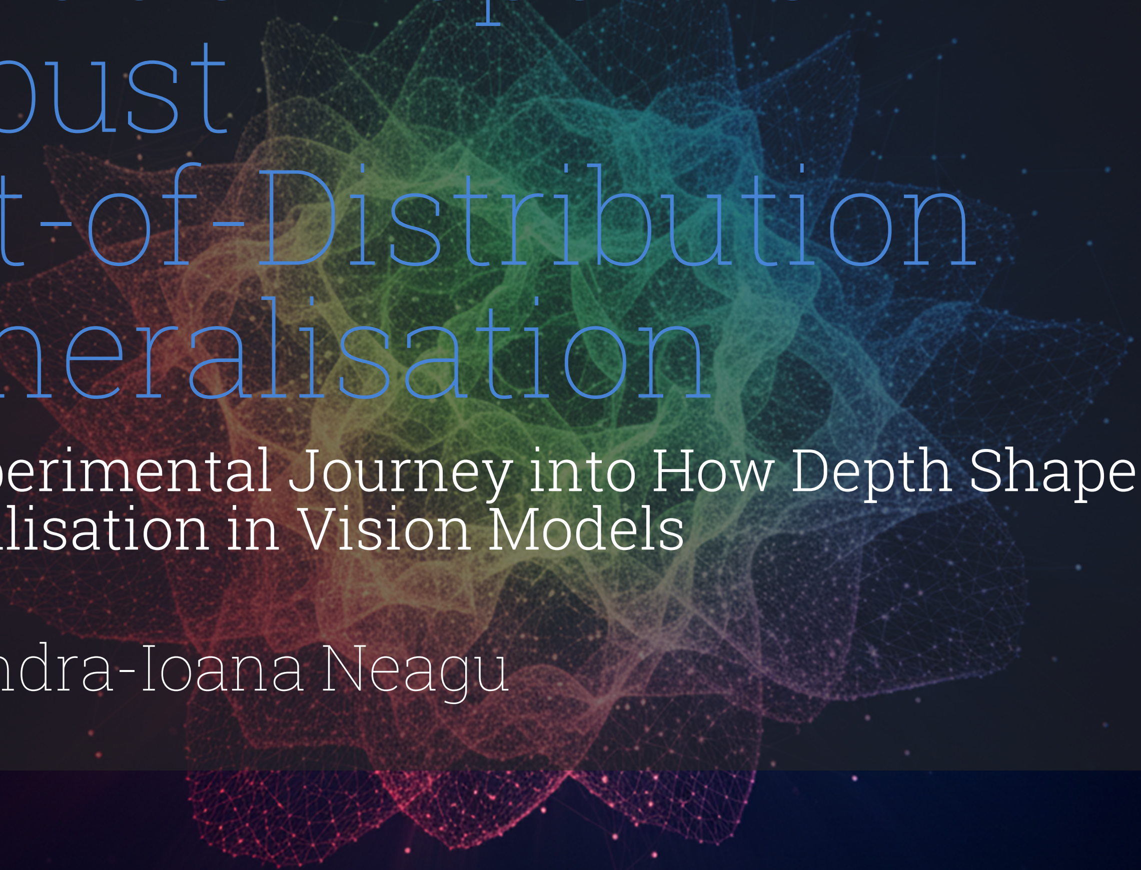
Understanding the Value of Depth: RGB-D Fusion and Pseudo-Depth for Robust Out-of-Distribution Generalisation



An Experimental Journey into How Depth Shapes Generalisation in Vision Models

Alexandra-Ioana Neagu

Delft University of Technology

Understanding the Value of Depth: RGB-D Fusion and Pseudo-Depth for Robust Out-of-Distribution Generalisation

An Experimental Journey into How Depth Shapes Generalisation in Vision Models

by

Alexandra-Ioana Neagu

Student Name
Alexandra-Ioana Neagu

Supervisors:	Dr. J. van Gemert, S. Gielisse
Graduation committee:	Dr. J. Van Gemert (Associate Professor, TU Delft)
	C. Brandt (Assistant Professor, TU Delft)
	S. Gielisse (PhD Candidate, TU Delft)
Faculty:	Electrical Engineering, Mathematics and Computer Science, Delft

Cover:	Art generated by Google Gemini (Modified by the author)
Style:	TU Delft Report Style, with modifications by Daan Zwaneveld

Preface

This thesis represents the culmination of my Master of Science studies at Delft University of Technology, conducted within the Computer Vision Lab of the Faculty of Electrical Engineering, Mathematics and Computer Science. The work presented here is the result of several months of research, experimentation, and reflection, and marks the final step of an academic journey that has been both intellectually demanding and deeply formative.

I would like to express my sincere gratitude to my supervisors, Dr. Jan van Gemert and Sander Gielisse, for their guidance, support, and trust throughout this project. I am particularly grateful to Dr. van Gemert for his ability to consistently challenge assumptions and steer discussions toward the underlying research question. His emphasis on conceptual clarity and scientific rigour strongly influenced how this work was framed, from the formulation of the problem to the interpretation of the results. I am equally thankful to Sander Gielisse for his day-to-day supervision, technical insight, and patience during our countless discussions and iterations. His feedback was invaluable in navigating both conceptual challenges and practical implementation details. Furthermore, I would like to thank Carolin Brandt for joining my thesis committee as an external member.

On a more personal note, I would like to thank my family for their unwavering support throughout my studies. Their encouragement and belief in me provided a constant source of motivation, especially during challenging periods. I am also grateful to my friends in the Netherlands and beyond, whose presence, conversations, and shared experiences made these years not only productive but genuinely meaningful.

Finally, I would like to acknowledge that this thesis is not only a research outcome, but also a learning process. Through this project, I gained a deeper understanding of experimental methodology, scientific writing, and the importance of asking precise questions when working with complex learning systems. I hope that the insights presented here contribute, in a small way, to ongoing discussions on robustness and generalisation in computer vision.

Alexandra-Ioana Neagu
Delft, February 2026

Contents

Nomenclature

Abbreviations

Abbreviation	Definition
CNN	Convolutional Neural Network
DNN	Deep Neural Network
IID	Independent and identically distributed
ID	In-distribution
OOD	Out-of-distribution
RGB	Red, Green, Blue (3-channel colour image)
RGB-D	RGB plus Depth (4-channel input/depth-augmented setting)
RQ	Research Question (RQ1, RQ2, RQ3, …)
SGD	Stochastic Gradient Descent
AdamW	Adam optimiser with decoupled weight decay
Datasets	
MNIST	Modified National Institute of Standards and Technology dataset
NICO++	Non-I.I.D. Image Dataset with Contexts++
CIFAR-10	10-class image classification benchmark dataset
ImageNet	Large-scale image classification dataset/benchmark
NYU Depth (V2)	NYU Depth Dataset (used as monocular depth benchmark)
KITTI	KITTI dataset (used as monocular depth benchmark)
Models	
ResNet-18	18-layer Residual Network
MobileNetV2	MobileNet version 2
ShuffleNetV2	ShuffleNet version 2
EfficientNet-B0	EfficientNet baseline variant B0

Symbols

Symbol	Definition	Unit
H	Image height	px
W	Image width	px
C	Number of channels (or logits dimension, depending on context)	–
K	Number of classes	–
$f : \mathbb{R}^{H\times W\times C} \to \{1,\ldots,K\}$	Image classifier mapping	–
$x \in \mathbb{R}^{3\times H\times W}$	RGB image input tensor	–
$d \in \mathbb{R}^{1\times H\times W}$	Depth map (ground-truth or predicted/pseudo-depth)	–
$o \in \mathbb{R}^{C}$	Class logits vector	–
y	Ground-truth class index (for cross-entropy)	–
o_c	c-th logit component	–
z_{rgb}	RGB feature representation $z_{\text{rgb}} = f_\theta(x)$	–
z_d	Depth feature representation $z_d = g_\phi(d)$	–
f_θ	RGB encoder with parameters θ	–
g_ϕ	Depth encoder with parameters ϕ	–
h_ψ	Classifier head with parameters ψ	–
L_{cls}	Cross-entropy classification loss	–
Δ	Context generalisation gap: Seen Accuracy $-$ Unseen Accuracy	%
$\sigma(\cdot)$	Nonlinear activation function (e.g., ReLU)	–
w, b	Neuron weights and bias	–
$W^{(l)}$, $b^{(l)}$	Layer-l weights and biases	–

Introduction

Modern computer vision systems are now widely used in everyday life. They help cars recognise pedestrians, assist robots in navigating indoor spaces, and allow software to classify images automatically. In many cases, these systems perform impressively well, *as long as the data they see at test time closely resembles the data they were trained on.*

However, a persistent problem emerges when these systems are exposed to **new situations**. A model trained to recognise objects in one environment may fail when the background, lighting, or context changes, even if the object itself remains the same. For example, a system that learns to recognise dogs mostly from images of dogs on grass may struggle to recognise a dog standing in water or snow. This phenomenon is known as **poor generalisation under distribution shift**, or more specifically, **out-of-distribution (OOD) failure**.

This thesis is motivated by a simple but important question: *How can we help vision models rely less on superficial visual cues, and instead learn representations that remain reliable when conditions change?*

Most image classifiers today are trained using **RGB images**, which encode colour and texture. While this information is rich, it can also be misleading. Models often learn shortcuts, statistical regularities in the training data that correlate with the correct label but are not truly essential. For instance, background colour, texture patterns, or lighting conditions may accidentally become strong predictors during training, even though they are irrelevant to the object itself.

Humans, by contrast, rely heavily on **structure and geometry**. We recognise objects by their shape, relative layout, and spatial extent, not only by their colour or texture. One way to encode such structural information in computer vision is through **depth**, a signal that describes how far different parts of the scene are from the camera.

Depth information has long been known to improve robustness in vision systems. However, most existing approaches assume access to specialised sensors (such as RGB-D cameras) or rely on complex model architectures. In many real-world scenarios, only a single RGB image is available.

This leads to the central idea explored in this thesis: *Can* ***estimated depth****, predicted from a single RGB image, act as a simple and practical signal that improves robustness under distribution shift, without redesigning models or requiring extra sensors?*

This thesis studies whether adding a **single predicted depth channel** to standard image classifiers can help them generalise better to unseen conditions. Rather than proposing a new model architecture, the work deliberately focuses on **minimal interventions**, standard convolutional neural networks (CNNs) trained from scratch with depth added only as an extra input channel. The goal is not to maximise performance at all costs, but to **understand causally** whether imperfect geometric information itself helps models move away from brittle, appearance-based shortcuts. To answer this, the thesis systematically compares **RGB-only models**, which see only colour images, and **RGB-D models**, which see colour plus estimated depth. It then evaluates how these models behave when the visual context changes, such as when background environments differ between training and testing.

To make the investigation as clear and interpretable as possible, the thesis progresses in stages. First, it uses **highly controlled toy experiments**, where the source of the distribution shift is explicitly designed. In these experiments, background colour acts as a known shortcut, and depth information is synthetically defined. This setting allows the effect of depth to be isolated very precisely.

Next, the thesis moves to a **real-world benchmark**, where objects appear in a variety of natural contexts. Here, depth is not measured directly, but predicted using a modern monocular depth estimation model. By gradually corrupting this depth signal with noise, the thesis further tests whether performance gains truly depend on geometric structure, or whether they arise merely from adding extra input dimensions. Together, these experiments form a coherent storyline, from simple and interpretable settings to realistic and noisy conditions, all aimed at understanding the role of depth in robust visual recognition.

This thesis report is organised into three main parts. **Part 1 (this chapter)** provides a high-level introduction and motivation. Its purpose is to explain the problem setting, the intuition behind the approach, and the overall structure of the work in a clear and accessible manner. **Part 2** presents the **technical background** required to understand the scientific contribution. This includes background on convolutional neural networks, domain generalisation and out-of-distribution robustness, shortcut learning, RGB-D vision, and depth estimation. These sections introduce the key concepts and terminology used later, without yet presenting new experimental results. **Part 3** contains the **scientific article** that forms the core of the thesis. This part presents the research questions, experimental design, results, and analysis. It builds directly on the concepts introduced in Part 2 and provides empirical evidence addressing the central question of the thesis.

Background

0.1. Image Classification as a Computer Vision Task

Image classification is one of the most fundamental and widely studied problems in computer vision. At its core, the task consists of assigning a discrete label to an input image, typically corresponding to the object, scene, or concept that the image depicts. Formally, given an image represented as a two-dimensional grid of pixel intensity values (with one or more colour channels), an image classification system learns a function $f : \mathbb{R}^{H \times W \times C} \rightarrow \{1, \ldots, K\}$, where H, W, and C are the image height, width, and number of channels, respectively, and K is the number of possible classes. Examples of both single-label and multi-label image classification are shown in Figure 1.

The significance of image classification goes beyond its apparent simplicity. It is a foundational problem upon which more complex computer vision tasks, such as object detection, semantic segmentation, and action recognition, are built. Successful solutions to image classification enable machines to interpret and reason about visual information, a prerequisite for applications in autonomous driving, medical image analysis, robotics, and many other domains.

Historically, image classification relied on manually designed feature extraction pipelines. Classical methods such as Scale-Invariant Feature Transform (SIFT) [46] and Histogram of Oriented Gradients (HOG) [7] produced hand-crafted descriptors to capture salient visual information. These features were then fed into machine learning models such as Support Vector Machines (SVMs) for classification. Although effective in constrained settings, such approaches struggled to generalise across diverse datasets and tasks, due to the limited expressiveness of hand-engineering [20].

The advent of deep learning fundamentally transformed the field. Convolutional Neural Networks (CNNs), first popularised with the LeNet architecture for handwritten digit recognition [41], demonstrated the ability to learn hierarchical feature representations directly from raw pixel data. This approach reached mainstream success with Krizhevsky et al.'s AlexNet (2012), which achieved a breakthrough performance in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [38]. CNNs learn multiple levels of abstraction, with lower layers detecting simple edges and textures, and deeper layers capturing increasingly complex patterns, such as object parts and global shapes [20].

Today, image classification remains a benchmark problem and a proving ground for novel computer vision models. Datasets such as CIFAR-10 [37], MNIST [10], and ImageNet [9] have become standard evaluation tools, each testing different aspects of model capacity, robustness, and scalability. State-of-the-art approaches extend beyond CNNs, incorporating attention mechanisms (e.g., Vision Transformers) [11] and multimodal learning strategies, yet the central challenge remains the same: mapping raw visual input to semantic categories reliably and efficiently.

In the context of this thesis, image classification serves as the foundation for examining how pseudo-depth cues influence generalisation under distribution shifts. Background-coloured datasets like the toy MNIST variants (see subsection 0.6.2 and 0.6.3), as well as natural-context datasets such as NICO++ (subsection 0.6.4), allow us to probe a specific failure mode of image classifiers: the tendency to rely on superficial correlations such as colour or texture. Because the classification pipeline is fixed and well understood, any performance change when augmenting RGB inputs with an estimated depth channel can be directly attributed to the added geometric signal, rather than to confounding factors in the task itself. Thus, image classification provides a controlled yet challenging environment for answering our research question: can depth-like structural information help CNNs focus on object geometry and thereby improve out-of-distribution robustness?

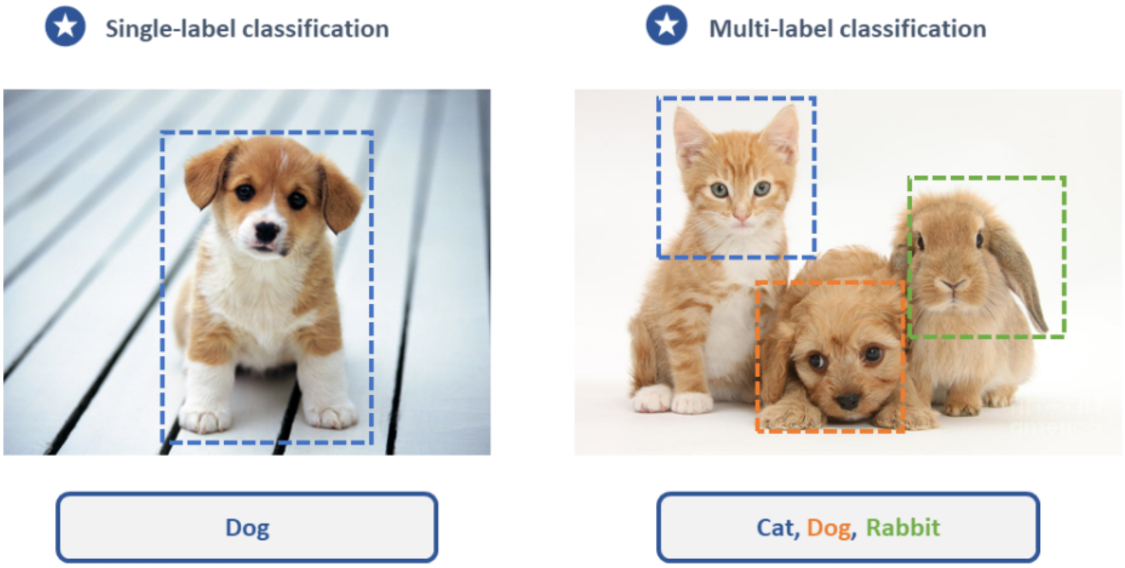


Figure 1: Examples of image classification, taken from [31]. (Left) *Single-label classification*, where the model assigns exactly one category to each image (e.g., ”Dog”). (Right) Multi-label classification, where multiple categories may simultaneously be present in the same image (e.g., ”Cat, Dog, Rabbit”). Dashed boxes indicate detected object regions associated with each label.

0.2. Deep Neural Networks

Deep Neural Networks (DNNs) are a class of machine learning models inspired by the structure and function of the human brain. They are composed of multiple layers of interconnected processing units, called artificial neurons, that collectively transform input data into meaningful output predictions. The “deep” in DNNs refers to the presence of multiple hidden layers between the input and output layers, which allows these models to learn increasingly abstract and complex representations of data [20].

A single artificial neuron can be described as a function that computes a weighted sum of its inputs, applies a bias term, and then passes the result through a non-linear activation function:

$$h(\mathbf{x}) = \sigma(\mathbf{w}^{\top}\mathbf{x} + b), \tag{1}$$

where

- $\mathbf{x} \in \mathbb{R}^d$ is the input vector,
- $\mathbf{w} \in \mathbb{R}^d$ are the learnable weights,
- $b \in \mathbb{R}$ is a bias term,
- $\sigma(\cdot)$ is a nonlinear activation function, such as ReLU

Stacking many such neurons together forms a layer. A feedforward neural network with L layers can be expressed as a nested composition of functions:

$$\mathbf{h}^{(1)} = \sigma(\mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)}), \quad \mathbf{h}^{(2)} = \sigma(\mathbf{W}^{(2)}\mathbf{h}^{(1)} + \mathbf{b}^{(2)}), \quad \ldots, \quad \hat{\mathbf{y}} = f(\mathbf{h}^{(L)}), \tag{2}$$

where $\hat{\mathbf{y}}$ is the model’s output (e.g., class probabilities in classification tasks), and $\mathbf{W}^{(l)}$, $\mathbf{b}^{(l)}$ are learnable parameters at layer l. Figure 2 illustrates a prototypical feedforward deep neural network.

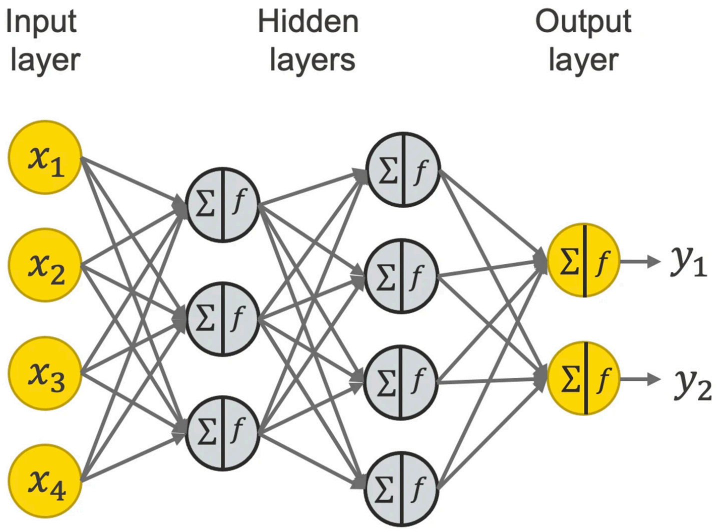


Figure 2: Schematic of a fully connected feedforward DNN, reproduced from [1]. The network consists of an input layer with four units, two hidden layers, and an output layer with two units. Each node represents an artificial neuron, and each directed edge denotes a learnable weighted connection between neurons in adjacent layers, illustrating the dense connectivity that allows the network to combine information from all inputs. Hidden layers apply affine transformations followed by nonlinear activation functions, enabling the network to learn hierarchical feature representations from the input.

While the formulation above provides a general description of deep networks, in this thesis, DNNs play a more specific role. They serve as the computational backbone for the classification models used in our pipelines. In practice, we do not rely on fully connected feedforward networks for image understanding. Instead, we use Convolutional Neural Networks (CNNs) (see section 0.4), which exploit spatial locality through weight sharing and are therefore better suited to visual data. These architectures extract progressively more abstract spatial features, edges, textures, and shapes that form the basis for the RGB(-D) classifiers evaluated throughout this work.

Recent work has shown that DNNs often rely on spurious correlations and shortcut cues that are sufficient for high in-distribution accuracy but fail under domain shift. These representation biases arise naturally from the statistical structure of RGB data and the inductive biases of convolutional architectures. From this perspective, auxiliary signals that emphasise more stable scene properties, such as geometry, may act as a mechanism to counteract shortcut learning rather than merely providing additional information.

0.2.1. Fully Connected Layers

Fully connected (dense) layers are among the simplest components of deep neural networks: each neuron receives input from every unit in the previous layer, allowing the model to combine all features jointly (see Figure 2). Because they make no assumptions about spatial or structural locality, dense layers are highly expressive universal approximators [28], but also extremely parameter-intensive. Mapping a flattened 28 $\times$ 28 MNIST image (d = 784) to even a moderate hidden layer of size k = 1000 already requires 785,000 weights, and scaling to larger images would result in hundreds of millions. This practical limitation is one of the reasons convolutional architectures, which exploit spatial structure, became the standard for image tasks.

In modern CNN-based classifiers, fully connected layers typically appear only at the final stage, transforming the learned spatial feature maps into class logits. This limited but crucial role is the one relevant to this thesis. The classifiers used in our RGB and RGB-D experiments rely on dense layers only as a final classification head, after convolutional layers have extracted object-level features. Likewise, when integrating pseudo-depth, the additional depth channel enriches the feature representation fed into these final fully connected layers, but does not alter their function. Thus, while dense layers are not central to the geometric reasoning explored in this work, they serve as the final decision-making stage that reflects the benefit (or lack thereof) of incorporating depth information into the learned representation.

0.2.2. Representation Depth in Neural Networks

In this subsection, we briefly discuss the notion of *architectural depth* in neural networks, which should not be confused with the notion of *geometric depth* used elsewhere in this thesis to refer to scene structure or depth maps.

Shallow neural networks (with one hidden layer) are universal function approximators [28], meaning they can approximate any continuous function on compact subsets of $\mathbb{R}^n$. However, they often require an impractically large number of units to model complex functions. Deeper networks, by contrast, can represent compositional hierarchies more efficiently. Each additional layer allows the network to capture higher-level abstractions by reusing and recombining features learned in earlier layers [20].

In the context of image classification, early layers typically learn low-level visual primitives such as edges or corners, intermediate layers capture textures or object parts, and deeper layers encode object-level or scene-level semantics. This hierarchical representation learning is a key advantage of DNNs over classical machine learning approaches that rely on hand-crafted features.

0.2.3. Training and Generalisation

The parameters of a DNN are learned by minimising a chosen loss function (e.g., cross-entropy for classification). The optimisation relies on the backpropagation algorithm to compute gradients efficiently [55]. However, the flexibility of DNNs also makes them prone to issues such as overfitting (memorising training data) or underfitting (failing to capture structure). See subsection 0.5.4 for more details on these.

Despite these hurdles, deep networks have achieved state-of-the-art performance across many domains: vision, speech, natural language processing, and even reinforcement learning [40]. Their success is attributed to both algorithmic innovations and the availability of large-scale datasets and computational resources, especially GPUs.

0.3. The Building Blocks of Convolutional Neural Networks

0.3.1. The Convolution Operation

At the heart of Convolutional Neural Networks (CNNs) lies the convolution operation. This operation is designed to extract local patterns from structured data, such as images, by systematically combining small, localised regions of the input with learnable filters (often called kernels). Unlike fully connected layers, where every input unit connects to every output unit, convolution restricts connections to local neighbourhoods. This makes CNNs computationally efficient and particularly well-suited to image data, where spatial locality matters [20].

Mathematically, given an input image $I \in \mathbb{R}^{H \times W}$ (for simplicity, a single-channel grayscale image), and a kernel $K \in \mathbb{R}^{m \times n}$, the convolution operation produces an output feature map $O \in \mathbb{R}^{H' \times W'}$ defined as:

$$O(i,j) = \sum_{u=0}^{m-1} \sum_{v=0}^{n-1} K(u,v) \cdot I(i+u, j+v), \tag{3}$$

where

- (i, j) indexes a location in the output,
- (u, v) indexes positions in the kernel,
- the kernel “slides” over the input image to compute a weighted sum at each location.

An illustration of this process is shown in Figure 3, where a 2 × 2 kernel is convolved with a 4 × 4 input. Each step corresponds to a local weighted sum, which forms one entry of the output feature map.

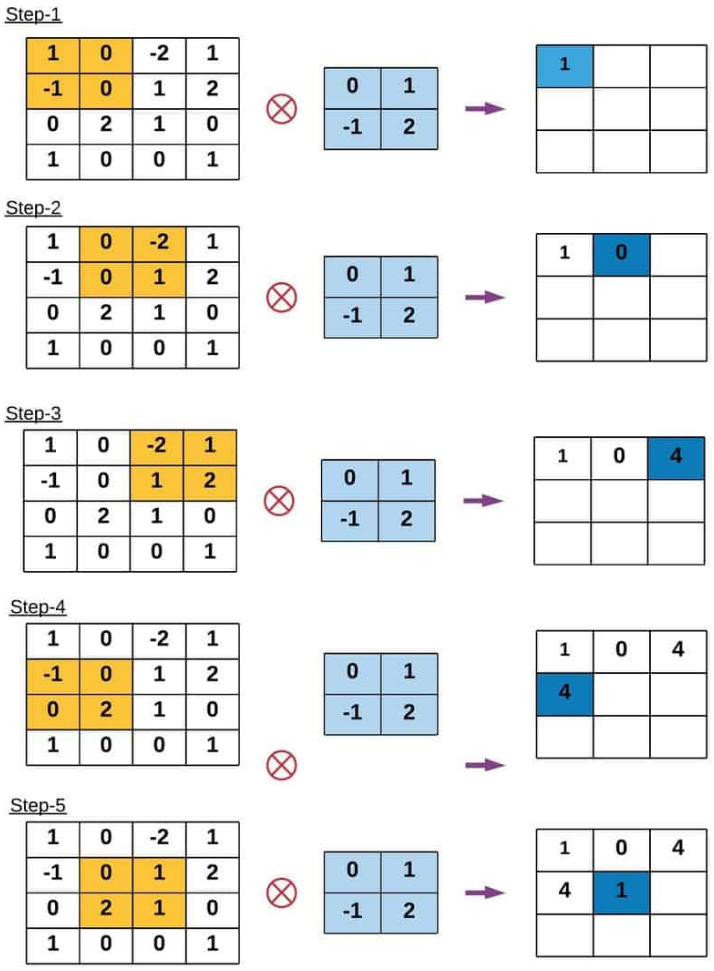


Figure 3: Example of the convolution operation, from [18]. A 2 × 2 kernel (centre) is applied to a 4 × 4 input matrix (left), producing a 3 × 3 output feature map (right). At each step, the kernel is multiplied element-wise with the corresponding region of the input, and the results are summed to yield a single value in the output. This process illustrates how convolution extracts local features by sliding the kernel across the input.

Although this operation is technically cross-correlation (since the kernel is not flipped), the deep learning literature conventionally refers to it as convolution [12].

The convolution can be interpreted as a mechanism for detecting patterns in the input. Each kernel specialises in responding strongly to certain structures, such as edges, corners, or textures, depending on how its weights are learned during training [41]. By applying multiple kernels in parallel, a convolutional layer produces a collection of feature maps, each emphasising a different aspect of the image.

Importantly, this operation leverages three advantageous properties:

- Parameter efficiency: Kernels are reused across the entire input, significantly reducing the number of parameters compared to fully connected layers.
- Translation equivariance: A feature detected in one region of the image will produce a similar response if it appears elsewhere, which is essential for recognising objects regardless of their position.
- Locality: By focusing on small neighbourhoods, convolutions capture spatially local dependencies that are fundamental to natural images.

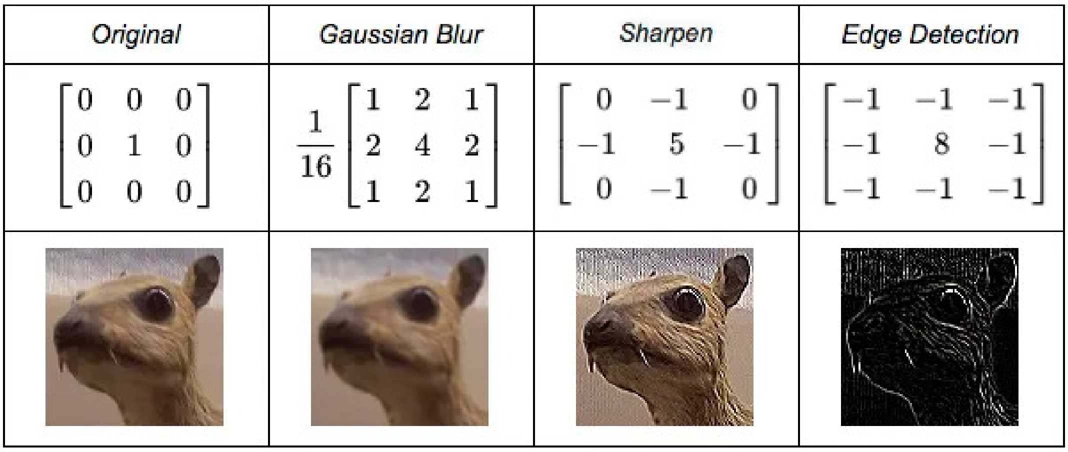
These properties explain why convolutional architectures have become a dominant paradigm for computer vision tasks, and in particular image classification [40].

0.3.2. Convolutional Kernels

A central component of CNNs is the convolutional kernel (or filter), a small matrix of learnable parameters applied to local regions of the input. Through the convolution operation, each kernel detects a particular pattern, such as edges, corners, or textures, based on how its weights evolve during training.

Formally, a kernel $K \in \mathbb{R}^{m \times n}$ slides across the input image $I \in \mathbb{R}^{H \times W}$, computing a dot product with each local patch (see Equation 3). Because the same weights are reused at every spatial location, kernels provide *translation equivariance*: the detector responds similarly to the same feature wherever it appears in the image [20]. A single convolutional layer typically includes many kernels running in parallel, each producing its own feature map. This collection of maps yields a rich, spatially structured representation of the input, from simple oriented edges in early layers to more complex motifs in deeper layers [40]. Figure 4 illustrates how different kernels emphasise different structures even when applied to the same image.

Kernel size is a key architectural choice. Modern CNNs favour small kernels (e.g., 3×3), since stacking several such layers increases the effective receptive field while keeping the parameter count low [58].



Original	*Gaussian Blur*	*Sharpen*	*Edge Detection*
$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$

Figure 4: Examples of convolutional kernels and their effects on an image, taken from [32]. Each kernel (top row) encodes a different operation, such as Gaussian blurring, sharpening, or edge detection. When applied to the same input image (bottom row), the resulting outputs highlight different aspects of the image, illustrating how kernels act as feature detectors.

0.3.3. The Receptive Field

In CNNs, the *receptive field* of a unit is the region of the input image that influences its activation [20]. In the first convolutional layer, this region corresponds directly to the kernel size (e.g., a 3×3 kernel observes a 3×3 patch). As layers are stacked, receptive fields expand: each deeper unit aggregates information from larger portions of the original image. This hierarchical growth is illustrated in Figure 5, where an activation in a later layer corresponds to a progressively larger input area.

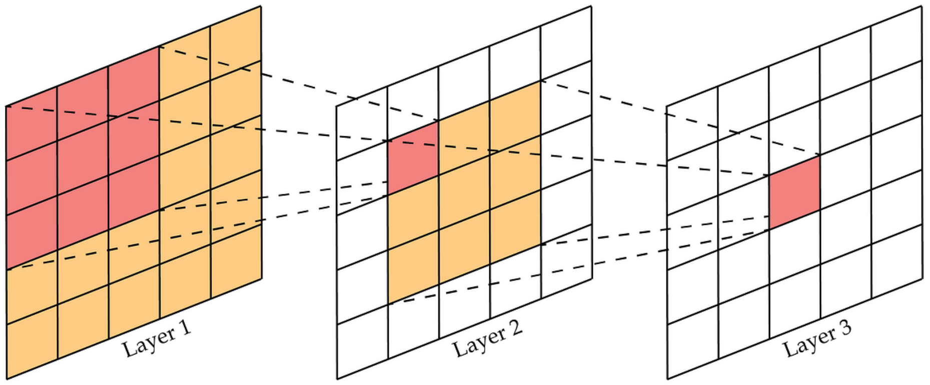


Figure 5: Receptive field expansion across layers. A deeper-layer activation (red) depends on a larger region of the input (red + orange). Example taken from [34].

Formally, if R_l denotes the receptive field at layer l with kernel size k_l and stride s_{l-1} in the previous layer, then:

$$R_l = R_{l-1} + (k_l - 1)\, s_{l-1}, \tag{4}$$

with $R_0 = 1$ for an individual input pixel [12]. Larger strides increase the rate at which receptive fields grow, as shown in Figure 6, where skipping input positions yields fewer outputs but a broader area of influence per activation.

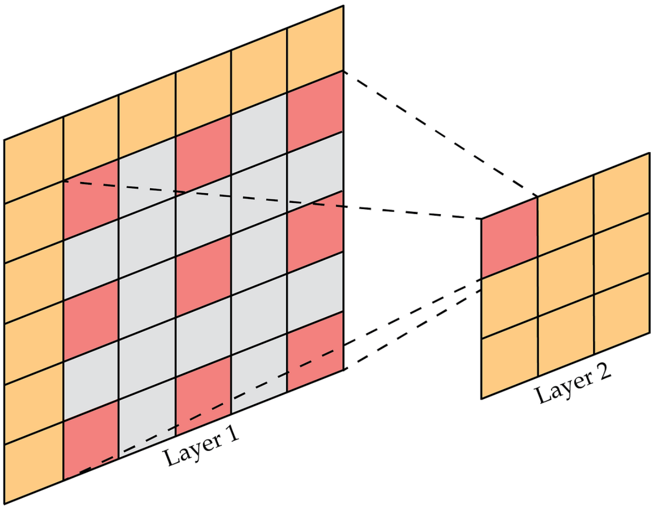


Figure 6: Effect of stride on receptive fields. A stride of 2 skips input positions, enlarging the receptive field of each output unit. Example taken from [34].

Although convolutional kernels are typically small, stacking multiple layers allows CNNs to learn both local features (edges, textures) and global structures (object shapes). In this thesis, this property is especially relevant because the pseudo-depth channel provides additional structural cues that become increasingly influential at deeper receptive fields, where the network integrates global shape information rather than superficial colour or texture.

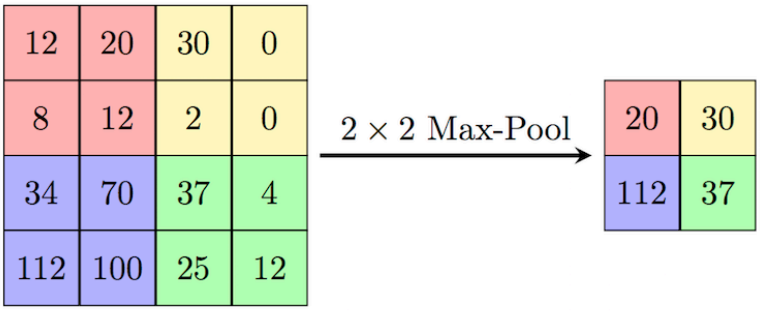
0.3.4. Pooling Layers

Pooling layers reduce the spatial resolution of feature maps while preserving the most informative activations. This makes representations more compact and computationally efficient, and introduces a degree of invariance to small spatial shifts [20].

Formally, for an input feature map $F \in \mathbb{R}^{H \times W}$, a pooling window of size $p \times p$, and stride s, the pooled output P is defined as

$$P(i,j) = \text{pool}(F(u,v) : u \in [is,\, is+p-1],\ v \in [js,\, js+p-1])\,, \tag{5}$$

where pool$(\cdot)$ is typically a max or average operator. Max pooling is the most common, highlighting the strongest local responses (see Figure 7), while global pooling collapses each channel to a single value and is often used instead of fully connected layers [44].



12	20	30	0
8	12	2	0
34	70	37	4
112	100	25	12

20	30
112	37

Figure 7: Example of 2×2 max pooling from [8]. For each non-overlapping 2×2 region in the input feature map, the maximum value is selected and propagated to the output. This operation reduces the spatial resolution of the feature map while preserving the strongest local activations, introducing a degree of translation invariance.

Although pooling supports efficient, translation-robust representations, it removes fine spatial detail and can hinder tasks requiring precise localisation. For this reason, modern CNNs sometimes replace pooling with strided convolutions [62]. In the models used throughout this thesis, pooling mainly serves to progressively condense RGB and pseudo-depth features before classification, allowing deeper layers to focus on more global object structure.

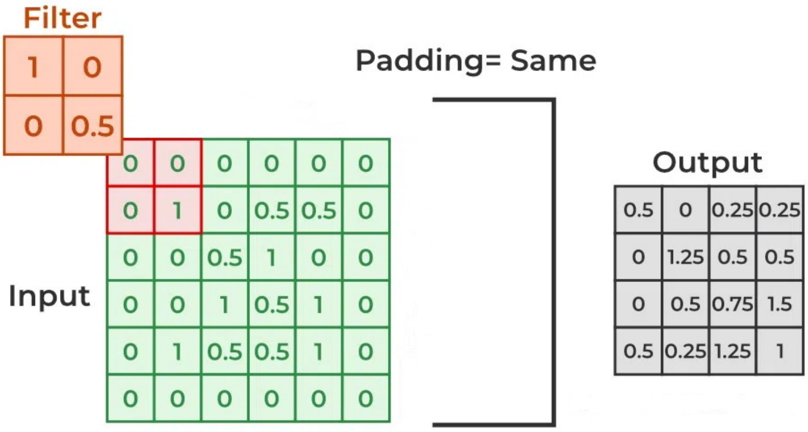
0.3.5. Padding

Convolutional kernels have limited coverage near image borders, which causes feature maps to shrink after each layer. Padding mitigates this by adding extra pixels, typically zeros, around the input so that boundary regions are processed in the same way as interior ones [20].

For an input of size $H \times W$, kernel size k, stride s, and padding p, the output resolution is

$$H' = \frac{H-k+2p}{s} + 1, \qquad W' = \frac{W-k+2p}{s} + 1. \tag{6}$$

Two common modes are *valid* padding ($p = 0$), where the output shrinks, and *same* padding, where p is chosen to preserve the input dimensions (see Figure 8).



1	0
0	0.5

0	0	0	0	0	0
0	1	0	0.5	0.5	0
0	0	0.5	1	0	0
0	0	1	0.5	1	0
0	1	0.5	0.5	1	0
0	0	0	0	0	0

0.5	0	0.25	0.25
0	1.25	0.5	0.5
0	0.5	0.75	1.5
0.5	0.25	1.25	1

Figure 8: Example of ”same” padding, adapted from [6]. Zeros are added around the input so that the convolution produces an output of the same spatial size.

Padding also affects how receptive fields grow. By preventing rapid spatial downsampling, it enables deeper CNN layers to integrate broader context. In the models used in this thesis, padding ensures that both RGB and pseudo-depth features maintain consistent spatial alignment throughout the network.

0.3.6. Activation Functions

Activation functions introduce the non-linearity that allows deep networks to learn complex mappings. Without them, a stack of convolutions or fully connected layers would reduce to a single linear transformation, severely limiting the model's capacity [20]. In CNNs, the activation is applied element-wise to the output of each layer.

The models used in this thesis rely exclusively on the Rectified Linear Unit (ReLU), defined as

$$\text{ReLU}(z) = \max(0, z). \tag{7}$$

ReLU is computationally simple, avoids vanishing gradients for positive inputs, and has become the standard choice for modern CNNs [50]. Its sparsifying effect (zeroing out negative activations) helps networks focus on salient features while keeping optimisation stable.

0.4. Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are a specialised class of DNNs designed to process data with a grid-like structure, most notably images [20, 41]. CNNs exploit the spatial structure of images by using convolutional operations, local connectivity, and shared weights to efficiently learn hierarchical representations. They are a dominant paradigm in computer vision and have achieved state-of-the-art results in tasks such as image classification, object detection, and segmentation [40].

0.4.1. Architecture Overview

A typical CNN consists of a sequence of layers that progressively transform raw image pixels into high-level representations suitable for classification or other tasks. The key building blocks, already introduced in the previous section, play the following roles:

- **Convolutional layers** apply *kernels* that detect local patterns in the input. Early kernels may respond to edges and simple textures, while deeper ones capture increasingly abstract features such as object parts or shapes.
- **Receptive fields** grow deeper into the network, allowing units to aggregate information from larger portions of the input. This hierarchical expansion enables CNNs to detect global structures while still leveraging local detail.
- **Pooling layers** reduce the spatial resolution of feature maps, making representations more compact and less sensitive to small translations or distortions.
- **Padding** ensures that convolutional kernels can be applied near image borders without shrinking the spatial size of feature maps.
- **Activation functions** introduce nonlinearity, enabling the network to learn complex mappings that cannot be represented by linear transformations alone.

Stacking these components results in a layered hierarchy: convolutional and pooling layers alternate to gradually abstract information, and activation functions are applied at each stage. Deeper in the architecture, feature maps become lower in resolution but richer in semantic content. The overall process is illustrated in Figure 9, where convolution, activation, pooling, and fully connected layers are combined into a complete architecture.

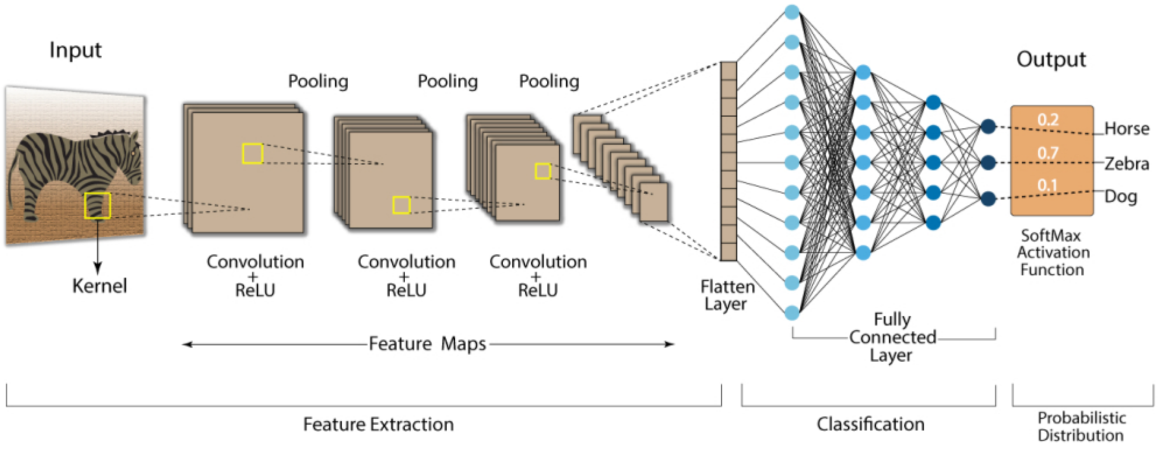


Figure 9: High-level architecture of a Convolutional Neural Network (CNN), from [59]. The input image is processed by a sequence of convolutional layers with nonlinear activation functions (e.g., ReLU) and pooling layers, which extract increasingly abstract feature maps. These are then flattened and passed through fully connected layers, which generate class scores that are converted into a probability distribution by a SoftMax function. This illustrates the end-to-end mapping from raw pixels to semantic class predictions.

After several stages of convolution, pooling, and activation, the high-level features are typically fed into fully connected layers (or global pooling layers) to produce the final output, for example, a probability distribution over object categories in an image classification task.

This design allows CNNs to automatically learn what features to detect and how to combine them, eliminating the need for manual feature engineering that characterised earlier computer vision approaches. Crucially, weight sharing across the image (via kernels) and hierarchical representation learning make CNNs both computationally efficient and highly expressive [20].

In the context of this thesis, CNNs serve as the backbone for the image classification models evaluated in the RGB and RGB-D settings. The same convolutional principles, local kernel operations, deep receptive fields, and hierarchical feature abstraction underpin both streams. This shared structure allows us to investigate how augmenting RGB features with a depth channel influences the learned representation and whether it improves out-of-distribution generalisation.

0.4.2. Convolutional Neural Network Architectures Used in This Thesis

This thesis evaluates several widely used CNN architectures that represent different design principles in modern deep learning. Rather than focusing on their computational efficiency, we use these models as diverse representatives of contemporary CNN design, allowing us to examine whether incorporating pseudo-depth influences classification performance across different architectural families.

ResNet-18 [24] is a canonical example of residual network design. Its skip connections enable stable gradient propagation through identity mappings, making it a robust and versatile backbone for image classification. Although smaller than deeper ResNet variants, ResNet-18 maintains strong representational capacity and serves as a reliable baseline architecture in many vision benchmarks.

EfficientNet [63] introduces a compound scaling strategy that systematically balances network depth, width, and input resolution. Instead of relying on ad-hoc scaling, EfficientNet uses a principled method to adjust model capacity in a coordinated manner. This results in a family of architectures that capture a broad range of representational characteristics.

MobileNet [29, 56] is built around depthwise separable convolutions, which factorise standard convolution into depthwise and pointwise components. This structural decomposition changes how features are extracted without altering the core convolutional principles. MobileNetV2 further introduces inverted residual blocks with linear bottlenecks, offering a distinct architectural style compared to traditional residual networks.

ShuffleNet [48, 70] employs group convolutions and channel shuffling to reorganise how information flows between feature channels. Its design explores alternative ways of distributing computation and feature mixing, making it architecturally complementary to both standard residual networks and MobileNet-style depthwise convolutions. ShuffleNet V2 further refines this design by addressing practical efficiency constraints such as memory access cost and channel imbalance, resulting in more stable and efficient feature propagation while preserving the core channel-shuffling principle.

Together, these architectures, ResNet-18, EfficientNet, MobileNet, and ShuffleNet, allow us to investigate our research question across different CNN design patterns and inductive biases. Because pseudo-depth is introduced via early fusion at the input level, its influence is exposed to the earliest feature extraction mechanisms of each architecture. By evaluating the effect of adding a pseudo-depth channel on models with distinct structural biases, we obtain a more robust understanding of whether depth cues consistently improve out-of-distribution generalisation.

0.5. Training Process

Once a network architecture has been defined, the next step is to train its parameters so that the model can perform the target task, such as image classification. Training iteratively adjusts the network's weights and biases so that its predictions match the ground-truth labels in the dataset [20].

The process consists of three main steps:

- **Forward pass**: Input images are propagated through the network, and each layer applies its operations (convolution, activation, pooling, etc.) to produce a prediction.
- **Loss computation**: A loss function measures the discrepancy between the prediction and the true label for the current batch.
- **Backward pass and update**: Using backpropagation, gradients of the loss with respect to each parameter are computed, and an optimisation algorithm (e.g., SGD) updates the weights to reduce future error.

This cycle repeats over many iterations grouped into *epochs*, where one epoch corresponds to a full pass through the training set. Over time, the parameters converge toward values that minimise the loss and improve performance on unseen data. A simplified overview of this loop is shown in Figure 10.

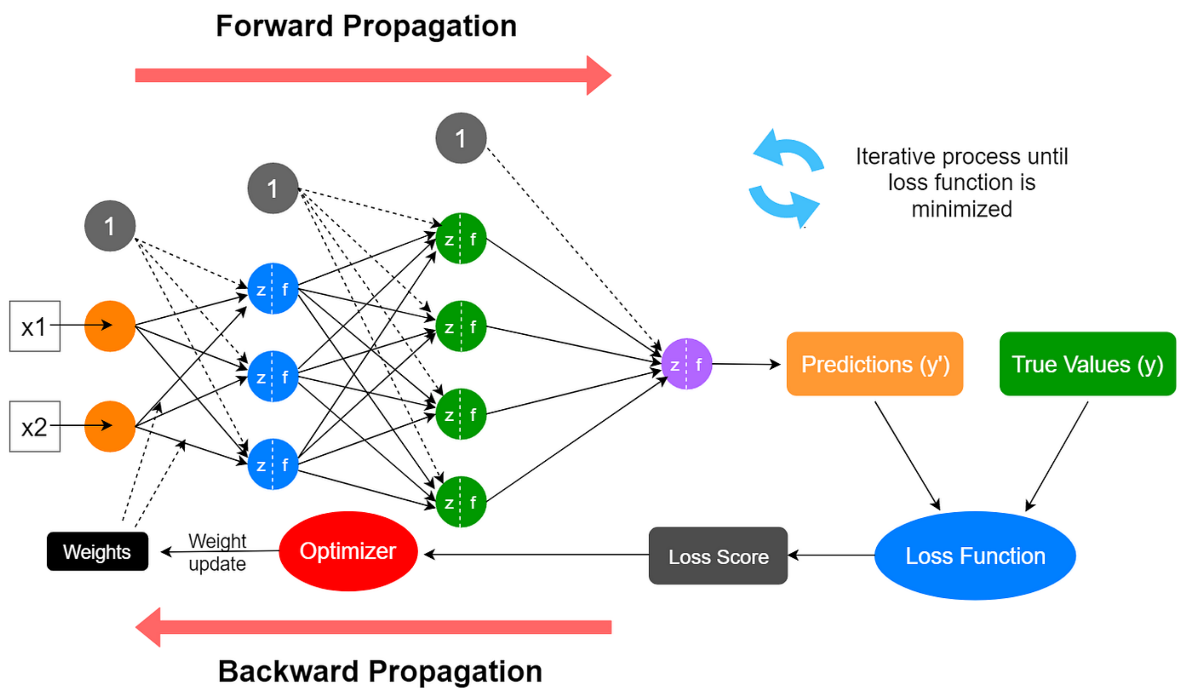


Figure 10: Overview of the training process, exemplified from [52]. During the forward pass, an input image x is processed by the network to produce a prediction y'. The loss function $L(y', y)$ compares the prediction to the ground-truth label y. During the backward pass, gradients of the loss with respect to the model parameters are computed via backpropagation and used by the optimiser to update the network weights. This forward–backward cycle is repeated iteratively until the training loss converges.

0.5.1. Loss Functions

During training, a neural network learns by minimising a loss function that quantifies the discrepancy between its predictions and the true labels [20]. For a dataset $\mathcal{D} = \{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})\}_{i=1}^{N}$ and model predictions $\hat{\mathbf{y}}^{(i)} = f(\mathbf{x}^{(i)}; \theta)$, the training objective is to minimise the average loss

$$\mathcal{L}(\theta) = \frac{1}{N} \sum_{i=1}^{N} \ell(\hat{\mathbf{y}}^{(i)}, \mathbf{y}^{(i)}). \tag{8}$$

In this thesis, we exclusively use the *cross-entropy loss*, the standard choice for classification. Given a predicted probability distribution $\hat{\mathbf{y}}$ and a one-hot encoded target $\mathbf{y}$, the per-sample loss is

$$\ell(\hat{\mathbf{y}}, \mathbf{y}) = -\sum_{j=1}^{K} y_j \log \hat{y}_j. \tag{9}$$

Cross-entropy encourages the model to assign high probability to the correct class and penalises confident but incorrect predictions.

The gradient of this loss provides the learning signal used during backpropagation. Because all experiments in this thesis (both RGB and RGB-D) are cast as multi-class image classification, cross-entropy is the appropriate and sufficient objective throughout.

0.5.2. Backpropagation

After the loss has been computed, the network's parameters must be updated in a direction that reduces this loss. This requires computing the gradient of the loss with respect to every weight in the model. Backpropagation provides an efficient way to do this by applying the chain rule through the layered structure of the network [55].

Given a network with a loss function $\mathcal{L}$ and parameters $\mathbf{W}^{(l)}$, backpropagation computes the gradient

$$\frac{\partial \mathcal{L}}{\partial \mathbf{W}^{(l)}} \tag{10}$$

by propagating error signals from the output layer backwards. Each layer reuses intermediate quantities such as activations and derivatives of the activation functions, avoiding redundant computation and making gradient evaluation tractable even for large CNNs.

In modern deep learning frameworks, backpropagation is implemented automatically and executed efficiently on GPUs. It is the mechanism that connects the loss function to the optimisation algorithm, enabling the model to update its parameters and learn from data. All training procedures in this thesis, both RGB and RGB-D, rely on this standard gradient-based update process.

0.5.3. Optimisation Algorithms

Training a neural network involves updating its parameters so as to minimise the loss,

$$\theta^{*} = \arg\min_{\theta} \mathcal{L}(\theta). \tag{11}$$

Figure 11 illustrates this general idea. Parameters are iteratively adjusted in directions that reduce the loss.

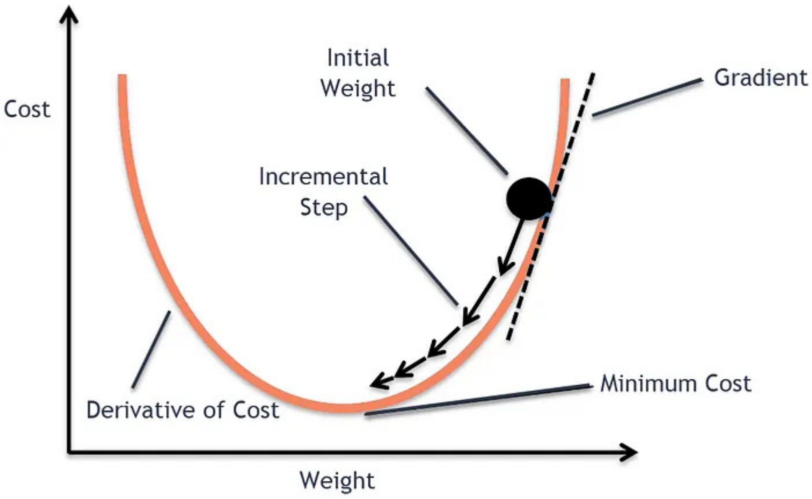


Figure 11: Conceptual illustration of parameter updates during optimisation, reproduced from [22]. The horizontal axis represents model parameters (weights), while the vertical axis denotes the loss (cost) function. The gradient indicates the direction of steepest ascent, and optimisation proceeds by iteratively updating parameters in the opposite direction toward a local minimum.

In this thesis, all experiments use the *AdamW* optimiser, a modern adaptive optimisation method that has become standard in deep learning. AdamW [45] extends the Adam algorithm [35] by maintaining moving averages of gradients and squared gradients,

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1)\, \nabla_\theta \mathcal{L}_t, \tag{12}$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2)\, (\nabla_\theta \mathcal{L}_t)^2, \tag{13}$$

which are bias-corrected and used to compute an adaptive update:

$$\theta_{t+1} = \theta_t - \eta \frac{m_t}{\sqrt{v_t} + \epsilon} - \eta\, \lambda\, \theta_t. \tag{14}$$

Here, θ_t denotes the model parameters at optimisation step t, and $\mathcal{L}_t$ is the training loss evaluated on the current mini-batch. The vectors m_t and v_t represent the exponentially decaying first- and second-order moment estimates of the gradients, respectively. The scalars β_1 and β_2 control the decay rates of these moving averages, η is the learning rate, ϵ is a small constant added for numerical stability, and λ is the weight decay coefficient controlling the strength of ℓ_2 regularisation applied to the parameters.

The important contribution of AdamW is the decoupled weight decay term $\lambda\theta_t$, which improves generalisation by regularising the parameters independently from the gradient update. Due to its stability and effectiveness across CNN architectures, AdamW is used for all RGB and RGB-D models evaluated in this thesis.

0.5.4. Regularisation Techniques

Neural networks contain large numbers of learnable parameters, which makes them highly flexible but also susceptible to *overfitting*, capturing noise or accidental regularities in the training data rather than learning generalisable patterns. Conversely, if a model lacks sufficient capacity or is trained inadequately, it may *underfit*, failing to capture the underlying structure in the data. These phenomena are illustrated in Figure 12.

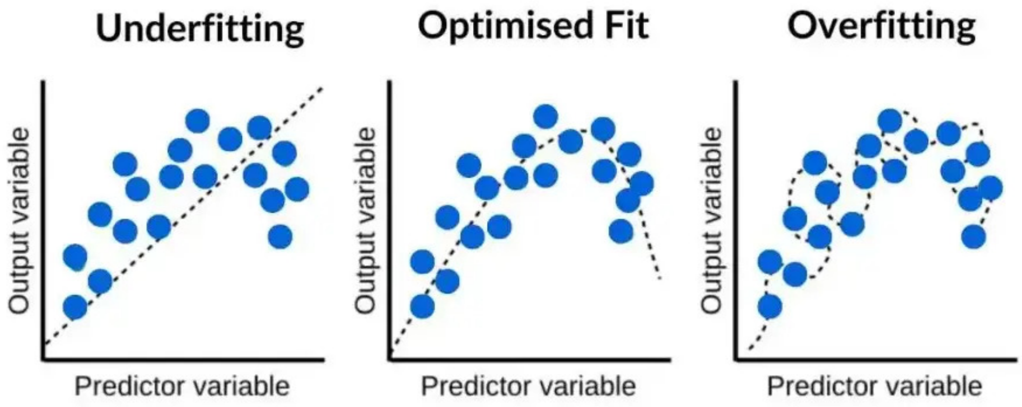


Figure 12: Illustration of underfitting, optimal fit, and overfitting, taken from [51]. The horizontal axis represents an input variable, while the vertical axis represents the predicted output. Underfitting corresponds to overly simple models, overfitting to overly complex models, and optimal fit balances bias and variance. Regularisation techniques help promote solutions that generalise well to unseen data.

Regularisation refers to strategies that improve generalisation by discouraging the model from relying too heavily on spurious correlations in the training set. In this thesis, the two forms of regularisation used throughout are *data augmentation* and *weight decay*.

Data augmentation applies label-preserving transformations to training images, such as random crops, flips, colour jitter, or geometric distortions, to expose the model to a wider range of appearance variations [57]. By effectively enlarging the dataset, augmentation reduces overfitting and encourages CNNs to learn representations that are robust to natural variability. This is particularly important in our experiments, where changes in background context or scene composition can create distribution shifts between training and testing.

Weight decay is a simple but effective form of regularisation used in all experiments through the AdamW optimiser. It penalises large parameter values by shrinking the weights toward zero at every update step. This discourages overly complex solutions and helps the model focus on more stable, generalisable patterns rather than memorising fine-grained noise.

Together, data augmentation and weight decay ensure that the RGB and RGB-D models in this thesis learn features that generalise beyond the specific instances encountered during training.

0.5.5. Hyperparameter Optimisation

Neural networks rely on hyperparameters, such as the learning rate, batch size, number of epochs, and weight decay, that shape the training process but are not learned from data. Selecting appropriate values can strongly influence convergence behaviour and final performance [20]. Techniques such as manual tuning, grid search [3], random search [3], or Bayesian optimisation [60] are commonly used to explore this configuration space.

In this thesis, extensive hyperparameter optimisation is deliberately not performed. Since our goal is to compare RGB models with their RGB-D counterparts under controlled and comparable conditions, it is essential that both versions of each architecture are trained with the *same* hyperparameters. This isolates the effect of adding a pseudo-depth channel without introducing differences caused by tuning. As such, only a standard and fixed training configuration is used throughout, and hyperparameter optimisation lies outside the scope of this work.

0.5.6. Putting It All Together

Training a neural network begins with parameter initialisation, which provides a reasonable starting point for optimisation. Modern frameworks use well-established initialisation schemes (such as Xavier [19], or He initialisation [25]) to maintain stable activation and gradient magnitudes across layers, enabling reliable convergence.

The components described in the previous subsections, loss computation, backpropagation, optimisation with AdamW, and regularisation through data augmentation and weight decay, together define the full training loop. In each iteration, the network performs a forward pass to generate predictions, the loss function evaluates their quality, gradients are computed through backpropagation, and the optimiser updates the parameters. This cycle is repeated over many mini-batches and across multiple epochs until the model reaches a stable solution.

Although modern deep learning libraries (e.g., *PyTorch*) automate these steps, understanding how they fit together is essential for designing controlled experiments. In this thesis, the same training loop and hyperparameters are applied consistently to both RGB and RGB-D versions of each model so that any performance differences can be attributed to the presence or absence of pseudo-depth information.

0.6. Datasets

The performance and generalisation ability of DNNs depend heavily on the quality and diversity of the datasets on which they are trained and evaluated. In this section, we describe the datasets used throughout this work, including both established benchmarks and custom variants derived from them. The datasets serve complementary roles: standard datasets, such as MNIST, provide controlled environments for methodological validation, while more complex or domain-specific datasets, such as NICO++, are employed to assess the models under realistic and context-rich conditions.

0.6.1. MNIST

The *Modified National Institute of Standards and Technology (MNIST)* [10] dataset is one of the most widely used benchmarks in machine learning and computer vision. It was developed as a standardised and simplified version of the original NIST dataset, providing a clean and accessible platform for evaluating image classification algorithms.

MNIST consists of 70,000 grayscale images of handwritten digits from 0 to 9, each of size 28×28 pixels. Of these, 60,000 images are designated for training and 10,000 for testing. Each image contains a single centred digit written by a different individual, with variations in handwriting style, thickness, and orientation. The pixel intensities are typically normalised to the range $[0, 1]$ or $[-1, 1]$ prior to training. Examples of images from the dataset are shown in Figure 13, illustrating the diversity of handwriting styles and digit shapes.

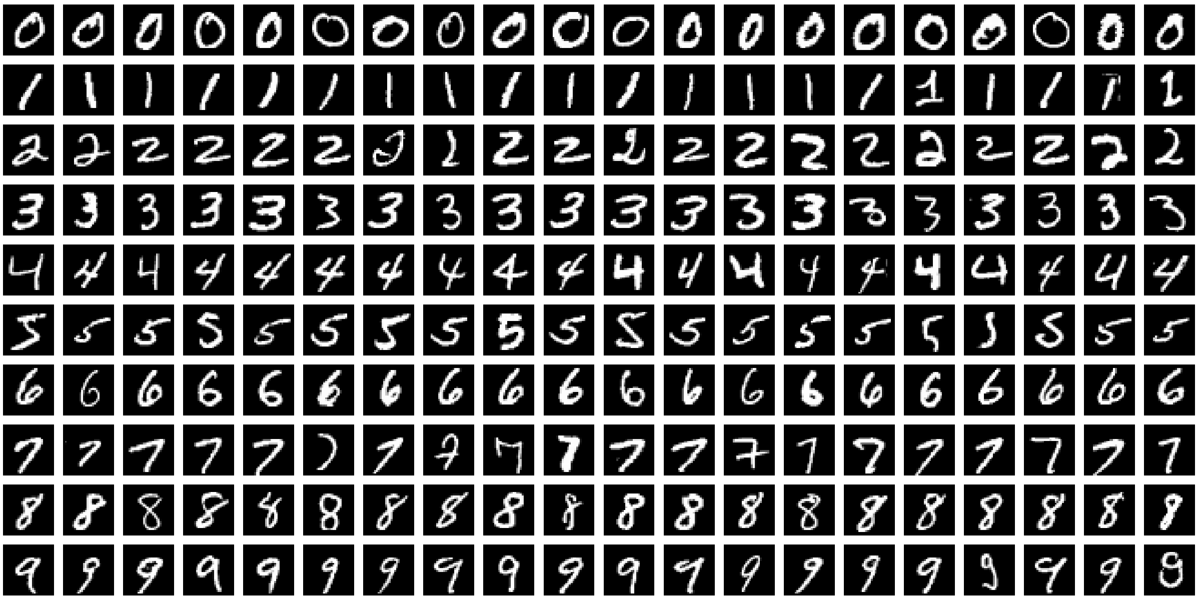


Figure 13: Example images from the MNIST dataset, taken from [49]. Each image is a 28×28 grayscale representation of a handwritten digit (0-9). The dataset includes a wide variety of handwriting styles, thicknesses, and orientations, making it a valuable benchmark for evaluating image classification models.

This dataset has served as a fundamental benchmark for early developments in neural networks, including CNNs, and remains a canonical example in the literature due to its simplicity and accessibility. Despite its relatively low complexity, MNIST captures key challenges of pattern recognition, such as intra-class variation and noise, making it a valuable testbed for validating new model architectures and training methods.

Over time, MNIST's role has evolved from being a competitive benchmark to serving as a baseline or sanity check for modern deep learning techniques. Models achieving near-perfect accuracy on MNIST are now expected, but its simplicity makes it an ideal starting point for prototyping and verifying experimental setups before scaling to more complex datasets.

0.6.2. Coloured-Background MNIST

To introduce controlled forms of contextual variation into the MNIST setting, we created a custom variant in which each handwritten digit is placed on a uniformly coloured background. This dataset preserves the original MNIST digit shapes but overlays them onto one of several background colours, enabling us to systematically study how models react to simple distribution shifts in visual context.

For the version used in our toy experiments, digits are rendered in white and placed on backgrounds of *red*, *green*, *blue*, or *black*. The colour serves as a contextual attribute that is not directly tied to the digit label but can nonetheless be exploited by a model as a shortcut if the training distribution is biased. For example, if digits are seen only on red and green backgrounds during training, a model may implicitly rely on these colours rather than fully learning the digit representation, which can lead to sharp degradation when tested on the unseen blue background.

Figure 14 illustrates representative samples from this dataset, showing how the same digits appear under different background colours.

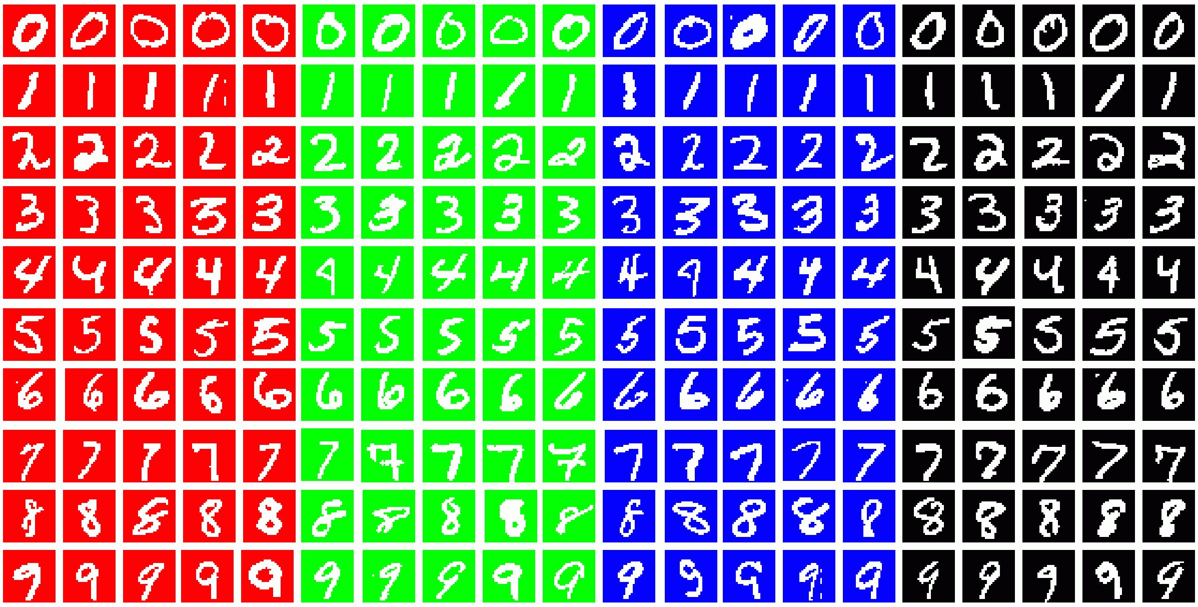


Figure 14: Example images from the custom coloured-background MNIST dataset. Each handwritten digit from MNIST is overlaid onto a uniformly coloured background (red, green, blue, or black). The digit itself remains unchanged, while the background provides a simple yet controlled contextual cue. This allows us to investigate how models respond to distribution shifts in background colour and whether depth information can help mitigate such shortcut dependencies.

0.6.3. Coloured-Background + Coloured-Digit MNIST

To further increase the complexity of the controlled MNIST setting and constrain the emergence of depth-only shortcuts, we introduce a second custom dataset in which both the digit and the background are colourised. In this variant, each MNIST digit is recoloured either *lime* or *orange*, and placed on one of four background colours: red, green, blue, or black. This yields eight distinct digit-background combinations (lime/orange × red/green/blue/black), each containing the same underlying MNIST digits, but with different colour attributes.

Unlike the simpler coloured-background dataset, here the class label depends not only on the digit identity but also on the digit colour. For example, a lime ”5” and an orange ”5” correspond to different classes. As a consequence, accurate classification requires the model to integrate information from the RGB channels (to detect the digit colour) and the depth channel (which encodes the digit’s shape). Depth alone is insufficient because it contains no information about the digit’s colour, while RGB alone can be misleading due to the presence of multiple background colours. This construction explicitly prevents solutions that rely solely on depth and provides a controlled setting for testing whether RGB-D models use their modalities in a complementary manner.

Figure 15 presents examples from this dataset, highlighting how digit identity and digit colour vary jointly across multiple background contexts.

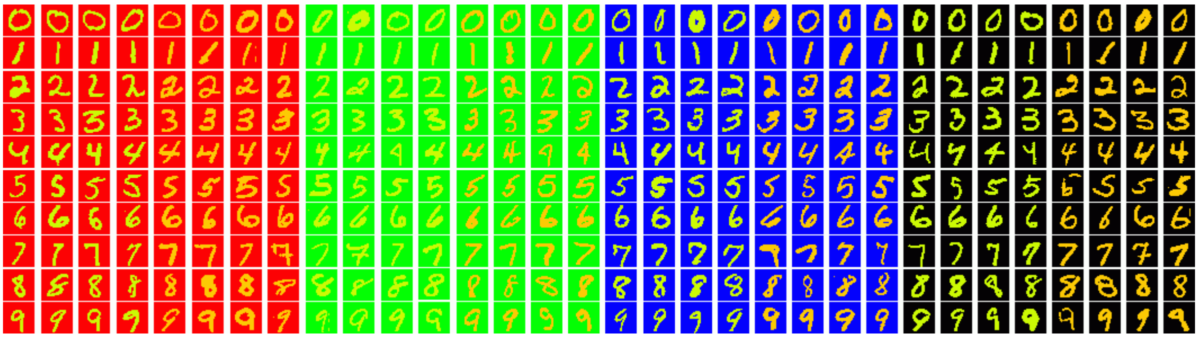


Figure 15: Example images from the Coloured-Background + Coloured-Digit MNIST dataset. Each handwritten digit is recoloured in either *lime* or *orange* and placed against backgrounds of different colours (red, green, blue, or black). This variant requires models to jointly infer both the digit identity and the digit's colour, preventing depth-only or background-only shortcut solutions and enabling controlled evaluation of RGB-depth complementarity under varying contextual conditions.

0.6.4. The NICO++ Benchmark Dataset

A significant challenge in developing robust computer vision models is the assumption that training and testing data are independently and identically distributed (I.I.D.). This assumption rarely holds true in real-world applications, where models encounter novel environments, lighting conditions, and object contexts not seen during training. This discrepancy, often termed a "distribution shift," can lead to a drastic drop in model performance. Models trained on biased datasets often learn to rely on spurious correlations, features that are correlated with the label in the training data but are not causally related to the object itself. For instance, a model might associate the presence of grass with the "cow" class, and subsequently fail to recognise a cow in a desert or indoor environment.

To address the limitations of traditional benchmarks and facilitate research in out-of-distribution (OOD) generalisation, specialised datasets have been developed. NICO++ (Non-I.I.D. Image Dataset with Contexts++) is one such benchmark, introduced by Zhang et al. [71]. This dataset is designed to provide a large-scale, fine-grained, and realistic testbed for evaluating the robustness and generalisation capabilities of models under non-I.I.D. conditions.

NICO++ is an extension of NICO [27], a dataset previously designed to address the same challenge, training and testing AI models in a way that forces them to learn the true features of an object, rather than just the context in which it appears. Both these datasets provide a controlled environment in which to systematically measure why a model might fail when it encounters a new domain.

The defining feature of NICO++ is its explicit annotation of "contexts". A context refers to the background, scene, or attribute that co-occurs with the main object. For example, an object from the class dog can be found in contexts such as on grass, in snow, or on a sofa. Each image in the dataset is labelled with both its primary object category and its associated context. Examples of images from the dataset are shown in Figure 16, illustrating how each object category is represented under diverse contextual conditions.

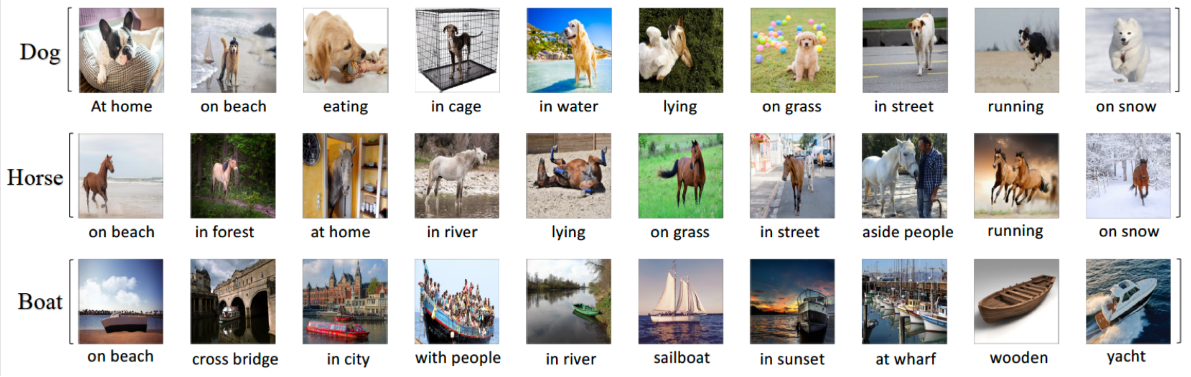


Figure 16: Example images from the NICO++ dataset. Each object category (e.g., dog, horse, boat) appears in multiple distinct contexts, such as ”on beach”, ”in city”, or ”on snow”. This design allows for the study of contextual bias and robustness. Figure taken from the original NICO paper [27].

The standard evaluation protocol for NICO++ involves splitting the data such that the contexts seen during training are disjoint from the contexts seen during testing for a given object category. This forces a model to learn the intrinsic features of the object itself, rather than relying on the shortcut features provided by the context. Our paper follows the same evaluation protocol.

For the experiments conducted in this paper, a curated subset of the full NICO++ dataset was employed. Since the primary goal of this research is to investigate the effect of pseudo-depth on the performance gap between in-distribution and out-of-distribution data for CNN models, and not to achieve state-of-the-art results on the full benchmark, a smaller, controlled environment is not only sufficient but also preferable for a clear analysis.

0.7. Out-of-Distribution Generalisation

Machine learning models, including DNNs, are typically trained and evaluated under the assumption that the training and test data are drawn from the same underlying distribution. In practice, however, this assumption rarely holds. When the data encountered during testing differs from the data seen during training, due to changes in context, environment, or the conditions in which the data was acquired, the model's performance often degrades significantly. This challenge is known as the out-of-distribution (OOD) generalisation problem [20, 53].

OOD generalisation refers to a model's ability to maintain reliable performance when exposed to data drawn from a different but related distribution than the training set. More formally, given a training distribution $P_{\text{train}}(X, Y)$ and a test distribution $P_{\text{test}}(X, Y)$, a model exhibits good OOD generalisation if it performs well even when $P_{\text{train}} \neq P_{\text{test}}$.

Models trained under the I.I.D. assumption often learn to take ”shortcuts” by exploiting spurious correlations in the training data. A spurious correlation is a statistical relationship between two variables that appears to be causal but is not. An example of this is a model trained to classify parrots. If the training dataset overwhelmingly contains images of parrots in the sky, the model may learn the simple rule that ”if there is a sky, it is a parrot”. This model might achieve very high accuracy on test data drawn from the same distribution (more parrots in the sky), but its performance would worsen when it encounters an OOD example, such as a parrot in a tree or in a cage indoors. The model hasn't truly learned to identify a ”parrot”. It has learned to identify a ”parrot-in-the-sky” scene. This failure to generalise beyond the training context is the core of the OOD problem.

There are several types of distribution shifts that can lead to OOD scenarios [36, 64]:

- **Covariate shift**: The input distribution $P(X)$ changes, while the conditional label distribution $P(Y \mid X)$ remains the same. An example of this is an image classifier trained on cats photographed indoors that will then perform poorly on outdoor cat images. This is the challenge that this paper aims to tackle.
- **Label shift**: The marginal distribution of labels $P(Y)$ changes, but the conditional distribution

$P(X \mid Y)$ does not. This can occur, for example, in a dataset with an altered frequency of certain classes compared to the training set.

- **Concept shift**: The underlying relationship between inputs and labels changes, i.e., $P(Y \mid X)$ differs. This happens when re-labelling criteria change, or object categories are redefined across domains.

In real-world visual recognition scenarios, covariate shift is pervasive and difficult to anticipate fully. Differences in lighting, weather, camera viewpoint, background clutter, or sensor characteristics can all induce substantial performance degradation [2]. CNN models exacerbate this vulnerability. Their limited capacity encourages reliance on the easiest discriminative signal present, often contextual or texture-based cues, rather than more robust shape- or geometry-based representations.

A wide range of strategies has been proposed to improve OOD robustness. Data augmentation and domain randomisation attempt to expose the model to sufficiently diverse conditions such that it cannot rely solely on superficial cues. Domain generalisation methods seek representations that are invariant across multiple training domains [66]. Causal and invariant learning approaches directly target features that are stable determinants of the label rather than those correlated with it [2]. While effective in certain settings, these strategies can be computationally costly or heavily dependent on large and diverse datasets.

Despite significant progress, achieving robust OOD generalisation remains an open research problem. It is especially relevant for applications in safety-critical domains such as autonomous driving, healthcare, and environmental monitoring, where real-world variability cannot be fully captured by training data.

A complementary line of work shows that incorporating additional modalities, especially depth, can improve robustness by encouraging models to rely on geometric structure rather than appearance alone. Depth varies predictably with scene geometry and object layout, and therefore provides cues that are less sensitive to background or style shifts. However, most existing RGB-D approaches assume access to ground-truth depth, which is rarely available in typical image classification tasks.

This gap motivates the central question of this thesis: *can pseudo-depth, estimated from a single RGB image, serve as a reliable auxiliary signal to improve OOD generalisation?* If depth provides a more stable cue than colour or texture, even an approximate estimate might help CNNs avoid shortcut learning and generalise more robustly under covariate shift. This hypothesis is tested across both controlled synthetic settings and real-world datasets (see section 0.6).

0.8. Shortcut Learning

Shortcut learning refers to the tendency of machine learning models to rely on superficial, easily exploitable patterns in the training data rather than learning the intended, semantically meaningful concepts [17]. Because deep networks are optimisation-driven systems, they naturally gravitate toward the simplest decision rule that minimises training loss. When such a rule correlates with the correct label in the training set, the model may appear to perform well under I.I.D. evaluation while failing once this correlation breaks. Shortcut learning is therefore one of the principal mechanisms underlying failures in OOD generalisation, as discussed in the previous section.

A shortcut typically arises when a dataset contains spurious correlations, statistical regularities that are predictive but not causally tied to the task. From the model's perspective, detecting these correlations is cheaper than learning the true concept. In the earlier ”parrot-in-the-sky” example, the intended, robust feature for identifying a parrot is its shape, plume patterns, and colours, and its avian anatomy. However, distinguishing the blue sky is far simpler than learning the rich and variable morphology of parrots. As a result, the model implicitly learns the rule “blue background → parrot,” which works for many training images but fails when the context changes (e.g., a parrot indoors or in a forest). This behaviour reveals that the model has not learned to recognise the object itself, but instead a contextual co-occurrence pattern.

Shortcut learning manifests in multiple forms. One prominent example is *texture bias*, where CNNs often rely on local texture cues instead of global shape structure. A network might identify an elephant from its wrinkled grey texture, but misclassify a smooth elephant statue or a toy with the wrong texture.

In this thesis, however, we focus primarily on *background-based shortcuts*, in which models entangle the identity of an object with the context in which it typically appears. This has been documented in several benchmarks, including ImageNet and NICO++, where objects such as cows, ships, or birds frequently appear in stereotypical environments (grass, water, and sky, respectively), encouraging models to learn contextual associations instead of object-specific features.

Background-based shortcuts are especially problematic for CNN architectures. These models often rely disproportionately on the most predictive and lowest-variance cues available, typically colour or background statistics, because these allow rapid training convergence. For example, a classifier might learn to associate the ”ship” class with water textures or the colour blue, or the ”dog” class with grassy environments. Such associations can be strong enough that the model incorrectly labels a beached ship or a dog in an indoor setting. In extreme cases, shortcuts can stem from minute dataset artifacts, such as watermarks or consistent sensor-specific noise, which become unintended but highly predictive signals.

The toy setup used in this work provides a distilled illustration of shortcut learning. When MNIST digits are composited onto coloured backgrounds, an RGB-only model quickly learns to use background colour as a proxy for the target label because the digit identity remains constant while the background varies predictably. The same issue appears in the real-world NICO++ dataset, where object categories are statistically tied to environments (e.g., ”cow” tending to appear on grass or ”car” on roads), making the background a dominant predictive feature unless counteracted.

Shortcut learning is thus a key reason why models that achieve high performance on standard benchmarks can be brittle and unreliable when deployed in unconstrained environments. This observation motivates approaches that encourage models to rely on more stable, causally meaningful cues. In this thesis we explore one such cue: depth. Because depth captures geometric structure largely independent of background appearance, incorporating even pseudo-depth estimates can discourage shortcut reliance and promote better OOD generalisation in CNNs.

0.9. RGB-D

0.9.1. Value of Depth

Depth provides a fundamentally different type of visual information than RGB, capturing the geometric layout of a scene rather than its appearance. While RGB encodes colour, texture, and illumination, depth maps represent the distance from the camera to each point in the scene, making explicit the spatial arrangement of surfaces and objects. This geometric modality gives models access to cues such as object shape, relative scale, orientation, and occlusion, signals that are often difficult or impossible to infer reliably from 2D appearance alone [21, 67].

This distinction is particularly valuable in scenarios where RGB information is heavily influenced by contextual or stylistic factors. Colour and texture can vary significantly with lighting, weather, camera sensors, or background changes, and CNNs are especially prone to exploiting these high-variance, superficial cues as shortcuts. Depth, by contrast, changes in structured and predictable ways. The geometry of an object is far less sensitive to background colour, clutter, or environmental noise. This makes depth a more causally relevant signal for object identity, and therefore a potentially powerful tool for improving robustness under distribution shift.

To illustrate this distinction, Figure 17 shows several examples from the NICO++ dataset alongside their corresponding monocular depth estimates. Although the RGB images vary substantially in background, lighting, and overall context, the estimated depth maps reveal stable geometric structure: the silhouette of the dog remains distinct regardless of the surrounding leaves, the cat retains its form under dramatic sunset illumination, and the motorcycle’s geometry is preserved despite heavy visual clutter and occlusion caused by the exhaust smoke. The same pattern holds for the horse and ship classes. These examples highlight how depth emphasises object shape and spatial layout while suppressing background variation, making it a particularly promising auxiliary modality for counteracting shortcut learning and improving OOD robustness.

So, in the context of OOD generalisation, depth helps mitigate the core failure mode discussed earlier, the entanglement of object identity with background context. A cow on grass and a cow on snow differ

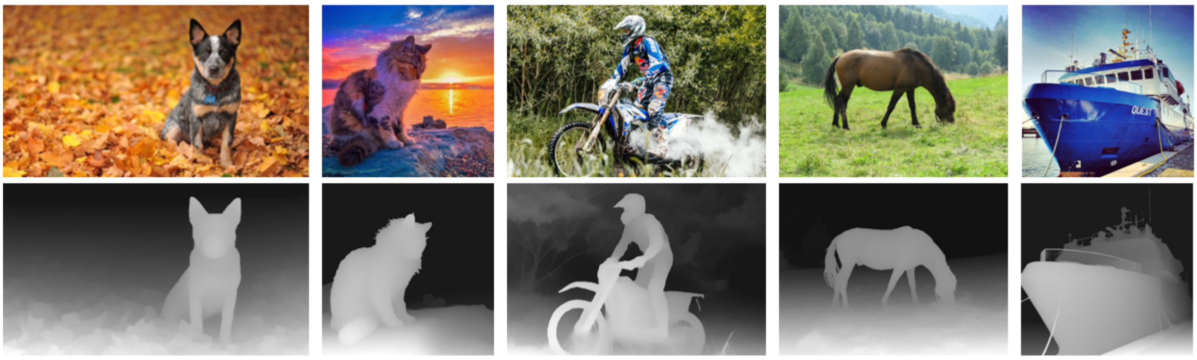


Figure 17: Examples from the NICO++ dataset (top row) and their corresponding monocular depth estimates generated using a state-of-the-art depth estimator (bottom row). Despite substantial variation in background, lighting, and environmental context across the RGB images, the predicted depth maps reveal consistent geometric structure for each object. This illustrates how depth highlights object shape and spatial layout while suppressing contextual variation, making it a valuable complementary modality for improving OOD robustness.

significantly in RGB space, but share a similar 3D shape. Likewise, a ship on water and a ship on land may look very different in colour or texture, yet maintain consistent geometric structure. By providing access to these shape-based cues, depth can encourage models to focus on object-centric features that generalise better across environments.

This value is clearly visible in controlled settings such as the toy MNIST experiments in this thesis. When digits are overlaid on coloured backgrounds, an RGB-only model often learns to rely on background colour as a shortcut. However, the depth maps (binary masks reflecting the foreground digit's silhouette) do not change with the background. This forces the RGB-D model to incorporate digit shape, rather than colour alone, leading to improved performance on unseen background colours.

In real-world datasets such as NICO++, the benefit of depth is more subtle but even more impactful. Here, distribution shifts arise from changes in background, weather, and context rather than from artificially controlled RGB cues. True depth is unavailable for NICO++ images, but depth estimation offers an appealing alternative. Even pseudo-depth tends to capture robust geometric structure, emphasising shape and object boundaries in a way that can complement or correct the biases of RGB features. Thus, integrating depth, whether ground-truth or estimated, provides a pathway toward learning more invariant, object-centric representations.

This thesis builds upon these insights by investigating whether pseudo-depth, produced by a monocular depth estimator, can meaningfully improve OOD robustness in CNNs. If geometric cues derived from estimated depth are sufficiently stable, they may help models reduce reliance on spurious RGB correlations and thereby narrow the seen–unseen generalisation gap. An important distinction is, however, that predicted depth is not treated as a replacement for RGB information nor as a precise geometric measurement, but as a structured auxiliary signal that may bias learned representations away from appearance-driven shortcuts and toward more stable spatial cues.

0.9.2. Complementarity of RGB and Depth

The combination of RGB and depth, commonly denoted as *RGB-D*, provides a substantially richer representation of visual scenes than either modality alone. RGB encodes appearance-based cues, colour, texture, shading, and illumination, while depth captures the underlying geometry of the scene, describing how far objects are from the camera and how surfaces are arranged in space. These two sources of information emphasise fundamentally different aspects of the visual world: RGB excels at describing *how* things look, while depth describes *where* things are. This makes the modalities highly complementary, a relationship that is particularly relevant to the problem of *shortcut learning* discussed earlier (see section 0.8). Because geometric cues change more predictably than RGB appearance under distribution shift, depth offers a more stable signal for mitigating shortcut reliance and improving OOD generalisation.

Depth proves especially useful in settings where RGB cues are ambiguous or unreliable. Variations in lighting, shadows, low contrast, or textureless regions can degrade appearance-based reasoning, yet they do not alter the physical 3D structure of the scene. Conversely, RGB often compensates for challenges faced by depth sensing, such as failures on reflective or transparent surfaces or loss of detail at long range. In practice, this complementarity allows RGB-D models to achieve superior performance across tasks such as object recognition, semantic segmentation, and scene understanding [5, 14, 61]. Depth discontinuities typically align with object boundaries, reinforcing shape-based cues that improve localisation and segmentation.

While the benefits of RGB-D fusion are well established, acquiring depth data through physical sensors introduces significant practical challenges. Structured-light or time-of-flight cameras are sensitive to environmental conditions such as illumination, material reflectivity, and operating distance, which can lead to noisy or incomplete depth maps. Sensor depth and RGB are often not perfectly aligned spatially or temporally, leading to fusion artefacts. Moreover, publicly available RGB-D datasets tend to be relatively small and narrow in domain compared to large-scale RGB datasets, limiting their diversity and the robustness of models trained on them.

These limitations have motivated a surge of interest in *monocular depth estimation*, where depth maps are predicted directly from RGB images [13, 54]. This approach bypasses the need for depth sensors and dramatically expands the applicability of geometric reasoning to datasets that contain only RGB imagery. For this thesis, this development is crucial. The NICO++ dataset contains no ground-truth depth, yet its OOD structure, objects appearing in systematically different contexts, makes it an excellent testbed for evaluating whether geometric cues can reduce shortcut learning. By supplementing RGB inputs with *pseudo-depth* estimated from a single image, we can construct an RGB-D representation without additional sensors and investigate whether even imperfect geometric information can guide CNNs toward more object-centric, context-invariant features.

0.9.3. Depth Estimation from RGB

Acquiring depth information typically requires specialised sensors such as structured-light or time-of-flight cameras. While these devices can provide accurate geometric measurements, they are expensive, power-intensive, and sensitive to environmental factors, including lighting conditions, material reflectivity, and measurement range. Furthermore, such sensors are not widely available to everyday users, in contrast to standard digital cameras. As a result, depth data are rarely available for large-scale or in-the-wild datasets. These limitations have driven significant interest in *monocular depth estimation*, the task of predicting dense, pixel-wise depth maps directly from a single RGB image.

Monocular depth estimation is inherently ill-posed. Multiple distinct 3D scenes can correspond to the same 2D projection. To overcome this ambiguity, learning-based approaches extract statistical regularities about geometry, such as edge discontinuities, shading patterns, texture gradients, and object shapes, from large image collections. Before the advent of deep learning, monocular depth estimation relied heavily on handcrafted cues derived from classical computer vision and human perception. Traditional approaches used shape-from-shading, shape-from-texture, defocus cues, and geometric priors to infer depth from a single image. These methods were grounded in assumptions such as Lambertian reflectance, uniform illumination, or known surface properties, which rarely held in complex real-world scenes. As a result, early monocular depth methods were brittle and often limited to controlled environments. They revealed the fundamental difficulty of the problem but also established the perceptual cues, shadows, contours, gradients, that later informed learning-based approaches.

Early deep learning approaches demonstrated that CNNs can learn hierarchical geometric features entirely from RGB input. Eigen *et al.* [13] pioneered this direction using a multi-scale architecture that jointly predicted coarse global layout and fine local detail and demonstrated that CNNs could learn depth cues directly from large annotated datasets such as NYU Depth and KITTI, without relying on hand-engineered constraints. This shift marked the beginning of fully data-driven depth prediction, where the network implicitly learned both local texture-depth relationships and global scene geometry. Subsequent methods improved accuracy by addressing key challenges such as scale inconsistency, boundary sharpness, and surface smoothness through encoder–decoder designs, residual connections, and geometry-aware losses incorporating surface normal regularisation [15, 39, 42]. These works showed that even without ground-truth 3D data, strong geometric priors can be inferred by learning from diverse annotated scenes.

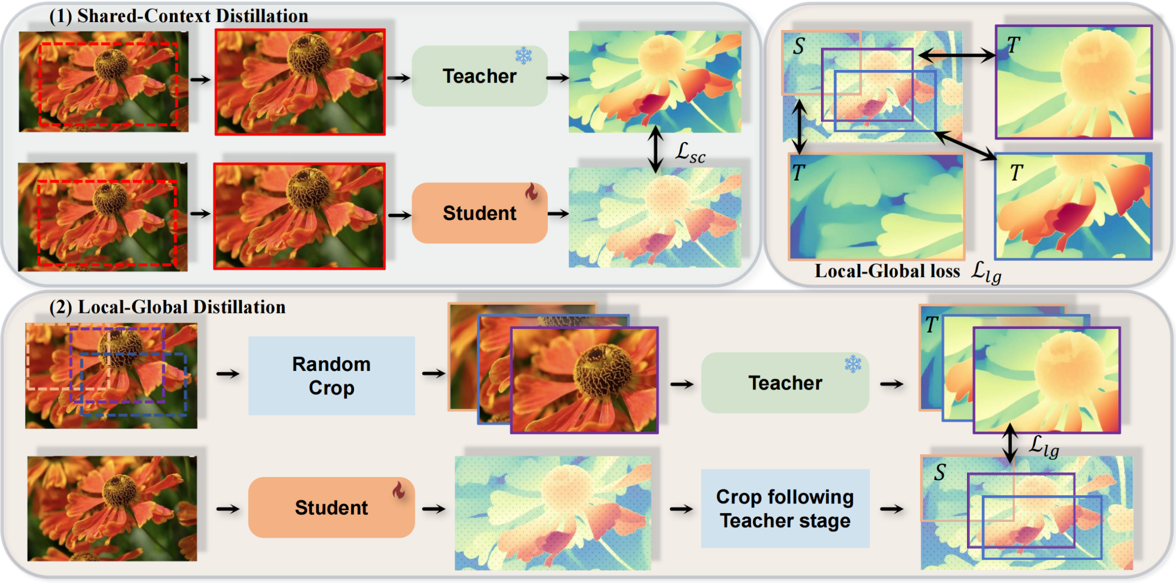

(1) Shared-Context Distillation
Teacher
Student
$\mathcal{L}_{sc}$
S
T
T
T
Local-Global loss $\mathcal{L}_{lg}$
(2) Local-Global Distillation
Random Crop
Teacher
T
$\mathcal{L}_{lg}$
Student
Crop following Teacher stage
S

Figure 18: Overview of the distillation strategy used in *Distill-Any-Depth* [26]. The framework employs a multi-teacher setup and transfers knowledge to a single student model using two complementary mechanisms: (1) *Shared-Context Distillation*, where teacher and student receive identically cropped regions to enforce consistent predictions under aligned context, and (2) *Local–Global Distillation*, where the teacher predicts depth for randomly cropped local patches while the student processes the full image, encouraging the student to reconcile fine-grained local structure with broader global scene layout. Together, these pathways allow the student to capture both local geometric detail and coherent global depth structure, resulting in highly generalisable pseudo-depth estimates for in-the-wild imagery. The teacher models provide supervisory depth predictions, while the student network learns to approximate these predictions under varying spatial context, enabling robust pseudo-depth estimation without ground-truth depth.

A parallel line of work began to explore *unsupervised* and *self-supervised* monocular depth estimation due to the scarcity of large labelled depth datasets. Techniques such as view synthesis, stereo supervision, and photometric reconstruction losses enabled models to learn depth without ground-truth labels, greatly expanding the scale and diversity of training data. This shift was historically important because it moved depth estimation toward broader generalisation across natural images and unconstrained environments.

The most recent generation of monocular depth models leverages large-scale pretraining and transformer-based backbones to capture long-range spatial dependencies that CNNs struggle to model. The Dense Prediction Transformer [54] demonstrated that transformers pretrained on massive vision datasets (e.g., ImageNet, JFT-300M) produce robust depth estimates by integrating contextual information across the entire image. Building upon this, modern frameworks such as *Distill-Any-Depth* [26] achieve state-of-the-art performance through a *multi-teacher distillation* approach. Instead of relying on a single supervisory signal, the method leverages a diverse ensemble of specialised depth ”expert” models, each trained on different datasets, sensing modalities, or scales, to generate heterogeneous depth predictions for the same RGB image. These predictions are then unified through a *cross-context distillation* strategy, in which a student model learns simultaneously from global (whole-image) depth predictions and local (patch-based) depth cues. This combination enables the student network to capture both fine-grained object geometry and coherent large-scale scene structure. Because the training supervision comes entirely from teacher models rather than explicit ground-truth depth, the resulting estimator generalises effectively across domains and demonstrates robustness to variations in lighting, style, context, and image composition. The overall distillation workflow of Distill-Any-Depth is illustrated in Figure 18, which shows how the model unifies teacher predictions using both shared-context and local-global training signals. Furthermore, the diversity and quality of the pseudo-depth produced by Distill-Any-Depth are illustrated in Figure 19, which shows consistent depth predictions across a wide range of indoor, outdoor, synthetic, and stylistically varied images.

The emergence of these large, general-purpose monocular depth estimators is particularly significant for research on robustness and OOD generalisation. Early depth predictors were too domain-specific to be useful in OOD contexts. Depth is largely invariant to superficial appearance changes, such as colour shifts, background variation, weather, or texture differences, that often cause RGB models to latch onto spurious correlations. In contrast, geometric structure changes far more slowly under distribution shift. This property makes pseudo-depth a compelling auxiliary signal for mitigating shortcut learning. Even though monocular depth estimates are not perfectly accurate, they preserve coarse object silhouette, relative scale, and spatial arrangement, exactly the types of cues that CNNs tend to ignore when relying solely on RGB appearance.

This is especially relevant for this thesis. The NICO++ dataset contains only RGB images but is explicitly designed to benchmark OOD generalisation, with object classes appearing across multiple, visually diverse contexts. Incorporating pseudo-depth allows us to construct RGB-D representations for NICO++ without requiring specialised hardware or dataset redesign. If the estimated depth maps capture object-centric geometry that remains stable across contexts, they may encourage the model to prioritise shape-based features over spurious RGB cues. Thus, monocular depth estimation not only mitigates the practical limitations of acquiring real depth data but also provides a principled and scalable way to introduce geometric signals into standard 2D datasets.

0.9.4. RGB-D Architectures and Fusion Strategies

Integrating depth information with RGB data has been a long-standing focus in computer vision, motivating the development of a wide range of neural architectures designed to exploit the complementary strengths of appearance and geometry. Because depth provides structural cues that are largely invariant to texture, lighting, or colour, while RGB encodes fine-grained appearance patterns that depth cannot capture, an important question in RGB-D learning concerns how these modalities should be fused within a model. Over the past decade, this question has led to several families of fusion strategies, each reflecting different assumptions about the nature of cross-modal interactions and different trade-offs between computational complexity, expressiveness, and noise robustness. The three canonical fusion paradigms, early fusion, late fusion, and intermediate or hybrid fusion, are illustrated in Figure 20, which provides a conceptual overview of how RGB and depth signals can be combined within a neural network.

Early approaches to RGB-D modelling typically adopted what is now known as *early fusion*. In this paradigm, the depth map is appended directly to the RGB channels to form a four-channel input, allowing a network to learn joint low-level features from the raw modalities [21]. The advantage of this approach lies in its simplicity. Depth is integrated at the very first layer of the network, enabling the model to discover correlations between texture and geometry automatically. Early fusion is computationally lightweight and requires minimal architectural modification, making it compatible with efficient CNN backbones. However, because all channels are processed together from the outset, early fusion can make the model sensitive to modality-specific noise. Structured-light and time-of-flight depth maps, for example, often contain artefacts or missing pixels, and estimated depth maps may introduce systematic errors that differ from those in RGB imagery. When modalities differ substantially in scale or noise statistics, early fusion risks allowing the richer or noisier modality to dominate the learned representation.

To address these limitations, later works explored *late fusion*, in which RGB and depth are processed in two independent streams before their feature hierarchies are merged at a higher level. Representative models such as FuseNet and others [4, 14, 21, 23] employ parallel CNN encoders, one dedicated to appearance and the other to depth, and combine their outputs via concatenation, element-wise addition, or learned gating. This architectural separation allows each stream to specialise in its modality. The RGB branch focuses on texture, colour, and shading, while the depth branch distils geometric structure, surface orientation, and discontinuities. Late fusion typically results in more interpretable representations and improved robustness to noise in either modality. The trade-off is increased computational cost, as dual-stream networks often require substantially more parameters and memory, making them less suitable for applications where efficiency is a priority.

As the field matured, researchers began to recognise that neither purely early nor purely late fusion could fully exploit the rich interplay between modalities. This motivated the development of *intermediate* or *hybrid fusion* strategies, in which RGB and depth features interact at multiple stages throughout the network. Cross-modal attention mechanisms [73], feature gating models, and multi-level skip connections [33] enabled deeper and more dynamic integration between appearance and geometry. These approaches proved especially effective for dense prediction tasks such as segmentation and saliency detection, where integrating geometric cues at different spatial scales can significantly enhance boundary localisation and region consistency. Hybrid fusion architectures reflect an understanding that RGB and depth can guide each other. Depth can refine structural information at early layers, while RGB can inform semantics at later layers [30].

More recent paradigms have pushed RGB-D fusion beyond static concatenation or additive combinations toward adaptive, context-aware mechanisms. Attention-based fusion models [65, 72] learn to modulate the contribution of each modality depending on spatial or semantic relevance. For example, the network may place greater emphasis on depth in cluttered regions or when RGB cues are degraded, and rely more heavily on RGB in texture-rich or high-detail areas. Other works introduce geometry-aware operators that use depth to reshape the network's internal computations [43, 47]. These models incorporate depth into convolutional receptive fields or normalisation schemes, allowing the network to become more sensitive to 3D structure. Transformer-based RGB-D architectures extend this idea even further by embedding modality-specific features into a unified token space, enabling long-range cross-modal interaction via self-attention [68, 69]. Such models demonstrate state-of-the-art performance in tasks requiring global coherence and detailed spatial reasoning.

Taken together, the history of RGB-D fusion shows a steady progression toward architectures that treat appearance and geometry not as independent signals, but as mutually informative modalities whose relationship varies across spatial scale, image content, and task demands. This evolution is relevant for this thesis because we adopt deliberately simpler fusion strategies in order to isolate the contribution of depth itself. In the toy experiments, we employ a late fusion design to keep the RGB and depth processing streams interpretable and to clearly observe the effect of adding shape-based cues. In the real-world NICO++ experiments, we use early fusion with pseudo-depth to minimise computational overhead, reduce architectural confounds, and evaluate whether even basic geometric information, estimated from a monocular depth model, can mitigate shortcut learning and improve robustness under distribution shift. While modern RGB-D systems increasingly rely on sophisticated cross-modal interactions, the simplified fusion strategies used in this thesis serve a different but complementary purpose, they highlight the intrinsic value of depth as a stabilising geometric signal, independent of architectural complexity.

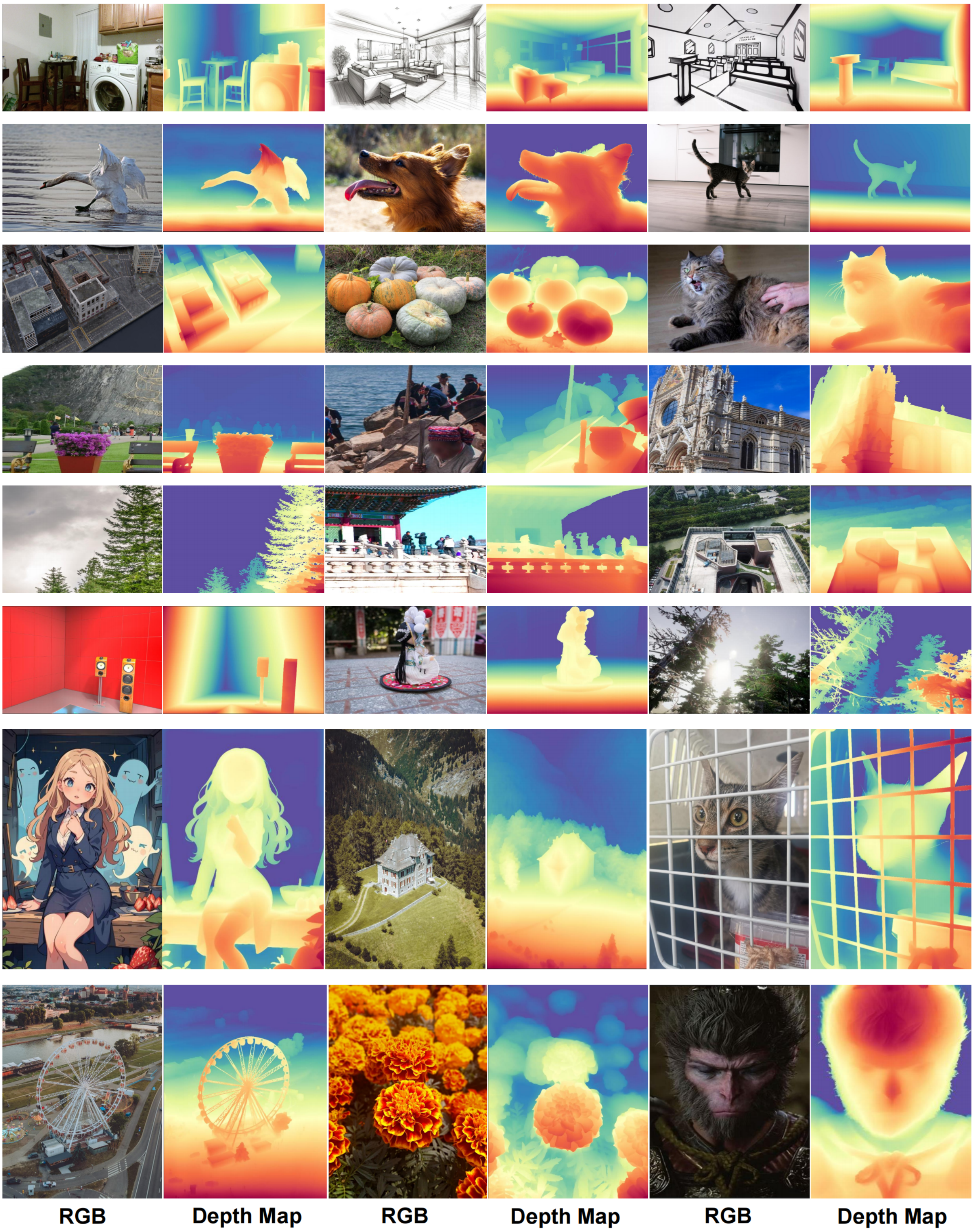


Figure 19: Example RGB images and corresponding pseudo-depth maps generated by *Distill-Any-Depth* [26]. The model produces coherent depth structure across a broad spectrum of content, including indoor environments, outdoor natural scenes, urban settings, stylised or cartoon-like images, and cases with strong viewpoint and illumination changes. Despite the variability in appearance, the predicted depth maps preserve object boundaries, relative distances, and overall scene layout, illustrating the robustness and versatility of modern monocular depth estimators.

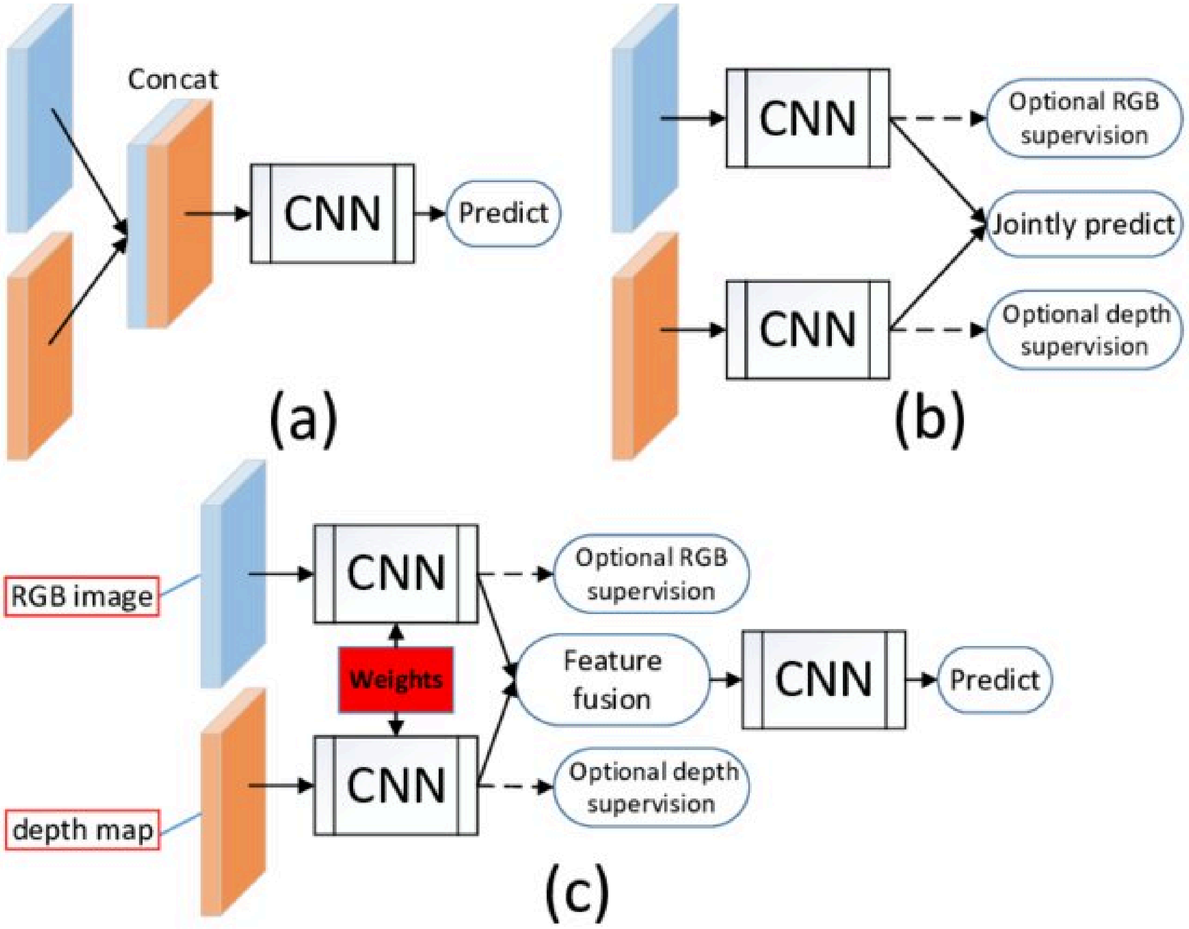


Figure 20: Illustration of common RGB-D fusion strategies, from [16]. *(a) Early fusion*: the depth map is concatenated with the RGB channels at the input, allowing a single CNN to learn joint low-level features. *(b) Late fusion*: RGB and depth streams are processed independently and merged only at higher layers, enabling modality-specific feature extraction before joint prediction. *(c) Intermediate or hybrid fusion*: features from the RGB and depth branches interact at multiple stages, often through shared weights, cross-modal connections, or attention-based mechanisms. These architectural patterns represent the major design choices in RGB-D perception and form the basis for many modern multi-modal models.

Scientific Article

Understanding the Value of Depth: RGB-D Fusion and Pseudo-Depth for Robust Out-of-Distribution Generalisation

Alexandra-Ioana Neagu
Delft University of Technology
Mekelweg 5, 2628 CD Delft

Abstract

Convolutional neural networks (CNNs) trained on RGB images (red, green, blue channels) often exhibit sharp performance degradation under distribution shifts, as they tend to rely on superficial appearance cues such as background or texture. While depth information is known to provide complementary geometric signals that can improve robustness, most existing approaches assume access to ground-truth depth or rely on complex RGB-D architectures, limiting their applicability in practice.

In this work, we investigate whether estimated depth, obtained from a monocular RGB image, can serve as a simple and effective auxiliary signal to improve out-of-distribution (OOD) generalisation in standard CNN classifiers. Using both controlled toy experiments and real-world evaluations on the NICO++ benchmark, we compare RGB-only models against RGB-D variants that incorporate a single predicted depth channel via minimal fusion. Our results show that pseudo-depth consistently reduces OOD performance gaps across multiple CNN backbones, without degrading in-distribution accuracy. We further demonstrate that these gains persist under moderate corruption of the depth signal and disappear when geometric structure is entirely removed, indicating that the improvements stem from meaningful geometric information rather than the mere presence of an additional input channel. Furthermore, we analyse these effects through class-resolved confusion matrices and qualitative input-level examples, showing that depth specifically attenuates structured semantic confusions under domain shift.

Taken together, our findings suggest that even imperfect, predicted depth can act as a lightweight geometric inductive bias, helping CNN classifiers move away from brittle appearance-based shortcuts and toward more robust representations under domain shift.

1. Introduction

In real-world visual recognition tasks, especially safety-critical ones, models must perform reliably not only on training-like data but also when exposed to unseen environments. Yet, convolutional neural networks (CNNs), including modern compact architectures commonly used in practice, often exhibit sharp drops in performance when evaluated on data drawn from different domains or distributions. This lack of robustness to out-of-distribution (OOD) shifts, arising from factors such as background bias, illumination, or viewpoint, remains a central challenge for domain generalisation in computer vision [16, 30]. While large-scale architectures, such as the newly conceptualised transformers, and training regimes can mitigate this issue through scale and diversity, standard CNN models trained under limited data or supervision are still prone to relying on shortcut correlations rather than causal cues [14]. Such models remain widely used due to their simplicity, efficiency, and ease of deployment across a range of application settings.

A promising line of work has shown that depth information can provide a complementary geometric signal to RGB cues (red, green, blue channels), improving model robustness and invariance to superficial domain features [9, 15, 23, 33]. However, such methods typically rely on ground-truth depth, available only for specific modalities (e.g., RGB-D sensors) or limited benchmarks like NYU Depth V2 [28] or KITTI [12]. In the vast majority of real-world applications, images are taken by regular, consumer-grade cameras that lack such specialised sensors, so more often than not, no ground-truth depth exists. An open question, therefore, is whether monocularly estimated depth, when used as a minimal auxiliary signal, can meaningfully improve out-of-distribution generalisation under contextual domain shifts in standard RGB classification pipelines.

This work investigates that question using standard CNN classification architectures trained under controlled conditions. Specifically, we ask:

- **RQ1**: What is the baseline in-distribution vs. OOD per-

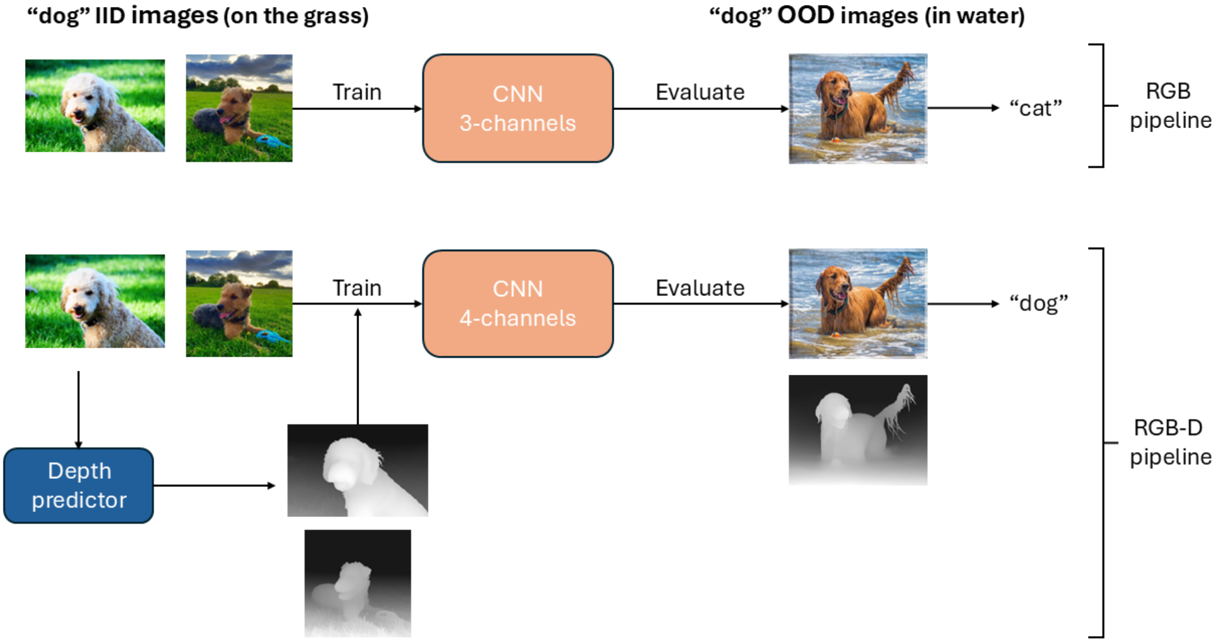


Figure 1. **Comparison between the RGB and RGB-D training pipelines used in this work.** The upper pipeline shows a CNN trained on RGB images (3 channels) of "dogs on grass", which fails to generalise when evaluated on out-of-distribution (OOD) images ("dogs in water"). The lower pipeline shows the RGB-D variant of the same CNN (4 channels), which receives both RGB and estimated depth inputs, yielding improved OOD performance.

formance gap for CNN classifiers trained solely on RGB data?

- **RQ2**: Can augmenting RGB images with a single estimated depth channel narrow this gap?
- **RQ3**: How does the quality of the depth estimate influence this potential improvement?

An overview of the proposed RGB-D OOD generalisation pipeline is illustrated in Figure 1.

We explore these questions through a series of controlled experiments on a modified version of the NICO++ dataset [36], a benchmark designed to test OOD robustness under domain shifts, as well as synthetic variants of the MNIST dataset [6] incorporating background and colour biases. Our approach compares several commonly used CNN backbones (ResNet-18, EfficientNet-B0, MobileNetV2, ShuffleNet V2) in their RGB and RGB-D configurations, using early fusion (depth concatenated at the input) to isolate the effect of geometric information while keeping architectural changes minimal. Depth maps are generated offline using state-of-the-art pretrained monocular estimators, and their impact is analysed both quantitatively, via OOD accuracy gaps and depth corruption ablations, and qualitatively, through class-resolved confusion patterns and representative input-level examples.

In summary, this work makes the following contributions:

1. Benchmarking OOD robustness of CNN classifiers under RGB-only training, quantifying baseline generalisation gaps across multiple architectures.
2. Introducing pseudo-depth augmentation as a simple, plug-and-play extension to RGB pipelines, requiring no architectural modification beyond input concatenation.
3. Empirically analysing the role of depth quality in OOD performance through controlled depth corruption, class-resolved confusion analysis, and qualitative input-level examples, showing that estimated depth can partially bridge the generalisation gap even without ground-truth supervision.

Together, these results aim to offer insight into how pseudo-depth can be leveraged to improve OOD generalisation efficiently, highlighting the role of geometric cues as a lightweight inductive bias (in the sense that it introduces geometric structure without increasing model capacity), that can complement appearance-based learning beyond controlled environments.

The remainder of this paper is organised as follows. Section 2 reviews related work on RGB-D fusion, domain generalisation, and efficiency-aware architectures. Section 3 describes the experimental methodology, including the controlled toy setup, the real-world NICO++ setting, and the generation and corruption of pseudo-depth signals. Section 4 presents experimental results addressing each research question, analysing both aggregate performance and depth-quality ablations. Section 5 discusses the implications, limitations, and broader significance of the findings, and section 6 concludes the paper. Additional analyses, including class-resolved confusion matrices and qualitative examples, and more in-depth results, are provided in the Appendix.

2. Related Work

In this section, we review three streams of literature most relevant to our research questions: RGB + depth fusion, domain generalisation and OOD robustness, and efficiency-aware architectures and fusion strategies, highlighting their strengths and limitations, and positioning our approach accordingly.

2.1. RGB-D fusion methods for recognition and segmentation

Methods that integrate RGB and depth data aim to exploit complementary geometric cues to improve recognition, segmentation, or detection performance, particularly in challenging visual conditions such as illumination changes, background clutter, or occlusion. Depth provides invariant structural information that complements appearance-based RGB features, helping models capture object shape, spatial layout, and relative distance.

A widely adopted design is the two-stream architecture, where RGB and depth images are processed by parallel encoder networks and fused at one or more stages, commonly through concatenation, summation, or attention-based feature aggregation [3, 8, 24, 25]. Early fusion combines the modalities at the input level, encouraging low-level feature alignment, while late fusion merges high-level representations, allowing the network to learn modality-specific semantics before jointly reasoning about them [10, 38].

More recent approaches introduce adaptive fusion mechanisms that learn how much to rely on each modality depending on scene context or signal quality. Cross-modal attention modules guide feature interaction between RGB and depth branches by selectively emphasising informative channels or spatial regions [2, 4, 19, 20, 29, 37]. Gating strategies achieve a similar goal by dynamically weighting modalities based on their reliability, mitigating the effect of noisy or missing depth data [5]. Transformer-based models further extend these ideas by employing self- and cross-attention layers to jointly encode RGB and depth tokens, enabling long-range context fusion [21, 34, 35].

Some works go further and adapt convolution operators to account for depth input by altering sampling offsets or receptive fields conditioned on depth. For example, Depth-Adapted CNN (Z-ACN) modifies 2D convolutions to sample from geometry-aware offsets computed from depth maps, effectively deforming receptive fields in response to local 3D structure. Experimental results show that this leads to improved segmentation performance in varying depth contexts [32]. More generally, Depth-Adapted CNNs for RGB-D Semantic Segmentation extend this idea by introducing depth-guided offsets and pooling operations to more tightly integrate geometry into feature extraction [33].

These methods assume available depth supervision (i.e. ground-truth depth maps) and often target dense prediction tasks such as segmentation or object detection. In contrast, our work focuses on (single-)image classification, and critically, uses estimated (pseudo-)depth. We also restrict ourselves to a plug-in early fusion rather than revising convolution operators, to preserve model simplicity and parameter efficiency.

2.2. Domain generalisation, OOD robustness, and multimodal cues

A growing field of research studies the domain shift/OOD robustness problem, especially in recognition tasks. Many methods rely on learning domain-invariant features, domain alignment, or adversarial training (e.g., invariant risk minimisation [1], domain adversarial networks [11]). Augmentation-based and regularisation techniques, including style randomisation, frequency perturbations, or texture-shape debiasing, have also been explored to reduce reliance on superficial cues [13, 18]. However, such methods often operate purely on RGB data and remain limited by the shortcomings of monocular appearance.

More recently, shortcut learning has been framed not only as a consequence of training objectives or insufficient regularisation, but also as a manifestation of inductive biases introduced by the input representation itself [14]. When models are trained solely on RGB data, the representation strongly emphasises texture, colour, and local appearance statistics, which can encourage reliance on spurious correlations that fail to transfer across domains [13]. From this perspective, improving robustness does not necessarily require more complex training schemes, but can also be achieved by altering the structure of the input signal to favour more transferable cues. Geometric information, such as depth, introduces shape- and layout-oriented structure at the representation level, potentially biasing models away from brittle appearance-based shortcuts.

In the multimodal and cross-domain context, RGB-D domain generalisation/domain adaptation methods have started to appear, notably for segmentation. For example, Depth-Sensitive Soft Suppression (DSSS) uses stylised depth augmentation and class-wise suppression to force the model to rely on less “sensitive” depth regions for domain-invariant representation, applied to RGB-D semantic segmentation tasks [31]. Also, some works in RGB-D domain adaptation explore cross-modality feature alignment or stylisation to mitigate domain shifts in depth and colour modalities [26].

These works show that depth maps can help in learning invariant shape-driven features less susceptible to superficial shifts. But they deal with segmentation or dense tasks, and often assume access to ground-truth depth. They also typically deploy heavier architectures. Our work is thus more constrained. We do not redesign network layers or require segmentation supervision or heavy architectures, but our hypothesis is that even minimal estimated geometric injection might improve robustness under shift.

A related line of work in robustness research emphasises the use of input-level ablations and negative controls to disentangle genuine information gain from incidental regularisation effects. For example, replacing informative signals with random or corrupted inputs can reveal whether observed improvements stem from meaningful structure or merely from increased input dimensionality. Such diagnostic interventions are particularly relevant in multimodal settings, where performance gains may otherwise be attributed to architectural changes or auxiliary channels rather than to the semantic content of the modality itself [18, 27].

2.3. Efficiency-aware architectures, fusion, and trade-offs

Compact CNN architectures are widely used in practice due to their favourable trade-offs between accuracy, computational cost, and ease of deployment. While such models are often trained under limited data or supervision, their behaviour under domain shift reflects broader challenges of shortcut learning in appearance-based recognition rather than properties specific to model size alone. There is comparatively less literature on how to fuse multimodal inputs (RGB + depth) in a manner that preserves efficiency while isolating the contribution of geometric cues.

While single-modality CNNs are well studied, there is relatively limited literature on how to fuse multimodal inputs such as RGB and depth efficiently. A central challenge lies in balancing computational cost with representational complementarity: many RGB-D networks rely on dual-stream encoders and attention-based fusion modules, substantially increasing model complexity and making it difficult to disentangle geometric benefits from architectural scaling.

Recent works have started exploring efficient RGB-D fusion under constrained settings. For instance, the Speed–Accuracy Tradeoff Network (SATNet) proposes a lightweight RGB-D saliency detection model that adjusts the degree of depth utilisation according to available computation, striking a balance between modality complementarity and inference cost [7]. Another work in camouflaged object detection builds a lightweight hybrid attention RGB-D network using MobileNetV2 backbone and boundary-aware modules to remain computationally efficient [22]. These methods show that RGB-D fusion can be achieved under efficiency constraints, but they primarily target pixel-wise tasks and often rely on task-specific attention mechanisms or multiple fusion stages, which are less suitable for studying robustness in classification. In contrast, our work adopts a deliberately minimal fusion strategy, concatenating estimated depth at the input, to study whether geometric cues alone can improve robustness under domain shift, without confounding effects from architectural redesign or increased model capacity.

3. Method

Our goal is to assess whether the inclusion of pseudo-depth cues can enhance classification performance in OOD settings. To this end, we designed two complementary experimental pipelines: a controlled toy setup and a real-world setup. The toy setup isolates the contribution of depth in a minimal environment, while the real-world setup evaluates the same idea under realistic conditions.

In both cases, we train standard CNN classifiers under controlled architectural settings, comparing a baseline RGB model against a model that also consumes (pseudo-)depth information. The RGB variant processes only the colour image $x \in \mathbb{R}^{3\times H\times W}$, whereas the RGB-D variant additionally takes a depth map $d \in \mathbb{R}^{1\times H\times W}$.

3.1. Toy setup: controlled RGB vs. RGB-D

The toy setup is designed as a controlled proof-of-concept to isolate the effect of depth cues on generalisation under a simple and fully interpretable setting. By using synthetic data with explicitly manipulated background contexts and depth signals, this setup allows us to examine whether depth improves robustness independently of architectural complexity, pre-training, or real-world confounds. The following subsections describe the model architectures and the experimental design used in this controlled scenario.

3.1.1. Model architecture

For the controlled experiment, we created a simple CNN trained from scratch to avoid any bias from pre-training. It consists of two convolutional blocks followed by fully connected layers producing class logits $\mathbf{o} \in \mathbb{R}^{C}$.

To study the contribution of depth, we designed a corresponding RGB-D version of it. This model has two parallel branches: one processing the RGB image x, the other processing its depth map d. Each branch replicates the simple CNN's structure and outputs a feature vector. The final fully connected layer that maps the concatenated feature vector to class logits is referred to as the classifier head. The two feature vectors coming from each branch are concatenated and passed through this classifier head (late fusion):

$$\mathbf{z}_{\text{rgb}} = f_\theta(x), \quad \mathbf{z}_{\text{d}} = g_\phi(d), \quad \mathbf{o} = h_\psi([\mathbf{z}_{\text{rgb}}; \mathbf{z}_{\text{d}}]), \tag{1}$$

where f_θ, g_ϕ denote the two CNN encoders and h_ψ the classifier head.

The training objective is the standard cross-entropy loss over the predicted logits:

$$\mathcal{L}_{\text{cls}} = -\log \frac{\exp(o_y)}{\sum_{c=1}^{C} \exp(o_c)}, \tag{2}$$

where y is the ground-truth class index and o_c the c-th logit.

Late fusion was selected in this phase to keep the architecture interpretable and to clearly separate the effect of the RGB and depth streams.

3.1.2. Experimental design and depth representation

To obtain a fully controlled environment, we created a custom variant of the MNIST dataset where each digit is white, and the background is uniformly colored.

The background colour acts as a ”context” variable that is irrelevant to the digit identity but provides a controlled source of distribution shift. We designed two complementary conditions:

- **Within-context (control) setup:** the model is trained and tested on digits with a black background.
- **Cross-context setup:** the model is trained only on digits with *red* and *green* backgrounds, and tested on digits with *red*, *green*, and *blue* backgrounds.

This structure allows us to measure how well the network generalises to unseen background colours (here, blue), which can act as a proxy for domain shift. All class distributions and background colours are balanced during training to ensure that accuracy differences reflect generalisation rather than imbalance.

Since real depth annotations are unavailable, we generated synthetic depth maps in a deterministic way. We assign a low depth value to the background and a high value to the digit, effectively creating a binary ”near-far” map:

$$d(i,j) = \begin{cases} 1, & \text{if } (i,j) \text{ belongs to the digit,} \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

Both RGB and RGB-D models are trained under identical optimisation settings and data balancing to isolate the impact of adding depth.

3.2. Real-world setup: early-fusion architectures

After confirming improvements in the toy scenario, we applied the approach (RGB vs. RGB-D experiments) to a real-world setting using the NICO++ dataset, which contains natural images of objects under varied contexts. To minimise architectural complexity and isolate the effect of geometric cues, we adopted an early-fusion strategy instead. The depth channel is concatenated with the RGB channels at the input, forming a 4-channel tensor $\tilde{x} = [x; d]$. This tensor is directly fed to a single backbone network, without duplicating weights as in late fusion:

$$\mathbf{o} = f_\theta(\tilde{x}), \quad \tilde{x} \in \mathbb{R}^{4 \times H \times W}. \tag{4}$$

This approach preserves a simple architectural design while allowing the network to learn cross-modal interactions from the first convolutional layer. We evaluated four commonly used CNN classification backbones in this configuration: *ResNet-18*, *EfficientNet-B0*, *MobileNetV2*, and *ShuffleNet V2*. We focus on the smallest standard variants of each backbone to limit model capacity and isolate the effect of depth as an auxiliary signal. Larger models may partially compensate for missing or noisy depth through increased representational power, obscuring modality-specific contributions. Using lightweight and widely adopted variants enables a more controlled and interpretable comparison across architectures. All networks were trained from scratch to ensure comparable initialisation and training conditions.

3.3. Pseudo-depth generation for real images

For the real-world setup, we did not rely on ground-truth depth. Instead, we generated pseudo-depth maps using the largest version of a state-of-the-art monocular depth estimator, Distill-Any-Depth [17]. This provided an additional channel $d = \mathcal{D}(x)$ per image, approximating relative depth from a single RGB input.

To study robustness to noisy depth signals, we further created corrupted variants using Gaussian noise:

$$\tilde{d} = \text{clip}(d + \sigma\,\varepsilon, 0, 1), \quad \varepsilon \sim \mathcal{N}(0, \mathbf{I}), \tag{5}$$

where σ controls the noise level (low, medium, or pure noise). This allowed us to analyse whether networks can still benefit from pseudo-depth even when the depth quality degrades.

3.4. Training objective and procedure

All models, both RGB and RGB-D, were trained using the same loss function in Equation 2. Optimisation followed standard stochastic gradient descent with adaptive learning rates and weight decay, ensuring identical training conditions across modalities. The detailed hyperparameters (learning rate, batch size, epochs, etc.) are provided in the next section.

During evaluation, accuracy was measured per domain (background colour or NICO++ context) for direct comparison of model generalisation performance. Although RGB-D models are trained for a larger maximum number of epochs than RGB models in some experiments to accommodate potentially slower convergence, this does not confer an optimisation advantage. In all cases, model selection is based on validation accuracy, and test performance is reported using the checkpoint with the highest validation accuracy for each run.

4. Experiments

This section presents the experimental evaluation of our approach and addresses the RQs outlined in the introduction. We structure the experiments to progress from a highly controlled toy setting to a realistic, large-scale benchmark, allowing us to first isolate causal effects and then assess their robustness under real-world conditions. Across both settings, we focus on measuring OOD generalisation under contextual shifts, comparing RGB-only models to their RGB-D counterparts and analysing how the inclusion and quality of predicted depth influence performance. Together, these experiments are designed to disentangle whether pseudo-depth acts merely as an additional input channel or as a meaningful auxiliary signal that improves robustness by mitigating reliance on spurious visual cues.

Across all experiments, we compare RGB models to their RGB-D counterparts under matched training protocols and report mean ± sample standard deviation over three runs. Unless stated otherwise, models are trained from scratch, evaluated on fixed held-out splits, and only the input modality (RGB vs. RGB-D) or depth quality is varied.

4.1. Toy setting

We first study a controlled toy setting based on a custom variation of MNIST to isolate the causal role of depth under a known contextual shift. By construction, background colour acts as a spurious cue for RGB models, while the pseudo-depth channel preserves digit shape independently of colour. This setting, therefore, serves as a diagnostic test of whether adding depth reduces sensitivity to background context before moving to a realistic benchmark.

4.1.1. Experimental setup

We use custom MNIST variants where digits are composited over uniform backgrounds (red/green/blue/black). Images are resized to 28×28 and normalised. Depth maps are pre-generated as binary near–far masks. For the "RGB" toy variant, training uses red + green backgrounds, and testing reports accuracies on blue (unseen), green, and red. A separate "black" toy variant is trained and evaluated on black backgrounds to provide an upper boundary on the model's performance. Each digit class appears 20 times in the training set (10 on red and 10 on green backgrounds, or 20 on black backgrounds), 10 times in the validation set (5 per red and green colour, and 10 for the black colour), and 10 times in each test set (blue, red, green, or black backgrounds), yielding 200, 100, and 100 images per split, respectively. All sets are strictly non-overlapping and balanced across classes and background colours.

Figure 2 shows representative samples from our custom MNIST toy dataset. In-distribution (ID) images use the red and green backgrounds observed during training, while the blue background is held out and constitutes the OOD context shift. For the RGB-D variant, the accompanying depth channel is not predicted but synthetically defined as a binary near-far mask. Digit foreground pixels are assigned a value of 1 (white) and background pixels a value of 0 (black). This construction preserves digit shape while removing colour, making depth informative about geometry but uninformative about the background context.

We instantiate a compact CNN for RGB and its RGB-D counterpart. RGB-D uses dual streams (one for RGB, one for single-channel depth) with feature concatenation before the classifier. The models are used both in the within-context and cross-context setups for consistency and are always trained from scratch.

For RGB runs, we train for 150 epochs with AdamW with a learning rate of $1e^{-3}$ and weight decay of $5e^{-4}$. For RGB-D runs, we train for 200 epochs with the same optimiser. Each setting is repeated for 3 iterations, and we report the mean ± sample standard deviation. Randomness is controlled by a base seed of 42 that is incremented per iteration.

4.1.2. Results on custom MNIST - Experiment 1

To validate our core hypothesis in a controlled setting before moving to the real-world NICO++ benchmark, we first conducted experiments on our synthetic MNIST-based dataset, reporting accuracy outcomes. Models were trained on digits placed on red and green backgrounds, and then evaluated on held-out red, green, and blue backgrounds. The blue background constitutes the OOD shift, as the model never observes this colour during training. We also included a black-background control experiment, in which both RGB and RGB-D versions were trained and tested on uniform black backgrounds, ensuring that no colour-based shortcut is available at all.

The results, averaged over three runs, are shown in Table 1. The RGB baseline exhibits strong performance on the seen red and green backgrounds but drops substantially on the unseen blue background (61.67% ± 23.12). The large standard deviation indicates substantial run-to-run variability, suggesting that RGB-only models are highly sensitive to stochastic factors when forced to generalise beyond the training contexts. This behaviour is consistent with reliance

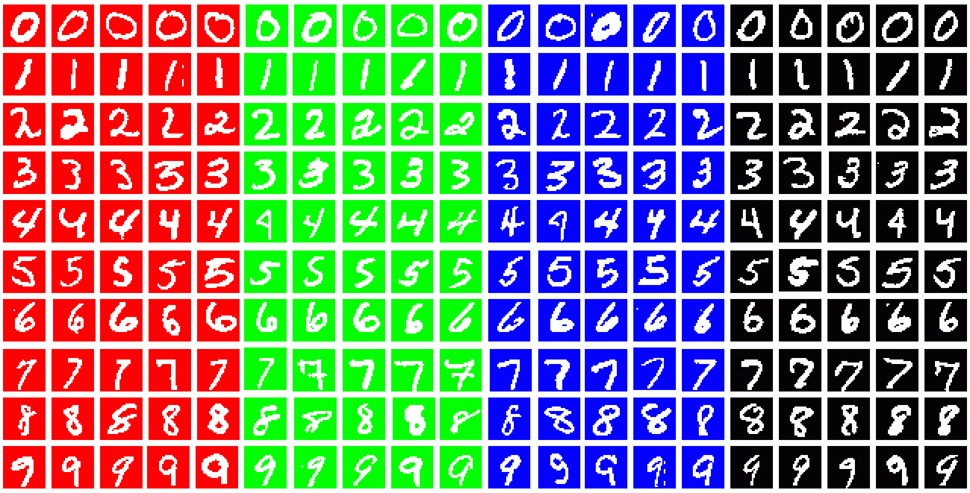


Figure 2. **Example images of the coloured-background toy MNIST.** Training/in-distribution (ID) images have backgrounds that are *red* and *green*. The *blue* background images are held out and used as the out-of-distribution (OOD) context at test time.

on colour-specific cues that fail under distribution shift.

In contrast, the RGB-D model shows a clear improvement on the unseen blue background, achieving a substantially higher mean accuracy (81.33% ± 4.16). Importantly, this improvement is accompanied by a dramatic reduction in variance across runs. While the RGB model's OOD performance varies widely depending on the training instance, the RGB-D model exhibits consistently strong generalisation under the same shift. Performance on the seen red and green backgrounds remains comparable between RGB and RGB-D, indicating that the gains under OOD conditions do not come at the expense of in-distribution accuracy.

On the black-background control, both models achieve similar accuracies, suggesting that the gains under distribution shift arise from the complementary depth information rather than added model capacity alone. Overall, these results show that incorporating depth information not only improves average OOD accuracy but also significantly stabilises performance across runs, providing an early indication that depth cues mitigate brittle reliance on background appearance.

4.2. Real-world setting

We then evaluate the same hypothesis on NICO++, a large-scale benchmark designed to probe OOD generalisation under natural contextual variation. Unlike the toy setting, shifts in NICO++ involve complex changes in texture, scene layout, and background statistics. We therefore use this setting to test whether the gains from pseudo-depth persist under realistic domain shift, and to quantify how depth quality affects the observed improvements.

4.2.1. Experimental setup

We use NICO++ with 20 object classes. Training contexts are "autumn", "dim", "rock", and "water", and unseen contexts are "grass" and "outdoor". For each (class, context) pair in the training set, we sample 70 images for training, 15 for validation, and 15 for ID testing ("seen"). For each (class, context) pair in the unseen set, we sample 15 images for OOD testing ("unseen"). This yields per class: 280 training (4 × 70), 60 validation (4 × 15), 60 seen-testing (4 × 15), and 30 unseen-testing images (2 × 15). All splits are disjoint and class-balanced within each context. To ensure sufficient and balanced data per category, we selected the 20 object classes in NICO++ with the largest number of available samples, thereby avoiding underrepresented or highly imbalanced classes. The choice of "seen" versus "unseen" contexts follows a similar rationale: we grouped "autumn", "dim", "rock", and "water" as "seen" contexts, while reserving "grass" and "outdoor" as "unseen" contexts because these two are semantically related and visually similar, which makes them a natural OOD target group and helps avoid bias that could arise from mixing them with the training domains.

Figure 3 shows representative RGB images from the NICO++ dataset alongside their corresponding monocular depth predictions. The first four examples illustrate ID training contexts, while the last two correspond to unseen OOD test contexts. The associated depth maps, generated using Distill-Any-Depth, capture coarse geometric structure that is largely invariant to background appearance, and are used as the additional input channel in our RGB-D experiments.

Inputs are resized/cropped to 224 × 224. RGB training uses random resized crop with a scale of 0.35 – 1, horizontal

Table 1. **Aggregated accuracy results (mean ± sample standard deviation (in %) over three runs) for the toy MNIST setup.** Models are trained on digits placed on red and green backgrounds and evaluated on in-distribution (ID) red, green, and unseen out-of-distribution (OOD) blue backgrounds, with a separate black-background control experiment included. The RGB-D model shows improved and more stable performance under the unseen blue background shift. Bold values indicate the results on the *Blue Test* set, which represents the unseen background OOD condition.

	Coloured Background Experiment					**Black Background Control**		
	Train / Val		**Test**			**Train / Val**		**Test**
	Train	**Val**	**Red (ID)**	**Green (ID)**	**Blue (OOD)**	**Train**	**Val**	**Black**
RGB	97.33 ± 3.06	86.33 ± 3.21	81.67 ± 5.86	83.67 ± 4.58	**61.67** ± **23.12**	99.67 ± 0.29	87.33 ± 5.51	84.33 ± 5.69
RGB-D	98.83 ± 2.02	87.33 ± 2.52	82.67 ± 1.53	83.33 ± 1.53	**81.33** ± **4.16**	95.50 ± 5.77	86.33 ± 4.04	84.00 ± 7.00

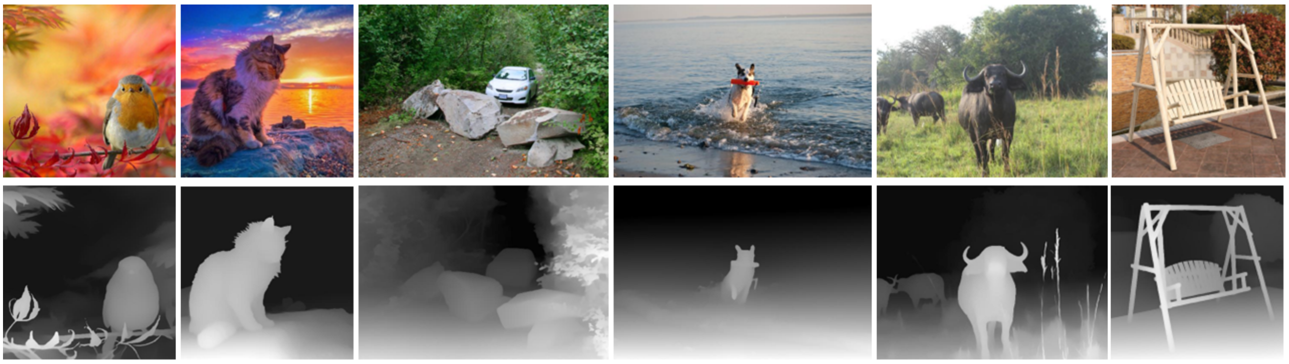


Figure 3. **Example images from the NICO++ dataset with predicted depth.** The top row shows RGB images from NICO++, while the bottom row shows the corresponding monocular depth maps generated using *Distill-Any-Depth*. The first four columns correspond to in-distribution (ID) training contexts: ”bird” in the ”autumn”, ”cat” in ”dim” lightning, ”car” on ”rock”, and ”dog” in ”water”. The last two columns correspond to out-of-distribution (OOD) test contexts held out during training: ”cow” on ”grass” and ”chair” ”outdoor”. Depth maps are generated offline and used as the additional input channel in the RGB-D models.

flip, and colour jitter, followed by ImageNet normalisation. Evaluation uses resizing to 256 and centre crop. For RGB-D, we use early fusion. The depth channel is concatenated with RGB to form a 4-channel tensor. To preserve cross-modal spatial alignment, we apply joint transforms that use the same sampled crop and flip to both RGB and depth. Per-modality normalisation is then applied before concatenation. Depth maps are precomputed monocular pseudo-depths, generated using a state-of-the-art monocular depth estimator at the time of writing this paper.

To investigate RQ3, we additionally construct three noisy variants of these pseudo-depth maps. Starting from the original grayscale depth map, we first normalise pixel values to the [0, 1] range. For the low-noise condition, we add pixel-wise Gaussian noise with zero mean and standard deviation $\sigma = 0.03$, and clip the result back to [0, 1]. For the medium-noise condition, we increase the standard deviation to $\sigma = 0.08$, again followed by clipping. Finally, for the pure-noise condition, we ignore the original depth map altogether and instead sample each pixel independently from a Gaussian distribution with mean 0.5 and standard deviation 0.5, clipping again to [0, 1]. The RGB images remain unchanged in all cases, and only the depth channel is perturbed. These three depth variants are used both at training and at test time in the corresponding noise-corruption experiments.

We evaluate four commonly used CNN classifiers, ResNet-18, EfficientNet-B0, MobileNetV2, and ShuffleNet V2, under two modalities: RGB (3-channel input) and RGB-D (4-channel early fusion by replacing the first convolution to accept 4 channels). Within each modality, all hyperparameters are identical across backbones.

We train with AdamW with learning rate $1e^{-3}$, weight decay $5e^{-4}$, and β = (0.9, 0.999), cosine LR with 5 warm-up epochs, batch size 32, input size 224, and the same seed and incrementing strategy used in the toy setup. All models train for 200 epochs, except for ShuffleNet in the RGB-D experiments, which trains for 300 epochs to accommodate its longer convergence time. We monitor validation accuracy each epoch and retain the checkpoint with the best validation performance. The final results are computed with this checkpoint. We report accuracy on the ”seen” contexts test set and the ”unseen” contexts test set. Each experiment is repeated 3 times with different randomisation, and we report the mean ± sample standard deviation of the results across these runs.

Of note is that in the toy setup, we adopted a late-fusion design as a straightforward way to verify whether the inclusion of (pseudo-)depth could improve generalisation. This configuration, with two parallel branches for RGB and depth, served as a controlled proof of concept to test if depth cues can enhance OOD performance. However, late fusion nearly doubles the number of parameters and computational cost, since both branches replicate the full backbone. For the real-world experiments, we therefore opted for an early-fusion strategy, concatenating the depth map with the RGB channels at the input, so that the model learns cross-modal features from the first layer while keeping the parameter increase negligible. This allows us to test whether the observed benefits of depth persist even under a minimal and practically relevant architectural configuration.

Lastly, in all experiments, the models were not tuned through explicit hyperparameter optimisation, as it was beyond the scope of our research. Instead, we trained each network for a sufficiently long schedule (e.g., 150 epochs for RGB, 200 epochs for RGB-D in the toy setup) to ensure convergence, while monitoring validation accuracy. The checkpoint with the highest validation performance was retained and used for final testing, and all reported results correspond to this best-validation model.

4.2.2. Results on NICO++

In the real-world setting, we now evaluate our approach on the NICO++ benchmark, where contextual variation is natural and more challenging, and evaluate whether the same RGB vs. RGB-D trends persist across multiple standard CNN backbones. For the real-world setup, we summarise OOD behaviour using the seen-unseen context generalisation gap Δ (Equation 6), and additionally analyse how this gap changes when the depth channel is progressively degraded with noise. We structure the analysis into a sequence of focused experiments that address our research questions.

4.2.2.1 Experiment 2: What is the baseline in-distribution vs. OOD performance gap for RGB-only models?

Before evaluating any depth-augmented variant, we must first quantify how strong the OOD gap is when using conventional RGB models. This establishes a reference point for later experiments and clarifies the extent to which colour-based shortcuts affect generalisation on NICO++.

We train four widely used CNN architectures, ResNet-18, EfficientNet-B0, MobileNetV2, and ShuffleNet V2, on the NICO++ training split described in 4.2.1. At test time, the models are evaluated both on the seen contexts and on held-out unseen contexts representing genuine OOD shifts in background not observed during training. The OOD generalisation behaviour is summarised via the difference:

$$\Delta = \text{Seen Accuracy} - \text{Unseen Accuracy}, \qquad (6)$$

where a larger Δ indicates stronger reliance on context-specific cues.

Table 2 reports the mean $\pm$ standard deviation of Δ across three independent runs for each architecture. All four RGB models exhibit a substantial OOD degradation, with accuracy gaps consistently around 10%. MobileNet, ShuffleNet, and ResNet-18 show the largest gaps (10.78%, 10.67%, and 10.66%, respectively), closely followed by EfficientNet (10.22%). Notably, the standard deviations are small across all models, below 1%, indicating that the observed OOD degradation is systematic rather than the product of training noise. In other words, the reliance on background cues is a stable and reproducible property of RGB-only training on NICO++.

Additionally, while our analysis focuses on the ”seen” - ”unseen” accuracy generalisation gap Δ, the underlying absolute test accuracies for each run and model, under all experimental conditions, are reported in the Appendix A.4 for transparency and reproducibility.

4.2.2.2 Experiment 3: Can augmenting RGB images with a single estimated depth channel narrow the OOD gap?

Having established the RGB baseline, we now turn to RQ2, which asks whether augmenting each input image with a single estimated depth channel can systematically reduce the OOD performance gap. The motivation is to test whether depth cues, though approximate and predicted rather than ground-truth, provide complementary information that discourages reliance on background colour or texture.

We evaluate the same four CNN architectures as before (ResNet-18, EfficientNet-B0, MobileNetV2, ShuffleNet V2), but now each model receives a 4-channel input comprising RGB plus a predicted depth map. The training and evaluation protocol remains unchanged from 4.2.1, and we again quantify robustness using the context generalisation gap Δ from Equation 6.

Table 2 again reports the mean $\pm$ standard deviation of Δ across three independent runs for the RGB-D models. In contrast to the RGB-only baselines, all architectures exhibit a consistently smaller OOD degradation. Across models, the gap decreases from roughly 10% (RGB) to around 8% when depth is added. The reduction is most pronounced for MobileNet and ResNet-18 (down to 8.17% and 8.16%), while EfficientNet also benefits, achieving the lowest variance among all models (8.92% $\pm$ 0.22). These findings indicate that depth cues, despite being predicted rather than measured, provide a meaningful signal that mitigates reliance on colour-based shortcuts.

When examining the variability across runs, ShuffleNet and ResNet-18 exhibit noticeably higher standard deviations (1.63% and 1.45%), suggesting that while depth generally improves their OOD performance, the extent of improvement varies more across random initialisations. This increased variance indicates a degree of sensitivity to optimisation dynamics in these architectures when depth is introduced. Nonetheless, the reduction in the mean OOD gap remains consistent across all four models.

Overall, the results provide a clear answer to RQ2: adding a single predicted depth channel consistently narrows the OOD generalisation gap across all tested CNN architectures.

4.2.2.3 Experiment 4: How does the quality of the depth estimate influence the improvement?

To better understand the robustness of the depth-augmented models, we go to RQ3 to investigate how sensitive the observed gains are to the quality of the depth estimate. While earlier results demonstrated that predicted depth maps consistently reduce the OOD generalisation gap, it is natural to ask whether this benefit depends on the fidelity of the depth signal or whether even approximate estimates are sufficient to discourage shortcut learning.

To this end, we explicitly corrupt the predicted depth maps with increasing levels of noise. A low-noise condition in which the structural content of the depth signal remains largely intact, a medium-noise condition that introduces substantial degradation, and a pure-noise condition in which the depth channel contains no geometric information at all. Each corrupted depth map is then fed to the RGB-D models during both training and testing, allowing us to directly assess how OOD generalisation depends on the available geometric signal. As in previous experiments, robustness is quantified by the context generalisation gap Δ from Equation 6, with lower values indicating better OOD generalisation.

The results, summarised in Table 2, reveal a clear and systematic pattern. Under low noise, most architectures preserve the gains observed with clean predicted depth. EfficientNet and MobileNet achieve the smallest gaps (8.72% and 9.25%, respectively), and their relatively low standard deviations indicate stable behaviour across runs. ResNet-18 also benefits from low-noise depth, although with higher variability. In contrast, ShuffleNet stands out in this regime. Despite low noise, it exhibits a substantially larger gap (11.33% ± 0.15), comparable to the RGB-only baseline, and shows little variance across runs. This indicates that, for ShuffleNet, adding depth under low-noise conditions does not yield the same degree of OOD improvement as observed for the other architectures.

When moving to medium noise, the gaps generally increase but remain consistently lower than those of RGB-only training. ResNet-18 achieves the smallest gap (8.84% ± 0.17), while MobileNet and EfficientNet exhibit larger gaps and higher variability, indicating that some architectures are more sensitive to imperfections in the depth channel than others. Interestingly, ShuffleNet now shows a reduced gap (8.92% ± 0.58) compared to the low-noise condition, indicating that its OOD performance is not monotonically related to depth quality. Nonetheless, even moderately degraded depth remains beneficial relative to omitting it entirely across architectures.

The pure-noise condition provides an important control. If the depth channel were merely acting as a regulariser or additional input dimension, we would expect performance to resemble that of the medium-noise condition. Instead, all models experience a clear degradation, with gaps rising to approximately 9–11%. This demonstrates that the improvements observed in earlier experiments are not due to the mere presence of an extra channel, but specifically arise from depth encoding meaningful geometric structure. The fact that adding completely uninformative depth worsens performance for several models highlights that the networks do attempt to use the depth signal, and are adversely affected when it becomes misleading.

The depth-quality ablation can be viewed as a controlled input-level intervention that isolates the role of geometric information in the observed robustness gains. By progressively degrading the depth channel while keeping the architecture, training procedure, and RGB inputs fixed, we explicitly decouple the presence of an additional channel from the presence of meaningful geometric structure. The degradation in OOD performance as depth quality decreases, and the disappearance of the benefit under pure noise, indicate that the improvements are not due to regularisation or increased input dimensionality. Instead, they arise specifically from depth encoding usable geometric cues. This experiment, therefore, provides evidence that pseudo-depth contributes as an informative auxiliary signal, rather than as a spurious or shortcut-inducing modality.

Overall, these results reveal that the quality of the depth estimate meaningfully shapes the extent of the OOD improvement. While coarse or noisy depth still provides useful inductive bias, the benefits diminish as noise increases, and disappear entirely when geometric structure is lost. These results further strengthen the interpretation that depth information, even when predicted and imperfect, can guide CNNs away from brittle context cues and toward more robust representations. A class-resolved analysis and qualitative input-level of these effects is provided in the Appendix.

Table 2. **NICO++ context generalisation gap across input modalities and depth quality.** We report the seen - unseen context gap $\Delta = \text{Acc}_{\text{seen}} - \text{Acc}_{\text{unseen}}$ (see Equation 6) (in %), as mean ± sample standard deviation over three runs. Lower Δ indicates better out-of-distribution (OOD) generalisation. *Clean* denotes predicted depth (RGB-D) without noise corruption. *Low/Medium* add varying levels of Gaussian noise to the depth channel. *Pure* replaces depth with random noise.

Backbone	**Baseline**		**Depth quality ablation (RGB-D)**		
	RGB	**RGB-D (Clean)**	**Low**	**Medium**	**Pure**
ResNet-18	10.66 ± 0.65	8.16 ± 1.45	9.36 ± 1.33	8.84 ± 0.17	10.53 ± 1.19
EfficientNet	10.22 ± 0.05	8.92 ± 0.22	8.72 ± 0.56	10.06 ± 1.28	9.56 ± 1.27
MobileNet	10.78 ± 0.63	8.17 ± 0.52	9.25 ± 1.04	9.91 ± 1.67	10.75 ± 1.26
ShuffleNet	10.67 ± 0.00	8.20 ± 1.63	11.33 ± 0.15	8.92 ± 0.58	8.36 ± 1.85

5. Discussion

This work set out to examine whether estimated depth, when used as a simple auxiliary signal, can improve the robustness of standard CNN classifiers under domain shift. Across both controlled toy experiments and real-world evaluations on NICO++, our results consistently show that augmenting RGB inputs with a single pseudo-depth channel reduces OOD performance gaps, without harming ID accuracy. Crucially, these gains are observed even though the depth signal is predicted rather than measured, and no architectural changes are introduced beyond minimal input fusion. Taken together, the findings suggest that even approximate geometric cues can act as an effective inductive bias, encouraging models to rely less on superficial appearance correlations and more on structural properties that transfer across environments.

5.1. Model scale and generality of findings

The experiments in this work focus on compact and commonly used CNN architectures, and do not include large-scale models such as vision transformers or foundation models pretrained on massive datasets. This choice was intentional. Restricting model scale allows us to isolate the empirical effect of introducing geometric information on robustness outcomes by holding architecture, optimisation, and data constant while varying only the input modality. Importantly, we do not claim that the observed improvements are specific to lightweight models, nor that larger models would fail to generalise without depth. Rather, our results support the interpretation that depth acts as a geometric inductive bias. It introduces structural cues that are less sensitive to contextual appearance shifts, independently of model size. While larger models may rely less on shortcut correlations due to their scale and data diversity, there is no reason to expect that geometric information would become irrelevant under domain shift. Verifying how pseudo-depth interacts with large-scale pretrained models remains an important direction for future work, but is beyond the scope of this study.

Furthermore, to verify that the observed improvements are not driven by the model collapsing onto depth as a shortcut, we conducted an additional controlled experiment in which depth alone is insufficient to solve the task. The results, reported in the Appendix A.1, confirm that effective performance requires integrating both RGB and depth cues, and that the model does not rely on depth in isolation.

To better understand how these robustness gains manifest at the class level, we further analyse the structure of the models' errors using aggregated confusion matrices, reported in the Appendix A.2. This analysis shows that the reduction in OOD error is not uniform across classes, but is driven by the attenuation of a small number of structured semantic confusions (e.g., within vehicle and animal categories). Importantly, these class-resolved patterns mirror the depth-quality ablation results: clean depth reduces dominant confusions, while progressively corrupted depth causes the confusion structure to revert to that of RGB-only models. This provides additional evidence that pseudo-depth contributes meaningful geometric information rather than acting as a generic regulariser.

These class-resolved effects are further illustrated through qualitative input-level examples. Figure 9 in the Appendix A.3 shows representative unseen-context images for which RGB-only models fail, while RGB-D models with clean estimated depth recover the correct class. In these examples, adding depth resolves the same animal–animal, vehicle–vehicle, and shape-driven confusions identified in the confusion-matrix analysis, while replacing depth with pure noise removes this benefit. These cases provide an interpretable, input-level view of how geometric cues reduce reliance on appearance-based shortcuts under domain shift.

5.2. Limitations

While the results consistently demonstrate that pseudo-depth can improve robustness under domain shift, several limitations of this study should be acknowledged. First, all real-world experiments rely on a single monocular depth estimation model to generate pseudo-depth maps. Although this estimator represents the state of the art at the time of writing, its predictions are inevitably imperfect and may encode biases specific to its training data. While our noise corruption experiments partially address robustness to degraded depth quality, we do not evaluate how different depth estimators, or depth models trained under different regimes, might affect downstream classification performance.

Second, depth maps are generated offline and treated as fixed inputs during classifier training. This design choice allows us to isolate the effect of depth as an auxiliary signal, but it precludes any end-to-end adaptation between depth prediction and classification. Joint optimisation could potentially amplify or attenuate the observed benefits, particularly under domain shift, but would introduce additional confounding factors that are outside the scope of this work.

Third, our evaluation is limited to image classification tasks. While classification provides a clean testbed for analysing shortcut reliance and context bias, it remains unclear to what extent the observed effects transfer to other vision tasks such as detection or segmentation, where depth information is already known to play a more explicit role. Finally, although the experiments span both synthetic and real-world datasets, they do not cover extreme distribution shifts or open-set conditions, which may require stronger forms of invariance than those induced by a single geometric cue.

5.3. Future work

Several directions for future work follow naturally from the findings of this study. A first direction concerns the interaction between pseudo-depth and model scale. While we deliberately focused on compact CNN architectures to isolate the effect of geometric cues, it would be valuable to examine whether similar robustness gains persist in larger, pretrained models, including vision transformers and hybrid architectures. Such an investigation could clarify whether geometric inductive biases remain beneficial when models have access to substantially greater capacity and visual diversity.

A second avenue is the integration of depth prediction and classification in an end-to-end framework. Joint optimisation may allow the depth estimator to adapt its representations to the needs of the downstream task, potentially improving robustness under domain shift. At the same time, end-to-end training raises important questions about stability and shortcut formation, which would require careful experimental design to disentangle.

Beyond architecture and training strategy, future work could explore alternative forms of geometric or structural supervision. While this work focuses on monocular depth, other signals such as surface normals, relative ordering, or coarse shape priors may provide complementary inductive biases that further reduce reliance on appearance-driven shortcuts. Finally, extending the analysis to additional tasks and more diverse domain shifts, including open-set and long-tail scenarios, would help assess the generality of the observed effects and clarify the broader role of geometry in robust visual recognition.

6. Conclusion

This paper examined whether estimated depth, obtained from monocular RGB images, can serve as a practical auxiliary signal to improve the robustness of CNN classifiers under domain shift. Through controlled toy experiments and real-world evaluations on NICO++, we showed that augmenting standard RGB pipelines with a single pseudo-depth channel consistently reduces out-of-distribution performance gaps, while preserving in-distribution accuracy. These improvements hold across multiple CNN architectures and persist even when depth estimates are noisy, provided that they retain meaningful geometric structure.

The key implication of these findings is that geometric cues need not be precise or sensor-derived to be useful. Even imperfect, predicted depth can act as a stabilising inductive bias, encouraging models to rely less on superficial appearance correlations and more on structural properties that transfer across environments. Importantly, this benefit is achieved without architectural redesign, additional supervision, or increased model capacity, making pseudo-depth augmentation a simple and accessible extension to existing RGB classification pipelines.

At the same time, our results do not suggest that pseudo-depth alone is sufficient for robust generalisation, nor that it replaces other strategies such as scale, data diversity, or explicit regularisation. Rather, the evidence indicates that geometry, even if approximate, provides complementary information that can meaningfully narrow robustness gaps in settings where models are otherwise prone to shortcut learning. In this sense, the contribution of this work lies not in proposing a new model, but in empirically demonstrating that even minimal geometric information can play a constructive role in improving out-of-distribution behaviour in practical vision systems.

References

[1] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019. 3

[2] Hong-Bo Bi, Zi-Qi Liu, Kang Wang, Bo Dong, Geng Chen, and Ji-Quan Ma. Towards accurate rgb-d saliency detection with complementary attention and adaptive integration. *Neurocomputing*, 439:63–74, 2021. 3

[3] Hao Chen, Youfu Li, Yongjian Deng, and Guosheng Lin. Cnn-based rgb-d salient object detection: Learn, select, and fuse. *International Journal of Computer Vision*, 129(7): 2076–2096, 2021. 3

[4] Xiaokang Chen, Kwan-Yee Lin, Jingbo Wang, Wayne Wu, Chen Qian, Hongsheng Li, and Gang Zeng. Bi-directional cross-modality feature propagation with separation-and-aggregation gate for rgb-d semantic segmentation. In *European conference on computer vision*, pages 561–577. Springer, 2020. 3

[5] Zuyao Chen, Runmin Cong, Qianqian Xu, and Qingming Huang. Dpanet: Depth potentiality-aware gated attention network for rgb-d salient object detection. *IEEE Transactions on Image Processing*, 30:7012–7024, 2020. 3

[6] Li Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012. 2

[7] Songsong Duan, Xi Yang, Nannan Wang, and Xinbo Gao. Lightweight rgb-d salient object detection from a speed-accuracy tradeoff perspective, 2025. 4

[8] Andreas Eitel, Jost Tobias Springenberg, Luciano Spinello, Martin Riedmiller, and Wolfram Burgard. Multimodal deep learning for robust rgb-d object recognition. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 681–687. IEEE, 2015. 3

[9] Andrea Ferreri, Silvia Bucci, and Tatiana Tommasi. Multimodal rgb-d scene recognition across domains. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2199–2208, 2021. 1

[10] Keren Fu, Deng-Ping Fan, Ge-Peng Ji, and Qijun Zhao. Jl-dcf: Joint learning and densely-cooperative fusion framework for rgb-d salient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3052–3062, 2020. 3

[11] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario March, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of machine learning research*, 17(59):1–35, 2016. 3

[12] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361, 2012. 1

[13] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. In *International conference on learning representations*, 2018. 3

[14] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020. 1, 3

[15] Giorgio Giannone and Boris Chidlovskii. Learning common representation from rgb and depth images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 0–0, 2019. 1

[16] Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. *arXiv preprint arXiv:2007.01434*, 2020. 1

[17] Xiankang He, Dongyan Guo, Hongji Li, Ruibo Li, Ying Cui, and Chi Zhang. Distill any depth: Distillation creates a stronger monocular depth estimator, 2025. 5

[18] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8340–8349, 2021. 3, 4

[19] Xinxin Hu, Kailun Yang, Lei Fei, and Kaiwei Wang. Acnet: Attention based network to exploit complementary features for rgbd semantic segmentation. In *2019 IEEE international conference on image processing (ICIP)*, pages 1440–1444. IEEE, 2019. 3

[20] Guanbin Li, Yukang Gan, Hejun Wu, Nong Xiao, and Liang Lin. Cross-modal attentional context learning for rgb-d object detection. *IEEE Transactions on Image Processing*, 28 (4):1591–1601, 2018. 3

[21] Peng Liu, Jinhong Deng, Lixin Duan, Wen Li, and Fengmao Lv. Segmenting anything in the dark via depth perception. *IEEE Transactions on Multimedia*, 2025. 3

[22] Yang Liu, Shuhan Chen, Haonan Tang, and Shiyu Wang. Lightweight hybrid attention rgb-d networks for accurate camouflaged object detection. *The Visual Computer*, pages 1–17, 2025. 4

[23] Jiajun Ma, Yanmin Zhou, Zhipeng Wang, Hongrui Sang, Rong Jiang, and Bin He. Geometric-aware rgb-d representation learning for hand–object reconstruction. *Expert Systems with Applications*, 257:124995, 2024. 1

[24] Seong-Jin Park, Ki-Sang Hong, and Seungyong Lee. Rdfnet: Rgb-d multi-level residual feature fusion for indoor semantic segmentation. In *Proceedings of the IEEE international conference on computer vision*, pages 4980–4989, 2017. 3

[25] Houwen Peng, Bing Li, Weihua Xiong, Weiming Hu, and Rongrong Ji. Rgbd salient object detection: A benchmark and algorithms. In *European conference on computer vision*, pages 92–109. Springer, 2014. 3

[26] Giulia Rizzoli, Donald Shenaj, and Pietro Zanuttigh. Source-free domain adaptation for rgb-d semantic segmentation with vision transformers. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 615–624, 2024. 4

[27] Kishore Sampath, Ayaazuddin Mohammad, Resmi Ramachandranpillai, et al. The multimodal paradox: How added and missing modalities shape bias and performance in multimodal ai. *arXiv preprint arXiv:2505.03020*, 2025. 4

[28] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *European conference on computer vision*, pages 746–760. Springer, 2012. 1

[29] Yingcai Wan, Qiankun Zhao, Jiqian Xu, Huaizhen Wang, and Lijin Fang. Dagnet: Depth-aware glass-like objects segmentation via cross-modal attention. *Journal of Visual Communication and Image Representation*, 100:104121, 2024. 3

[30] Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, Tao Qin, Wang Lu, Yiqiang Chen, Wenjun Zeng, and Philip S Yu. Generalizing to unseen domains: A survey on domain generalization. *IEEE transactions on knowledge and data engineering*, 35(8):8052–8072, 2022. 1

[31] Binbin Wei, Yuhang Zhang, Shishun Tian, Muxin Liao, Wei Li, and Wenbin Zou. Depth-sensitive soft suppression with

rgb-d inter-modal stylization flow for domain generalization semantic segmentation, 2025. 4

[32] Zongwei Wu, Guillaume Allibert, Christophe Stolz, and Cédric Demonceaux. Depth-adapted cnn for rgb-d cameras. In *Proceedings of the Asian Conference on Computer Vision*, 2020. 3

[33] Zongwei Wu, Guillaume Allibert, Christophe Stolz, Chao Ma, and Cédric Demonceaux. Depth-adapted cnns for rgb-d semantic segmentation. *arXiv preprint arXiv:2206.03939*, 2022. 1, 3

[34] Zongwei Wu, Zhuyun Zhou, Guillaume Allibert, Christophe Stolz, Cédric Demonceaux, and Chao Ma. Transformer fusion for indoor rgb-d semantic segmentation. *Computer Vision and Image Understanding*, 249:104174, 2024. 3

[35] Bowen Yin, Xuying Zhang, Zhongyu Li, Li Liu, Ming-Ming Cheng, and Qibin Hou. Dformer: Rethinking rgbd representation learning for semantic segmentation. *arXiv preprint arXiv:2309.09668*, 2023. 3

[36] Xingxuan Zhang, Yue He, Renzhe Xu, Han Yu, Zheyan Shen, and Peng Cui. Nico++: Towards better benchmarking for domain generalization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16036–16047, 2023. 2

[37] Qiankun Zhao, Yingcai Wan, Jiqian Xu, and Lijin Fang. Cross-modal attention fusion network for rgb-d semantic segmentation. *Neurocomputing*, 548:126389, 2023. 3

[38] Xiaoqi Zhao, Lihe Zhang, Youwei Pang, Huchuan Lu, and Lei Zhang. A single stream network for robust and real-time rgb-d salient object detection. In *European conference on computer vision*, pages 646–662. Springer, 2020. 3

A. Appendix

This appendix provides supporting analyses that expand on the main paper's quantitative results. Specifically, A.1 introduces a controlled stress test of modality usage in the toy setting. By making the label depend on digit identity and colour, it prevents a depth-only shortcut and checks that RGB-D models do not collapse onto depth when RGB is necessary. A.2 then provides a class-resolved view of the NICO++ results via row-normalised confusion matrices, highlighting which structured confusions dominate under contextual shift and how clean versus corrupted pseudo-depth changes these patterns. A.3 complements this with a small set of deterministic input-level examples that illustrate the same idea, that depth helps and noise removes the benefit, at the level of individual predictions. Finally, A.4 reports the full per-run ”seen”/”unseen” accuracies underlying the aggregated metrics in the main paper to support transparency and reproducibility.

Unless noted otherwise, all appendix results follow the same reporting conventions as the main paper (mean ± sample standard deviation over three runs) and use the same data splits, models, and evaluation protocols.

A.1. Controlled experiment with coloured digits and coloured backgrounds

To further assess whether our RGB-D classifier might be relying disproportionately on depth in the toy experiment, we designed a second controlled experiment using a more challenging version of our custom MNIST dataset. In this variant, both the digit and the background are colour-coded, and the class label depends not only on the digit identity but also on the digit's colour. For example, a lime ”3” and an orange ”3” correspond to different classes. Consequently, even though the shape of the digit remains visible in the depth map, the depth channel alone is not informative enough to determine the correct class. Successful classification requires the model to combine digit shape (available in depth) with digit colour (available only in RGB). This construction explicitly prevents any shortcut solution based solely on depth.

In this dataset, digits appear in two colours, lime or orange, and are placed against red, green, blue, or black backgrounds, producing eight combinations (lime/orange × red/green/blue/black). Training uses lime/orange digits only on red and green backgrounds. Validation is balanced across digit colours and red/green backgrounds, without exposure to blue backgrounds, and testing is performed on unseen digit instances evaluated on multiple backgrounds. As in the main toy setup, we also construct a matched black-background control. Figure 4 shows representative samples from this coloured-digit MNIST variant. All splits are non-overlapping. Depth maps are constructed identically to the main toy experiment, encoding the digit silhouette with a constant depth value and a uniform background plane.

The same late fusion RGB-D model and hyperparameters used in the main toy experiment were applied here. Because digit identity is entangled with both its colour and its depth boundary structure, the depth channel alone is not sufficient to solve the classification task. Therefore, this experiment tests whether the RGB-D model is genuinely using multi-modal information or whether it collapses onto a depth-only shortcut.

For this experiment, as with the others, the reported accuracy for each test condition (red, green, blue, black) reflects how well the model can classify digit + digit-colour combinations aggregated over the corresponding background colours. That is, each score represents classification performance for a given digit-colour combination averaged across multiple background contexts.

Results averaged over three runs are shown in Table 3. As expected, performance on the coloured-digit experiment is substantially lower than in the simpler background-only setup due to the increased task difficulty. Importantly, the model does not exhibit high accuracy on any of the coloured test sets, demonstrating that it does not rely purely on depth cues. Instead, the drops confirm that classification requires integrating both modalities, and the RGB-D model does not exploit depth as an isolated shortcut. Performance on the black-background control is again higher and more stable, confirming that the depth modality does not dominate the model's decision-making when colour information is essential for solving the task.

A.2. Confusion matrix analysis across modalities and depth corruption

To better understand how estimated depth affects recognition under domain shift, we report row-normalised confusion matrices aggregated across three runs (mean ± sample standard deviation) for each backbone, for both the seen and unseen contexts test splits, and for RGB-D under progressively corrupted depth (no-, low-, medium-, and pure-noise). Because each row is normalised, diagonal entries directly reflect per-class recall, while off-diagonal mass indicates systematic confusions between specific class pairs. For each run, confusion matrices are first row-normalised to obtain per-class conditional probabilities and are then averaged across runs. Reported sample standard deviations are computed across these normalised matrices.

Table 4 summarises the macro recall implied by the confusion-matrix diagonals on the unseen contexts split across backbones and depth-corruption conditions.

Across backbones, the RGB-only models exhibit a clear structural change from the seen to the unseen split. Matrices become less diagonal, and probability mass concentrates into a small number of semantically plausible confusions (e.g., within vehicle-like categories and within animal-like

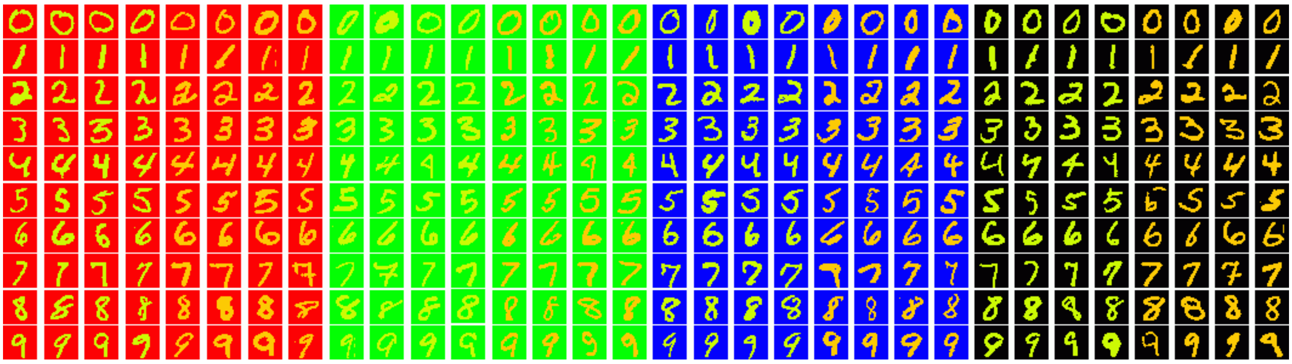


Figure 4. **Example images from the coloured-digit MNIST variant with coloured backgrounds.** Digits appear in two foreground colours (lime and orange) and are placed on coloured backgrounds. The in-distribution (ID) training data consists of images with *red* and *green* backgrounds, containing both lime and orange digits. The out-of-distribution (OOD) setting corresponds to the *blue* background, which is never observed during training. A separate *black* background is used as a control condition, in which no colour-based shortcut is available.

Table 3. **Results for the coloured-digit MNIST experiment (mean ± sample standard deviation (in %) over three runs).** This variant requires combining RGB and depth information, as neither modality alone suffices to identify the label. **Bold** values indicate the results on the *Blue Test* set, which represents the unseen background out-of-distribution (OOD) condition.

	Coloured-Digit Experiment					**Black Background Control**		
	Train / Val		**Test**			**Train / Val**		**Test**
	Train	**Val**	**Red (ID)**	**Green (ID)**	**Blue (OOD)**	**Train**	**Val**	**Black**
RGB-D	100.00 ± 0.00	58.83 ± 2.31	51.67 ± 8.50	53.67 ± 7.51	**50.67 ± 10.97**	99.67 ± 0.58	76.17 ± 4.19	73.33 ± 1.15

categories). When a clean estimated depth is added via early fusion, this concentration is reduced on the unseen split. Several classes recover diagonal mass, and the dominant off-diagonal confusions weaken, consistent with depth contributing complementary shape and layout cues that are less sensitive to appearance shifts.

The depth-quality ablation provides an interpretable ”input-level intervention” on these error patterns. Under low and medium depth noise, the confusion matrices progressively revert toward the RGB-only structure: diagonal gains shrink, and the dominant confusion pairs re-emerge. Under pure-noise depth, improvements largely vanish (and in some cases reverse), indicating that the observed robustness gains are not explained by adding an extra channel alone, but depend on geometric structure in the depth signal. In this sense, the confusion matrices provide a qualitative, class-resolved analysis of model behaviour that complements the aggregate depth-corruption results reported in the main paper.

Figure 5, 6, 7, and 8 show the full set of aggregated confusion matrices for each backbone and condition. We also report class-level analyses for the four models, from highly lightweight (ShuffleNet, MobileNet) to higher-capacity architectures (ResNet-18, EfficientNet).

A.2.1. EfficientNet: class-level confusion analysis

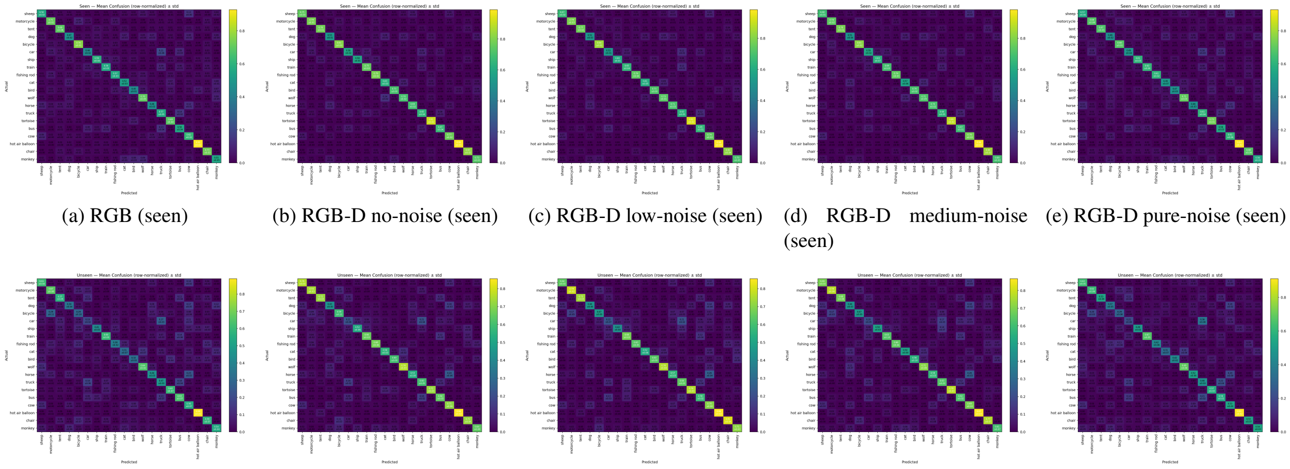
In the RGB-only setting, EfficientNet exhibits a substantial drop in macro recall from the seen to the unseen contexts split (0.606 to 0.502), indicating sensitivity to contextual domain shift. This degradation is dominated by a small number of structured semantic confusions rather than being uniformly distributed across classes. In particular, vehicle-related classes (e.g., car → truck, bicycle → motorcycle) and animal-related classes (e.g., horse → cow, dog → cow) account for a large fraction of the off-diagonal mass on the unseen split.

When a clean estimated depth is introduced via early fusion, unseen macro recall increases markedly to 0.633. This improvement is accompanied by consistent gains in per-class recall for geometry-relevant categories such as bicycle, cat, horse, chair, and motorcycle, as well as by a reduction in the dominant confusion pairs observed in the RGB baseline. Importantly, these changes are stable across runs, as indicated by low standard deviations, suggesting that depth contributes reliable complementary information rather than acting as a stochastic regulariser.

As depth quality degrades, this benefit progressively diminishes. Under low- and medium-noise depth, macro recall on the unseen split remains above the RGB baseline,

Backbone	RGB	RGB-D (Clean)	RGB-D (Low)	RGB-D (Medium)	RGB-D (Pure)
ResNet-18	0.521	0.624	0.624	0.631	0.511
EfficientNet	0.502	0.633	0.628	0.618	0.514
MobileNet	0.417	0.592	0.583	0.565	0.424
ShuffleNet	0.424	0.554	0.544	0.543	0.427

Table 4. **Macro recall (mean of the diagonal entries in the row-normalised confusion matrices) on the unseen contexts split, aggregated over three runs, for RGB and RGB-D under progressively corrupted noisy depth.**



(a) RGB (seen)

(b) RGB-D no-noise (seen)

(c) RGB-D low-noise (seen)

(d) RGB-D medium-noise (seen)

(e) RGB-D pure-noise (seen)

(f) RGB (unseen)

(g) RGB-D no-noise (unseen)

(h) RGB-D low-noise (unseen)

(i) RGB-D medium-noise (unseen)

(j) RGB-D pure-noise (unseen)

Figure 5. **Row-normalised confusion matrices (mean ± std over three runs) for EfficientNet across RGB and RGB-D depth-corruption conditions on seen and unseen contexts.**

but several previously suppressed confusions (e.g., horse → cow, dog → cow) re-emerge. In the pure-noise condition, the advantage of RGB-D largely disappears. Unseen macro recall (0.514) approaches that of the RGB-only model, and the confusion structure closely matches that observed without depth.

A.2.2. MobileNet: class-level confusion analysis

MobileNet exhibits substantially stronger sensitivity to contextual domain shift than higher-capacity backbones. In the RGB-only setting, macro recall drops from 0.520 on the seen contexts split to 0.417 on the unseen contexts split, with several classes collapsing to very low recall (e.g., car, dog, cat). This degradation is dominated by a small number of structured semantic confusions, most notably within vehicle classes (car ↔ truck, truck ↔ bus) and within animal classes (horse → cow, dog → sheep).

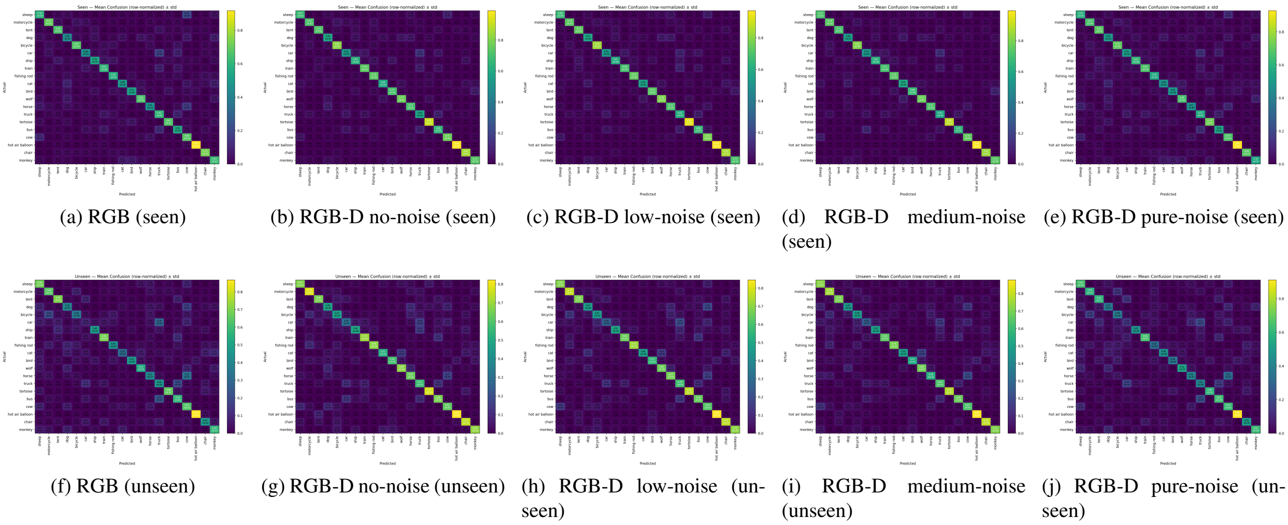
Introducing clean estimated depth via early fusion yields a large recovery in unseen performance, increasing macro recall to 0.592. This improvement is driven by pronounced gains in geometry-relevant categories such as bicycle, chair, motorcycle, and tortoise, as well as by a reduction in the dominance of the most frequent confusion pairs. The magnitude of this gain is notably larger than for EfficientNet, suggesting that depth cues are particularly beneficial for lower-capacity models that are more prone to shortcut learning.

As depth quality is progressively degraded, the benefit diminishes but does not vanish immediately. Under low- and medium-noise depth, unseen macro recall remains well above the RGB baseline, although several dominant confusions re-emerge with increasing probability. In the pure-noise condition, the advantage of RGB-D disappears almost entirely. Unseen macro recall (0.424) closely matches the RGB-only result, and the confusion structure becomes nearly identical.

A.2.3. ResNet-18: class-level confusion analysis

In the RGB-only setting, ResNet-18 exhibits a moderate degradation under contextual domain shift, with macro recall decreasing from 0.610 on the seen split to 0.521 on the unseen split. The most prominent failure modes involve vehicle classes (e.g., car → truck, bus ↔ truck), animal classes (e.g., horse → cow), and visually similar object pairs such as bicycle → motorcycle.

Adding clean estimated depth via early fusion increases



(a) RGB (seen)

(b) RGB-D no-noise (seen)

(c) RGB-D low-noise (seen)

(d) RGB-D medium-noise (seen)

(e) RGB-D pure-noise (seen)

(f) RGB (unseen)

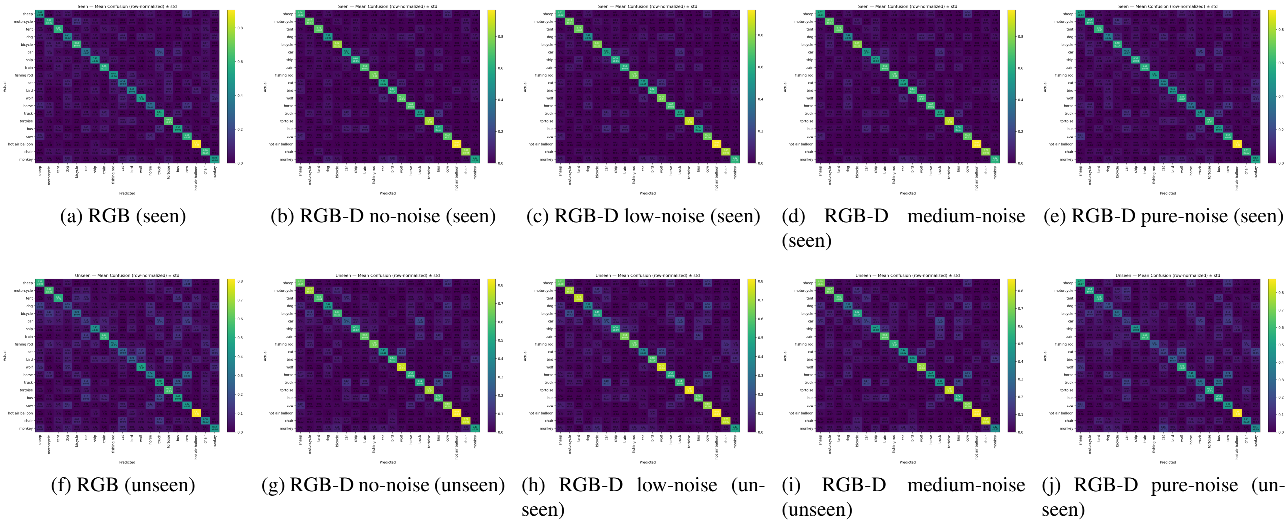
(g) RGB-D no-noise (unseen)

(h) RGB-D low-noise (unseen)

(i) RGB-D medium-noise (unseen)

(j) RGB-D pure-noise (unseen)

Figure 6. **Row-normalised confusion matrices (mean ± std over three runs) for ResNet-18 across RGB and RGB-D depth-corruption conditions on seen and unseen contexts.**



(a) RGB (seen)

(b) RGB-D no-noise (seen)

(c) RGB-D low-noise (seen)

(d) RGB-D medium-noise (seen)

(e) RGB-D pure-noise (seen)

(f) RGB (unseen)

(g) RGB-D no-noise (unseen)

(h) RGB-D low-noise (unseen)

(i) RGB-D medium-noise (unseen)

(j) RGB-D pure-noise (unseen)

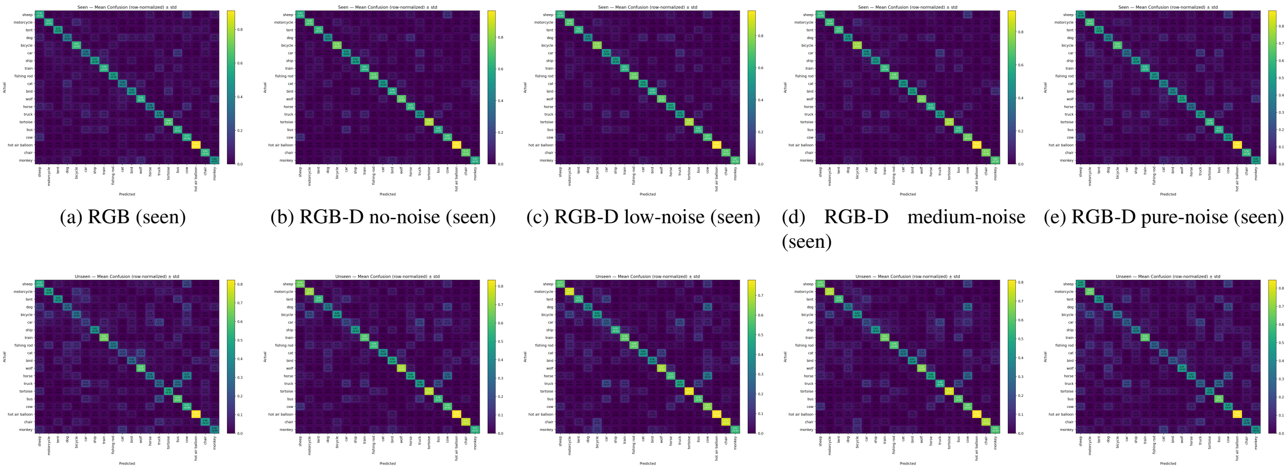
Figure 7. **Row-normalised confusion matrices (mean ± std over three runs) for MobileNet across RGB and RGB-D depth-corruption conditions on seen and unseen contexts.**

unseen macro recall to 0.624. This improvement is accompanied by consistent per-class gains for categories with distinctive geometric structure, including car, bicycle, cat, and chair, as well as by a reduction in the dominance of the most frequent confusion pairs. Compared to MobileNet, the magnitude of the gain is smaller, but the effect is highly stable across runs, indicating that ResNet-18 is less brittle while still benefiting from complementary depth cues.

As depth quality degrades, the advantage of RGB-D diminishes gradually. Under low- and medium-noise depth, unseen macro recall remains close to the no-noise condition, although several semantic confusions re-emerge. In the pure-noise condition, the benefit of depth disappears. Unseen macro recall (0.511) closely matches the RGB-only baseline, and the confusion structure reverts to that observed without depth.

A.2.4. ShuffleNet: class-level confusion analysis

ShuffleNet exhibits the strongest sensitivity to contextual domain shift among the evaluated backbones. In the RGB-only setting, macro recall decreases from 0.531 on the seen split to 0.424 on the unseen split, with several classes collapsing to very low recall. The dominant failure modes again correspond to highly structured semantic confusions,



(a) RGB (seen) (b) RGB-D no-noise (seen) (c) RGB-D low-noise (seen) (d) RGB-D medium-noise (seen) (e) RGB-D pure-noise (seen)

(f) RGB (unseen) (g) RGB-D no-noise (unseen) (h) RGB-D low-noise (unseen) (i) RGB-D medium-noise (unseen) (j) RGB-D pure-noise (unseen)

Figure 8. **Row-normalised confusion matrices (mean ± std over three runs) for ShuffleNet across RGB and RGB-D depth-corruption conditions on seen and unseen contexts.**

most prominently within vehicle classes (truck ↔ bus, car → truck), animal classes (horse → cow), and visually similar object pairs such as bicycle → motorcycle.

Adding clean estimated depth via early fusion increases unseen macro recall to 0.554. Although the absolute performance remains lower than for higher-capacity models, this improvement is driven by consistent per-class gains in geometry-sensitive categories such as bicycle, chair, motorcycle, and tortoise, as well as by a reduction in the dominance of the most frequent confusion pairs.

As depth quality degrades, the benefit of RGB-D diminishes steadily. Under low- and medium-noise depth, unseen macro recall remains above the RGB baseline, but dominant confusions progressively re-emerge and variability across runs increases. In the pure-noise condition, the advantage of RGB-D disappears almost entirely: unseen macro recall (0.427) closely matches the RGB-only result, and the confusion structure reverts to that observed without depth.

A.2.4.1 Summary across backbones

Table 5 summarises the most frequent unseen-context confusion pairs aggregated across backbones, illustrating how clean depth consistently reduces the dominance of the same semantic shortcuts observed in the RGB-only setting.

Across all evaluated architectures, the confusion matrix analyses reveal a consistent and highly structured pattern of failure under contextual domain shift. In the RGB-only setting, performance degradation on the unseen split is dominated by a small number of semantically meaningful confusions, most prominently within vehicle classes (e.g., car ↔ truck, truck ↔ bus), animal classes (e.g., horse → cow), and visually similar object pairs such as bicycle → motorcycle. Introducing clean estimated depth via early fusion consistently reduces the dominance of these confusions and increases per-class recall for geometry-relevant categories, with the magnitude of the gain scaling inversely with model capacity (largest for MobileNet and ShuffleNet, smaller but more stable for ResNet-18 and EfficientNet). As depth quality is progressively degraded, these benefits diminish in a smooth and interpretable manner, and under pure-noise depth, the confusion structure reliably reverts to that of the RGB-only baseline. Taken together, these results demonstrate that the observed robustness gains arise from meaningful geometric information in the depth signal rather than from the addition of extra input channels, and that this effect generalises across model architectures.

A.3. Qualitative input-level analysis on NICO++

To complement the quantitative results and the class-resolved confusion-matrix analysis, we provide qualitative input-level examples illustrating how estimated depth affects individual predictions under domain shift. Figure 9 shows three representative samples from the unseen contexts split of NICO++, covering an animal category, a vehicle category, and a geometry-dominated object.

In all three cases, the RGB-only ResNet-18 model fails under the unseen context, producing semantically plausible but incorrect predictions (e.g., animal-to-animal or vehicle-to-vehicle confusions). When a clean estimated depth channel is added via early fusion, the model recovers the correct class. The corresponding predicted depth maps exhibit stable geometric structure that is largely invariant to background appearance, suggesting that shape and spatial cues

Condition (unseen)	**Confusion pair**	**Mean ± std (across backbones)**
RGB	horse → cow	0.31 ± 0.05
	car → truck	0.22 ± 0.05
	truck → bus	0.21 ± 0.05
	bicycle → motorcycle	0.19 ± 0.04
	dog → cow	0.18 ± 0.04
RGB-D (Clean)	horse → cow	0.25 ± 0.04
	car → truck	0.17 ± 0.03
	truck → bus	0.15 ± 0.03
	bicycle → motorcycle	0.14 ± 0.03
	dog → cow	0.14 ± 0.03

Table 5. **Top five unseen-context confusion pairs aggregated across backbones for RGB and RGB-D (no noise).** Values report the mean ± sample standard deviation of the confusion probability across ResNet-18, EfficientNet, MobileNet, and ShuffleNet.

help disambiguate these categories. When the depth channel is replaced with pure noise, this benefit disappears, and the model again produces incorrect predictions, mirroring the degradation observed in the depth-quality ablation experiments.

The selected examples are chosen deterministically from the unseen split by lexicographic filename order, subject to the condition that the RGB-only model fails while the RGB-D (no noise) model succeeds. To illustrate that this behaviour is not specific to a single architecture, predictions from the compact MobileNet model with clean depth input are also shown. MobileNet correctly classifies all three examples, consistent with the cross-backbone trends reported in the quantitative results.

Together with the confusion-matrix analysis, these examples provide an interpretable, input-level view of the role played by pseudo-depth. They illustrate that the observed robustness gains arise from resolving specific, structured confusions through geometric information, rather than from the mere presence of an additional input channel.

A.4. Per-run seen and unseen test accuracies on NICO++

This subsection reports the full per-run test accuracies for all NICO++ experiments discussed in the main paper. For each model and experimental condition (RGB, RGB-D with clean depth, and RGB-D under varying levels of depth corruption), we report accuracy on both ”seen” (ID) and ”unseen” (OOD) contexts for each individual run, together with the mean and sample standard deviation across runs. Table 6, 7, 8, 9, and 10 report these results. These tables are provided to ensure transparency and reproducibility, and to allow readers to directly inspect the absolute performance values underlying the aggregate OOD gap metrics reported in the main text.

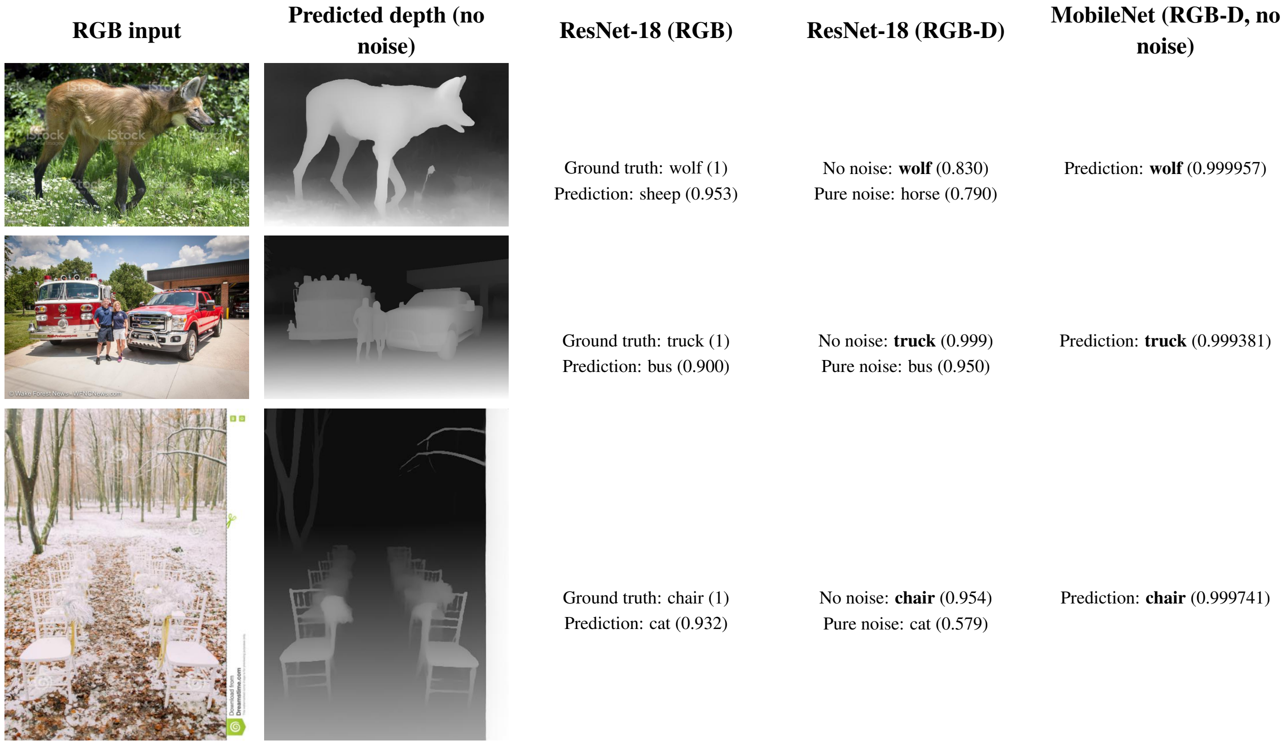

RGB input
Predicted depth (no noise)
ResNet-18 (RGB)
ResNet-18 (RGB-D)
MobileNet (RGB-D, no noise)
Ground truth: wolf (1)
Prediction: sheep (0.953)
No noise: **wolf** (0.830)
Pure noise: horse (0.790)
Prediction: **wolf** (0.999957)
Ground truth: truck (1)
Prediction: bus (0.900)
No noise: **truck** (0.999)
Pure noise: bus (0.950)
Prediction: **truck** (0.999381)
Ground truth: chair (1)
Prediction: cat (0.932)
No noise: **chair** (0.954)
Pure noise: cat (0.579)
Prediction: **chair** (0.999741)

Figure 9. **Qualitative input-level examples on unseen contexts (NICO++).** Each row shows the RGB input, its predicted (no noise) depth map, and the corresponding predictions. For ResNet-18, adding clean depth (RGB-D no noise) corrects the RGB-only error in all three cases, while replacing depth with pure noise removes this benefit. MobileNet (RGB-D, no noise) also predicts the correct class for all three examples, illustrating that the same qualitative behaviour occurs in a compact architecture. Correct predictions are shown in bold. Confidence values denote the predicted-class softmax probability.

Table 6. **RGB baseline accuracy (in %) on NICO++ (Seen/Unseen contexts).** Per-run results are reported for each model, followed by the mean ± sample standard deviation across the three runs.

Model	**Run 1**		**Run 2**		**Run 3**		**Mean ± Std**	
	Seen	Unseen	Seen	Unseen	Seen	Unseen	Seen	Unseen
ResNet-18	61.83	51.50	62.08	50.67	62.08	51.83	62.00 ± 0.14	51.33 ± 0.60
EfficientNet	61.08	50.83	59.50	49.33	60.92	50.67	60.50 ± 0.87	50.28 ± 0.82
MobileNet	53.75	42.33	51.92	41.17	51.17	41.00	52.28 ± 1.33	41.50 ± 0.72
ShuffleNet	53.50	42.83	52.50	41.83	53.67	43.00	53.22 ± 0.63	42.55 ± 0.63

Table 7. **RGB-D (clean depth) accuracy (in %) on NICO++ (Seen/Unseen contexts).** Per-run results are reported for each model, followed by the mean ± sample standard deviation across the three runs.

Model	**Run 1**		**Run 2**		**Run 3**		**Mean ± Std**	
	Seen	Unseen	Seen	Unseen	Seen	Unseen	Seen	Unseen
ResNet-18	72.50	64.67	71.92	62.17	72.08	65.17	72.17 ± 0.30	64.00 ± 1.61
EfficientNet	71.58	62.83	72.17	63.00	72.17	63.33	71.97 ± 0.34	63.05 ± 0.25
MobileNet	67.42	59.83	67.00	58.67	66.42	57.83	66.95 ± 0.50	58.78 ± 1.00
ShuffleNet	63.17	56.83	63.00	54.17	63.92	54.50	63.36 ± 0.49	55.17 ± 1.45

Table 8. **RGB-D accuracy (in %) on NICO++ with low-noise depth corruption (Seen/Unseen contexts).** Per-run results are reported for each model, followed by the mean ± sample standard deviation across the three runs.

Model	**Run 1**		**Run 2**		**Run 3**		**Mean ± Std**	
	Seen	Unseen	Seen	Unseen	Seen	Unseen	Seen	Unseen
ResNet-18	70.92	62.67	73.00	64.00	71.33	60.50	71.75 ± 1.10	62.39 ± 1.77
EfficientNet	70.92	62.83	71.67	62.50	72.08	63.17	71.56 ± 0.59	62.83 ± 0.34
MobileNet	67.17	57.83	68.25	58.00	67.17	59.00	67.53 ± 0.62	58.28 ± 0.63
ShuffleNet	65.25	53.83	65.33	54.17	65.25	53.83	65.28 ± 0.05	53.94 ± 0.20

Table 9. **RGB-D accuracy (in %) on NICO++ with medium-noise depth corruption (Seen/Unseen contexts).** Per-run results are reported for each model, followed by the mean ± sample standard deviation across the three runs.

Model	**Run 1**		**Run 2**		**Run 3**		**Mean ± Std**	
	Seen	Unseen	Seen	Unseen	Seen	Unseen	Seen	Unseen
ResNet-18	72.00	63.00	72.17	63.50	71.67	62.83	71.95 ± 0.25	63.11 ± 0.35
EfficientNet	71.33	59.83	71.92	62.33	72.25	63.17	71.83 ± 0.47	61.78 ± 1.74
MobileNet	67.50	55.67	65.75	56.67	66.00	57.17	66.42 ± 0.95	56.50 ± 0.76
ShuffleNet	63.00	53.67	63.67	54.50	63.08	54.83	63.25 ± 0.37	54.33 ± 0.60

Table 10. **RGB-D accuracy (in %) on NICO++ with pure-noise depth input (Seen/Unseen contexts).** Per-run results are reported for each model, followed by the mean ± sample standard deviation across the three runs.

Model	**Run 1**		**Run 2**		**Run 3**		**Mean ± Std**	
	Seen	Unseen	Seen	Unseen	Seen	Unseen	Seen	Unseen
ResNet-18	61.58	50.17	62.00	51.00	61.17	52.00	61.58 ± 0.42	51.06 ± 0.92
EfficientNet	60.58	50.33	62.42	54.33	59.83	49.50	60.94 ± 1.33	51.39 ± 2.58
MobileNet	51.92	42.50	53.42	41.50	54.08	43.17	53.14 ± 1.11	42.39 ± 0.84
ShuffleNet	51.25	44.00	50.67	43.33	51.33	40.83	51.08 ± 0.36	42.72 ± 1.67

Bibliography

[1] Rana Muhammad Adnan Ikram et al. “Novel evolutionary-optimized neural network for predicting landslide susceptibility”. In: *Environment, Development and Sustainability* 26.7 (2024), pp. 17687–17719.

[2] Martin Arjovsky et al. “Invariant risk minimization”. In: *arXiv preprint arXiv:1907.02893* (2019).

[3] James Bergstra and Yoshua Bengio. “Random search for hyper-parameter optimization”. In: *The journal of machine learning research* 13.1 (2012), pp. 281–305.

[4] Hao Chen, Youfu Li, and Dan Su. “RGB-D saliency detection by multi-stream late fusion network”. In: *International conference on computer vision systems*. Springer. 2017, pp. 459–468.

[5] Xiaotian Chen, Xuejin Chen, and Zheng-Jun Zha. “Structure-aware residual pyramid network for monocular depth estimation”. In: *arXiv preprint arXiv:1907.06023* (2019).

[6] *CNN | Introduction to Padding - GeeksforGeeks — geeksforgeeks.org*. https://www.geeksforgeeks.org/machine-learning/cnn-introduction-to-padding/. [Accessed 07-11-2025].

[7] Navneet Dalal and Bill Triggs. “Histograms of oriented gradients for human detection”. In: *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR’05)*. Vol. 1. Ieee. 2005, pp. 886–893.

[8] Siva Krishna Dasari, Koteswara Rao Chintada, and Muralidhar Patruni. “Flue-cured tobacco leaves classification: A generalized approach using deep convolutional neural networks”. In: *Cognitive science and artificial intelligence: Advances and applications*. Springer, 2017, pp. 13–21.

[9] Jia Deng et al. “Imagenet: A large-scale hierarchical image database”. In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, pp. 248–255.

[10] Li Deng. “The MNIST Database of Handwritten Digit Images for Machine Learning Research [Best of the Web]”. In: *IEEE Signal Processing Magazine* 29.6 (2012), pp. 141–142. DOI: 10.1109/MSP.2012.2211477.

[11] Alexey Dosovitskiy et al. “An image is worth 16x16 words: Transformers for image recognition at scale”. In: *arXiv preprint arXiv:2010.11929* (2020).

[12] Vincent Dumoulin and Francesco Visin. “A guide to convolution arithmetic for deep learning”. In: *arXiv preprint arXiv:1603.07285* (2016).

[13] David Eigen, Christian Puhrsch, and Rob Fergus. “Depth map prediction from a single image using a multi-scale deep network”. In: *Advances in neural information processing systems* 27 (2014).

[14] Andreas Eitel et al. “Multimodal deep learning for robust RGB-D object recognition”. In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2015, pp. 681–687.

[15] Huan Fu et al. “Deep ordinal regression network for monocular depth estimation”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 2002–2011.

[16] Keren Fu et al. “Siamese network for RGB-D salient object detection and beyond”. In: *IEEE transactions on pattern analysis and machine intelligence* 44.9 (2021), pp. 5541–5559.

[17] Robert Geirhos et al. “Shortcut learning in deep neural networks”. In: *Nature Machine Intelligence* 2.11 (2020), pp. 665–673.

[18] Anirudha Ghosh et al. “Fundamental concepts of convolutional neural network”. In: *Recent trends and advances in artificial intelligence and Internet of Things*. Springer, 2019, pp. 519–567.

[19] Xavier Glorot and Yoshua Bengio. “Understanding the difficulty of training deep feedforward neural networks”. In: *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings. 2010, pp. 249–256.

[20] Ian Goodfellow et al. *Deep learning*. Vol. 1. 2. MIT press Cambridge, 2016.

[21] Saurabh Gupta et al. “Learning rich features from RGB-D images for object detection and segmentation”. In: *European conference on computer vision*. Springer. 2014, pp. 345–360.

[22] Saad Hikmat Haji and Adnan Mohsin Abdulazeez. “Comparison of optimization techniques based on gradient descent algorithm: A review”. In: *PalArch’s Journal of Archaeology of Egypt/Egyptology* 18.4 (2021), pp. 2715–2743.

[23] Caner Hazirbas et al. “Fusenet: Incorporating depth into semantic segmentation via fusion-based cnn architecture”. In: *Asian conference on computer vision*. Springer. 2016, pp. 213–228.

[24] Kaiming He et al. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[25] Kaiming He et al. “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification”. In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 1026–1034.

[26] Xiankang He et al. *Distill Any Depth: Distillation Creates a Stronger Monocular Depth Estimator*. 2025. arXiv: 2502.19204 [cs.CV]. URL: https://arxiv.org/abs/2502.19204.

[27] Yue He, Zheyan Shen, and Peng Cui. “Towards non-iid image classification: A dataset and baselines”. In: *Pattern Recognition* 110 (2021), p. 107383.

[28] Kurt Hornik. “Approximation capabilities of multilayer feedforward networks”. In: *Neural networks* 4.2 (1991), pp. 251–257.

[29] Andrew G Howard et al. “Mobilenets: Efficient convolutional neural networks for mobile vision applications”. In: *arXiv preprint arXiv:1704.04861* (2017).

[30] Nianchang Huang et al. “Middle-Level Feature Fusion for Lightweight RGB-D Salient Object Detection”. In: *IEEE Transactions on Image Processing* 31 (2022), pp. 6621–6634. ISSN: 1941-0042. DOI: 10.1109/tip.2022.3214092. URL: http://dx.doi.org/10.1109/TIP.2022.3214092.

[31] *Image Classification: Applications & Best Practices in 2026 — research.aimultiple.com*. https://research.aimultiple.com/image-classification/. [Accessed 02-01-2026].

[32] Abhishek Jain. *All about convolutions, kernels, features in CNN*. Feb. 2024. URL: https://medium.com/@abhishekjainindore24/all-about-convolutions-kernels-features-in-cnn-c656616390a1.

[33] Jindong Jiang et al. “Rednet: Residual encoder-decoder network for indoor rgb-d semantic segmentation”. In: *arXiv preprint arXiv:1806.01054* (2018).

[34] Reza Kalantar. *Receptive Field in Deep Convolutional Networks — rekalantar*. https://medium.com/@rekalantar/receptive-fields-in-deep-convolutional-networks-43871d2ef2e9. [Accessed 07-11-2025].

[35] Diederik P Kingma. “Adam: A method for stochastic optimization”. In: *arXiv preprint arXiv:1412.6980* (2014).

[36] Pang Wei Koh et al. “Wilds: A benchmark of in-the-wild distribution shifts”. In: *International conference on machine learning*. PMLR. 2021, pp. 5637–5664.

[37] Alex Krizhevsky, Geoffrey Hinton, et al. “Learning multiple layers of features from tiny images”. In: (2009).

[38] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “Imagenet classification with deep convolutional neural networks”. In: *Advances in neural information processing systems* 25 (2012).

[39] Iro Laina et al. “Deeper depth prediction with fully convolutional residual networks”. In: *2016 Fourth international conference on 3D vision (3DV)*. IEEE. 2016, pp. 239–248.

[40] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. “Deep learning”. In: *nature* 521.7553 (2015), pp. 436–444.

[41] Yann LeCun et al. “Gradient-based learning applied to document recognition”. In: *Proceedings of the IEEE* 86.11 (2002), pp. 2278–2324.

[42] Jin Han Lee et al. “From big to small: Multi-scale local planar guidance for monocular depth estimation”. In: *arXiv preprint arXiv:1907.10326* (2019).

[43] Haoang Li et al. “Pg-slam: Photo-realistic and geometry-aware rgb-d slam in dynamic environments”. In: *IEEE Transactions on Robotics* (2025).

[44] Min Lin, Qiang Chen, and Shuicheng Yan. “Network in network”. In: *arXiv preprint arXiv:1312.4400* (2013).

[45] Ilya Loshchilov and Frank Hutter. *Decoupled Weight Decay Regularization*. 2019. arXiv: 1711.05101 [cs.LG]. URL: https://arxiv.org/abs/1711.05101.

[46] David G Lowe. “Distinctive image features from scale-invariant keypoints”. In: *International journal of computer vision* 60.2 (2004), pp. 91–110.

[47] Jiajun Ma et al. “Geometric-aware RGB-D representation learning for hand–object reconstruction”. In: *Expert Systems with Applications* 257 (2024), p. 124995.

[48] Ningning Ma et al. “Shufflenet v2: Practical guidelines for efficient cnn architecture design”. In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 116–131.

[49] *MNIST database - Wikipedia — en.wikipedia.org*. https://en.wikipedia.org/wiki/MNIST_database. [Accessed 07-11-2025].

[50] Vinod Nair and Geoffrey E Hinton. “Rectified linear units improve restricted boltzmann machines”. In: *Proceedings of the 27th international conference on machine learning (ICML-10)*. 2010, pp. 807–814.

[51] Neri Van Otten. *Weight Decay In Machine Learning And Deep Learning Explained & How To Tutorial*. https://spotintelligence.com/2024/05/02/weight-decay/. [Accessed 07-11-2025].

[52] Rukshan Pramoditha. *Overview of a Neural Network’s Learning Process — medium.com*. https://medium.com/data-science-365/overview-of-a-neural-networks-learning-process-61690a502fa. [Accessed 07-11-2025].

[53] Joaquin Quiñonero-Candela et al. *Dataset shift in machine learning*. Mit Press, 2022.

[54] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. “Vision transformers for dense prediction”. In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, pp. 12179–12188.

[55] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. “Learning representations by back-propagating errors”. In: *nature* 323.6088 (1986), pp. 533–536.

[56] Mark Sandler et al. “Mobilenetv2: Inverted residuals and linear bottlenecks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 4510–4520.

[57] Connor Shorten and Taghi M Khoshgoftaar. “A survey on image data augmentation for deep learning”. In: *Journal of big data* 6.1 (2019), pp. 1–48.

[58] Karen Simonyan and Andrew Zisserman. “Very deep convolutional networks for large-scale image recognition”. In: *arXiv preprint arXiv:1409.1556* (2014).

[59] Ambarish Singh. *Deep Dive into the World of CNNs — ai.plainenglish.io*. https://ai.plainenglish.io/deep-dive-into-the-world-of-cnns-8cf22cd84e7. [Accessed 07-11-2025].

[60] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. “Practical bayesian optimization of machine learning algorithms”. In: *Advances in neural information processing systems* 25 (2012).

[61] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. “Sun rgb-d: A rgb-d scene understanding benchmark suite”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 567–576.

[62] Jost Tobias Springenberg et al. “Striving for simplicity: The all convolutional net”. In: *arXiv preprint arXiv:1412.6806* (2014).

[63] Mingxing Tan and Quoc Le. “Efficientnet: Rethinking model scaling for convolutional neural networks”. In: *International conference on machine learning*. PMLR. 2019, pp. 6105–6114.

[64] Rohan Taori et al. “Measuring robustness to natural distribution shifts in image classification”. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 18583–18599.

[65] Hardik Uppal et al. “Two-level attention-based fusion learning for rgb-d face recognition”. In: *2020 25th international conference on pattern recognition (ICPR)*. IEEE. 2021, pp. 10120–10127.

[66] Jindong Wang et al. “Generalizing to unseen domains: A survey on domain generalization”. In: *IEEE transactions on knowledge and data engineering* 35.8 (2022), pp. 8052–8072.

[67] Weiyue Wang and Ulrich Neumann. “Depth-aware cnn for rgb-d segmentation”. In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 135–150.

[68] Bo-Wen Yin et al. *DFormerv2: Geometry Self-Attention for RGBD Semantic Segmentation*. 2025. arXiv: `2504.04701 [cs.CV]`. URL: `https://arxiv.org/abs/2504.04701`.

[69] Jiaming Zhang et al. “CMX: Cross-modal fusion for RGB-X semantic segmentation with transformers”. In: *IEEE Transactions on intelligent transportation systems* 24.12 (2023), pp. 14679–14694.

[70] Xiangyu Zhang et al. “Shufflenet: An extremely efficient convolutional neural network for mobile devices”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 6848–6856.

[71] Xingxuan Zhang et al. “Nico++: Towards better benchmarking for domain generalization”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2023, pp. 16036–16047.

[72] Li Zhong et al. “Attention-based fusion network for RGB-D semantic segmentation”. In: *Neurocomputing* 608 (2024), p. 128371.

[73] Hao Zhou et al. “RGB-D co-attention network for semantic segmentation”. In: *Proceedings of the Asian conference on computer vision*. 2020.

Source Code Example

We show in this Appendix a snippet of the source code. For the full code, see the author's GitHub repository. ***This code implements synchronised data augmentation and loading for RGB-D inputs.*** *It ensures that both RGB images and their corresponding depth maps undergo identical random transformations, such as cropping and flipping, so they remain pixel-aligned.*

```
# Pre-create a RandomResizedCrop object just for parameter sampling
rrc = transforms.RandomResizedCrop(INPUT_SIZE, scale=(0.35,1.0))

## Synchronising RandomResizedCrop
def joint_random_resized_crop(rgb, depth):
    i, j, h, w = rrc.get_params(rgb, rrc.scale, rrc.ratio)
    rgb_cropped   = F.resized_crop(rgb, i, j, h, w, size=(INPUT_SIZE, INPUT_SIZE))
    depth_cropped = F.resized_crop(depth, i, j, h, w, size=(INPUT_SIZE, INPUT_SIZE))
    return rgb_cropped, depth_cropped

## Synchronising RandomHorizontalFlip
def joint_random_hflip(rgb, depth, p=0.5):
    if random.random() < p:
        rgb   = F.hflip(rgb)
        depth = F.hflip(depth)
    return rgb, depth


## Putting it all together: a JointTransform class
class JointTrainTransform:
    def __init__(self, rgb_train_tf, depth_train_tf, rrc_scale=(0.35,1.0), p_flip=0.5):
        self.rgb_tf   = rgb_train_tf
        self.depth_tf = depth_train_tf
        self.rrc      = transforms.RandomResizedCrop(INPUT_SIZE, scale=rrc_scale)
        self.p_flip   = p_flip

    def __call__(self, rgb: Image.Image, depth: Image.Image):
        # 1) Joint RandomResizedCrop
        i, j, h, w = self.rrc.get_params(rgb, self.rrc.scale, self.rrc.ratio)
        rgb, depth = (
            F.resized_crop(rgb, i,j,h,w, size=(INPUT_SIZE, INPUT_SIZE)),
            F.resized_crop(depth, i,j,h,w, size=(INPUT_SIZE, INPUT_SIZE))
        )
        # 2) Joint RandomHorizontalFlip
        if torch.rand(1).item() < self.p_flip:
            rgb, depth = F.hflip(rgb), F.hflip(depth)
        # 3) Per-modality additional transforms
        rgb   = self.rgb_tf(rgb)
        depth = self.depth_tf(depth)
        # 4) Concat
        return torch.cat([rgb, depth], dim=0)

# Wrap them into a joint test transform
# This just runs each branch and concatenates - no random ops needed
class JointTestTransform:
    def __init__(self, rgb_tf, depth_tf):
        self.rgb_tf   = rgb_tf
        self.depth_tf = depth_tf

    def __call__(self, rgb, depth):
        rgb   = self.rgb_tf(rgb)
        depth = self.depth_tf(depth)
        return torch.cat([rgb, depth], dim=0)

class RGBDDataset(Dataset):
    def __init__(self, samples, joint_transform, rgb_root, depth_root):
        self.samples         = samples
```

```python
        self.joint_transform = joint_transform
        self.rgb_root        = rgb_root
        self.depth_root      = depth_root

    def __len__(self):
        return len(self.samples)

    def __getitem__(self, idx):
        rgb_path, label = self.samples[idx]
        # Correctly compute the path of the RGB file relative to rgb_root
        rel_path = os.path.relpath(rgb_path, self.rgb_root)
        rel_dir  = os.path.dirname(rel_path)      # e.g. "autumn/cat"
        base     = os.path.basename(rel_path)     # e.g. "autumn_000136.jpg"
        depth_name = f"depth_map_{base}"
        depth_path = os.path.join(self.depth_root, rel_dir, depth_name)
        rgb   = Image.open(rgb_path).convert("RGB")
        depth = Image.open(depth_path).convert("L")
        rgbd  = self.joint_transform(rgb, depth)
        return rgbd, label
```