Robust Explainable Image Classification from Synthetic Adversarial Images

Minor Thesis
submitted in partial fulfilment of
the requirements for the degree of
Master of Computer Science

Lucas Fern[†]
School of Computing and Information Systems
The University of Melbourne

Supervised by

Kris Ehinger
School of Computing
and Information Systems
The University of Melbourne

Tim Miller
School of Electrical Engineering
and Computer Science
The University of Queensland

Submitted: October, 2023

† lucas@lfern.com
0009-0008-1066-4552

Abstract

Image classification algorithms are deployed in a range of real-world environments. In many of these contexts, it is critical that these algorithms perform reliably and are trusted by users in order to maintain safe and effective operation. Despite this, existing computer vision models often fail when provided with input images that exhibit challenging characteristics such as unusual object poses, image backgrounds, or lighting configurations.

These variations occur frequently in the real world, and as such, it is important that models develop *invariance* to these parameters. This research aims to achieve this with the use of synthetic image data. Firstly, a large data set of synthetic images is produced, which includes significant variation across all of the aforementioned parameters, and contains more classes (48) and images (880,050) than similar existing data sets.

To gain a deeper understanding of the limitations of existing classification models, the data set is then used to evaluate four image classifiers on their invariance to the labelled parameters. This evaluation highlights that model performance depends significantly on object pose, validating findings by Alcorn et al. (2019), and shows that specific backgrounds and lighting conditions also have a considerable effect on classification accuracy.

In this evaluation, the *Swin Transformer V2 (Large)* model (Z. Liu et al., 2022) performs most accurately, and is most invariant to the changing parameters. As such, I produce a new model, STRobE, short for Swin Transformer (Robust and Explainable), by modifying the *Swin Transformer V2* and retraining it on the synthetic data set, with the aim of increasing its parameter-invariance *and* explainability.

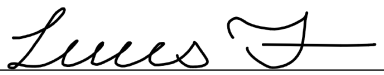
Explainability promotes further understanding of machine learning models, and in this model, explanations are facilitated by the extensive labelling of the synthetic data set. Using these labels, the STRobE model is trained to predict the pose of the image subject, the background class, and the lighting direction in images it classifies. These predictions are then used to synthesize image-based explanations representing the model's understanding of its inputs.

While the STRobE model did not train to convergence within the time frame of the project, the partially trained model demonstrates that training on highly diverse data sets results in significantly improved robustness. The model also learns to effectively predict most labelled parameters, facilitating an evaluation of its image-based explanations. These explanations are shown to provide value by highlighting potential reasons for type-II errors.

Declaration

I, *Lucas Fern*, declare that this thesis titled, *Robust Explainable Image Classification from Synthetic Adversarial Images* and the work presented in it are my own. I confirm that:

- The thesis comprises only my original work towards the *Master of Computer Science*;
- Due acknowledgement has been made in the text to all other material used; and
- The thesis is fewer than the maximum word limit in length, including text and headings from the start of the introduction to the end of the conclusion, and exclusive of text in figures, tables, bibliographies, and appendices.

Signed: 

Date: 30/10/2023

Acknowledgements

I begin by acknowledging the traditional owners of the land on which I completed this research, the Wurundjeri and Bunurong People of the Kulin Nation. I pay my respects to their Elders past, present, and emerging, and recognise that sovereignty was never ceded.

To Kris and Tim, my supervisors. Thank you both for your support and advice throughout this project, both on a technical and personal level. You are both wonderful teachers and mentors. Kris, you provided my first introduction to machine learning, and I look back on this as something that defined my trajectory through higher education. Tim, I credit you for bridging my passion for AI with my commitment to ethics and my desire to cause positive change. It was a pleasure learning from you over multiple semesters and a privilege to teach alongside you this year.

To my parents, thank you for raising me in a way that cultivated curiosity, a passion for continuous learning, and my attention to detail. Thank you for supporting me throughout my education and while I completed this project. To my brother Elliot, thank you for your support, and for the knowledge and experiences we share.

To my wider family and mentors. Thank you for the guidance and inspiration you provide, and how you've shaped me into the person I am today.

To my friends. Thanks for being supportive, and for reminding me what life is like outside of my research. You've made this year of my life and education truly enjoyable.

Finally, to everyone who provided feedback on this thesis. Kris Ehinger, Tim Miller, Amy Mendelsohn, Jake Godfrey, Felix Harvey, Lex Gallon, and Dad–thank you. This final copy contains countless amendments based on your advice.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In the wider landscape of artificial intelligence research, the evolution of computer vision models reflects the significant advancements made in machine learning in recent years. As deep learning research continues to accelerate (Maslej et al., 2023), computer vision models are increasingly approaching, and in some cases surpassing, the performance of the human visual system. However, the proficiency with which humans are able to identify and label objects, especially under adverse conditions, highlights a significant limitation on the capabilities of vision models.

Image classification, a subfield of computer vision that is concerned with the categorisation of images into predefined classes, is a specific domain where this limitation is especially evident. At their core, image classification algorithms aim to tell a user *what is depicted in an image,* and artificial intelligence (AI) systems capable of performing this task have applications across many domains. These applications range from everyday conveniences like photo album organisation, to safety-critical uses such as disease detection in medical images, and autonomous navigation (Dollar, Wojek, Schiele, & Perona, 2011; Esteva et al., 2017; Janowczyk & Madabhushi, 2016). It's in these later applications, where decisions may have profound consequences, that the reliability of image classification algorithms is paramount.

To formally evaluate the reliability of these algorithms in this research, we consider specific parameters that are known to cause significant changes to the appearance of an image. These include (a) object pose (the position *and* orientation of the image subject), (b) image background, and (c) lighting direction. These scene parameters (defining the environment in which the image is captured, as opposed to camera parameters) are sources of within-category variation, as they result in changes to an image without altering the class label. These three specific features are hereafter referred to as the *explanation parameters*.

A classification algorithm that performs accurately and consistently across the entire range of any given parameter is considered *invariant* to that parameter, and models with greater invariance to changing image parameters are considered more *robust.* Such models are not only safer to deploy in the real world due to their reliable performance under adverse condi-

tions, but they are also perceived as more trustworthy by users (Madry, Makelov, Schmidt, Tsipras, & Vladu, 2017; Ribeiro, Singh, & Guestrin, 2016). Despite this, a review by Drenkow, Sani, Shpitser, and Unberath (2022) concludes that significant further research is required to achieve robustness to even minor, natural variations in image parameters.

This not only suggests that existing computer vision models may perform unreliably when deployed in the real world, but also indicates that there is room to engineer machine learning models that are ascribed a greater level of trust by users. While producing parameter invariant models is one way to achieve this, the field of Explainable AI (XAI) is also concerned with producing trustworthy algorithms by making them more interpretable to humans. In addition to being more trustworthy, XAI systems promote accountability and transparency, yet despite these benefits, state-of-the-art deep learning models are notoriously *uninterpretable* (Y. Zhang, Tiňo, Leonardis, & Tang, 2021).

In an attempt to address these deficiencies in both the explainability *and* robustness of image classification algorithms, this research aims to answer the following two questions:

Question 1: How robust are state-of-the-art image classification algorithms to variation in the explanation parameters?

Question 2: Can training these models on *synthetic images* improve their robustness, and allow them to *explain* the object pose, image background, and lighting direction that they identify in an image?

These research questions are answered in stages by pursuing the following three objectives:

Objective 1 (Image Synthesis): Generate a synthetic image classification data set containing images of objects under challenging conditions. The parameters that will be varied and annotated are: (a) object pose, (b) image background, and (c) lighting direction (the explanation parameters).

Objective 2 (Existing Model Evaluation): Use the synthetic data set produced for Objective 1 to evaluate the robustness of state-of-the-art image classifiers to variation in the explanation parameters. Here, failure cases serve as counterfactual examples showing the conditions to which the classifier is not invariant.

Objective 3 (Model Training): Modify one of the image classification algorithms benchmarked for Objective 2 to increase its explainability and robustness. Add additional explanatory outputs, and train the model to predict the explanation parameters using the data set produced for Objective 1. Use these explanatory outputs to synthesise explanatory images by providing them as input to the existing image synthesis pipeline.

By completing these objectives, this research makes various contributions to the fields of computer vision and XAI. By producing a data set of accurately labelled synthetic images, containing objects under challenging conditions, Objective 1 (Image Synthesis) addresses a gap in current computer vision research. While other data sets that control for certain specific parameters exist in both the real image and synthetic domains (Alcorn et al., 2019; Barbu et

al., 2019), the data set produced in this research is both larger in size and contains labels for more image parameters. Complete details about the synthesis and distribution of this data set are provided in Chapter 3.

This large repository of synthetic images provides a controlled environment in which the robustness of current and future image classification models can be evaluated. We begin this evaluation in Chapter 4 by assessing the parameter invariance of four state-of-the-art image classification algorithms. By analysing the performance of a representative sample of classification models, this investigation makes multiple contributions. Firstly, by assessing the ways that these models respond to object pose, we validate findings by Alcorn et al. (2019) who find that computer vision models are prone to misclassification on images with even minor pose variation.

Following this, the impact of the other explanation parameters are investigated, producing novel results about the ways that classification models respond to changes in lighting conditions and background. Again we find that existing models respond to changed lighting conditions and backgrounds in a significant way. This suggests that there is value to be gained by training a parameter-invariant model using the synthetic data set.

This is addressed as Objective 3, where the *Swin Transformer V2* model (Z. Liu et al., 2022) is modified to predict the explanation parameters of the images it classifies. This model, designed to be both *robust* and capable of producing image-based *explanations*, is referred to as STRobE, short for Swin Transformer (Robust and Explainable). While this model did not learn to make meaningful predictions of the explanation parameters within the time frame of this project, the architecture and training process are summarised in Chapter 5 to suggest directions for future research. Additionally, the image-based explanation technique is evaluated by assessing explanatory images produced using manual annotations of challenging images.

Throughout each stage of this project, various avenues for future research are presented. These include suggestions for producing synthetic data sets that are more photorealistic, for evaluating existing models across other image parameters, and notably, for producing a more successful robust model. These limitations and directions for future work are presented in Chapter 6. To facilitate a complete understanding of this project and these suggestions for future work, we begin with a review of relevant background literature in Chapter 2.

Chapter 2

Background

The three objectives presented in Chapter 1 provide a rough outline for this literature review. As Objective 2 (Existing Model Evaluation) and Objective 3 (Model Training) involve the evaluation and training of image classification models, Section 2.1 identifies a set of image classifiers that are most relevant to these goals. As this research focuses on improving the robustness and explainability of these models, existing techniques for achieving this are reviewed in Sections 2.2 and 2.3 respectively.

Since synthetic images will be used to evaluate robustness and improve explainability, Section 2.4 investigates existing approaches for image synthesis. Finally, multi-task learning is discussed in Section 2.5 as the explainable model produced for Objective 3 (Model Training) will be required to produce multiple explanatory outputs.

2.1 Image Classification

Image classification is a computer vision task where an algorithm is required to assign a label to an input image from a predefined set of categories. This fundamental computer vision task is applied in many domains, ranging from medical imaging, where it is used to detect and assist in the diagnosis of diseases (e.g. Esteva et al., 2017; Gulshan et al., 2016; Janowczyk & Madabhushi, 2016), to autonomous navigation, where it has been used for pedestrian detection (Dollar et al., 2011) and to identify traffic signs (Sermanet & LeCun, 2011), among other applications. Image classification is an essential tool in these domains because of its ability to automate tasks which are time consuming and may be prone to errors when performed by humans, however, the safety-critical nature of these applications underscores the importance of classification algorithms that perform reliably, even under challenging conditions.

In real-world applications of classification models, image conditions may vary significantly between inputs. This can be a result of different cameras and camera configurations, or a result of objects and scenes that are positioned, illuminated, or constructed in different ways. In high-stakes applications like those mentioned above, it is important that classification

algorithms perform reliably in order to preserve safety and trust. It is therefore important that these models are not overly sensitive to changes in the aforementioned image conditions, for example, an autonomous vehicle should be able to identify pedestrians regardless of the lighting conditions or the pedestrian's pose.

While models that perform well under diverse conditions are essential for building safety and trust, these properties are further enhanced with the use of *explainable models*. In the context of image classification, users may show increased trust in models when they understand the image features and decision processes that resulted in the provided classification result.

Achieving explainability and invariance to changing image parameters are challenging problems in computer vision due to the drastic changes in appearance that can be caused by lighting conditions, backgrounds, and object poses, among other variables. The existing research into producing robust and explainable models is covered later in this chapter, and as such, the remainder of this section provides a high level overview of the history of image classification, presenting a range of techniques that represent the broad landscape of classification approaches.

2.1.1 Evolution of Image Classification Techniques

There have been significant developments in the field of image classification since the problem was first approached. While initial approaches relied on handcrafted features, modern techniques utilise a variety of sophisticated deep learning methods. The evolution of these techniques can broadly be categorised into three eras: pre-deep learning, followed by the rise of Convolutional Neural Networks (CNNs), followed more recently by the emergence of transformer-based architectures. Key developments from each era are summarised below, emphasising the continuing desire for interpretable models that are robust to changing image conditions.

2.1.1.1 Pre-deep learning Methods

Prior to the introduction of deep learning, image classification primarily relied on handcrafted features and traditional machine learning algorithms. A widely used technique from this era, which is applied to this day, is the Scale-Invariant Feature Transform (SIFT) (Lowe, 2004). As the name suggests, this method identifies features in an image that are invariant to scale and rotation, and partially invariant to both viewpoint and illumination. While SIFT is not a classification algorithm by itself, SIFT features can and have been used as inputs to other classification algorithms. This demonstrates a focus on parameter invariance since the early days of computer vision.

There are many other methods for automatic feature extraction including the Histogram of Oriented Gradients (HOG) (Dalal & Triggs, 2005), which produces features by analysing image gradients in small regions of the input, as well as GLOH, (Mikolajczyk & Schmid, 2005), FAST, (Rosten & Drummond, 2006), BRIEF, (Calonder, Lepetit, Strecha, & Fua, 2010), BRISK, (Leutenegger, Chli, & Siegwart, 2011), and ORB (Rublee, Rabaud, Konolige, &

Bradski, 2011). The introduction of each of these techniques is motivated by an improvement to efficiency, or importantly for this research, improved invariance to image conditions.

Features extracted using these methods were often used for classification by providing them as input to a Visual Bag of Words (VBoW) classifier (Csurka, Dance, Fan, Willamowski, & Bray, 2004). This light-weight method can be used to perform classification with arbitrary feature descriptions by finding their nearest-neighbours in feature-space (Guo, Wang, Bell, Bi, & Greer, 2003). While this method is relatively interpretable when compared to the wider image classification landscape, this approach has largely been superceded by the deep learning methods introduced in the following sections.

2.1.1.2 CNN-based Methods

While developments in computer vision have been divided into different eras for the purpose of this overview, it is important to note that there is significant overlap between the time periods of each era, and techniques of all descriptions continue to be developed to this day. It is important to emphasise this since CNNs were first proposed by LeCun, Bottou, Bengio, and Haffner in 1998. While CNN-based methods existed throughout the 2000s, limitations on hardware, research, and data sets restricted the ability to train deeper models on large amounts of data.

Around 2010, multiple developments converged, resulting in significant and rapid development of deep learning algorithms. Two critical developments in this period were the introduction of large data sets such as ImageNet (Deng et al., 2009), and the speed increases found when training neural network models on Graphics Processing Units (GPUs) (Chellapilla, Puri, & Simard, 2006; Raina, Madhavan, & Ng, 2009). This resulted in the introduction of modern deep learning architectures for image classification, such as *AlexNet* by Krizhevsky, Sutskever, and Hinton (2012). This model, containing approximately 60 million parameters across 650,000 neurons, achieved top-1 and top-5 error rates of 37.5% and 17.0% on the ImageNet benchmark, considerably exceeding the existing state of the art.

In the years following the introduction of this model, many improvements have been made to CNN architectures. *VGGNet* (Simonyan & Zisserman, 2014) demonstrates the effectiveness of deep architectures, *ResNet* (K. He, Zhang, Ren, & Sun, 2016) introduces the concept of *skip connections* to alleviate the vanishing gradient problem that can arise with deep architectures, and *DenseNet* (Huang, Liu, Van Der Maaten, & Weinberger, 2017) builds further connectivity between layers to both enable skip connections and enhance information sharing.

While this summary hardly scratches the surface of modifications made to the CNN architecture, it is relevant to note that performance improvements tend to arise as a result of training larger and more efficient models on larger and more diverse data sets (Dawson, Dubrule, & John, 2023; Gong, Zhong, & Hu, 2019). This is somewhat different to what is seen with pre-deep learning models, which tend to increase performance by explicitly crafting features

that are more invariant to changing image conditions. As a result of this, the evaluation performed for Objective 2 (Existing Model Evaluation) of this project will select CNN models based not on which are explicitly designed to be invariant to changing conditions (Section 2.2 demonstrates that very few models are explicitly designed for this goal), but instead to select a *diverse* sample of models that represent the state of the field at their time of release.

To select diverse models, it is critical to consider the primary ways that CNN architectures differ from one another. A key consideration here is that models are often released in different sizes to cater to different computational constraints. On the smallest scale, models specifically designed for mobile and edge computing devices, such as the *MobileNet* family of models (Howard et al., 2017; Sandler, Howard, Zhu, Zhmoginov, & Chen, 2018) prioritise efficiency. The primary *MobileNet V2* model contains only 3.4 million parameters (considerably less than the 60 million of the early *AlexNet* model), and performs only 3.4% of the computation per inference (measured in Multiply-Add operations) when compared with *ResNet-101*.

On the other extreme, many models are released with larger variants that often scale up depth or the amount of feature representations that are learned. *ResNet* is one such example, which exists in 18, 34, 50, 101, and 152-layer variants. The largest of these models, *ResNet-152*, contains a similar number of parameters to *AlexNet* at 58 million (T. Chen, Kornblith, Swersky, Norouzi, & Hinton, 2020), but achieves state-of-the-art performance in 2016 due to various architectural improvements.

Based on this overview of influential CNN models, *MobileNet V2* and *ResNet-152* are selected for evaluation for Objective 2 (Existing Model Evaluation). It is believed that by evaluating these two models that share the CNN architecture, but otherwise differ in many ways, the most information will be gained about the invariance of CNN models to changing image conditions.

2.1.1.3 Transformer-based Methods

More recently, the *transformer* module, initially applied in the field of Natural Language Processing (NLP) (Vaswani et al., 2017), has been adapted to produce high-performing computer vision models. The architecture of transformer-based models is significantly different from CNN-based ones, and recent research by Raghu, Unterthiner, Kornblith, Zhang, and Dosovitskiy (2021) finds "striking differences" between the representations that are learned by CNN and Vision Transformer (ViT) architectures. As such it is important that models of this variety are also included in the evaluation performed for this research. Similar to how CNN-based models were introduced and selected above, this section presents a brief overview of transformer-based models in image classification, and selects two models for evaluation that embody different goals and approaches.

The initial *ViT* model, introduced by Dosovitskiy et al. (2020), was the first to apply the transformer architecture to computer vision tasks. Their approach of dividing an input image into patches and processing these image patches with a transformer module achieved compet-

itive performance with state-of-the-art CNN-based models on release. The *ViT* architecture is visualised and explained further in Figure 2.1.

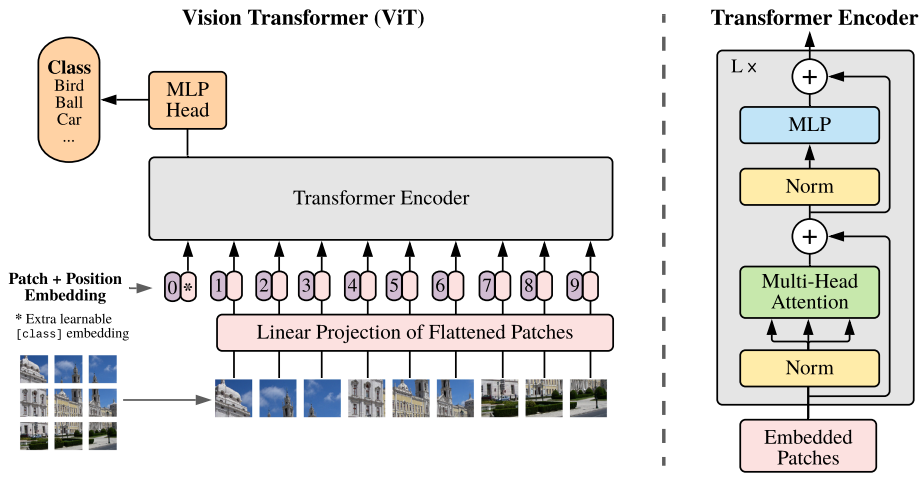


Figure 2.1: The architecture of the *ViT* model. The left side shows the construction of the *ViT* model, demonstrating how inputs are divided into fixed-size patches and projected into a linear representation before being input to the transformer encoder. The transformer encoder comprises multiple transformer *modules* (proposed by Vaswani et al., 2017). These are shown on the right, including the multi-head attention layer that allows the model to attend to arbitrary regions of the input image. *Image from Dosovitskiy et al. (2020).*

A key point of difference between CNN and transformer architectures is the *attention mechanism* present in the transformer module. While CNNs perform convolutions on local regions of an image during processing, gradually increasing the receptive field in each layer, the attention mechanism allows transformer-based models to capture global context by attending to arbitrary regions of the image throughout the inference process.

In the years since the release of *ViT*, many high-performing models have been produced by adapting and improving its architecture. One such example is the *Swin Transformer* family of models (Z. Liu et al., 2022, 2021), which introduce hierarchical image patches, encouraging the model to make further use of global image context. Transformer in Transformer (TNT) (Han et al., 2021) is another modified architecture that uses an additional transformer block which operates *within* each image patch to attend to finer details in the input.

While Vision Transformers have achieved state-of-the-art performance across many tasks, it is recognised that ViT models demand larger training data sets to achieve comparable performance with CNN-based models, and that their large receptive field demands more computation for inference. Certain models, such as the Data Efficient Vision Transformer (DEiT) (Touvron et al., 2021) attempt to solve these issues within the framework of a traditional Vision Transformer. DEiT addresses the need for large training data sets using *knowledge distillation* (Hinton, Vinyals, & Dean, 2015), however, other researchers have attempted to combine elements from CNN and transformer architectures, producing Hybrid Vision Transformers (HVTs).

HVTs such as *ConViT* (d'Ascoli et al., 2021) are explicitly designed to introduce features

of CNN models into the ViT architecture. A relevant example is the translation invariance that is built into CNN-based models due to the nature of convolutional layers. *ConViT* incorporates this inductive bias by introducing *Gated Positional Self-Attention (GPSA) layers*. These layers are capable of behaving as convolutional layers, but may also learn to recover the expressivity of a transformer attention layer by learning its own *gating parameter*.

Finally, the demand for mobile-scale models also exists for HVTs. While the full transformer module is large in size, and is therefore generally unsuitable for mobile applications, hybrid models can be more efficient and are therefore able to be condensed to a mobile scale. Mobile HVTs include the *MobileViT* family of models (Mehta & Rastegari, 2021, 2022), and the *EfficientFormer* (Y. Li et al., 2022), among others.

For the purposes of evaluating existing models, it is considered appropriate to select one large Vision Transformer model, and one mobile-scale HVT model. As such, the *Swin Transformer V2* and *MobileViT V2* models have been selected based on their high performance within their respective categories. While this selection of two models does not constitute a comprehensive representation of the Vision Transformer space, the use of one transformer-based and one hybrid model was considered acceptable and ensures a manageable scope for the evaluation process.

2.2 Invariance to Image Parameters

Despite the advanced level of classification performance achieved by the state-of-the-art computer vision models introduced above, they are known to perform poorly on images that are captured under challenging conditions. To measure this, image parameters can be systematically varied, and the change in model performance over this parameter space can be assessed. For the purposes of this research, the image parameters that are considered for evaluation include object pose, image background, and lighting direction (the explanation parameters).

Generally it is desirable for classification models to perform accurately on inputs throughout the entire image parameter-space, as this increases their ability to generalise to unseen data, resist adversarial attacks (Madry et al., 2017), and be trusted by users (Ribeiro et al., 2016). Models that perform consistently over the distribution of a given parameter are considered *invariant* to that parameter. Similarly, *robustness* is used in some existing research to refer to models that preserve performance across alterations in image conditions (Drenkow et al., 2022).

Despite the aforementioned benefits of robust models, a 2022 review concludes that robustness is the subject of a disproportionately small amount of computer vision research, and that a significant research gap still exists (Drenkow et al., 2022). This project aims to fill said research gap, and as such, existing research on invariance to the explanation parameters is summarised in the following sections.

2.2.1 Invariance to Object Pose (Rotation)

Pose, usually measured in six dimensions, refers to both the position and orientation of an object in an image. Pose invariance therefore considers how a model responds to the subject appearing in unusual or unseen positions or orientations. This is a challenging problem for multiple reasons. Intuitively, many 3D objects look considerably different when projected into 2D images from different orientations. Even slight changes can lead to significant variation in appearance. In addition to this, many objects tend to be photographed from particular poses in real images. This favoured rotation, referred to as the *canonical pose*, tends to be represented most frequently in real-image data sets, and other poses may be significantly under-represented in comparison.

The fact that these problems are inherent to real images and image data sets means that rotation invariance cannot be easily achieved with changes to model architecture in the same way that translation invariance can. Looking at pose invariance in image classification models, work by Alcorn et al. (2019) finds that CNN-based models such as *ResNet* (K. He et al., 2016) and *InceptionV3* (Szegedy, Vanhoucke, Ioffe, Shlens, & Wojna, 2016) perform poorly when exposed to objects in strange poses. They suggest that augmenting training data with diverse poses has the potential to improve robustness.

This is corroborated by H. Yu and Oh (2021) who use image sequences obtained from moving vehicles to produce data sets with greater representation of poses. They show that representing and annotating more diverse poses in training data results in higher accuracy on pose estimation tasks.

An alternative approach to producing pose invariant models is to design model architectures with specific elements to reduce pose dependence. To this end, Dong and Lin (2020) propose an object detection algorithm that is robust to image-plane rotation by modifying the architecture of a CNN model, however this method does not extend to 3D pose, as changes in 3D pose fundamentally alter the information contained in the image. In general, it is recognised that further work is required to produce classification algorithms that are robust to even minor, natural variations in object pose (Alcorn et al., 2019; Engstrom, Tran, Tsipras, Schmidt, & Madry, 2019).

2.2.2 Background Invariance

Background invariance is a similarly challenging problem for computer vision models, as backgrounds may contain confounding features and objects, or may be highly correlated with specific object classes. Taking the *toothbrush* object class as an example, models may learn to associate the class with features present in bathrooms. Not only may this result in misclassification of *toothbrushes* that appear against other backgrounds, but it may result in type-II errors for other objects being photographed against bathroom backgrounds (K. Xiao, Engstrom, Ilyas, & Madry, 2020).

More so than the other explanation parameters, robustness to background variation is ad-

dressed in multiple ways in existing work. On ImageNet images (Deng et al., 2009), K. Xiao et al. (2020) propose a method for separating the foreground and background signal using various methods of masking and obscuring the image subject. With this approach, they demonstrate that models are prone to misclassifying images based on the background, even when they classify the foreground of the images correctly. They additionally show that models achieve a non-trivial classification accuracy on ImageNet images based on the background of the images alone.

While K. Xiao et al. find no significant performance differences when substituting ImageNet backgrounds between classes, other existing research finds that background randomisation in synthetic images *is* an effective method of achieving background invariance (Horn & Houben, 2020; Tobin et al., 2017). To this end, Horn and Houben (2020) propose a method for automated compositing of subjects onto novel backgrounds. They demonstrate that substituting underrepresented object classes onto backgrounds from other images is an effective method of generating balanced training data, and consequently show a significant performance increase on the *German Traffic Sign Recognition Benchmark* classification task (Stallkamp, Schlipsing, Salmen, & Igel, 2011).

In the domain of synthetic imagery, Tobin et al. (2017) demonstrate that robustness to background changes can be achieved with *domain randomisation*. Their research suggests that training models on synthetic data with significant variation in image backgrounds may cause the real world to "appear to the model as just another variation", allowing for effective transfer to real images with a reduced need for photorealistic synthetic data.

The approach taken in this research, presented in Chapter 3, relies on synthesising training data with diverse backgrounds via image compositing. This combines elements from all the above works (Horn & Houben, 2020; Tobin et al., 2017; K. Xiao et al., 2020).

2.2.3 Invariance to Lighting (Illuminant) Direction

While robustness to lighting direction is not a primary focus of any research uncovered in this review, unusual lighting conditions are known to cause significant changes in the appearance of objects, which can result in incorrect classifications by computer vision algorithms (Barbu et al., 2019). To address this, existing research with synthetic imagery, such as that by Tobin et al. (2017) and Mitash, Bekris, and Boularias (2017) varies lighting direction as a method of domain randomisation. While this is done to improve the transfer of the models from the synthetic to the real-world domain, the specific effects of different lighting directions on the performance of the models are not evaluated. This said, it is recognised that randomised lighting conditions have a positive effect on the robustness of trained models (Mitash et al., 2017).

Looking into other computer vision applications, there is some existing work which aims to estimate the illuminant direction of individual images for the purposes of forensics and computer graphics (Gardner et al., 2017; Johnson & Farid, 2005). While this work does not

provide insight into the effects of these conditions on vision models, these applications do motivate the inclusion of lighting direction as a valuable output of the model produced for Objective 3 (Model Training).

Overall, due to limited existing research on the effects of illuminant direction, this project is expected to make a significant contribution by furthering what is known about the effects of lighting conditions on classification performance.

2.3 Explainable AI

Explainable AI (XAI) systems are AI systems that humans maintain *intellectual oversight* of. This generally means that the decisions or predictions made by the model are understandable or interpretable by humans, which cannot be said of many state-of-the-art computer vision models. These properties are desirable, as explanations promote trust, accountability, and transparency, which allow for more responsible deployment of AI systems. With this being said, *uninterpretable* machine learning models have been deployed in real-world contexts at an increasing rate over recent years, highlighting the importance of XAI research.

As discussed in Section 2.1, the state-of-the-art in computer vision is dominated by deep learning models. While there is a considerable body of existing research that attempts to provide interpretability for these models (summarised in Section 2.3.2) humans are far from having a complete understanding of their decision making processes. This is a primary motivation for focusing on explainability when developing a model for Objective 3 (Model Training). As such, Section 2.3.1 introduces the various ways that XAI approaches are categorised, and Section 2.3.2 provides an overview of existing techniques for explaining computer vision models, identifying a gap to be filled by this research.

2.3.1 Classifying Approaches to Explainability

Commonly encountered explainability methods can first be categorised according to when the explanation is generated relative to the model output (Du, Liu, & Hu, 2019):

Intrinsic interpretability is achieved when a model is constructed in a way that is inherently understandable. This includes models such as decision trees and linear models, where the logic performed by the algorithm can easily be traced by a human.

Although some argue that intrinsically interpretable models may sacrifice performance for accurate, understandable explanations (Du et al., 2019), researchers including Rudin (2019) debate this claim.

Post-hoc interpretability requires the construction of a second model to provide explanations for a model that is inherently uninterpretable. This approach is often applied to deep learning models so that explanations can be achieved without sacrificing performance (Du et al., 2019; Vale, El-Sharif, & Ali, 2022).

Both intrinsic and post-hoc explanation techniques have been applied to computer vision models. Intrinsic techniques tend to impose constraints on certain layers of the network to coerce models into producing interpretable outputs (e.g. Alvarez Melis & Jaakkola, 2018; Sabour, Frosst, & Hinton, 2017; Q. Zhang, Wu, & Zhu, 2018). Post-hoc approaches vary widely in implementation, and multiple approaches are summarised in Section 2.3.2 below. Both approaches are considered relevant to Objective 3 (Model Training) of this research.

Beyond this, explainability methods can be further categorised based on whether they provide explanations at the level of individual instances, or look at the model and its parameters as a whole (Du et al., 2019). In this case:

Local explanations examine the result for individual instances, and help to understand the causal relationship between the input to the model and the corresponding prediction.

Global explanations inspect the structures and parameters present in the model, facilitating an understanding of its inner mechanisms.

Previous work looking at global explainability for neural network models has focused on training intrinsically explainable models to mimic the behaviour of the network. These include classification and regression trees, interpretable tree ensembles, and rule-based algorithms (Arbatli & Akin, 1997; Hara & Hayashi, 2016; Loh, 2011). In the context of this project, however, *local explanations* are considered most relevant, since understanding the inner workings of state-of-the-art image classification models to the level that is required for global explanation is unrealistic. Training interpretable models to mimic their behaviour is also considered outside the scope of this research.

Finally, explanation techniques are often categorised as model-specific or model-agnostic. These distinctions are relevant when considering the applicability, versatility, and complexity of the explanation technique, and can be summarised as follows:

Model-specific explanation techniques are designed around a specific model or class of models. These techniques utilise the internal structures, mechanisms, and parameters of the models they are designed for, and therefore provide insights that are specifically tailored to the architecture of the model.

Model-agnostic techniques, on the other hand, do not access the internal states of the model. Instead, these methods are applicable to all models that share the necessary input-output interface (e.g. image classification models, image-to-text models).

While model-specific techniques often yield more detailed insights due to their access to the internal states of a model, their applicability is limited to the models they are designed for, and they are often not easily adapted to other model architectures. Conversely, model-agnostic approaches are broadly applicable but may provide less detailed and specific explanations when compared to model-specific techniques.

In the context of this project, both approaches are considered relevant, and various existing methods for generating explanations of computer vision models are therefore summarised in

the following section. These techniques span many of the categorisations presented above, showcasing the landscape of existing explanation methods and demonstrating the potential for the development of a novel explanation technique in this research.

2.3.2 Explanation Techniques for Computer Vision Models

In this section, a summary of existing explanation techniques for computer vision models is presented. From this summary, it will be seen that many existing explanation methods focus on visualising the important regions in an input image. These techniques are summarised in Section 2.3.2.1. Various other methods, including alternative image-based methods and text-based methods are summarised in Section 2.3.2.2.

2.3.2.1 Saliency Mapping Techniques

As an explanation technique, saliency maps visualise the regions in an image that contribute most significantly to the output of the computer vision model. While saliency maps have been used since at least 1998 to visualise the regions that *humans* attend to in images (Itti et al., 1998), they were first applied as a form of post-hoc, local explanation by Simonyan, Vedaldi, and Zisserman (2013). Since this introduction, various techniques for saliency mapping have been proposed. These iterations and improvements are summarised in the following paragraphs, and an example saliency map from Zhou, Khosla, Lapedriza, Oliva, and Torralba (2016) is shown in Figure 2.2.

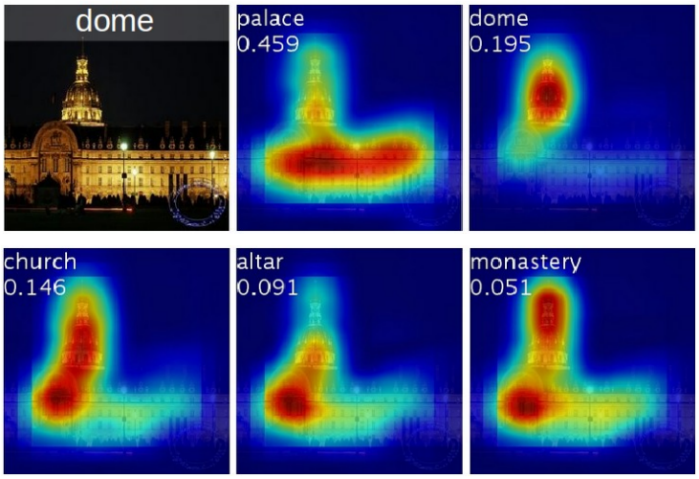


Figure 2.2: An example saliency map produced with Class Activation Mapping (CAM). The visualisation shows the regions of the input image that contribute most to producing each of the top five classification outputs. *Image from Zhou et al. (2016).*

Gradient-based Methods: Gradient-based methods use the gradients of the the model's output with respect to the input image to identify the most salient regions. Gradient-weighted Class Activation Mapping (Grad-CAM) is a popular technique in this category that can be applied to a range of CNN-based models (Selvaraju et al., 2017). However, gradient-based techniques are not entirely model agnostic as they rely on interrogating the gradients of

the neural network model. This is not applicable in all cases. Other popular techniques in the gradient-based classification include Layer-wise Relevance Propagation (LRP) (Bach et al., 2015), CAM (Zhou et al., 2016, see Figure 2.2), and Integrated Gradients (IG) (Sundararajan, Taly, & Yan, 2017) (see also Simonyan et al., 2013).

While gradient-based methods can be simple and effective tools for generating visual explanations, many techniques in this category are limited in their ability to explain the complex relationships between input features and model predictions. The value provided by many early gradient-based techniques has also been questioned in more recent work (Sundararajan et al., 2017).

Perturbation-based Methods: Perturbation-based methods provide explanations for an input X by producing alternative inputs X_i' with small perturbations (such as occlusions in the case of computer vision models) to the input features. These input variants are passed into the model, and the change in the output space is observed to evaluate which perturbations result in the most significant differences. When applied to computer vision models, this provides insights into the salient regions of the input image while remaining model-agnostic, as the technique relies only on the input and output format of the model.

Local Interpretable Model-agnostic Explanations (LIME) (Ribeiro et al., 2016) is a popular perturbation-based explanation method, which fits an interpretable model to explain the effect of the perturbations by locally approximating the result of the original model around the specific instance. Other perturbation-based methods include Shapely Additive Explanations (SHAP) (Lundberg & Lee, 2017) and Occlusion Sensitivity (Zeiler & Fergus, 2014, pp. 5–7).

While perturbation-based explanations are a popular technique since they are model-agnostic (therefore requiring no modification to the model), these methods can be relatively unstable due to their reliance on random perturbations. Generating explanations can also be expensive since it relies on generating multiple perturbed inputs, passing these through the model, and finally interpreting the results (Du et al., 2019; Ribeiro et al., 2016).

Attention-based Methods: Attention-based methods can also be used to produce saliency maps. These methods usually rely on attention mechanisms that are built into the model, as is done by Jetley, Lord, Lee, and Torr (2018). More recently, attention mechanisms are included in models like *ViT* that implement the transformer architecture (Dosovitskiy et al., 2020; Z. Liu et al., 2022, 2021; Mehta & Rastegari, 2021). The transformer modules present in *ViT* were used by Playout, Duval, Boucher, and Cheriet (2022), who generated high resolution saliency maps with their proposed technique *Focused Attention*. They implement multiple existing methods of saliency mapping, including Attention Rollout (Abnar & Zuidema, 2020), LRP (Bach et al., 2015), and their own Focused Attention approach (Playout et al., 2022), and demonstrate the superior interpretability of Vision Transformers with Focused Attention in a survey of domain experts.

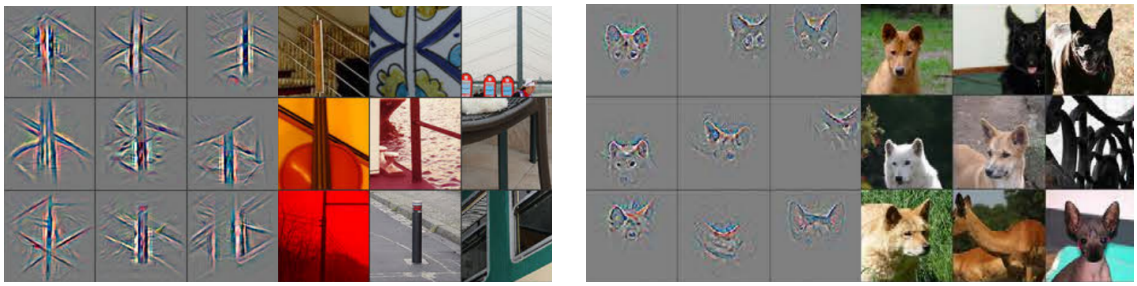
Attention-based techniques *can* provide valuable insights, but while Playout et al. (2022)

demonstrate that the resulting heatmaps can be of a higher quality than those output by other techniques, the validity of attention based explanations has been questioned by researchers including Jain and Wallace (2019). Their research finds that learned attention weights are often uncorrelated with gradient-based measures of feature importance, concluding that it is not yet clear what relationship exists between attention weights and model outputs, and that attention modules should not be interpreted as providing meaningful explanations of a model's decision-making process.

2.3.2.2 Alternative Explanation Techniques

While the aforementioned techniques are mainly used for saliency mapping, alternative explanation approaches have been implemented which provide interpretability in other ways. Some such methods are summarised in the remainder of this section.

Visualising Hidden Layer Activations: While saliency mapping techniques mentioned above produce explanations by visualising the salient regions in an input image, it is also possible to visualise the *features* that are learned by certain hidden layers of computer vision models. To this end, Zeiler and Fergus (2014) introduce a model-specific technique using *Deconvolutional Networks* that projects activations from hidden layers back into pixel space in order to visualise the regions of an input image that are causing activation of a specific layer. An example of their feature visualisations are shown in Figure 2.3, demonstrating an alternative method for visually explaining computer vision models.



(a) Layer 3. (b) Layer 5.

Figure 2.3: Deconvolutional network feature visualisations. Each figure shows one randomly sampled feature map from the respective layer of their trained model. The right half of each image shows the nine input image patches that caused the highest activation of that feature, and the left half visualises the model's activation using their deconvolutional network approach. *Images from Zeiler and Fergus (2014).*

Example-based Methods: Example-based explanation techniques aim to explain a model's output by making reference to exemplar instances. In the field of computer vision, multiple types of example-based explanation are used. Firstly, *counterfactual examples* provide an explanation of a classification output by producing slightly modified images that result in different classification outputs. Goyal et al. (2019) achieve this by modifying images with patches of images from other classes until a different classification result is produced.

Other researchers use alternative methods such as generative models to produce counterfactual examples (C.-H. Chang, Creager, Goldenberg, & Duvenaud, 2018).

Prototype examples are an alternative method of example-based explanation, in which a classification output is explained by drawing similarities to prototypical examples of the predicted class. C. Chen et al. (2019) implement a prototype-based explanation technique which identifies specific regions in input images, and links these image segments directly to segments of images of the predicted class. Compared to other techniques, example-based methods can be relatively inexpensive to produce and easy to interpret without technical knowledge, however these methods have also been criticised for their inability to illuminate the inner workings of black-box systems.

Text-based Descriptions: While the aforementioned techniques all produce image-based explanations, it is also possible to provide interpretations of model processes using text-based descriptions. These are natural language justifications that explain a model output by *describing* the image features that contributed to the prediction. Hendricks et al. (2016) implement test-based explanation by mapping images directly to textual descriptions using an image-to-text deep learning model. With this model they find no need to explicitly identify features in an intermediate stage. In 2018, they extend on this work by explicitly linking the objects or features referenced in the textual description to regions of the image (Hendricks et al., 2018).

While these methods require training data with detailed annotations, ground-truth labels *are* available for the synthetic data set produced in this research. As such, the explanation technique introduced in Section 5.1.2 follows a similar methodology that is extended to produce a novel image-based explanation technique.

Concept-based Explanations: Concept-based Explanations recognise that computer vision models leverage complex features based on pixel values that are not easily understood by humans. Instead of attempting to interpret these features directly, these techniques view the state of the model as a vector space E_m, and define another vector space E_h which is spanned by human-understandable *concepts*. E_h may be defined using image prototypes that embody *concepts* (i.e. 'smooth', 'stripy', etc.) The explanation is then generated by learning a function $g : E_m \rightarrow E_h$ that maps the representation of the model's state to the human-understandable concept-space (Kim et al., 2018).

Various concept-based explanation techniques have been proposed and applied to computer vision models. Testing with Concept Activation Vectors (TCAV) (Kim et al., 2018) is one such technique that provides explanations in an arbitrary, user defined concept-space. TCAV achieves this by using images that embody each concept to learn Concept Activation Vectors (CAVs) that represent the model's response to the concept. These CAVs are then used to deduce the concepts that the model is responding to in an input image.

Network Dissection (Bau, Zhou, Khosla, Oliva, & Torralba, 2017) is another concept-based

explanation technique in the computer vision domain, which learns a *predefined* set of concepts using a broadly and densely labelled training data set. Another method, introduced by Q. Zhang, Yang, Ma, and Wu (2019) uses a customised loss function to encourage learning of specific features (e.g. 'beak' and 'wing' for bird classification), and explains the learned features by associating them with image features using a decision tree model.

2.3.2.3 Summary of Explanation Techniques

Based on the above overview of XAI techniques in computer vision, it is clear that significant research effort has gone into producing visualisations of both salient image regions and intermediate feature representations of computer vision models. It is additionally clear that other explanation techniques are valuable in a variety of applications.

Since explainability is a focus of this research, various methods of providing explanations are used throughout the project. In the evaluation of existing models conducted for Objective 2 (Existing Model Evaluation) and presented in Section 4.2, synthetic *counterfactual examples* are used to showcase failure conditions of existing classification models. In addition to this, a novel image-based explanation technique is proposed in service of Objective 3 (Model Training). This technique, which draws from existing image-, example-, and concept-based methods, is presented in Section 5.1.2.

2.4 Synthetic Image Generation

Synthetic image generation is the process of creating artificial images, often using computer graphics, simulated environments, or generative models. Synthetic imagery is generally utilised where large-scale collection or annotation of data is expensive or impossible. This is true in this project, where training a model to be robust to variation in the explanation parameters requires a data set annotated with these features.

Such a data set of real images has not been published at the time of writing, and would be both expensive and challenging to annotate. As such, synthetic imagery is considered appropriate for both evaluating the robustness of existing models for Objective 2 (Existing Model Evaluation), and as a component of the training data for a new model produced to achieve Objective 3 (Model Training).

2.4.1 Synthetic Image Quality

To create a high quality synthetic data set in this research, the following properties identified by Man and Chahl (2022) will be considered. The methodology proposed in Section 3.1 was selected based on it's ability to best satisfy these criteria:

Domain Gap: The domain gap is the difference between the distribution of real-world and synthetic images. These differences include the presence of artefacts, or the absence of certain features that are found in real images (Nikolenko, 2021). A large domain gap often

results in significant performance decreases when transferring algorithms to real data, since the model may learn features or patterns that are specific to the synthetic domain. To reduce the domain gap, properties like *photorealism* and *data diversity* should be increased in the synthetic data set (Sun & Saenko, 2014; Tsirikoglou, Eilertsen, & Unger, 2020), and *bias* should be decreased (Man & Chahl, 2022).

In this context, photorealism is a property of images that contain a high level of detail that is representative of real-world images. Realistic textures, lighting, shadows, and reflections all contribute to improving photorealism and make photorealistic renders challenging to distinguish from real images. As a result of this, said images are ideal for evaluating and training computer vision models as there is a relative absence of synthetic artefacts that may influence the performance of the model. While photorealism has been suggested as an effective method for improving synthetic data by reducing the domain gap (Tsirikoglou et al., 2020), other research suggests that photorealism is unnecessary for algorithms to successfully translate to the real world (Sun & Saenko, 2014; Tobin et al., 2017).

While the *visual* properties required to achieve photorealistic images are crucial for minimising the domain gap, it is similarly important to consider larger distributional differences between real and synthetic data sets. One such aspect is the diversity of the data set. Data diversity refers to the variety of different scenes, objects, textures, lighting conditions, poses, and other features present in the synthetic data. Real image data sets such as ImageNet (Deng et al., 2009) contain a high degree of variation in each of these features, and ensuring similarly high diversity in synthetic data is essential for training robust models that generalise effectively to new, unseen information (Man & Chahl, 2022).

Related to the property of diversity, another key feature that contributes to an increased domain gap is *bias*. Biases are systematic patterns in synthetic images that deviate from the real-world distribution. Bias may arise as a result of imbalanced classes or features in synthetic data sets, and is known to result in poor performance in situations that were poorly represented in training data (Man & Chahl, 2022).

The methodology for rendering the synthetic data set in this research is presented in Section 3.1. The approach is designed specifically to reduce the domain gap while including deviations from real-world data sets that contribute to the objectives of this research. To balance effectively between photorealism, diversity, and bias, input data sets to the synthesis pipeline are compared and selected in Section 3.1.6. Following the image synthesis, the distribution of various parameters in the synthetic data sets are validated and inspected for diversity and bias in Section 3.2. Finally, existing research by Man and Chahl (2022) and Anderson, Ziolkowski, Kennedy, and Apon (2022) shows that models trained on synthetic data transfer significantly better to the real-world domain if they are fine-tuned on real images. As such, the training methodology proposed in Section 5.1.4 incorporates fine-tuning on ImageNet (Deng et al., 2009).

Computational Resource Requirements: The resource requirements for an image synthesis pipeline should also be considered, as rendering complex scenes can be computationally expensive. A pipeline that is too complex may limit the amount or resolution of images that can be generated, which raises issues since larger data sets have greater potential for diversity. Broadly, photorealistic images are more computationally demanding since simulating accurate lighting, materials, and interactions between the two is a resource-intensive process (Movshovitz-Attias, Kanade, & Sheikh, 2016). Computational resource requirements vary significantly between synthesis techniques, and as such, the complexity of each approach is discussed when looking at specific approaches in Sections 2.4.2.1–2.4.2.3.

2.4.2 Image Synthesis Techniques

The following sections investigate the various options available for generating synthetic images, ranging from manual synthesis in 3D rendering engines, to synthesis via generative models. The suitability of each approach for the objectives of this research is evaluated according to the metrics presented in Section 2.4.1. Ultimately, a hybrid approach between manual synthesis and composite imagery is be shown to be appropriate for this research.

2.4.2.1 Manual Image Synthesis

Manual image synthesis is the process of producing 3D environments and capturing synthetic images from within them. These environments can vary from small scale *scenes*, which may only look realistic from a single perspective, to *worlds* which are large and diverse environments that can be used to capture a variety of images (Man & Chahl, 2022). Many 3D video games provide good examples of virtual worlds, and there is a significant body of research using synthetic imagery captured from games such as *Grand Theft Auto V* (Johnson-Roberson et al., 2016; Richter, Vineet, Roth, & Koltun, 2016). For this project however, capturing images from game environments is considered inappropriate due to the biased distribution of game assets, and the considerable effort required to interface with games in this way.

Returning to manual generation of novel scenes, existing work demonstrates that manual image synthesis can be done using a variety of 3D design tools, including Blender (e.g. Riegler, Urschler, Ruther, Bischof, & Stern, 2015; Ruiz, Fontinele, Perrone, Santos, & Oliveira, 2019) and Unreal Engine (e.g. Qiu & Yuille, 2016; Tremblay, To, & Birchfield, 2018). These software tools provide an accessible interface for producing 3D environments, and can be interfaced with programmatically, allowing image synthesis pipelines to be largely automated.

In the context of this research, there are a few advantages of manual synthesis that are worth considering. Primarily, creating scenes from 3D models and capturing them from within a modelling engine allows for detailed annotation to be integrated directly into the rendering process (Hinterstoisser et al., 2013; Ruiz et al., 2019; Tobin et al., 2017; Xu, Lin, Zhang, Wang, & Li, 2022). This means that properties such as the location, orientation, scale, and bounding box of objects can be annotated with perfect accuracy in the manually synthesised

images. Additionally, properties like lighting, illumination, and background can be precisely controlled, which is essential for accurate annotation.

On the contrary, there are some disadvantages of manual image synthesis when compared with other techniques. One considerable disadvantage is that 3D models, backgrounds, and other objects must be sourced, and that the quality of the resulting images (especially the properties of photorealism and diversity) is directly dependent on the quality and diversity of these assets. Some high quality and diverse data sets are identified in Section 3.1.6.

Assuming access to suitable sets of objects and backgrounds, manual image synthesis is a promising avenue with the ability to produce photorealistic images, with a high level of data diversity, and low bias. The computational resource requirements for manual synthesis vary significantly depending on the complexity of images and rendered scenes, however simple scenes can be rendered on consumer-level hardware at HD resolution in under a second.

2.4.2.2 Synthetic Composite Imagery

Composition is a simple method of image synthesis that works by taking a 2D image of a subject and overlaying it on a background image (Man & Chahl, 2022). Other elements of a scene may also be overlaid, such as rain or fog (Sakaridis, Dai, & Van Gool, 2018). This is computationally inexpensive, as no 3D rendering is required, however the lack of interaction between the subject and the background reduces the potential for photorealism.

In existing research by Horn and Houben (2020), synthetic composite images were used to create a large, diverse, and balanced data set from a small and imbalanced one. This demonstrates the potential to achieve extremely high diversity by compositing a subject onto numerous backgrounds, which can be easily scaled due to the inexpensive nature of the compositing process. In addition to this, research by Tobin et al. (2017) suggests that randomising backgrounds in a similar way reduces the requirement for photorealistic images. The combination of high diversity, low resource requirements, and a reduced need for photorealism make image composition a valuable technique for use in this project.

2.4.2.3 Synthesis With Generative Models

More recently, Variational Autoencoders (VAEs) and Generative Adversarial Networks (GANs) have become widely used for producing synthetic images (e.g. Dhariwal & Nichol, 2021; Karras, Laine, & Aila, 2019; Ramesh, Dhariwal, Nichol, Chu, & Chen, 2022; Ramesh et al., 2021; Rombach, Blattmann, Lorenz, Esser, & Ommer, 2022; Saharia et al., 2022; J. Yu et al., 2022). The most widely used implementations of these models perform text-to-image synthesis, where a text-based prompt is provided and a corresponding image is produced conditional on the provided prompt. These text-to-image models consist of an encoder network that learns to produce feature representations from the input text, and a generator model that maps this latent representation to an output image.

Implementations of these models, including Stable Diffusion (Rombach et al., 2022) and

DALL-E (Ramesh et al., 2022, 2021) are capable of producing photorealistic images with extreme diversity. However, there are critical limitations of these models that reduce their suitability for this project. The primary limitation is that the encoder-decoder architecture makes no guarantees that the output image adheres to the specifications of the input prompt. This problem is known as image-text alignment, and is identified by Frolov, Hinz, Raue, Hees, and Dengel (2021) as a key challenge for text-to-image models, especially as prompts increase in complexity and begin to describe scenes with multiple objects.

Poor image-text alignment is a significant barrier for using generative models to synthesise complex data sets because these data sets require accurate labelling. To produce a data set containing images of objects in challenging poses, contexts, and lighting conditions for Objective 1 (Image Synthesis), it would be necessary to provide a generative model with an appropriate prompt *and* to validate that each produced image aligned with the provided prompt. Each image would be required to meet a strict set of requirements governing the the class and pose of the image subject, the image background, and the lighting conditions. These demands are considered unrealistic for current generative models, and validating individual output images is considered infeasible in the time frame of this project.

In addition to this, the specific requirement to represent poses, backgrounds, and lighting directions that are not frequently observed in real-world images is a further reason to believe that generative models may be unable to produce high quality data for this research. This is because computer vision models are known to work most reliably within the distribution of their training data, and cannot be guaranteed to generalise to cases outside of this distribution. Since the purpose of the synthetic data set is to represent poses that are infrequently captured, it is considered inappropriate to rely on generative models to synthesise this data. As such, generative synthesis was not used. It should, however, be reconsidered in future research as the technology continues to develop.

2.5 Multi-task Learning

Multi-Task Learning (MTL) is a learning paradigm where a single model is trained to perform multiple tasks simultaneously. This allows the model to leverage complex latent representations to solve multiple tasks. MTL is covered in this literature review because Objective 3 (Model Training) involves modifying an existing model such that it produces additional explanatory outputs (i.e. completes *multiple tasks*). The resulting model should:

(a) Maintain similar image classification performance to the model being adapted; and

(b) Produce a supplementary output for each of the explanation parameters.

2.5.1 Multi-task Learning Paradigms

While it is clear that MTL is relevant to this project based on the multitude of model outputs, it is worth investigating the various MTL paradigms in order to select the most relevant and

effective one for use in this research. A survey by Y. Zhang and Yang (2021) identifies various approaches that are widely used in existing research. Of these approaches, only some are applicable to deep learning models, and of those, *Feature Learning* is considered most relevant to this research.

While alternative approaches, including *Task Clustering* (Bakker & Heskes, 2003) and *Task Relation Learning* (Long, Cao, Wang, & Yu, 2017) have been applied to neural network models, these approaches add significant complexity and are not expected to provide sufficient advantages to justify deviating from the successful and widely used *Feature Learning* approach. In the remainder of this section, the feature learning approach is discussed in more detail to justify the use of this paradigm in service of Objective 3 (Model Training).

Feature Learning: Feature learning is a popular MTL paradigm which assumes that multiple outputs of the model can leverage a shared feature representation to produce their predictions. This is intuitive for tasks that are closely related, as latent features learned in service of one task may be similarly relevant for solving related tasks. In addition to this, requiring that the model completes additional related tasks may cause it to learn features that would not have otherwise been learned by a single-task model, but that are nonetheless valuable for increasing single-task performance. While all implementations of feature learning involve a shared feature representation, Y. Zhang and Yang (2021) further divide this approach into two distinct sub-categories, the *Feature Transformation* approach, and the *Feature Selection* approach.

The *Feature Transformation* paradigm exists when models produce a shared feature representation, then the various outputs of the model are produced by taking linear or non-linear transformations of this latent representation. This approach is most commonly found in neural network models, which learn latent representations in their hidden layers that are transformed by subsequent output layers to produce the model's predictions (e.g. S. Li, Liu, & Chan, 2014; W. Liu, Mei, Zhang, Che, & Luo, 2015; W. Zhang et al., 2015; Z. Zhang, Luo, Loy, & Tang, 2014).

The *Feature Selection* approach involves a similar shared feature representation, however, instead of producing outputs using arbitrary transformations of the latent representation, regularisation techniques are used to select specific features that are relevant to each task. An advantage of this approach is that it is more interpretable since only a subset of features are used for any one task. This approach is, however, not widely used in conjunction with deep learning models, and is considered inappropriate for this research due to the fact that current state-of-the-art image classification models are all neural network based.

In summary, *Feature Transformation Learning* is the most relevant MTL paradigm for this research due to its widespread application to deep learning models. The primary reasons for this are that implementing this approach is straightforward and places no restrictions on the features that are used by the model. The specific architecture and training process used to implement this MTL approach are discussed further in Section 5.1.

2.6 Key Insights from the Literature

Before approaching the objectives of this project in the following chapters, this section distils the core findings of the literature review, and summarises their implications for this research. In Section 2.1, an overview of the history and current state of image classification was provided. Resulting from a survey of high-performing and historically significant models with diverse architectures, four models were selected for analysis in Objective 2 (Existing Model Evaluation). These models are (1) *MobileNet V2* (Sandler et al., 2018), (2) *ResNet-152* (K. He et al., 2016), (3) *MobileViT V2* (Mehta & Rastegari, 2022), and (4) the *Swin Transformer V2 (Large)* (Z. Liu et al., 2022).

Following this, existing research into parameter-invariant computer vision was summarised in Section 2.2. This section concludes that, while some existing models have been evaluated on their invariance to pose and background variation, there is room to expand this evaluation to other models, architectures, and parameters. This research specifically includes the parameter of lighting direction as no existing research was identified in this area.

Next, when attempting to train the STRobE model for Objective 3 (Model Training), we aim to not only achieve invariance to the explanation parameters, but to also achieve *explainability*. As such, existing techniques for producing explainable computer vision models were summarised in Section 2.3. Based on gaps in the existing literature, we conclude that there is potential to implement a novel example-based explanation technique using synthetic images.

Since each of the evaluation, model training, and explainability objectives of this research depend on synthetic images, existing methods for image synthesis were summarised in Section 2.4. A hybrid approach between *manual synthesis* and *composite imagery* is found to be appropriate for the goals of this research. Using these techniques, a data set of synthetic images is produced in Chapter 3, which is used both to evaluate existing models in Chapter 4, and to train the STRobE model in Chapter 5.

Chapter 3

Generating a Synthetic Image Data Set

As discussed in Section 2.4, synthetic data sets are indispensable in scenarios where real-world data is either unavailable or challenging to procure due to factors like cost and complexity. In this research, synthetic images are employed primarily because no real-world data set exists that meets the specific requirements of the project. Since this research involves evaluating the robustness of existing models to variations in the explanation parameters, a data set is needed where these parameters are controlled and labelled for every image. Additionally, to enable comprehensive evaluations of existing models, and the training of a new model, this data set must represent a highly diverse range of image configurations.

While images captured in the real world and labelled accurately represent the gold standard of image classification data sets, creating a real image data set of a sufficient size and quality for this research is infeasible for multiple reasons. Primarily, capturing and accurately labelling images is labour-intensive, and therefore expensive. This is especially true in this project, where images must contain variation in object pose, image background, and lighting direction. Moreover, manually labelling these parameters for existing images is challenging and prone to errors.

Synthetic imagery provides an avenue for solving both of these problems, as the synthesis pipeline can be automated for large sets of objects and backgrounds and accurate labelling can be directly integrated into the synthesis pipeline. Additionally, manual synthesis allows for fine-grained control of the explanation parameters such that the data set is optimally distributed for evaluating existing models and training new ones. In this chapter, the image synthesis process is described in detail, including the data sets and technologies that were used in the process. Following this, the synthetic data set is validated, confirming that it is suitable for both evaluating and training image classification models.

3.1 Image Synthesis Methodology

In this section, an overview of the synthesis pipeline is provided, including a discussion of the software and data sets used as inputs, and the rendering parameters used throughout the process. At a high level, the synthesis pipeline has two stages: rendering the image subjects, and compositing the subjects onto backgrounds. These stages are examples of manual synthesis (Section 2.4.2.1) and composite imagery (Section 2.4.2.2) respectively. The decision to use these techniques, as well as other design decisions made when constructing the image processing pipeline, are justified in the following sections.

3.1.1 Software

As stated above, the hybrid image synthesis approach starts with a rendering phase, which begins by importing 3D models of image subjects into the computer graphics software *Blender* (Blender Online Community, 2018). In the context of this research, *Blender* is used for rendering 3D models under controlled variation in orientation and lighting conditions. *Blender* was selected for the rendering phase over other computer graphics software such as *Unreal Engine* or *Unity* due to its open-source nature and the completeness of its Python interface.

3.1.2 Image Size and Count

The size of the synthetic images is set at 224×224 pixels. This value was chosen as it is a popular input resolution for many computer vision models, and this relatively small size results in short synthesis and processing times. It may be valuable to produce higher resolution images in future research, but 224 pixels was considered appropriate for the objectives of this project.

To determine the overall number of images in the synthetic data sets, it is relevant to look at the selected data set of 3D models (ShapeNetCore, see Section 3.1.6). In ShapeNetCore, the number of included models varies significantly between classes. To compensate for this and achieve a balanced distribution of classes in the final synthetic image data set, a variable amount of images are captured of each 3D model. The desired number of images per object class was set at 15,000, which is expected to be a sufficient size for training a new model, while maintaining tractable synthesis and training times.

To ensure that at least 15,000 images are synthesised for each object class, the number of images captured of each 3D model is $n_i = \max\left(10, \left\lceil \frac{15{,}000}{m_i} \right\rceil\right)$, where i is an object class containing m_i 3D models. This means that, for the vast majority of classes that contain fewer than 1,500 models, $\frac{15{,}000}{m_i}$ images are captured per model. For the few classes with more than 1,500 models, a minimum of 10 images is captured per object to ensure that each object is still imaged in a diverse range of poses. The method for sampling optimally distributed poses for each object is discussed in the following section.

3.1.3 Pose Sampling

In the rendering phase, images are captured of each object under different poses and lighting conditions. In this context, the *pose* or *orientation* of an object is defined by its facing direction (comprising rotation along its local *yaw* and *pitch* axes) and rotation along its local *roll* axis, requiring three parameters to define. To demonstrate the impact that pose has on the appearance of a 3D object, Figure 3.1 shows sample renders of an *airplane* model representing facing directions across the pitch and yaw axes.

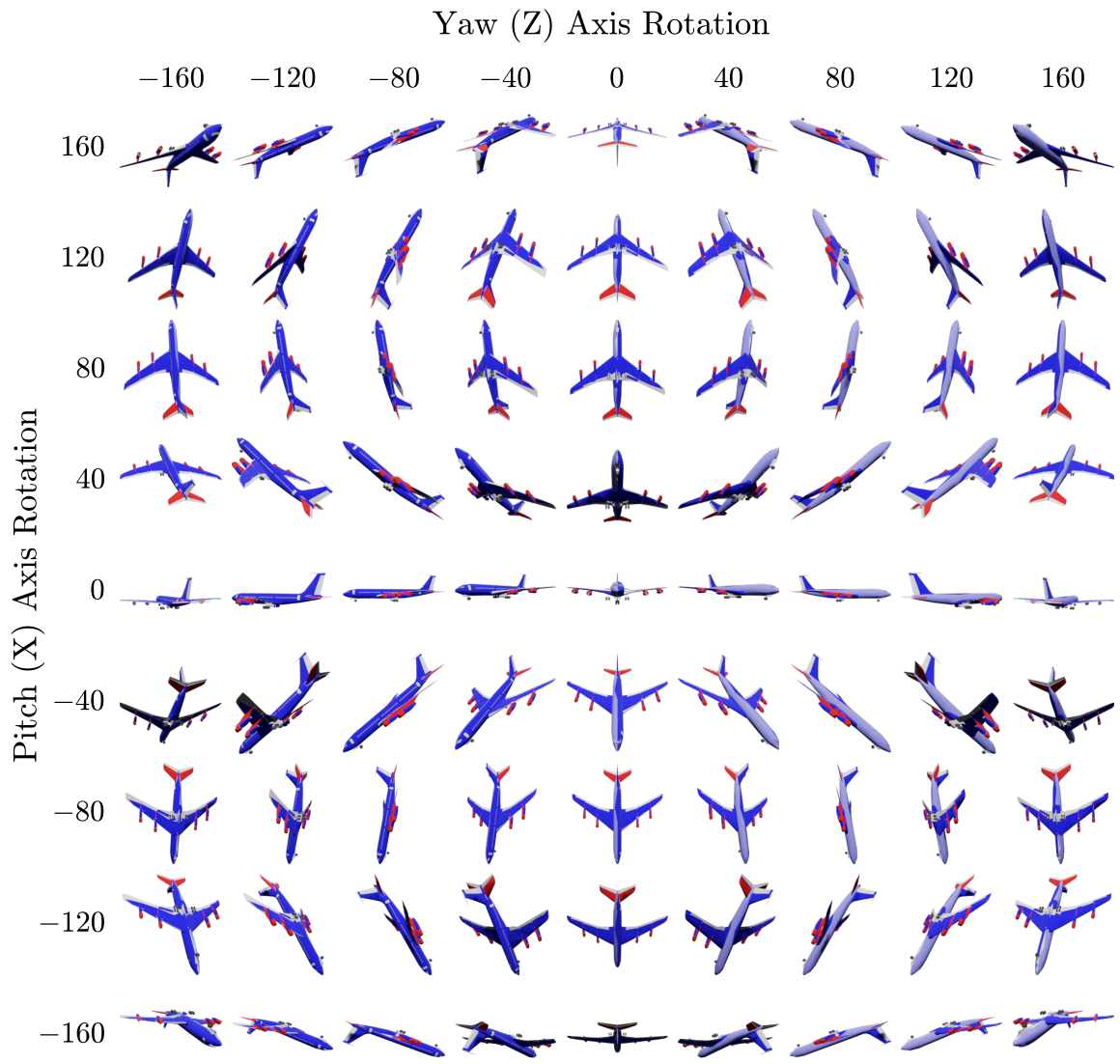


Figure 3.1: Rotation samples across the yaw and pitch axes, visualised in 3D space using an *Airplane* model from ShapeNetCore.

Including diverse poses like these in the synthetic data set facilitates evaluation on poses that are not frequently observed in real-image data sets. Additionally, the representation of infrequently observed poses is expected to provide value when training a new model for Objective 3 (Model Training). As such the distribution of poses in the data set is critically important. The remainder of this section therefore provides justification for the method

selected for sampling these poses. The various methods that were considered are evaluated against the following desirable properties:

Representation: The distribution of poses should be highly diverse, sampling a comprehensive range of facing directions and roll rotations. Ideally, in a system with perfect representation, any arbitrary pose could be sampled.

Uniformity: The system should not be biased towards selecting specific poses.[1]

Complexity: The pose should be represented in an intuitive format with minimal redundancy. This is to make predicting these poses as easy as possible for a model trained on the synthetic data set.

Three pose sampling approaches were considered for this project, which are denoted *Uniform Facing Angles*, *Uniform Axial Sampling*, and *Spherically-Distributed Sampling* in the sections below. In these sections, each approach is described and evaluated against the above criteria, providing justification for the use of the *Spherically-Distributed Sampling* approach.

Uniform Facing Angles: With this approach, the object's pose is defined by its facing direction using two parameters a (azimuth) and e (elevation). To produce at least n_i views of each object, $\lceil\sqrt{n_i}\rceil$ equally spaced rotation increments are sampled along the object's local Z (yaw/azimuth) and X (pitch/elevation) axes. To ensure that these poses are distributed uniformly along each axis of rotation, a random azimuth offset o_a is first sampled from $O \sim \text{Unif}\left(0, \frac{360}{\lceil\sqrt{n_i}\rceil}\right)$ and a random elevation offset o_e is also sampled from O. Using these offsets, the set of at least n_i poses $P : \mathcal{P}([0,360) \times [0,360))$ for object i would be generated according to:

$$P = \left\{ \left(\frac{360}{\lceil\sqrt{n_i}\rceil} x + o_a, \frac{360}{\lceil\sqrt{n_i}\rceil} y + o_e \right) \middle| (x,y) \in \{0,1,\ldots,\lceil\sqrt{n_i}\rceil\} \times \{0,1,\ldots,\lceil\sqrt{n_i}\rceil\} \right\}$$

Evaluating this approach against the criteria presented above:

Representation: The poses represented are somewhat diverse, including any possible facing direction. A limitation of this sampling approach is that objects are not rotated along their local *roll* axis. This means that each facing direction can only be sampled in an *'upright'*, or *'upside-down'* orientation (which occurs when the pitch rotation is between 90 and 270 degrees).

Uniformity: With this approach, the parameter values a and e are each uniformly distributed over the entire $[0°, 360°]$ range, as this is expected to reduce bias in any models trained on the data set (H. He & Garcia, 2009). This being said, the facing directions of the objects *are* more concentrated towards the poles on the vertical axis (shown in

[1] There is additional complexity associated with the property of uniformity, as different measures of uniformity compete when sampling poses. For example, sampling yaw and pitch rotations from independent uniform distributions causes polar bias in the facing directions.

Figure 3.2 and hereafter referred to as polar bias). Despite the apparent concentration of points around these poles, it is important to note that rotation along the yaw axis ensures that even when the facing angle is similar, the view of each object differs significantly between images.

Complexity: Simplicity is the primary advantage of this approach. While the alternative approaches, discussed next, offer more complete representation of poses *and* allow for more uniform sampling, those approaches require more variables to define the object's pose, and have more redundant configurations.

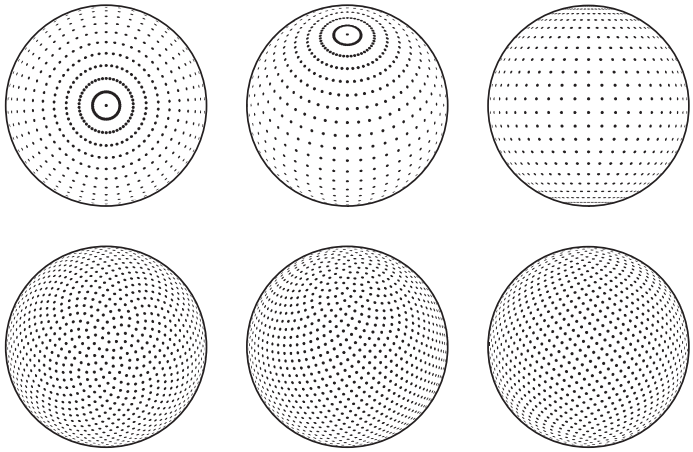


Figure 3.2: Uniform Axial (a.k.a. azimuth-elevation) sampling (top) and Spherically-Distributed (a.k.a Fibonacci lattice) sampling (bottom). Orthographic projections, centred at the pole (left), 45° elevation (middle) and equator (right). *Image from González (2010).*

Uniform Axial Sampling: This sampling method extends the aforementioned *Uniform Facing Angles* approach by adding a third parameter to represent rotation along the object's local *roll* (Y) axis. This can be added into the previous system while maintaining the same amount of images per object by sampling a random roll rotation r_r from $Y \sim \text{Unif}(0, 360)$ for each (r_a, r_e) pair. This simple extended approach evaluates as follows:

Representation: This approach is capable of representing any pose with its three parameters.

Uniformity: Again with this approach the parameter values are uniformly distributed over the entire $[0°, 360°]$ range. This approach suffers from the same polar bias as the previous approach, though this approach maintains a high diversity of yaw *and* roll configurations around the poles.

Complexity: While this approach adds only a single additional parameter to the system, the poses that can now be represented are significantly more complex, and there exist redundant configurations since certain poses can be achieved in multiple ways. Ultimately, the complexity of poses generated by this method was the primary reason that the system was not used for producing the final synthetic data set.

Spherically-Distributed Sampling: While the previous methods both suffer from polar bias, approaches such as Fibonacci lattice sampling (González, 2010) (Figure 3.2) achieve a more uniform distribution of facing directions across the surface of the sphere. With this approach, a large set of azimuth, elevation pairs is first generated using the Fibonacci lattice algorithm. This set $F = \{(r_{a,1}, r_{e,1}), \ldots, (r_{a,P}, r_{e,P})\}$, containing $P = 2S + 1$ facing directions (where $S \in \mathbb{N}$ and rotations are defined in radians) is generated using the Fibonacci lattice algorithm as follows:

$$F = \left\{ \left(2\pi j \phi^{-1}, \arcsin\left(\frac{2j}{2S+1} \right) \right) \,\middle|\, j \in \{-S, \ldots, S\} \right\}, \text{ where } \phi = \frac{1+\sqrt{5}}{2}$$

In this formula, the *golden ratio* (ϕ) is used when defining the rotation increments along the azimuth axis, as ϕ is the "most irrational number", which ensures that clumping of lattice points never occurs (González, 2010).

With facing directions generated using this approach, images are then synthesised by sampling n_i facing angles from F (without replacement) and using these to define the facing angle of the object in the rendered image. Optionally, a random roll rotation r_r can be sampled from a desired distribution and applied to the object. An evaluation of the properties of this approach is provided below:

Representation: In theory it is possible to sample any facing direction on the sphere with this approach, however in practice the set F is finite with cardinality $P = 2S + 1$. By selecting a large value of S (100,000 is used in this research) we ensure that the facing directions present in F are distributed densely and uniformly across the entire sphere. In practice, this makes it possible to sample all *visually distinct* facing directions during rendering.

The representation of roll rotations depends on if and how these are sampled for each facing angle. One method would be to sample a roll-axis rotation r_r from $X \sim \text{Unif}(0^c, 2\pi^c)$ for each image. Under this approach, any roll rotation could be represented making this method capable of producing arbitrary poses. In practice, sampling such diverse roll-axis rotations results in highly complex poses with the potential to parameterise certain rotations in multiple ways. Because of this, the approach implemented in this research is to randomly select a roll-axis rotation from $\{0^c, \pi^c\}$. This results in the object either being *'upright'*, or *'upside-down'*, resulting in an identical configuration space to the *Uniform Facing Angles* approach.

Uniformity: The primary advantage of this method is that it avoids polar bias by sampling fewer pitch (elevation) rotations near the vertical poles (see Figure 3.2). Looking at the Fibonacci lattice formula for the azimuth angles ($r_{a,j} = 2\pi j \phi^{-1}$) it is clear that these are linearly distributed over j, however this is not true of the elevation angles defined by $r_{e,j} = \arcsin\left(\frac{2j}{2S+1}\right)$, which are instead concentrated around 0 due to the arcsin transformation.

This transformation is responsible for the reduction in polar bias, but appears to also introduce a non-uniformly distributed parameter which may be harmful when using the data set for Objective 3 (Model Training) (H. He & Garcia, 2009). Fortunately the arcsin transformation can be reversed at training time by simply taking the sin of the pitch angles, resulting in a uniform distribution for model training. With this addition, the *Spherically-Distributed Sampling* approach maximises uniformity, having uniformly distributed facing angles *and* parameters.

Complexity: With this approach, the complexity depends on the roll rotations that are used. If roll rotations are sampled over the entire $[0^c, 2\pi^c)$ range, the representation of poses is identical to that of *Uniform Axial Sampling*. In this scenario, the poses with three degrees of freedom are highly complex and contain redundancy.

If instead, the roll-axis rotation is always either 0^c or π^c, the configuration space is identical to that of *Uniform Facing Angles*. As such these rotations can be defined without the need for an additional parameter since a roll rotation of π^c is equivalent to adding π^c to both the yaw and pitch-axis rotations before they are applied in that order. This method, capable of representing complex poses with only two parameters and no redundancy was ultimately selected for the synthesis process.

Applying Poses for Rendering: Ultimately, the *Spherically-Distributed Sampling* approach was used in the rendering process to generate the set $P = \{(r_a, r_e) \mid r_a, r_e \in [0^c, 2\pi^c)\}$ of n_i poses for each object. The implementation of this process is summarised in pseudocode in Algorithm 1. At a high level, each image is set up by resetting the object to its default position, then an (r_a, r_e) pair is taken from P and applied to the object, rotating it first around the Z (yaw/azimuth) axis by r_a^c, then around its local X (pitch/elevation) axis by r_e^c. Before capturing the image, a lighting configuration is initialised as described in the following section.

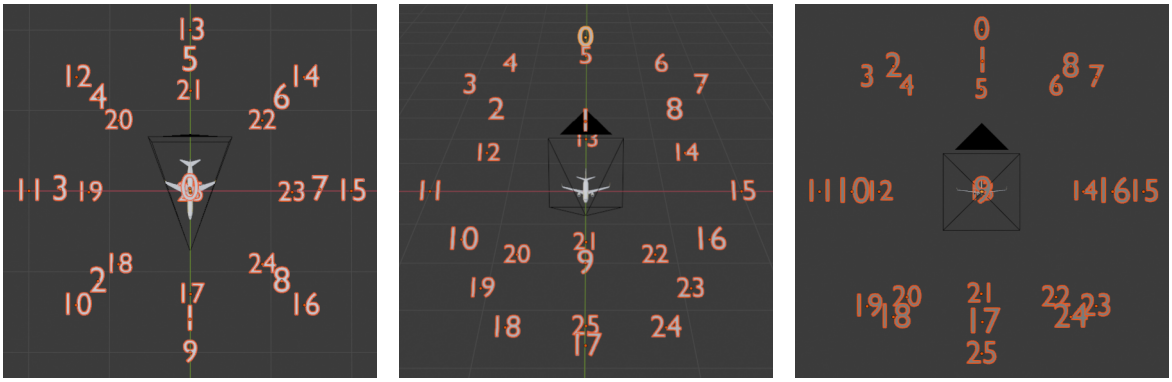
3.1.4 Lighting Configurations

The variation of lighting directions is a crucial aspect of the rendering phase. In real images, the direction of lighting can significantly impact the appearance of objects in an image. Therefore, to simulate a wide range of real-world conditions, and test the invariance of classification models to lighting directions, it is essential to vary lighting directions in the synthetic data set.

To best simulate a real-world lighting setup, two light sources are placed in the *Blender* scene. One light provides a base level of global illumination, and another directional light is positioned in one of 26 predetermined locations around the object. The set L, containing these 26 poses is illustrated in Figure 3.3 and defined as:

$$L = (A \times E) \cup \{(0, 90), (0, -90)\} \text{ where } A = \{-135, -90 \ldots, 90, 135, 180\}$$
$$\text{and } E = \{-45, 0, 45\}$$

After sampling a lighting configuration (l_a, l_e) from L, the directional light is positioned by adding a light source behind the camera, and rotating it around the origin (the centre of the 3D model) by l_a° along the azimuth/yaw axis, followed by l_e° along the elevation/pitch axis. A new (l_a, l_e) pair is sampled (with replacement) from L for each of the n_i poses of each object.



(a) Top view (b) 45° elevation front view (c) Front view

Figure 3.3: The 26 lighting configurations in the *Blender* scene. Note that these light positions can be grouped into sets such as *'Above'*, *'Below'*, *'Left'*, *'Right'*, etc. which enables novel evaluation of the robustness of image classifiers to these directional groups.

With the directional light placed in the scene, the camera is translated such that the object fills the frame, and a 224×224px image is rendered against a transparent background for each of the n_i scene configurations. These transparent images then proceed into the compositing phase of the synthesis pipeline to generate the final data set.

3.1.5 Compositing Process

In the compositing phase, backgrounds are added to the transparent object renders produced in the previous stage. To increase diversity in the final data set, the size of the image subject and its position on the background is varied between images.

To perform the compositing process for a given rendered image $\boldsymbol{i}$, a background class is first selected uniformly and at random from the possible classes in the background data set (this is the SUN397 data set, which is discussed further and justified in Section 3.1.6). A random background image is sampled uniformly and at random from this class, and scaled down such that a 224×224px square can be positioned in the image with no more than one degree of freedom. This can be seen in the central pane of Figure 3.4.

Following this, a 224×224px square image $\boldsymbol{b}_{\text{SUN}}$ is extracted from the scaled background. The rendered image $\boldsymbol{i}$ is then scaled to a random square size between 90 and 224 pixels, and the scaled $\boldsymbol{i}$ is composited onto the the background $\boldsymbol{b}_{\text{SUN}}$ at a location which is also selected uniformly and at random. The use of uniformly distributed parameters in the compositing stage is an intentional choice to optimise the data set for Objective 3 (Model Training).

In addition to the image on SUN backgrounds, a second composite image is produced by

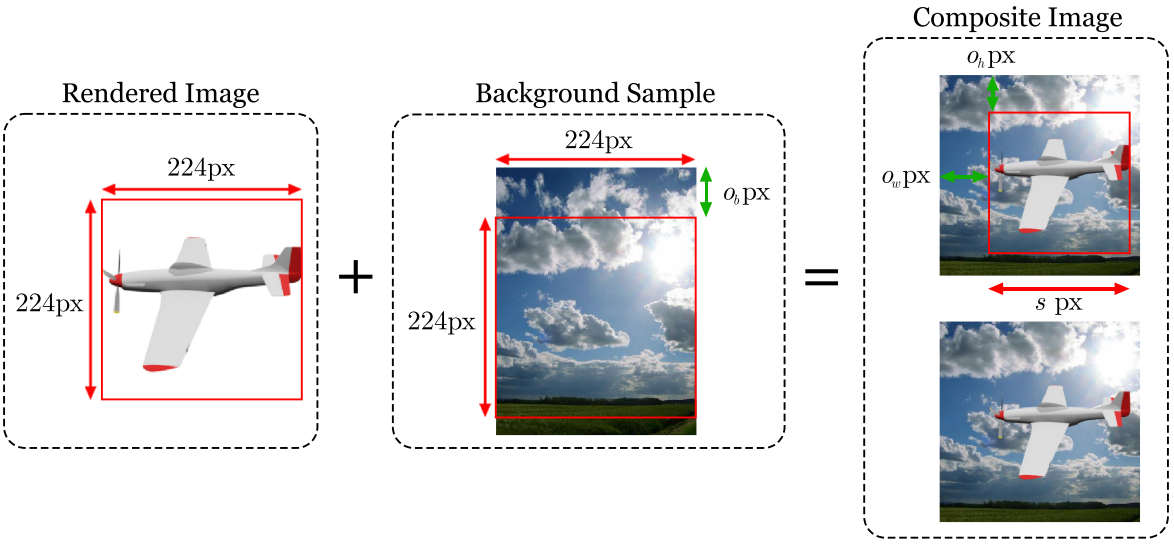


Figure 3.4: An illustration of the image compositing process, starting with a rendered object image, sampling a background, and compositing the two to produce the final synthetic image.

compositing $\boldsymbol{i}$ onto a plain white background $\boldsymbol{b}_{\text{white}}$ in an identical position. Again, this process is described algorithmically in Section 3.1.8, and the compositing process with $\boldsymbol{b}_{\text{SUN}}$ is shown in Figure 3.4.

3.1.6 Data Sets for Image Synthesis

The selection of appropriate input data sets is critical to the success of the image synthesis process. The data sets play a primary role in producing high quality synthetic images, and selecting high quality 3D models *and* backgrounds is important for reducing the domain gap. For this project, the ShapeNetCore (A. X. Chang et al., 2015) and SUN (J. Xiao, Hays, Ehinger, Oliva, & Torralba, 2010) data sets were chosen due to their specific characteristics that align with the research objectives.

3.1.6.1 3D Object Data Sets

To select the most appropriate data set of 3D models for this research, desirable properties of such a data set were first identified. Based on the existing research into synthetic image data sets (presented in Section 2.4), as well as the specific requirements of this research, it was determined that an ideal set of 3D models would contain:

1. High quality, photorealistic 3D models that appear realistic from all perspectives.
2. Multiple instances of each object class, with more being better.
3. High quality annotations of the 3D models, including a canonical pose for each object class, and the alignment of each object relative to this pose.
4. A relatively large amount of classes, ideally classes that overlap with those present in real image data sets.

ShapeNetCore: While no data set could be found that met all the above criteria, ShapeNetCore presented the best trade-off between these properties. ShapeNetCore is a data set of 3D Computer-Aided Design (CAD) models, containing 51,649 unique objects across 55 object categories. It is a subset of the larger ShapeNet data set in which all objects have been manually verified and aligned with the other objects in the same category (the default alignments can be seen in Figure 3.5).

This alignment is one of the most important properties identified above (item 3), and is a key advantage for this project as it enables the pose of the objects to be annotated automatically in synthesised images. Another advantage of ShapeNetCore is its large size, which contributes significantly to developing a synthetic data set with high-diversity. This results in a more robust evaluation of existing models, and provides more information to new models trained on the data set (Gong et al., 2019). Unfortunately many of the models in ShapeNetCore are not photorealistic and contain simple textures due to their purpose as CAD models. Additionally, of the 55 classes in the data set, only 48 of these overlap with ImageNet (Deng et al., 2009).

Google Scanned Objects: The Google Scanned Objects data set (Downs et al., 2022), which contains 1030 high-fidelity scans of household objects, was another candidate data set that was ultimately not used in the synthesis pipeline. While this data set provides high-fidelity scans of 3D objects, along with realistic textures (both of which are valuable for enhancing photorealism), there are a few reasons it was ultimately not used. Firstly, the objects in the data set are not aligned, which is a requirement for accurately labelling the orientation of objects in synthetic images. While manual alignment is possible, the second limitation of this data set is that there is very low diversity within object classes, as each ImageNet class represented in the data set presents at most a few instances. It was considered that the ShapeNetCore data set could be augmented with the addition of Scanned Objects where the classes between the data sets overlapped, however ultimately it was decided that this would not provide sufficient benefit to justify the manual effort required.

OmniObject3D: Another similar data set of 3D models produced by T. Wu et al. (2023) is OmniObject3D. This is another data set of high-fidelity 3D scans (similar to Google Scanned Objects), containing 6,000 scans across 190 object categories. While this now exists as a promising repository of photorealistic 3D models, the data set was released shortly after the commencement of this research and was therefore not included.

3.1.6.2 Background Data Sets

For the image backgrounds used during the compositing phase (see Sections 3.1.5 and 3.1.8), the MIT Scene Understanding (SUN) Database is used. Specifically, the SUN397 data set was selected as this 397-class subset of the complete SUN database contains a more tractable amount of classes, with at least 100 images per background class.

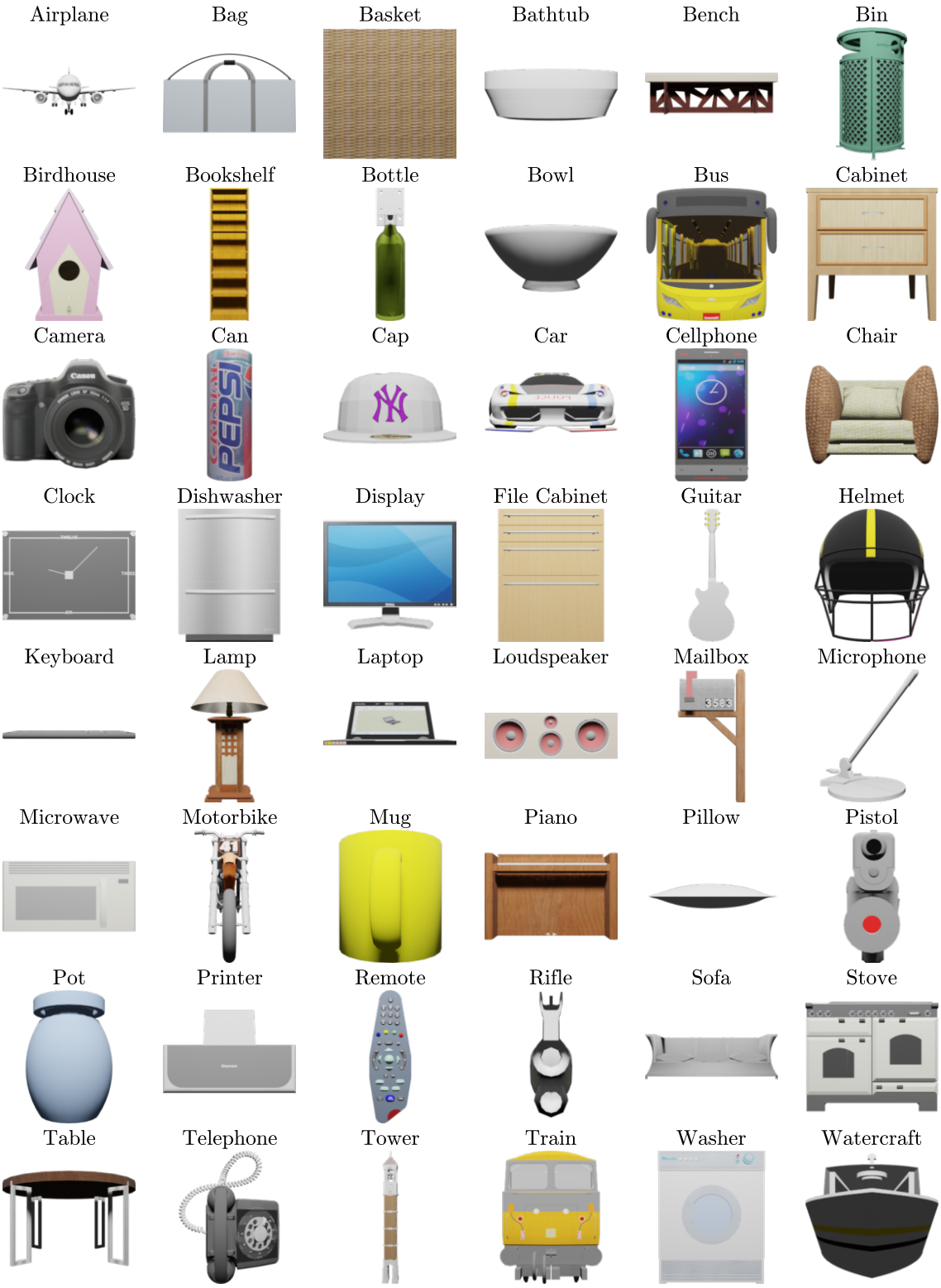


Figure 3.5: The default poses (Pitch and Yaw rotation of 0°) of the 48 ShapeNetCore objects included in the synthetic data sets (one object sampled per class).

In total, this data set contains over 108,000 images, spanning many types of indoor, outdoor, urban, and natural environments. In addition to this high diversity, the SUN data set is categorised hierarchically, allowing for labelling of specific background categories, as well as broader classes such as indoor and outdoor. These hierarchical labels provide advantages when evaluating models for Objective 2 (Existing Model Evaluation), as this allows for more nuanced evaluation of the features that existing models are responding to with their classifications. Additionally, granular labels provide benefits when training a new model for Objective 3 (Model Training) as the learning task can be made easier by choosing broader labels, or more informative by opting for more finely detailed outputs.

While it was believed that SUN397 met all the criteria of a high quality background data set at the time of selection, it was, in retrospect, not optimised for use off-the-shelf. The 397 classes present in the data set are simply too numerous to draw strong conclusions about the impact of different background classes (as seen in Section 4.2.4). The many-to-many relationship between background classes and hierarchical labels also reduces the reliability of drawing conclusions based on labelled groups of background classes. As a result, it may be appropriate to select an alternative data set or a subset of classes from SUN397 in future research.

With this said, the choice of the ShapeNetCore and SUN data sets does allow for the production of a highly diverse data set, containing both a large amount of images and background classes. While some models in ShapeNetCore are *not* photorealistic, and may not have accurate materials and textures, the more photorealistic scanned object data sets were not large enough and had other limitations that made them unsuitable for this research. With these data sets selected and retrieved they were then verified and preprocessed. The steps undertaken to achieve this are covered in the following section.

3.1.6.3 Preprocessing

ShapeNetCore: Before using the ShapeNetCore models in the synthesis pipeline, two preprocessing steps were performed. Firstly, a mapping was created between the 55 classes present in ShapeNetCore and the 1000 classes of the ImageNet data set. This mapping is necessary for defining correct outputs when the synthetic images are used as input to ImageNet classifiers, and the complete mapping can be found in Appendix A.1. The seven classes that did not have corresponding classes in the ImageNet data set[2] were then removed from ShapeNetCore, as they would not be useful for training or evaluation of ImageNet models. This results in the removal of only 2,290 models from the initial data set of 51,649, leaving 49,359 models.

Following this, all models were converted to `.gltf` format using the command line tool

[2]The seven classes removed from ShapeNetCore are *bed, earphone, faucet, jar, knife, rocket,* and *skateboard.* While there are near matches for some of these classes (e.g. *letter opener* for *knife*, and *studio couch/day bed* for *bed* they were ultimately excluded since this produces a more strictly correct data set which will still be of a sufficient size.

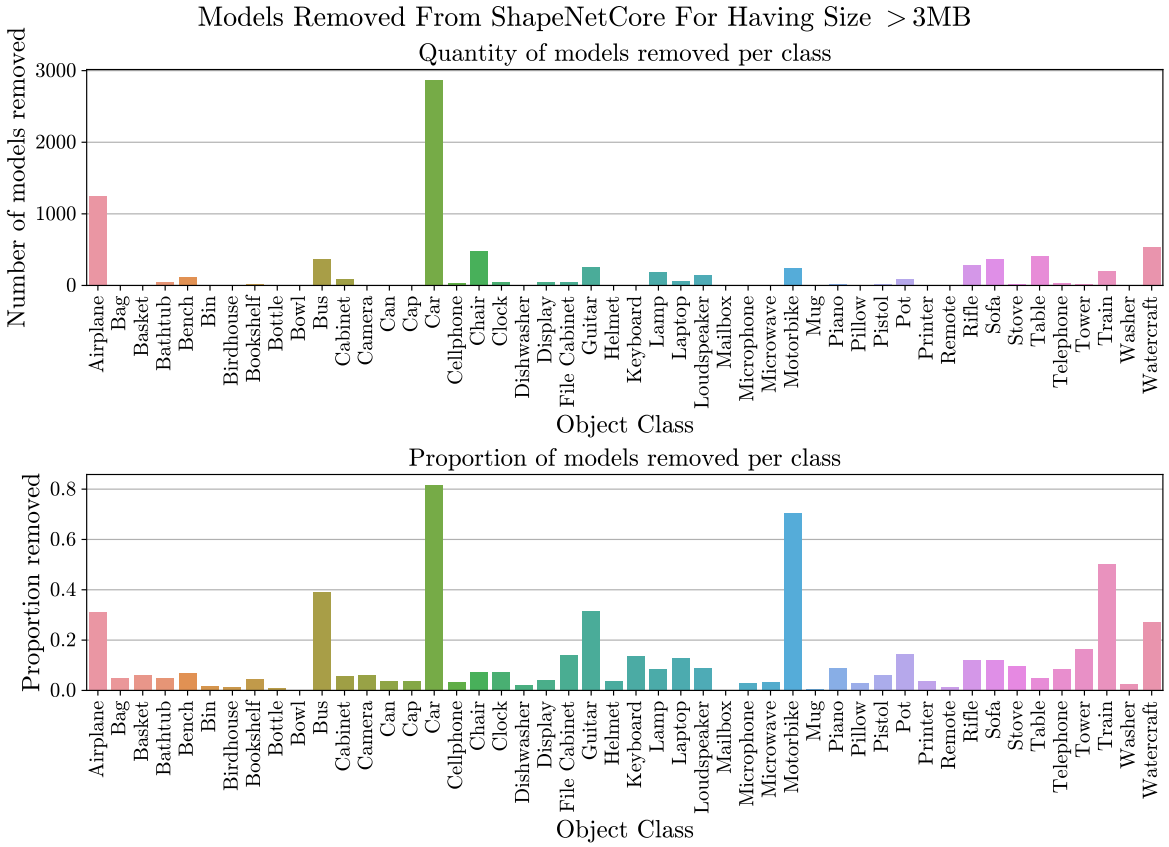


Figure 3.6: The distribution of models that were excluded from the rendering process for having a size greater than 3MB. It is clear from this figure that certain classes such as *car* and *motorbike* have a significantly greater proportion of their models removed due to the higher average quality of these models.

`obj2gltf`[3], as models in this format loaded more quickly and correctly into *Blender*. At this stage, models with `.gltf` files larger than 3MB were removed from the data set, as large models caused significant delays in loading times which would have added over a week to the duration of the rendering process. The choice to remove larger models was made to adhere to the time constraints on the project, and the distribution of models removed is shown in Figure 3.6. This is a considerable limitation of this research, as this resulted in the removal of 8,387 of the remaining 49,359 models (leaving 40,972 models).

SUN Database: The SUN data set is used for the background images in the compositing process, and requires a single preprocessing pass to resize images and remove those with insufficient resolution. To do this, each image is processed iteratively, and:

1. Images smaller than 224×224 pixels are removed.
2. Images larger than 224×224 pixels are scaled down (using Lanczos resampling) such that either the width or height of the resulting image (whichever is smaller) becomes 224 pixels, and the aspect ratio is maintained.

[3] Available at `https://github.com/CesiumGS/obj2gltf`

With the data sets now selected and preprocessed, they are fed into the image processing pipeline that was described in Section 3.1. The specific process used for rendering the image subjects from ShapeNetCore models is described precisely in Section 3.1.7, and the compositing process is described in Section 3.1.8.

3.1.7 Object Rendering Algorithm

With the data sets preprocessed, the ShapeNetCore models are ready to be rendered into `.png` images with transparent backgrounds. The approach for this, which was described and justified in Sections 3.1.3 and 3.1.4 is described in pseudocode in Algorithm 1.

3.1.8 Background Compositing Algorithm

After rendering synthetic images of the ShapeNetCore objects according to the algorithm defined in the previous section, these images proceed into the compositing phase of the synthesis pipeline. In this phase, each rendered image $\boldsymbol{i}$ output from Algorithm 1 proceeds through the following process to produce a final composite image for the synthetic data set:

1. An integer s is randomly sampled from the range $[90, 224]$. The 224×224 pixel rendered image $\boldsymbol{i}$ is scaled down to $s \times s$ pixels (using Lanczos resampling). Here the range imposed on s sets a lower bound of 90×90px on the size of the subject in the final composited images.

2. A random background class is sampled uniformly from the 397 classes of the SUN397 data set to ensure a uniform distribution of background classes.

3. A random background image $\boldsymbol{b}$ is sampled uniformly and with replacement from the preprocessed images in that background class. Note that the preprocessed images have already been resized (as described in Section 3.1.6.3) such that the width, height, or both are 224px.

4. A 224×224 pixel patch $\boldsymbol{b}_{\text{SUN}}$ is taken from $\boldsymbol{b}$ for use as the background in the composited image. By performing this sampling here instead of during the preprocessing stage it becomes possible to sample various patches from each background image.

5. A 224×224 pixel image of pure white pixels $\boldsymbol{b}_{\text{white}}$ is also created.

6. A horizontal offset o_w and a vertical offset o_h (both integers) are sampled independently from the range $[0, 224 - s]$. Together these define the position of the scaled image subject in the resulting composite images.

7. Composite images $\boldsymbol{i}_{\text{SUN}}$ and $\boldsymbol{i}_{\text{white}}$ are created by taking the backgrounds $\boldsymbol{b}_{\text{SUN}}$ and $\boldsymbol{b}_{\text{white}}$ and overlaying the scaled image subject o_w pixels from the left edge, and o_h pixels from the top edge of the background.

8. These composite images are saved along with their corresponding metadata. This metadata includes the background class label, the scale of the image subject (s), and

Algorithm 1 The rendering process implemented in *Blender* that is the first phase of the image synthesis pipeline. This pseudocode shows how the approach described in Sections 3.1.3 and 3.1.4 was implemented.

Input: A set M of ShapeNetCore models in `.gltf` format, paired with their class label: $M = \{(m_1, c_1), \ldots, (m_{N_m}, c_{N_m})\}$.

Output: A set rendered image files, paired with their class and parameter labels: $I_{\text{rendered}} = \{(\boldsymbol{i}_1, L_{\text{rendered},1}), \ldots, (\boldsymbol{i}_N, L_{\text{rendered},N})\}$.

procedure RENDERIMAGESUBJECTS(M)
  $S \leftarrow 100,000$ ▷ $P = 2S + 1$ is the total amount of facing directions we sample from for each render.
  $F \leftarrow \left\{ \left(2\pi j \phi^{-1}, \arcsin\left(\frac{2j}{2S+1} \right) \right) \,\middle|\, j \in \{-S, \ldots, S\} \right\}$
  $A \leftarrow \{-180, -135, \ldots, 135, 180\}$ and $E \leftarrow \{-180, -135, \ldots, 135, 180\}$
  $L \leftarrow (A \times E) \cup \{(0, 180), (0, -180)\}$ ▷ L contains the yaw-pitch rotation pairs that produce the 26 lighting configurations.
  Initialise a global *("sun")* light facing downwards with an intensity of $1W/m^2$.
  Initialise a *"point"* light at the origin with an intensity of $200W$.
  Position a camera at $(x, y, z) = (0, 1, 0)$, facing the origin.
  Set the output resolution to 224×224 pixels.
  Set the output format to `.png` with transparent backgrounds enabled.
  $I \leftarrow \{\}$
  for (m, c_o) **in** M
    if SIZEOF(m) > 3MB
      continue
    $m_l \leftarrow$ LOADMODEL(m)
    $N_m \leftarrow \max(10, \frac{15,000}{\text{NUMOBJECTSINCLASS}(c_o)})$
    $P \leftarrow F$.RANDOMSAMPLE(N_m)
    for (r_a, r_e) **in** P
      with 50% probability: ▷ 50% chance of a π^c roll rotation.
        $r_a \leftarrow r_a + \pi^c$ and $r_e \leftarrow r_e + \pi^c$
      $(l_a, l_e) = l \leftarrow$ UNIFORMSAMPLE(L)
      Set the azimuth of the point light source to l_a.
      Set the elevation of the point light source to l_e.
      Rotate m_l to its default orientation $(0, 0, 0)$.
      Rotate m_l by r_a along the yaw (Z) axis then r_e along the pitch (X) axis.
      Translate the camera such that the object fits the frame exactly.
      $\boldsymbol{i} \leftarrow$ RENDERIMAGE()
      $L_{\text{rendered}} \leftarrow \{c_o, l, r_a, r_e\}$ ▷ The metadata, including object class label c_o, lighting configuration l, and azimuth and elevation rotations r_a and r_e is collated.
      I_{rendered}.INSERT($(\boldsymbol{i}, L_{\text{rendered}})$) ▷ The image is stored with its metadata.
    DELETEMODEL(m_l)

the position offsets (o_w and o_h), as well as the metadata generated for the image subject in the rendering phase.

3.2 Validating Synthetic Data

After producing the synthetic data set, the images were validated to ensure the desired properties and distribution were achieved. In this section, the results of this validation are presented. This involves looking at the distribution of the explanation parameters as well as additional parameters such as those that define the position of the image subject in the composited image.

ShapeNetCore Class Distribution: To perform an effective evaluation of existing models *and* successfully train a new model, it was important to achieve a relatively uniform distribution of classes in the synthetic data sets. Using the *Spherically-Distributed Sampling* approach described in Section 3.1.3 it was possible to ensure at least 15,000 images were captured of each class, with more images for classes that presented many models in ShapeNetCore. The complete distribution of object classes across both the composited data sets (SUN and white backgrounds) is shown in Figure 3.7. The oversampling of certain classes (including *table*, *chair*, and *airplane*) will prove advantageous when evaluating existing models, but is addressed when using the data set for Objective 3 (Model Training) in Section 5.1.4.1.

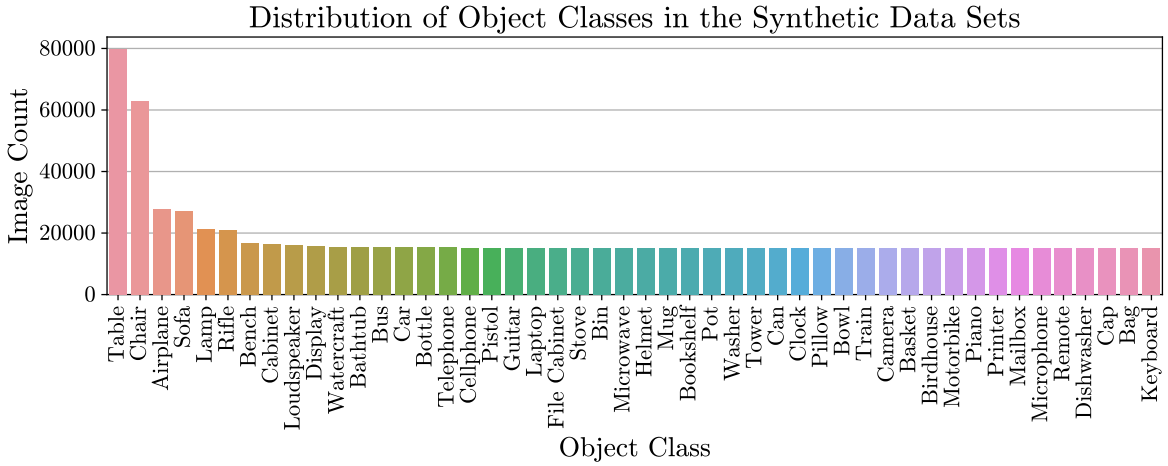


Figure 3.7: The distribution of ShapeNetCore classes across the images in the final synthetic data sets.

Rotation: The rotation of the image subject is a critical explanation parameter to learn successfully for Objective 3 (Model Training). To maximise the chance that the model effectively learns to predict this parameter, it was considered important to achieve a uniform distribution of rotations across both 3D space and parameter space. Figure 3.1 shows sample images across the pitch-yaw parameter space, demonstrating that this sampling method produces many unique views of each object, and that both of these rotation parameters cause a

noticeable change in the appearance of the object.

The default pose of each object is defined by the ShapeNetCore data set, and the default poses for all object classes in the synthetic data set can be seen in Figure 3.5. As justified in Section 3.1.3, the *Spherically-Distributed Sampling* approach is used to sample rotations for each object in the synthetic images. This was done to achieve facing directions that are distributed uniformly across the sphere (Figure 3.2), as well as uniformly distributed parameters defining these rotations. Figure 3.8 shows the distribution of these parameters, demonstrating that the yaw rotations are uniformly distributed, and that while the pitch rotations are *not*, a sin transformation of these values results in a uniform distribution suitable for model training.

Lighting Configurations: The lighting direction is the second explanation parameter varied in the synthetic images. Lighting directions are sampled for each image as described in Section 3.1.4, with the 26 possible positions shown in Figure 3.3. Figure 3.9a confirms that in the synthetic data set the lighting configurations are uniformly sampled from the 26 possible positions. Figure 3.9b shows how these 26 positions can be grouped based on their position relative to the 3D model. These groups will later be used to draw more general conclusions about the impact of lighting directions on classification performance.

Subject Scale and Location: While the scale and position of the image subject in the composited images is not considered one of the primary explanation parameters, the model developed for Objective 3 (Model Training) will still be designed to predict these attributes. By outputting a prediction for the scale and location of the image subject, the outputs from the developed model can be used to synthesise a complete explanatory image by using these predictions as input to the synthesis pipeline. As such, the distribution of scales and offset locations is shown in Figure 3.10.

Note that the X and Y offsets are dependent on the scale of the image subject so that the rendered subject is contained entirely within the bounds of the image, as such, the distribution of offsets is non-uniform. Since the scale of the image subject varies between 90 and 224 pixels (inclusive), there are $224 - 90 + 1 = 135$ possible scales, and the distribution of offsets follows the PMF

$$f_X(n) = f_Y(n) = \frac{1}{135} \sum_{i=n}^{135} i^{-1}, \quad \text{for } n \in \{0, 1, \ldots, 134, 135\}$$

Figure 3.10b shows the related distribution of the centre points of the composited image subjects, coloured according to the subject scale. Note that centre points are concentrated around the centre of the image, as larger image subjects cannot be positioned close to the edges while remaining within the image boundaries. This figure demonstrates the lack of correlation between the horizontal and vertical offsets.

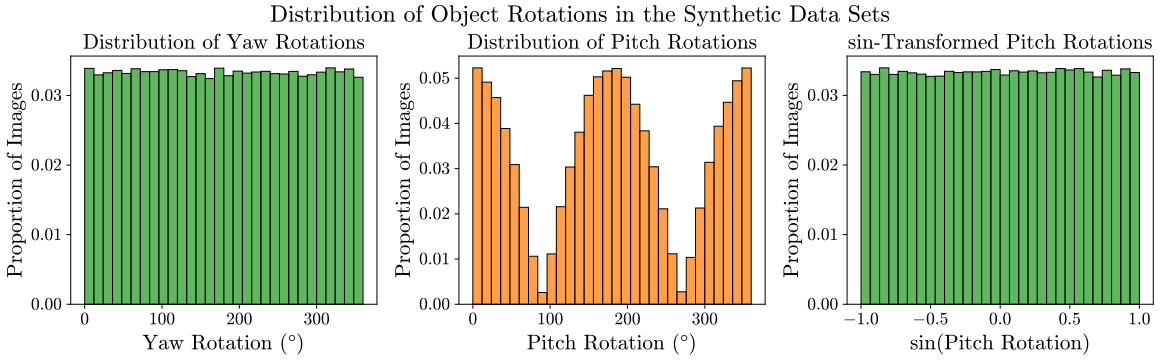
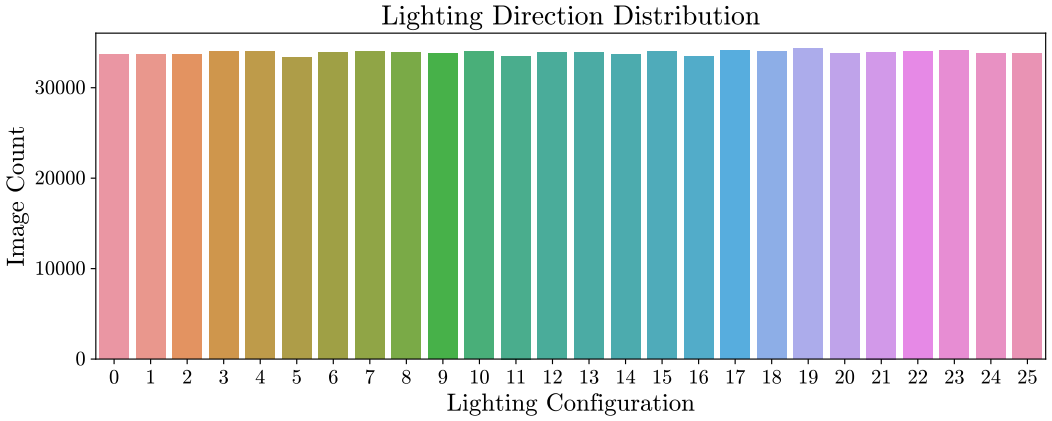
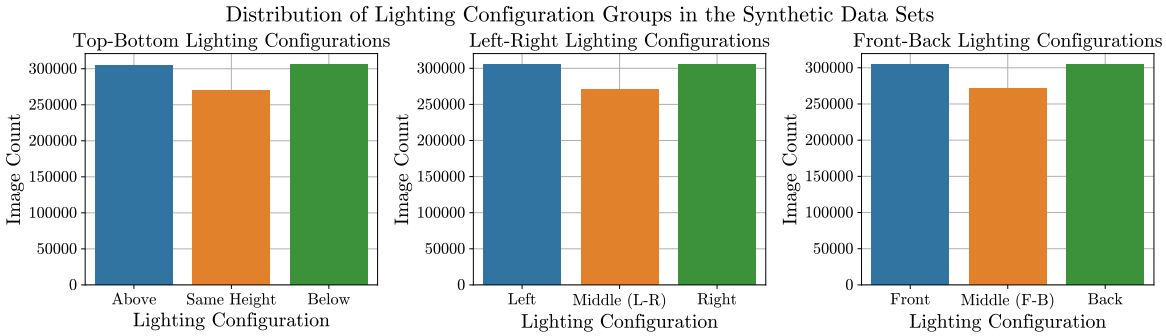


Figure 3.8: The distribution of the object rotations in the final synthetic data set. Yaw rotations and transformed pitch rotations (green) follow a uniform distribution, while pitch rotations (orange – generated with the Fibonacci lattice algorithm) do not.

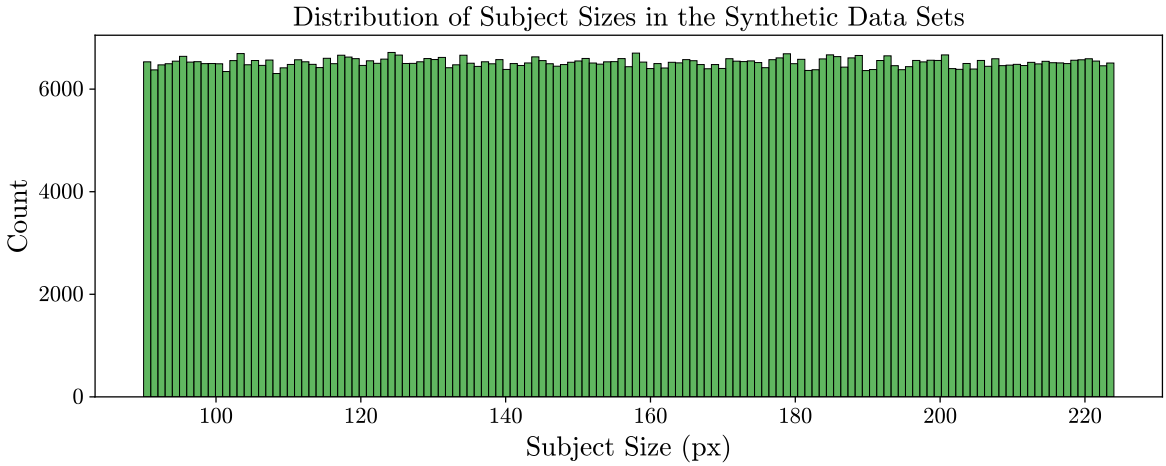


(a) The distribution of images across the 26 lighting configurations.

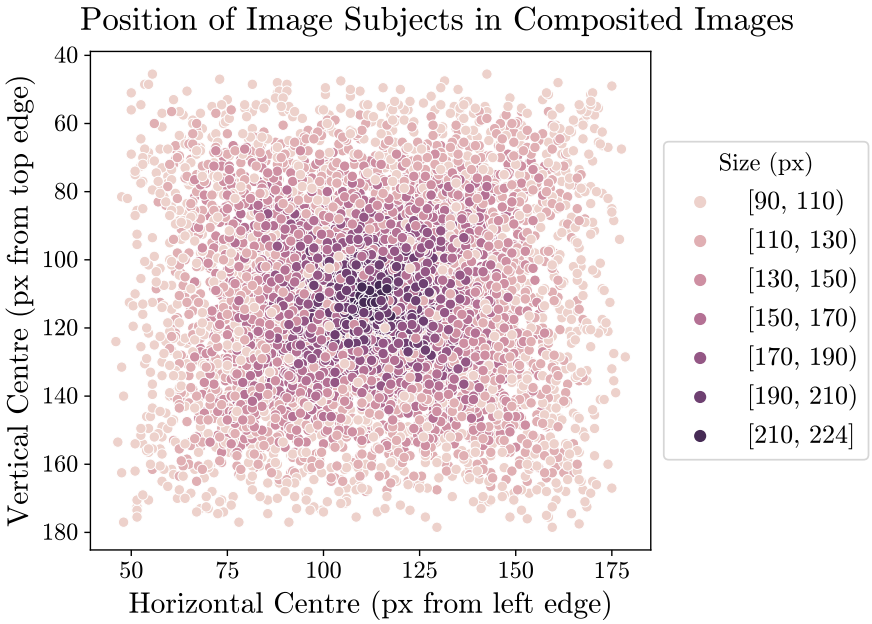


(b) The distribution of lighting configurations across groups in the three axes. Note that the *'Middle'* group in each set contains one fewer sample since the object occupies the position in the centre of the scene.

Figure 3.9: Distribution of lighting configurations across all images in the synthetic data set. Refer to Figure 3.3 for a visualisation of the 26 lighting positions.



(a) The distribution describing the sizes of rendered image subjects in the final composited images across the synthetic data sets.



(b) The distribution of composited image subject centres in the synthetic data set (sample of 10,000). The centre point of the image subject is dependent on both the horizontal and vertical offsets, and the subject size (which is randomly sampled between 90 and 224 pixels, and is shown in the legend).

Figure 3.10: Distribution of rendered image subject scales and locations.

Chapter 4

Robustness Evaluation of SOTA Models

When deploying image classifiers in real-world applications it is critical that users are aware of their limitations so that accuracy can be maximised and failure cases can be avoided. As computer vision models including image classifiers are deployed in more contexts it has become increasingly important to understand how state-of-the-art models respond to changing conditions in their inputs.

This research is motivated by the knowledge that existing models are *not* be invariant to changes in the explanation parameters (namely object pose, image background, and lighting direction). If state-of-the-art models were already invariant to these parameters there would be no need to train a new model for Objective 3 (Model Training). However research by both Alcorn et al. (2019) and K. Xiao et al. (2020) suggests that this is not the case, motivating the need for the thorough evaluation that is performed as Objective 2 (Existing Model Evaluation).

As such, Section 4.1 presents the methodology for evaluating the parameter invariance of state-of-the-art image classification models. This includes both the selection of models that were evaluated, and the way that they were evaluated using the data set produced in the previous chapter. Section 4.2 then presents the results of this evaluation, drawing conclusions about how models respond to the various explanation parameters, and using failure cases as a form of example-based explanation.

4.1 Methodology for Evaluating Existing Models

While some existing literature considers the impact of individual parameters like pose (Alcorn et al., 2019) and background (K. Xiao et al., 2020) on model performance, a comprehensive evaluation investigating a broad combination of parameters has not previously been performed. This evaluation aims to fill that gap and produce novel results, especially when

evaluating parameters that have not been studied in existing research, such as tolerance to changes in lighting direction. The evaluation serves not only to test the hypothesis that state-of-the-art models are *not* robust to the explanation parameters, but also to select the model that will be built upon for Objective 3 (Model Training).

For this evaluation, the synthetic data set produced for Objective 1 (Image Synthesis) plays a critical role. To assess the robustness of each model to changes in the explanation parameters each model is tested on the entire synthetic data set, generating insights into the performance of the model throughout the parameter-space. By identifying specific areas in the explanation parameter-space where models struggle to produce correct results, this evaluation aims to provide value to both model developers and end-users. For developers, this research highlights areas in which models and data sets can be further refined to increase performance. For end-users it provides a guide to the limitations of existing models to educate their deployment in real-world scenarios.

In the following sections, justification will be provided for the specific models that were evaluated, then the evaluation process will be described.

4.1.1 Software and Model Selection

The four models selected for this evaluation were introduced in Section 2.1, and were chosen for their widespread use and/or state-of-the-art performance. These are:

- *MobileNet V2* (Sandler et al., 2018),
- *ResNet-152* (K. He et al., 2016),
- *MobileViT V2* (Mehta & Rastegari, 2022), and
- *Swin Transformer V2 (Large)* (Z. Liu et al., 2022).

As mentioned in Section 2.1, *MobileNet V2* and *ResNet* are both CNN-based models, while the *Swin Transformer V2* is transformer-based, and *MobileViT V2* is hybrid. It was considered important to select models representing diverse architectures to test the hypothesis that these architectural differences result in different responses to variation in the explanation parameters. For a similar reason, two of the models selected are *mobile* models, designed to be small and computationally efficient. It is hypothesised that the light-weight nature of these models will result in reduced robustness to variation in the explanation parameters, and evaluating two such models allows for this to be tested.

To ensure consistency across models, the four selected classifiers were evaluated using a modular evaluation pipeline implemented in Python with the HuggingFace `transformers` library (Wolf et al., 2020). This software was chosen as it provides a unified interface to generate predictions from each model, and allows for the synthetic data set to be loaded in a format that is compatible with all selected classifiers without modification.

As such, the four selected models are all available via HuggingFace, and the specific variants

used are:

***MobileNet V2*:** Version 1.4 of the *MobileNetV2* model, which takes input images of 224×224 pixels and is pre-trained on ImageNet by Sandler et al. (2018).

***ResNet*:** The 152-layer version of ResNet, which is the deepest version of the model, pre-trained on ImageNet by K. He et al. (2016). This version of ResNet (unlike the shallower 18 and 34 layer versions) uses *bottleneck blocks* to reduce the dimensionality and consequently the resource requirements of hidden layers in the model.

***MobileViT V2*:** Version 2.0 of *MobileViT V2*, also pre-trained on ImageNet, but at a resolution of 256×256 pixels (Mehta & Rastegari, 2022). The model is nonetheless suited for the 224×224 pixel data set as this is managed by the model's image preprocessor.

***Swin Transformer V2*:** The *Large* version of the *Swin Transformer V2* model. While this is smaller than the *Huge* and *Giant* variants, it was the largest version available for evaluation and retraining. It is also pre-trained on ImageNet, but at a resolution of 192×192 pixels, and is then fine-tuned on images at a larger 256×256 pixel resolution (Z. Liu et al., 2022).

The specific model variants were selected based on which achieved the highest accuracy on ImageNet. Naturally this resulted in selection of the largest available variant for each model. The following section describes how the synthetic data set was used to evaluate these models.

4.1.2 Evaluation Process

Using the unified interface provided by the `transformers` library, the evaluation process proceeds according to the following simple steps.

1. The two synthetic data sets are loaded with their metadata using the HuggingFace `datasets` library. These are the data sets of rendered ShapeNetCore objects on SUN and white backgrounds, and each contains 880,050 images across the 48 classes.

2. Each of the four models are initialised with weights from pre-training on ImageNet. Classification pipelines are created for each model, which take images from the data sets, and return the model's class predictions and their associated probabilities.

3. Each ImageNet classifier is evaluated on both synthetic data sets. For each instance in the data sets the top-5 predictions are recorded, as are the predicted class probabilities (SoftMax outputs).

4. In a subsequent post-processing step, the mapping between ShapeNetCore classes and ImageNet classes (described in Section 3.1.6.3 and defined in Appendix A.1) is used to derive two boolean columns for each class. These derived columns are top-1 and top-5 correctness, which are `true` if and only if the model correctly identified the class in its top-1 or top-5 predictions respectively.

In the following section, the data collected during this evaluation is used to generate insights into the performance of each model under variation in the explanation parameters.

4.2 Existing Model Evaluation Results

In this section, the results produced by the evaluation process outlined above are analysed. First, the overall classification performance of the four models on the synthetic data set is presented, followed by an investigation of how the models respond to variation in each of the explanation parameters.

4.2.1 Performance Across Classification Models

The classification performance of the four models is presented in Figure 4.1 and Table 4.1 which visualise the top-1 accuracy of each model between the ImageNet data set and the synthetic data set produced in this research. For this, ImageNet top-1 accuracies are taken from the authors of the models (*MobileNet V2 (1.4)* (Sandler et al., 2018), *Swin Transformer V2 (Large)* (Z. Liu et al., 2022), *ResNet-152* (K. He et al., 2016), and *MobileViT V2 (2.0)* (Mehta & Rastegari, 2022)).

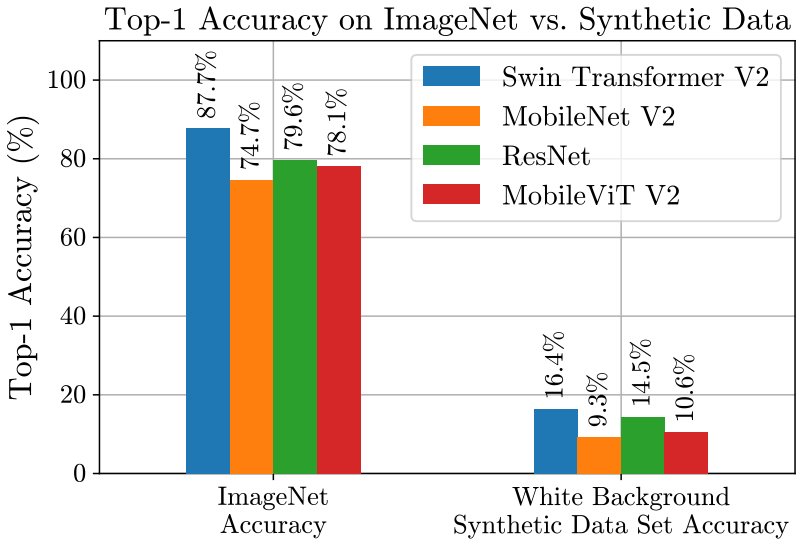


Figure 4.1: Top-1 classification accuracy for the four benchmarked models on ImageNet and the synthetic data set with white backgrounds.

While it is difficult to draw robust conclusions based on this sample size of four models, these visualisations suggest that models which perform better on ImageNet also perform better on synthetic data. The percentage changes listed in Table 4.1 additionally suggest that models with better performance on ImageNet transfer more effectively to the synthetic domain. That is to say, the highest performing models also see the lowest percentage change in accuracy. It is therefore hypothesised that as computer vision models develop more robust feature representations to achieve higher performance on real images, these improvements positively impact their ability to transfer to the synthetic domain.

Table 4.1: Comparison of top-1 classification accuracy across models on ImageNet vs. the synthetic data set with white backgrounds.

Model	**Accuracy on ImageNet (%)**	**Accuracy on Synthetic Images with White Backgrounds (%)**	**% Change in Accuracy**
Swin Transformer V2 (Large)	87.7	16.4	-81.3
MobileNet V2 (1.4)	74.7	9.3	-87.6
ResNet-152	79.6	14.5	-81.7
MobileViT V2 (1.0)	78.1	10.6	-86.4

4.2.2 Classification Performance on Synthetic Images

When comparing the performance of the four classification models in the previous section, it is clear that all four models see a significant decrease in accuracy between ImageNet and the synthetic data set. The *Swin Transformer V2* model, which performs best on both tasks, also sees the smallest percentage difference of 81.3%, while the worst performing model, *MobileNet V2*, sees the largest percentage difference of 87.6%. With this in mind, it is important to now consider *why* the models exhibit such a significant decrease in accuracy on the synthetic data.

Due to the superior performance of the *Swin Transformer V2* model, most of the results presented in the following sections are drawn from the *Swin Transformer V2* evaluation results. With this said, the conclusions of Sections 4.2.2.1–4.2.2.2 apply across all four models that were evaluated, and additional visualisations pertaining to the other models can be found in Appendix B.

4.2.2.1 Classification models transfer successfully to synthetic images

While model performance is significantly lower on the synthetic data sets, it is not immediately clear whether this is due to (a) an inability for the models (trained on real images) to transfer to the synthetic domain, and/or (b) the classification task being more challenging due to variation in the explanation parameters. However, by thoroughly investigating of the effects of variation across each parameter in Sections 4.2.3–4.2.6 we conclude that classification models *do* transfer successfully to the synthetic domain, and that challenging image conditions in the synthetic images are responsible for most of the decrease in performance.

Support for this conclusion can be found in Section 4.2.3, and specifically in Figure 4.6. This figure demonstrates that when presented with ideal poses of objects in the synthetic images, the *Swin Transformer V2* model is capable of accuracies approaching 100% on certain objects. This clearly suggests that variation in object pose is more responsible for reduced accuracy than the domain gap between synthetic and real images. Similar but less pronounced patterns are observed with the other explanation parameters, and will be explored in more detail in the relevant sections.

4.2.2.2 Varied explanation parameters are responsible for type-II errors

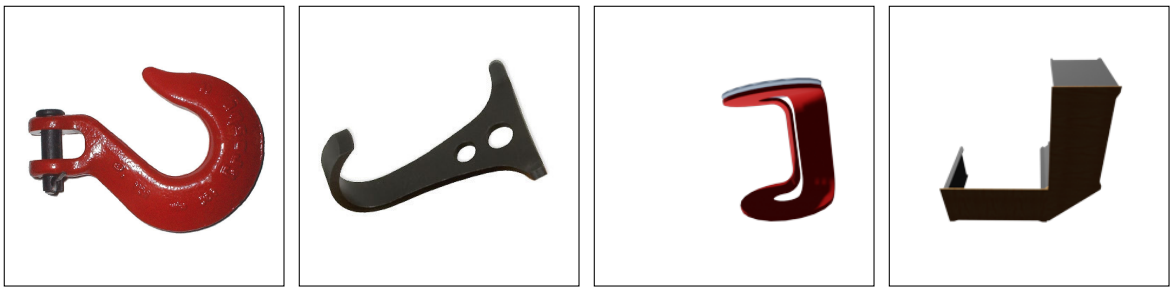
Having concluded that classification algorithms perform suitably well on the synthetic data, it is worth looking at broad patterns that appear throughout the entire evaluation before looking more deeply at the effects of the individual explanation parameters in later sections. To this end, Table 4.2 shows the top-5 classification results of the *Swin Transformer V2* for five of the object classes in the synthetic data set. Items coloured green are considered correct according to the mapping defined in Appendix A.1, and items coloured red are considered incorrect. By inspecting these results and the results across other classes and classifiers some clear patterns emerge, especially concerning the type-II (false negative) errors of the models.

Firstly, the *nail* class appears as a frequent false negative result for many object classes. By inspecting the images that this output arises on it is clear that *nail* is often predicted for images that contain either (a) a view of the image subject that reduces it to a circular shape, or (b) a pose that results in a rod-like shape. Based on both intuition and inspection of the ImageNet training set, it is likely that these poses correspond to images of nail heads (circular) and full nails (rod-like). Since the synthetic data set contains various objects in unusual poses and under lighting conditions that obscure identifying features it is unsurprising that many views reduce to simple shapes that are easily confused as *nails*. Figure 4.2b supports this by showing instances of synthetic *table* images that present these unusual views and are consequently misclassified as *nails*.

Table 4.2: Top-5 most frequent classification outputs from the *Swin Transformer V2* model for selected ShapeNetCore objects on white backgrounds.

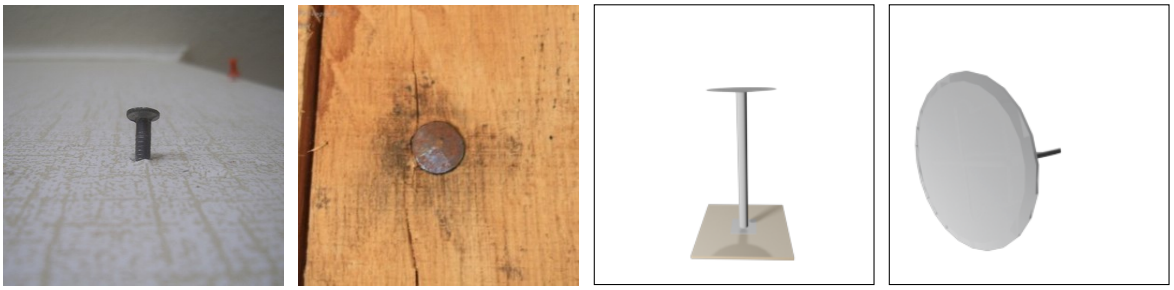
	Top-5 Most Common Class Labels (*Swin Transformer V2*)									
Object Class	**1**		**2**		**3**		**4**		**5**	
	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5
Airplane	*Missile*		*Airliner*		*Warplane*		*Wing*		*Airship*	
	38.6%	81.0%	28.2%	55.5%	15.9%	79.2%	3.2%	64.0%	2.3%	27.8%
Chair	*Folding Chair*		*Hook*		*Rocking Chair*		*Guillotine*		*Barber Chair*	
	17.2%	37.5%	10.5%	35.1%	6.5%	26.1%	5.5%	19.9%	3.0%	10.2%
Lamp	*Spotlight*		*Nail*		*Table Lamp*		*Plunger*		*Microphone*	
	10.8%	39.4%	8.3%	21.5%	8.0%	24.8%	7.9%	27.3%	6.0%	28.0%
Rifle	*Assault Rifle*		*Rifle*		*Missile*		*Hook*		*Microphone*	
	31.4%	70.2%	24.5%	70.0%	10.2%	50.1%	4.1%	28.1%	3.3%	19.4%
Table	*Guillotine*		*Desk*		*Hook*		*Dining Table*		*Nail*	
	9.9%	30.0%	9.1%	32.8%	8.5%	33.8%	5.5%	20.5%	5.1%	17.2%

Similarly, the *hook* class appears as a common false negative option for many object classes. It is suggested that, similar to the *nail* class, this arises under poses and lighting conditions that hide identifying features. The *hook* class, however, is observed for classes that reduce to shapes containing bends and curves. One such example is the *chair* class, which from side views may present as a simple shape with various curves. Additionally, the *table* class



(a.i) ImageNet *hook* (a.ii) ImageNet *hook* (a.iii) Synthetic *table* (a.iv) Synthetic *table*

(a) ImageNet *hooks*, and synthetic *table* images that are misclassified as *hooks*.



(b.i) ImageNet *nail* (b.ii) ImageNet *nail* (b.iii) Synthetic *table* (b.iv) Synthetic *table*

(b) ImageNet *nails*, and synthetic *table* images that are misclassified as *nails*.

Figure 4.2: Sample images from the *hook* and *nail* classes of ImageNet (left two columns), and synthetic images of *tables* that are misclassified as these objects (right two columns).

is commonly confused with *hook*, which may initially be unintuitive, but begins to make sense when considering the diverse yaw *and* pitch rotations sampled in the synthetic data set. Figure 4.2a supports this with more unusual perspectives of *tables* which result in misclassification as *hooks*.

While this section has looked briefly at how variation in the explanation parameters causes confusion and type-II error with specific object classes, the following sections will investigate more deeply how variation across the explanation parameters causes systematic failure across the various classification models.

4.2.3 Invariance to Object Pose (Rotation)

The pose of the image subject (used here to describe its 3D rotation) plays a significant role in how the object appears in an image. In data sets such as ImageNet, the photos in each object class tend to showcase a specific perspective of the object, with limited representation of dramatically different poses. Certain objects, especially those smaller in size, tend to be photographed from above, while large or airborne objects (e.g. *airplane*) are more often imaged from other perspectives. The perspective from which a given object is most often photographed will hereafter be referred to as its *canonical pose*.

It is hypothesised that because training data most often showcases images in their canonical

pose, these poses should be classified most accurately during evaluation. Conversely, it is expected that unusual poses, such as views from underneath and behind objects will be classified with a significantly lower accuracy. More broadly, the results presented in this section highlight the specific regions of rotation-space that models can be expected to perform correctly in for each object. This is valuable for many real world applications which require computer vision models to be invariant to pose changes for safe operation. The following sections (4.2.3.1–4.2.3.5) present five key conclusions about the pose invariance of the four models.

4.2.3.1 Classification models are easily confused by object rotations

Existing work by Alcorn et al. (2019) suggests that neural network models are easily confused when the image subject is transformed by rotation and translation. This research validates the existing conclusion that classifiers are not invariant to rotation along the yaw and pitch axes by testing a more diverse sample of 3D models and object classes.

In Figure 4.3, each row contains four images of the same model sampled from ShapeNetCore. In each image, the model is subject to a different rotation along the pitch and yaw axes, and is presented at a different scale and position in the composited image.

When inspecting the predictions output by the *Swin Transformer V2* model, it is clear that specific poses cause the model to produce incorrect outputs, sometimes with a high level of confidence. This is true even when other poses are classified correctly. In many of the incorrectly classified examples the model's output is understandable based on the appearance of the object in its pose (e.g. *crutch* in row 3, column 3 of Figure 4.3).

4.2.3.2 Correct classifications are localised in rotation-space

Knowing that classification models perform more accurately with objects in specific orientations, one may next ask which rotations are most desirable for maximising classification performance. To this end, Figure 4.4 shows the mean accuracy across all classes over the entire rotation-space. In this figure the leftmost plot shows the distribution of images captured across each object's pitch and yaw axes, discretised into 10° intervals. Note that the non-uniform distribution present in this plot arises because the pose sampling method was selected to distribute facing directions uniformly around the sphere, rather than distributing angles uniformly along the pitch and yaw axes. The middle and rightmost plots then show the top-1 and top-5 accuracy of the *Swin Transformer V2* model for the images in those orientations, averaged with equal weight for each of the 48 object classes.

Based on this visualisation there is a clear concentration of classification accuracy in the region surrounding 0° pitch and yaw rotation. While this result depends entirely on the default alignment of the objects (shown in Figure 3.5), the ShapeNetCore data set is aligned such that the models are *upright* and, where applicable, have their *'front'* facing the camera (A. X. Chang et al., 2015). This distribution therefore suggests that models are most accurately classified when viewed from the front, and the slight bias in favour of downwards

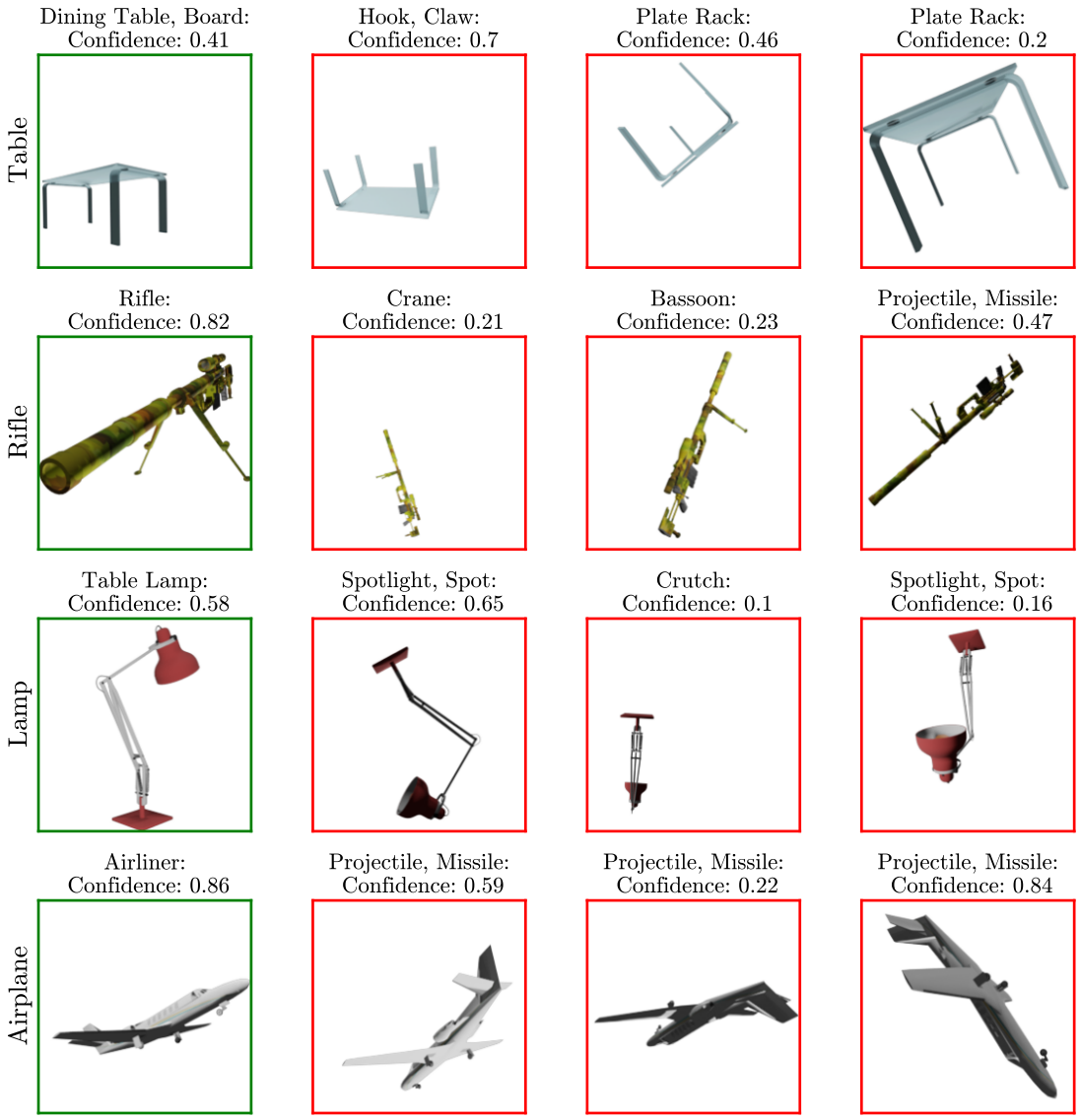


Figure 4.3: Sample objects that are classified correctly by the *Swin Transformer V2* in certain orientations (left column), and unsuccessfully in others (columns 2-4).

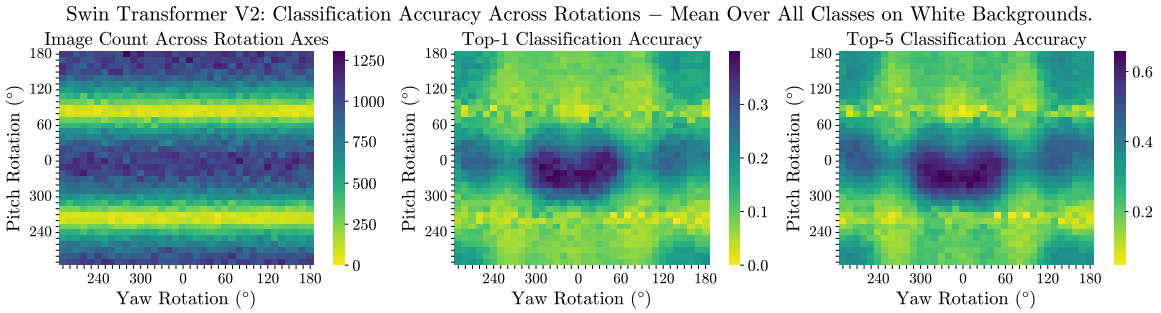


Figure 4.4: Mean classification accuracy of the *Swin Transformer V2* model across rotation-space. In this figure the mean is computed across classes, meaning that accuracy was calculated per-class then averaged across the 48 classes. Rotations are measured in 10° increments relative to the default orientations shown in Figure 3.5.

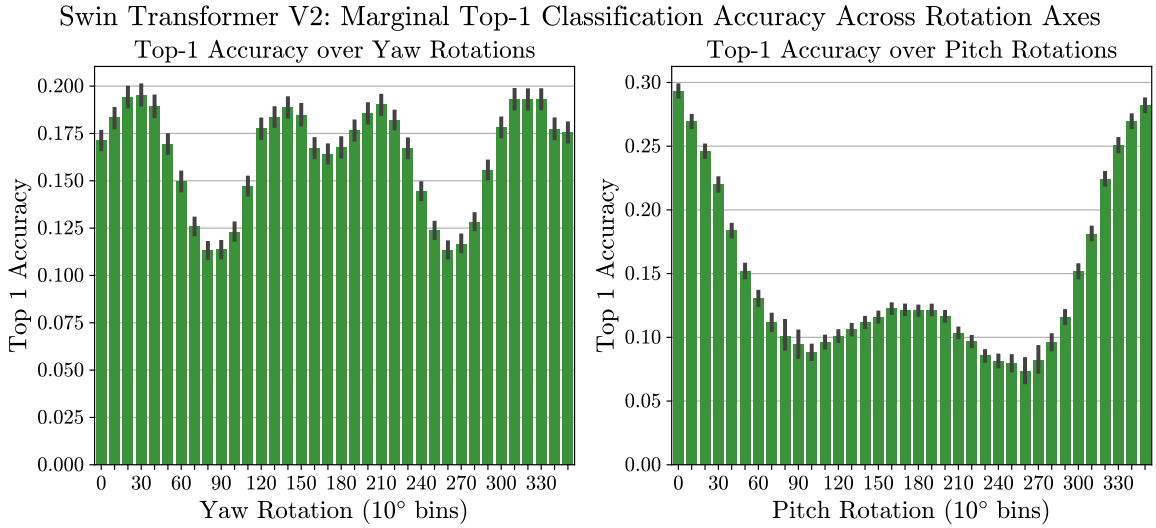


Figure 4.5: Marginal distribution of accuracy across the individual yaw and pitch axes with 95% Confidence Interval (CI) for the entire white background data set. Each bin labelled i spans the range $[i°, i + 10°)$.

pitch rotations at this angle implies a similar bias towards viewing objects from above. It is suggested that these biases are due to identifying features of ShapeNetCore objects most often being located on the front or top, as opposed to the back or bottom.

Looking at the marginal distributions of top-1 accuracy over rotations in each axis (Figure 4.5) and testing differences in proportion using a two-sample Z-test provides similar results. Looking first at yaw rotations it is interesting to note that classification accuracy in the 0° yaw rotation bin is worse than accuracy at ±30° by at least 1.5% for the *Swin Transformer V2* model on the synthetic data set ($t(73, 474) = -2.49, p < 0.01$). Across all objects the ±90° yaw rotations (corresponding to views directly from either side of the object) result in the most significant decreases in classification accuracy, showing a drop of over 7% when compared to the highest performing ±30° orientations ($t(97, 784) = -3.84, p < 0.001$).

Looking next at rotations along the pitch (elevation) axis it is clear that objects are classified most accurately when subject to elevation angles around 0°. Performance decreases monotonically as the pitch angle deviates towards 90° or −90°, with the maximum difference of at least 20% occurring between the 0° and 260° bins ($t(41, 803) = 2.54, p < 0.01$). Looking specifically at ±90° pitch angles there is no significant difference between 'up' and 'down' orientations ($t(6, 572) = -1.73, p = 0.084$).

As the pitch angle deviates further past ±90°, classification accuracy increases until the image subject is inverted at 180°. This 180° pitch rotation (which rotates the object *'upside down'*) results in a reduction in classification accuracy of at least 16% when compared to 0° ($t(76, 584) = -2.92, p < 0.001$) and a 2.4% increase when compared with 270° ($t(41, 629) = 2.49, p < 0.01$).

4.2.3.3 Classification performance across rotation-space depends on object class

Intuitively, one may expect that certain image subjects would be accurately classified in some orientations, and very challenging to accurately label in others. In this section some specific object classes are selected to show the significant interaction between object class and rotation in predicting classification accuracy.

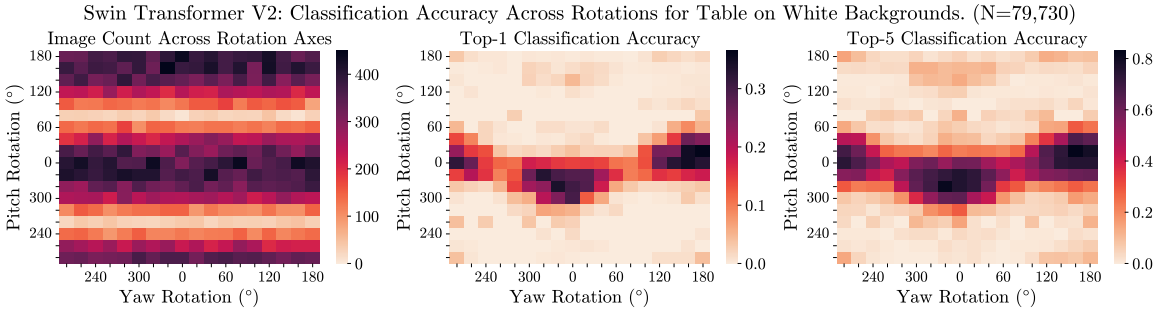
To investigate this hypothesis, Figure 4.6 shows a visualisation analogous to Figure 4.4 but for four individual classes selected from ShapeNetCore. Across these four classes, considerably different distributions are observed for accurate classifications across rotation space. Looking individually at the four cases:

Table: The distribution of accurate classification regions for the table class, shown in Figure 4.6a, displays a somewhat *sinusoidal* shape along the yaw rotation axis, centred around 0° pitch rotation. This pattern often often occurs for objects that (a) exhibit rotational symmetry around the vertical axis, and (b) are most often imaged from a slightly positive elevation (see also *bowl*). In this pattern, the band of high classification accuracies is curved since the positive camera elevation represented heavily in training data for these objects can be simulated by either pitching the object down, or by rotating the object 180° along the yaw axis then applying a positive local pitch rotation.

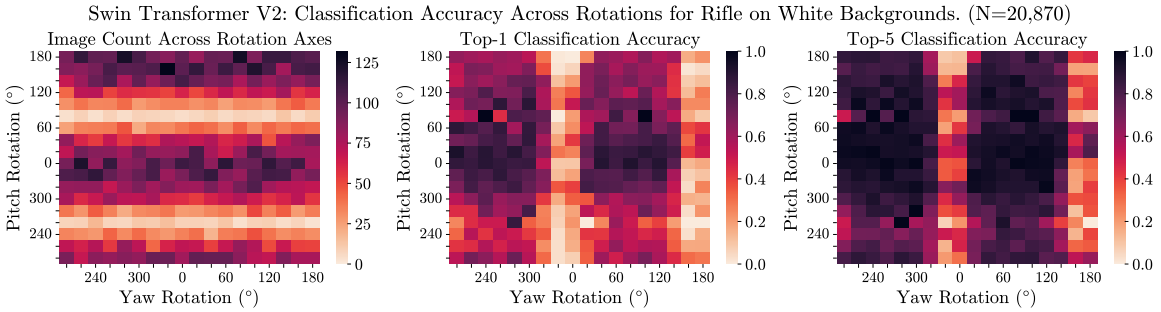
Rifle: The *rifle* class (Figure 4.6b) shows relatively consistent performance across all pitch rotations, but is classified very poorly at yaw rotations of approximately 0° or 180°. Referring to Figure 3.7, these angles correspond to the views from either direction of the barrel of the rifle, meaning that many of the visual features present on the sides of the object are obscured. Additionally, yaw rotations in this plane tend to produce degenerate views of classes such as *rifle* and *pistol* since from the front or bottom side many of these models appear as simple circular or rectangular shapes with relatively uniform colour.

Lamp: This class, shown in Figure 4.6c, demonstrates another pattern of accuracy across rotation-space that is shared with various other classes. Lamp models are classified most effectively at pitch rotations around 0°, with relatively uniform performance along the yaw axis. This pattern is characteristic of classes like *lamp* and *tower* that have a high level of rotational symmetry along their vertical axis, but unlike the *table* class, there is less bias towards photographing these objects from a positive elevation.

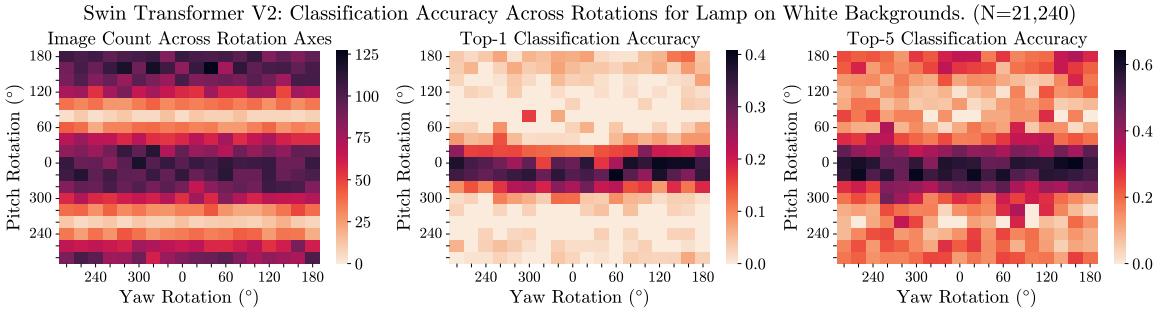
Remote: The classification accuracy for the *remote* class, shown in Figure 4.6d shows yet another distribution that is shared widely between object classes. The default orientation of the *remote* class presents the buttons toward the camera (see Figure 3.5). In and around this orientation, the objects present many identifiable features that are captured in the synthetic images. However, in other poses where this 'front' face is obscured, classes such as *remote* present relatively few identifiable features. As a result, the distribution of classification accuracy for this object class is centred on the two



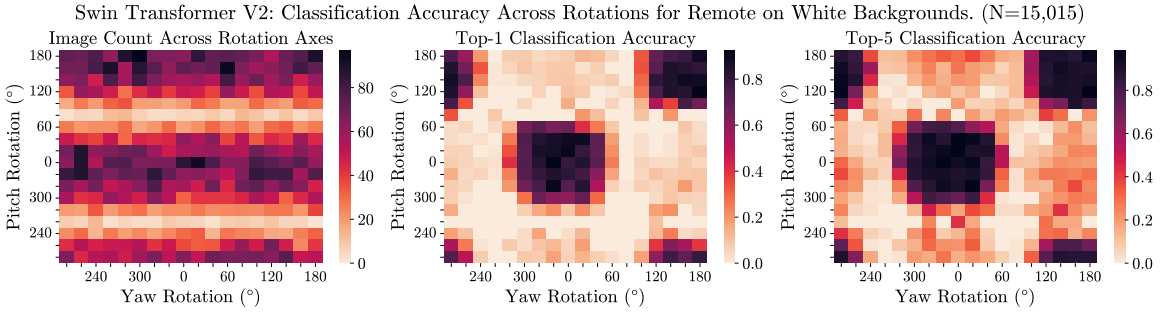
(a) *Swin Transformer V2:* Table accuracy over rotations on white backgrounds.



(b) *Swin Transformer V2:* Rifle accuracy over rotations on white backgrounds.



(c) *Swin Transformer V2:* Lamp accuracy over rotations on white backgrounds.



(d) *Swin Transformer V2:* Remote accuracy over rotations on white backgrounds.

Figure 4.6: Classification accuracy of the *Swin Transformer V2* model over varying object rotations on white backgrounds. Rotations are measured in 20° increments relative to the default orientations shown in Figure 3.5.

regions of rotation-space that present this front face to the camera. Other classes that exemplify this pattern include *microwave*, *washing machine*, *file cabinet*, and *clock*.

Based on these results it is clear that classification performance across rotation-space is heavily dependant on the class of the object. The observations made in this section suggest that classification performance is maximised when the object is in a pose that (a) is represented heavily in training data, and/or (b) presents a view of the object with many identifying features. These results validate and build upon previous results by Alcorn et al. (2019), and serve as evidence for the hypothesis that model accuracy is maximised when objects appear in their canonical pose.

4.2.3.4 Canonical pose influences rotation invariance

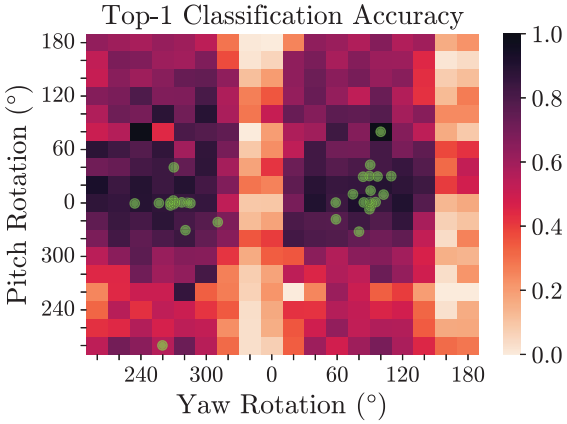
In the previous section, it was mentioned that classification accuracy on synthetic images is highest for poses that occur frequently in training data. To formalise this observation, 30 ImageNet images were randomly sampled from the *rifle* and *remote* classes of ImageNet, and their poses were manually annotated by estimating the facing direction of the object using azimuth and elevation angles. This estimation was done without referencing the accuracy heat maps shown in Figure 4.6 to avoid bias, and the estimated poses were subsequently annotated on the top-1 classification accuracy visualisations to produce Figures 4.7a.ii and 4.7b.ii.

This small sample of manually annotated images across the two classes leads to a few interesting conclusions about classification performance and the quality of the synthetic data set. Firstly, there is a relatively strong correlation between poses represented in ImageNet and regions of high classification accuracy. This again suggests that (a) poses that are heavily represented in model training data are classified more accurately in synthetic images, and/or (b) ImageNet images tend to contain the most identifying features of the classes they represent. Considering the results more carefully suggests that option (b) may be more likely. This is because many of the poses that are classified effectively in the synthetic data sets are *not* represented well in ImageNet. These include rotations that place the objects upside-down, but keep identifying features in view (e.g. (azimuth, elevation) = $(180°, 180°)$ for *remote* which maintains the front view but rotates 180° along the roll axis). The fact that this pose is classified accurately without significant representation in training data suggests that it is the presence of identifying features that is most important for accurate classifications.

Additionally, in Figure 4.7b.ii, there are multiple ImageNet *remotes* that were annotated as having pitch and yaw rotations outside of the regions of high classification accuracy. This highlights a limitation of the chosen *Spherically Distributed Pose Sampling* approach, as this method represents rotations using only two rotational axes. To illustrate the problem with this, Figure 4.8 shows a discrete representation of the *remote* rotation-space. Comparing this with Figure 4.7b.ii it is clear that regions of high classification accuracy correspond very closely with rotations that present the buttons of the remote.

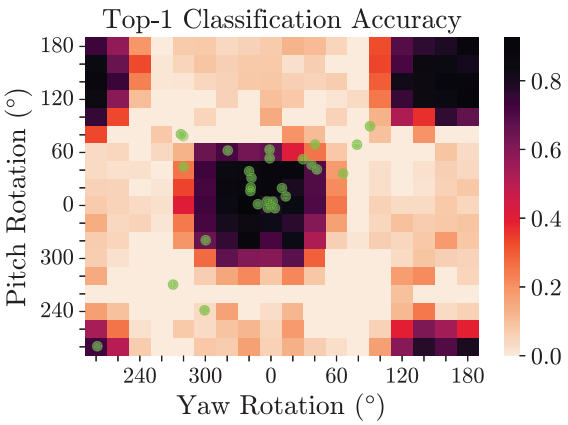
(a.i) Sample of ImageNet *rifles*.

(b.i) Sample of ImageNet *remotes*.



(a.ii) Manually annotated poses of 30 *rifles* on top-1 *rifle* accuracy over rotations.

(a) Comparison for *rifle*.



(b.ii) Manually annotated poses of 30 *remotes* on top-1 *remote* accuracy over rotations.

(b) Comparison for *remote*.

Figure 4.7: Comparison of poses represented in ImageNet vs. top-1 accuracy over synthetic images.

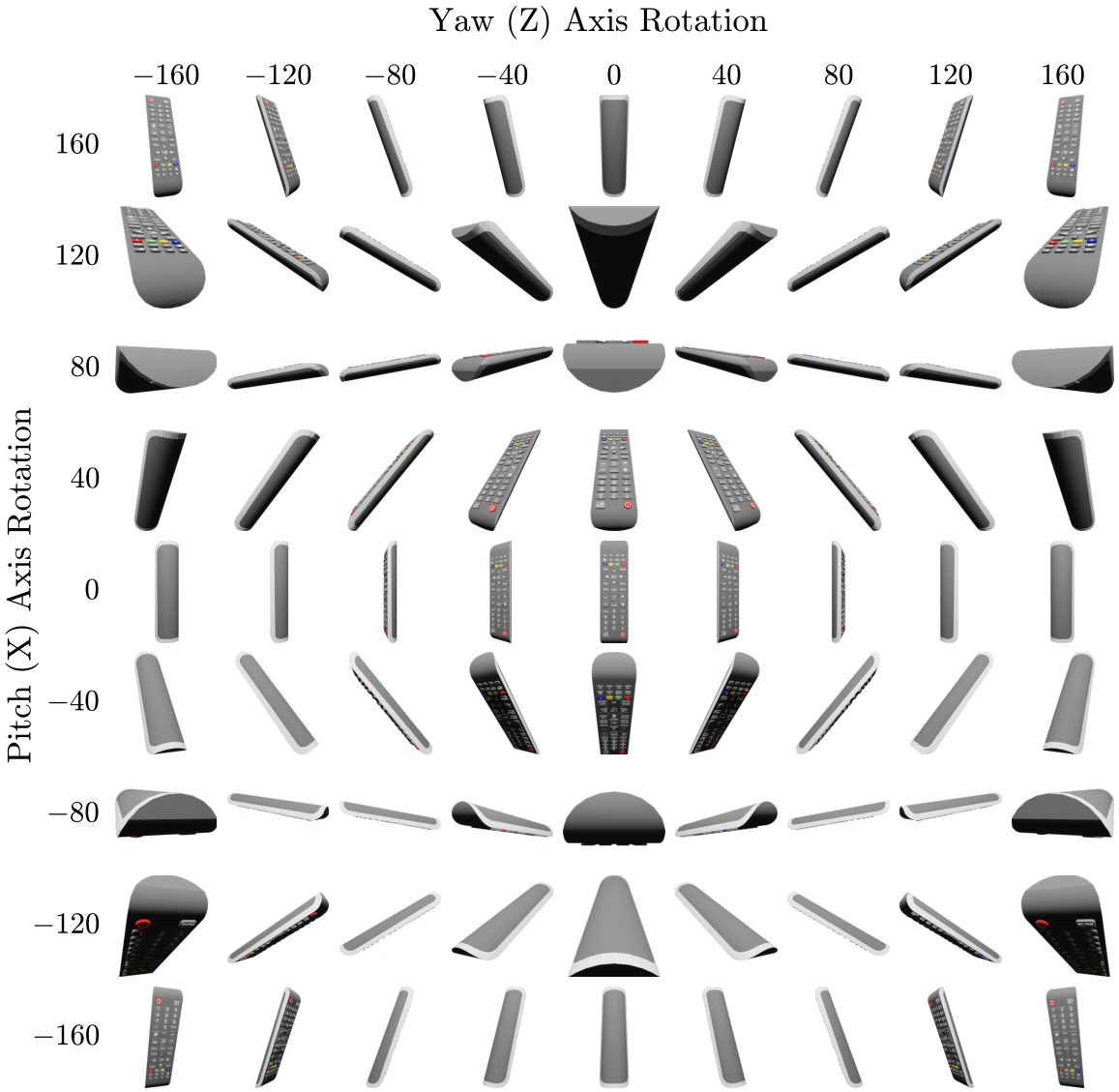


Figure 4.8: Samples of *remote* rotations across the yaw and pitch axes demonstrating the limitations of combining the *Spherically Distributed Rotation Sampling* approach with the default poses from ShapeNetCore.

The reason that real images are often classified correctly in regions of pitch-yaw rotation-space that are not classified accurately across the synthetic data set is that remotes in real images are often subject to roll rotations that bring identifying features into view. Consider, for example, the view of the remote at $(\text{pitch}, \text{yaw}) = (120°, 40°)$ in Figure 4.8. Were this remote subject to an additional roll rotation, the identifying features (buttons) would be brought into the view of the camera. This is to say that when the 3D rotation of an object in a real image is projected into the 2D pitch-yaw representation, information about the roll rotation is lost, and is not able to be recovered using the *Spherically Distributed Pose Sampling* approach. Suggestions for how to manage this in future research are provided in Section 6.2.1.

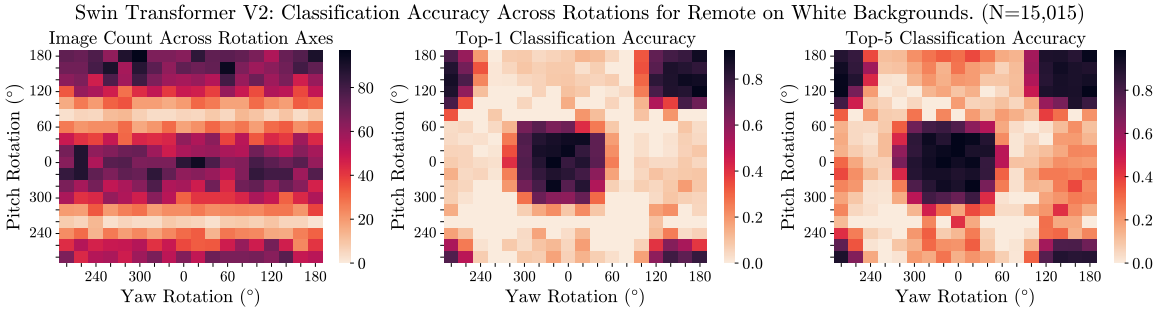
4.2.3.5 Challenging poses are shared between classification models

A key question posed in this research is whether specific classification models or computer vision architectures are more invariant to changes in the explanation parameters than others. To investigate this, the evaluation approach outlined in Section 4.1 tests four different models, with the specific variants listed in Section 4.1.1. While the results presented in Sections 4.2.3.1–4.2.3.4 specifically address the *Swin Transformer V2* model, the same visualisations are presented for the *MobileNet V2*, *ResNet*, and *MobileViT V2* models in Appendix B.1.1.

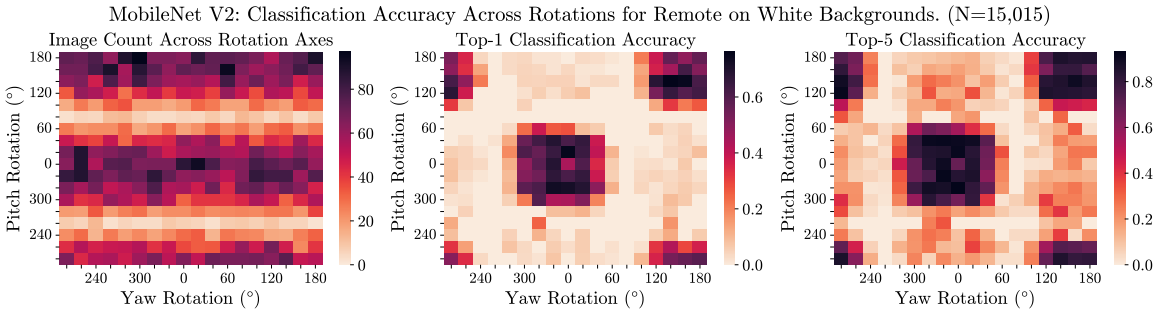
Comparing the *Swin Transformer V2* results presented in the above sections with the results for the other models presented in Appendix B.1.1 it is clear that poses that are challenging for the *Swin Transformer V2* are also difficult to classify for the other models. Looking first at Figures B.1–B.3, which correspond to Figure 4.5, all models clearly exhibit the same pattern in classification accuracy over both rotational axes. All models exhibit the same multimodal marginal distribution of accuracy across the yaw axis, with has four peaks ±30° and ±150°, and local minima at ±90°, 0°, and 180°.

As expected, a similar marginal distribution is observed across the pitch axis for the three models, showing a bimodal distribution with the largest peak at 0° pitch rotation, and a second, smaller peak centred on 180° (where the models are upside-down). Across both rotational axes the shape of the distribution is shared between models, and the magnitude of the classification accuracy is proportional to the relative accuracy of each model on the synthetic data set.

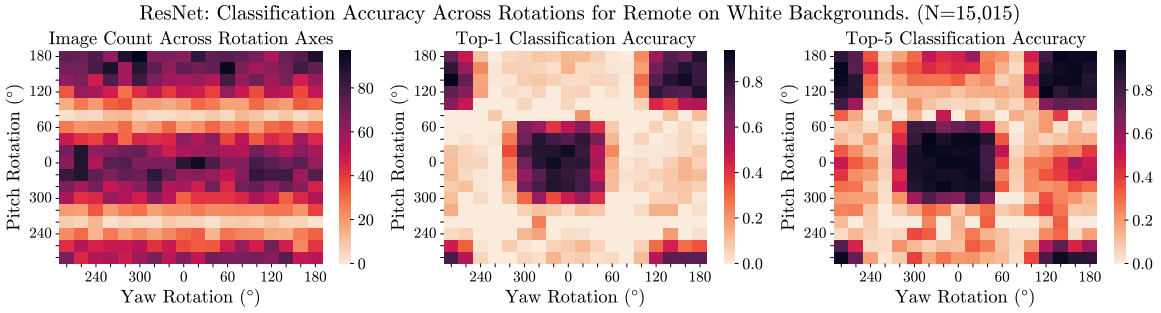
A similar result is observed when looking at the performance of each model on specific object classes. To this end, Figure 4.9 shows the performance of all models on the *remote* class across rotation space and Figures B.7–B.9 show the *table*, *rifle*, and *lamp* classes. Across all these examples, as well as additional object classes that are not presented in this report, classification models all clearly demonstrate similar patterns of classification performance across rotation-space on individual objects. This supports the conclusions presented in Section 4.2.3.4, which established that high classification accuracy is dependent on the visibility of identifying features of the object class.



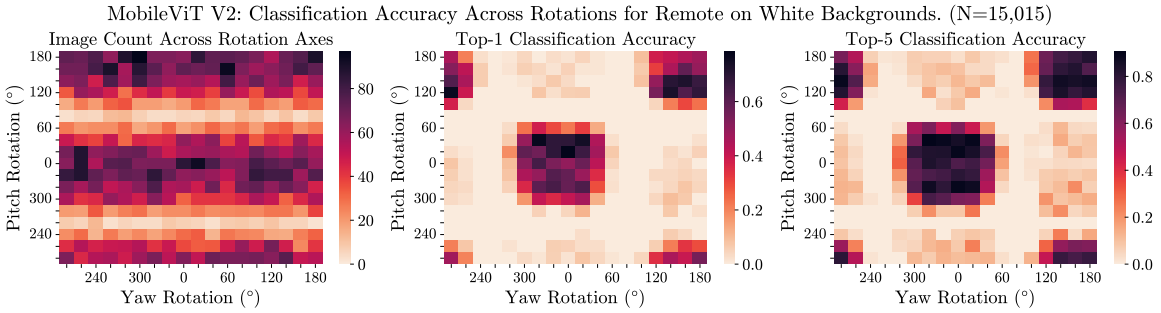
(a) *Swin Transformer V2:* Remote accuracy over rotations on white backgrounds.



(b) *MobileNet V2:* Remote accuracy over rotations on white backgrounds.



(c) *ResNet:* Remote accuracy over rotations on white backgrounds.



(d) *MobileViT V2:* Remote accuracy over rotations on white backgrounds.

Figure 4.9: Classification accuracy of all models over rotation-space for *remotes* on white backgrounds.

4.2.4 Background Invariance

The backdrop against which an object is presented plays an important role in how the overall image is perceived by image classification models. The context provided by the background can either aid or inhibit the model's ability to make correct classifications. K. Xiao et al. (2020) show both sides of this story, demonstrating that backgrounds alone are sufficient to make correct classifications in many cases, and conversely, show that confounding backgrounds may result in incorrect classifications even when the image subject is classified correctly.

In this section, state-of-the-art models are evaluated on the synthetic data sets (recall that two data sets were produced in Section 3.1, both containing equivalent objects, with one having white backgrounds and the other having backgrounds sampled from the SUN397 data set). The analysis performed validates findings by K. Xiao et al. (2020) and suggests various new conclusions about the effects of different background categories. These results are presented in the following sections.

4.2.4.1 Classification models perform better on white vs. SUN backgrounds

To begin with a simple and intuitive conclusion, this evaluation finds a significant difference in performance between synthetic images with white backgrounds, and those with backgrounds sampled from the SUN397 data set. This is shown in Figure 4.10, and the percentage decrease in accuracy for each model is shown in Table 4.3, demonstrating that mobile-sized models are relatively more sensitive to the presence of backgrounds. This provides clear support for the conclusion by K. Xiao et al. (2020) that "more accurate models ... have greater robustness to changes in image background."

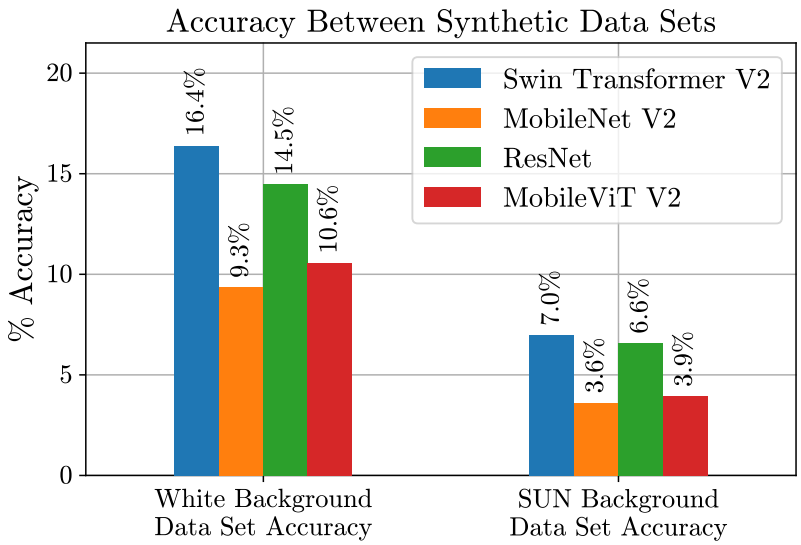


Figure 4.10: Comparison of top-1 classification accuracy for the four benchmarked models on synthetic images with white backgrounds vs. backgrounds from the SUN397 data set.

This is similar to the drop in performance observed between ImageNet and synthetic images

Table 4.3: Comparison of top-1 classification accuracy across models on the synthetic data sets with white backgrounds vs. backgrounds from the SUN397 data set.

Model	Accuracy on White BGs (%)	Accuracy on SUN BGs (%)	% Change in Accuracy
Swin Transformer V2 (Large)	16.4	7.0	-57.3
MobileNet V2 (1.4)	9.3	3.6	-61.3
ResNet-152	14.5	6.6	-54.5
MobileViT V2 (1.0)	10.6	3.9	-63.2

in Table 4.1, however in this case, the transformer-based models appear to be slightly more sensitive than their CNN-based counterparts. It is possible that the global context emphasised in transformer-based models causes them to be more responsive to image backdrops, but more research is required to verify this conclusion as the difference between the models is relatively small, and investigating their internal processes is outside of the scope of this research.

Looking at the interaction between backgrounds and image subjects in Figure 4.11 shows that the performance of the *Swin Transformer V2* model decreases in the presence of SUN backgrounds for all classes except *can*. This is a relatively unremarkable result since the classification accuracy for *cans* on *white* backgrounds was so low that even a small number of confounding backgrounds could result in the observed 165.6% increase in top-5 accuracy. While all other classes see a decrease in top-5 accuracy of between -3.3% and -82.3%, it is difficult to generate much insight from these results, as it is challenging to separate the effect of confounding objects appearing in background images. This problem is discussed further in Section 4.2.4.3.

4.2.4.2 Classification accuracy is higher on outdoor SUN backgrounds

While the SUN data set contains too many classes to produce meaningful results about the effect of individual background categories on classification performance, the hierarchical annotations of the backgrounds *do* facilitate evaluation over broader groups of background classes. The broadest category annotated on the SUN397 data set states whether each scene class is Indoor, Outdoor (Natural), or Outdoor (Man-made), and analysing the results across these features does highlight significant conclusions.

Looking first at the overall impact of these broad background categories, Figure 4.12 shows the top-1 and top-5 classification accuracy of the *Swin Transformer V2* model across all classes, broken down across the various indoor and outdoor background types. In both top-1 and top-5 cases, Outdoor (Natural) backgrounds are shown to result in significantly higher classification accuracy, followed by Outdoor (Man-made) backgrounds, which again result in better performance than Indoor backgrounds. The same trend is observed for individual object classes, including classes that would typically appear outdoors (e.g. *mailbox*; see Figure 4.13a) and those that would typically appear indoors (e.g. *remote*; see Figure 4.13b).

It is suggested that Outdoor (Natural) backgrounds result in the best classification perfor-

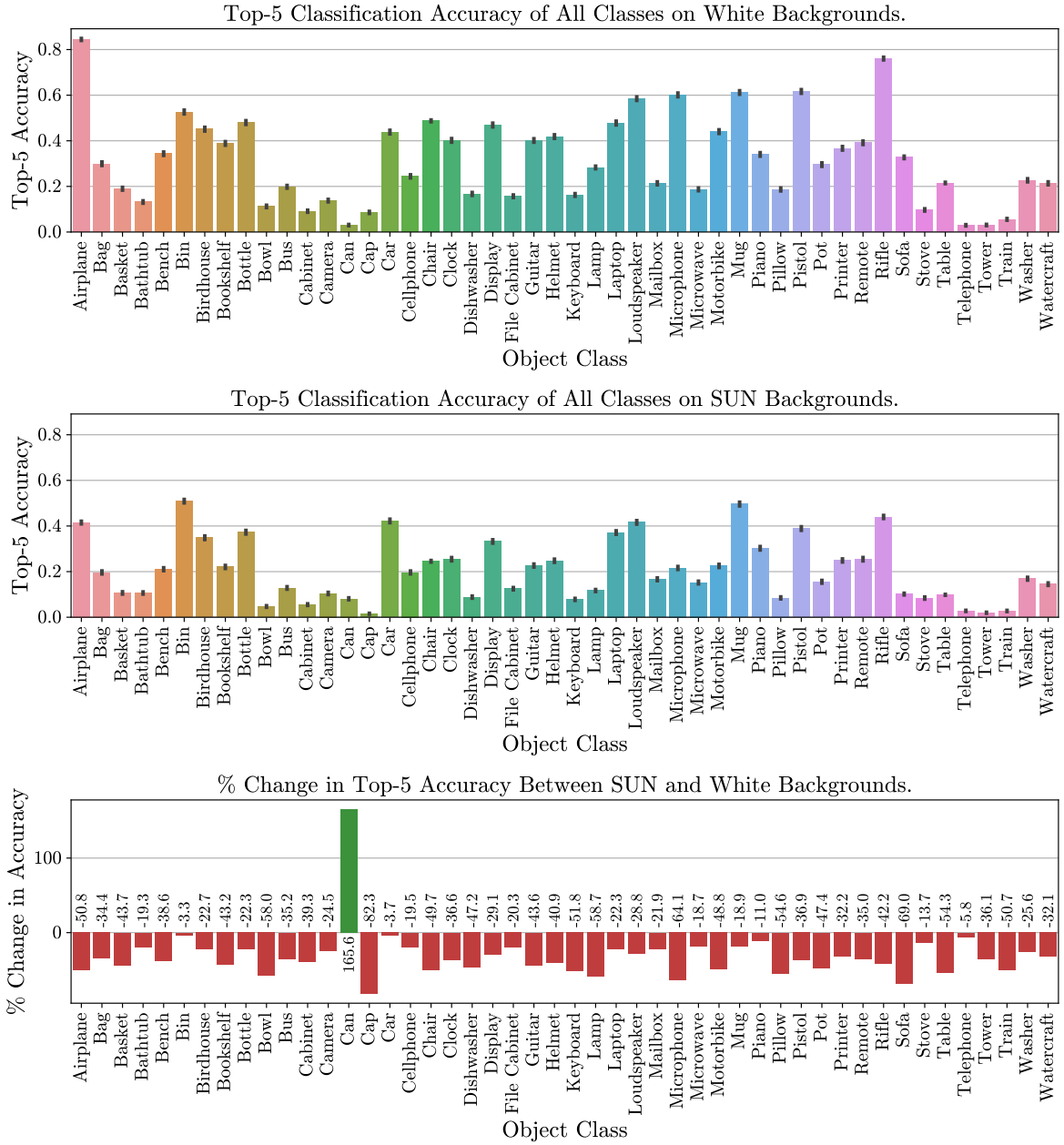


Figure 4.11: Comparison of top-5 classification accuracy for the *Swin Transformer V2* on white vs. SUN397 backgrounds, broken down by object class.

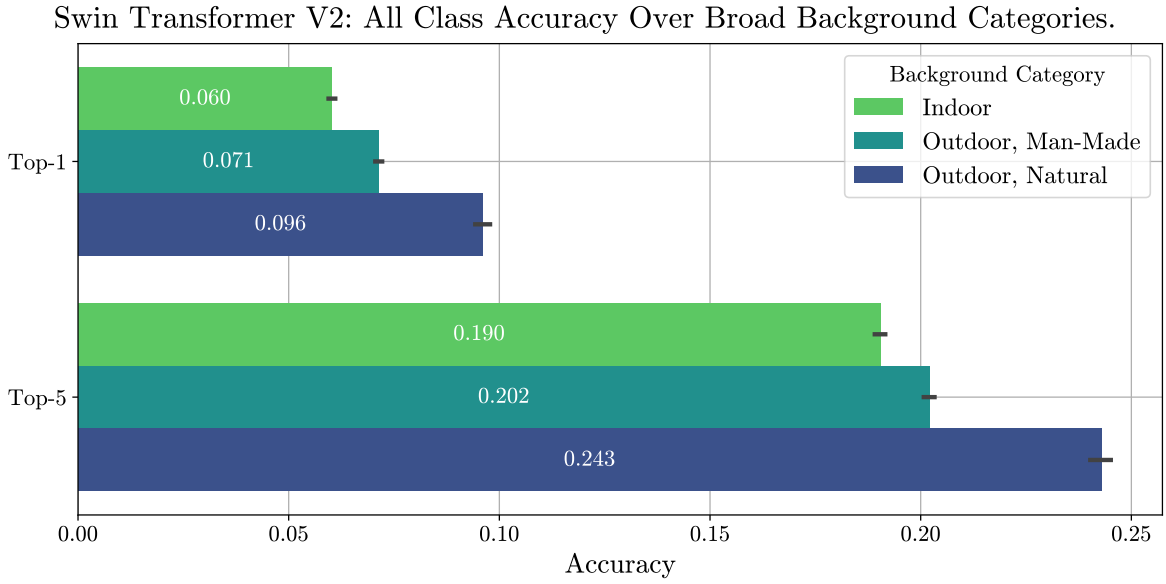


Figure 4.12: Comparison of classification accuracy across Indoor, Outdoor (Natural), and Outdoor (Man-made) backgrounds for the *Swin Transformer V2* model across the entire synthetic data set.

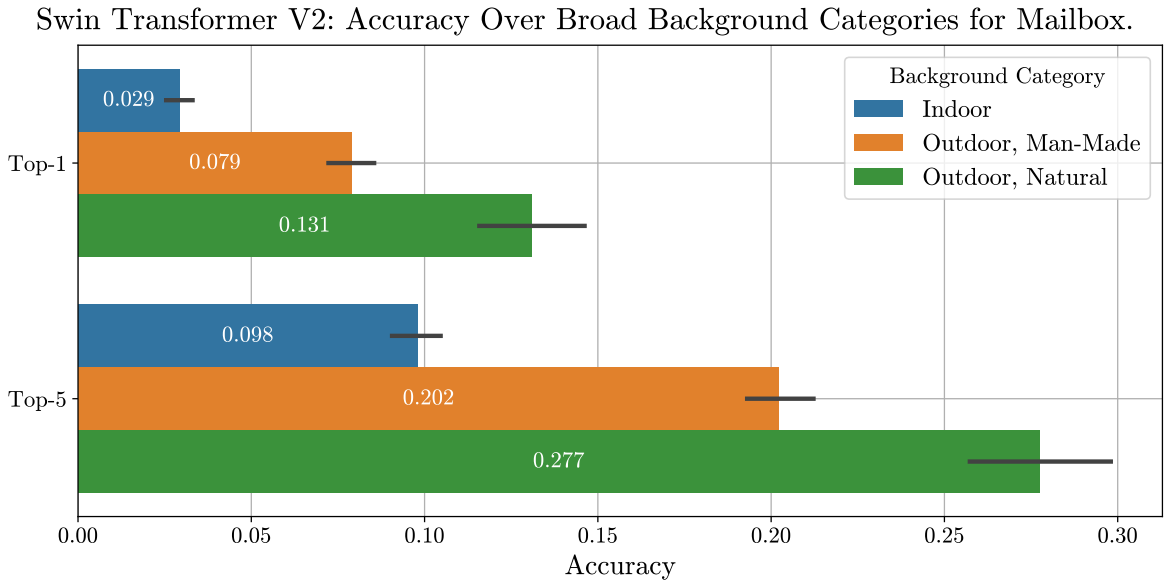
mance due to the relative absence of confounding objects and features in the background images. On the other extreme, Indoor scenes are highly likely to contain other object classes present in ShapeNetCore, which may result in incorrect classifications when they are detected by the model. To further test this hypothesis, the *sky* background class was identified as an Outdoor (Natural) scene with extremely low presence of confounding objects. This is validated in Table 4.4, which shows that none of the top-5 most common labels predicted for *sky* backgrounds in the absence of a composited subject are objects contained in ShapeNetCore.

Using this information, Figure 4.14 compares the accuracy with which all classes are classified when presented on different background types. Despite the relatively small sample size of *sky* backgrounds, classification accuracy against these backdrops is shown to be significantly higher than both the mean accuracy over all backgrounds, *and* the mean accuracy for Outdoor (Natural) backgrounds. This lends further evidence to the hypothesis that confounding objects present in backgrounds result in incorrect classifications, which is elaborated on in the following section.

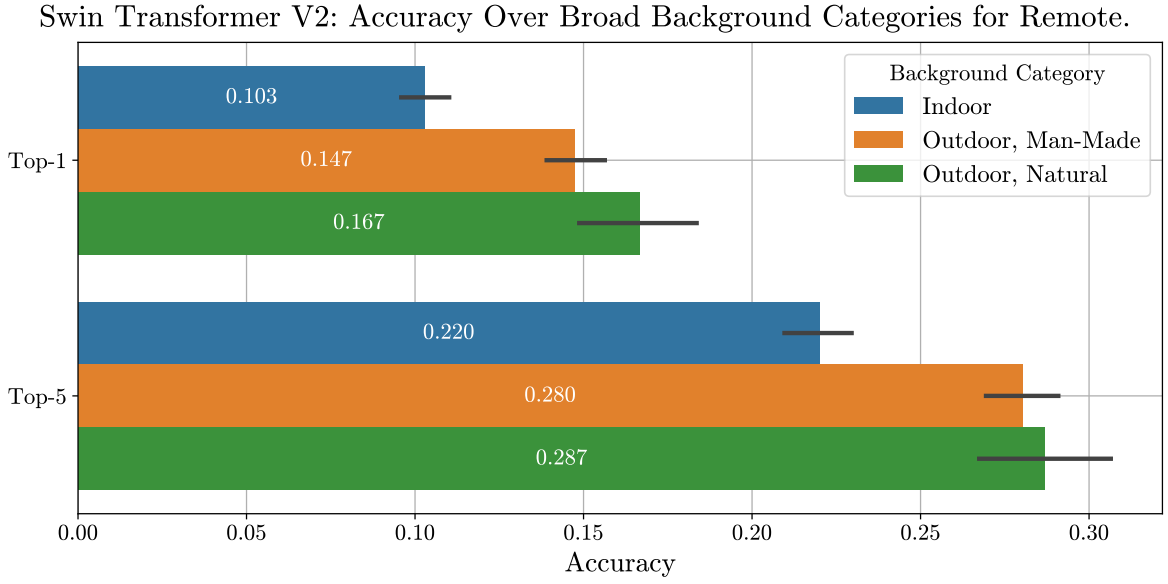
4.2.4.3 Objects present in background images cause type-II errors

Images from the SUN397 data set were used as image backgrounds due to the diverse and real-world nature of the images, however, it is these same properties that result in a highly complex synthetic data set. In the previous section, it was suggested that Outdoor (Natural) backgrounds resulted in the highest classification accuracy due to a relative absence of confounding objects and features in the images. This section provides further evidence for that conclusion, and investigates the causes of type-II (false negative) errors in more detail.

Drawing attention again to Table 4.4, it's clear that objects are more commonly identified



(a) Top-1 and top-5 accuracy for *mailbox*, a typically *outdoor* class.



(b) Top-1 and top-5 accuracy for *remote*, a typically *indoor* class.

Figure 4.13: Comparison of classification accuracy across Indoor, Outdoor (Natural), and Outdoor (Man-made) backgrounds for the *Swin Transformer V2* model on specific object classes in the synthetic data sets.

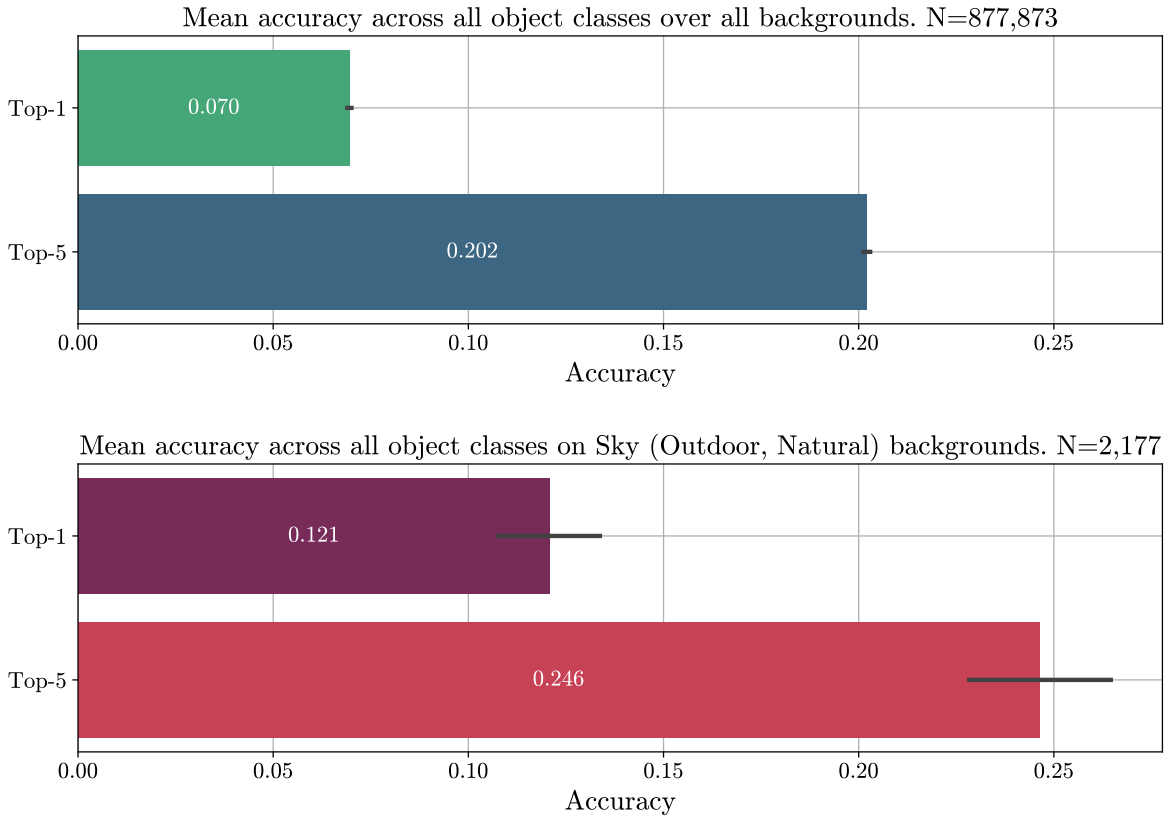


Figure 4.14: Comparison of classification accuracy across all object classes for the *Swin Transformer V2* model. The *sky* background class is generally absent of confounding objects, and results in the highest accuracy (bottom) when compared to both the average background (middle) and the average background across each broad indoor/outdoor category (Figure 4.12).

Table 4.4: Top-5 most frequent classification outputs from the *Swin Transformer V2* model on five selected background classes from the SUN397 data set.

	Top-5 Most Common Class Labels (*Swin Transformer V2*)									
BG	**1**		**2**		**3**		**4**		**5**	
Class	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5
Airport	*Airliner*		*Crane*		*Snowplough*		*Coast*		*Pier*	
	58.4%	72.7%	5.2%	15.6%	3.9%	5.2%	3.9%	25.9%	2.5%	9.1%
Cafeteria	*Library*		*Restaurant*		*Dining Table*		*Folding Chair*		*Desk*	
	38.1%	92.4%	37.1%	88.6%	12.4%	90.5%	3.8%	25.7%	1.9%	38.1%
Crosswalk	*Traffic Light*		*Street Sign*		*Taxi*		*Tram*		*Pole*	
	29.3%	66.0%	11.7%	50.0%	5.9%	27.1%	3.2%	21.3%	2.1%	23.4%
Ocean	*Coast*		*Lakeside*		*Jetty*		*Sandbar*		*Headland*	
	51.6%	88.2%	10.6%	54.7%	5.9%	40.6%	5.1%	42.1%	5.1%	44.5%
Sky	*Coast*		*Volcano*		*Geyser*		*Lakeside*		*Lighthouse*	
	16.1%	55.4%	13.1%	36.3%	11.3%	22.6%	9.5%	48.8%	7.7%	42.9%

in Indoor and Outdoor (Man-made) background classes. On the other hand, looking at the classification results for *ocean* and *sky* backgrounds shows a higher proportion of scene-type results (e.g. *coast*, *lakeside*, *headland*). As such, Outdoor (Natural) environments are less likely to be mistaken for objects in the ShapeNetCore data set. Conversely, the *cafeteria* background is often classified as *dining table* and *desk*, which are two classes that overlap with the ShapeNetCore *table* class.

To demonstrate that these issues result in type-II errors, Figure 4.15 investigates cases where *rifle* objects are misclassified as *airplanes*. The top graph shows that across the entire synthetic data set, this specific misclassification occurs as the top-1 result in 0.9% of cases. The bottom graph demonstrates that this misclassification occurs in 40.5% of cases on backgrounds that are themselves frequently classified as *airplanes*. The specific background classes selected for this analysis are listed in the figure caption.

This result is clearly significant, and while it may be expected, there are a few reasons that the magnitude of this effect is so pronounced. To explain this, it is important to consider that the composited subjects in these synthetic images appear in unusual poses, and under other challenging configurations of the explanation parameters. As such, the *Swin Transformer V2* model is less likely to identify the image subject, and more likely to respond to features in the image background. Additionally, while the composited subject is very rarely presented in its canonical pose, the SUN backgrounds almost always do present objects in a canonical orientation. This is a simple reality of using complex, real-world scene images as image backgrounds, and is one reason that simpler background data sets are recommended for future research in Section 6.1.2.

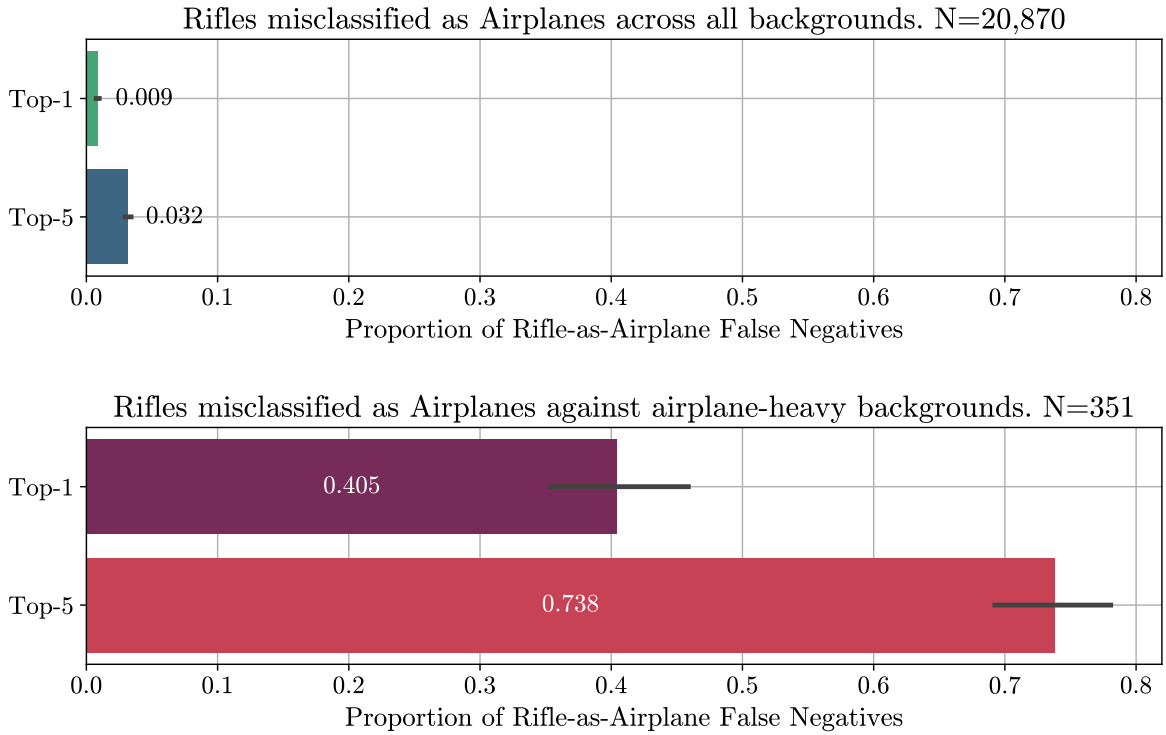


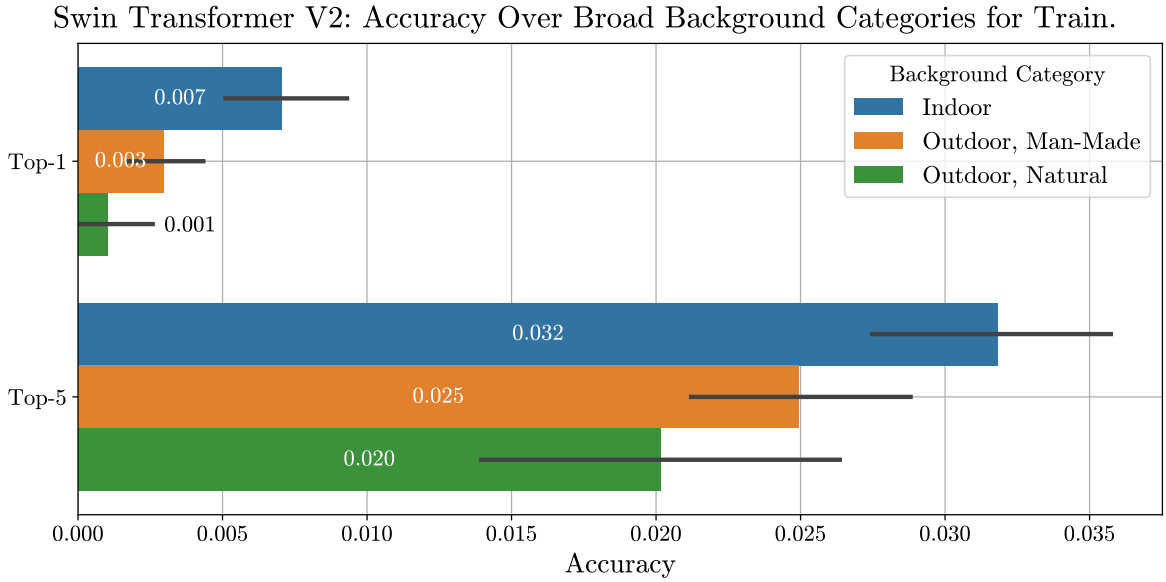
Figure 4.15: The proportion of *rifle* images that are misclassified as *airplanes* by the *Swin Transformer V2* across all backgrounds (top) and across backgrounds from classes that are already frequently classified as *airplanes*. These background classes are: *airfield, airplane cabin, airport, arrival gate, control tower, hangar indoor, hangar outdoor, heliport,* and *runway*.

4.2.4.4 Interactions between specific backgrounds and objects are inconclusive

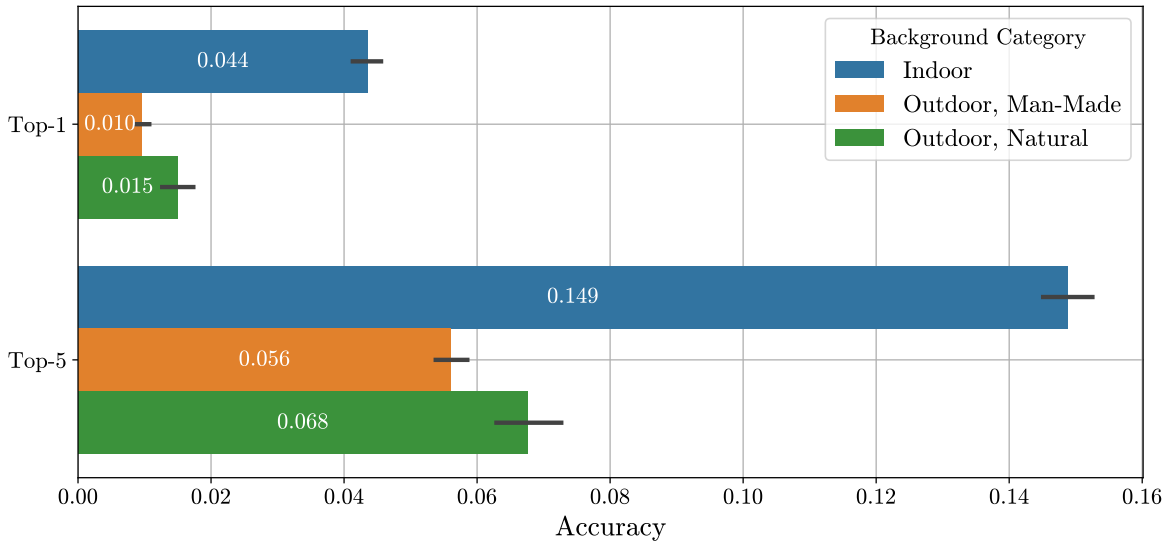
The above sections demonstrate a complex relationship between image subjects and background classes. In some cases this relationship aligns with expected results (e.g. *mailboxes* being classified better against Outdoor backgrounds in Figure 4.13a), and in others, image backgrounds containing confounding classes make it challenging to draw significant conclusions. This section rounds out the analysis of backgrounds by highlighting other sources of inaccuracy, and suggesting how they may be addressed in future research.

To this end, Figure 4.16a shows two sample classes that go against the trend of higher accuracy being observed on Outdoor (Natural) backgrounds. Looking first at Figure 4.16a, it can be seen that the *train* object class appears to be best classified indoors, however the classification accuracy on this object class is so low (averaging 0.45% top-1 and 2.7% top-5 accuracy) that these results are relatively insignificant. This is one limitation of the synthetic data set. The images are often so challenging to classify that the resulting analysis has weak predictive power.

Figure 4.16b shows that *table* is another class that is best classified against indoor backgrounds. However, as discussed above, *tables* appear in many Indoor background images



(a) Classification performance across background categories for synthetic *train* images. While this class does not follow the common trend of being best classified against Outdoor (Natural) backgrounds, the classification accuracy is so low that the results are relatively insignificant.



(b) Classification performance across background categories for synthetic *table* images. While non-trivial accuracy is achieved on this class, the results are significantly confounded by the presence of *tables* in background images. This is especially true of *indoor* scenes, as demonstrated with the *cafeteria* background class in Table 4.4.

Figure 4.16: Comparison of classification accuracy across Indoor, Outdoor (Natural), and Outdoor (Man-made) backgrounds for the *Swin Transformer V2* model on synthetic *train* and *table* images. These visualisations demonstrate some limitations of performing this analysis across such diverse background classes and objects.

(e.g. *cafeteria* shown in Table 4.4). As a result, it is recommended that future research uses background data sets that contain fewer confounding objects (see Section 6.1.2).

4.2.5 Invariance to Lighting Direction

It is well known by photographers that the position and configuration of lights in a scene (including factors like brightness, colour, and diffusion) has a significant influence on the appearance of objects in an image. Since these parameters are often varied in the real world it is critical to understand how computer vision models respond to changes in these parameters. Specifically, it is important to consider how lighting configurations both positively and negatively influence the accuracy of vision models.

When computer vision models are deployed in real-world contexts such as autonomous navigation they are bound to encounter lighting from different directions as they explore their environment. As such, this research focuses specifically on the impact of lighting *direction* on the performance of image classification models. The methodology used to gather the results presented in this section is described in Section 4.1.2. In summary, each image in the synthetic data sets is taken with a fill light of constant brightness, and a key light which is randomly positioned in one of 26 predefined locations around the model. The specific lighting positions used are described and justified in Section 3.1.4 (see Figure 3.3).

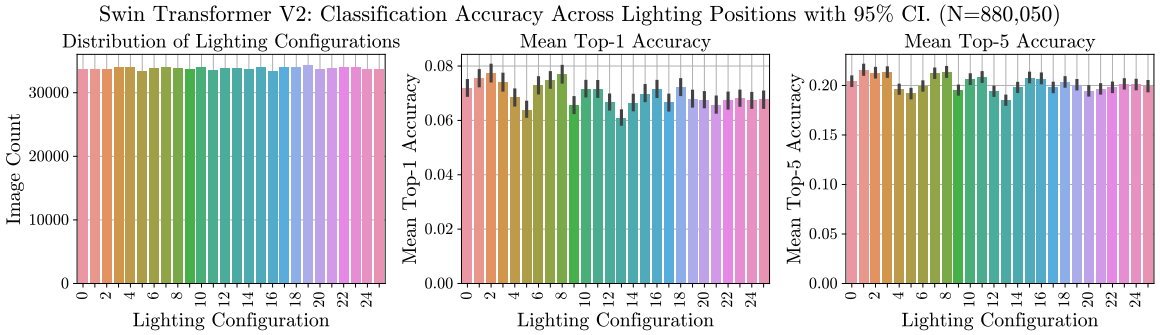
The following sections present the key findings evaluating the four selected classification models on the synthetic data sets. For the evaluation across lighting directions, significant variation was observed between classification models. As such, visualisations for the most accurate *Swin Transformer V2* model are presented in this section, and additional visualisations for the other models are presented in Appendix B.1.2.

4.2.5.1 Lighting direction significantly influences classification accuracy

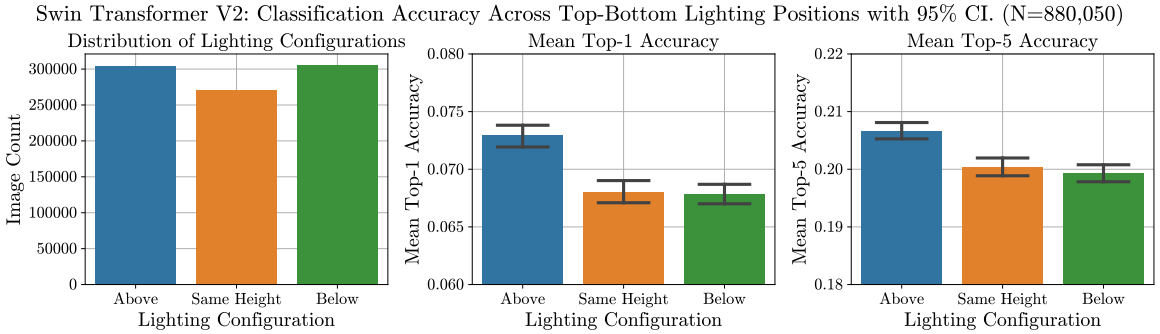
The first thing to consider when looking at the influence of lighting directions is whether varying the position of the key light impacts the classification accuracy of the model. Under the null hypothesis, the distribution of classification accuracy across lighting directions is uniform. Intuitively, Figure 4.17a shows that this is not the case. Accounting for the 95% CIs in that figure, some lighting configurations result in significantly lower classification accuracy than others. To formalise this, the goodness-of-fit of a uniform distribution was tested using a Kolmogorov-Smirnov test, which rejects the null hypothesis ($\text{KS}(\text{uniform}) = 0.83, p < 0.001$).

Looking more closely at Figure 4.17a it is worth considering which lighting configurations are most and least favourable for classification. It appears that configurations 5, 9, and 13 perform significantly worse than most other positions. Referring to Figure 3.3 we see that configuration five is directly behind the object at an elevation of 45°, 9 is directly in front of the object (behind the camera), and 13 is directly behind the object with no elevation. Samples under each of these configurations are presented in Figure 4.18.

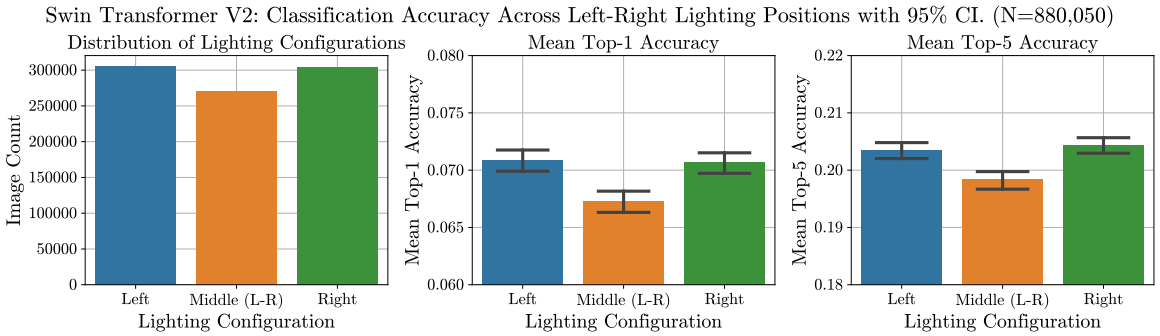
A few observations can be made from these sample images, which generalise to the rest of the



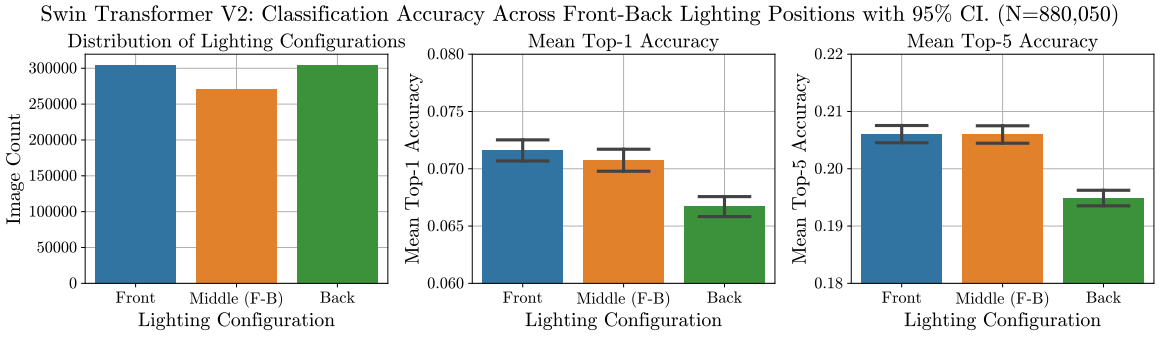
(a) *Swin Transformer V2:* All-class mean accuracy over the 26 lighting configurations.



(b) *Swin Transformer V2:* All-class mean accuracy over the Top-Bottom lighting groups.

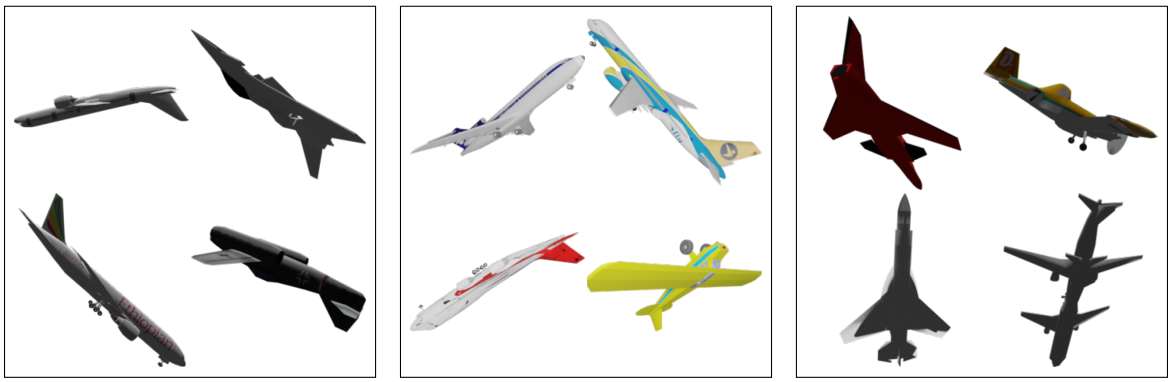


(c) *Swin Transformer V2:* All-class mean accuracy over the Left-Right lighting groups.



(d) *Swin Transformer V2:* All-class mean accuracy over the Front-Back lighting groups.

Figure 4.17: Mean classification accuracy of the *Swin Transformer V2* model over various lighting conditions on SUN backgrounds.



(a) Lighting Configuration 5 (b) Lighting Configuration 9 (c) Lighting Configuration 13

Figure 4.18: *Airplane* image samples under the worst performing lighting configurations.

synthetic data set. Firstly, looking at Figures 4.18a and 4.18c, in which the key light source is positioned behind the object, the image subjects are noticeably dark. This lack of illumination on the surfaces visible to the camera obscures key features of the objects, making them harder to classify. This is suggested to be the reason that these lighting configurations result in poor classification performance. Similarly, lighting configuration 9 (shown in Figure 4.18b) positions the key light behind the camera. In this configuration, the entire visible surface of the object is illuminated, and the lack of cast shadows makes it challenging to identify features that *would* cast shadows were the light to be coming from above.

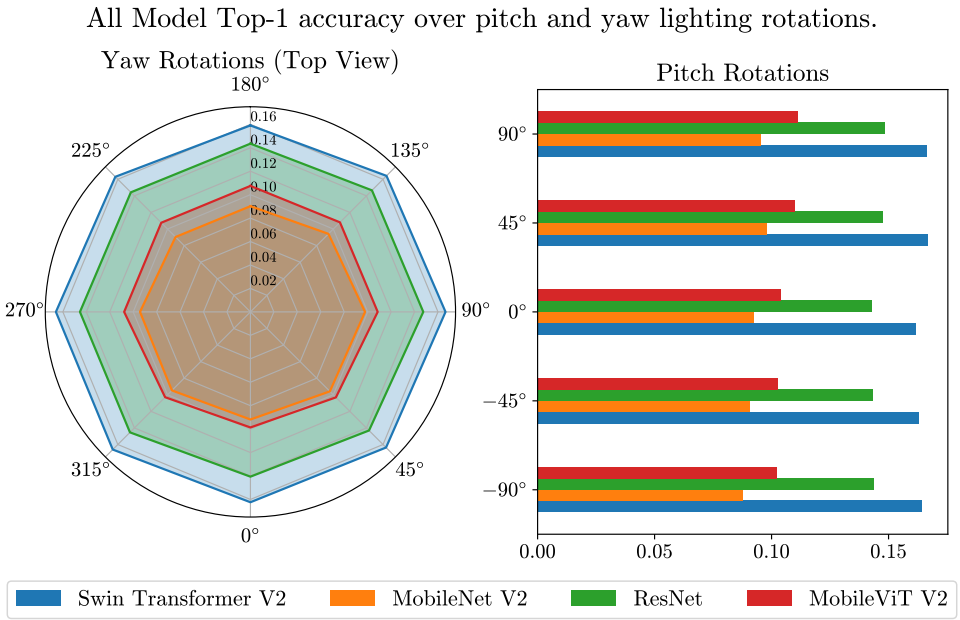


Figure 4.19: Marginal top-1 classification accuracy of all classification models over lighting rotations in the yaw axis *(left)* and pitch axis *(right)*.

This result is further validated in Figure 4.19 which shows the marginal distribution of top-1 accuracy for all four models over lighting rotations in the yaw and pitch axes. For yaw rotations, the camera is positioned at 0° and the radar chart shows a top-down perspective

of the scene. The radar chart validates that lighting from in front of and behind the object (0° and 180° respectively) result in lower classification accuracy than other rotations, with the highest accuracy being observed for all models at 90° and 270° of yaw rotation. Looking at pitch rotation, it appears that most models respond best to lighting from above, which is discussed in more detail below.

In addition to these conclusions about specific lighting configurations and angles, broader conclusions can be drawn by analysing the results over different configuration sets. By analysing these results for the *Swin Transformer V2* in Figure 4.17, and across all four models in Figures B.10–B.13 the following conclusions were drawn:

Lighting from above is favoured by all models: By identifying the nine lighting positions above and below the object, and the eight positions at an equal elevation, Figure B.11 shows the classification performance of all four models across these three groups. From this, it can be seen that all models achieve superior classification performance on images which are lit from above, compared to images that are lit from both other altitudes. This difference in both top-1 *and* top-5 accuracy is found to be statistically significant (top-1: $t(880,048) = 9.67, p < 0.001$, top-5: $t(880,048) = 9.19, p < 0.001$). Interestingly, *MobileNet V2* shows a significant preference for lighting from 0 elevation when compared to lighting from below, yet this is not demonstrated by any other models.

The symmetry of lighting configurations and object rotations in the synthetic data sets suggests that this preference for lighting from above is not caused by the illumination of identifying features (as this would be equally likely with the opposite configuration by symmetry). Instead this may suggest that the prevalence of lighting from above in real training images results in a bias towards images that are lit from above in testing.

Lighting from the sides is favoured by all models: Similar to the previous conclusion, a significant preference is observed when considering lighting that originates to the left or right of a object, versus being in the same lateral plane. Referring to Figure B.12 it is clear that all four models exhibit a preference for lighting from the left *or* right of a model when compared to the middle. It is suggested that this difference may arise due to (a) the *middle* group containing all three of the worst performing individual configurations (5, 9, and 13) outlined above, and potentially (b) lighting from outside the object's lateral plane being featured more heavily in the ImageNet training set.

When comparing lighting from the left vs. right of a model there exists no significant difference between the two groups for most of the classifiers evaluated. The sole exception for this is *MobileNet V2*, which exhibits superior top-1 classification accuracy with lighting that originates from the right ($t(880,048) = 2.21, p < 0.05$). It is recommended that this conclusion not be taken in isolation to suggest an overall preference for lighting from the right, as *MobileNet V2* does not exhibit the same significant trend in its top-5 results, nor do any other models.

Models respond differently to changing lighting depths: Finally, looking at variation of lighting positions across the depth plane (Figure B.13) some variation is observed between models. All models except *MobileViT V2* show no clear preference between lighting from in front of the object (smaller depth) and lighting in the same depth plane as the object. Across all three of these models there is a significant decrease in accuracy for lighting from behind the models.

The *MobileViT V2* model, on the other hand, shows a significant preference for lighting in the object's depth plane, but no significant preference for lighting that originates in front vs. behind the object. This research is unable to identify any specific features of this model's architecture or training process that would explain this difference, but it is observed over both the top-1 and top-5 accuracies, and is found to be significant (top-1: $t(880,048) = 4.46, p < 0.001$, top-5: $t(880,048) = 5.11, p < 0.001$).

In this section, novel results have been presented about the influence of lighting direction on the classification performance of four image classification models. It is clear that, at least on synthetic images, the position of the light source in a scene has a significant impact on the ability for classification models to produce accurate outputs. In Section 6.2, suggestions are provided for how these results could be extended in future research.

4.2.6 Scale Invariance

While object scale was not one of the primary explanation parameters investigated in this research, the compositing methodology described in Section 3.1.5 does involve systematic variation and labelling of the scale of the image subject. Intuitively it expected that images which feature their subjects at a large scale will be classified more accurately by computer vision models since (a) more pixels of the image will represent identifiable features of the image subject, and (b) fewer pixels will be representing background content that may confuse the classification algorithm or contain objects of other classes.

The following subsections present conclusions about how model performance varies as the scale of the image subject is changed. Note that this evaluation follows the methodology outlined in Section 4.1, using the data set as described in Section 3.1. As such, all images in the evaluation set are 224×224 pixels, and the scale of the image subject is varied between 90×90 pixels and 224×224 pixels.

4.2.6.1 Classification accuracy varies approximately linearly with scale

Aligning with intuition, Figure 4.20 shows how both top-1 and top-5 accuracy varies for the *Swin Transformer V2* model as the scale of the image subject is changed. For image subjects at a scale of 90×90 pixels (covering at most $\sim 40\%$ of the width or height of the full image), the model achieves a mean top-1 accuracy of around 2.5% on SUN backgrounds. For models at a scale of 200×200 pixels, the accuracy is around 10%. Fitting a linear model to predict

top-1 accuracy based on the size of image subject suggests that the relationship follows:

$$\% \text{ Accuracy} = -3.58 + 0.067 \times \texttt{Object Size (px)} \text{ for } \texttt{Object Size (px)} \in [90, 224]$$

The R^2 of this model is 0.986, suggesting that the scale of the image subject is a very strong predictor of classification accuracy for the *Swin Transformer V2*.

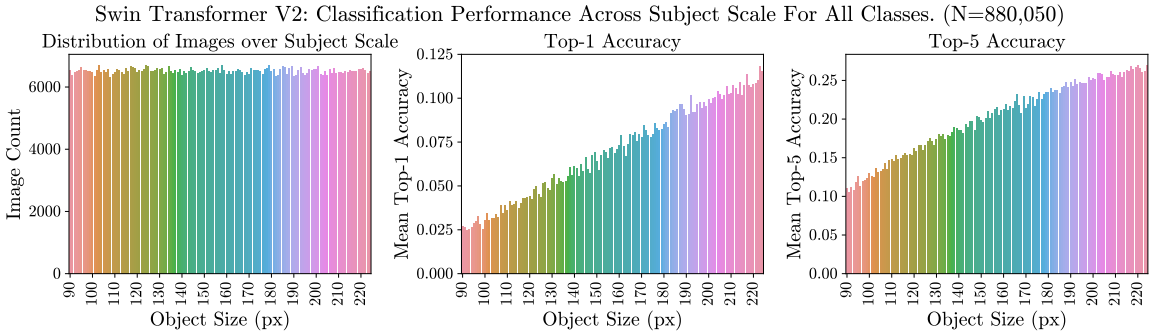


Figure 4.20: *Swin Transformer V2:* All-class mean accuracy over scale of the image subject.

4.2.6.2 Classification models respond similarly to object scale

In the previous section, it was shown that the classification accuracy of the *Swin Transformer V2* model varies approximately linearly with the scale of the image subject. By performing the same evaluation for the other three classification models, a similar conclusion is drawn. Referring to Figure B.14 a similar linear relationship is observed in both the top-1 and top-5 accuracies. Interestingly, performance of the other models appears to plateau somewhat once the image subject has reached a scale of around 200 (out of a maximum 224) pixels. At this scale it is possible that so much of the image is occupied by the image subject that further increasing the scale results in diminishing returns for classification accuracy. Alternatively, it is possible that the architecture of the *Swin Transformer V2* model, specifically designed to include detection windows at a wide variety of sizes (Z. Liu et al., 2022, 2021), is better suited to take advantage of the full scale features presented in these large-object images.

Chapter 5

Producing a Parameter-Invariant, Explainable Image Classifier

Objective 3 (Model Training) of this research aims to address the issue of invariance to changing image parameters by training an image classification model on synthetic data containing high representation of challenging image conditions. Specifically, this project aims to produce a model that is invariant to changes in object pose, image background, and lighting direction (the explanation parameters) using the data set synthesised as described in Section 3.1.

Additionally, the model produced aims to contribute to the field of XAI by implementing a novel method of image-based explanation. By training a model to predict the explanation parameters for images it classifies, these predictions can be interpreted as additional information about what the model sees in the input image. These additional outputs can then be used to synthesise an image-based explanation of what the model claims to be seeing. More details about this explanation technique are provided in Section 5.1.2.

Considering the multiple objectives of *robustness* and *explainability*, the model proposed and trained in this research is hereafter referred to as STRobE, short for Swin Transformer (Robust and Explainable). In the sections that follow, justification will be provided for the base model that was selected and built upon, then the model architecture, loss function, and training process will be outlined. Following this, in Sections 5.2.2 and 5.2.4, the STRobE model is evaluated to determine the success of the training process.

5.1 Model Design and Development

The model development process involves multiple key tasks, which are summarised in the following sections. Firstly, a base model is selected, which serves as the foundational computer vision architecture on which the STRobE model is built. Following this, the modifications made to facilitate image-based explanation are described and justified, and finally, the training process is detailed in Section 5.1.4.

5.1.1 Base Model Selection

In order to produce the most successful model within the constraints of this project it was considered most appropriate to select an existing computer vision model and modify it to suit this task, rather than to produce a brand new model. The *Swin Transformer V2* was selected as the base model for multiple reasons.

Firstly, this model achieved the best performance when evaluating the robustness of existing models in Chapter 4. The *Swin Transformer V2* model achieves the highest classification accuracy on both real and synthetic images (see Table 4.1) *and* presents the highest level of classification performance across all variations in the explanation parameters. It is hypothesised that the larger learning capacity of this model and the more complex feature representations that it learns will make it most suitable for learning invariance to the explanation parameters when fine-tuning.

Secondly, the model presents some specific architectural features that are expected to be advantageous for producing effective predictions of the parameters. The primary contribution made by the *Swin Transformer V2* on top of the *Vision Transformer* architecture is the inclusion of hierarchical feature maps that encourage the model to make greater use of global image context, rather than prioritising information localised in small image patches. This blend of local and global context is expected to provide value for predicting the explanation parameters, since understanding a complex parameter such as the pose of an image subject may require the model to look at features that are far apart in pixel space.

5.1.2 Explanatory Outputs

With the base model selected, it is next important to understand the explanatory outputs that will be added in order to guide the development of the model architecture. For this, the background content on parameter invariance, Explainable AI, and Multi-Task Learning is relevant (see Sections 2.2, 2.3, and 2.5 respectively).

While the additional outputs added to the classification model in this research are referred to as *explanatory*, they are designed to serve a dual purpose. Clearly, providing supplementary output features that describe additional image parameters increases the explanatory power of the model. However, a potential side effect of the Multi-Task Learning process is that feature representations learned for certain explanatory outputs may *also* be valuable for producing classification results that are less sensitive to variation in the explanation parameters.

Additionally, it is expected that training the model on a data set containing significant representation of unusual poses, lighting conditions, and backgrounds should increase the *invariance* of the model to changes in these parameters. This is supported by H. Yu and Oh (2021), who demonstrate that representing additional poses in training data results in greater pose invariance. Intuitively it is expected that this property may extend to the other explanation parameters.

5.1.2.1 Explanatory Model Outputs

The first explanatory component of the STRobE model is provided in the form of categorical or numeric estimates for each of the explanation parameters. By labelling each of the rendering parameters explicitly during both stages of the synthetic data generation process (see Sections 3.1.7 and 3.1.8) it is possible to train the model to estimate each of the explanation parameters using supervised learning. As such, the proposed model (whose architecture is detailed in Section 5.1.3 and shown in Figure 5.2) provides predictions for the following features:

1. Object class,
2. Image background,
3. Lighting position,
4. The 2D rotation of the object, described using azimuth and elevation angles relative to its default ShapeNetCore pose (shown in Figure 3.5),
5. The position of the image subject within the image, and
6. The scale of the image subject.

These explanatory predictions are local and model specific. There are both intrinsic and post-hoc elements of this approach. On one hand, the explanations can be interpreted as intrinsic since the model is being coerced to learn directly interpretable features. On the other hand, it can be considered post-hoc, as the internal classification process of the model remains untraceable by humans. Clearly the approach embodies elements of both paradigms.

The explanations can be interpreted in a few ways in the context of existing XAI research. One interpretation is as a form of *concept-based explanation*. In this framing, the individual explanation parameters are the *concepts*, and the model provides interpretations of the model's classification output by providing additional information as to how the instance is viewed in this concept-space (explanation parameter-space).

Alternatively, it is also possible to conceptualise these predictions as a rudimentary text-based explanation. Clearly these parameter estimates could easily be converted into an explanatory sentence by simply substituting them into a format such as *"the model sees [W object] in [X pose], against [Y background], lit under [Z lighting condition]..."*. However, this approach adds little value over presenting the parameter values in isolation. Instead, the final explanations produced by the STRobE model take a logical follow-up step, and use these parameters to synthesise image-based explanations.

5.1.2.2 Explanatory Images

To understand this step of the explanation process, it is relevant to note that the outputs of the model described in the above section are the same inputs that are supplied to the image synthesis process detailed in Section 3.1. With this information, the image-based

explanation process is simple. The outputs from the model are simply used as inputs to the image synthesis pipeline, producing a new synthetic image based on the parameters estimated by the retrained model. An example of this is shown in Figure 5.1.

(a) ImageNet *dishwasher*

```
object class: stove
background: coffee shop
yaw rotation: 180°
pitch rotation 30°
lighting conf.: 1
scale: 125px
X offset: 75px
Y offset: 125px
```

(b) Explanatory model outputs

(c) Image-based explanation

Figure 5.1: A sample image from ImageNet with its corresponding output from the explanatory model and explanatory image generator.

The explanatory image is intended to act as a visualisation of the classification model's understanding of the image. In the example shown in Figure 5.1, the model incorrectly classifies the *dishwasher* as a *stove*, and provides additional information about the background, lighting, and pose of the object it identified. While this incorrect classification result may be unintuitive when provided with only the class label or textual outputs, the image-based explanation contextualises these predictions. The synthetic image highlights similarities between the input image (where the *dishwasher* appears to be in use as a table) and the selected 3D *stove* model, which has a similar protruding top surface. While this provides some intuition for the image-based explanations produced, the technique is evaluated in more detail in Section 5.2.4.

5.1.3 Model Architecture

Inspired by other models that are built on top of the base *Swin Transformer*, the model produced for this project consists of the base *Swin Transformer V2* model with an additional lightweight prediction head. Existing research that builds upon the *Swin Transfomer* and other similar image encoders suggests that producing an effective downstream model requires only a simple prediction head that conforms to the outputs of the encoder and produces the desired outputs for the task (Xie et al., 2022). In their 2022 paper proposing *SimMIM*, Xie et al. show that a simple linear layer used as an output head is sufficient to perform the complex task of Masked Image Modelling (MIM). They additionally suggest that the prediction head can be of arbitrary form while maintaining its level of performance.

Based on these findings, the prediction head used in the STRobE model is structured as shown in Figure 5.2. The custom output heads are joined to the *Swin Transformer V2* model by detaching the existing ImageNet classification head, and replacing this with a custom multi-task network. Note that an additional hidden layer is used in the object classification head

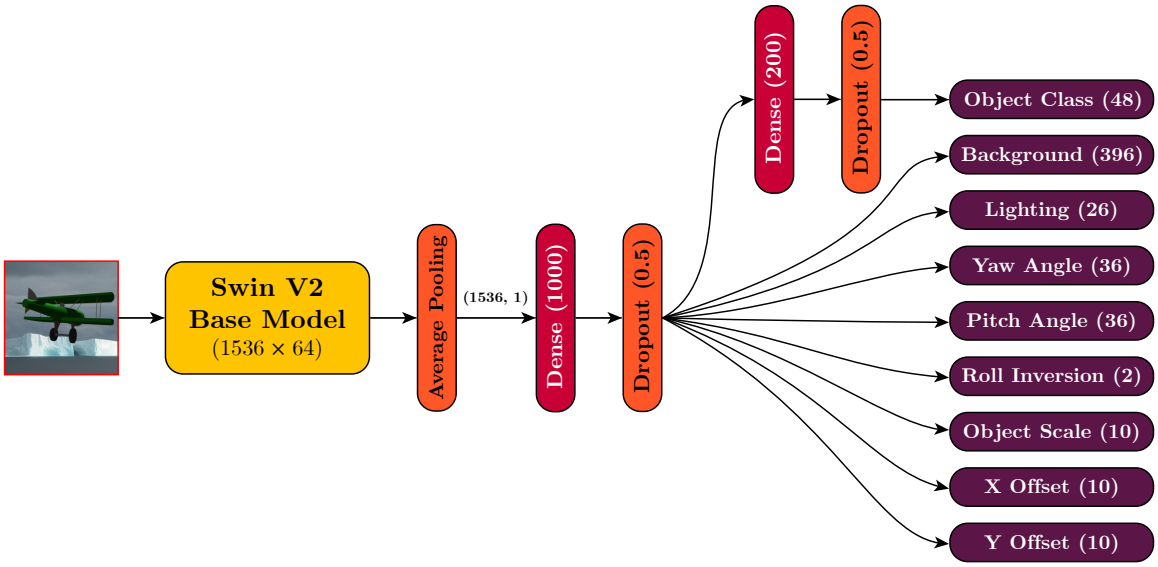


Figure 5.2: A visual representation of the STRobE model architecture.

to facilitate the multi-data set training approach described in Section 5.1.4.

This specialised network takes the 1536×64 dimensional latent representation output by the *Swin Transformer V2* base model and starts by performing average pooling to reduce the dimensionality to 1536×1. Following this is a dense layer of 1000 neurons (using ReLU activation and 50% dropout) that is shared among all prediction heads. As Figure 5.2 shows, the output from this layer then branches to each of the individual output heads. These task-specific output heads are designed as follows:

Primary Image Classification Head: The classification head responsible for predicting the class of the image subject (out of the 48 classes that are shared between ImageNet and ShapeNetCore), involves an additional dense layer with 200 neurons (also using ReLU activation with 20% dropout). On top of this is the linear output layer, producing the classification result among the 48 object classes. This output head is structured with an additional hidden layer to provide more parameters for training with the multi-stage training process described in Section 5.1.4.

Discrete Explanatory Outputs: In addition to the primary object classification head, there are two additional output heads that perform a traditional classification task on the image background and lighting configuration of the scene. For these tasks, linear output layers are attached directly to the previous dense layer, and the number of nodes in each head was selected as follows:

Background (396): The background classification head contains 396 neurons corresponding to the 396 classes containing more than 100 suitably sized images in the SUN database[1]. In hindsight, it may have been beneficial to select a smaller data set of

[1]The remaining 1 class out of the 397 did not contain 100 images of at least 224×224 pixel resolution.

background classes, or a subset of this 396 to simplify this task.

Lighting (26): The lighting classification head has 26 neurons corresponding to the 26 lighting configurations used in the synthetic image data set. The distribution of lighting configurations is discussed in Section 3.1.4 and shown in Figure 3.3. It is hoped that this output head will learn to predict the approximate location of the key light source in a scene.

Continuous and Large Range Explanatory Outputs: The other output heads are responsible for producing the remaining explanatory outputs that will describe the position and orientation of the image subject. Since the rotations and offsets are either continuous, or discrete numeric values with large ranges, a choice must be made to either treat the tasks as continuous and use regression models, or to discretise the continuous values and treat them as classification problems.

Generally, discrete classification tasks are simpler when the prediction function to estimate is complex, however discretising the data results in a loss of information that is undesirable in some cases. In this model, these outputs are discretised to simplify the task and consequently increase the chance of training a successful model within the time constraints of the project. This means that the remaining output heads are also categorical, and the amount of bins (neurons) selected for each remaining output head is chosen to balance task difficulty with information loss as follows:

Pitch and Yaw Rotation Angle (36): When combined, the pitch and yaw rotation angles define the facing direction of the image subject. When labelling poses during the image synthesis process (as described in Section 3.1.3) a two-variable representation was used to minimise the complexity of evaluating existing models. This two variable representation was defined as:

$$\text{(azimuth, elevation)} \in \{(r_a{}^\circ, r_e{}^\circ) \mid r_a \in [0, 360], r_e \in [0, 360]\}$$

However, when using the same data set to train the STRobE model, it is considered more appropriate to use a three-variable representation containing a binary roll parameter to reduce the complexity of poses with a pitch rotation in the 90°–270° range (where the object is *'upside down'*). Under this new representation, an equivalent set of rotations are represented using three variables as:

$$\text{(azimuth, elevation, roll)} \in \{(r_a{}^\circ, r_e{}^\circ, r_r{}^\circ) \mid r_a \in [0, 360], r_e \in [-90, 90], r_r \in \{0, 180\}\}$$

This representation was selected since it simplifies the notion of a facing direction as the model no longer needs to consider the impact of an object with a pitch rotation that brings it across the vertical axis. Instead a simple binary output is used to represent whether the object has been subject to a 180° roll (equivalent to adding 180° to both the azimuth and elevation angles).

Following this change, the number of bins for the rotation parameters was set to 36 so that each *yaw rotation* bin spans 10°. The number of bins for pitch rotation was then set to the same value. 10° was considered a suitable granularity to produce explanatory images that would correspond convincingly to the input image.

Roll Rotation (2): As specified in the previous step, a roll rotation flag is used to simplify the representation of poses. The roll rotation takes a value of either 0° or 180°, and as such, this is treated as a two-way (binary) classification task. While binary classification can be achieved using a single output neuron with sigmoid activation, two neurons are used in this model so that Categorical Cross-Entropy Loss can be used. The complete loss function is described in Section 5.1.4.2.

Object Scale (10): As outlined in Section 3.1.5, the image compositing process places rendered object images onto randomly selected backgrounds at a randomised scale and location. In the synthetic data sets, this random scale is sampled uniformly from the range 90–224 pixels. When training a model to predict these values it was considered appropriate to discretise this range into 10 bins, simplifying the task of predicting the object's scale. As such, this output head is a linear layer with 10 neurons.

X and Y Offset (10): Finally, the compositing process requires two offsets describing the distance of the image subject from the left and top of the image. These offsets follow the distribution shown in Figure 3.10b. Again, these values are discretised into 10 bins for the predictive model, where each bin spans between 13 and 14 pixels. This granularity is considered appropriate for generating informative explanatory images.

With the structure of the output head determined, the model was constructed in PyTorch (Paszke et al., 2017). The specific variant of the *Swin Transformer V2* model used for training was the same version that performed best in Objective 2 (Existing Model Evaluation) (the *Swin Transformer V2 (Large)* model; Z. Liu et al., 2022). Once the model was implemented, the training process proceeded as outlined in the following section.

5.1.4 Training Process

Given the multi-task nature of the STRobE model and the use of synthetic data, the training process draws on considerable existing research into MTL and Parameter-Efficient Fine-Tuning (PEFT). While initial experiments used a simpler training process, involving fine-tuning only on the synthetic data set, this simple approach ran into multiple issues. These are summarised and addressed in the following paragraphs.

Model Size and Computational Requirements: A key issue encountered during training is the size of the selected base model, the *Swin Transformer V2 (Large)*. While the large size of this model is a key contributor to its high performance, it also makes the process of retraining all parameters of the model time-consuming. To alleviate this and achieve faster training, PEFT is used, and the specific PEFT method implemented in this research is Low-Rank Adaptation (LoRA), proposed by Hu et al. (2021).

This approach, originally applied to Large Language Models (LLMs), suggests that rather than retraining all parameters in a transformer-based model, the pre-trained parameters can be frozen and instead, new trainable transformation matrices can be injected into the transformer's attention mechanism. If these matrices are optimised instead of the entire model, the number of trainable parameters is drastically reduced while maintaining fine-tuning quality. In this research, LoRA is implemented using HuggingFace's PEFT library (Wolf et al., 2020), which brings the number of trainable parameters down from 197,476,306 (100%) to 5,701,820 (2.84%).

Catastrophic Forgetting: Another problem with the simple training approach tested initially is known as *catastrophic forgetting*, which describes how neural network models, when trained on multiple tasks sequentially, forget how to perform the initial task (French, 1999). Various approaches have been proposed to solve this problem (see Kirkpatrick et al., 2017; Serra, Suris, Miron, and Karatzoglou, Serra et al.), and the one implemented in this research is based on the widely used *replay* method (Hayes, Kafle, Shrestha, Acharya, & Kanan, 2020; Kemker, McClure, Abitino, Hayes, & Kanan, 2018).

To implement replay into the training process of the STRobE model, training alternates between synthetic images and ImageNet images in each batch. The model first trains on a batch of synthetic images, using a loss function and optimiser that updates the weights of the model, *including* the explanatory output heads. Following this, the model trains on a batch of ImageNet images, with a loss function and optimiser that *exclude* the explanatory output heads. Since the *Swin Transformer V2* model is pre-trained on ImageNet, the ImageNet batch provided in each epoch reinforces the existing features learned by the model, reducing the potential for catastrophic forgetting (Hayes et al., 2021).

While there exist many more complex methods for implementing replay, such as replaying latent representations from the hidden layers of the model, this simple approach based on replaying raw inputs is known as *partial replay* and has been used in many successful incremental learning models (e.g. Castro, Marín-Jiménez, Guil, Schmid, & Alahari, 2018; Rebuffi, Kolesnikov, Sperl, & Lampert, 2017). Since implementing this for the STRobE model requires multiple data sets, loss functions, and optimisers, as well as a non-standard training process, details for all of these elements are provided in the following sections.

5.1.4.1 Training Data

Synthetic Data Set: Preparation of the synthetic data set is relatively straightforward. The only preprocessing step required is to address the class imbalance shown in Figure 3.7. Since there are only a few classes that have more than the defined minimum of 15,000 images, images are randomly discarded from classes that contain more than this amount of instances. The resulting data set has exactly 15,000 images for each of the 48 object classes. These are loaded using a custom data set class that (a) applies the image preprocessing steps used by Z. Liu et al. (2022) for the original *Swin Transformer V2* model, and (b) loads and

discretises the explanatory outputs into the format described in Section 5.1.3.

ImageNet Data Set: Considering the unique nature of the task being learned in this research, the ImageNet data set also requires minor preprocessing prior to use. This involves discarding the classes from ImageNet that are not *also* present in ShapeNetCore. This relies on the mapping between classes defined in Appendix A.1. The images are then loaded using a custom data set class that attaches the class labels and applies the *Swin Transformer V2* image preprocessor.

5.1.4.2 Loss Functions

The loss functions serve as the optimisation objectives during model training, quantifying the difference between the model's output and the desired labels. For the STRobE model, which utilises multiple output heads and completes diverse prediction tasks, the loss functions define which tasks should be prioritised by assigning different weights to each task. As mentioned above, one loss function is required for training the explanatory outputs, and a second is required for training the classification head on ImageNet (the *replay* process). The following paragraphs present the selection of loss metrics and provide justification for the relative weight assigned to each task.

Multi-task Loss: The multi-task loss function is used when training on synthetic data to optimise all trainable weights of the model simultaneously. Considering that all the output heads are categorical (as justified in Section 5.1.3), the choice of loss function is relatively straightforward. The most commonly used loss function for multi-class classification problems is *Categorical Cross-Entropy Loss* (CCEL), defined as:

$$\text{CCEL} = -\sum_{i=1}^{C} y_i \log(p_i)$$

where C is the number of classes, y_i is the true label (1 if the instance belongs to class i, otherwise 0), and p_i is the predicted probability of the instance belonging to class i. In specific contexts, other loss functions may be more appropriate for certain modelling tasks, however all outputs in this model are relatively simple and the data set was specifically designed to contain features that are uniformly distributed. As such, there was no anticipated need to deviate from this popular choice for any of the output heads.

With the loss function determined for each individual output head, the multi-task loss function (defining the overall loss that will be optimised by the model) can be produced. The multi-task loss function is defined as a simple linear combination of the losses on each individual task. For this, weights were determined for each output based on the perceived importance of each feature to the ultimate objective of producing valuable explanatory images.

In this implementation, it was considered most important that the model correctly predicts the class of the image subject, however, suggestions for how this may be approached in

future work are provided in Section 6.3.1. As such, the primary object classification head is assigned a relative weight of 1. Predicting the explanation parameters is considered the second most important objective, therefore the heads for predicting the background, lighting configuration, and rotations of the object are given a relative weight of 0.5. Finally, the position and scale of the image subject on the composited background is considered least important, and these parameters are therefore given a relative weight of 0.25. The overall multi-task loss function is thus defined as:

$$\begin{aligned}\text{Multi-task Loss} = 1 &\times \text{CCEL}(\text{Object Class}) \\ &+ 0.5 \times \text{CCEL}(\text{Background}) \\ &+ 0.5 \times \text{CCEL}(\text{Lighting Configuration}) \\ &+ 0.5 \times \text{CCEL}(\text{Yaw Angle}) \\ &+ 0.5 \times \text{CCEL}(\text{Pitch Angle}) \\ &+ 0.5 \times \text{CCEL}(\text{Roll Inversion}) \\ &+ 0.25 \times \text{CCEL}(\text{Object Scale}) \\ &+ 0.25 \times \text{CCEL}(\text{X Offset}) \\ &+ 0.25 \times \text{CCEL}(\text{Y Offffset})\end{aligned}$$

Single-task (Classification-only) Loss: The loss function used for training the primary classification head on ImageNet is comparatively simple. For this task, the Categorical Cross-Entropy Loss is measured for the predicted object class, and the other outputs are ignored (as labels do not exist for these outputs in ImageNet). The weight of this task is set at 1, equal to the weight assigned in the multi-task loss function, however, tuning this parameter may prove valuable in future research. With this weight, the classification-only loss is defined as:

$$\text{Classification-only Loss} = 1 \times \text{CCEL}(\text{Object Class})$$

5.1.4.3 Implementation Overview

With the data sets and loss functions established, the training process (including LoRA-based PEFT) proceeds as follows:

1. The data sets are initialised as described in Section 5.1.4.1.
2. The loss functions are created as described in Section 5.1.4.2
3. The model architecture described in Section 5.1.3 and shown in Figure 5.2 is initialised using PyTorch (Paszke et al., 2017), using the `microsoft/swinv2-large-patch4-window12to16-192to256-22kto1k-ft` base model, pre-trained on ImageNet.
4. LoRA is applied to the model to reduce the number of trainable parameters. The

LoRA hyperparameters are set at $r = 20$ and $\alpha = 20$ based on the original research by Hu et al. (2021). A dropout rate of 0.2 is also applied to the LoRA layers to prevent overfitting.

5. An Adam optimiser is created for training on synthetic data. This optimiser applies to all the trainable parameters of the model and is initialised with a learning rate of 0.01. LoRA promotes the use of a relatively large learning rate, which is reduced by 50% every three epochs using a learning rate scheduler.

6. A second Adam optimiser is created for the ImageNet task with the same learning rate and scheduler. This optimiser applies only to the parameters in the primary object-classification head and the shared trainable layers of the network. The explanatory output heads are *not* affected by this optimiser.

7. Training proceeds for 15 epochs, distributed among three Nvidia A100 GPUs. In each epoch, batches of size 3×32 (32 instances per GPU) are processed simultaneously from each data set.

    7.1. The synthetic batch is processed first, followed by the ImageNet batch, each using the relevant optimiser to update the model weights.

    7.2. Loss and accuracy are logged for each task on every training batch.

    7.3. At the end of each epoch, the model performance is evaluated on both synthetic and ImageNet validation data sets.

    7.4. Learning rates are halved every 3 batches.

8. The final model weights are saved and exported.

Following the completion of this training process, the retrained model was evaluated based on both its classification and explanatory performance. The evaluation method and metrics are presented in the following section.

5.2 Evaluating the STRobE Model

To determine the success of the model development and training process, the final stage of this research is to evaluate the performance of the STRobE model and the degree to which it achieves its intended objectives. A comprehensive evaluation of the model first requires an assessment of its performance, including the degree to which it is invariant to the explanation parameters. This evaluation will facilitate conclusions about the effectiveness of the synthetic data set and how it enables the training of robust classification models.

Following this, is it important to assess the value provided by the explanatory outputs of the model. This includes both an evaluation of the raw explanatory predictions that STRobE produces, and of the explanatory images that are produced when using these predictions as

input to the image synthesis pipeline. The methodology and results of this evaluation across both objectives is outlined in the following sections.

5.2.1 Methodology for Evaluating Performance and Parameter Invariance

As a primary objective of the retraining process, the STRobE model was first evaluated on its robustness to variation in the explanation parameters. This was done by evaluating the model on a held-out test set of synthetic images, however, an alternative evaluation process is proposed for use in future research in Section 6.3.2.

Since the STRobE model performs its primary classification task across the 48 classes that intersect between the ShapeNetCore and ImageNet data sets, it could not be directly compared with the *Swin Transformer V2* model, which performs 1000-class classification on ImageNet labels. To perform a fair comparison between the models, a reduced version of the *Swin Transformer V2* was produced, which maps the outputs of the original model into the output-space of the STRobE model and discards predictions of other classes. This 48-class variant of the pre-trained model is hereafter referred to as the *Reduced Swin V2* model.

These models are evaluated on a test set of 131,213 synthetic images that were not included in the training or validation data of the STRobE model. This evaluation follows a methodology very similar to the one used for evaluating the four existing models, which is detailed in Section 4.1.2. The only differences here being that no test set with white backgrounds is used, and a different set of models are evaluated.

By collecting the top-5 predictions of each model over the test set, additional result columns are engineered which label the top-1 and top-5 correctness of each model on each image. These are used to produce visualisations and derive insights about the accuracy and parameter invariance of the models in the following section.

5.2.2 Classification Performance and Parameter Invariance Results

In this section, the results from the model training process are presented. Due to time constraints on the project, the STRobE model was unable to train to convergence, training for only three epochs on each of the synthetic and ImageNet training data sets. Because of this limitation, it unlikely that the evaluated weights result in optimal performance. Nonetheless, the accuracy of the STRobE model across each of its outputs is presented in Section 5.2.2.1. While it is very likely that the model would perform better with further training, we demonstrate that even in this state, the STRobE model is significantly more invariant to the explanation parameters than the *Reduced Swin Transformer V2*. These results are shown in Section 5.2.2.2.

5.2.2.1 Accuracy and Bias

Figure 5.3 shows a comparison of the classification accuracy between the *Reduced Swin Transformer V2* and STRobE models on the synthetic data test set. It is clear from this visual-

isation that the STRobE model performs more accurate classification for the vast majority of object classes. This is supported by the overall accuracies of the two models presented in Table 5.1.

Model	Top-1 Accuracy with 95% CI	Top-5 Accuracy with 95% CI
Reduced Swin Transformer V2	20.85% (20.63, 21.07)	42.86% (42.59, 43.13)
STRobE	35.26% (35.00, 35.52)	69.87% (69.61, 70.10)

Table 5.1: The overall accuracies of the STRobE and *Reduced Swin V2* models on the synthetic test set.

It is suggested that part of the reason for this increased performance is that the STRobE model is trained on similar synthetic images, transferring the model more effectively to the synthetic domain. However, in addition to this, the results in Section 5.2.2.2 demonstrate that the STRobE model is significantly more invariant to changes in the explanation parameters, which also results in a considerable performance increase.

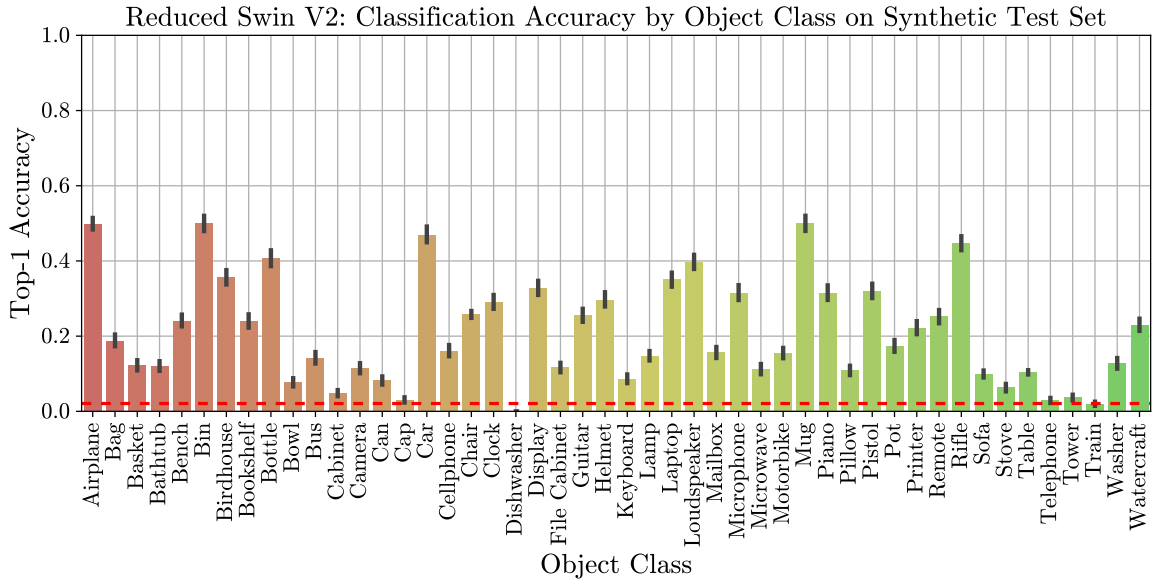
Neither model is significantly biased in their distribution of predicted class labels, however, there are some explanatory outputs of the STRobE model for which biases and inaccuracies remain after the three epoch training run. Figure 5.4 demonstrates this for rotations, showing that when predicting the roll rotation of objects, the model tends to over-predict that models are in an upright orientation (Figure 5.4a).

Similarly, Figure 5.4b shows that there are certain pitch and yaw rotation increments that are rarely predicted by the model. Despite this, the performance of the model at predicting facing directions is promising, with the diagonal lines in the confusion matrices suggesting that predictions are often correct, but are sometimes off by 180°, and that models are sometimes misclassified as their reflection.

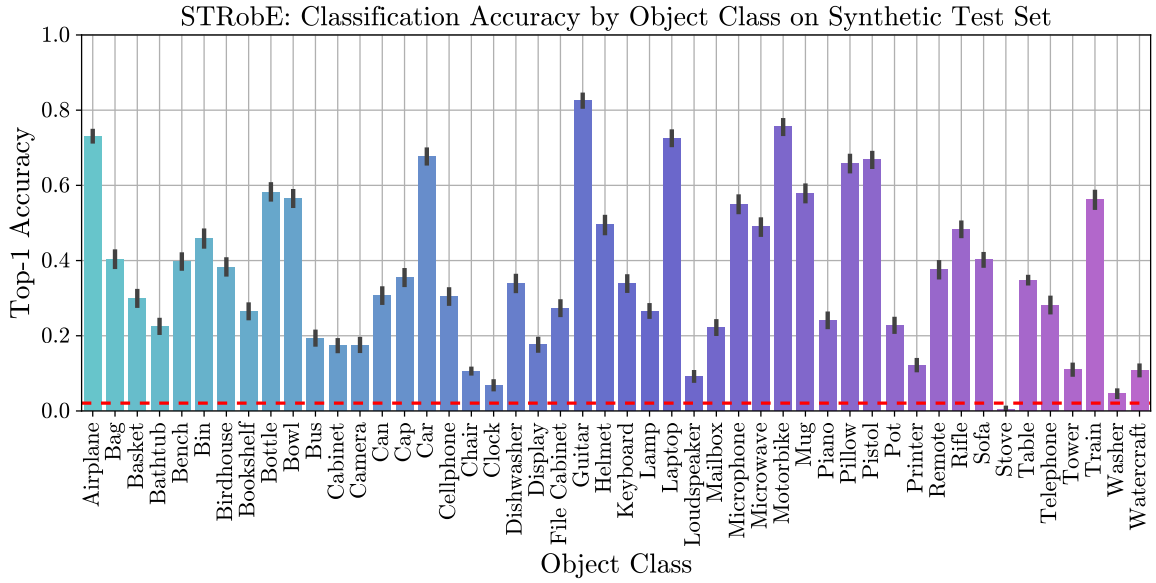
Lastly, Figure 5.6 shows the accuracy of the STRobE model at predicting lighting directions. By comparing this with the locations of the lighting configurations shown in Figure 5.5, it is clear that the model has trouble distinguishing between lighting directions that are nearby in 3D space. At this early stage of training, the model also appears to rarely predict specific configurations like 7 and 8. Nonetheless, the model achieves a non-trivial top-1 accuracy of 7.06% (95% CI: (6.92, 7.20)) when predicting lighting directions.

5.2.2.2 Parameter Invariance

The STRobE model demonstrates a significantly improved level of invariance to each of the explanation parameters when compared with the *Reduced Swin Transformer V2* model. In this section, the performance of both models is evaluated over each of the explanation parameters using the same evaluation methods applied to existing models in Chapter 4.

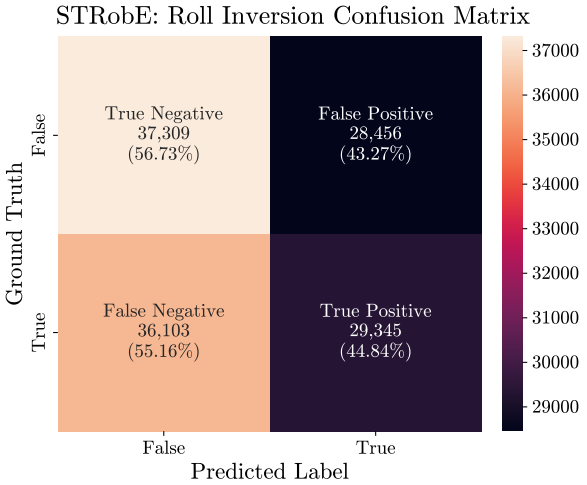


(a) Per-class accuracy for the *Reduced Swin Transformer V2* model.

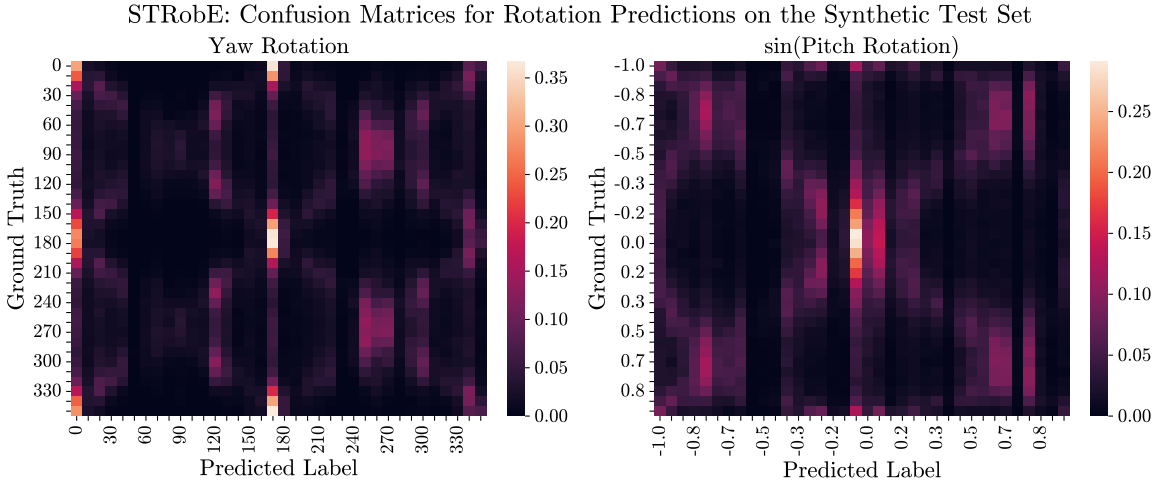


(b) Per-class accuracy for the STRobE model.

Figure 5.3: Comparison of top-1 accuracy over the 48 object classes for the STRobE and *Reduced Swin Transformer V2* models. The red dashed line represents the expected performance of a random classifier.

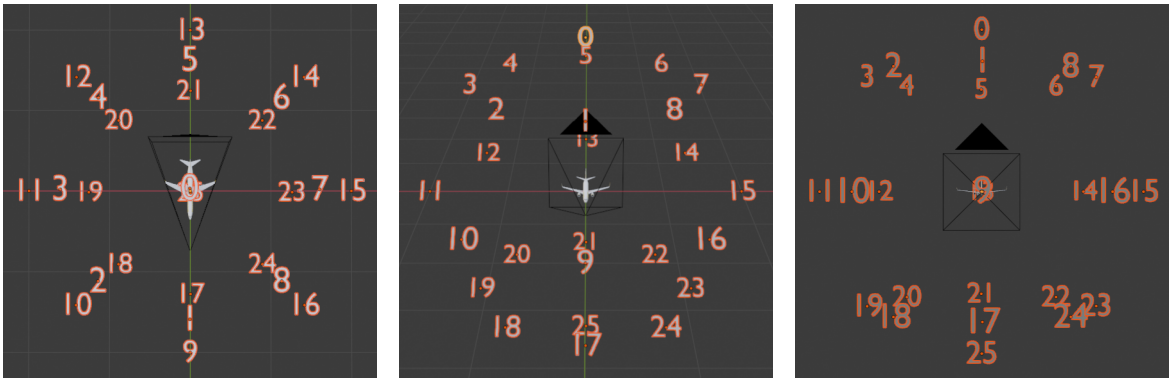


(a) Confusion matrix for the binary roll rotation classification task.



(b) Confusion matrices for the pitch and yaw rotation classification tasks.

Figure 5.4: Confusion matrices showing the distribution of rotation predictions by the STRobE model on the synthetic image test set.



(a) Top view (b) 45° elevation front view (c) Front view

Figure 5.5: The 26 lighting configurations used in the synthetic data sets.

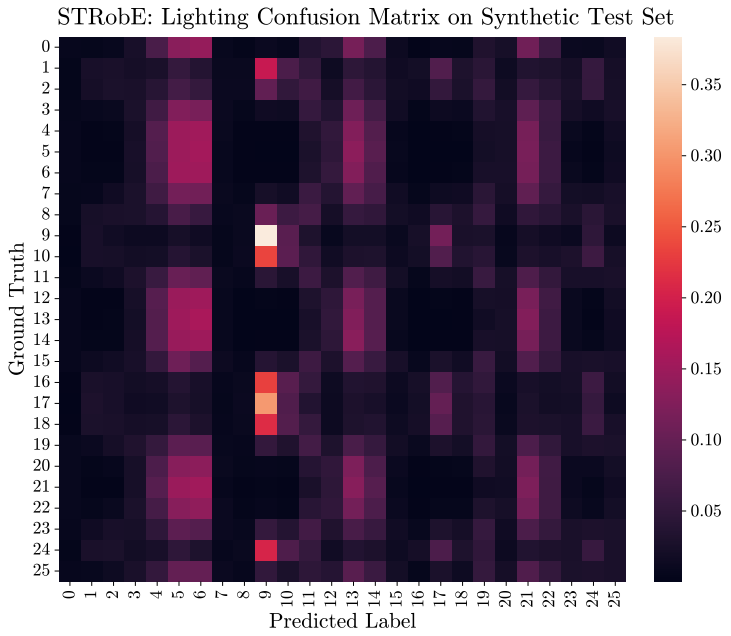


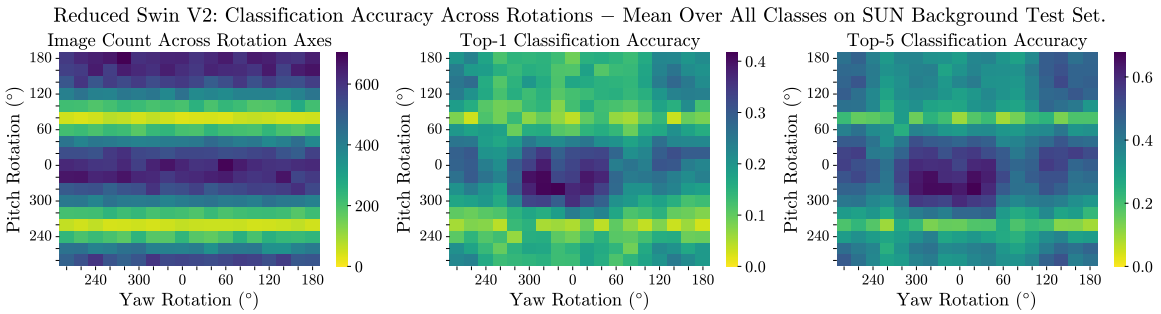
Figure 5.6: Confusion matrix showing the distribution of lighting predictions by the STRobE model on the synthetic image test set. Tick labels correspond to the lighting positions shown in Figure 3.3.

Pose Invariance: The STRobE model is considerably better at classifying objects in areas of rotation-space where the *Reduced Swin V2* model performs poorly. Figure 5.7 shows that this is true over both axes of rotation. While regions of maximum classification accuracy for the *Reduced Swin V2* model are densely concentrated around the central 0° rotations of the objects, the STRobE model demonstrates similar accuracy throughout rotation space. Objects with a pitch of ±90° are an exception to this, showing bands of lower accuracy.

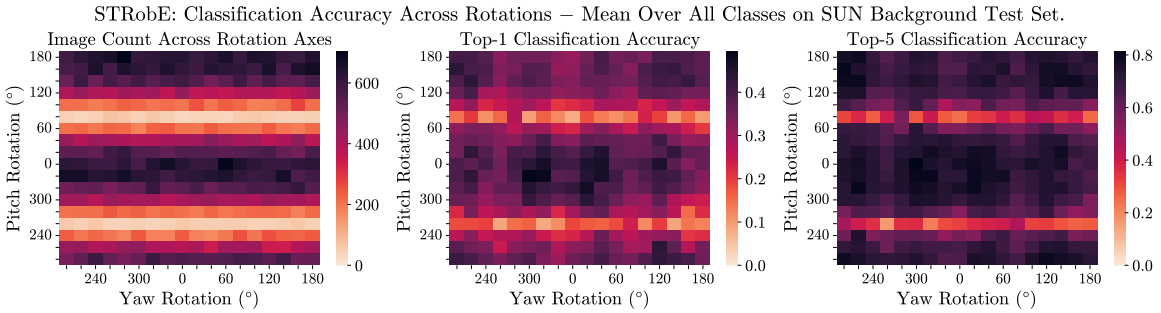
These ±90° pitch rotations are represented relatively less than other pitch rotations in the synthetic training data due to the use of *Spherically-Distributed Pose Sampling*. The lower performance on these rotations suggests that *Uniform Axial Sampling* may result in even greater pose invariance.

This is further supported by Figure 5.8b, which shows that classification performance across the pitch axes increases significantly for regions of pitch rotation space that are represented heavily in training data, and that the effect for rotations around ±90° is comparatively small. Overall, we conclude that greater representation of challenging poses in training data results in models with superior rotation invariance.

Background Invariance: Classification performance of the STRobE model varies significantly less between Indoor, Outdoor (Man-Made), and Outdoor (Natural) backgrounds than it does for the *Reduced Swin V2* model. This is shown in Figure 5.9, which demonstrates that while the *Reduced Swin V2* model responds significantly better on Outdoor (Natural) and Outdoor (Man-Made) backgrounds, the STRobE model does *not* respond differently to these background classes in a statistically significant way.



(a) Accuracy over rotation space for the *Reduced Swin Transformer V2* model.



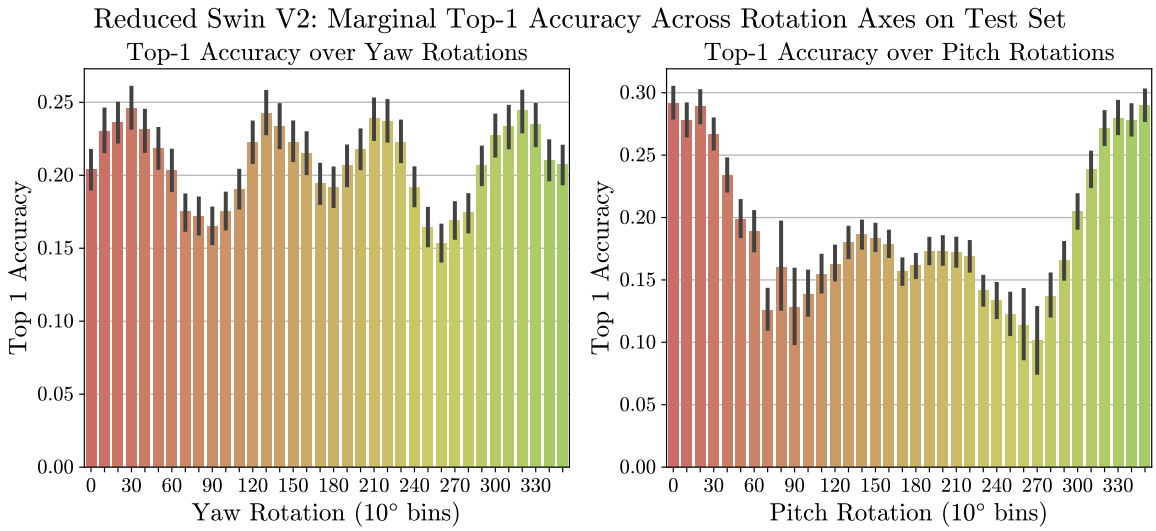
(b) Accuracy over rotation space for the STRobE model.

Figure 5.7: Comparison of classification accuracy over rotation space for the STRobE and *Reduced Swin Transformer V2* models, demonstrating the superior rotation invariance of the STRobE model.

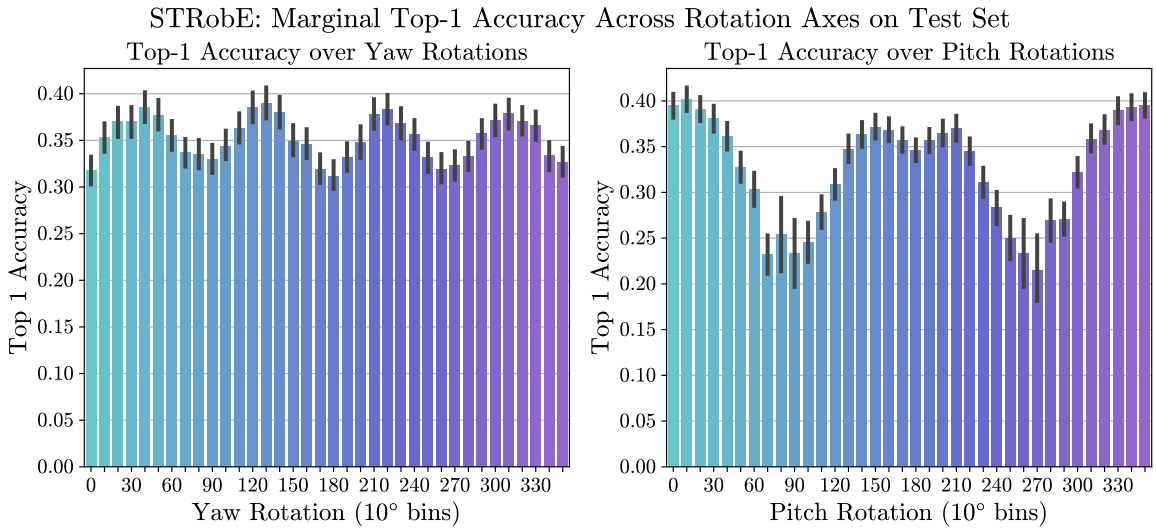
This result supports existing conclusions by Tobin et al. (2017), showing that background randomisation encourages models to attend to the features of the image subject, rather than relying on properties of the image background. It is noteworthy that this training process appears to completely eliminate the effect of confounding objects appearing in image backgrounds, which was proposed as the most significant reason for type-II errors in Section 4.2.4.

Invariance to Lighting Direction: Similar to both pose and background, invariance to lighting direction also appears to be improved in a significant way by training on synthetic data with diverse lighting configurations. Figure 5.10 compares the accuracy of the STRobE and *Reduced Swin V2* models over the 26 lighting configurations shown in Figure 5.5. While the *Reduced Swin V2* model is not affected by lighting to an extreme extent, the STRobE model still demonstrates a considerably flatter distribution, especially with its top-5 accuracy.

Particularly noteworthy is the performance across groups of lighting positions in the image's depth plane (shown in Figure 5.11). Along this axis, the *Reduced Swin V2* model performs significantly worse with lights that are positioned *behind* the image subject. It was suggested in Section 4.2.5 that this is due to specific lighting configurations that obscure identifying features of objects. While this is supported by these results (since these configurations are still the most challenging for the STRobE model), training on synthetic images does appear to have reduced this difference in a significant way.

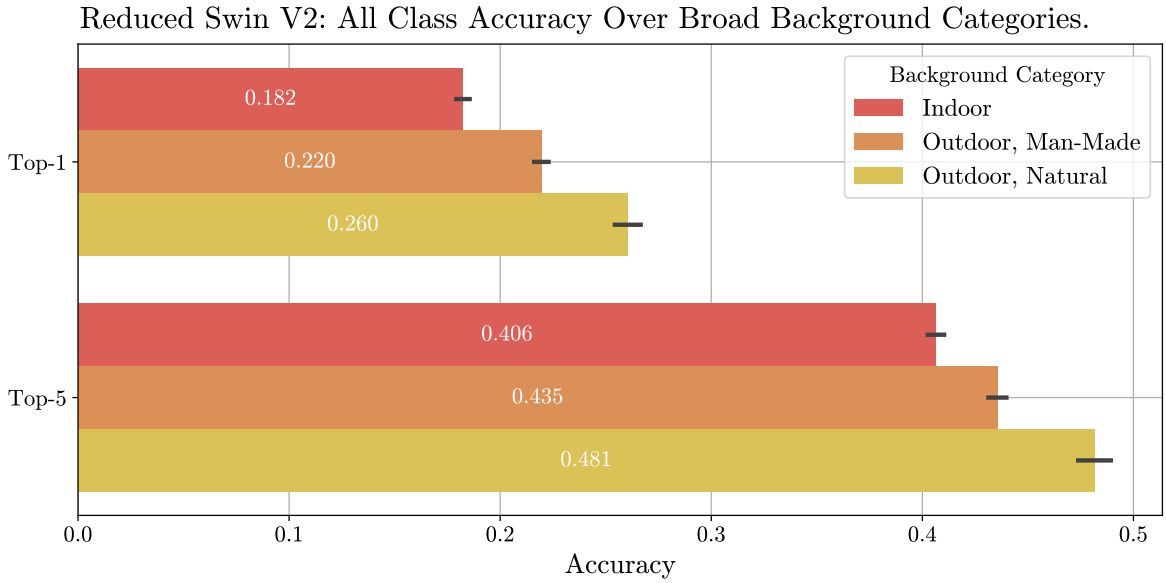


(a) Marginal accuracy over rotation axes for the *Reduced Swin Transformer V2* model.

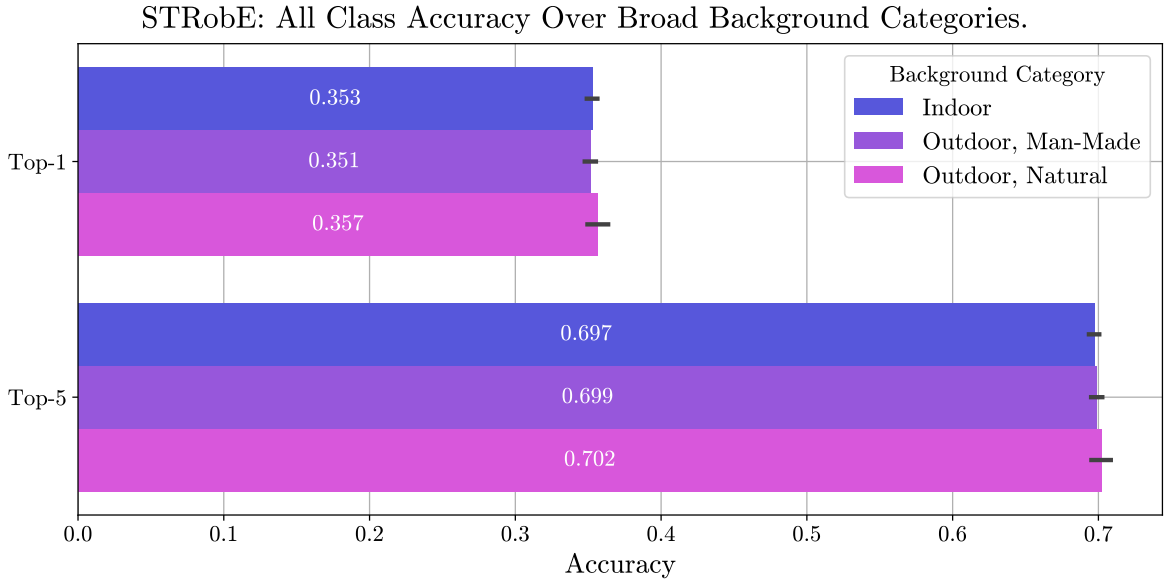


(b) Marginal accuracy over rotation axes for the STRobE model.

Figure 5.8: Comparison of classification accuracy over individual rotation axes for the STRobE and *Reduced Swin Transformer V2* models.

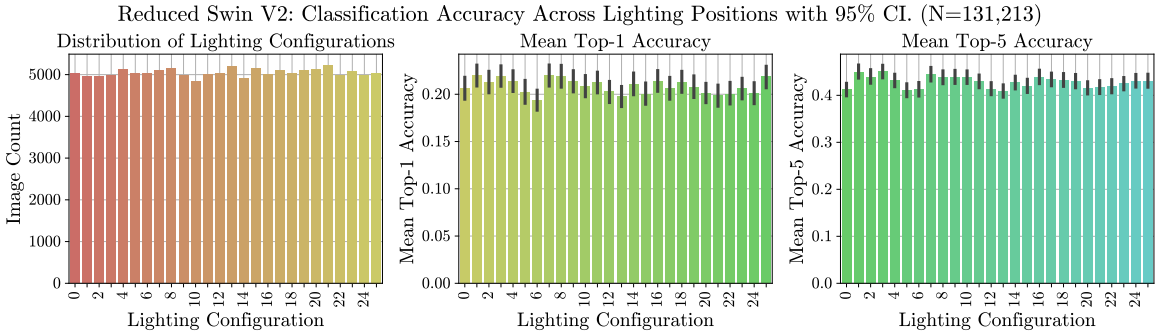


(a) Accuracy over broad background categories for the *Reduced Swin Transformer V2* model.

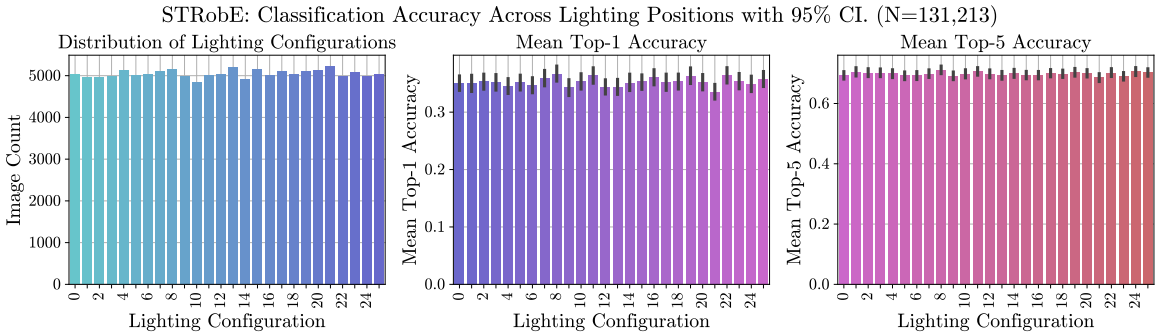


(b) Accuracy over broad background categories for the STRobE model.

Figure 5.9: Comparison of classification accuracy over Indoor, Outdoor (Man-Made), and Outdoor (Natural) backgrounds for the STRobE and *Reduced Swin Transformer V2* models.

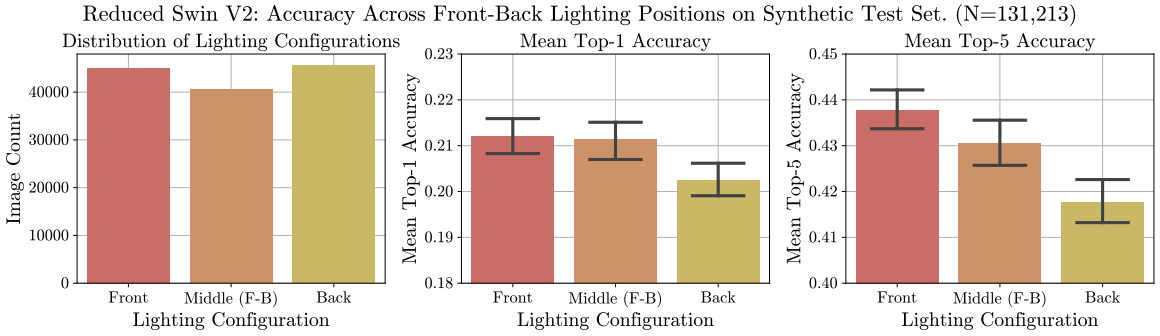


(a) Accuracy over individual lighting configurations for the *Reduced Swin Transformer V2* model.

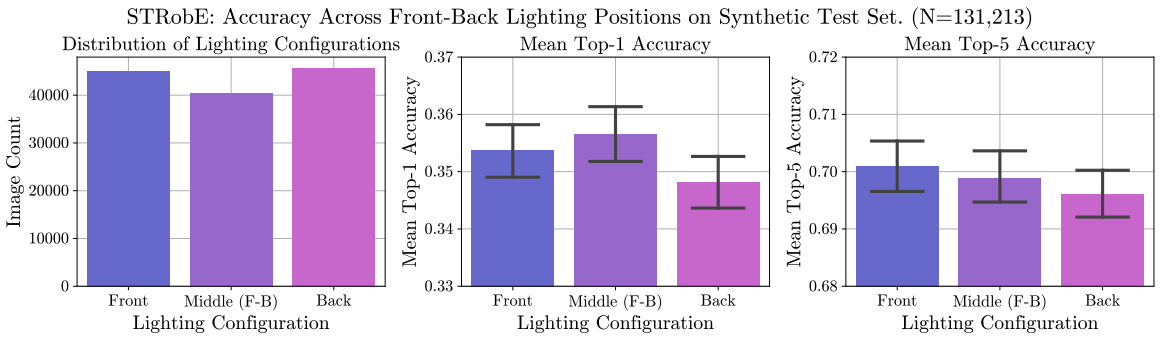


(b) Accuracy over individual lighting configurations for the STRobE model.

Figure 5.10: Comparison of classification accuracy over the 26 lighting configurations in the synthetic test data set for the STRobE and *Reduced Swin Transformer V2* models.



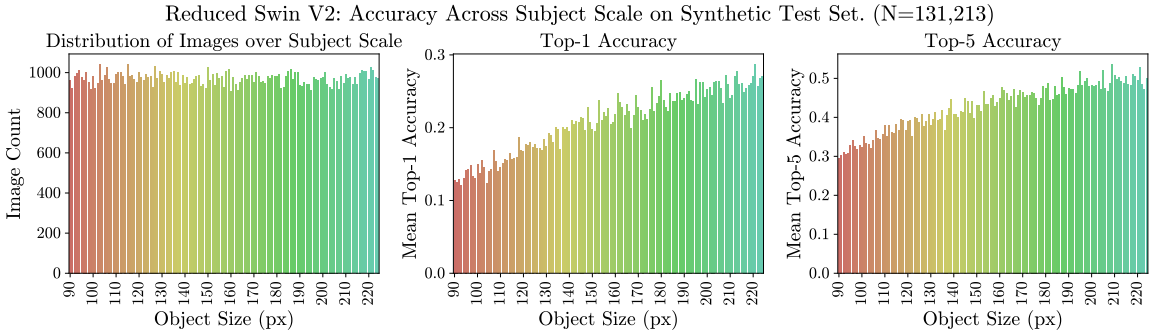
(a) Accuracy over front-back lighting groups for the *Reduced Swin Transformer V2* model.



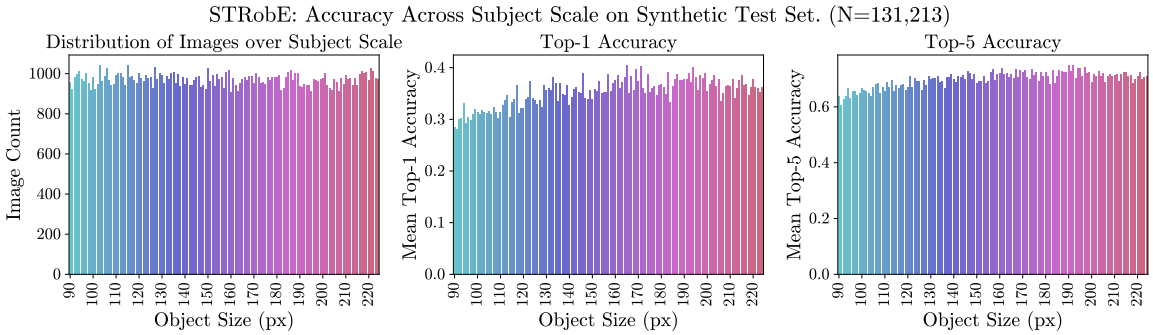
(b) Accuracy over front-back lighting groups for the STRobE model.

Figure 5.11: Comparison of classification accuracy over front-to-back lighting groups for the STRobE and *Reduced Swin Transformer V2* models.

Scale Invariance: While scale was not one of the primary explanation parameters, scale was varied in the synthetic data set as an augmentation, and to facilitate the prediction of object scales in real images. A side effect of this is a considerably improved ability to accurately classify small-scale image subjects. This is demonstrated in Figure 5.12, which shows not only an overall increase in accuracy for the STRobE model, but a disproportionately large increase for small-scale image subjects. This again suggests that greater representation of diverse scales in training data results in greater invariance to scale in resulting models.



(a) Accuracy over scale for the *Reduced Swin Transformer V2* model.



(b) Accuracy over scale for the STRobE model.

Figure 5.12: Comparison of classification accuracy over the scale of the image subject for the STRobE and *Reduced Swin Transformer V2* models.

5.2.3 Methodology for Evaluating Explanatory Outputs

The explanatory outputs are the second proposed contribution of the STRobE model. As described in Section 5.1.2, the model is designed to output predictions for the explanation parameters for each image it classifies. These explanatory outputs can then be fed into the image synthesis pipeline to produce a synthetic representation of how the model perceives the input image.

These image-based explanations are intended to provide users with a deeper understanding of the model outputs, especially in cases where the class label or other explanatory outputs are assigned incorrectly. To determine the value provided by this method, a qualitative evaluation of the image-based explanation technique is performed by synthesising explanatory images for instances from a real-image data set. The selected data set for this evaluation is ObjectNet

(a.i) ObjectNet *chair*

```
object class: table
background: bathroom
yaw rotation: 180°
pitch rotation: 0°
roll rotation: 180°
lighting conf.: 5
scale: 90px
X offset: 60px
Y offset: 50px
```

(a.ii) STRobE-generated labels

(a) The original *chair* image from ObjectNet, and it's corresponding STRobE output.

(b.i) Explanatory image 1

(b.ii) Explanatory image 2

(b.iii) Explanatory image 3

(b) Image-based explanations generated using the explanatory labels.

Figure 5.13: A sample *chair* image from the ObjectNet data set, accompanied with explanatory outputs from STRobE and image-based explanations generated using these labels.

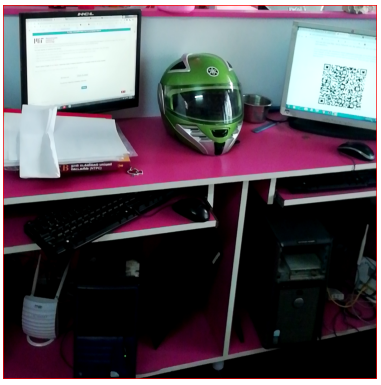
(Barbu et al., 2019), which is designed specifically to test computer vision models with unique viewpoints and backgrounds.

The evaluation is performed by passing real images from ObjectNet into the STRobE model, and feeding the explanatory outputs produced by the model into the image synthesis pipeline, producing explanatory visualisations. Samples from this evaluation are presented in Figures 5.13 and 5.14, and these samples are discussed further in the following section.

5.2.4 Explainability Evaluation Results

By passing ObjectNet images from various classes into the STRobE model and using the resulting outputs to produce large quantities of image-based explanations, various conclusions were drawn about the value of the proposed image-based explanation technique. This section presents those conclusions, making reference to Figures 5.13 and 5.14, which show two of the aforementioned ObjectNet images and their corresponding synthetic image explanations.

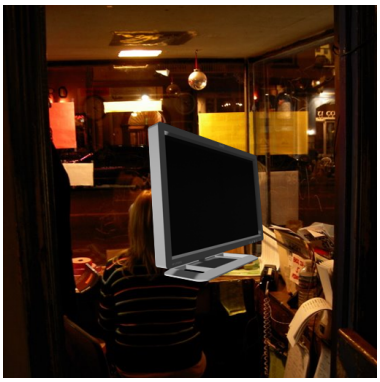
The left image in Figures 5.13a and 5.14a are the instances sampled from the ObjectNet data



```
object class: display
background: ticket booth
yaw rotation: 310°
pitch rotation: 0°
roll rotation: 180°
lighting conf.: 18
scale: 110px
X offset: 80px
Y offset: 80px
```

(a.i) ObjectNet *helmet* (a.ii) STRobE-generated labels

(a) The original *helmet* image from ObjectNet, and it's corresponding STRobE output.



(b.i) Explanatory image 1 (b.ii) Explanatory image 2 (b.iii) Explanatory image 3

(b) Image-based explanations generated using the explanatory labels.

Figure 5.14: A sample *helmet* image from the ObjectNet data set, accompanied with explanatory outputs from STRobE and image-based explanations generated using these labels.

set. The right panels show the labels that were assigned to these images by the STRobE model. The second row of each figure (Figures 5.13b and 5.14b) shows three explanatory images that were generated by the image synthesis pipeline using these STRobE outputs as input. The following conclusions are drawn based on the large set of synthetic images that were generated.

5.2.4.1 Synthetic image-based explanations can explain type-II errors

Evident especially in Figure 5.13 is that this style of image-based explanation provides value by explaining certain instances of type-II error. In this example, the upside-down image of the *chair* is misclassified as a *table* by the STRobE model. While this result may initially be unintuitive, the image-based explanations provided in Figures 5.13b.i and 5.13b.ii display similar characteristics to the ObjectNet image, suggesting the features of the chair that the model may be misinterpreting as *table* features.

These explanations are valuable in cases where variation in the explanation parameters causes an object to appear like another object. Errors caused by this are often unintuitive to identify

and explain, and this form of image-based explanation provides genuine value by highlighting these cases. Figure 5.1 shows another example, demonstrating how a *dishwasher* may appear similar to a *stove* when subject to specific rotations and scene configurations.

While this form of image-based explanation does help to explain this type of error where objects are confused for other classes, it provides less value when the false negative result is caused by confounding objects in the input image. Figure 5.14 exemplifies this, as the STRobE model misclassifies this instance due to the presence of a *display* in the image.

In this case, the image-based explanations provided in Figure 5.14b *do* highlight the presence of the confounding object in the image, but provide little value towards explaining why the desired image subject was overlooked. In cases like these, saliency mapping techniques (discussed in Section 2.3.2.1) are expected to provide greater explanatory value as they are specifically designed to identify the salient regions of input images.

5.2.4.2 Explanations do not always intuitively correspond to input images

The diverse images and 3D models present in the SUN and ShapeNetCore data sets allow for explanatory images to contain high variation in appearance. While this is valuable in that it facilitates the production of very specific synthetic images, the granularity of the labels on these inputs is often too coarse to produce explanatory images that correspond closely to the input image. Similarly, the specificity of labels output by the STRobE model may be too coarse to precisely define the desired format of explanatory images.

This is highlighted in Figures 5.13b.iii and 5.14b.iii which display relatively little similarity to their corresponding ObjectNet images. In Figure 5.13b.iii, the specific *table* object and background image selected happen to correspond poorly to the input image. While the other explanatory images in Figure 5.13b show relatively high correspondence, the high diversity in the input data sets means that this is often not the case.

Figure 5.14 demonstrates a different issue, whereby valuable explanatory images require accurate predictions from the STRobE model. In this example, the background, yaw rotation, and offset of the *display* identified in the input image are somewhat incorrect, again resulting in explanatory images with relatively low correspondence. It is suggested that this would improve if the STRobE model was trained to convergence.

Overall, these results demonstrate the tension between prediction complexity and the generation of highly specific synthetic images. On one hand, having relatively coarse-grained classes of image subjects and backgrounds increases the chance of these outputs being learned effectively by STRobE or future explainable models. On the other hand, these coarse labels result in explanatory images that do not always correspond with the source image, limiting their explanatory value. Some suggestions to improve this image-based explanation technique are therefore proposed in Section 6.4.1.

Chapter 6

Limitations and Future Directions

In the course of this research, several limitations were encountered across different stages of the project. This section provides a systematic overview of these obstacles, clarifying the scope and boundaries of the findings, and providing directions and suggestions for future research.

6.1 Image Synthesis

The image synthesis process, while serving its intended purpose, has areas for potential improvement. Based on the results found when evaluating existing models in Section 4.2, a primary consideration for producing a synthetic data set of higher quality in future research is further reduction of the domain gap. The following short sections present individual limitations and suggestions for future work.

6.1.1 Photorealism should be further emphasised in future research

As mentioned in Section 2.4.1, photorealistic renders contain a high level of realistic detail, making them challenging to distinguish from real images and therefore ideal for evaluation and training of computer vision models. While varying the explanation parameters in this research naturally conflicts with the objective of photorealism (since this variation involves the intentional use of out-of-distribution poses, backgrounds, and lighting configurations) the synthetic images present some additional unrealistic properties that could be addressed in future research.

Some key limitations of this research that could be addressed in future work are:

A lack of interaction between image subjects and backgrounds: This arises due to the nature of the synthesis process, involving separate rendering and compositing phases. This split approach to adding backgrounds provides considerable advantages in terms of speed and background diversity (as subjects can be inexpensively composited onto multiple backgrounds), but is unable to simulate interaction between image

subjects and scenes. One such interaction lacking in the synthetic images is *gravity*. In most real images, the image subject interacts with the ground and/or other supporting surfaces, which is not possible with the compositing methodology.

Instead, future data sets may opt for virtual environments or worlds in which simulated image subjects could be positioned and oriented, subject to simulated gravity. These simulated environments may also encourage other photorealistic interactions, such as (a) lighting conditions that are consistent between the image subject and the virtual environment (b) occlusion by other objects in the scene, and (c) objects that are not always framed entirely within the image bounds (though this could also be achieved with the compositing method). While a synthesis pipeline based on simulated environments would result in considerably increased rendering time and complexity, it may prove worthwhile to achieve a reduction in the domain gap across these many important dimensions.

Limited realism of ShapeNetCore models: While the ShapeNetCore data set provides multiple advantages, especially when comparing its size and annotation quality to other data sets, the 3D models it contains are not designed to maximise photorealism. Instead, the objects contained in the data set are usually CAD models that often have geometries and textures that are not faithful to real world objects. As such, alternative data sets of 3D models should be considered for future research, and may improve the quality of resulting synthetic data sets in ways other than simply increasing photorealism. These other benefits are discussed in the following section.

6.1.2 Alternative input data sets should be considered

3D Object Data Sets: As mentioned above, the ShapeNetCore data set is not perfectly tailored for this research due to a lack of photorealism, however, there are other limitations in addition to this. Firstly, the classes present in ShapeNetCore are considerably imbalanced, as shown in Figure 6.1. While this imbalance is managed in this project by sampling additional poses of classes with fewer models, this results in certain 3D objects being contained in more images than others. On the extremes, this results in only 10 images (the defined minimum) per object for each class containing more than 1,500 models (e.g. *table*, *chair*, *airplane*) and 278 images per class for the least-numerous class, *cap*.

Remaining on the topic of the 3D object data set, there are additional distributional factors that limit the suitability of ShapeNetCore. Another such limitation is the limited overlap between ShapeNetCore and ImageNet. Recall that the entire ShapeNetCore data set contains 55 classes, of which only 48 were identified as overlapping with ImageNet classes (see Appendix A.1). The one-to-many relationship of this mapping means that these models span 88 of the 1000 classes present in ImageNet. Clearly the evaluation performed for Objective 2 (Existing Model Evaluation) would be strengthened by considering a wider range of classes. It may be particularly relevant to consider plant and animal classes which are prevalent in ImageNet and do not appear in ShapeNetCore.

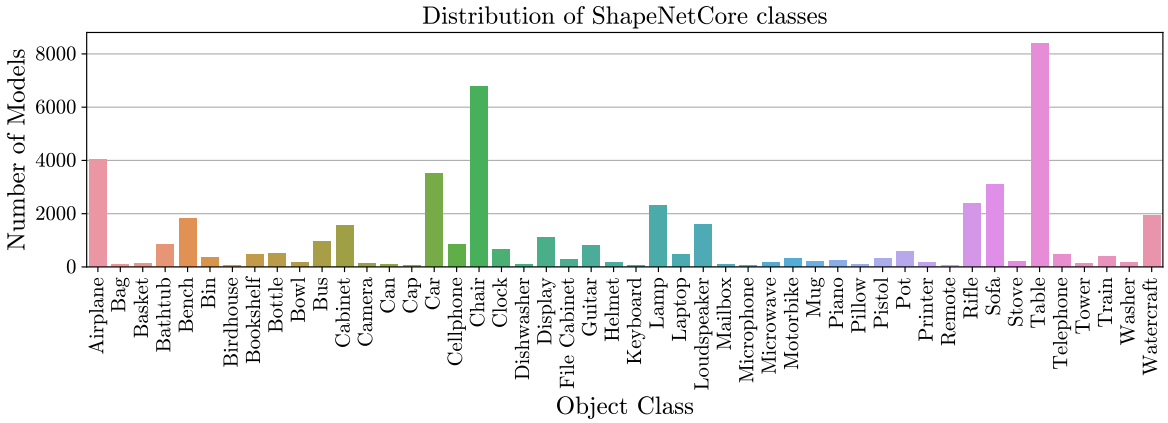


Figure 6.1: The distribution of ShapeNetCore models across the 48 classes used for image synthesis.

Background Data Sets: If future research is to use composited backgrounds, there is also room to improve the choice of background data set. In this project, the primary limitation of using SUN397 is an over-abundance of diversity and complexity. The 397 classes are simply too numerous to draw meaningful or statistically significant conclusions about the interactions between individual background and object classes. This is because the sample size of each background-object pair is simply too small, even in a synthetic data set of over 880,000 instances.

Additionally, many of the SUN images are already relatively complex scenes, that often already contain a clear image subject. This is especially true of Indoor backgrounds, and of Outdoor (Man-made) environments to a lesser extent. This causes problems when evaluating classification models as these models tend to identify and respond to features present in the backgrounds, rather than acknowledging the composited image subject. It is suggested that future work addresses this by selecting a background data set that contains fewer classes, as well as backgrounds that are specifically selected to reduce instances of confounding objects. This may be done by taking a subset of SUN397, or by finding a new data set entirely.

6.1.3 Synthesis with generative models should be reconsidered

Despite concluding in Section 2.4.2.3 that generative image models (e.g. VAEs and GANs) were not yet suitable for producing accurately labelled synthetic data sets, the pace of research in this area suggests that this should be reconsidered in future work. The current limitation of these models is *image-text alignment* (the degree to which the output image adheres to the provided input prompt), as existing models do not guarantee prompt adherence to a sufficient level.

Researchers including X. Wu, Sun, Zhu, Zhao, and Li (2023) demonstrate that aligning text-to-image models with *human preferences* is an area of active research, however, it is currently unclear how models such as *Stable Diffusion* may be modified to achieve sufficient image-text alignment for producing strictly labelled synthetic data. In addition to this, the fact

that training data for these models does not contain significant representation of challenging conditions raises uncertainty about the ability of generative models to accurately synthesise images of objects in said conditions. Rendered synthetic images remain a promising avenue for producing this training data.

6.2 Evaluation of Existing Models

The evaluation of existing models performed in service of Objective 2 (Existing Model Evaluation) supports existing research into the parameter invariance of computer vision models, and provides novel conclusions about the influence of the explanation parameters. With this being said, the results produced also highlight limitations of the evaluation methodology. The following sections therefore provide suggestions for how further insights can be gained in future research.

6.2.1 Evaluation parameters could be varied over larger domains

In the previous section, general limitations of the synthetic data set were identified, focusing on quality and diversity. However, it is also worth considering how effectively this data set facilitates the evaluation of existing models. Reflecting on this suggests that more information could have been gained by varying the explanation parameters over larger domains.

For example, using the image synthesis methodology described in Section 3.1, the 3D models from ShapeNetCore are randomly rotated along their local yaw and pitch axes for each image. They are not rotated along their local *roll* axis. While this was done to make the resulting data set more suitable for fine-tuning a classification model, it clearly limits the ability to evaluate the effect of roll rotations on classification performance.

Similar observations can be made about the other parameters, for example:

- Lighting directions could be represented using continuous angles rather than discrete positions to facilitate more fine-grained analysis.
- The scale of the image subject could be varied further beyond the 90–224 pixel range used in this research, including both image subjects smaller than 90×90 pixels, and subjects that overflowed the 224×224 pixel image.
- Offsets could similarly be adjusted to allow for image subjects that were not contained entirely within the bounds of the image.

6.2.2 Future research should evaluate other image parameters

In addition to evaluating the existing explanation parameters over larger domains, it is also worth considering extending this research to additional image parameters. This would result in a more comprehensive understanding of how computer vision models respond to changing parameters, and may highlight additional interactions between parameters.

Suggestions for future research include:

- Camera configurations could be varied, as modern rendering software allows for the simulation of parameters such as the F-stop. Analysing combinations including camera parameters would facilitate evaluation of invariance to *depth of field* and other related image parameters.
- As mentioned in Section 6.1.1, occlusion by other objects could be incorporated and evaluated, especially if rendering is done using virtual environments.
- Other lighting parameters could be considered, including lighting intensity, colour, and the number of lights, which were all constant in this research.

6.2.3 Consider interaction between specific objects and background classes

As mentioned in Section 6.1.2, the number of classes present in the SUN397 data set is simply too large to extract meaningful results about the interaction between individual objects and background classes. This limits the effectiveness of the evaluation performed on existing models as it results in an insufficient sample size when considering interaction between specific objects and background classes.

The related recommendation is that future research prioritises evaluation of this interaction. This may involve a comparison of classification results for specific objects appearing on canonical backgrounds (those against which they frequently occur in real images) versus backgrounds they are rarely observed on. Existing research by K. Xiao et al. (2020) suggests that classification performance may improve significantly when objects appear against canonical backgrounds, but this could not be validated in a significant way in this research.

6.2.4 Evaluation should be performed for more classification models

A final way that future research could produce a more powerful evaluation of existing computer vision models would be to evaluate a larger sample of models than the four used in this research. While the four selected models represent a relatively diverse sample of the current image classification landscape, evaluating more models across the spectrum of CNN and transformer-based architectures would facilitate more robust conclusions.

In such a future evaluation, it may be valuable to not only consider the level of parameter invariance that results from different architectures, but to also consider differences in the training processes of pre-trained models. While this research establishes an initial method for performing such an investigation, the scope of the investigation could not be expanded within the time constraints of the project.

6.3 Parameter-Invariant Image Classification

When training the STRobE model, various difficulties were encountered. While the issue of catastrophic forgetting was resolved with the use of *replay*, and the number of trainable

parameters was reduced using LoRA, the following sections present additional suggestions for how this process could be extended upon and improved in future research.

6.3.1 STRobE training should be continued and further improved

The primary limitation on the performance of the STRobE model was the inability to train the model to convergence within the time frame of this research. Due to a repeated need to modify and restart the training process, the final model presented in this research trained for only three epochs. It is possible that maximising the models accuracy could require an order of magnitude more epochs, which could be verified easily with more training time.

As well as simply training for longer, there is room for further experimentation with the hyperparameters and configuration of the STRobE model. Recall that in Section 5.2.2, certain tasks, including the prediction of roll rotations, were not learned effectively during the training process. While it is possible that this would change further into training, it is recommended that future research considers alternative ways of assigning weights in the multi-task loss function (Section 5.1.4.2) to encourage effective learning of all tasks. In the implementation of STRobE, weights are assigned according to the perceived importance of each task, however, future work may consider weighting these tasks according to task difficulty to ensure that gains on simpler tasks are not prioritised.

It is also recommended that future research includes more experimentation with design decisions such as the optimisers, learning rate, and LoRA hyperparameters (r and α), as there was insufficient time to iterate upon the selected parameter values in this research.

6.3.2 Evaluate future parameter-invariant models on ObjectNet

Another key limitation of this research is that the STRobE model was not evaluated on real images. Initially, an evaluation using the ObjectNet test set (Barbu et al., 2019) was proposed, however, as a result of limited training success until late in the project, this evaluation could not be performed.

ObjectNet is a real-image test set containing images of objects in intentionally challenging poses and contexts. Both pose and background are labelled in this data set, allowing for evaluation of the corresponding explanatory output heads of the STRobE model. Performing this evaluation would contribute significantly to the field of image classification, as it would highlight whether training on challenging poses in synthetic data translates to improved parameter-invariance in real images.

6.3.3 There is tension between learning difficulty and explanation quality

Also worth discussing for future research is the importance of considering the tension between explanation quality and learning difficulty. While decreasing granularity and specificity of the explanatory outputs makes the tasks easier to learn, it results in lower quality explanatory outputs and consequently worse image-based explanations. The results achieved after only

three epochs of training the STRobE model suggest that there is sufficient ability to learn each of the tasks labelled in this research.

As such, it may be valuable to use more specific labels for backgrounds and object classes in future research, specifying elements like colour and defining features, since simply providing the class label results in extremely diverse explanatory images. This is discussed further as a suggestion for improving image-based explanations in the following section.

6.4 Synthetic Image-Based Explanations

Based on evaluation of the image-based explanations produced using the image synthesis pipeline, the following recommendations are made for future implementations of similar explanation techniques. Note that the suggestions provided in Sections 6.1.1 and 6.1.2 for improving photorealism and data set selection respectively, would also result in improvements to the image-based explanation technique, but will not be repeated in this section.

6.4.1 Image-based explanations should be more specific

One of the key limitations of the image-based explanation approach is that explanatory images rarely correspond closely to the appearance of the input image. This is true even when the object class and explanation parameters are labelled correctly. As discussed in Section 5.2.4.2, this is because there is significant variation *within* object and background classes, meaning that even when these properties are predicted correctly, this does not guarantee that a suitable instance will be selected by the image synthesis pipeline.

There is potential to resolve this by labelling more features of the objects and backgrounds (e.g. colours, textures, as is done with concept-based explanations), allowing for the selection of more specific instances when generating explanations. If there is no requirement that the model produces explanatory outputs that are directly interpretable, then this could be further extended by substituting conceptual features for latent representations of the objects and backgrounds. With this extension, the image-based explanation technique develops further similarities to the example-based method proposed by C. Chen et al. (2019), which uses similar latent representations from input images to match image features with those present in prototype images. In this case, those prototype images (and 3D models) would then be used to synthesise image-based explanations.

6.4.2 Image-based explanations should be evaluated with a user study

Another limitation of this research is that the image-based explanation technique was not evaluated quantitatively with a user study. A user study utilising the ObjectNet data set (Barbu et al., 2019) *was* proposed for this project, but was not implemented due to time constraints and the limited success of the model training process. As such, the proposed methodology for a simple controlled study is presented below as a suggestion for future evaluation of similar explanation techniques:

1. The ObjectNet data set is retrieved, and classes from the ObjectNet data set that are not predicted by the model are discarded.
2. A representative sample $I_O = \{i_1^O, \ldots, i_n^O\}$ of the remaining images from ObjectNet are passed through the model, generating predicted class labels $C_O = \{c_1^O, \ldots, c_n^O\}$ explanatory outputs $E_O = \{e_1^O, \ldots, e_n^O\}$.
3. These textual explanations (E_O) are used as input to the image synthesis pipeline generating explanatory images $I_E = \{i_1^E, \ldots, i_n^E\}$.
4. The group of users is divided into two groups, A (control) and B (treatment).
5. Group A is presented with input images i_j^O and their predicted class label c_j^O produced by the model (not the complete explanatory outputs e_j^O, or the explanatory images i_j^O). They are asked to evaluate:
   - Their understanding of the model's output without access to the explanation.
6. Group B is presented with input images i_j^O, as well as the corresponding explanatory images i_j^E and are asked to evaluate:
   - Their understanding of the model's output *with* access to the explanation image.
   - The similarity between i_j^O and the explanatory image i_j^E, measured across various dimensions including composition and photorealism.

By evaluating the extent to which the treatment group perceives a greater understanding of the model's classification output, the value provided by the image-based explanations can be quantified, which was not possible in this research.

Chapter 7

Conclusion

Considering the wide-ranging and important scenarios in which computer vision algorithms are deployed in the real world, this research is motivated by the need to improve the robustness and explainability of these models. In this project, we first set out to determine how robust existing image classification models are to variation in the explanation parameters. Based on the work performed for Objective 2 (Existing Model Evaluation), it is clear that even for state-of-the-art models, classification performance changes significantly as a result of even minor variations in pose, background, and lighting direction.

In this evaluation, we find that classification models transfer successfully to synthetic images, and using this fact, draw various conclusions about invariance to each of the explanation parameters. When evaluating across pose variations, we find that accurate classifications are highly localised in rotation space, and that these regions of high accuracy are highly correlated with the object class. This suggests that identifying features of objects are more responsible for accurate classification than representation of specific poses in training data. Evaluation across image backgrounds suggests that backgrounds containing confounding objects are a primary cause of type-II error, and that outdoor backgrounds (especially natural ones) maximise classification performance due to a relative absence of these confounding objects.

When assessing the impact of lighting direction, we find that lights placed directly in front of, or behind objects result in the worst classification accuracy. We suggest that this is due to a lack of illumination on key features when the light is behind the object, and a lack of cast shadows when the light is directly in front. Finally, we evaluate the ways that classification models respond to the scale of image subjects, finding that classification accuracy varies approximately linearly with object scale over the range of sizes included in the synthetic data set.

Following this conclusion that models are *not* invariant to changes in the explanation parameters, we set out to produce a parameter-invariant and explainable model (STRobE) in Objective 3 (Model Training). By training the STRobE model on the synthetic data set, we conclude that representing diverse image configurations in training data results in sig-

nificantly improved robustness to the explanation parameters. Despite limitations on the training process, we show that this applies across pose, background, lighting direction, and object scale.

We additionally demonstrate that synthetic images serve not only as a way to train more parameter-invariant models, but that image synthesis can be used to produce explanations for image classification models. These explanations provide value in highlighting causes of type-II error, but do not consistently correspond to input images due to the coarse granularity of class and background labels.

While various ways to build upon and improve these results are presented in Chapter 6, this research has significant ramifications for the training and deployment of future image classification algorithms. Primarily, this research suggests that training models on data sets with more diverse representation of poses, backgrounds, lighting directions, and object scales has significant potential to improve their invariance to these parameters. It is recommended that future data sets specifically include these parameters in their design, and that synthetic data sets are considered for their potential to vary and label these parameters in a systematic way.

It's evident from this research that the journey towards truly robust and explainable computer vision models is multifaceted and ongoing. Based on our evaluation, we suggest that designers and users of computer vision models be more conscious of the ways that these algorithms respond to challenging inputs. By acknowledging and reflecting on these limitations, we hope that existing models can be deployed more safely, and that more robust and explainable models can be developed in the future.

References

Abnar, S., & Zuidema, W. (2020). Quantifying attention flow in transformers. *arXiv preprint arXiv:2005.00928*.

Alcorn, M. A., Li, Q., Gong, Z., Wang, C., Mai, L., Ku, W.-S., & Nguyen, A. (2019). Strike (with) a pose: Neural networks are easily fooled by strange poses of familiar objects. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (pp. 4845–4854).

Alvarez Melis, D., & Jaakkola, T. (2018). Towards robust interpretability with self-explaining neural networks. *Advances in neural information processing systems*, *31*.

Anderson, J. W., Ziolkowski, M., Kennedy, K., & Apon, A. W. (2022). Synthetic image data for deep learning. *arXiv preprint arXiv:2212.06232*.

Arbatli, A. D., & Akin, H. L. (1997). Rule extraction from trained neural networks using genetic algorithms. *Nonlinear Analysis: Theory, Methods & Applications*, *30*(3), 1639–1648.

Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.-R., & Samek, W. (2015). On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, *10*(7), e0130140.

Bakker, B., & Heskes, T. (2003). Task clustering and gating for bayesian multitask learning. *Journal of Machine Learning Research*, *4*.

Barbu, A., Mayo, D., Alverio, J., Luo, W., Wang, C., Gutfreund, D., ... Katz, B. (2019). Objectnet: A large-scale bias-controlled dataset for pushing the limits of object recognition models. *Advances in neural information processing systems*, *32*.

Bau, D., Zhou, B., Khosla, A., Oliva, A., & Torralba, A. (2017). Network dissection: Quantifying interpretability of deep visual representations. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 6541–6549).

Blender Online Community. (2018). Blender - a 3d modelling and rendering package [Computer software manual]. Stichting Blender Foundation, Amsterdam. Retrieved from http://www.blender.org

Calonder, M., Lepetit, V., Strecha, C., & Fua, P. (2010). Brief: Binary robust independent elementary features. In *Computer vision–eccv 2010: 11th european conference on computer vision, heraklion, crete, greece, september 5-11, 2010, proceedings, part iv 11* (pp. 778–792).

Castro, F. M., Marín-Jiménez, M. J., Guil, N., Schmid, C., & Alahari, K. (2018). End-to-

end incremental learning. In *Proceedings of the european conference on computer vision (eccv)* (pp. 233–248).

Chang, A. X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., ... Su, H. (2015). Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*.

Chang, C.-H., Creager, E., Goldenberg, A., & Duvenaud, D. (2018). Explaining image classifiers by counterfactual generation. *arXiv preprint arXiv:1807.08024*.

Chellapilla, K., Puri, S., & Simard, P. (2006). High performance convolutional neural networks for document processing. In *Tenth international workshop on frontiers in handwriting recognition.*

Chen, C., Li, O., Tao, D., Barnett, A., Rudin, C., & Su, J. K. (2019). This looks like that: deep learning for interpretable image recognition. *Advances in neural information processing systems*, *32*.

Chen, T., Kornblith, S., Swersky, K., Norouzi, M., & Hinton, G. E. (2020). Big self-supervised models are strong semi-supervised learners. *Advances in neural information processing systems*, *33*, 22243–22255.

Csurka, G., Dance, C., Fan, L., Willamowski, J., & Bray, C. (2004). Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, eccv* (Vol. 1, pp. 1–2).

Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. In *2005 ieee computer society conference on computer vision and pattern recognition (cvpr'05)* (Vol. 1, pp. 886–893).

Dawson, H. L., Dubrule, O., & John, C. M. (2023). Impact of dataset size and convolutional neural network architecture on transfer learning for carbonate rock classification. *Computers & Geosciences*, *171*, 105284.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 ieee conference on computer vision and pattern recognition* (pp. 248–255).

Dhariwal, P., & Nichol, A. (2021). Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, *34*, 8780–8794.

Dollar, P., Wojek, C., Schiele, B., & Perona, P. (2011). Pedestrian detection: An evaluation of the state of the art. *IEEE transactions on pattern analysis and machine intelligence*, *34*(4), 743–761.

Dong, Z., & Lin, B. (2020). Learning a robust cnn-based rotation insensitive model for ship detection in vhr remote sensing images. *International Journal of Remote Sensing*, *41*(9), 3614–3626.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... Gelly, S. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Downs, L., Francis, A., Koenig, N., Kinman, B., Hickman, R., Reymann, K., ... Vanhoucke, V. (2022). Google scanned objects: A high-quality dataset of 3d scanned household items. In *2022 international conference on robotics and automation (icra)* (pp. 2553–

2560).

Drenkow, N., Sani, N., Shpitser, I., & Unberath, M. (2022). A systematic review of robustness in deep learning for computer vision: Mind the gap? *arXiv preprint arXiv:2112.00639*.

Du, M., Liu, N., & Hu, X. (2019). Techniques for interpretable machine learning. *Communications of the ACM*, *63*(1), 68–77.

d'Ascoli, S., Touvron, H., Leavitt, M. L., Morcos, A. S., Biroli, G., & Sagun, L. (2021). Convit: Improving vision transformers with soft convolutional inductive biases. In *International conference on machine learning* (pp. 2286–2296).

Engstrom, L., Tran, B., Tsipras, D., Schmidt, L., & Madry, A. (2019). Exploring the landscape of spatial robustness. In *International conference on machine learning* (pp. 1802–1811).

Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., & Thrun, S. (2017). Dermatologist-level classification of skin cancer with deep neural networks. *nature*, *542*(7639), 115–118.

French, R. M. (1999). Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, *3*(4), 128–135.

Frolov, S., Hinz, T., Raue, F., Hees, J., & Dengel, A. (2021). Adversarial text-to-image synthesis: A review. *Neural Networks*, *144*, 187-209. doi: https://doi.org/10.1016/j.neunet.2021.07.019

Gardner, M.-A., Sunkavalli, K., Yumer, E., Shen, X., Gambaretto, E., Gagné, C., & Lalonde, J.-F. (2017). Learning to predict indoor illumination from a single image. *arXiv preprint arXiv:1704.00090*.

Gong, Z., Zhong, P., & Hu, W. (2019). Diversity in machine learning. *Ieee Access*, *7*, 64323–64350.

González, Á. (2010). Measurement of areas on a sphere using fibonacci and latitude–longitude lattices. *Mathematical Geosciences*, *42*, 49–64.

Goyal, Y., Wu, Z., Ernst, J., Batra, D., Parikh, D., & Lee, S. (2019). Counterfactual visual explanations. In *International conference on machine learning* (pp. 2376–2384).

Gulshan, V., Peng, L., Coram, M., Stumpe, M. C., Wu, D., Narayanaswamy, A., ... others (2016). Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. *jama*, *316*(22), 2402–2410.

Guo, G., Wang, H., Bell, D., Bi, Y., & Greer, K. (2003). Knn model-based approach in classification. In *On the move to meaningful internet systems 2003: Coopis, doa, and odbase: Otm confederated international conferences, coopis, doa, and odbase 2003, catania, sicily, italy, november 3-7, 2003. proceedings* (pp. 986–996).

Han, K., Xiao, A., Wu, E., Guo, J., Xu, C., & Wang, Y. (2021). Transformer in transformer. *Advances in Neural Information Processing Systems*, *34*, 15908–15919.

Hara, S., & Hayashi, K. (2016). Making tree ensembles interpretable. *arXiv preprint arXiv:1606.05390*.

Hayes, T. L., Kafle, K., Shrestha, R., Acharya, M., & Kanan, C. (2020). Remind your neural network to prevent catastrophic forgetting. In *European conference on computer vision*

(pp. 466–483).

Hayes, T. L., Krishnan, G. P., Bazhenov, M., Siegelmann, H. T., Sejnowski, T. J., & Kanan, C. (2021). Replay in deep learning: Current approaches and missing biological elements. *Neural computation*, *33*(11), 2908–2950.

He, H., & Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering*, *21*(9), 1263–1284.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 770–778).

Hendricks, L. A., Akata, Z., Rohrbach, M., Donahue, J., Schiele, B., & Darrell, T. (2016). Generating visual explanations. In *Computer vision–eccv 2016: 14th european conference, amsterdam, the netherlands, october 11–14, 2016, proceedings, part iv 14* (pp. 3–19).

Hendricks, L. A., Hu, R., Darrell, T., & Akata, Z. (2018). Grounding visual explanations. In *Proceedings of the european conference on computer vision (eccv)* (pp. 264–279).

Hinterstoisser, S., Lepetit, V., Ilic, S., Holzer, S., Bradski, G., Konolige, K., & Navab, N. (2013). Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes. In *Computer vision–accv 2012: 11th asian conference on computer vision, daejeon, korea, november 5-9, 2012, revised selected papers, part i 11* (pp. 548–562).

Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Horn, D., & Houben, S. (2020). Fully automated traffic sign substitution in real-world images for large-scale data augmentation. In *2020 ieee intelligent vehicles symposium (iv)* (pp. 465–471).

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., ... Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., ... Chen, W. (2021). Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 4700–4708).

Itti, L., Koch, C., & Niebur, E. (1998). A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on pattern analysis and machine intelligence*, *20*(11), 1254–1259.

Jain, S., & Wallace, B. C. (2019). Attention is not explanation. *arXiv preprint arXiv:1902.10186*.

Janowczyk, A., & Madabhushi, A. (2016). Deep learning for digital pathology image analysis: A comprehensive tutorial with selected use cases. *Journal of pathology informatics*, *7*(1), 29.

Jetley, S., Lord, N. A., Lee, N., & Torr, P. H. (2018). Learn to pay attention. *arXiv preprint arXiv:1804.02391*.

Johnson, M. K., & Farid, H. (2005). Exposing digital forgeries by detecting inconsistencies in lighting. In *Proceedings of the 7th workshop on multimedia and security* (pp. 1–10).

Johnson-Roberson, M., Barto, C., Mehta, R., Sridhar, S. N., Rosaen, K., & Vasudevan, R. (2016). Driving in the matrix: Can virtual worlds replace human-generated annotations for real world tasks? *arXiv preprint arXiv:1610.01983*.

Karras, T., Laine, S., & Aila, T. (2019). A style-based generator architecture for generative adversarial networks. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (pp. 4401–4410).

Kemker, R., McClure, M., Abitino, A., Hayes, T., & Kanan, C. (2018). Measuring catastrophic forgetting in neural networks. In *Proceedings of the aaai conference on artificial intelligence* (Vol. 32).

Kim, B., Wattenberg, M., Gilmer, J., Cai, C., Wexler, J., & Viegas, F. (2018). Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International conference on machine learning* (pp. 2668–2677).

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., ... others (2017). Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, *114*(13), 3521–3526.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, *25*.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2324.

Leutenegger, S., Chli, M., & Siegwart, R. Y. (2011). Brisk: Binary robust invariant scalable keypoints. In *2011 international conference on computer vision* (pp. 2548–2555).

Li, S., Liu, Z.-Q., & Chan, A. B. (2014). Heterogeneous multi-task learning for human pose estimation with deep convolutional neural network. In *Proceedings of the ieee conference on computer vision and pattern recognition workshops* (pp. 482–489).

Li, Y., Yuan, G., Wen, Y., Hu, J., Evangelidis, G., Tulyakov, S., ... Ren, J. (2022). Efficientformer: Vision transformers at mobilenet speed. *Advances in Neural Information Processing Systems*, *35*, 12934–12949.

Liu, W., Mei, T., Zhang, Y., Che, C., & Luo, J. (2015). Multi-task deep visual-semantic embedding for video thumbnail selection. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 3707–3715).

Liu, Z., Hu, H., Lin, Y., Yao, Z., Xie, Z., Wei, Y., ... Guo, B. (2022). Swin transformer v2: Scaling up capacity and resolution. In *International conference on computer vision and pattern recognition (cvpr)*.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., ... Guo, B. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the ieee/cvf international conference on computer vision* (pp. 10012–10022).

Loh, W.-Y. (2011). Classification and regression trees. *Wiley interdisciplinary reviews: data mining and knowledge discovery*, *1*(1), 14–23.

Long, M., Cao, Z., Wang, J., & Yu, P. S. (2017). Learning multiple tasks with multilinear relationship networks. *Advances in neural information processing systems*, *30*.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, *60*, 91–110.

Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Advances in neural information processing systems*, *30*.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., & Vladu, A. (2017). Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*.

Man, K., & Chahl, J. (2022). A review of synthetic image data and its use in computer vision. *Journal of Imaging*, *8*(11), 310.

Maslej, N., Fattorini, L., Brynjolfsson, E., Etchemendy, J., Ligett, K., Lyons, T., ... others (2023). The ai index 2023 annual report. *arXiv preprint arXiv:2310.03715*.

Mehta, S., & Rastegari, M. (2021). Mobilevit: light-weight, general-purpose, and mobile-friendly vision transformer. *arXiv preprint arXiv:2110.02178*.

Mehta, S., & Rastegari, M. (2022). Separable self-attention for mobile vision transformers. *arXiv preprint arXiv:2206.02680*.

Mikolajczyk, K., & Schmid, C. (2005). A performance evaluation of local descriptors. *IEEE transactions on pattern analysis and machine intelligence*, *27*(10), 1615–1630.

Mitash, C., Bekris, K. E., & Boularias, A. (2017). A self-supervised learning system for object detection using physics simulation and multi-view pose estimation. In *2017 ieee/rsj international conference on intelligent robots and systems (iros)* (pp. 545–551).

Movshovitz-Attias, Y., Kanade, T., & Sheikh, Y. (2016). How useful is photo-realistic rendering for visual learning? In *Computer vision–eccv 2016 workshops: Amsterdam, the netherlands, october 8-10 and 15-16, 2016, proceedings, part iii 14* (pp. 202–217).

Nikolenko, S. I. (2021). *Synthetic data for deep learning* (Vol. 174). Springer.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., ... Lerer, A. (2017). Automatic differentiation in pytorch. *OpenReview*.

Playout, C., Duval, R., Boucher, M. C., & Cheriet, F. (2022). Focused attention in transformers for interpretable classification of retinal images. *Medical Image Analysis*, *82*, 102608.

Qiu, W., & Yuille, A. (2016). Unrealcv: Connecting computer vision to unreal engine. In *Computer vision–eccv 2016 workshops: Amsterdam, the netherlands, october 8-10 and 15-16, 2016, proceedings, part iii 14* (pp. 909–916).

Raghu, M., Unterthiner, T., Kornblith, S., Zhang, C., & Dosovitskiy, A. (2021). Do vision transformers see like convolutional neural networks? *Advances in Neural Information Processing Systems*, *34*, 12116–12128.

Raina, R., Madhavan, A., & Ng, A. Y. (2009). Large-scale deep unsupervised learning using graphics processors. In *Proceedings of the 26th annual international conference on machine learning* (pp. 873–880).

Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., & Chen, M. (2022). Hierarchical text-conditional image generation with clip latents, 2022. *OpenAI*, *7*.

Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., ... Sutskever, I. (2021). Zero-shot text-to-image generation. In *International conference on machine learning* (pp. 8821–8831).

Rebuffi, S.-A., Kolesnikov, A., Sperl, G., & Lampert, C. H. (2017). icarl: Incremental classifier and representation learning. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 2001–2010).

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 1135–1144).

Richter, S. R., Vineet, V., Roth, S., & Koltun, V. (2016). Playing for data: Ground truth from computer games. In *Computer vision–eccv 2016: 14th european conference, amsterdam, the netherlands, october 11-14, 2016, proceedings, part ii 14* (pp. 102–118).

Riegler, G., Urschler, M., Ruther, M., Bischof, H., & Stern, D. (2015). Anatomical landmark detection in medical applications driven by synthetic data. In *Proceedings of the ieee international conference on computer vision workshops* (pp. 12–16).

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., & Ommer, B. (2022). High-resolution image synthesis with latent diffusion models. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (pp. 10684–10695).

Rosten, E., & Drummond, T. (2006). Machine learning for high-speed corner detection. In *Computer vision–eccv 2006: 9th european conference on computer vision, graz, austria, may 7-13, 2006. proceedings, part i 9* (pp. 430–443).

Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. (2011). Orb: An efficient alternative to sift or surf. In *2011 international conference on computer vision* (pp. 2564–2571).

Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature machine intelligence*, *1*(5), 206–215.

Ruiz, M., Fontinele, J., Perrone, R., Santos, M., & Oliveira, L. (2019). A tool for building multi-purpose and multi-pose synthetic data sets. In *Vipimage 2019: Proceedings of the vii eccomas thematic conference on computational vision and medical image processing, october 16–18, 2019, porto, portugal* (pp. 401–410).

Sabour, S., Frosst, N., & Hinton, G. E. (2017). Dynamic routing between capsules. *Advances in neural information processing systems*, *30*.

Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E. L., ... Salimans, T. (2022). Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, *35*, 36479–36494.

Sakaridis, C., Dai, D., & Van Gool, L. (2018). Semantic foggy scene understanding with synthetic data. *International Journal of Computer Vision*, *126*, 973–992.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 4510–4520).

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2017). Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the ieee international conference on computer vision* (pp. 618–626).

Sermanet, P., & LeCun, Y. (2011). Traffic sign recognition with multi-scale convolutional networks. In *The 2011 international joint conference on neural networks* (pp. 2809–2813).

Serra, J., Suris, D., Miron, M., & Karatzoglou, A. (2018). Overcoming catastrophic forgetting with hard attention to the task. In *International conference on machine learning* (pp. 4548–4557).

Simonyan, K., Vedaldi, A., & Zisserman, A. (2013). Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*.

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Stallkamp, J., Schlipsing, M., Salmen, J., & Igel, C. (2011). The german traffic sign recognition benchmark: a multi-class classification competition. In *The 2011 international joint conference on neural networks* (pp. 1453–1460).

Sun, B., & Saenko, K. (2014). From virtual to reality: Fast adaptation of virtual object detectors to real domains. In *Bmvc* (Vol. 1, p. 3).

Sundararajan, M., Taly, A., & Yan, Q. (2017). Axiomatic attribution for deep networks. In *International conference on machine learning* (pp. 3319–3328).

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 2818–2826).

Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W., & Abbeel, P. (2017). Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 ieee/rsj international conference on intelligent robots and systems (iros)* (pp. 23–30).

Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., & Jégou, H. (2021). Training data-efficient image transformers & distillation through attention. In *International conference on machine learning* (pp. 10347–10357).

Tremblay, J., To, T., & Birchfield, S. (2018). Falling things: A synthetic dataset for 3d object detection and pose estimation. In *Proceedings of the ieee conference on computer vision and pattern recognition workshops* (pp. 2038–2041).

Tsirikoglou, A., Eilertsen, G., & Unger, J. (2020). A survey of image synthesis methods for visual machine learning. In *Computer graphics forum* (Vol. 39, pp. 426–451).

Vale, D., El-Sharif, A., & Ali, M. (2022). Explainable artificial intelligence (xai) post-hoc explainability methods: Risks and limitations in non-discrimination law. *AI and Ethics*, 1–12.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, *30*.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., ... Funtowicz, M. (2020). Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations* (pp. 38–45).

Wu, T., Zhang, J., Fu, X., Wang, Y., Ren, J., Pan, L., ... Qian, C. (2023). Omniobject3d: Large-vocabulary 3d object dataset for realistic perception, reconstruction and generation. *arXiv preprint arXiv:2301.07525*.

Wu, X., Sun, K., Zhu, F., Zhao, R., & Li, H. (2023). Human preference score: Better aligning text-to-image models with human preference. In *Proceedings of the ieee/cvf international conference on computer vision* (pp. 2096–2105).

Xiao, J., Hays, J., Ehinger, K. A., Oliva, A., & Torralba, A. (2010). Sun database: Large-scale scene recognition from abbey to zoo. In *2010 ieee computer society conference on computer vision and pattern recognition* (pp. 3485–3492).

Xiao, K., Engstrom, L., Ilyas, A., & Madry, A. (2020). Noise or signal: The role of image backgrounds in object recognition. *arXiv preprint arXiv:2006.09994*.

Xie, Z., Zhang, Z., Cao, Y., Lin, Y., Bao, J., Yao, Z., ... Hu, H. (2022). Simmim: A simple framework for masked image modeling. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (pp. 9653–9663).

Xu, Y., Lin, K.-Y., Zhang, G., Wang, X., & Li, H. (2022). Rnnpose: Recurrent 6-dof object pose refinement with robust correspondence field estimation and pose optimization. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (pp. 14880–14890).

Yu, H., & Oh, J. (2021). Self-supervised learning of 3d object understanding by data association and landmark estimation for image sequence. *arXiv preprint arXiv:2104.07077*.

Yu, J., Xu, Y., Koh, J. Y., Luong, T., Baid, G., Wang, Z., ... Ayan, B. K. (2022). Scaling autoregressive models for content-rich text-to-image generation. *arXiv preprint arXiv:2206.10789*, *2*(3), 5.

Zeiler, M. D., & Fergus, R. (2014). Visualizing and understanding convolutional networks. In *Computer vision–eccv 2014: 13th european conference, zurich, switzerland, september 6-12, 2014, proceedings, part i 13* (pp. 818–833).

Zhang, Q., Wu, Y. N., & Zhu, S.-C. (2018). Interpretable convolutional neural networks. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 8827–8836).

Zhang, Q., Yang, Y., Ma, H., & Wu, Y. N. (2019). Interpreting cnns via decision trees. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (pp. 6261–6270).

Zhang, W., Li, R., Zeng, T., Sun, Q., Kumar, S., Ye, J., & Ji, S. (2015). Deep model based transfer and multi-task learning for biological image analysis. In *Proceedings of the 21th acm sigkdd international conference on knowledge discovery and data mining* (pp. 1475–1484).

Zhang, Y., Tiňo, P., Leonardis, A., & Tang, K. (2021). A survey on neural network in-

terpretability. *IEEE Transactions on Emerging Topics in Computational Intelligence*, *5*(5), 726–742.

Zhang, Y., & Yang, Q. (2021). A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering*, *34*(12), 5586–5609.

Zhang, Z., Luo, P., Loy, C. C., & Tang, X. (2014). Facial landmark detection by deep multi-task learning. In *Computer vision–eccv 2014: 13th european conference, zurich, switzerland, september 6-12, 2014, proceedings, part vi 13* (pp. 94–108).

Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., & Torralba, A. (2016). Learning deep features for discriminative localization. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 2921–2929).

Acronyms

AI Artificial Intelligence. 1, 12

BG Background. 67

BRIEF Binary Robust Independent Elementary Features. 5, 121

BRISK Binary Robust Invariant Scalable Keypoints. 5

CAD Computer-Aided Design. 34

CAM Class Activation Mapping. vii, 14, 15

CAV Concept Activation Vector. 17

CCEL Categorical Cross-Entropy Loss. 82, 84, 85

CI Confidence Interval. 53, 70, 88, 127, 128

CNN Convolutional Neural Network. 5–10, 14, 45, 62, 104, 122

DEiT Data Efficient Vision Transformer. 8

FAST Features from Accelerated Segment Test. 5, 121

FN False Negative. 49, 64, 99

GAN Generative Adversarial Network. 21, 102

GLOH Gradient Location and Orientation Histogram. 5

GPSA Gated Positional Self-Attention. 9

GPU Graphics Processing Unit. 6, 86

Grad-CAM Gradient-weighted Class Activation Mapping. 14

HOG Histogram of Oriented Gradients. 5

HVT Hybrid Vision Transformer. 8, 9

IG Integrated Gradients. 15

LIME Local Interpretable Model-agnostic Explanations. 15

LLM Large Language Model. 83

LoRA Low-Rank Adaptation. 82, 83, 85, 86, 105

LRP Layer-wise Relevance Propagation. 15

MIM Masked Image Modelling. 79

MTL Multi-Task Learning. 22, 23, 77, 82

NLP Natural Language Processing. 7

ORB Oriented FAST and Rotated BRIEF. 5

PEFT Parameter-Efficient Fine-Tuning. 82, 83, 85

PMF Probability Mass Function. 41

ReLU Rectified Linear Unit. 80

SHAP Shapely Additive Explanations. 15

SIFT Scale-Invariant Feature Transform. 5

STRobE Swin Transformer (Robust and Explainable). i, viii, ix, 3, 24, 76, 78–84, 86–99, 104–106, 108

SUN Scene Understanding. vii, ix, 32–34, 36–38, 40, 46, 61–64, 67, 71, 74, 80, 99, 102, 104, 132

TCAV Testing with Concept Activation Vectors. 17

TNT Transformer in Transformer. 8

VAE Variational Autoencoder. 21, 102

VBoW Visual Bag of Words. 6

ViT Vision Transformer. vii, 7–9, 15, 77, 120

XAI Explainable AI. 2, 12, 18, 76–78

Glossary

Azimuth An angular measurement, specifically the *horizontal* angle from a specified direction. In this project, the azimuth refers to the rotation of a 3D model along it's local *yaw* axis, relative to it's default orientation in the ShapeNetCore data set. 28, 30–32, 39, 56, 78, 81, 122, 123

Canonical Pose The canonical pose of an object class is the pose from which it is most frequently photographed in real images. 10, 33, 50, 56, 67

Elevation An angular measurement, specifically the *vertical* angle from a specified direction. In this project, the elevation refers to the rotation of a 3D model along it's local *pitch* axis, relative to it's default orientation in the ShapeNetCore data set. 28, 30–32, 39, 53, 56, 78, 81, 122

Explanation Parameters The explanation parameters are (a) object pose (the position and orientation of the image subject), (b) image background, and (c) lighting direction. This project investigates how classification models respond to variation in these parameters (Objective 2 (Existing Model Evaluation)) and aims to train a model that performs better when presented with challenging configurations of the explanation parameters (Objective 3 (Model Training)). 1–3, 9, 10, 18, 22, 24, 25, 40, 41, 44, 45, 47–50, 59, 67, 74, 76–78, 85–88, 96, 98, 100, 103, 106, 108, 109

Facing Direction A angular measurement across multiple axes, combining the azimuth and elevation of an object. The facing direction is independent of rotation along the object's local *roll* axis. 27, 28, 30, 39, 41, 51, 56, 81, 88

Inductive Bias The inductive biases of a machine learning model are the assumptions that the model makes in order to predict outputs that were not encountered in training. In Convolutional Neural Networks, translation invariance is an inductive bias that is built into the architecture of the model due to the way that convolutional layers operate on an input without regard for their current position in the image. 9

Polar Bias The phenomenon by which uniform random sampling of azimuth and elevation angles on a sphere results in a higher density of points near the poles on the vertical

axis when compared to the equator. This is due to the fact that a given change in the azimuth angle corresponds to a smaller change in the actual facing direction near the poles. 28–31

Appendix A

Methodology

A.1 Mapping Between ShapeNetCore and ImageNet Classes

Listed below are the 55 ShapeNetCore classes and the corresponding classes that were identified in ImageNet. The ImageNet classes are listed with their class ID and a textual description of that class ID, and each ShapeNetCore class may correspond to 0 or more ImageNet classes.

Airplane: 404 (airliner), 895 (warplane, military plane).

Bag: 636 (mailbag, postbag), 728 (plastic bag).

Basket: 790 (shopping basket).

Bathtub: 435 (bathtub, bathing tub, bath, tub).

Bed: *None*

Bench: 703 (park bench).

Bicycle: 444 (bicycle-built-for-two, tandem bicycle, tandem), 671 (mountain bike, all-terrain bike, off-roader).

Bin: 412 (ashcan, trash can, garbage can, wastebin, ash bin, ash-bin, ashbin, dustb, trash barrel, trash bin).

Birdhouse: 448 (birdhouse).

Bookshelf: 453 (bookcase), 454 (bookshop, bookstore, bookstall).

Bottle: 440 (beer bottle), 455 (bottlecap), 737 (pop bottle, soda bottle), 898 (water bottle), 907 (wine bottle).

Bowl: 659 (mixing bowl), 809 (soup bowl).

Bus: 654 (minibus), 779 (school bus), 874 (trolleybus, trolley coach, trackless trolley).

Cabinet: 495 (china cabinet, china closet), 648 (medicine chest, medicine cabinet).

Camera: 732 (Polaroid camera, Polaroid Land camera), 759 (reflex camera).

Can: 653 (milk can).

Cap: 433 (bathing cap, swimming cap), 793 (shower cap).

Car: 705 (passenger car, coach, carriage), 751 (racer, race car, racing car), 817 (sports car, sport car).

Cellphone: 487 (cellular telephone, cellular phone, cellphone, cell, mobile phone).

Chair: 765 (rocking chair, rocker), 423 (bar-

ber chair), 559 (folding chair).

Clock: 409 (analog clock), 530 (digital clock), 892 (wall clock).

Dishwasher: 534 (dishwasher, dish washer, dishwashing machine).

Display: 664 (monitor).

Earphone: *None*

Faucet: *None*

File Cabinet: 553 (file, file cabinet, filing cabinet).

Guitar: 402 (acoustic guitar), 546 (electric guitar).

Helmet: 560 (football helmet), 518 (crash helmet).

Jar: *None*

Keyboard: 508 (computer keyboard, keypad).

Knife: *None*

Lamp: 846 (table lamp), 619 (lampshade, lamp shade).

Laptop: 620 (laptop, laptop computer).

Loudspeaker: 632 (loudspeaker, speaker, speaker unit, loudspeaker system, speaker system).

Mailbox: 637 (mailbox, letter box).

Microphone: 650 (microphone, mike).

Microwave: 651 (microwave, microwave oven).

Motorbike: 670 (motor scooter, scooter).

Mug: 504 (coffee mug).

Piano: 579 (grand piano, grand), 881 (upright, upright piano).

Pillow: 721 (pillow).

Pistol: 763 (revolver, six-gun, six-shooter).

Pot: 738 (pot, flowerpot).

Printer: 742 (printer).

Remote: 761 (remote control, remote).

Rifle: 413 (assault rifle, assault gun), 764 (rifle).

Rocket: *None*

Skateboard: *None*

Sofa: 831 (studio couch, day bed).

Stove: 827 (stove).

Table: 532 (dining table, board), 736 (pool table, billiard table, snooker table).

Telephone: 528 (dial telephone, dial phone), 707 (pay-phone, pay-station).

Tower: 900 (water tower).

Train: 466 (bullet train, bullet).

Washer: 897 (washer, automatic washer, washing machine), 534 (dishwasher, dish washer, dishwashing machine).

Watercraft: 554 (fireboat), 576 (gondola), 625 (lifeboat), 693 (paddle, boat paddle), 814 (speedboat), 833 (submarine, pigboat, sub, U-boat).

Appendix B

Additional Model Evaluation Results

B.1 Evaluation of Existing Models

B.1.1 Invariance to Object Pose (Rotation)

Mean Accuracy Over Rotation-space: Corresponds with the *Swin Transformer V2* results presented in Figure 4.4. In these figures the mean is computed across classes, meaning that accuracy was calculated per-class then averaged across the 48 classes. Rotations are measured in 10° increments relative to the default orientations shown in Figure 3.5. *MobileNet V2:* Figure B.1, *ResNet:* Figure B.2, *MobileViT V2:* Figure B.3.

Marginal Accuracy Over Axes of Rotation: Corresponds with Figure 4.5. In these visualisations each bin labelled i spans the range $[i°, i + 10°)$. *MobileNet V2:* Figure B.4, *ResNet:* Figure B.5, *MobileViT V2:* Figure B.6.

Accuracy Over Rotation-space for Specific Objects: Corresponds with the *Swin Transformer V2* results in Figure 4.6, comparing the results across the four models. In these visualisations rotations are measured in 20° increments relative to the default orientations shown in Figure 3.5. *Table:* Figure B.7, *Rifle:* Figure B.8, *Lamp:* Figure B.9, *Remote:* Figure 4.9.

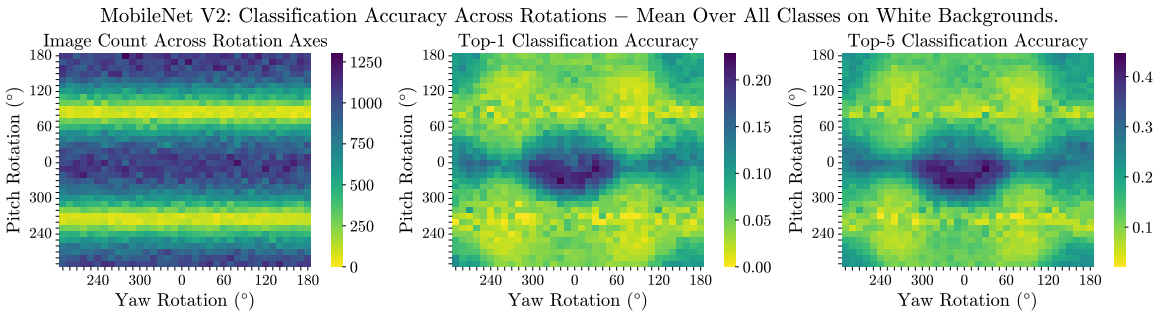


Figure B.1: Mean classification accuracy of the *MobileNet V2* model across rotation-space.

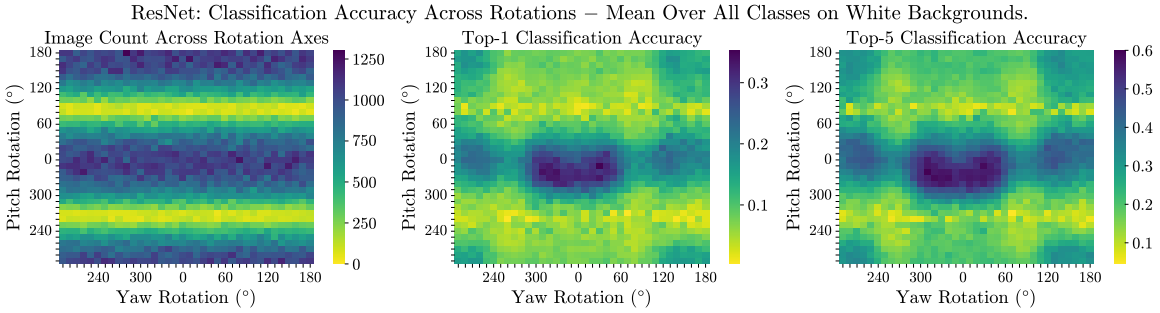


Figure B.2: Mean classification accuracy of the *ResNet* model across rotation-space.

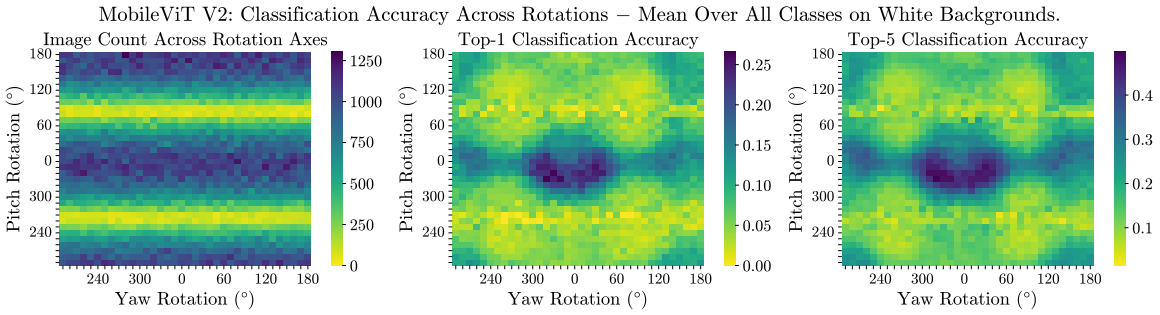


Figure B.3: Mean classification accuracy of the *MobileViT V2* model across rotation-space.

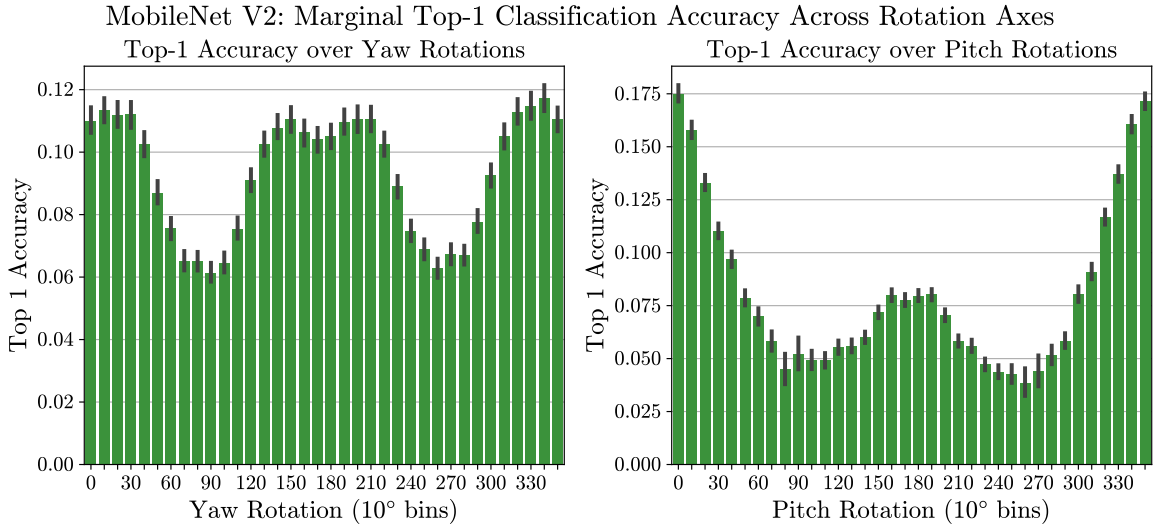


Figure B.4: *MobileNet V2:* Marginal distribution of accuracy across the individual yaw and pitch axes with 95% Confidence Interval (CI) for the entire white background data set.

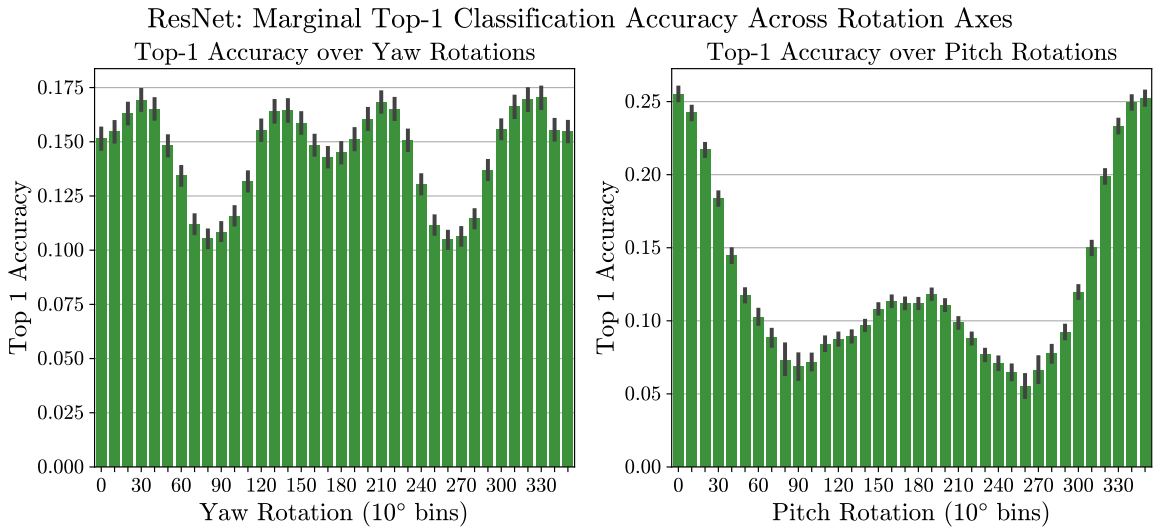


Figure B.5: *ResNet:* Marginal distribution of accuracy across the individual yaw and pitch axes with 95% CI for the entire white background data set.

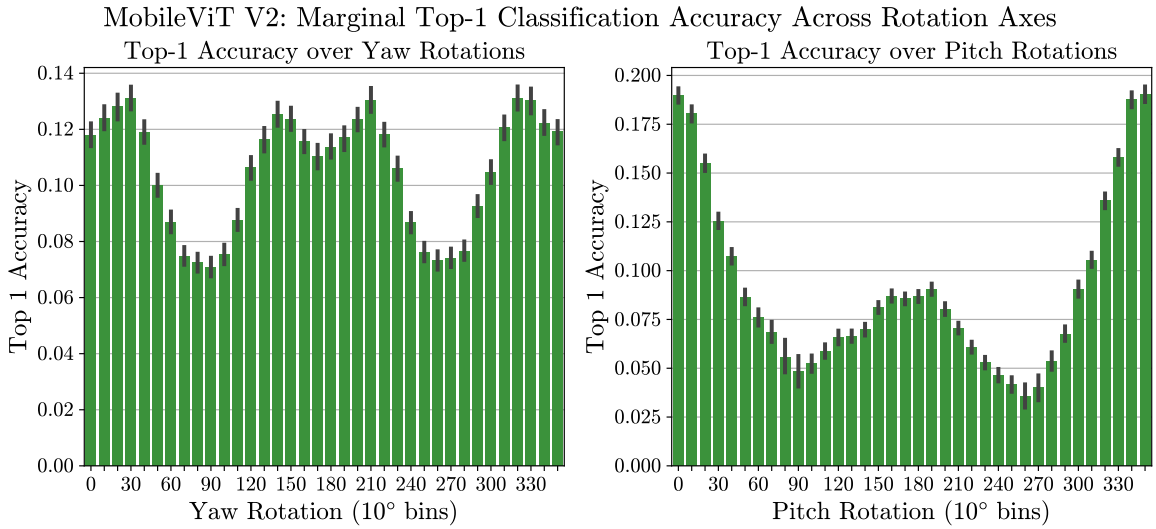
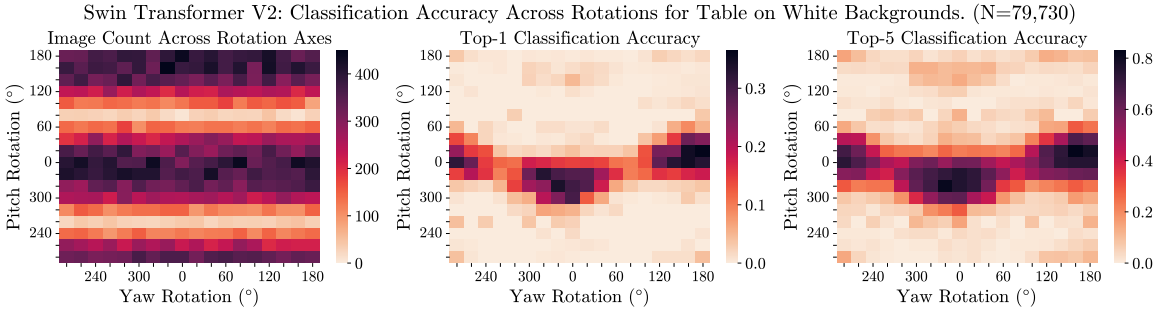
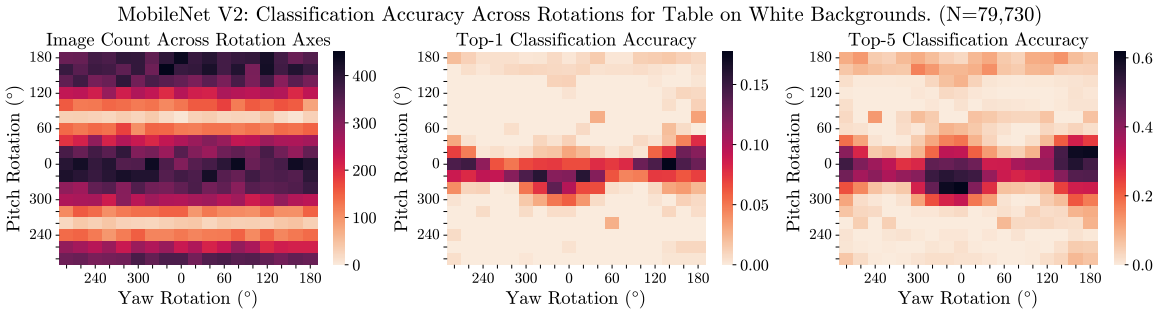


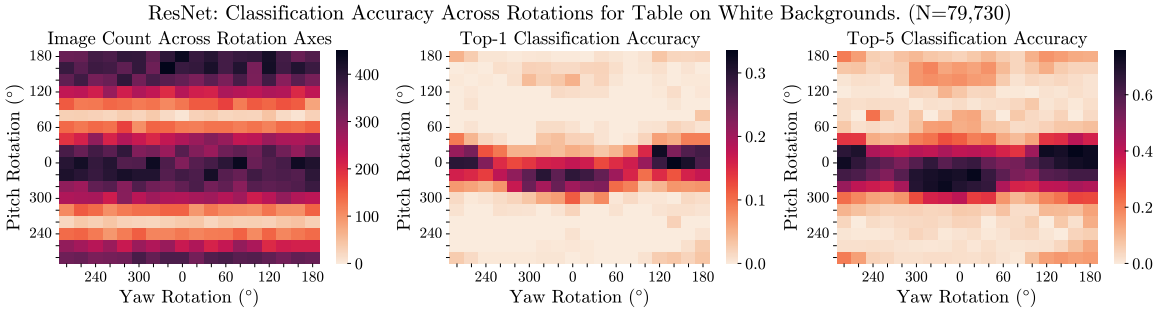
Figure B.6: *MobileViT V2:* Marginal distribution of accuracy across the individual yaw and pitch axes with 95% CI for the entire white background data set.



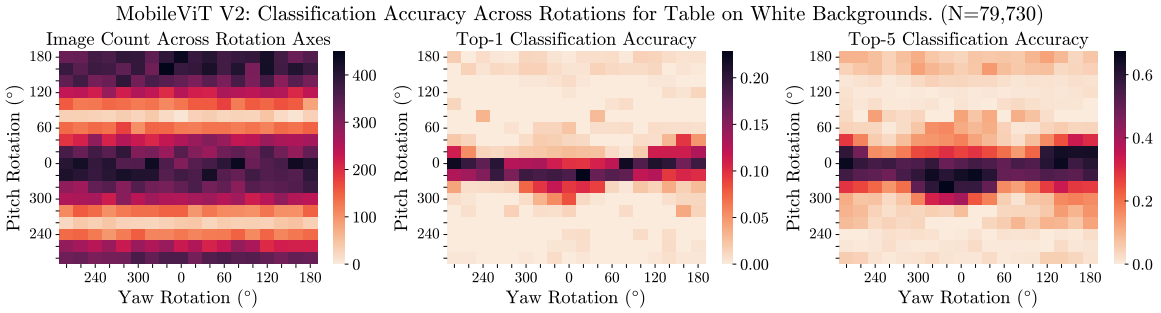
(a) *Swin Transformer V2:* Table accuracy over rotations on white backgrounds.



(b) *MobileNet V2:* Table accuracy over rotations on white backgrounds.

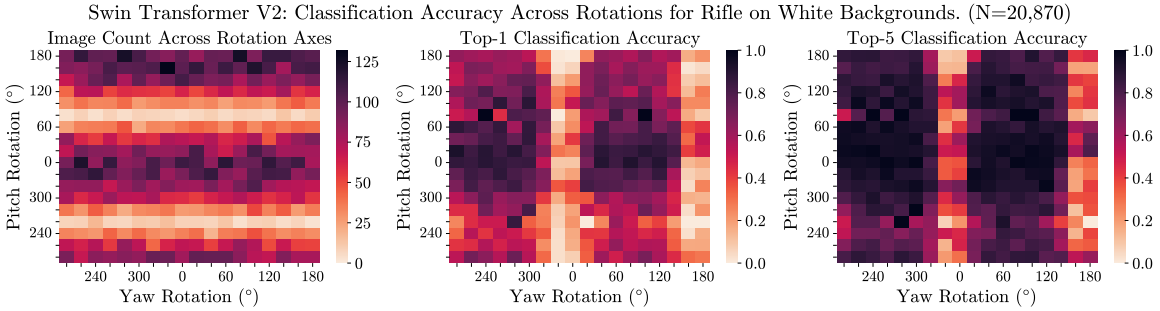


(c) *ResNet:* Table accuracy over rotations on white backgrounds.

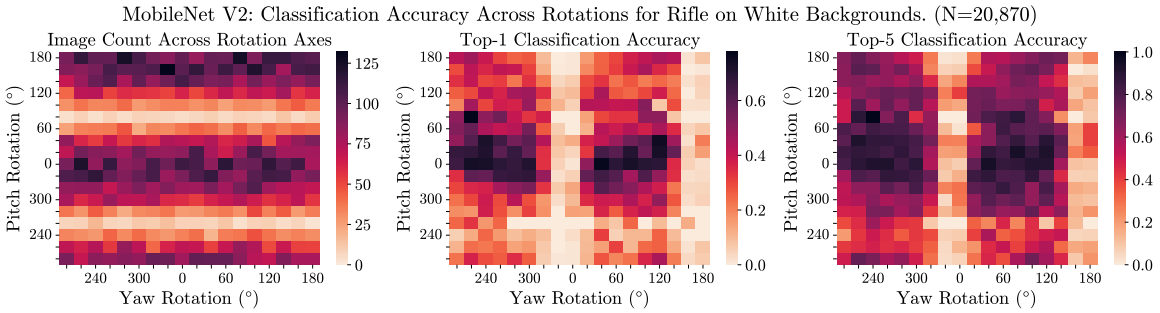


(d) *MobileViT V2:* Table accuracy over rotations on white backgrounds.

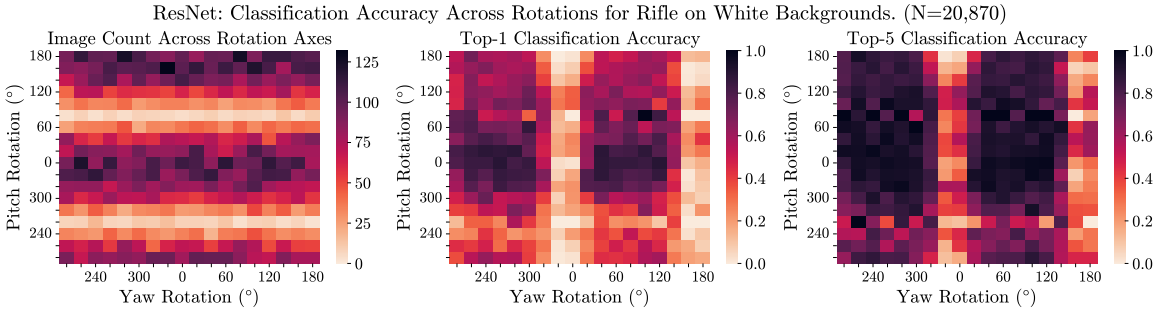
Figure B.7: Classification accuracy of all models over rotation-space for *tables* on white backgrounds.



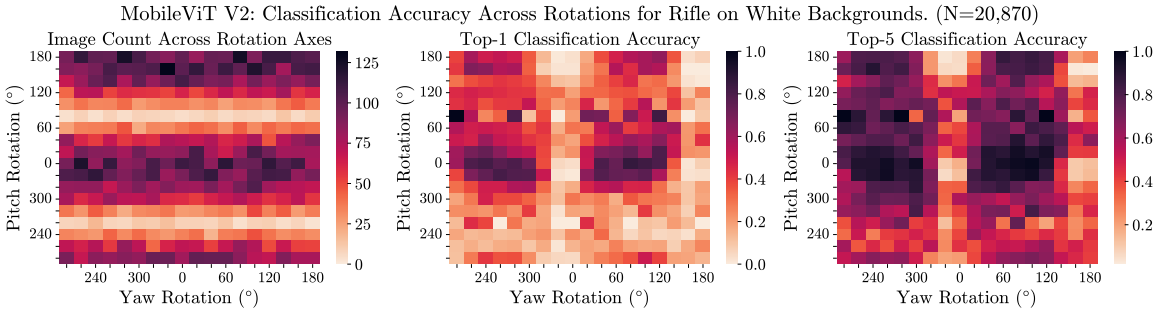
(a) *Swin Transformer V2:* Rifle accuracy over rotations on white backgrounds.



(b) *MobileNet V2:* Rifle accuracy over rotations on white backgrounds.

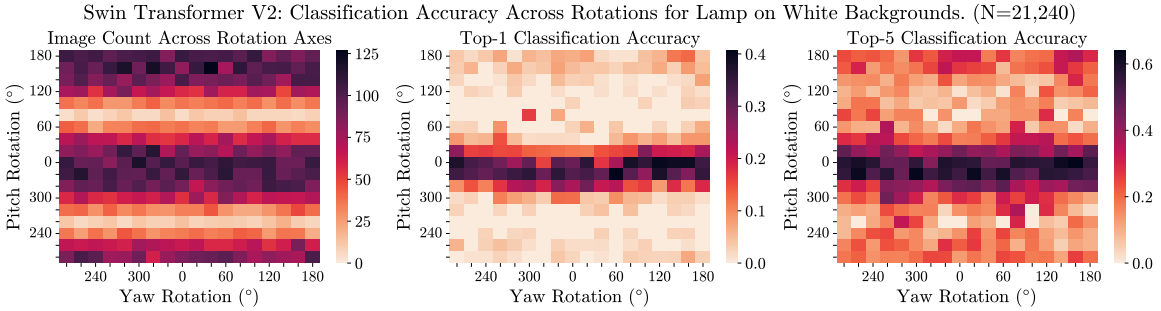


(c) *ResNet:* Rifle accuracy over rotations on white backgrounds.

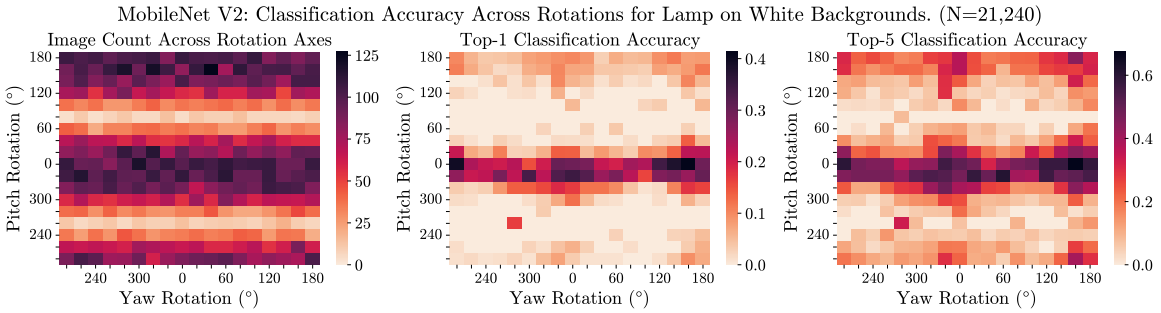


(d) *MobileViT V2:* Rifle accuracy over rotations on white backgrounds.

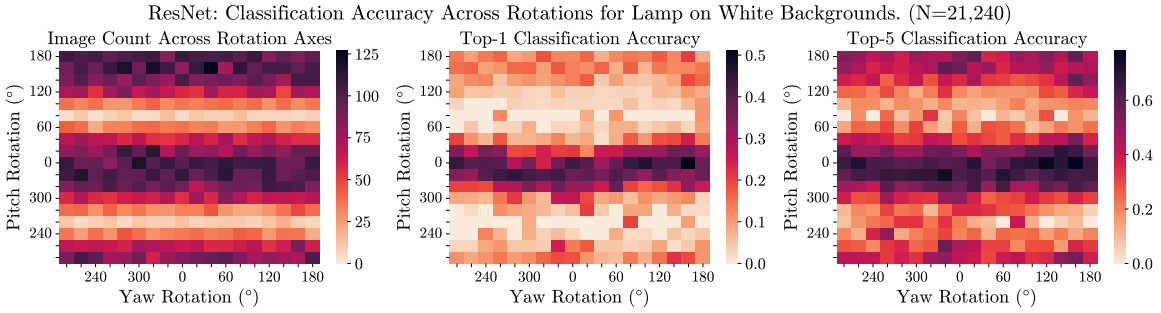
Figure B.8: Classification accuracy of all models over rotation-space for *rifle* on white backgrounds.



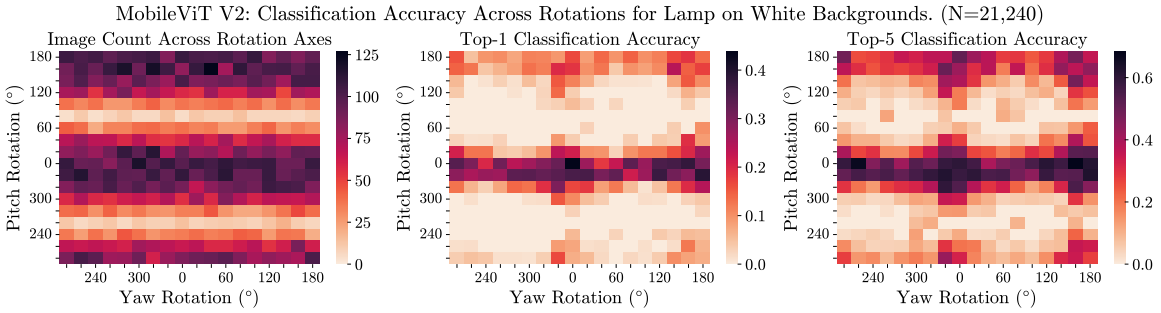
(a) *Swin Transformer V2:* Lamp accuracy over rotations on white backgrounds.



(b) *MobileNet V2:* Lamp accuracy over rotations on white backgrounds.



(c) *ResNet:* Lamp accuracy over rotations on white backgrounds.



(d) *MobileViT V2:* Lamp accuracy over rotations on white backgrounds.

Figure B.9: Classification accuracy of all models over rotation-space for *lamps* on white backgrounds.

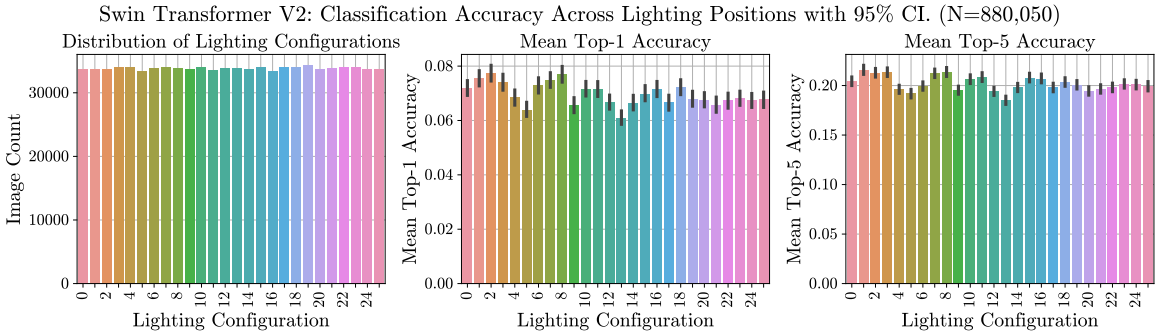
B.1.2 Invariance to Lighting Direction

Classification Accuracy Over Individual Lighting Configurations: The visualisations in Figure B.10, corresponding to Figure 4.17a compare the mean accuracy of each model over each of the 26 individual lighting positions. The accuracy measurements are taken on the synthetic data set with Scene Understanding (SUN) backgrounds.

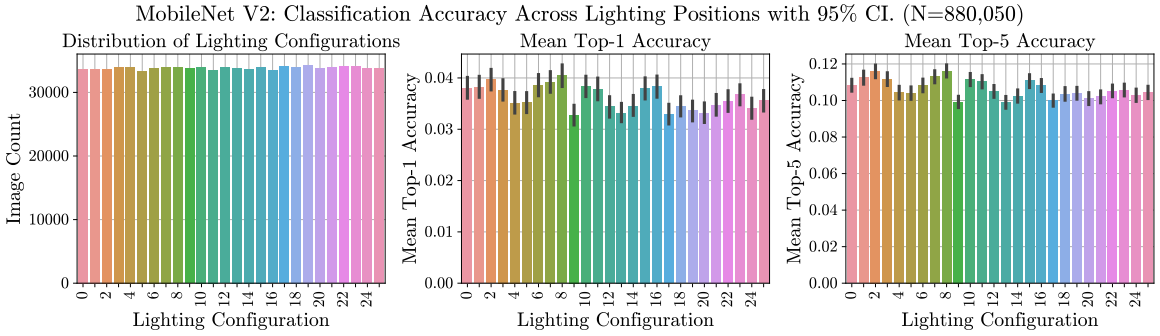
Classification Accuracy Over Configuration Groups: Figures B.11, B.12, and B.13 compare the classification accuracy between the four models on different directional groups of lighting configurations. These correspond to Figures 4.17b–4.17d presented for the *Swin Transformer V2* model in Section 4.2.5. The accuracy measurements are taken on the synthetic data set with SUN backgrounds.

B.1.3 Scale Invariance

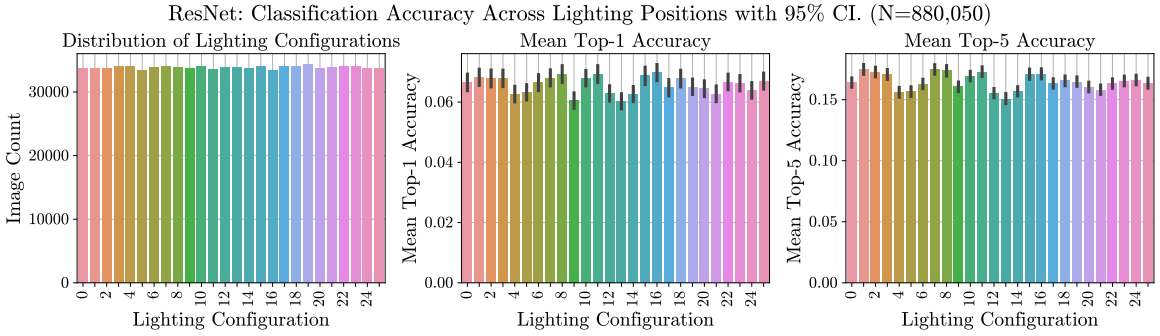
Classification Accuracy Over Scale: Figure B.14 visualises the accuracy across the four classification models showing how they respond to varying the scale of the image subject between 90 and 224 pixels in a 224×224 pixel synthetic image. These figures correspond to Figure 4.20 and show the results for the synthetic data set with SUN backgrounds.



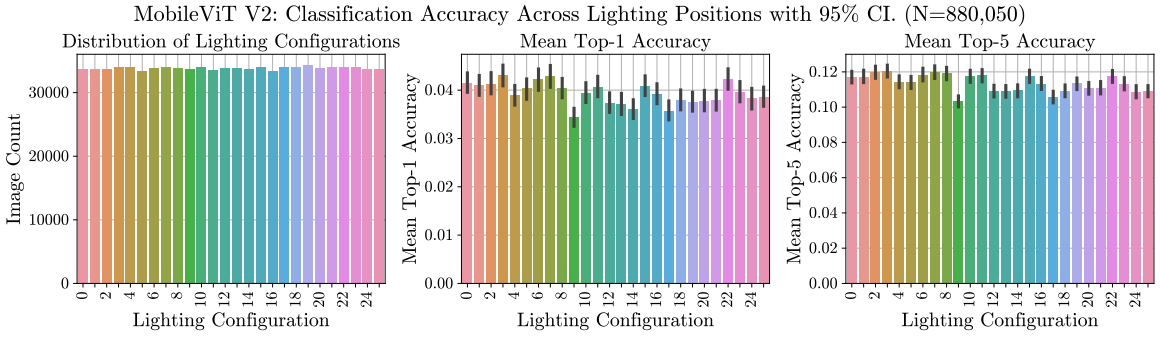
(a) *Swin Transformer V2:* All-class mean accuracy over the 26 lighting configurations.



(b) *MobileNet V2:* All-class mean accuracy over the 26 lighting configurations.

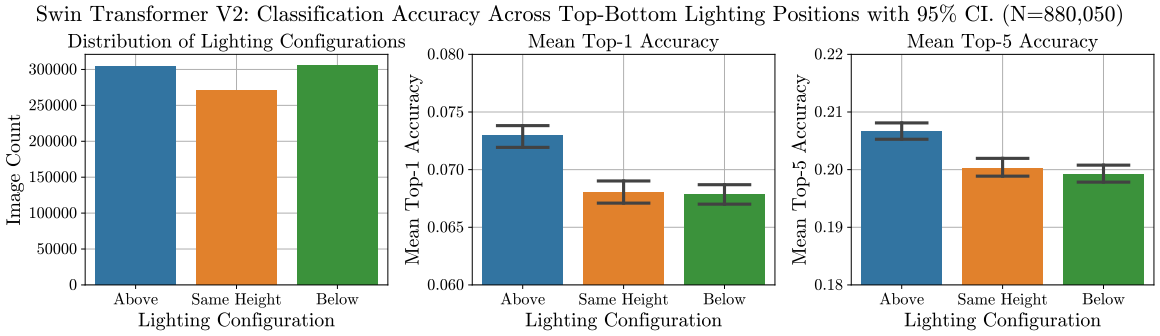


(c) *ResNet:* All-class mean accuracy over the 26 lighting configurations.

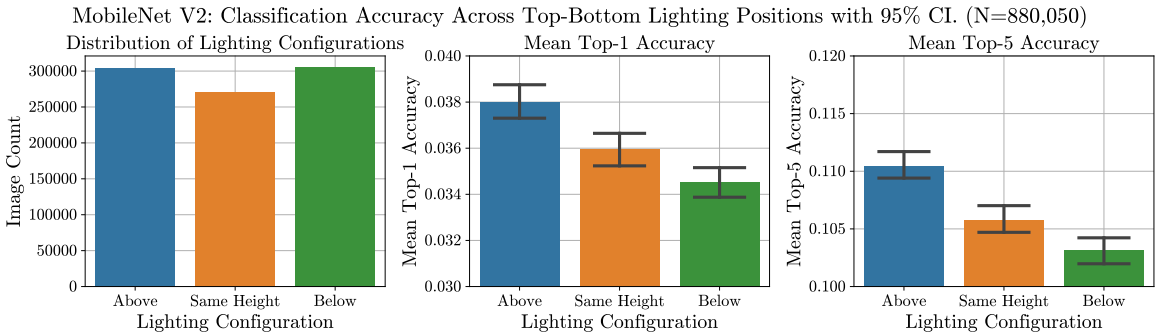


(d) *MobileViT V2:* All-class mean accuracy over the 26 lighting configurations.

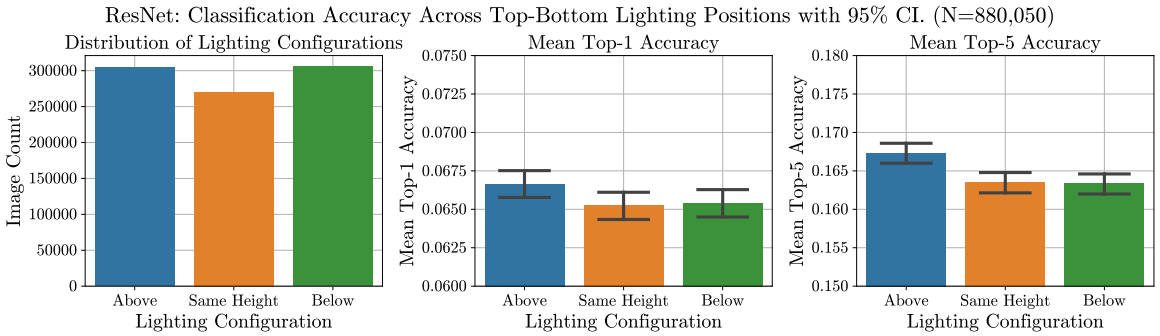
Figure B.10: Mean accuracy over the 26 lighting configurations for the four evaluated models.



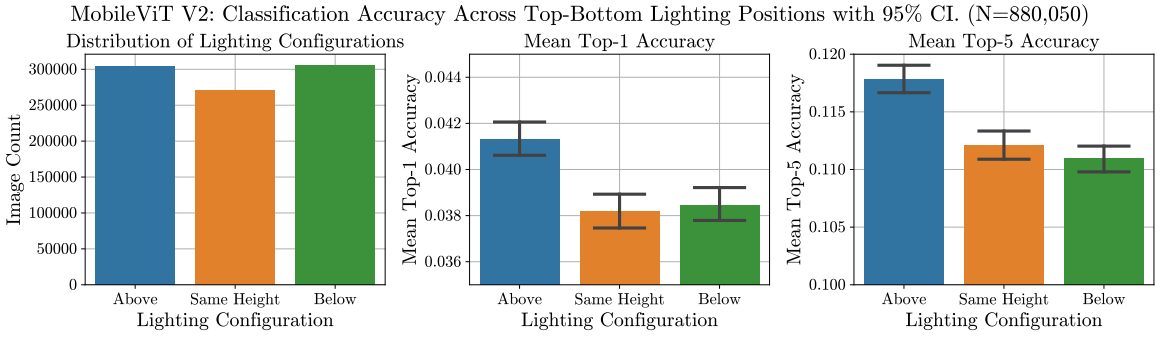
(a) *Swin Transformer V2:* All-class mean accuracy over the Top-Bottom lighting groups.



(b) *MobileNet V2:* All-class mean accuracy over the Top-Bottom lighting groups.

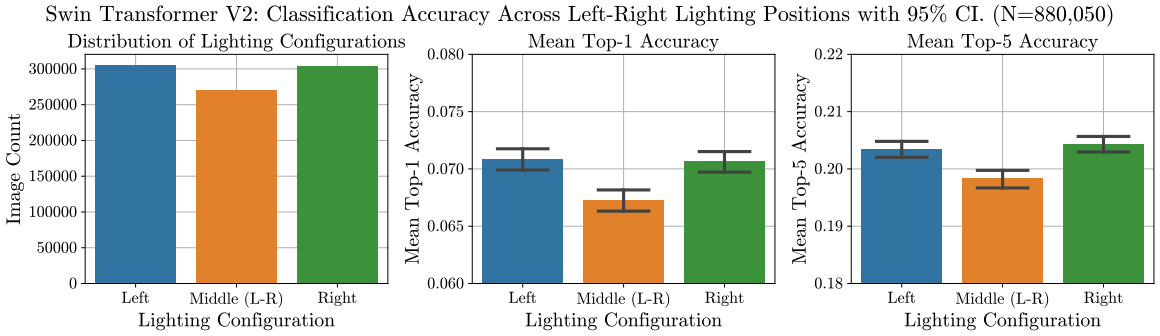


(c) *ResNet:* All-class mean accuracy over the Top-Bottom lighting groups.

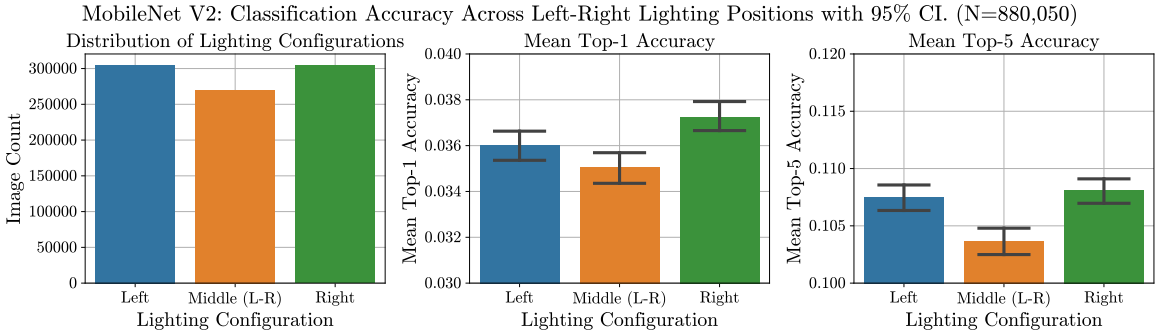


(d) *MobileViT V2:* All-class mean accuracy over the Top-Bottom lighting groups.

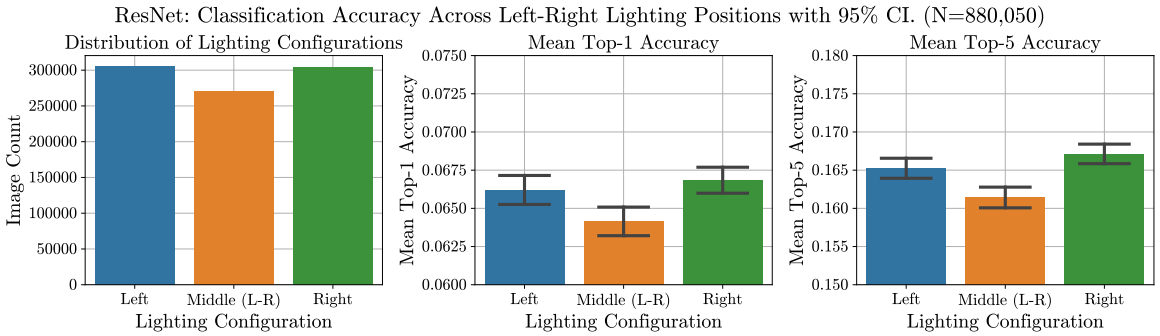
Figure B.11: Mean accuracy over the Top-Bottom lighting groups for the 4 evaluated models. *NB: axes are truncated and axis limits vary between models.*



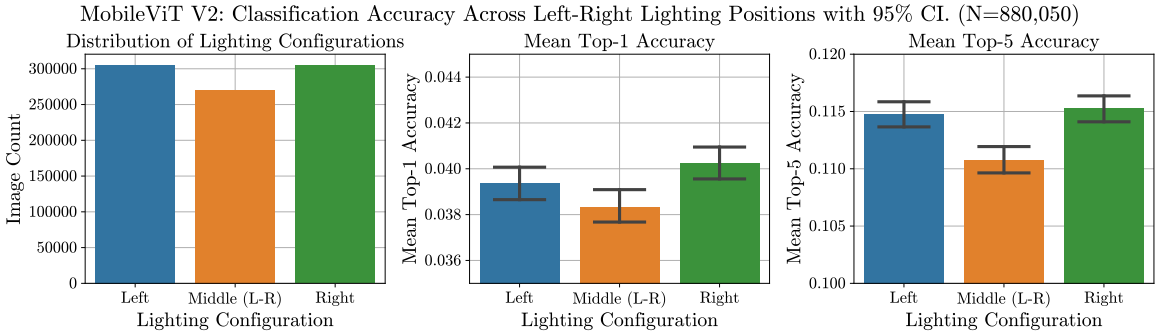
(a) *Swin Transformer V2:* All-class mean accuracy over the Left-Right lighting groups.



(b) *MobileNet V2:* All-class mean accuracy over the Left-Right lighting groups.

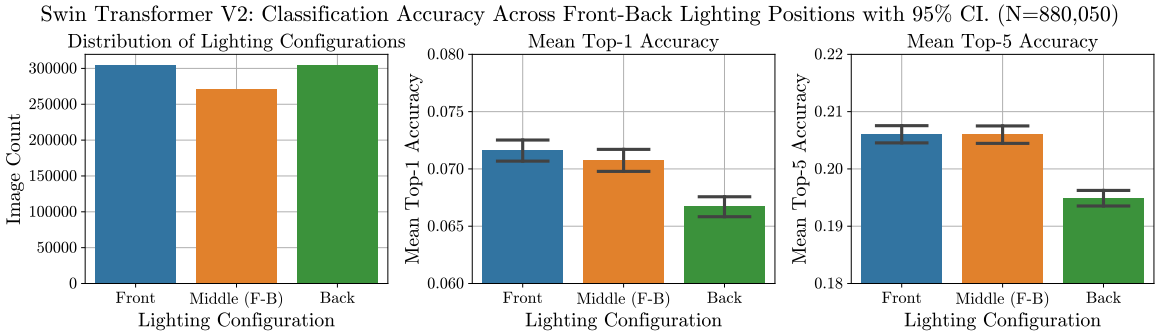


(c) *ResNet:* All-class mean accuracy over the Left-Right lighting groups.

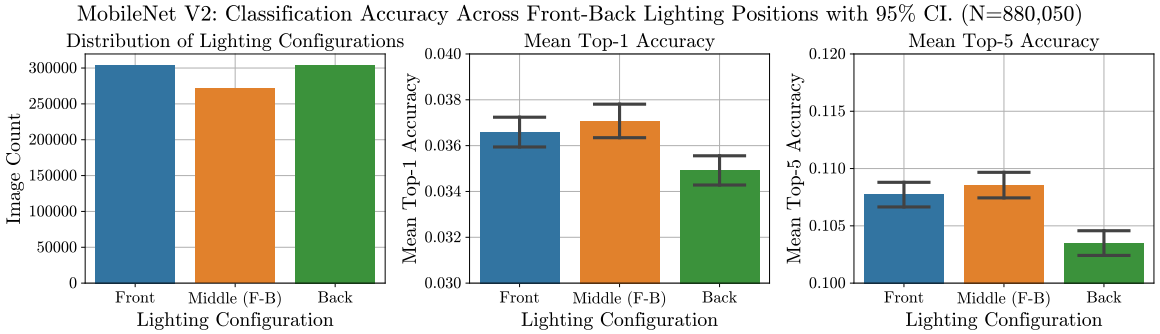


(d) *MobileViT V2:* All-class mean accuracy over the Left-Right lighting groups.

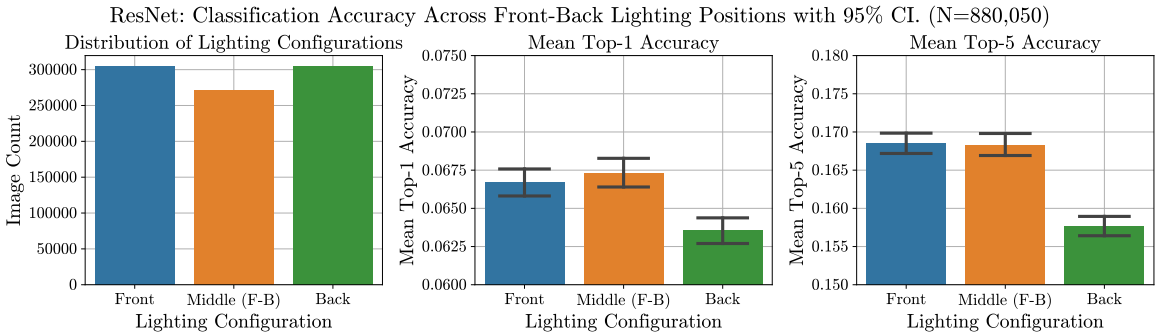
Figure B.12: Mean accuracy over the Left-Right lighting groups for the 4 evaluated models. *NB: axes are truncated and axis limits vary between models.*



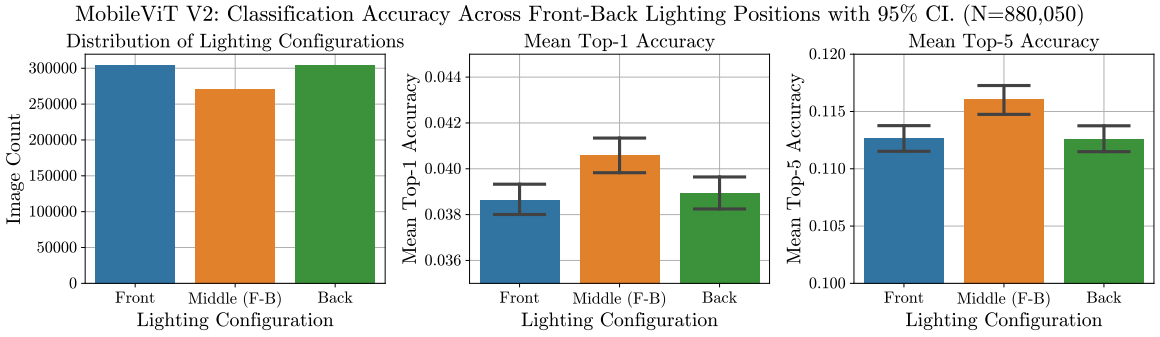
(a) *Swin Transformer V2:* All-class mean accuracy over the Front-Back lighting groups.



(b) *MobileNet V2:* All-class mean accuracy over the Front-Back lighting groups.

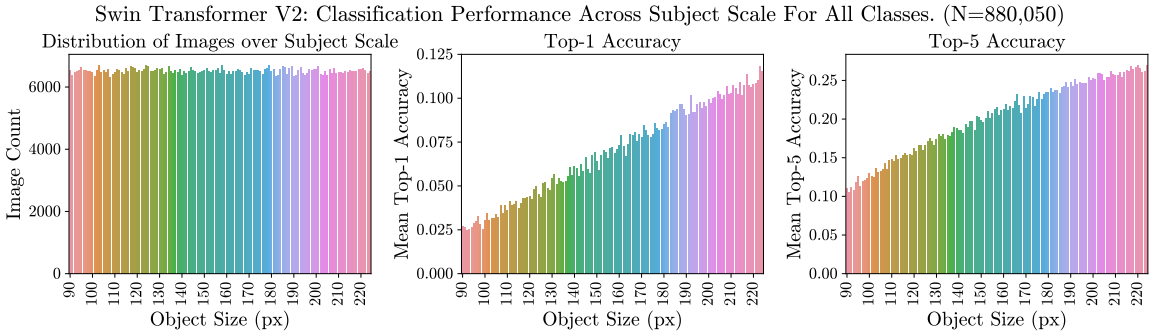


(c) *ResNet:* All-class mean accuracy over the Front-Back lighting groups.

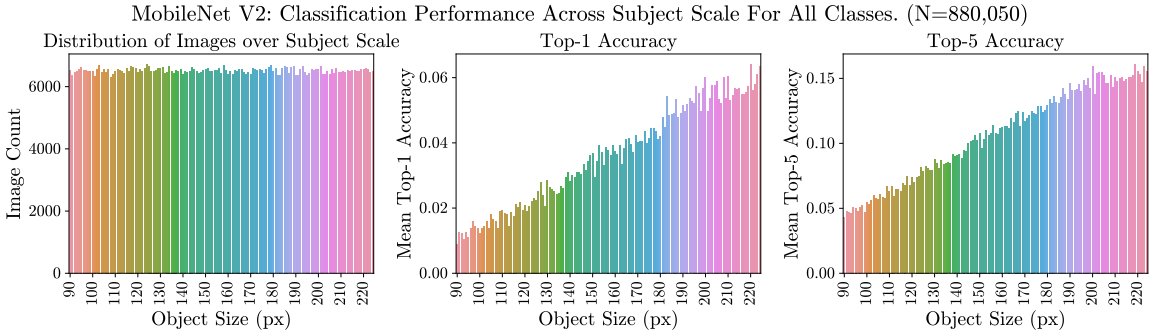


(d) *MobileViT V2:* All-class mean accuracy over the Front-Back lighting groups.

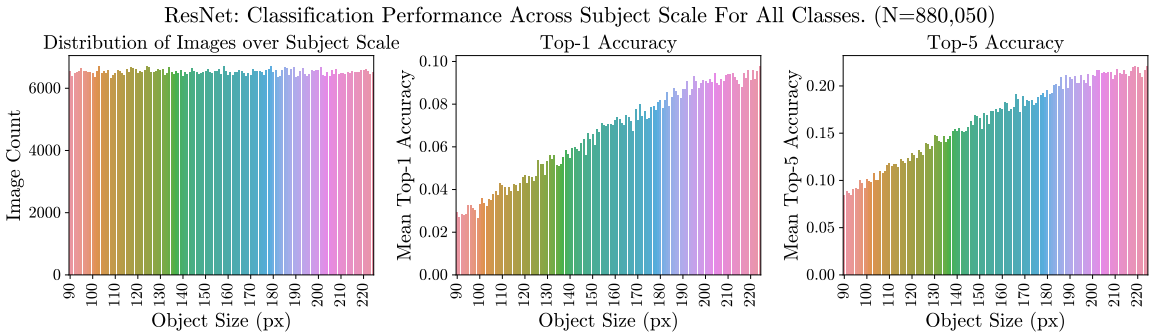
Figure B.13: Mean accuracy over the Front-Back lighting groups for the 4 evaluated models. *NB: axes are truncated and axis limits vary between models.*



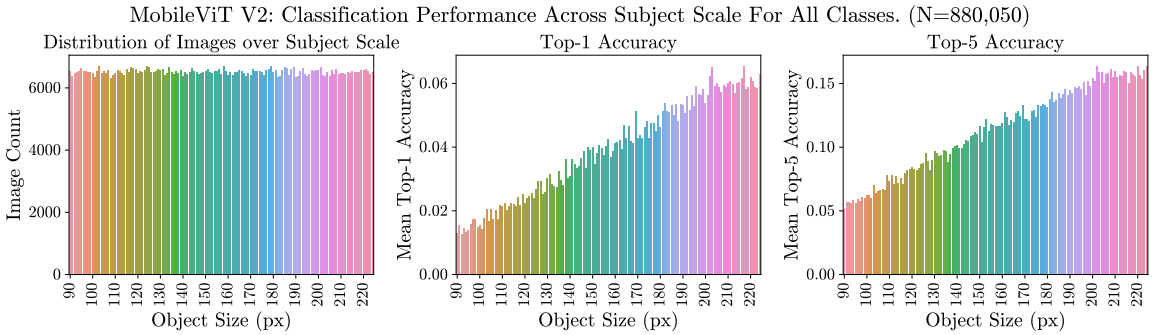
(a) *Swin Transformer V2:* All-class mean accuracy over the scale of the image subject.



(b) *MobileNet V2:* All-class mean accuracy over the scale of the image subject.



(c) *ResNet:* All-class mean accuracy over the scale of the image subject.



(d) *MobileViT V2:* All-class mean accuracy over the scale of the image subject.

Figure B.14: Comparison across all models of mean accuracy over the scale of the image subject in composited images.