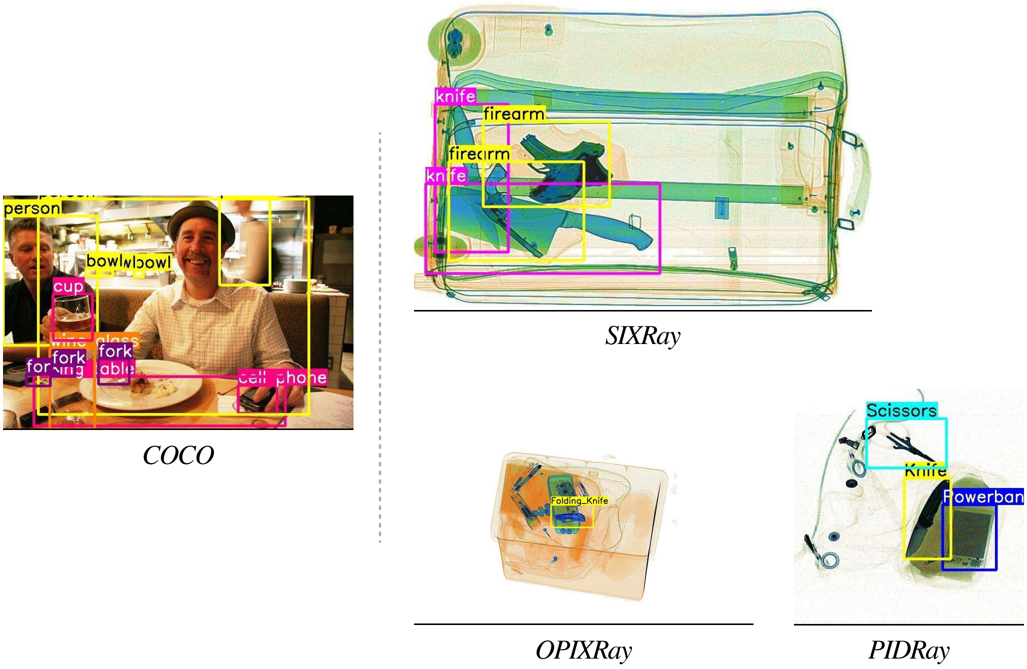
The rapid progress in automatic prohibited object detection within the context of X-ray security screening, driven forward by advances in deep learning, has resulted in the first internationally recognised, application-focused object detection performance standard (ECAC Common Testing Methodology for Automated Prohibited Item Detection Systems). However, the ever-increasing volume of detection work in this application area is highly reliant on a limited set of large-scale benchmark detection datasets that are specific to this domain. This study provides a comprehensive quantitative analysis of the underlying distribution of the prohibited item instances in three of the most prevalent X-ray security imagery benchmarks and how these correlate against the detection performance of six state-of-the-art object detectors spanning multiple contemporary object detection paradigms. We focus on object size, location and aspect ratio within the image in addition to looking at global properties such as image colour distribution. Our results show a clear correlation between false negative (missed) detections and object size with the distribution of undetected items being statistically smaller in size than those typically found in the corresponding dataset as a whole. For false positive detections, the size distribution of such false alarm instances is shown to differ from the corresponding dataset test distribution in all cases. Furthermore, we observe that one-stage, anchor-free object detectors may be more vulnerable to the detection of heavily occluded or cluttered objects than other approaches whilst the detection of smaller prohibited item instances such as bullets remains more challenging than other object types.

B.1 Introduction

X-ray security screening is widely used in aviation and other transportation domains, with a recent focus on the development of automatic identification of prohibited items within complex and cluttered X-ray images using a range of object detection approaches [189]. These developments have now led to changes in international aviation security regulations resulting in the first international security equipment standard for automatic prohibited item detection - the European Civil Aviation Conference (ECAC) Common Testing Methodology for the integration of Automated Prohibited Item Detection Systems (APIDS), which provides certified performance compliance for X-ray security scanner systems in the area of automated threat object detection (ECAC APIDS) and possibly represents one of the first, if not the first, internationally recognised performance standard for object detection algorithm performance [A27].

Within this context, prior work has investigated the performance of deep learning-based detectors for security inspection and threat-item detection within X-ray security imagery [188], [A28–A32]. Furthermore, recent work has seen the introduction of new paradigms for object detection, such as the use of Vision Transformers [7] and anchor-free models [46], [A33, A34]. However, the performance of all of these object detection approaches is very dependent on the availability of suitable X-ray security imagery datasets with sufficient object annotations, diversity and scale which has often been lacking within the common public X-ray dataset resources [189], [A35, A36].

Previous works have investigated the use of transfer learning to overcome the rel-



COCO

SIXRay

OPIXRay

PIDRay

Figure B.1: Typical images from X-ray datasets SIXray, OPIXray and PIDray.

atively small size of X-ray security datasets for image classification [219] and object detection [A30, A37] and report that a pre-trained model on a large-scale dataset such as ImageNet [79] or MS-COCO [88] results in higher detection performance despite the cross-over from perspective projection photographic imagery to parallel projection transmission imagery. However, pre-training on such datasets could induce dataset bias that may not hold for the target dataset [A38] which exhibits many differences from photographic image (object detection) datasets (Fig. B.1). For instance, X-ray images are semi-transparent transmission imagery, meaning that objects appear translucent and visually blended front-to-back whereas, in natural photographic images, foreground objects visually occlude background objects. As a result, the creation of dedicated X-ray security datasets has been an important step in the development of APIDS-capable approaches but in itself is inherently challenging due to the requirement for concurrent access to an X-ray security scanner, a diverse range of suitable prohibited threat items and similarly a suitably diverse set of passenger bags in which to em-place them. As a result, a limited number of large-scale benchmark datasets have emerged [116, 119], [A1, A39] upon which the relative performance analysis of APIDS capable approaches is now largely

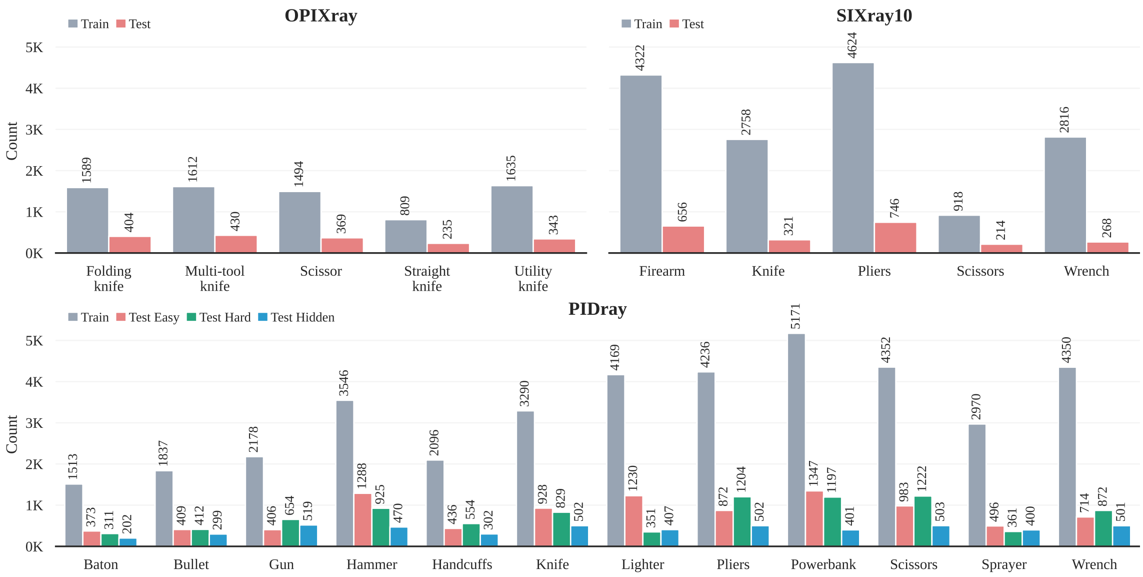


Figure B.2: PIDray, OPIXray, SIXray10 dataset statistics: class-wise prohibited item instances within {*Train*, *Test*} data splits.

reliant [189], [A11, A30, A35, A36]. Consequently, a statistical review of these benchmark dataset resources and their differences from more conventional object detection benchmark datasets [88], is an important step in improving the effectiveness of object detectors when applied to X-ray security prohibited item detection.

Beyond the specifics of X-ray imagery, multiple studies [A40–A42] provide ample evidence of dataset bias on common object recognition datasets, causing an inclination towards highly biased object detection models. In this regard, dataset bias refers to systematic errors in a dataset affecting the generalisation ability of learning-based algorithms, resulting in poor performance on models developed beyond the original dataset domain (distribution mismatch between dataset and the task) [A41, A43].

The majority of the methods for object detection bias mitigation utilise dataset re-sampling to adjust the relative frequencies of dataset samples, improving the model generalisation performance [A44–A46]. For instance, *REPAIR* [A44] removes the representation bias by learning a probability distribution over the dataset that favours hard instances for a given representation. On the other hand, *AFLITE* [A47] introduces adversarial filters designed to detect different types of dataset bias to eliminate noisy labels and feature distribution skewness before training the model.

Despite the study of dataset bias becoming particularly relevant for prohibited object detection, existing studies [A38, A48–A50] on dataset bias have been conducted on natural photographic (visible spectrum) datasets, such as the PASCAL Visual Object Classes [78], ImageNet [79] or COCO [88]. Furthermore, as the prohibited object detection literature commonly adopts pre-trained contemporary detection architectures [188, 219], there is an increasing possibility of encountering the aforementioned dataset biases and risks in X-ray security imagery.

Against this background, in this study, we analyze the underlying statistical trends of the image samples and object instances within the most extensive, and commonplace, X-ray security imagery datasets and their resultant impact on a suite of representative object detectors, providing extensive quantitative analysis on failure modes and potential sources of detection bias.
Our key contributions are as follows:

- A statistical evaluation of three of the most extensive and commonly used X-ray security imagery benchmark datasets, namely OPIXray [119], SIXray [116] and PIDray [A1], based on image and object instance properties, including image colour and object bounding box (location) distribution, highlighting the key differences against a standard natural image dataset (COCO [88]).

- A reference performance benchmark of six contemporary object detectors spanning different paradigms:- Cascade R-CNN [157] (multi-stage), Deformable DETR [225] (Transformer-based detection head), FSAF [A34] (anchor-free and online feature selection), Faster R-CNN [2] with Swin Transformers [7] (two-stage detector with a Vision Transformer-based backbone), YOLOX [46] (state-of-the-art one stage real-time) and CenterNet [A33] (keypoint-based).

- A quantitative investigation on the failure modes of the six different object detectors considered showing a correlation of the false negative and false positive detection occurrences against ground truth for the purpose of detection bias identification. Additionally, a class-wise analysis of the distribution of object instances within the training and testing sets for further understanding of detector performance and bias.

B.2 Evaluation Methodology

We present our evaluation methodology spanning down-selected datasets (Appendix B.2.1), object detectors (Appendix B.2.2) and object instance statistical analysis (Appendix B.2.3 in addition to implementation details (Appendix B.2.4).

B.2.1 Datasets

To assess the performance and potential dataset bias of X-ray security imagery, we analyse three of the most extensive, and commonly used, prohibited item detection datasets which are characteristically diverse, covering different X-ray scanners, prohibited item distribution and reflective of a likely real-world scenario.

OPIXray [119] consists of $8,885$ X-ray images with five classes of prohibited items (*folding knife, straight knife, scissor, utility knife, multi-tool knife*) and represents cluttered and overlapping stream-of-commerce baggage items.

SIXray [116] consists of $1,059,231$ images with $8,929$ X-ray images containing at least one prohibited item among five classes (*gun, knife, wrench, pliers, scissors*) originating from stream-of-commerce baggage and parcel X-ray scans collected from several subway stations. In this appendix, the *SIXray* partition is used, containing the $8,929$ images with prohibited items and $10\times$ images without.

PIDray [A1] is a large-scale prohibited items dataset including 12 classes of prohibited items (*baton, bullet, gun, hammer, handcuffs, knife, lighter, pliers, power bank, scissors, sprayer, wrench*) and $124,486$ images coming from three different scenarios (airports, subway stations and railway stations). The testing partitions are divided into easy (exactly one prohibited item), hard (two or more objects in the same image) or hidden (purposely hidden objects within the bag contents).

The distribution of prohibited item objects within these datasets is illustrated in Fig. B.2 with a comparison of their colour characteristics further shown in Fig. B.3.

B.2.2 Object Detection

To provide our performance benchmark, we down-select six state-of-the-art object detection architectures spanning differing detection paradigms (e.g. single-stage, multi-stage, deep convolutional neural networks, vision Transformers).

Cascade R-CNN (CR-CNN) [157]: is a modification of the R-CNN [71] that resolves the trade-off of having to choose between low Intersection over Union (IoU) thresholds that generate imprecise detections and high IoU thresholds that negatively affect performance. It does so by training a sequence of detectors one after the other, each with a progressively higher IoU threshold, to become more discerning in identifying false positives.

FSAF [A34]: is a single-stage object detection framework that uses feature selection on multiple anchor-free branches to overcome issues with heuristic-based feature selection and overlap-dependent anchor sampling. FSAF is built on a feature pyramid architecture and has been shown to improve object detection accuracy with minimal additional inference time.

Deformable DETR (DDETR) [225]: is an extension of the Detection Transformer (DETR) object detection model, which uses a transformer architecture to model sequential relationships between features that uses a deformable attention mechanism. Deformable DETR improves convergence by having attention modules focus only on adjacent features and addresses the issue of detecting objects at different scales. It retains the benefits of DETRs transformer-based architecture while achieving these improvements.

Faster R-CNN w/ Swin Transformer (FRCNNw/ST) [7]: Liu *et al.* introduced the Swin Transformer, a vision Transformer with shifted windows, which shows significant detection performance gains when used as a backbone for object detection. It is used in conjunction with Faster R-CNN [2], an anchor-based two-stage detector that uses a region proposal network.

YOLOX [46]: follows the success of the YOLO family of detectors, and is an anchor-free architectural variant of YOLOv3 [75] consisting of a decoupled detection head (*i.e.*, separated networks for classification and bounding box regression) and a strong label assignment and achieves state-of-the-art performance at real-time (YOLOX-S version).

CenterNet [A33]: converts the detection task to a keypoint detection by predicting the centre of the objects and regressing the remaining parameters. It achieves

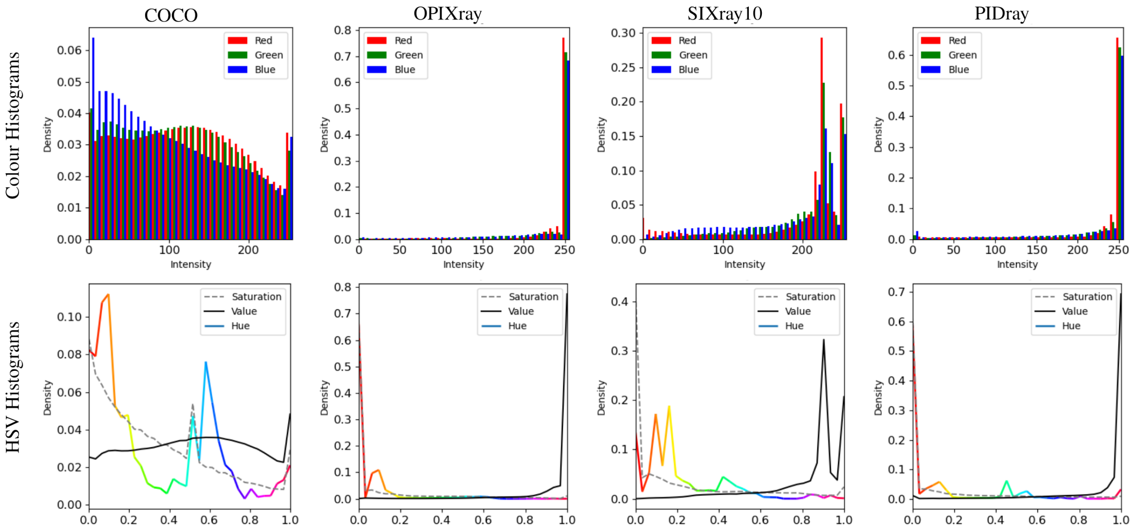


Figure B.3: RGB and HSV histograms for X-Ray datasets: OPIXray [119], SIXray10 [116] and PIDray [A1]; compared with COCO dataset [88].

a great speed-accuracy trade-off and can be used for other tasks such as 3D and keypoint detection.

B.2.3 Object Instance Analysis

In order to investigate the effect of the underlying distribution of object instances on detector performance, a statistical analysis of the distribution of three spatial parameters is performed: object area, centre and aspect ratio. In this context, *area* of an object refers to the total number of pixels that its bounding box occupies; *centre* is the geometrical centroid of the bounding box relative to the image and *aspect ratio* is the ratio of width to height. Regarding the centre, we report the Euclidean distance from the image centre. Our analysis aims to uncover the distribution of the location and size of objects within the sample images and how this potentially differs from a natural images dataset such as COCO [88]. Furthermore, the distribution of these parameters for false positive and false negative detection results is also performed.

B.2.4 Implementation Details

The training of the detector architectures (Appendix B.2.2) is implemented using the MMDetection framework [224]. All detectors are pre-trained on the COCO

Table B.1: Detectors training details.

Architecture	Optimiser	Epochs	Lr
CR-CNN [157]	SGD	20	10^{-2}
FSAF [A34]	SGD	20	10^{-2}
DDETR [225]	Adam [65]	50	10^{-4}
FRCNNw/ST [7]	AdamW [227]	30	10^{-4}
YOLOX [46]	SGD	20	10^{-3}
CenterNet [A33]	SGD	20	2×10^{-3}

dataset [88]. Training details are implemented using the default configurations with a few modifications, shown in Table B.1.

Standard data augmentation techniques as described in the original works are used. All training is carried out using an NVIDIA GeForce RTX 2080 Ti.

B.3 Evaluation Results

We present our evaluation spanning dataset analysis (Appendix B.3.1), detection performance (Appendix B.3.2) and detection relative to dataset object instance distributions (Appendix B.3.3).

B.3.1 Dataset Analysis

The colour analysis of the X-ray datasets compared to the COCO dataset is shown in Fig. B.3 in the form of RGB and HSV histograms. It is observed from the RGB histogram that while the COCO dataset has a seemingly uniform distribution across the intensity values, X-ray datasets are highly skewed to high values on the three RGB and HSV channels (mostly because of the white background). OPIXray and PIDray show higher peaks at 255 since they have large background regions. In contrast, SIXray10, where baggage images tend to occupy the full image plane, shows a peak at slightly smaller values, corresponding to the green, blue and orange colours of a typical bag (this peak is also observed for OPIXray and PIDray, albeit significantly lower). Additionally, the hue component distribution on the COCO dataset shows peaks at the orange (most likely corresponding to a range of lighter skin tones, since *person* is the most common category) and blue (sky in outdoor

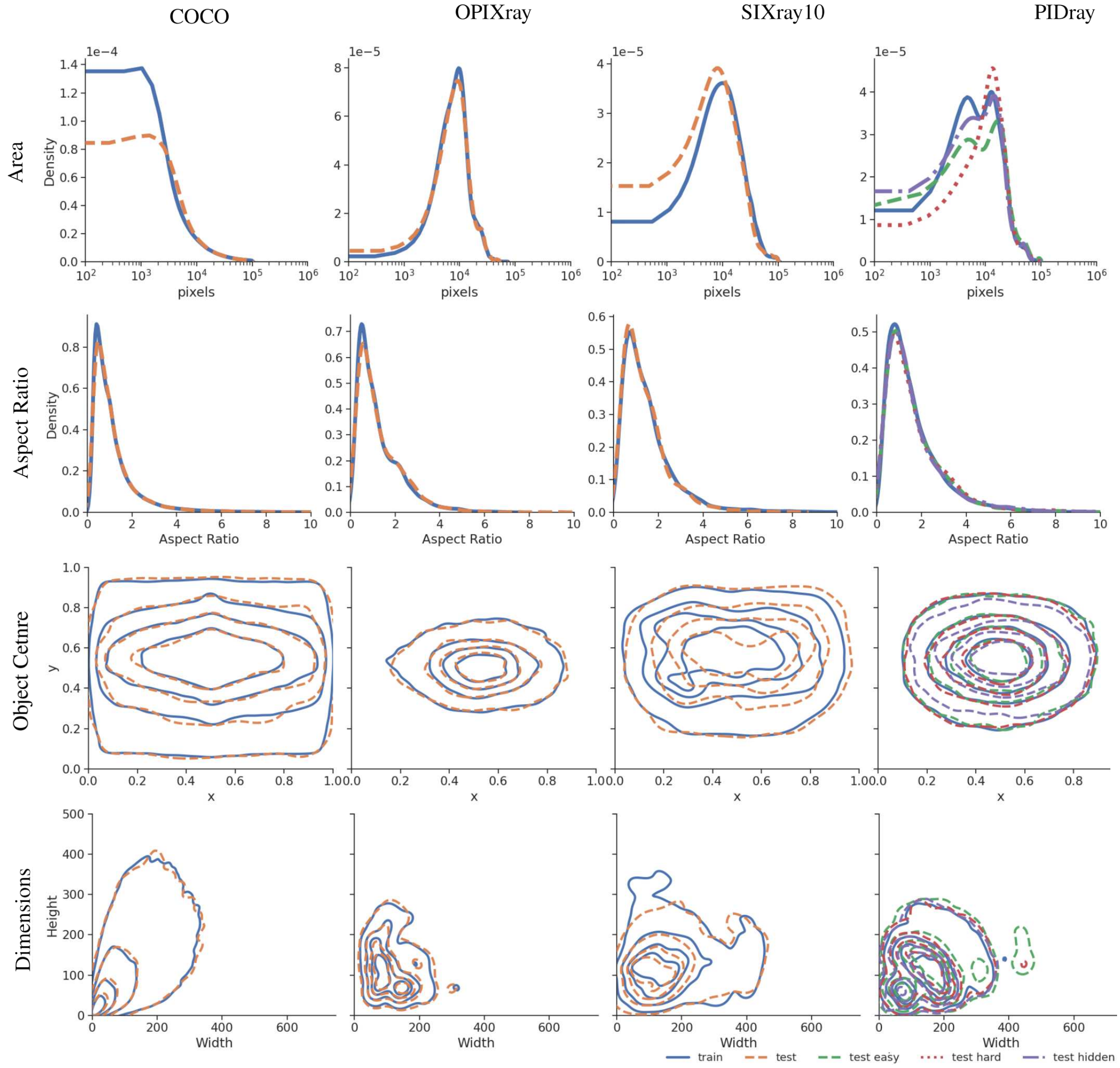


Figure B.4: Density estimation (using a Gaussian kernel) of the area, dimensions, bounding box centre and aspect ratio of the ground truth bounding boxes on OPIXray, SIXray10, PIDray and COCO.

images) colours, whilst the saturation mostly decreases towards bright colours, with one peak at high saturation values, indicating a high relatively presence of pure colours. On the other hand, the X-ray datasets are generally not saturated images with peaks at the blue and orange colours, having an additional peak with a hue component of zero (corresponding to the white background).

The object parameters distribution is presented in Fig. B.4. The dimensions, centre, aspect ratio and area, are shown as contour plots, where each contour represents the probability mass of lying among different density levels (10%, 30%, 50%, 70% and 90%) with densities obtained via Gaussian kernel density estimation. It

Table B.2: AP @ IoU=0.5 comparison for the *OPIXray* dataset.

Model	Folding	Straight	Scissor	Utility	M-tool	mAP
CR-CNN	0.934	0.771	0.961	0.836	0.949	0.890
FSAF	0.821	0.804	0.956	0.805	0.868	0.851
DDETR	0.909	0.774	0.963	0.859	0.934	0.888
FRCNNw/ST	0.945	0.842	0.977	0.854	0.959	0.915
YOLOX	0.908	0.801	0.974	0.859	0.935	0.896
CenterNet	0.911	0.758	0.977	0.820	0.909	0.875

Table B.3: AP @ IoU=0.5 comparison for the *SIXray10* dataset.

Model	Firearm	Knife	Wrench	Pliers	Scissors	mAP
CR-CNN	0.882	0.824	0.838	0.882	0.873	0.860
FSAF	0.894	0.776	0.792	0.885	0.898	0.849
DDETR	0.913	0.934	0.910	0.944	0.960	0.932
FRCNNw/ST	0.897	0.856	0.899	0.920	0.947	0.904
YOLOX	0.909	0.869	0.891	0.907	0.938	0.903
CenterNet	0.906	0.862	0.887	0.918	0.908	0.896

is observed from the area and dimensions distributions (Fig. B.4, upper two rows) that the COCO dataset has a higher concentration of small objects, while X-ray datasets have clear peaks at 10^4 pixels. This variation is explicable in relation to the perspective image view of the COCO images that gives rise to perspective foreshortening (*i.e.*, objects further away appear smaller) whilst the parallel projection of the X-ray scan alleviates any such perspective effects. Ultimately, pre-training on the COCO dataset may leverage this prior information and hence a bias to predict small objects can be induced (see Appendix B.3.3). The distribution of the object bounding box centres reveals that while objects tend to appear near the image centre in all datasets, they are constrained into the scanned region in the X-ray datasets, with OPIXray being the most constrained case (given the small size of bags in this dataset). Additionally, the distribution of the test sets is presented. A careful examination exhibits small variances in the area between the test and training sets on the SIXray10 and PIDray datasets, while other object parameters retain similar distributions. Finally, no significant difference is found with respect to aspect ratio.

Table B.4: AP @ IoU=0.5 comparison for the *PIDray* dataset. Three reported values are evaluated on {*easy/hard/hidden*} test sets.

Model	Baton	Pliers	Hammer	Powerbank	Scissors	Wrench	Gun	Bullet	Sprayer	HandCuffs	Knife	Lighter	mAP
CR-CNN	.985/.933/.357	.999/.965/.916	.960/.898/.774	.953/.951/.753	.958/.926/.735	.984/.969/.930	.158/.416/.655	.945/.873/.332	.775/.892/.544	.989/.983/.989	.379/.630/.479	.843/.741/.125	.827/.848/.633
FSAF	.982/.940/.357	.999/.970/.890	.965/.906/.719	.952/.965/.672	.924/.931/.621	.979/.957/.942	.088/.307/.550	.950/.909/.264	.748/.866/.595	.988/.982/.990	.279/.615/.474	.855/.765/.114	.809/.843/.599
DDETR	.989/.952/.589	.999/.983/.941	.971/.945/.860	.969/.968/.723	.970/.968/.845	.987/.983/.981	.099/.337/.645	.966/.877/.384	.950/.914/.703	.988/.986/.990	.578/.724/.537	.872/.781/.388	.861/.868/.716
FRCNNw/ST	.988/.976 /.717	.990/.979/.949	.988/.952 /.921	.969/.978 /.835	.981/.963/.910	.988/.987/.990	.506/.579/.756	.962/.872/.505	.958/.943/.676	.988/.986/.990	.692/.753/.620	.867/.787/.906	.906/.896/.765
YOLOX	.986/.958/.615	.989/.986/.883	.969/.943/.826	.964/.966/.737	.982/.964/.840	.958/.987/.978	.334/.472/.666	.960/.902/.393	.905/.928/.676	.989/.986/.990	.670/.707/.525	.846/.795/.213	.879/.883/.695
CenterNet	.977/.935/.935	.990/.975/.914	.972/.908/.655	.952/.955/.649	.967/.933/.649	.983/.970/.963	.278/.441/.568	.891/.748/.207	.732/.863/.334	.989/.987/.989	.439/.605/.362	.851/.723/.143	.835/.837/.566

B.3.2 Detection Performance

The detection performance across the OPIXray, SIXray10 and PIDray datasets is shown in Tables B.2 - B.4. In the X-ray security detection context, being able to detect an object is more important than how accurate the bounding box is, hence we report class-wise average precision (AP) and mean AP (mAP) across all classes considering an IoU threshold of 0.5. In general, Transformer-based detectors achieve the highest detection performances, with Faster R-CNN w/Swin Transformers illustrating superior detection for the OPIXray and PIDray datasets, and Deformable DETR on SIXray10. On the other hand, FSAF and CenterNet detectors perform the weakest. On an analysis of the test splits of PIDray (Table B.4), it is further observed that these two detectors have a significantly lower mAP for the hidden (heavily occluded object) test split, making them unreliable object detectors within this context. Interestingly, the mAP does not exhibit a notable change between the easy and hard splits (some classes increase their AP while others decrease it), indicating that the evaluated detectors are not heavily affected by the number of objects in them (the hard split contains exclusively more than one item). This is also observed by Song *et al.* [A51]. Additionally, some categories are more difficult to detect than lesser dangerous objects (*e.g.*, *Gun* vs *Wrench* in PIDray), demonstrating that a class-wise analysis is needed in order to create tailored object detectors that identify more important items.

B.3.3 Detection Performance Instance Analysis

The distributions of the ground truth bounding box properties presented in Appendix B.3.1, including area, centre and aspect ratio, indicate that there is no significant distribution variance within the training and testing X-ray security datasets. Accordingly, we question *Can the detectors perform reliably on objects that belong to*

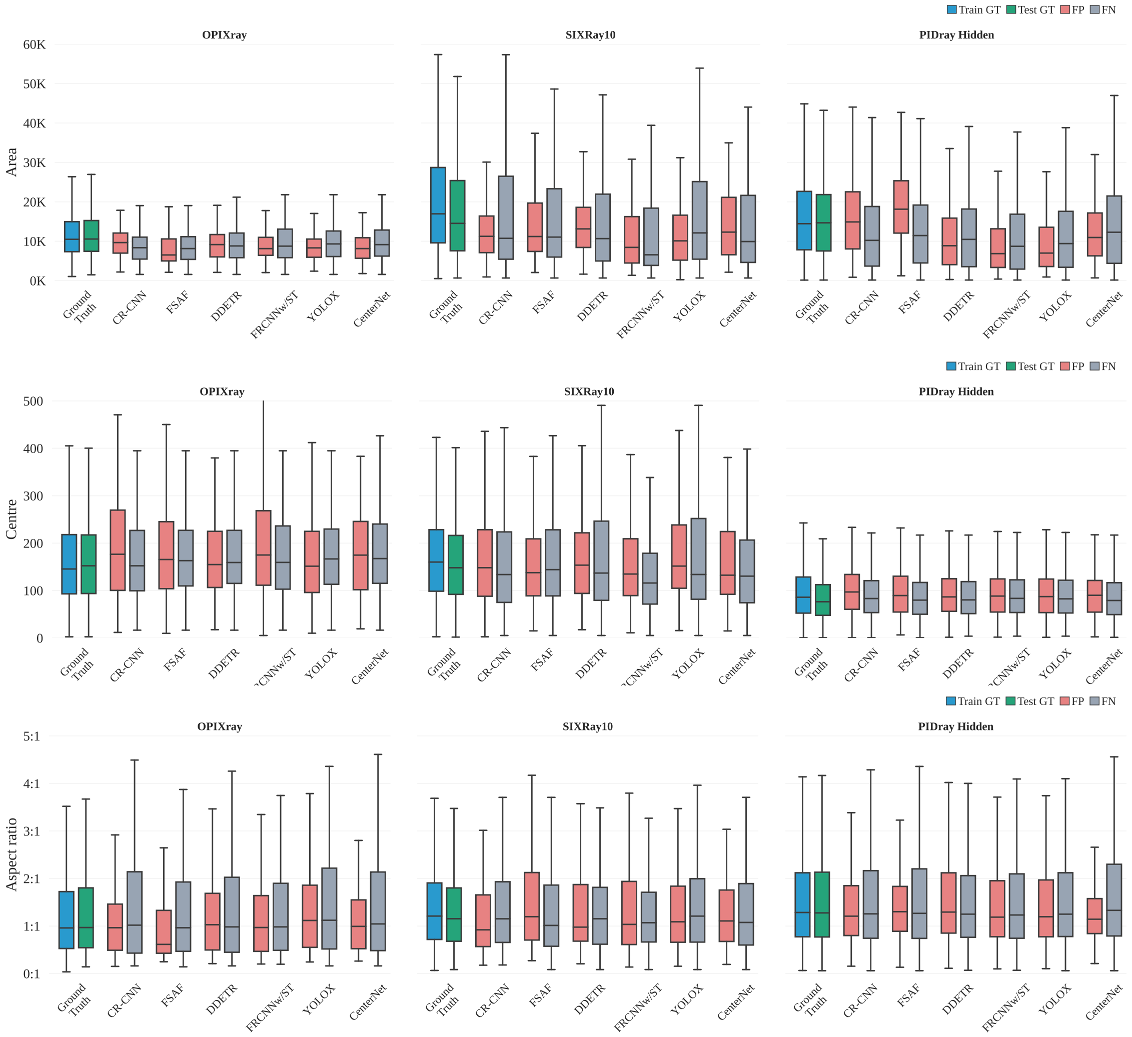


Figure B.5: The distribution of detector performance across object instance parameters regarding the evaluated sets including train, test ground truth, and predicted false positive and false negative sets.

the same training distribution? If not, how do the predictions vary across the selected object instance parameters? Subsequently, we evaluate the distribution of selected properties within False Negative (FN) and False Positive (FP) predictions from the chosen detectors and demonstrate the skewness of these distributions within training and testing splits (Fig. B.5). Regarding the *area*, it is observed that the median value of the area of FN samples across all datasets and detectors is smaller than that of the test and train distributions, indicating that undetected objects tend to have a smaller area (pixels) compared to the ground truth set area. In addition, the distribution of area in FP samples differs from the test distribution, with lower or higher variations depending on the detector and datasets. Notably, the FSAF

Table B.5: The Area Percentile Change on categories of PIDray $\{hidden,\ hard,\ easy\}$ sets from top to bottom, each cell depicts the $1-(median(set)/median(test))$ meaning that red colour cells have a larger change of the area among object categories.

Detector	Set	Baton	Bullet	Gun	Hammer	HandCuffs	Knife	Lighter	Pliers	Powerbank	Scissors	Sprayer	Wrench
GT	Train	-0.09	-0.41	0.07	0.19	0.08	0.02	-0.16	-0.24	0.15	-0.12	0.12	0.05
CR-CNN	FN	0.17	0.54	-0.81	-0.02	0.40	0.13	0.03	0.51	0.15	0.19	-0.03	0.34
	FP	0.04	-0.62	-0.12	0.23	0.62	0.44	-0.21	-0.39	-0.26	0.02	0.27	0.85
FSAF	FN	0.09	0.26	-0.02	-0.05	0.20	0.13	0.02	0.51	0.18	0.07	-0.01	0.15
	FP	0.27		-0.04	0.22	0.65	0.52	-0.45	-0.38	-0.35	0.29	0.22	0.92
DDETR	FN	0.18	0.59	-0.87	-0.11	0.23	0.21	0.09	0.52	0.07	0.21	-0.01	0.45
	FP	-0.06	0.24	0.33	0.34		0.67	-0.01	0.06	0.14	0.33	0.23	0.80
FRCNNw/ST	FN	0.18	0.68	-1.16	-0.08	- 24.26	0.29	0.07	0.24	0.04	0.28	0.02	0.86
	FP	0.35	-0.40	0.56	0.51	0.49	0.61	0.03	0.06	0.40	0.44	0.32	0.67
YOLOX	FN	0.21	0.58	-0.95	-0.16	0.01	0.27	0.03	0.38	0.10	0.24	-0.08	0.65
	FP	0.40	-0.03	0.30	0.40	0.78	0.46	0.10	0.19	0.29	0.44	0.40	0.72
CenterNet	FN	0.09	0.34	-0.62	-0.08	0.09	0.13	0.03	0.41	0.10	0.11	-0.01	0.42
	FP	0.41	-0.11	0.25	0.44	-0.20	0.42	0.02	0.07	0.30	0.15	0.30	0.58
GT	Train	0.02	-0.01	0.56	0.22	-0.15	-0.17	-0.10	-0.13	0.00	0.16	0.06	0.04
CR-CNN	FN	0.13	0.42	-0.05	0.13	0.47	0.01	0.19	0.09	0.09	0.28	-0.03	0.27
	FP	0.51	-2.43	0.59	0.48	0.00	0.04	-0.09	-0.05	0.10	0.08	0.22	0.11
FSAF	FN	0.19	0.25	-0.04	0.01	0.29	0.04	0.18	0.14	0.22	0.19	0.02	0.25
	FP	0.15	-2.22	0.71	0.35	-0.11	-0.10	-0.06	0.08	0.02	0.14	0.40	0.12
DDETR	FN	0.08	0.35	-0.05	0.06	0.25	-0.05	0.14	0.01	0.03	0.26	0.05	0.25
	FP	0.30	-1.54	0.50	0.40	-0.04	0.42	0.18	0.05	0.21	0.33	0.25	0.05
FRCNNw/ST	FN	0.13	0.48	-0.05	0.08	0.33	-0.09	0.16	0.09	0.11	0.28	-0.27	0.30
	FP	0.21	-1.93	0.52	0.53	0.01	0.43	-0.11	0.01	0.31	0.31	0.21	0.22
YOLOX	FN	0.13	0.53	-0.07	0.05	0.56	0.00	0.17	0.20	0.21	0.41	-0.16	0.47
	FP	0.31	-1.48	0.62	0.44	0.02	0.46	0.03	0.14	0.31	0.24	0.21	0.26
CenterNet	FN	0.30	0.44	-0.05	0.11	0.33	0.06	0.14	0.14	0.15	0.35	-0.02	0.30
	FP	-0.16	-1.94	0.75	0.42	-0.04	0.04	0.04	-0.17	-0.07	0.17	0.09	0.36
GT	Train	-0.05	-0.24	0.57	0.32	0.09	-0.21	0.04	0.03	0.13	0.31	0.44	0.11
CR-CNN	FN	0.38	0.74	-0.01	-0.09	-0.14	-0.09	0.23		0.31	0.46	0.04	0.36
	FP	0.52	-2.76	0.65	0.67	0.50	0.18	-0.07	-0.04	0.13	0.16	0.69	0.22
FSAF	FN	0.33	0.71	-0.00	-0.09	-0.06	-0.17	0.23	-0.07	0.27	0.34	0.08	0.38
	FP	0.63	-2.59	0.66	0.26	0.40	0.03	-0.09	0.38	0.02	0.10	0.72	0.48
DDETR	FN	0.33	0.77	-0.00	-0.18	0.03	-0.26	0.25	-0.51	0.21	0.45	0.08	0.14
	FP	0.18	-2.85	0.67	0.61	0.03	0.45	0.13	0.39	0.25	0.32	0.69	0.33
FRCNNw/ST	FN	-0.99	0.79	-0.02	0.05	-0.97	0.10	0.24	0.67	0.39	0.63	0.13	-0.38
	FP	0.39	-2.57	0.66	0.58	0.28	0.44	-0.67	0.05	0.28	0.25	0.69	0.34
YOLOX	FN	0.07	0.77	-0.05	-0.07	-0.97	0.21	0.26	-0.21	0.33	0.37	0.22	0.00
	FP	0.32	-2.08	0.62	0.76	-0.06	0.39	-0.53	0.49	0.27	0.36	0.71	0.23
CenterNet	FN	0.43	0.66	-0.01	-0.09	0.03	-0.04	0.18	-0.34	0.21	0.35	0.05	0.53
	FP	0.48	-2.57	0.66	-0.19	0.24	0.31	-0.46	0.37	0.30	0.20	0.67	0.43

detector on the PIDray Hidden (heavily occluded) set shows the most significant difference, where higher area size samples are mismatched. Conversely, the smallest distribution difference between the test and training sets was observed in the OPIXray dataset, resulting in smaller changes in predictions regarding their area. Concerning the *centre* parameter, we observed a slight increase in the median value of the distance of the FP predictions centre location from the centre of the image on the OPIXray dataset, while the rest did not exhibit any obvious trend. This indicates that while objects are usually constrained within an enclosed region, this does not affect modern detectors. As for the *aspect ratio*, the FN distribution in the OPIXray dataset shows a larger spread in aspect ratios than in the test set.

Furthermore, we explore the distribution shifts towards properties within class-level object bounding boxes within the datasets. As the area distribution exhibits the most significant changes in predictions, we focus our investigation on this parameter via the use of the PIDray test set (since it is the most challenging). Specifically, we first calculate the median area values of each class in the train, test, FN, and FP prediction sets. Subsequently, the relative error of the median $(1-(median_{set}(FP)/median_{area}(test)))$ of FN, FP and train ground truth with respect to the test ground truth is calculated (Table B.5), enabling us to determine *the relative change of the area among object categories regarding the evaluated sets.* Accordingly, negative values indicate that larger areas were miss-matched (FP/Test), or undetected (FN/Test), while positive values refer to smaller area predictions compared to test distribution within these classes.

From Table B.5, it is seen that the FP predictions for the bullet object category tend to be mismatched with larger area bounding boxes in all three PIDRay test sets. This can be explained given that bullets have small ground truth bounding boxes and small variations in the predicted bounding boxes give rise to high IoU. Conversely, wrenches are mismatched against smaller objects in the PIDray hidden (heavily occluded) data spit. It should be noted however that as some classes have fewer FN and FP depending on their performance, as the wrench category (Table B.4). With respect to the gun category, the distribution of the FN in the hidden set is significantly smaller, meaning that either the detector cannot locate highly cluttered guns and/or that they are just partially detected with smaller bounding boxes, having a similar problem with the IoU as in the bullets (but not as drastic). Finally, the highest difference is found in the FN for Faster R-CNN w/Swin Transformer on the handcuff category of the PIDray hidden set. This, however, corresponds to a single instance and is attributable to handcuffs being the only deformable object (due to the linking chain between the bracelets), resulting in variable object geometry and hence bounding box annotations.

B.4 Conclusion

In this appendix, we statistically evaluate three X-ray security imagery datasets, namely OPIXray [119], SIXray [116] and PIDray [A1]. The performance of six contemporary detectors operating with different deep learning paradigms is also evaluated, finding that Vision-Transformers-based detectors are the most reliable detectors and, conversely, one-stage anchor-free detectors have the worst performance, especially for heavily occluded objects. In addition, an analysis of the distribution of the properties of false positives and false negatives shows a bias towards smaller mismatches and undetected instances. It is also found that small categories, such as bullets, may be predicted with unrealistic sizes leading to lower overall detection performance. These results emphasize the importance of X-ray security image benchmark dataset analysis as a factor in the improvement of current and future object detectors in this context.

Appendices References

[A1] S. U. Khan, I. U. Khan, I. Ullah, N. Saif, and I. Ullah, "A Review of Airport Dual Energy X-ray Baggage Inspection Techniques: Image Enhancement and Noise Reduction," *Journal of X-ray Science and Technology*, vol. 28, pp. 481–505, 2020. A.1

[A2] A. Mouton and T. Breckon, "A Review of Automated Image Understanding within 3D Baggage Computed Tomography Security Screening," *Journal of X-ray Science and Technology*, vol. 23, no. 5, pp. 531–555, 2015. A.1

[A3] Z. Ying, R. Naidu, and C. Crawford, "Dual Energy Computed Tomography for Explosive Detection," *Journal of X-ray Science and Technology*, vol. 14, pp. 235–256, 2006. A.1

[A4] H. E. Martz and S. M. Glenn, "Dual-energy x-ray radiography and computed tomography," tech. rep., Lawrence Livermore National Lab.(LLNL), Livermore, CA (United States), 2019. A.1, A.3, A.3

[A5] B. Abidi, Y. Zheng, A. Gribok, and M. Abidi, "Ieee conference on computer vision and pattern recognition workshops (cvprw)," in *Screener Evaluation of Pseudo-Colored Single Energy X-ray Luggage Images*, pp. 35–35, 2005. A.1

[A6] K. Dmitruk, M. Mazur, M. Denkowski, and P. Mikolajczak, "Method for Filling and Sharpening False Colour Layers of Dual Energy X-ray Images," *IFAC and IEEE Conference on Programmable Devices and Embedded Systems*, vol. 48, no. 4, pp. 342–347, 2015. A.1

[A7] S. Anwar, M. Tahir, C. Li, A. Mian, F. S. Khan, and A. W. Muzaffar, "Image Colorization: A Survey and Dataset," *arXiv preprint arXiv:2008.10774*, 2020. A.1, A.2

[A8] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-Image Translation with Conditional Adversarial Networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1125–1134, 2017. A.1, A.2, A.4.3

[A9] N. Bhowmik, Y. Gaus, and T. Breckon, "On the Impact of Using X-Ray Energy Response Imagery for Object Detection via Convolutional Neural Networks," in *IEEE International Conference on Image Processing (ICPR)*, pp. 1224–1228, 2021. A.1, A.5.1, A.5.2, B.1

[A10] L. D. Griffin, M. Caldwell, and J. T. A. Andrews, "COMPASS-XP," in *Zenodo*, 2019. A.1

[A11] Z. Cheng, Q. Yang, and B. Sheng, "Deep Colorization," in *IEEE International Conference on Computer Vision (CVPR)*, pp. 415 – 423, 2015. A.2

[A12] R. Zhang, P. Isola, and A. A. Efros, "Colorful Image Colorization," in *European Conference on Computer Vision (ECCV)*, pp. 649 – 666, Springer, 2016. A.2

[A13] K. Nazeri, E. Ng, and M. Ebrahimi, "Image Colorization Using Generative Adversarial Networks," in *International Conference on Articulated Motion and Deformable Objects*, pp. 85–94, Springer, 2018. A.2

[A14] M. Limmer and H. P. Lensch, "Infrared Colorization Using Deep Convolutional Neural Networks," in *IEEE International Conference on Machine Learning and Applications*, pp. 61–68, IEEE, 2016. A.2

[A15] Q. Song, F. Xu, and Y.-Q. Jin, "Radar Image Colorization: Converting Single-Polarization to Fully Polarimetric Using Deep Neural Networks," *IEEE Access*, vol. 6, pp. 1647–1661, 2018. A.2

[A16] M. Chouai, M. Merah, J.-L. Sancho-Gomez, and M. Mimi, "Dual-energy X-ray Images Enhancement Based on a Discrete Wavelet Transform Fusion Technique for Luggage Inspection at Airport," in *International Conference on Image and Signal Processing and their Applications*, pp. 1–6, 2019. A.2

[A17] R. Kayalvizhi, Amit kumar, S. Malarvizhi, A. Topkar, and P. Vijayakumar, "Raw Data Processing Techniques for Material Classification of Objects in Dual Energy X-ray Baggage Inspection Systems," *Radiation Physics and Chemistry*, vol. 193, p. 109512, 2022. A.2

[A18] R. E. Alvarez and A. Macovski, "Energy-selective Reconstructions in X-ray Computerised Tomography," *Physics in Medicine and Biology*, vol. 21, no. 5, pp. 733–744, 1976. A.3

[A19] L. Jiang, B. Dai, W. Wu, and C. C. Loy, "Focal Frequency Loss for Generative Models," *arXiv preprint arXiv:2012.12821*, 2021. A.4.2

[A20] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," *arXiv preprint arXiv:1505.04597*, 2015. A.4.3

[A21] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks," in *IEEE International Conference on Computer Vision (ICCV)*, pp. 2242–2251, 2017. A.4.3

[A22] D. Ulyanov, A. Vedaldi, and V. S. Lempitsky, "Instance Normalization: The Missing Ingredient for Fast Stylization," *arXiv preprint arXiv:1607.08022*, 2016. A.4.3

[A23] Gilardoni, "Fep me 640 amx," 2018. `"https://www.gilardoni.it/en/security/x-ray-solutions/automatic-detection-of-explosives/fep-me-640-amx/"`. (accessed: 22.03.2022). A.5.1

[A24] J. Wang, K. Chen, R. Xu, Z. Liu, C. C. Loy, and D. Lin, "CARAFE: Content-Aware ReAssembly of Features," *arXiv preprint arXiv:1905.02188*, 2019. A.5.2, A.4

[A25] E. Secretariat, "CEP Implements Testing of APIDS and EVD, Extending the Programme to Nine Categories of Security Equipment," tech. rep., European Civil Aviation Conference, February 2023. B.1

[A26] M. Subramani, K. Rajaduari, S. D. Choudhury, A. Topkar, and V. Ponnusamy, "Evaluating One Stage Detector Architecture of Convolutional Neural Network for Threat Object Detection Using X-Ray Baggage Security Imaging," *Rev. d'Intelligence Artif.*, vol. 34, no. 4, pp. 495–500, 2020. B.1

[A27] Z. Liu, J. Li, Y. Shu, and D. Zhang, "Detection and Recognition of Security Detection Object Based on YOLO9000," in *IEEE International Conference on Systems and Informatics (ICSI)*, pp. 278–282, 2018. B.1

[A28] T. W. Webb, N. Bhowmik, Y. F. A. Gaus, and T. Breckon, "Operationalizing Convolutional Neural Network Architectures for Prohibited Object Detection in X-ray Imagery," *IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 610–615, 2021. B.1

[A29] B. Gu, R. Ge, Y. Chen, L. Luo, and G. Coatrieux, "Automatic and Robust Object Detection in X-Ray Baggage Inspection Using Deep Convolutional Neural Networks," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 10, pp. 10248–10257, 2021. B.1

[A30] A. Chang, Y. Zhang, S. Zhang, L. Zhong, and L. Zhang, "Detecting Prohibited Objects with Physical Size Constraint from Cluttered X-ray Baggage Images," *Knowledge-Based Systems*, vol. 237, p. 107916, 2022. B.1

[A31] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as Points," *arXiv preprint arXiv:1904.07850*, 2019. B.1, B.2.2, B.1

[A32] C. Zhu, Y. He, and M. Savvides, "Feature Selective Anchor-free Module for Single-shot Object Detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 840–849, 2019. B.1, B.2.2, B.1

[A33] D. Mery, D. Saavedra, and M. Prasad, "X-ray Baggage Inspection with Computer Vision: A Survey," *IEEE Access*, vol. 8, pp. 145620–145633, 2020. B.1

[A34] M. Rafiei, J. Raitoharju, and A. Iosifidis, "Computer Vision on X-Ray Data in Industrial Production and Security Applications: A Comprehensive Survey," *IEEE Access*, vol. 11, pp. 2445–2477, 2023. B.1

[A35] Y. F. A. Gaus, N. Bhowmik, S. Akcay, and T. Breckon, "EvaLuating the Transferability and Adversarial Discrimination of Convolutional Neural Networks for Threat Object Detection and Classification within X-ray Security Imagery," in *IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 420–425, 2019. B.1

[A36] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann, and W. Brendel, "ImageNet-trained CNNs are Biased Towards Texture: Increasing Shape Bias Improves Accuracy and Robustness," *arXiv preprint arXiv:1811.12231*, 2018. B.1

[A37] L. Zhang, L. Jiang, R. Ji, and H. Fan, "PIDray: A Large-scale X-ray Benchmark for Real-World Prohibited Item Detection," *arXiv preprint arXiv:2211.10763*, 2022. B.1, B.2.1, B.3, B.4

[A38] V. Riffo, H. Lobel, and D. Mery, "GDXray: The Database of X-ray Images for Nondestructive Testing," *Journal of Nondestructive Evaluation*, vol. 34, no. 4, p. 42, 2015. B.1

[A39] J. Ponce, T. L. Berg, M. Everingham, D. A. Forsyth, M. Hebert, S. Lazebnik, M. Marszalek, C. Schmid, B. C. Russell, A. Torralba, *et al.*, "Dataset Issues in Object Recognition," *Toward Category-level Object Recognition*, pp. 29–48, 2006. B.1

[A40] A. Torralba and A. A. Efros, "Unbiased Look at Dataset Bias," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1521–1528, IEEE, 2011. B.1

[A41] A. Khosla, T. Zhou, T. Malisiewicz, A. A. Efros, and A. Torralba, "Undoing the Damage of Dataset Bias," in *European Conference on Computer Vision (ECCV)*, pp. 158–171, 2012. B.1

[A42] T. Tommasi, N. Patricia, B. Caputo, and T. Tuytelaars, "A deeper look at dataset bias," *Domain Adaptation in Computer Vision Applications*, pp. 37–55, 2017. B.1

[A43] Y. Li and N. Vasconcelos, "REPAIR: Removing Representation Bias by Dataset Resampling," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9572–9581, 2019. B.1

[A44] W. Cai, R. Encarnacion, B. Chern, S. Corbett-Davies, M. Bogen, S. Bergman, and S. Goel, "Adaptive Sampling Strategies to Construct Equitable Training Datasets," in *Conference on Fairness, Accountability, and Transparency*, pp. 1467–1478, 2022. B.1

[A45] Y. Li and N. Vasconcelos, "Background Data Resampling for Outlier-aware Classification," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13218–13227, 2020. B.1

[A46] R. Le Bras, S. Swayamdipta, C. Bhagavatula, R. Zellers, M. Peters, A. Sabharwal, and Y. Choi, "Adversarial Filters of Dataset Biases," in *International Conference on Machine Learning (ICML)*, pp. 1078–1088, Pmlr, 2020. B.1

[A47] Z. Zhu, L. Xie, and A. L. Yuille, "Object Recognition with and without Objects," *arXiv preprint arXiv:1611.06596*, 2016. B.1

[A48] A. Barbu, D. Mayo, J. Alverio, W. Luo, C. Wang, D. Gutfreund, J. Tenenbaum, and B. Katz, "Objectnet: A Large-scale Bias-controlled Dataset for Pushing the Limits of Object Recognition Models," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 32, 2019. B.1

[A49] P. Stock and M. Cisse, "Convnets and ImageNet Beyond Accuracy: Understanding Mistakes and Uncovering Biases," in *European Conference on Computer Vision (ECCV)*, pp. 498–512, 2018. B.1

[A50] B. Song, R. Li, X. Pan, X. Liu, and Y. Xu, "Improved YOLOv5 Detection Algorithm of Contraband in X-ray Security Inspection Image," in *International Conference on Pattern Recognition and Artificial Intelligence (PRAI)*, pp. 169–174, 2022. B.3.2