Developing a Task-Agnostic Data Augmentation Method via Foreground Extraction, Object Relocation, and Generative Inpainting

Master Thesis

Master of Science in Applied Computer Science

Roza Gaisina

April 13, 2026

Supervisor:

1st: Prof. Dr. Christian Ledig
2nd: Sebastian Dörrich

Chair of Explainable Machine Learning
Faculty of Information Systems and Applied Computer Sciences
Otto-Friedrich-University Bamberg

Abstract

Understanding how image classification models utilize visual information remains a central challenge in computer vision Geirhos et al. (2018). This thesis investigates whether object-centric data augmentation can serve as a structured framework for analyzing model behavior under controlled visual manipulations. To this end, a modular augmentation pipeline, *sam2aug*, is developed that integrates segmentation, inpainting, and object-level transformations. The pipeline enables the generation of controlled datasets that isolate specific visual factors, including texture, shape, background context, and object scale.

Experiments on ImageNet-pretrained convolutional neural networks and Vision Transformers reveal consistent trends across all settings. Under standard conditions, model predictions are largely driven by texture cues. However, when structural information is preserved or emphasized, the proportion of shape-consistent predictions increases. In addition, Vision Transformer models (in particular, ViT-Base) exhibit greater stability under contextual and geometric perturbations than convolutional architectures. Additional experiments show that model predictions are sensitive to background context and object scale, with performance decreasing under distribution shifts and size reductions.

Overall, the results demonstrate that object-centric augmentation provides a flexible and interpretable framework for analyzing model behavior and offers new insights into the robustness and decision strategies of modern vision models.

Contents

List of Figures

List of Tables

List of Acronyms

AdaIN	Adaptive Instance Normalization
CNN	Convolutional Neural Network
COCO	Common Objects in Context
FFC	Fast Fourier Convolution
ILSVRC	ImageNet Large Scale Visual Recognition Challenge
LaMa	Large Mask Inpainting
NST	Neural Style Transfer
ReLU	Rectified Linear Unit
ResNet	Residual Network
RGB	Color model in which the red, green, and blue are primary colors
SAM	Segment Anything Model
VGG	Visual Geometry Group (network)
ViT	Vision Transformer
WNID	WordNet ID

Notation

Images, Masks, and Transformations

I	Input image
I_{out}	Transformed image
$\hat{I}$	Stylized image (e.g., via NST)
D	Donor (texture or style) image
D'	Donor image combined with edge map
M	Binary segmentation mask
M^{dil}	Dilated segmentation mask
α	Soft alpha mask for blending ($\alpha \in [0, 1]$)
B	Bounding box
B_{tight}	Tight bounding box derived from mask
k	Dilation kernel size
r	Relative object size ratio

Style Transfer and Feature Representations

f_c	Content feature representation
f_s	Style feature representation
$\mu(\cdot)$	Channel-wise mean of features
$\sigma(\cdot)$	Channel-wise standard deviation of features
$G_l(\cdot)$	Gram matrix of feature activations at layer l
w_l	Weight for layer l in style loss
L_{content}	Content loss
L_{style}	Style loss
L_{TV}	Total variation loss
$\lambda_c, \lambda_s, \lambda_{tv}$	Weights for loss components
$\text{AdaIN}(\cdot)$	Adaptive Instance Normalization transformation

Labels and Predictions

y	Ground-truth class label
y_{shape}	Shape label (object class)
y_{texture}	Texture label (donor class)
$\hat{y}_1$	Top-1 predicted class
$\hat{Y}_5$	Set of top-5 predicted classes
$P(y)$	Parent category of class y

Evaluation Metrics

N	Number of samples in dataset
N_{shape}	Number of shape-consistent predictions
N_{texture}	Number of texture-consistent predictions
$\mathbf{1}[\cdot]$	Indicator function (1 if condition true, 0 otherwise)
Shape Bias	Calculated shape bias
$p(c \mid x)$	Softmax output for class c given input x
$\Delta P_{\text{correct}}$	Change in softmax output for correct class

Dataset and Transformations

x	Original input image
x'	Transformed image (e.g., rescaled or relocated)
c	Class index
S	Set of scaling factors
$\text{area}(\cdot)$	Area of an object or image

1 Introduction

Deep learning and deep model architectures have played a critical role in advancing image classification (He et al., 2016; Krizhevsky et al., 2012), enabling high-performing models. Also, the availability of large annotated datasets (e.g., ImageNet (Russakovsky et al., 2015), MS COCO (Lin et al., 2014)) and their combination with effective training strategies contributed to this success (Krizhevsky et al., 2012).

One strategy to improve generalization and reduce overfitting during training is to increase the diversity of the training data by augmenting existing samples (Perez and Wang, 2017), particularly in settings with limited data or strong biases. Traditional augmentation methods operate directly in image space and include geometric transformations (e.g., rotation, scaling, and cropping) and photometric transformations (e.g., color jittering and noise injection) (Shorten and Khoshgoftaar, 2019). As summarized by Kumar et al. (2024) and Naveed et al. (2024), these approaches aim to introduce invariances to common variations in real-world data while preserving semantic labels. More advanced strategies extend this idea by combining information from multiple images, for example, through sample-mixing techniques such as Mixup and CutMix, or by using generative models to synthesize novel training samples. These methods have been shown to improve robustness and generalization across a range of tasks (Kumar et al., 2024).

Despite their effectiveness, most existing augmentation techniques share a fundamental limitation: they operate on the entire image, thereby providing only limited control over which visual factors are modified. As a result, they are well-suited for improving performance, but less suitable for analyzing model behavior under controlled conditions. In particular, when studying how models rely on specific visual cues, such as shape or texture, it is necessary to manipulate these factors independently.

This limitation is particularly relevant to shape-texture bias in image classification models. Prior work has shown that convolutional neural networks trained on ImageNet often rely strongly on local texture cues, whereas human perception is more strongly driven by global object shape (Geirhos et al., 2018; Tartaglini et al., 2022). The commonly used cue-conflict paradigm addresses this question by generating images in which shape and texture cues are deliberately placed in conflict, typically using neural style transfer (Gatys et al., 2016). While this approach has provided valuable insights, subsequent studies have highlighted limitations in stimulus generation and evaluation methodology, including that stylization is applied to the entire image rather than selectively to the object of interest (Burgert et al., 2025).

These observations underscore the need for more controlled, modular data transformation pipelines that enable independent manipulation of individual visual factors. In particular, an object-centric approach enables targeted modification of appearance while preserving other aspects of the scene. Such a framework enables systematic investigation of how models respond to changes in texture, context, and scale under controlled conditions.

In this thesis, we propose a task-agnostic object-centric data augmentation pipeline that combines foreground extraction, object relocation, and generative inpainting. The pipeline leverages modern segmentation models to isolate object regions, applies object-level transformations, and reconstructs the background using inpainting techniques. This modular design enables the creation of multiple controlled dataset variants, including shape-texture cue-conflict, background-shift, and object-rescaling datasets.

Using this framework, we conduct a series of experiments to analyze model behavior across different architectures, including convolutional neural networks and Vision Transformers. The evaluation focuses on shape-texture bias and robustness under contextual and geometric transformations. By comparing model predictions across systematically varied conditions, the thesis aims to provide a more fine-grained understanding of how architectural design and input transformations influence visual decision-making in modern image classification systems.

Overall, this work contributes both a methodological framework for controlled object-centric data augmentation and an empirical analysis of model behavior under structured transformations. By bridging data augmentation, generative image manipulation, and bias analysis, the thesis seeks to advance understanding of how deep neural networks use visual cues and how these behaviors can be studied in a controlled, interpretable manner.

This work is structured as follows. Section 2 introduces the theoretical background relevant to this work, including modern image classification architectures, data augmentation techniques, style transfer, and the core components of the proposed approach, namely image segmentation and inpainting. Section 3 reviews related work on shape-texture bias in deep neural networks, discusses existing cue-conflict evaluation paradigms, and highlights their limitations, highlighting the need for more controlled experimental frameworks. Section 4 presents the proposed object-centric task-agnostic data augmentation pipeline, describing its design, implementation, and modular components, including dataset construction and evaluation metrics. Section 5 describes the experimental setup and presents the achieved results. Section 6 reports and analyzes the results, discusses limitations, and outlines directions for future work. Finally, Section 7 concludes the thesis by summarizing the key findings, and the Appendix provides additional insights and a more detailed analysis of the experimental results.

2 Theoretical Background

This chapter introduces the theoretical foundations required for the methodology and experiments of this thesis. Since the goal of this work is to develop an object-centric data augmentation approach and to analyze how image classification models respond to controlled manipulations of object appearance, context, and scale, the discussion begins with the fundamentals of modern image classification and the architectural principles of convolutional neural networks and Vision Transformers. It then reviews data augmentation as a general strategy for improving model robustness and motivating controlled transformations of training or evaluation data. We also describe the basics of style transfer, which help manipulate appearance cues while preserving structure for the shape-texture bias experiment.

The second part of the chapter covers the technical components that enable the proposed object-centric pipeline: image segmentation to isolate object regions and image inpainting to reconstruct backgrounds after object removal. Together, these topics provide the conceptual basis for the *sam2aug* pipeline and for the experimental analysis of shape-texture bias, contextual robustness, and scale sensitivity presented in later chapters.

2.1 Image Classification

Image classification is the task of assigning an image to one of a predefined set of categories (Goodfellow et al., 2016). In modern computer vision, this task is predominantly addressed with deep learning models that learn hierarchical feature representations directly from data. Since this thesis compares convolutional and transformer-based architectures under controlled input transformations, it is important to understand how these model families differ in their representation learning and inductive biases.

Instead of relying on manually designed features, deep neural networks learn multiple levels of abstraction through stacked nonlinear transformations (LeCun et al., 2015). This paradigm shift has led to major improvements in visual recognition performance across many tasks.

The foundations of deep learning were summarized in the overview by LeCun et al. (2015). The authors describe how deep neural networks can learn representations at multiple levels of abstraction, enabling models to transform raw sensory input such as images into increasingly complex feature representations. In the context of computer vision, these learned representations typically progress from low-level features (edges, corners, and textures) to mid-level patterns (parts and shapes) and finally to high-level semantic concepts corresponding to object categories.

Within image classification research, two major architectural paradigms have emerged in recent years: convolutional neural networks (CNNs) and transformer-based vision models. While CNNs dominated computer vision for more than a decade (Krizhevsky et al., 2012; He et al., 2016), more recent architectures such as Vision

Transformers (ViTs) adopt ideas originally developed for natural language processing and provide an alternative approach to visual representation learning (Dosovitskiy et al., 2020).

2.1.1 Convolutional Neural Networks

Convolutional neural networks are specialized deep neural networks designed to process grid-structured data (Goodfellow et al., 2016), such as images. Images possess a strong spatial structure, with neighboring pixels highly correlated and patterns often repeating across different locations. CNNs exploit these properties through local receptive fields, weight sharing, and hierarchical feature extraction (Goodfellow et al., 2016).

Instead of connecting every neuron to all input pixels, convolutional layers apply small learnable filters that slide across the image. Each filter detects specific patterns such as edges, textures, or shapes. Because the same filter is applied across the entire image, the network learns translation-invariant feature detectors that respond to the same pattern regardless of its spatial position (Goodfellow et al., 2016).

Deep Convolutional Architectures The effectiveness of deep CNNs for large-scale image recognition was demonstrated by the landmark architecture AlexNet, proposed by Krizhevsky et al. (2012). Their model achieved significant improvements on the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) (Russakovsky et al., 2015) in 2012 by combining deep convolutional layers with large-scale GPU training and rectified linear unit (ReLU) activations (Krizhevsky et al., 2012). This result marked the beginning of the modern era of deep learning in computer vision.

Subsequent research focused on improving network depth and architectural design. The VGG network, introduced by Simonyan and Zisserman (2014), demonstrated that deeper architectures with small convolutional filters can significantly improve performance. Their design uses stacks of 3×3 convolutional layers, which allows deeper networks while keeping the number of parameters manageable (Simonyan and Zisserman, 2014).

However, increasing network depth introduced new challenges during training, particularly the vanishing gradient problem, which makes optimization difficult in very deep networks (Huang et al., 2017). This issue was addressed by the Residual Network (ResNet) architecture proposed by He et al. (2016). ResNet introduced skip connections, also known as residual connections, that allow gradients to propagate more effectively through deep networks. Instead of directly learning a mapping $H(x)$, the network learns a residual function $F(x) = H(x) - x$, which is then added back to the input. This design enables the successful training of networks with more than 100 layers and significantly improves performance on ImageNet classification tasks (He et al., 2016).

Residual architectures have since become a standard backbone for many computer vision systems and are widely used in classification, detection, and segmentation models(Mahajan et al., 2018; He et al., 2020).

Limitations of Convolutional Architectures Despite their success, CNNs exhibit architectural biases that influence the types of visual information they prioritize. CNNs are often reported to emphasize local texture patterns more strongly than global shape, although the extent of this effect depends on the training data and evaluation protocol (Geirhos et al., 2018). Several studies have shown that CNNs trained on ImageNet often rely strongly on texture cues for object recognition, which may differ from human perceptual strategies (Geirhos et al., 2018; Tartaglini et al., 2022).

Additionally, CNNs have a limited ability to model long-range dependencies in images because information must propagate through many layers before distant spatial regions can interact. While techniques such as larger receptive fields and dilated convolutions partially address this limitation, they do not fully capture global relationships in the image.

These limitations motivated research into alternative architectures capable of modeling global interactions more directly.

2.1.2 Vision Transformers

Transformer Architecture Transformers were first introduced to natural language processing by Vaswani et al. (2017). A critical advantage of transformer architecture is the self-attention mechanism, which allows each element of an input sequence to attend to all other elements.

Self-attention computes pairwise interactions between elements of the input sequence. Given the query, key, and value representations derived from the input, attention weights determine how strongly each element influences the others. As a result, transformers can integrate information across the entire input context in a single layer (Vaswani et al., 2017).

Vision Transformer (ViT) The transformer architecture was later adapted for image recognition in the Vision Transformer (ViT) model proposed by Dosovitskiy et al. (2020). Instead of processing raw pixels directly with convolutions, the Vision Transformer first divides an image into fixed-size patches. Each patch is flattened and projected into a vector representation that serves as a token in a sequence, analogous to word tokens in language models (Dosovitskiy et al., 2020).

Positional embeddings are added to the patch representations to preserve spatial information. A stack of transformer encoder layers then processes the resulting sequence of patch embeddings, each comprising multi-head self-attention and feed-forward networks (Dosovitskiy et al., 2020).

This architecture allows the model to directly capture global interactions between different regions of the image. Unlike CNNs, which gradually build global context across multiple layers, Vision Transformers can model long-range dependencies at every layer via attention mechanisms (Vaswani et al., 2017).

Experiments on large-scale datasets demonstrate that Vision Transformers can achieve competitive or superior performance compared to convolutional networks when trained with sufficient data and computational resources (Dosovitskiy et al., 2020).

2.1.3 Differences Between CNNs and Vision Transformers

CNNs and Vision Transformers differ fundamentally in how they process visual information. CNNs incorporate strong inductive biases through convolution and pooling operations. These biases encode assumptions about locality and translation invariance, which help CNNs learn effectively even with moderate amounts of training data (Dosovitskiy et al., 2020). However, these architectural constraints may also limit their ability to capture global relationships.

Vision Transformers, in contrast, rely on attention-based global interactions and impose fewer assumptions about spatial structure. While this flexibility enables more expressive representations, it also means that ViTs typically require larger training datasets to achieve optimal performance (Dosovitskiy et al., 2020).

Empirical studies have also shown that these architectural differences influence how models utilize visual cues. Transformer-based models often exhibit stronger sensitivity to global shape information (Naseer et al., 2021), whereas CNNs tend to rely more on local texture features (Geirhos et al., 2018).

Understanding the differences between these architectures is essential when studying perceptual biases in computer vision models. In particular, architectural design choices may influence whether models rely more heavily on local texture cues or on global shape information.

2.2 Data Augmentation

Data augmentation is a commonly used technique in machine learning that increases the diversity of training data by applying transformations to existing samples while preserving their semantic labels (Kumar et al., 2024). This approach improves generalization and reduces overfitting by enabling models to learn invariances to variations commonly encountered in real-world data (Shorten and Khoshgoftaar, 2019). Overfitting is a term used to describe the effect in which a model memorizes specific details rather than learning general patterns and therefore performs poorly on unseen data.

In computer vision, augmentation is especially effective because many transformations can be applied without altering the image's semantic content. For example, objects generally remain recognizable under moderate changes in orientation, scale,

or illumination. By exposing a model to such variations during training, augmentation encourages the learning of feature representations that are robust to irrelevant visual changes (Kumar et al., 2024).

Traditional methods operate directly in pixel space and are typically categorized into geometric and photometric transformations. Geometric transformations, such as rotation, scaling, cropping, and translation, modify spatial structure while preserving object identity. Photometric transformations, including brightness and contrast adjustments or color perturbations, alter pixel intensities without changing spatial relationships. These techniques improve robustness to variations in viewpoint, illumination, etc. (Shorten and Khoshgoftaar, 2019).

Beyond these classical approaches, more advanced augmentation strategies have been proposed. Sample-mixing methods generate new training examples by combining multiple images. For instance, Mixup creates synthetic samples through linear interpolation of images and labels, encouraging smoother decision boundaries and improved generalization (Naveed et al., 2024). Related approaches, such as CutMix, extend this idea by replacing image regions, further enhancing robustness (Naveed et al., 2024).

More recently, generative augmentation techniques have emerged that use generative models to synthesize new training samples. Approaches based on generative adversarial networks or diffusion models can produce realistic images that extend the dataset's diversity beyond simple transformations of existing samples. These methods can be particularly beneficial in scenarios where datasets are small or imbalanced. However, the generated samples must remain semantically consistent with their labels and should not introduce artifacts that could negatively affect model training (Kumar et al., 2024).

Overall, data augmentation is a fundamental component of modern deep learning pipelines. While most methods operate on entire images, certain applications require more controlled manipulations of specific visual factors. This requirement motivates object-centric augmentation, in which transformations are applied to segmented object regions rather than to the full image. Such an approach is particularly suitable for constructing controlled evaluation datasets.

2.3 Style Transfer

Neural style transfer (NST) refers to methods that synthesize an image by combining the content structure of one image with the visual style of another. In this context, "content" denotes higher-level spatial structure and object layout, whereas "style" captures appearance statistics such as color, texture, and brush strokes. Jing et al. (2018) and Cai et al. (2023) describe NST as a central problem in image synthesis enabled by deep representations and categorize existing approaches into optimization-based, feed-forward, arbitrary-style, and more recent generative methods.

2.3.1 Foundational optimization-based style transfer

Modern NST is commonly traced back to the work of Gatys et al. (2015, 2016). Their method uses a pretrained convolutional neural network (Visual Geometry Group network (VGG)) to separate content and style representations. Content is encoded by feature activations from deeper layers, while style is represented by correlations between feature maps, captured via Gram matrices across multiple layers. The Gram matrix summarizes how different visual patterns co-occur in an image, providing a representation of texture that is independent of the exact spatial arrangement. Stylization is formulated as an optimization problem in which the output image is iteratively updated to minimize a weighted combination of content and style losses (Gatys et al., 2015, 2016).

This approach demonstrated that deep features enable a meaningful separation of structure and appearance. However, it is computationally expensive because each image requires iterative optimization, which limits scalability (Dumoulin et al., 2016).

Li et al. (2017a) provided a theoretical interpretation, showing that Gram-matrix matching corresponds to distribution alignment in feature space and is related to the maximum mean discrepancy with a second-order polynomial kernel, which explains why matching second-order statistics effectively transfers visual style.

2.3.2 Fast feed-forward style transfer

Subsequent work aimed to retain the perceptual quality of optimization-based NST while improving efficiency. Johnson et al. (2016) introduced feed-forward image transformation networks trained with perceptual losses derived from pretrained features. Their method approximates the Gatys optimization objective with a single forward pass and achieves comparable qualitative results while running much faster.

Further improvements highlighted the importance of normalization layers. Ulyanov et al. (2017) showed that replacing batch normalization with instance normalization significantly improves stylization quality, establishing it as a key architectural component.

Dumoulin et al. (2016) extended this idea by learning a shared network for multiple styles using conditional instance normalization. Rather than training one model per style, their approach learns style-specific normalization parameters, making multi-style transfer more scalable while keeping inference fast.

2.3.3 Arbitrary style transfer

Early feed-forward methods were limited to a fixed set of styles, which motivated the development of approaches capable of handling arbitrary styles at test time. Huang and Belongie (2017) addressed this limitation with Adaptive Instance Normalization (AdaIN), a simple mechanism that directly adapts the statistics of content features

to match those of a given style image, enabling fast and flexible stylization without requiring per-style training.

Similarly, Li et al. (2017b) proposed Universal Style Transfer, which aligns feature distributions between content and style images via whitening and coloring operations, generalizing to arbitrary styles while remaining closely related to earlier methods that match feature statistics.

Chen and Schmidt (2016) introduced a complementary perspective by focusing on local structure. The patch-based method they developed performs style transfer by matching and replacing local feature patterns, highlighting that stylization can be guided not only by global statistics but also by local correspondences.

2.3.4 Transformer- and diffusion-based methods

More recent work extends style transfer beyond convolutional architectures. Deng et al. (2022) proposed StyTr2, a transformer-based framework that explicitly models long-range dependencies between content and style representations. This design aims to better preserve global structure and fine details compared to earlier approaches.

In parallel, diffusion-based methods have been explored for style transfer. Zhang et al. (2023) introduced an inversion-based approach that leverages diffusion models for stylization, reflecting a broader shift toward using generative models to improve realism and controllability.

In this thesis, style transfer is relevant not primarily as an image synthesis task, but as a mechanism for generating controlled conflicts between shape and texture cues. The key requirement is therefore not only visual realism, but also preservation of recognizable object structure under altered surface appearance.

2.4 Image Segmentation

Image segmentation is an important task in computer vision that partitions an image into meaningful regions corresponding to objects or surfaces (Minaee et al., 2021). Segmentation provides pixel-level predictions, enabling precise localization of objects within a scene. This capability is essential for applications such as autonomous driving, medical imaging, and scene understanding (Minaee et al., 2021).

Before the rise of deep learning, segmentation methods relied primarily on low-level image features such as intensity, color, and texture. One influential classical approach is the active contour model, commonly referred to as snakes. Introduced by Kass et al. (1988), this method represents object boundaries as energy-minimizing curves that evolve under the influence of internal smoothness constraints and external image forces. The contour deforms until it aligns with salient image structures such as edges or boundaries (Kass et al., 1988).

Another important direction in classical segmentation research is focused on boundary detection and region grouping. Arbelaez et al. (2010) proposed a framework that combines local image cues with global contour information to generate hierarchical segmentations of images. Their method demonstrated that accurate boundary detection followed by hierarchical region grouping can produce meaningful object-level partitions of images (Arbelaez et al., 2010).

Further advances integrated appearance models, shape priors, and contextual relationships between object classes. The TextonBoost framework is a representative example that combines texture features with spatial context to perform joint object recognition and segmentation. In this approach, pixel-level classification is refined using contextual constraints to produce more coherent segmentation results (Shotton et al., 2006).

Although these classical approaches achieved promising results, they relied heavily on hand-crafted features and often required complex pipelines consisting of multiple processing stages.

Modern segmentation methods can be broadly categorized by the level of detail they produce. Semantic segmentation assigns a class label to every pixel in the image but does not distinguish between individual instances of the same object class. Instance segmentation extends this formulation by identifying and separating individual object instances. Panoptic segmentation combines both approaches by simultaneously labeling all pixels and distinguishing object instances from background or "stuff" classes, such as sky or road (Kirillov et al., 2019).

2.4.1 Deep Learning for Image Segmentation

The introduction of deep learning significantly transformed segmentation research by enabling models to learn hierarchical feature representations directly from data. Convolutional neural networks can capture complex visual patterns and perform dense pixel-level predictions within a unified architecture (Sultana et al., 2020).

Modern segmentation methods are typically based on fully convolutional architectures, which replace traditional fully connected layers with convolutional layers to preserve spatial resolution, thereby allowing networks to generate prediction maps that assign labels to individual pixels (Long et al., 2015).

According to Minaee et al. (2021), deep learning-based segmentation approaches have rapidly become the dominant paradigm due to their ability to learn robust representations from large datasets and to generalize across diverse visual domains.

The development of large annotated datasets has played an important role in enabling this progress. For example, the Microsoft COCO dataset provides extensive pixel-level annotations for everyday objects in complex scenes, supporting research on both semantic and instance segmentation tasks (Lin et al., 2014).

2.4.2 Promptable Segmentation Models

Recent research has introduced a new generation of foundation models for segmentation that can generalize across categories and tasks. Instead of predicting segmentation masks for a fixed set of predefined classes, these models allow segmentation to be guided by user-provided prompts such as points, bounding boxes, or existing masks (Kirillov et al., 2023).

The Segment Anything Model (SAM) (Kirillov et al., 2023) and its successor, SAM 2 (Ravi et al., 2024), represent this paradigm shift. As illustrated in Figure 1, SAM 2 extends promptable segmentation to both images and videos by combining an image encoder with a prompt encoder and a mask decoder, augmented by a memory mechanism. The memory component enables the model to store information from previous frames and propagate segmentation masks consistently across time (Ravi et al., 2024).

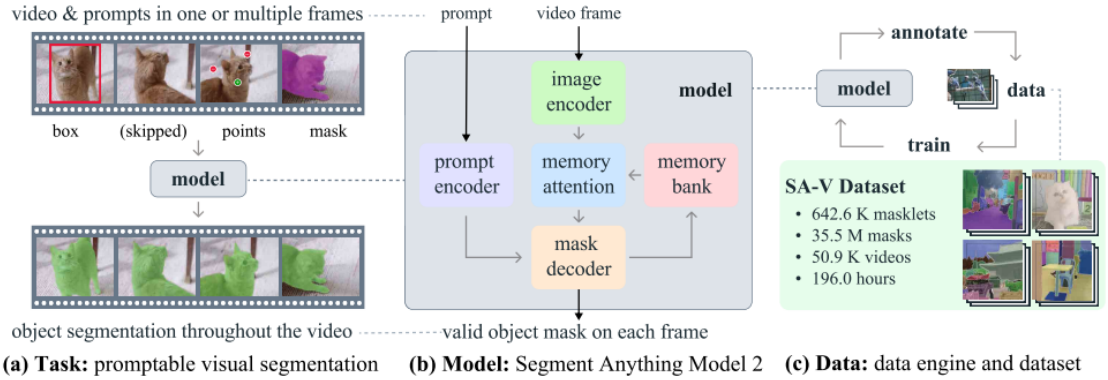


Figure 1: Overview of the Segment Anything Model 2 (SAM 2) architecture. The model combines an image encoder, a prompt encoder, and a mask decoder with a memory mechanism, enabling consistent segmentation across video frames (Ravi et al., 2024).

In contrast to traditional segmentation models that operate on individual images, SAM 2 processes video data as a sequence and leverages temporal context through its memory attention mechanism, thereby enabling it to generate coherent object masks across multiple frames given sparse user input, such as a single prompt in a single frame (Ravi et al., 2024).

In the context of this thesis, segmentation is not used as an end task, but as an enabling and the most important step for object-centric manipulation. Accurate object masks are required to separate foreground objects from background content and apply localized transformations. The promptable and temporally consistent segmentation capabilities of SAM 2 make it particularly suitable for constructing controlled datasets with manipulated object properties.

2.5 Image Inpainting

Image inpainting is the task of reconstructing missing or corrupted regions of an image such that the result appears visually plausible and consistent with the surrounding context (Quan et al., 2024). Given an image with masked regions, the goal is to estimate the missing pixel values based on the available information. Successful inpainting requires both local consistency, such as the continuation of edges and textures, and global semantic coherence, ensuring that the reconstructed content aligns with the overall scene structure (Quan et al., 2024).

Historically, image inpainting methods have evolved from classical signal-processing approaches to modern deep learning-based generative models (Quan et al., 2024).

2.5.1 Classical Inpainting Methods

Early approaches are based on diffusion processes, which propagate pixel information from the boundaries of the missing region inward. These methods are often formulated using partial differential equations and are effective for reconstructing small defects by preserving local structures such as edges and intensity gradients(Xu et al., 2023). However, they struggle with larger missing regions, as they rely solely on local information.

To overcome these limitations, patch-based methods were introduced. These approaches fill missing regions by copying similar patches from other parts of the image(Criminisi et al., 2004). By leveraging existing textures, they can produce more realistic results for larger regions. However, their performance depends on the availability of suitable source patches and may degrade when the missing region contains unique or complex structures (Quan et al., 2024).

2.5.2 Deep Learning-Based Inpainting

Deep learning has significantly advanced image inpainting by enabling models to learn high-level visual representations from large datasets. Early neural approaches, such as context encoders (Pathak et al., 2016), use encoder-decoder architectures to predict missing regions based on surrounding context.

Modern methods extend this idea using convolutional neural networks and generative modeling techniques. These models can infer semantically meaningful content and reconstruct large missing regions while maintaining global consistency. As a result, deep learning-based approaches have become the dominant paradigm in image inpainting (Quan et al., 2024).

2.5.3 Generative Inpainting

One of the key challenges in inpainting is modeling long-range dependencies across the image. To address this, recent methods incorporate attention mechanisms and

multi-scale feature representations (Quan et al., 2024), allowing information from distant regions to influence reconstruction.

Generative approaches, often based on adversarial training, further improve realism by encouraging synthesized content to resemble natural images (Pathak et al., 2016). More recent developments include transformer-based and diffusion-based models, which enhance global coherence and visual quality (Quan et al., 2024).

2.5.4 Large-Mask Inpainting

A particularly challenging scenario arises when large portions of an image are missing. In such cases, local contextual information may be insufficient for reconstructing the missing content.

An example of this global image-structure modeling approach is the LaMa (Large Mask Inpainting) model proposed by Suvorov et al. (2021). LaMa introduces fast Fourier convolutions (FFCs), which allow information to propagate across the entire spatial domain of the image. As shown in Figure 2, the LaMa architecture is a feed-forward ResNet-like inpainting network that integrates Fast Fourier Convolutions (FFC) to capture both local and global image context. By leveraging Fourier-based operations with an image-wide receptive field, the model can propagate information across the entire image and generate coherent reconstructions even in the presence of large, irregularly missing regions. The spectral transform module enables global information propagation by processing feature maps in the frequency domain.

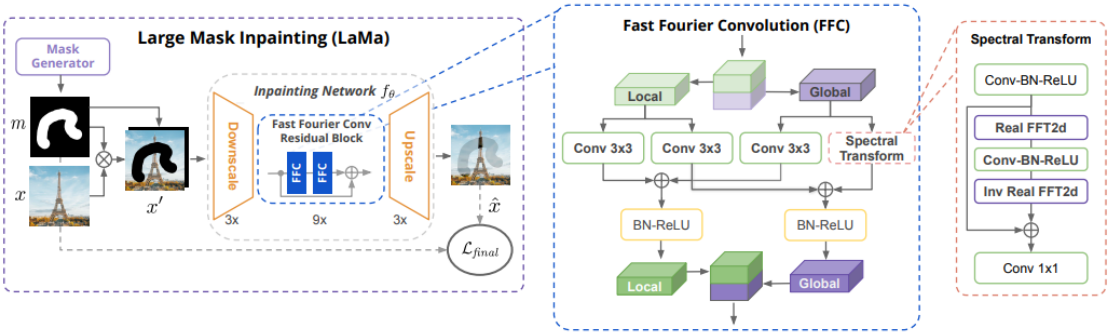


Figure 2: Overview of the LaMa inpainting architecture. The model combines a convolutional encoder-decoder structure with Fast Fourier Convolutions (FFC), enabling the propagation of both local and global information (Suvorov et al., 2021).

In addition, LaMa employs training strategies that improve robustness to varying mask sizes, making it particularly suitable for large-mask inpainting scenarios (Suvorov et al., 2021).

In this thesis, the inpainting model LaMa is used as a preprocessing step to remove objects from images while preserving plausible background content.

The concepts introduced in this chapter provide the technical basis for the experimental framework developed in this thesis. In particular, the differences between

CNNs and Vision Transformers motivate the comparison of architectural biases, while segmentation, inpainting, and style transfer form the operational components of the proposed object-centric augmentation pipeline. Building on this foundation, the following chapter reviews prior work on shape-texture bias and situates the present work within that literature.

3 Related Work

This chapter reviews prior work most relevant to the research questions of this thesis. Since the experimental analysis focuses on shape-texture bias and robustness under controlled object-centric transformations, the chapter first discusses studies on cue utilization in deep neural networks, with particular emphasis on the cue-conflict paradigm introduced by Geirhos et al. (2018). It then considers subsequent work that has questioned the interpretation and reliability of such evaluations, including studies on alternative experimental designs and architectural differences between convolutional and transformer-based models.

3.1 Shape and Texture Bias in Deep Neural Networks

Understanding which visual cues drive object recognition in deep neural networks has become an important topic in computer vision research. Human object recognition is known to rely primarily on global object shape, whereas modern convolutional neural networks (CNNs) trained on large-scale datasets such as ImageNet often rely heavily on local texture statistics. This discrepancy between human and machine perception has motivated a substantial body of research investigating the inductive biases of deep learning models (Geirhos et al., 2018; Tartaglini et al., 2022; Burgert et al., 2025; Kim et al., 2026).

A study by Geirhos et al. (2018) demonstrated that standard ImageNet-trained CNNs exhibit a strong texture bias in object recognition tasks. In this work, the authors introduced the cue-conflict evaluation paradigm, in which images are constructed so that the shape of one object category is combined with the texture of another. These hybrid images were generated using neural style transfer and allow direct measurement of whether a model prediction is driven primarily by shape information or by texture statistics (Geirhos et al., 2018).

The results demonstrated that human observers predominantly classify cue-conflict stimuli by object shape, whereas CNNs trained on ImageNet frequently classify them by texture cues. These findings suggested that CNNs implicitly learn representations that emphasize local texture patterns rather than global object structure (Geirhos et al., 2018).

To generate the cue-conflict stimuli used in their experiments, Geirhos et al. (2018) employed the neural style transfer method introduced by Gatys et al. (2015, 2016).

Although the cue-conflict paradigm has become widely used for analyzing visual biases in neural networks, subsequent work has identified several limitations in this methodology (Burgert et al., 2025; Kim et al., 2026).

3.2 Alternative Interpretations and Methodological Critiques

Several studies have questioned whether the strong texture bias reported for ImageNet-trained CNNs reflects an intrinsic architectural limitation or is partly shaped by the

design of the evaluation protocol itself (Burgert et al., 2025; Kim et al., 2026). This line of work suggests that conclusions about shape-texture bias depend not only on a model's internal representations but also on how cue-conflict stimuli are generated and how model behavior is measured.

A notable example is the work of Tartaglini et al. (2022), who re-examined the shape-texture bias question using an evaluation procedure inspired by developmental psychology experiments on human shape bias. In such experiments, participants are typically presented with an anchor object and asked to decide whether another object with the same shape or the same texture belongs to the same category. By adapting this paradigm to deep neural networks and measuring similarity in feature representations rather than classification outputs, the authors found that models may exhibit stronger shape sensitivity than previously reported. Their results indicate that at least part of the texture bias observed in earlier studies may arise from the specific construction of cue-conflict stimuli and from the choice of evaluation metric, rather than from a complete inability of CNNs to encode shape information (Tartaglini et al., 2022).

An important methodological point raised in this context concerns the stylization procedure used in the original cue-conflict datasets. Since texture is transferred to the entire image, including the background, the resulting stimuli may unintentionally amplify texture cues while reducing the perceptual visibility of the object shape, making it more difficult for models to predict object-level cue utilization.

Related concerns were raised by Kim et al. (2026), who critically reassessed the reliability of stylization-based cue-conflict benchmarks. The authors argue that such datasets may produce unstable or ambiguous measurements for several reasons. First, stylized images do not always yield perceptually clear and separable shape and texture cues. Second, relative metrics based on shape-versus-texture ratios may obscure models' absolute sensitivity to each cue. Third, evaluation on a restricted subset of predefined classes may distort the interpretation of predictions by ignoring the full classifier label space. To address these limitations, the authors propose the REFINED-BIAS benchmark, which aims to generate more clearly defined cue-conflict stimuli and evaluate model behavior using ranking-based metrics over the full class distribution (Kim et al., 2026).

Taken together, these studies show that measurements of shape-texture bias are highly sensitive to both stimulus generation and evaluation design and motivate the development of more controlled and modular pipelines for constructing diagnostic datasets. In particular, the object-centric approach adopted in this thesis, which applies texture manipulations only to segmented object regions while preserving background context, demonstrates its value.

3.3 Architectural Differences Between CNNs and Vision Transformers

Recent research has also investigated how architectural differences influence cue utilization. In particular, transformer-based architectures for vision processing have been shown to exhibit different inductive biases compared to convolutional neural networks (Naseer et al., 2021).

The study by Naseer et al. (2021) analyzed the internal representations and decision behavior of Vision Transformers (ViTs). Their results suggest that Vision Transformers often exhibit greater shape bias than convolutional architectures.

This difference is typically attributed to the transformer's architectural properties. While CNNs rely on local convolutional filters that emphasize spatially localized features, Vision Transformers process images as sequences of patches and integrate information globally through self-attention mechanisms (Naseer et al., 2021), thereby capturing long-range dependencies and global shape information more effectively than conventional convolutional architectures.

3.4 Motivation for the Present Work

The limitations of prior approaches described in this chapter motivated us to use our object-centric pipeline to extend prior work on shape-texture bias toward a more modular, controlled experimental framework for analyzing model behavior. Instead of manipulating the full image, the proposed pipeline first segments the foreground object and then selectively applies transformations to the object region, enabling texture modifications while preserving background context, or, conversely, manipulating context and scale while keeping object appearance fixed.

The following chapter, therefore, introduces the proposed *sam2aug* framework and describes how it is used to construct datasets for experiments on shape-texture bias, contextual robustness, and object scale.

4 Methods

4.1 Overview of the Approach

The objective of this work was to design, implement, and experimentally validate a segmentation-based data augmentation pipeline for image classification. The central research goal is to assess whether object-centric augmentation – based on explicit segmentation, background reconstruction, and controlled object manipulation – can serve as a structured framework for analyzing model behavior beyond conventional image-level augmentation techniques.

To this end, a task-agnostic augmentation framework, referred to as *sam2aug*, was developed. The pipeline integrates automatic object segmentation using a state-of-the-art foundation model, background reconstruction via inpainting, and controlled object transformation and relocation. It is designed to operate as a standalone image processing tool that enables systematic manipulation of visual factors while preserving semantic coherence.

The implementation follows several key design principles. First, the system adopts a modular architecture in which segmentation, inpainting, and relocation are implemented as independent components, enabling analysis of individual stages and facilitating extensibility. Second, the pipeline is compatible with large-scale datasets such as MS COCO (Lin et al., 2014) and ImageNet (Russakovsky et al., 2015), enabling its use in realistic experimental settings. Third, the framework is distributed as an open-source Python package installable via the pip package manager, ensuring accessibility and reproducibility. Finally, reproducibility is further supported through structured experiment configurations and the logging of metadata for each processing step.

The methodological approach is based on object-centric augmentation. Instead of applying global image transformations such as cropping, flipping, or color perturbations, the pipeline operates directly on object instances within an image. In a typical processing sequence, foreground objects are first segmented using a dedicated segmentation model. The segmented object is then removed from the image, resulting in a background representation with missing regions. These regions are subsequently reconstructed using an inpainting model. The extracted object can then be optionally transformed through geometric operations such as rotation, scaling, and translation, before being reinserted into the reconstructed background in a controlled manner. This process enables explicit manipulation of scene composition while maintaining visual plausibility.

4.2 *sam2aug*: Object-Centric Augmentation Pipeline

4.2.1 Implementation Details

The core processing logic of the *sam2aug* framework is implemented in the *AugmentationPipeline* class, which serves as an orchestration layer. This class is responsible

for validating inputs, invoking the segmentation module, deriving object and background representations from predicted masks, optionally performing background reconstruction, and applying object transformations and relocation.

The pipeline takes two primary inputs: an RGB image and a bounding box that specifies the target object's approximate location. The bounding boxes are derived from dataset annotations (e.g., the ImageNet validation subset) and serve as prompts for the segmentation model.

The system supports multiple execution modes, including segmentation only, segmentation combined with inpainting, segmentation with relocation, and the full pipeline that integrates all processing stages. For each processed object, the pipeline produces intermediate and final outputs that can be analyzed independently, enabling detailed investigation of each transformation step. The overview of the process and the intermediate processing outputs is shown in Figure 3. Given an input image and a bounding-box prompt, a segmentation model (SAM 2) predicts an object mask, which is then used to isolate the object from the background. The background can be reconstructed via inpainting, and the extracted object can optionally undergo geometric transformations (e.g., scaling and rotation). The object is then reinserted into the reconstructed background to produce the final composited image.

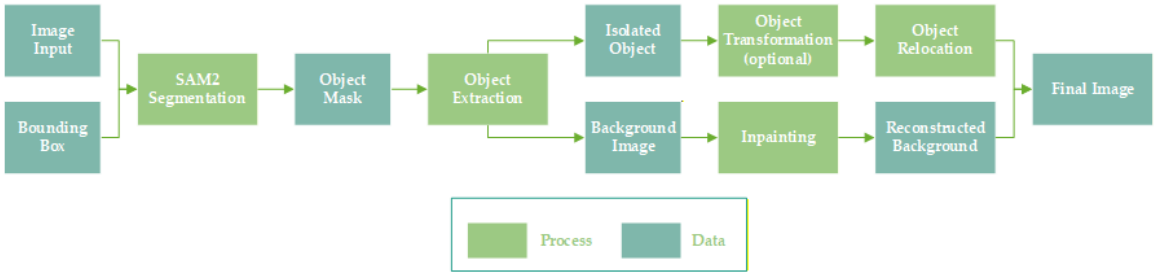


Figure 3: Overview of the *sam2aug* object-centric augmentation pipeline. Green blocks denote processing steps, while blue blocks represent intermediate data representations.

Object Segmentation The first stage of the pipeline is object segmentation, implemented by the *Segmenter* class, which wraps the Segment Anything 2 (SAM2) image predictor (Ravi et al., 2024). Given an input image and a bounding box prompt, the model predicts a binary mask that identifies the pixels belonging to the object.

The segmentation mask defines the object's spatial extent at the pixel level and serves as the central representation throughout the pipeline. It is reused in subsequent steps, including object extraction, background reconstruction, and object compositing.

Object Extraction and Background Removal Based on the predicted segmentation mask, the pipeline separates the object from the original image and constructs two complementary representations. First, an isolated object image is generated by retaining only the pixels corresponding to the object while removing all other regions. Second, a background image is created by removing the object region from the original image, resulting in a scene with missing pixels at the object's location. These representations enable independent manipulation of object and background content in later stages of the pipeline.

Background Reconstruction via Inpainting If enabled, the missing object region in the background image is reconstructed using an inpainting model implemented in the *Inpainter* module. The pipeline integrates the LaMa inpainting model (Suvorov et al., 2021), which can synthesize plausible background content in large regions of missing content.

To improve reconstruction quality, the segmentation mask can be optionally expanded prior to inpainting. The dilation step slightly enlarges the removed region to ensure that residual object pixels or segmentation artifacts are not preserved, preventing residual foreground information from leaking into the reconstructed background and potentially introducing artifacts or bias in subsequent analysis. Inpainting results without a segmentation mask dilation are shown in the Appendix (Figure 12).

The output of this stage is a reconstructed image that closely resembles the original scene but without the removed object.

Object Transformation Before relocation, the extracted object and its corresponding mask can be transformed using a random affine transformation. This transformation includes rotation, isotropic scaling, and translation. The rotation angle is sampled uniformly from the range $[-15, 15]$, while the scaling factor is drawn from the interval $[0.7, 1.3]$. Horizontal and vertical translations are sampled relative to the object dimensions. The selected parameter ranges introduce moderate geometric variations while preserving the object's recognizability and semantic identity. It is done to ensure that performance changes can be attributed to controlled transformations rather than to excessive distortions or loss of visual information.

By restricting transformations to this range, the experiment balances variability and realism, enabling a systematic analysis of geometric robustness under plausible object perturbations.

Object Relocation The relocation stage reinserts the transformed object into a target image, which may either be the reconstructed background or another provided scene. Prior to placement, the object can be rescaled to match the target image dimensions. To prevent the relocated object from becoming excessively small and visually insignificant, the applied scaling factor is constrained by a lower bound of 0.2, ensuring that the object retains at least 20% of its original size during relocation.

The relocation position is determined by configurable strategies, such as centered or random placement, or predefined anchor locations (e.g., top left, bottom right). The object is composited onto the target image using a soft blending mask derived from the segmentation mask, ensuring smooth transitions between the object and the background and avoiding visible artifacts. This step enables controlled manipulation of scene composition, thereby isolating contextual factors while preserving object identity.

Pipeline Outputs For each processed image, the pipeline produces a structured set of outputs, including the segmentation mask, the extracted object image, the background with the object removed, the reconstructed background (if inpainting is enabled), the transformed object, and the final composited image (Figure 4). In addition, metadata describing the processing configuration and runtime statistics is recorded. These outputs enable detailed analysis of each transformation step and support reproducibility.

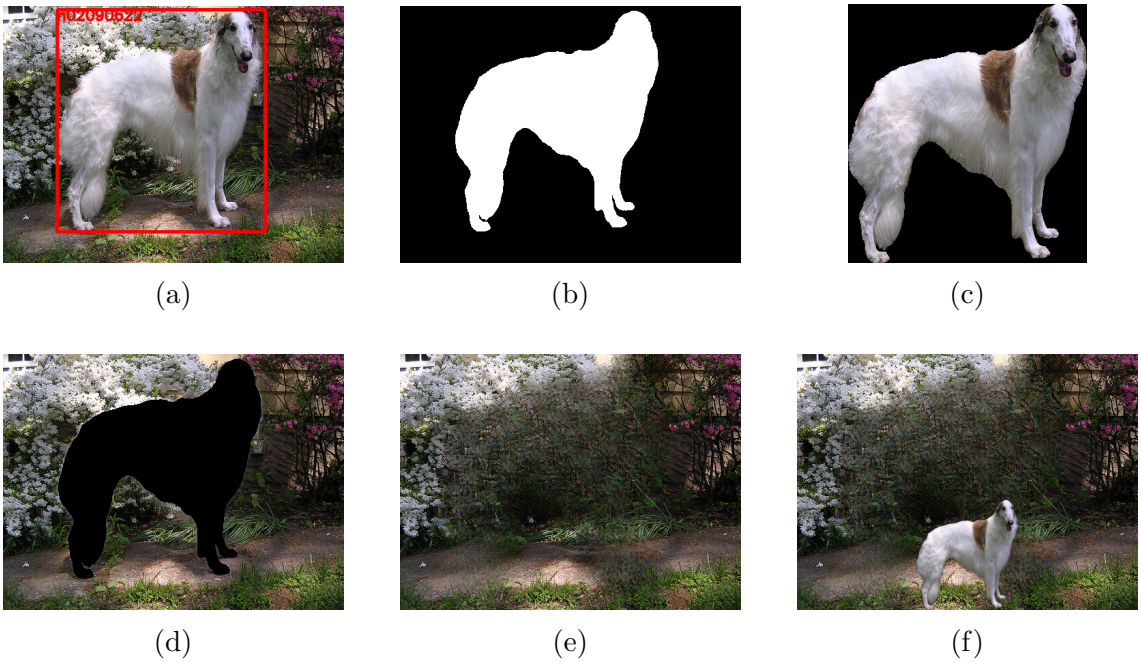


Figure 4: Illustration of intermediate outputs of the *sam2aug* pipeline for a single example: (a) original input image with bounding box overlay; (b) predicted segmentation mask obtained from the bounding box prompt; (c) extracted object based on the segmentation mask; (d) image with the object removed, where the masked region is set to zero; (e) reconstructed background generated via inpainting; (f) final composited image obtained by inserting the transformed object into the reconstructed background with anchor "bottom".

4.2.2 Implementation Availability

The complete implementation of the *sam2aug* framework is publicly available as an open-source project on GitHub via `https://github.com/roza-gaisina/sam2aug`. The repository contains all components required to reproduce the dataset generation and evaluation procedures described in this thesis.

4.3 Dataset Generation Using *sam2aug*

The *sam2aug* pipeline serves as the foundation for generating controlled datasets used in the experimental evaluation. By applying object-centric transformations, the pipeline enables the construction of datasets that isolate specific visual factors such as texture, context, and object scale.

4.3.1 Image Selection Constraints

The dataset construction process begins with selecting suitable input images. Since the segmentation model requires bounding-box prompts, only datasets with bounding-box annotations were considered. During development, the MS COCO dataset was used for pipeline testing and validation, while all experiments were conducted on the ImageNet validation dataset.

The experiments are based on the 16-category ImageNet subset introduced by Geirhos et al. (2018): *airplane, bear, bicycle, bird, boat, bottle, car, cat, chair, clock, dog, elephant, keyboard, knife,* and *truck*. According to Geirhos et al. (2018), classification of the single classes to parent categories was based on a lexical database WordNet (Miller, 1995).

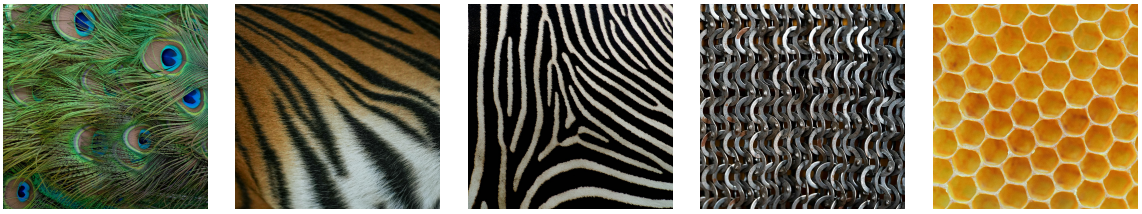
To ensure consistency across categories, a single representative class was selected for each parent category. This decision was motivated by the imbalance in the number of subclasses across categories, which would otherwise introduce variability in sampling and evaluation (e.g., the *dog* class with over 100 subclasses, the *knife* class with 1 subclass).

Additional selection criteria were applied to ensure data quality and pipeline suitability. All images were required to contain a clearly identifiable single object, and only samples that were correctly classified by all evaluated models were retained. Furthermore, constraints on bounding box size were introduced to ensure that objects were neither too small nor too large relative to the image. This was necessary to guarantee reliable segmentation and sufficient background context for inpainting. The list of selected classes for this experiment is provided in the Appendix (Table 4).

All selected images were manually inspected to verify segmentation quality and to exclude ambiguous cases or images containing multiple relevant objects. The category “oven” was excluded from the dataset because its bounding boxes were

systematically large, thereby degrading segmentation quality and limiting the applicability of shape-based manipulations. Samples of single images and their masks of the class "oven" are shown in the Appendix chapter A.1 (Figure 13).

For the shape-texture experiments, donor textures were selected from ImageNet classes characterized by strong visual patterns and high contrast. These include peacock, tiger, honeycomb, zebra, and chain mail. These textures were chosen to provide clear and distinguishable appearance cues (Figure 5).



(a) (b) (c) (d) (e)

Figure 5: Examples of donor textures used in the shape-texture bias experiments: (a) peacock feathers, (b) tiger fur, (c) zebra stripes, (d) chain mail, (e) honeycomb.

4.3.2 Object Segmentation and Mask Construction

For all dataset variants, object-level manipulation is performed using segmentation masks generated from bounding-box prompts. Given an input image I and a bounding box B, a segmentation model produces a binary mask $M \in \{0,1\}^{H \times W}$ indicating the object region:

$$M = \text{Segmenter}(I, B) \tag{1}$$

To ensure complete object coverage and avoid boundary artifacts, the mask is optionally dilated, resulting in an expanded mask M^{dil}:

$$M^{\text{dil}} = \text{dilate}(M, k) \tag{2}$$

where k denotes the dilation radius (in pixels), controlling the extent to which the mask is expanded.

The binary mask is converted into a continuous alpha mask $\alpha \in [0,1]^{H \times W}$ using Gaussian smoothing:

$$\alpha = \mathcal{G}(M^{\text{dil}}) \tag{3}$$

This soft mask enables smooth transitions between modified and unmodified regions during compositing and prevents visible seams at object boundaries.

A tight bounding box B_{tight} is computed from the mask:

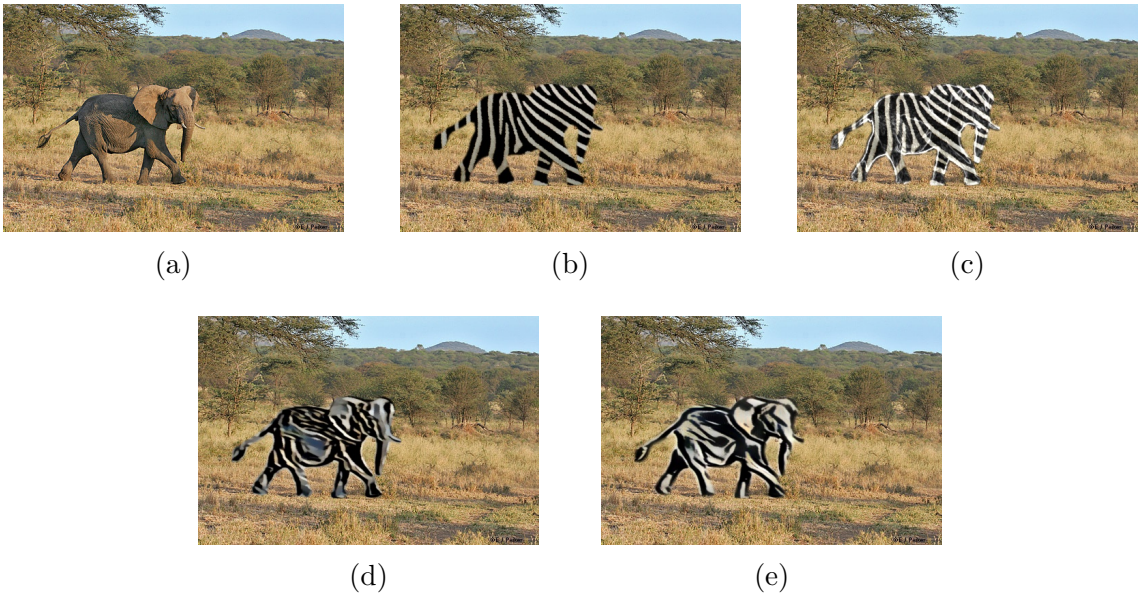
$$B_{\text{tight}} = \text{bbox}(M^{\text{dil}}) \tag{4}$$

This bounding box is used to resize donor textures to match the object's spatial extent and resolution.

4.3.3 Shape-Texture Dataset Variants

As *sam2aug* has a modular structure and allows using individual modules, we use only the segmentation part of the pipeline for this dataset generation.

Using the *Segmenter* module, four shape-texture datasets were generated, each corresponding to a different strategy for combining object shape and texture information. While the underlying segmentation and blending procedure is shared across all variants, the transformation applied to the object region differs in the extent to which it preserves spatial structure while modifying texture, enabling a controlled analysis of the relative importance of structural versus appearance cues. Figure 6 shows representative examples of the generated dataset variants. All images from the experimental datasets are available in the GitHub repository.



(a) (b) (c)

(d) (e)

Figure 6: Representative examples of the four shape-texture dataset variants generated by the pipeline: (a) original image, (b) Texture Only, (c) Texture + Edges, (d) Texture NST, (e) Texture AdaIN.

Texture Only In the Texture Only dataset, the object's appearance is fully replaced with a donor texture while preserving its spatial structure. The donor image is resized to match the object's tight bounding box and denoted as D. The final image is obtained by alpha blending:

$$I_{\text{out}} = (1 - \alpha) \odot I + \alpha \odot D \tag{5}$$

This formulation closely follows the cue-conflict paradigm introduced by Geirhos et al. (2018), with the important distinction that texture replacement is applied only to the object region while preserving the original background.

Texture + Edges To reintroduce structural information, the Texture + Edges dataset augments the donor texture with edge cues extracted from the original object. Let E denote the edge map extracted from the original image using the Sobel operator. The Sobel operator estimates image gradients by applying discrete convolutional filters that approximate first-order spatial derivatives in the horizontal and vertical directions (Kanopoulos et al., 1988):

$$E = \sqrt{(\partial_x I)^2 + (\partial_y I)^2} \tag{6}$$

where $\partial_x I$ and $\partial_y I$ denote the partial derivatives of the image intensity I with respect to the horizontal (x) and vertical (y) directions, respectively. These derivatives approximate local intensity changes and highlight object boundaries.

The edge map is normalized, restricted to the object region, and optionally smoothed. It is then combined with the donor texture, resulting in D':

$$D' = \text{clip}(D + \alpha \odot E) \tag{7}$$

The resulting image I_{out} is obtained through alpha blending:

$$I_{\text{out}} = (1 - \alpha) \odot I + \alpha \odot D' \tag{8}$$

This approach preserves local structural information while maintaining the overall texture transformation.

Texture NST (Neural Style Transfer) In the Texture NST condition, the object region is stylized using optimization-based neural style transfer following Gatys et al. (2016). Let I_c denote the content image (original object region), I_s the style image (donor texture), and $\hat{I}$ the optimized output image. Instead of directly replacing pixel values, the stylized image $\hat{I}$ is obtained by minimizing a weighted loss function:

$$\hat{I} = \arg\min_I \left(\lambda_c \, \mathcal{L}_{\text{content}}(I, I_c) + \lambda_s \, \mathcal{L}_{\text{style}}(I, I_s) + \lambda_{tv} \, \mathcal{L}_{\text{TV}}(I)\right) \tag{9}$$

The style loss $\mathcal{L}_{\text{style}}$ is computed using Gram matrices:

$$\mathcal{L}_{\text{style}} = \sum_l w_l \, \|G_l(I) - G_l(I_s)\|_F^2 \tag{10}$$

where $G_l(\cdot)$ denotes the Gram matrix computed from feature activations at layer l.

The stylization is applied only to the segmented object region and subsequently blended back into the original image using the alpha mask.

Texture AdaIN The Texture AdaIN condition applies fast style transfer using Adaptive Instance Normalization (Huang and Belongie, 2017). Let f_c and f_s denote feature representations of the content and style images, respectively. Then the transformation is defined as follows:

$$\text{AdaIN}(f_c, f_s) = \sigma(f_s) \left(\frac{f_c - \mu(f_c)}{\sigma(f_c)} \right) + \mu(f_s) \tag{11}$$

The transformed features f are interpolated with the original content features:

$$f = \lambda \cdot \text{AdaIN}(f_c, f_s) + (1 - \lambda) \cdot f_c \tag{12}$$

The resulting representation is decoded into an image and blended into the original image using the alpha mask. In contrast to NST, AdaIN operates in feature space and enables faster stylization at the cost of reduced structural control.

4.3.4 Dataset Generation for Object Relocation and Rescaling

In addition to the shape-texture datasets, two further experimental datasets were constructed. For the object relocation experiment, segmented objects were inserted into novel background environments, enabling analysis of contextual robustness. For the object rescaling experiment, objects were resized and reinserted into the original scene after background reconstruction, allowing controlled variation of object scale.

To ensure valid and interpretable transformations for object relocation and rescaling experiments, additional constraints were applied during dataset construction. A key requirement is that objects remain clearly recognizable after transformation while sufficient background context is preserved for inpainting. To enforce this, images were filtered based on the relative size of the object r:

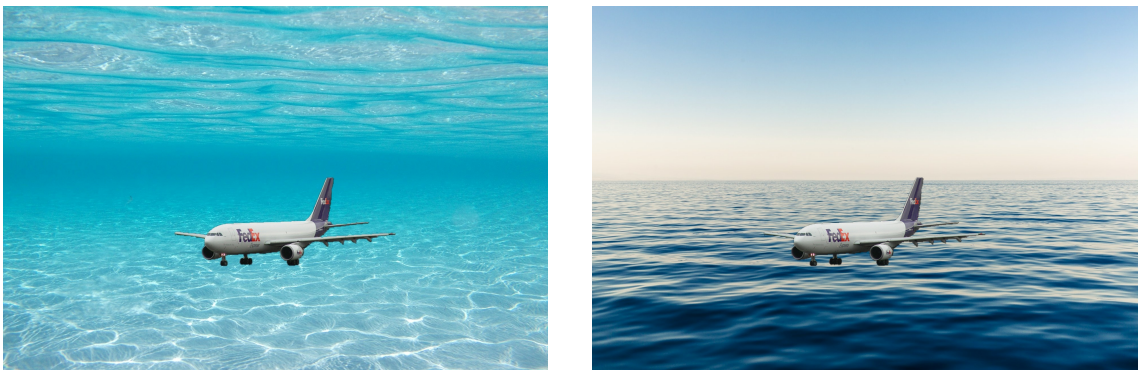
$$r = \frac{\text{area}(B)}{\text{area}(I)}, \quad 0.25 \leq r \leq 0.75 \tag{13}$$

This constraint avoids extremely small objects that would become unrecognizable after scaling, as well as overly large objects that leave insufficient background for reconstruction.

Furthermore, only images in which all objects were fully visible were retained. Images in which the bounding box intersected the image boundary were excluded to prevent truncation artifacts during relocation. Examples of object truncation can be seen in Figure 14.

After automatic filtering, all remaining samples were manually inspected to ensure segmentation quality and absence of ambiguous object boundaries. Several categories were excluded due to constraints named above, including *brown bear, beer bottle, computer keyboard,* and *cleaver*. From the remaining categories, a fixed number of images per class was selected to ensure balanced evaluation ($N = 5$). The list of selected classes for this experiment is provided in the Appendix (Table 5).

Object Relocation with Background Shift In the object relocation experiment, segmented objects are inserted into novel background environments to evaluate robustness to contextual changes using *Segmenter* and *Relocator* modules of *sam2aug*. The generation process consists of extracting the object, selecting a target background, and compositing the object into the new scene. Objects are placed onto predefined background images representing distinct environments, specifically open water and water surface. To ensure consistency across samples, object size is controlled using an area-based scaling factor, and objects are placed at a fixed central position. The final image is generated using soft blending to ensure visual consistency. This setup enables controlled analysis of context sensitivity and generalization to out-of-distribution backgrounds. Figure 7 illustrates the object relocation setup with background shift.



(a) (b)

Figure 7: Example images from the object relocation with background shift dataset. Segmented objects are inserted into novel background environments ((a) open-water and (b) water-surface scenes) while preserving their appearance.

Object Rescaling on Inpainted Background For this experiment, we use all three modules of the *sam2aug* pipeline: *Segmenter*, *Relocator*, and *Inpainter*. The object rescaling experiment isolates the effect of object size while keeping all other factors constant. The object is first segmented and removed from the original image, and the background is reconstructed using inpainting. The extracted object is then resized using predefined scale factors and reinserted into the bottom of the reconstructed background.

To improve inpainting quality, the segmentation mask indicating the region to be inpainted is dilated prior to reconstruction, preventing residual artifacts from influencing the reconstructed background with $k = 35$ pixels:

$$M^{\text{dil}} = \text{dilate}(M, k) \tag{14}$$

By keeping the background constant and varying only object size, this setup enables a controlled analysis of scale sensitivity and robustness to geometric transformations. Figure 8 presents the object rescaling process across different scale factors.

(a) (b) (c) (d) (e)

Figure 8: Example images from the object rescaling dataset. Objects are resized and reinserted into the original scene after background reconstruction: (a) original image, (b) scale 100%, (c) scale 75%, (d) scale 50%, (e) scale 25%.

4.4 Evaluation Framework

4.4.1 Shape-Texture Bias Evaluation

To analyze model decision strategies, we adopt the cue-conflict paradigm introduced by Geirhos et al. (2018). In this setting, each image contains two competing semantic cues: a shape label corresponding to the object's global structure and a texture label corresponding to its surface appearance.

Model predictions are evaluated with respect to both labels to determine whether decisions are driven by shape or texture information.

Top-1 and Top-5 Decision Metrics We begin by analyzing the model's top-1 prediction, denoted by $\hat{y}_1$; y_{shape} denotes the ground-truth class corresponding to the object's shape, and y_{texture} the class associated with the donor texture. Each prediction is categorized based on its alignment with the shape or texture label:

$$\text{decision}_{\text{top-1}} = \begin{cases} \text{shape} & \text{if } \hat{y}_1 = y_{\text{shape}} \\ \text{texture} & \text{if } \hat{y}_1 = y_{\text{texture}} \\ \text{other} & \text{otherwise} \end{cases} \tag{15}$$

This categorization yields three mutually exclusive outcomes: the prediction is consistent with the object's shape, consistent with its texture, or unrelated to both.

While top-1 predictions provide a clear and interpretable signal, they discard information about alternative hypotheses considered by the model. To capture this, we additionally analyze the top-5 predictions, denoted by the set $\hat{Y}_5$.

The decision rule is extended as follows:

$$\text{decision}_{\text{top-5}} = \begin{cases} \text{shape} & \text{if } y_{\text{shape}} \in \hat{Y}_5 \wedge y_{\text{texture}} \notin \hat{Y}_5 \\ \text{texture} & \text{if } y_{\text{texture}} \in \hat{Y}_5 \wedge y_{\text{shape}} \notin \hat{Y}_5 \\ \text{both} & \text{if } y_{\text{shape}}, y_{\text{texture}} \in \hat{Y}_5 \\ \text{other} & \text{otherwise} \end{cases} \tag{16}$$

This formulation allows us to distinguish between cases in which the model strongly favors one cue and cases in which it simultaneously represents both shape and texture hypotheses among its most probable predictions. In particular, the "both" category provides insight into situations where the model is uncertain or encodes multiple competing interpretations.

Shape Bias To summarize model behavior across the dataset, we compute the shape bias, defined as:

$$\text{Shape Bias} = \frac{N_{\text{shape}}}{N_{\text{shape}} + N_{\text{texture}}} \tag{17}$$

where N_{shape} and N_{texture} denote the number of predictions consistent with shape and texture, respectively. This definition follows the formulation used in prior work of Geirhos et al. (2018) and provides a normalized measure of the model's preference between the two competing cues.

Evaluation on Original Images For non-stylized (original) images, the distinction between shape and texture is not meaningful, as both cues correspond to the same object. In this case, evaluation reduces to standard classification metrics. The top-1 accuracy is defined as:

$$\text{Top-1 Accuracy} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}\left[\hat{y}_1^{(i)} = y^{(i)}\right] \tag{18}$$

where N denotes the number of samples in the dataset.

This metric serves primarily as a sanity check because one of the selection criteria for the experiment dataset images was that the objects in the original images were correctly predicted by all 4 selected models.

Semantic Robustness Metrics ImageNet classification is inherently fine-grained, and models may produce predictions that are semantically close to the ground truth but not identical. For example, predicting *African elephant* instead of *Indian elephant* constitutes a strict classification error, yet remains semantically meaningful.

To address this, we introduce parent-category-based metrics that group related classes into broader semantic categories as described by Geirhos et al. (2018).

Parent Top-1 Match Let $P(y)$ denote a mapping from a class label y to its corresponding parent category. We define a parent-level correctness indicator for the top-1 prediction:

$$\text{Parent Top-1} = \mathbf{1}\left[P(\hat{y}_1) = P(y_{\text{shape}})\right] \tag{19}$$

This metric evaluates whether the model correctly identifies the broader semantic category, even if it fails at the fine-grained class level.

Parent Top-5 Match

Similarly, for the top-5 predictions:

$$\text{Parent Top-5} = \mathbf{1}\left[\exists \hat{y} \in \hat{Y}_5 : P(\hat{y}) = P(y_{\text{shape}})\right] \tag{20}$$

This metric provides a more permissive measure of semantic correctness, capturing whether the correct parent category appears among the model's most likely hypotheses.

Parent Probability Mass

To obtain a continuous measure of semantic alignment, we compute the total probability mass assigned to the correct parent category:

$$\text{Parent Mass} = \sum_{c \in P(y_{\text{shape}})} p(c \mid x) \tag{21}$$

where $p(c \mid x)$ denotes the softmax output of the model for class c given input x. This metric measures how much of the model's output distribution is assigned to classes within the correct parent category, even when the exact class is not predicted. The evaluation framework evaluates robustness to fine-grained classification errors and assesses whether models preserve coarse semantic understanding.

This evaluation is very helpful, as standard accuracy metrics alone are insufficient to characterize the internal decision strategies of modern vision models. In particular, shape-texture bias evaluation enables a direct analysis of whether models rely on global shape information or local texture cues, a key difference between human and machine vision (Geirhos et al., 2018).

4.4.2 Evaluation under Contextual and Geometric Transformations

To complement the shape-texture analysis, additional metrics are introduced to evaluate model behavior under controlled contextual and geometric transformations. For both the object relocation and object rescaling experiments, classification accuracy is evaluated after transformation. The top-1 accuracy measures whether the predicted label matches the ground truth, while top-5 accuracy evaluates whether the correct label appears among the five most probable predictions.

Prediction Confidence Change To assess changes in model confidence, the difference in predicted probability for the correct class before and after transformation is computed. This metric quantifies the extent to which the model's confidence is affected by the applied perturbation.

Let x denote the original image and x' the transformed image. Let y be the ground-truth label and let $p(c \mid x)$ denote the predicted probability assigned by the model to class c given input x.

The change in predicted probability for the correct class is defined as:

$$\Delta P_{\text{correct}} = p(y \mid x') - p(y \mid x) \tag{22}$$

This metric measures how the model's confidence in the correct class changes under transformation. Negative values indicate a decrease in confidence, while values close to zero indicate robustness to the applied perturbation.

Furthermore, prediction stability is quantified by the frequency with which the predicted class label changes between the original and transformed images, providing insight into the sensitivity of model decisions to controlled modifications.

Prediction Change Rate Let $\hat{y}_1(x)$ denote the top-1 predicted class for input x. The prediction change rate is defined over a dataset of size N as:

$$\text{Change Rate} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1} \left[\hat{y}_1(x^{(i)}) \neq \hat{y}_1(x'^{(i)}) \right] \tag{23}$$

where $\mathbf{1}[\cdot]$ is the indicator function, which equals 1 if the condition is true and 0 otherwise.

This metric quantifies how frequently the model changes its predicted class after transformation, independently of whether the prediction is correct. A higher change rate indicates greater sensitivity to applied perturbation.

Together, these metrics provide a comprehensive evaluation of robustness by combining correctness, confidence, and prediction stability.

The datasets and evaluation framework described in this chapter form the basis for the experimental analysis presented in the following chapter. Using the *sam2aug* pipeline, controlled datasets are generated to systematically investigate shape-texture bias, contextual robustness, and sensitivity to object scale in modern image classification models.

5 Experiments and Results

5.1 Experimental Setup

In this chapter, we describe experiments designed to directly evaluate the object-centric transformations introduced in Chapter 4. Each experiment isolates a specific visual factor by leveraging a corresponding component of the *sam2aug* pipeline. In particular, the shape-texture datasets (Section 4.3.3) are used to analyze cue-conflict behavior, the relocation mechanism (Section 4.3.4, paragraph 1) is used to study contextual robustness, and the rescaling procedure (Section 4.3.4, paragraph 2) enables controlled evaluation of scale sensitivity. This design ensures that each experimental result can be directly attributed to a specific transformation applied at the object level.

Four ImageNet-pretrained models were evaluated, namely ResNet-18, ResNet-50, ViT-Tiny, and ViT-Base. These models were selected to enable a comparison between convolutional neural networks and Vision Transformers of different capacities, following commonly adopted pairings in the literature: ResNet-18 versus ViT-Tiny (Takahashi et al., 2024) and ResNet-50 versus ViT-Base (Kawadkar, 2025).

The evaluation is conducted on a curated subset of ImageNet validation images, serving as a control condition, and on multiple datasets generated by the *sam2aug* pipeline. These datasets are designed to isolate specific visual factors, including shape-texture conflicts, background context, and object scale. Model predictions are evaluated using the metrics defined in Chapter 4, including top-1 and top-5 decision categories (shape, texture, other), the derived shape-bias measure, and parent-level semantic robustness metrics.

All experiments were conducted on a remote Ubuntu server equipped with an Intel Xeon W-2265 CPU and an NVIDIA RTX A5000 GPU. The implementation uses Python 3.10.19 and PyTorch 2.9.0 with CUDA 13.0 support.

5.2 Shape-Texture Bias Experiments

This experiment directly builds on the dataset variants introduced in Section 4.3.3, where different strategies for modifying object appearance are defined. It follows the cue-conflict paradigm of Geirhos et al. (2018), where each sample contains a shape label (object identity) and a conflicting texture label derived from a donor image.

5.2.1 Comparison Across Dataset Variants

Shape Bias To compare model behavior under different shape-texture conditions, we first analyze the shape bias across all dataset variants (Figure 9). This metric provides a compact and interpretable measure of whether model decisions are primarily driven by shape or texture cues.

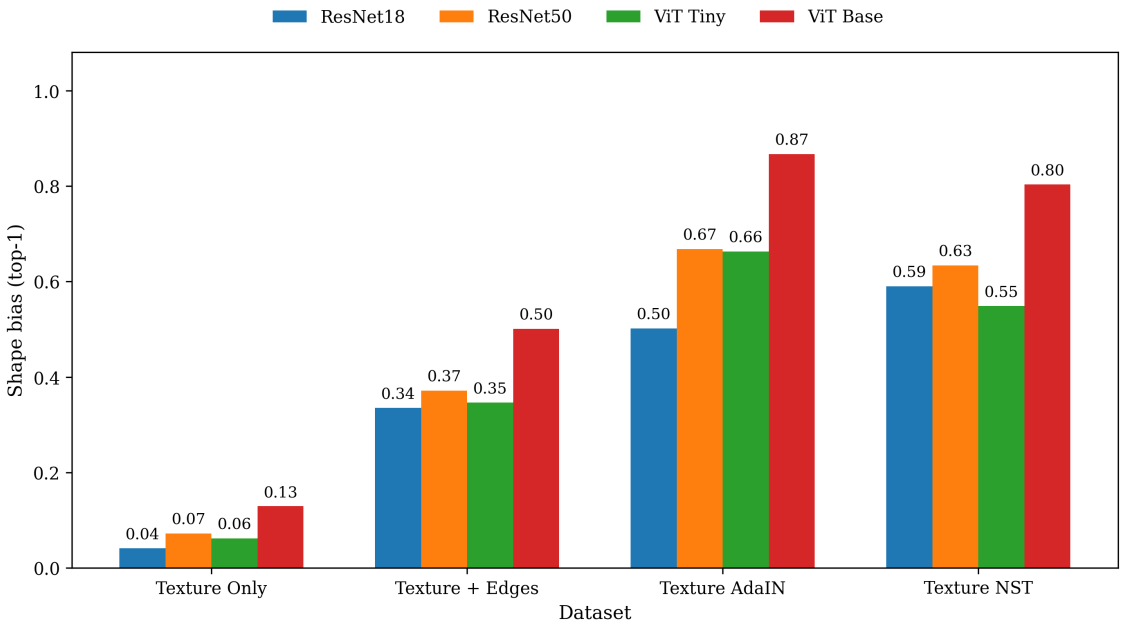


Figure 9: Shape bias computed from top-1 predictions across all dataset variants (x-axis) across evaluated models (bar color).

Across all models, shape bias increases consistently from the Texture Only condition to the structurally enriched conditions (Texture + Edges, NST, and AdaIN). This finding indicates that increasing structural consistency shifts model decisions from texture-based toward shape-based cues.

In the Texture Only condition, all models exhibit strong texture bias, with shape bias close to zero. The lowest value is observed for ResNet-18 (0.04), whereas ViT-Base reaches 0.13.

Introducing structural information (Texture + Edges) substantially increases shape bias. For example, CNN-based models achieve values of around 0.34-0.37, whereas ViT-Base achieves 0.50.

This trend continues for stylized variants. In the Texture NST condition, shape bias remains high across all models (e.g., 0.59 for ResNet-18 and 0.80 for ViT-Base). The highest values are observed for Texture AdaIN, with ViT-Base reaching 0.87, indicating the strongest shape preference across all conditions.

These results indicate that increasing the structural consistency of the object representation leads to a systematic shift from texture-based to shape-based decision-making.

The same trend is observed for top-5 shape bias (Figure 15 in Appendix), where values are consistently higher but follow the same ordering across conditions.

Top-1 Decision Distribution The distribution of top-1 predictions (shape, texture, other) across dataset variants is shown in Figure 10. This analysis provides a more detailed view of model behavior beyond the aggregated shape-bias metric.

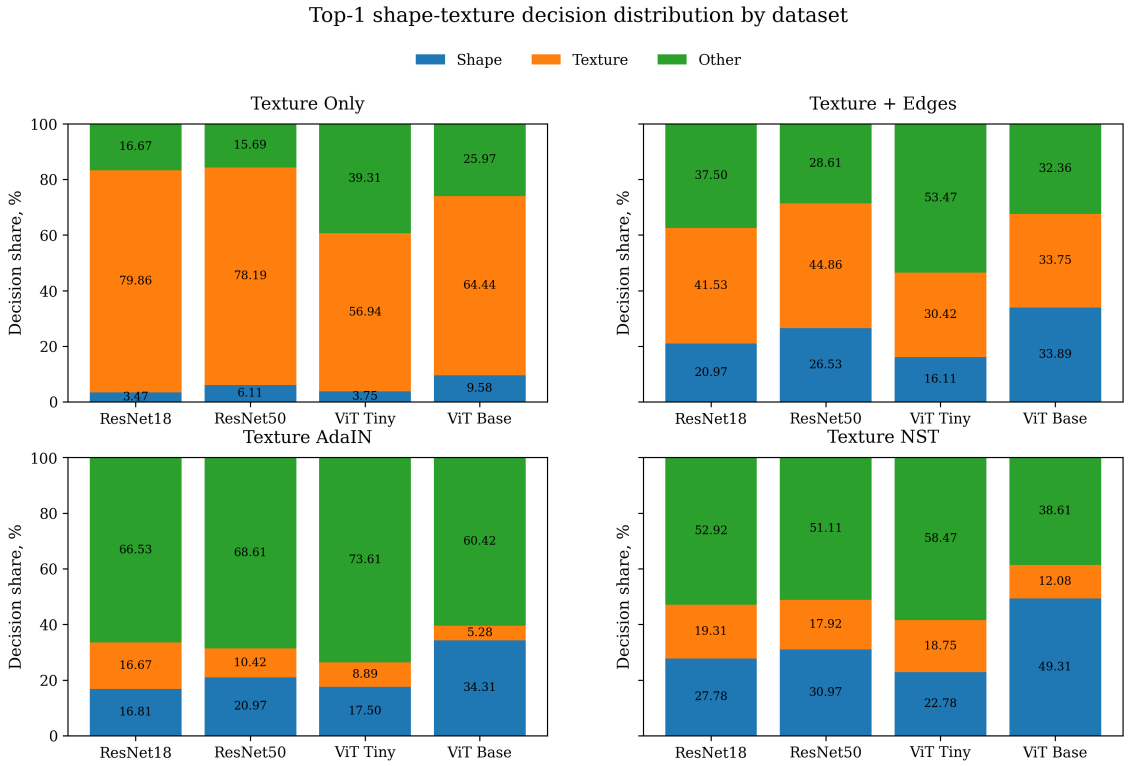


Figure 10: Distribution of top-1 predictions across shape, texture, and other categories for all dataset variants and models. Each bar represents the proportion of decisions aligned with shape or texture labels, or assigned to neither category for a single model.

In the Texture Only dataset, all models are dominated by texture-consistent predictions, which account for the majority of decisions (e.g., up to 79.86% for ResNet-18). Shape-consistent predictions remain rare across all models, consistently below 10%.

In the Texture + Edges dataset, the proportion of texture-based predictions decreases, while shape-consistent predictions increase across all models. At the same time, the proportion of "other" predictions increases, indicating greater uncertainty or ambiguity in model decisions.

In the Texture AdaIN dataset, texture-based predictions are further reduced, and the majority of predictions fall into the "other" category (e.g., up to 73.61% for ViT-Tiny). While shape-consistent predictions increase compared to the Texture Only condition, they remain lower than in the Texture + Edges dataset for most models.

In contrast, the Texture NST dataset shows a clear increase in shape-consistent predictions, while texture-based predictions remain low. For several models, partic-

ularly Vision Transformers, shape becomes the dominant non-"other" decision (e.g., 49.31% for ViT-Base).

These results highlight that changes in shape bias are driven not only by shifts between shape and texture predictions, but also by the proportion of "other" decisions, which is particularly relevant for stylization-based transformations.

5.2.2 Model Family Comparison

To better understand architectural differences, results are aggregated across model families in the Figure 11.

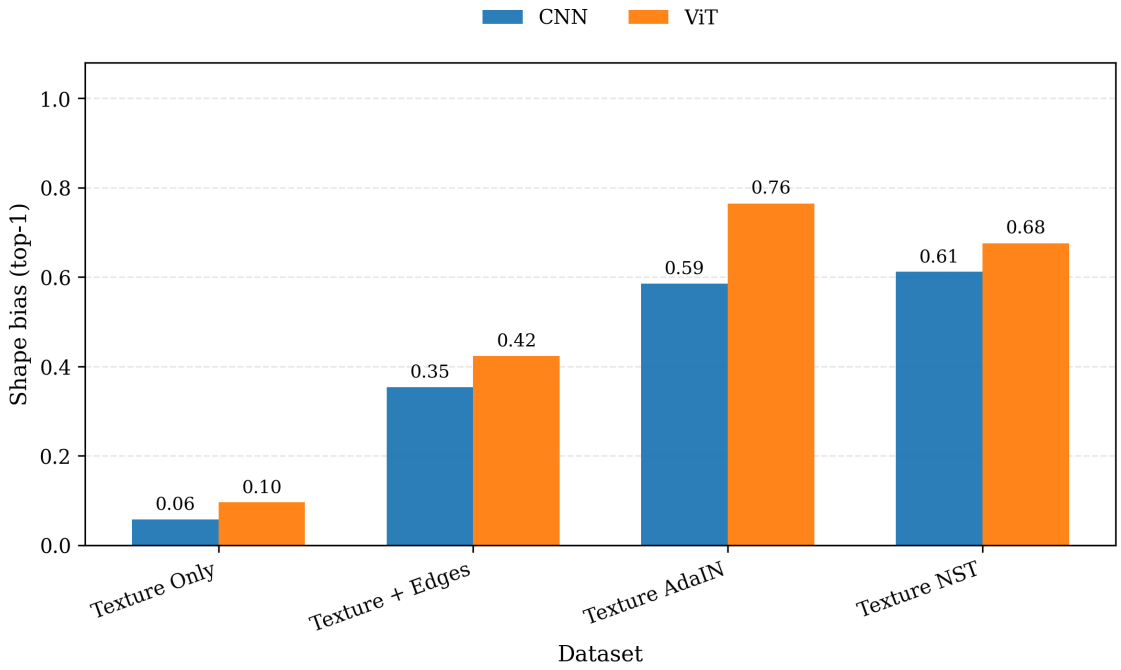


Figure 11: Comparison of top-1 shape bias between convolutional neural networks (ResNet-18, ResNet-50) and Vision Transformers (ViT-Tiny, ViT-Base) across dataset variants. Bars represent the average shape bias within each model family.

Aggregating results across model families (Figure 11) shows that Vision Transformers consistently exhibit higher shape bias than convolutional neural networks across all datasets.

While both families show low shape bias in the Texture Only condition (CNNs: 0.06, ViTs: 0.10), the gap increases for all structurally enriched variants. For example, in the Texture AdaIN condition, CNNs reach 0.59, whereas Vision Transformers achieve 0.76.

These results suggest that Vision Transformers may rely more strongly on global representations than convolutional networks, with ViT-Base consistently exhibiting the highest shape bias.

5.2.3 Semantic Robustness

In addition to the decision-based analysis, model behavior was also evaluated using parent-level semantic metrics as defined in Section 4.4.1. These metrics provide a complementary perspective by assessing whether predictions remain semantically consistent at a coarser, parent category level, even when fine-grained classification fails. While the overall trends observed in these metrics are consistent with the shape-bias results, they do not provide additional insight beyond the primary analysis of decision distributions and shape bias. For this reason, detailed results, including parent top-1 and top-5 matches as well as probability mass, are reported in the Appendix (Table 6).

5.2.4 Summary of Key Findings

Across all experiments, three consistent patterns emerge. First, texture bias dominates in the absence of structural cues, as observed in the Texture Only condition. Secondly, introducing structural information (edges or stylization) systematically increases shape bias, with the strongest effect observed for NST and AdaIN. Third, Vision Transformers exhibit a consistently stronger reliance on shape information than CNNs, both at the decision level and in terms of semantic robustness.

Detailed class-wise information is provided in the Appendix (Figures 16- 19), where additional variability across object categories and models can be observed.

5.3 Object Relocation with Background Shift Experiment

This experiment uses the relocation mechanism and the dataset construction procedure described in Section 4.3.4 to isolate the effect of background context.

The robustness to contextual changes will be evaluated by leveraging the relocation component of the *sam2aug* pipeline. Objects are extracted from their original images and inserted into novel background environments, specifically open-water and water-surface scenes, while preserving their appearance.

Results and Observations Table 1 presents the quantitative evaluation of classification performance, confidence, and prediction stability under background shift conditions. Bold values indicate the best-performing results for each metric, corresponding to the highest accuracies and confidence measures and the lowest error-related metrics.

Relocating objects into novel backgrounds reduces model performance across all architectures. ResNet-18 shows the strongest degradation, achieving 43.64% top-1 accuracy on the open-water background, while ViT-Base maintains the highest performance at 83.64%.

Top-5 accuracy remains high across all models, reaching 98.18% for ViT-Base for both backgrounds and 98.18% for ResNet-50 on the water-surface background, indicating that the correct class is often retained among candidate predictions despite decreases in top-1 accuracy.

All models exhibit reduced confidence in predictions after relocation. The largest decrease is observed for ResNet-18 (-54.49%), whereas ViT-Base shows a smaller reduction (-19.82%). Prediction change rates further confirm that convolutional networks are more sensitive to contextual changes than Vision Transformers.

Table 1: Quantitative evaluation of classification performance and confidence under background shift conditions (open water vs. water surface) across different model architectures

	ResNet-18		ViT-Tiny		ResNet-50		ViT-Base	
Metric	OpenW.	Surf.	OpenW.	Surf.	OpenW.	Surf.	OpenW.	Surf.
Top-1 Acc (gen.)	43.64%	65.45%	52.73%	50.91%	72.73%	74.55%	**83.64%**	**83.64%**
Top-5 Acc (gen.)	80.00%	90.91%	72.73%	81.82%	92.73%	**98.18%**	**98.18%**	**98.18%**
Mean P(correct) (orig.)	88.48%	88.48%	85.42%	85.42%	**94.72%**	**94.72%**	89.47%	89.47%
Mean P(correct) (gen.)	33.99%	50.07%	36.46%	42.08%	62.65%	**72.45%**	61.55%	69.65%
ΔP(correct)	-54.49%	-38.41%	-48.96%	-43.34%	-32.07%	-22.27%	-27.93%	**-19.82%**
Prediction change rate	56.36%	34.55%	47.27%	49.09%	27.27%	25.45%	**16.36%**	**16.36%**

The results demonstrate that background context plays a significant role in visual recognition. Convolutional models are slightly more sensitive to contextual changes, whereas Vision Transformers (especially ViT-Base) are more robust and stable under these conditions.

Additional class-wise results are provided in Appendix section 4.3.4 (Table 7).

5.4 Object Rescaling Experiment

This experiment is based on the rescaling procedure defined in Section 4.3.4, which isolates object size while keeping the background constant through inpainting. Objects are extracted, resized, and reinserted into the original scene while keeping the background unchanged.

Results and Observations Table 2 summarizes the Top-1 accuracy across different object scales. Model performance decreases monotonically as object size is reduced. At the original scale, all models achieve high accuracy, with ViT-Base reaching 98.18%.

At 50% scale, performance drops substantially, with ResNet-18 decreasing to 50.91%, while ViT-Base maintains a higher accuracy of 83.64%. At 25% scale, degradation becomes severe across all models, with ResNet-18 dropping to 9.09% and ViT-Base achieving 32.73%.

Table 2: Top-1 results across object scales

Model	Original	Scale 100	Scale 75	Scale 50	Scale 25
ResNet-18	100,00%	89,09%	83,64%	50,91%	9,09%
ViT-Tiny	100,00%	92,73%	81,82%	61,82%	16,36%
ResNet-50	100,00%	92,73%	**92,73%**	76,36%	16,36%
ViT-Base	100,00%	**98,18%**	90,91%	**83,64%**	**32,73%**

Prediction confidence follows a similar trend. Table 3 reports the mean predicted probability assigned to the correct class across different object scaling conditions.

The largest decrease in prediction probability on original images to the scale of 25% is observed for ResNet-18 (-82.95%), while ViT-Base shows a smaller reduction (-68.24%). Interestingly, ResNet-50 yields the highest predicted probability for the original images and for scale factors of 100% and 75%. However, for more challenging scale factors (50% and 25%), ViT-Base exhibits greater stability.

Table 3: Mean predicted probability for the correct class across original and rescaled object conditions

Model	Original	Scale 100	Scale 75	Scale 50	Scale 25
ResNet-18	88.48%	82.97%	71.99%	40.84%	5.53%
ViT-Tiny	85.42%	78.81%	66.85%	47.44%	10.65%
ResNet-50	**94.72%**	**89.36%**	**86.18%**	66.01%	13.50%
ViT-Base	89.47%	84.38%	80.07%	**68.35%**	**21.23%**

The results indicate that object size strongly influences recognition performance. All models struggle with small-scale objects, although Vision Transformers exhibit greater robustness compared to convolutional models.

5.5 Online Data Augmentation for the Image Classification Task

In addition to evaluation experiments, we explored using the segmentation and manipulation pipeline *sam2aug* as an online data augmentation method during model training. The goal was to dynamically generate augmented samples by applying object-level transformations, such as relocation and rescaling, during training.

The pipeline requires object segmentation followed by background reconstruction using the LaMa inpainting model. However, LaMa operates on a single-image inference basis and does not support efficient batch processing. As a result, the augmentation

pipeline introduced significant computational overhead. In practice, the generation of augmented samples became a bottleneck, substantially increasing training times. Due to these constraints, this approach was not pursued in the main experimental evaluation.

5.6 Summary of Experimental Findings

Across all experiments, several consistent patterns emerge. Models exhibit a strong texture bias under standard conditions, which can be reduced through transformations that preserve structural information. Vision Transformer models consistently demonstrate higher shape bias and greater robustness than convolutional architectures. Furthermore, both contextual changes and object scale significantly affect model predictions, revealing limitations in generalization under distribution shifts.

These findings provide a comprehensive basis for interpreting model behavior and are further analyzed in the following discussion chapter.

6 Discussion

6.1 Overview of Findings

The objective of this thesis was to investigate whether object-centric augmentation can be used as a structured framework for analyzing model behavior under controlled visual manipulations. By combining segmentation, inpainting, and object-level transformations, the proposed *sam2aug* pipeline enables the systematic isolation of key visual factors, including texture, shape, context, and object scale.

Across all experiments, several consistent patterns can be observed. First, all evaluated models rely heavily on texture cues under standard conditions, consistent with prior work Geirhos et al. (2018). Second, this behavior can be systematically influenced through controlled transformations that preserve or emphasize structural information. Third, differences between model architectures emerge across all settings, with Vision Transformer models generally exhibiting greater robustness to the applied perturbations.

The following sections discuss these findings in more detail for each experimental setup.

Shape-Texture Bias and the Role of Structural Information The results of the shape-texture bias experiments provide clear evidence that all evaluated ImageNet-trained models are sensitive to texture manipulations. However, the degree and nature of this sensitivity vary substantially across architectures and transformation methods.

In the Texture Only condition, predictions are largely aligned with the texture label across all models, consistent with previously reported observations of texture bias in convolutional neural networks. At the same time, the results indicate that this behavior is not fixed. When additional structural information is introduced, the proportion of shape-consistent predictions increases. In the Texture + Edges condition, the inclusion of edge information leads to a noticeable shift toward shape-based decisions, suggesting that models retain sensitivity to structural cues but do not prioritize them under standard conditions.

A further shift is observed in the Texture NST condition, where optimization-based stylization preserves global structure while modifying surface appearance. In this setting, shape-consistent predictions become more frequent, and semantic alignment improves, indicating that preserving spatial structure is important for enabling shape-based recognition. In contrast, the Texture AdaIN condition leads to a different pattern. While the influence of texture is reduced, a substantial proportion of predictions falls into the "other" category, suggesting that feature-space stylization may disrupt class-discriminative information more strongly, thereby increasing uncertainty.

Overall, these observations suggest that model behavior depends not only on architecture but also on how visual information is presented. By controlling the balance between texture and structure, one can systematically influence prediction patterns.

Context Sensitivity and Distribution Shift The object relocation experiment highlights the influence of background context on model predictions. Across all models, relocating objects into novel environments reduces both accuracy and prediction confidence, indicating that models incorporate contextual information alongside object-specific features when making predictions. The observed differences across background types further indicate that semantically plausible contexts can partially support recognition, whereas less familiar contexts lead to greater performance degradation. The effect is more pronounced for convolutional models, which exhibit higher rates of prediction change under background shifts. Vision Transformers, especially ViT-Base, show comparatively greater stability, indicating a higher degree of invariance to contextual changes.

These findings suggest that distribution shifts in background context can significantly affect model behavior, even when the object itself remains unchanged.

Sensitivity to Object Scale The object rescaling experiment isolates the effect of object size while keeping all other factors constant. Across all models, accuracy decreases with object size, with a particularly pronounced decline at smaller scales.

This behavior indicates that model predictions depend on the availability of sufficient visual detail. As the object becomes smaller, relevant features may become less distinguishable, leading to reduced confidence and increased misclassification. ViT-Base again demonstrates greater robustness, maintaining higher accuracy and smaller confidence reductions than convolutional models. However, all models exhibit substantial performance degradation at the smallest scales, suggesting a general limitation in handling extreme variations in scale.

Architectural Differences: CNNs vs Vision Transformers Differences between convolutional neural networks and Vision Transformers are consistently observed across all experiments. Vision Transformer models, particularly ViT-Base, exhibit greater shape bias in the shape-texture experiments, suggesting a stronger reliance on structural information than convolutional models.

This difference is also reflected in the model robustness experiments. In the object relocation setting, convolutional models exhibit larger decreases in accuracy and higher rates of prediction change when objects are placed in novel environments. Vision Transformers, in contrast, maintain more stable predictions and exhibit smaller confidence changes. A similar trend is observed in the object rescaling experiment. While all models exhibit performance degradation as object size decreases, ViT-Base consistently achieves higher accuracy and smaller reductions in confidence than convolutional networks and ViT-Tiny.

These results are consistent with the interpretation that transformer-based architectures capture more global representations, whereas convolutional models rely more heavily on local features Naseer et al. (2021). As a result, Vision Transformers appear less sensitive to localized perturbations and contextual changes under the evaluated conditions.

Semantic Robustness and Parent-Level Evaluation The introduction of parent-level metrics provides additional insights into model behavior beyond strict top-1 accuracy. While fine-grained classification performance degrades under challenging conditions, models often continue to assign probability mass to semantically related classes, particularly in settings where structural information is preserved, such as the Texture NST condition. In these cases, parent-level accuracy and probability mass remain relatively high, indicating that models retain a degree of coarse semantic understanding even when exact class predictions are incorrect.

These observations suggest that standard accuracy metrics alone may not fully capture the extent to which models preserve meaningful information under perturbations. Evaluating performance at multiple levels of semantic granularity provides a more comprehensive view of model behavior.

6.2 Limitations

Several limitations should be considered when interpreting the results. The use of SAM 2 for segmentation requires bounding-box prompts or point coordinates, limiting the experiments to datasets with available annotations. In addition, segmentation quality depends on image characteristics, which influenced the selection of suitable samples.

The use of LaMa for inpainting introduced an additional constraint. Inpainting quality depends strongly on the amount of background context available after object removal. For this reason, images containing objects that occupied too large a fraction of the image could not be used reliably. In this work, this issue was addressed by restricting the bounding-box area to at most 75% of the image area. While this improved the quality of the inpainting results, it also reduced the pool of eligible images.

These technical constraints necessitated substantial manual filtering and quality control. Images and segmentation results were visually inspected, and only cases suitable for the ImageNet validation set were retained. This procedure inevitably introduced selection bias, as the final dataset does not represent a random sample of ImageNet images. Although the original-image subset was further constrained to include only images that all evaluated models correctly classified, this does not eliminate the bias introduced by manual selection and filtering. A similar limitation applies to the selection of donor textures. They were selected based on visual properties such as contrast and structure, and may not fully represent the diversity

of real-world textures. Furthermore, in the shape-texture experiments, the original background was preserved, meaning that additional contextual cues may have influenced model predictions. Finally, the LaMa inpainting model does not support efficient batch processing, which limits scalability and prevents integrating the pipeline as an online data augmentation method during training.

6.3 Future Work

Future work can address these limitations and extend the proposed framework. One direction is the use of segmentation methods that do not require explicit prompts (e.g., SAM 3 (Carion et al., 2025)), thereby enabling fully automated dataset generation. Another important extension is integrating more efficient inpainting models that support batch processing, enabling the incorporation of object-centric transformations into training pipelines as a form of data augmentation.

Future work could also explore a broader, more systematic selection of donor textures, for example, by defining explicit criteria for texture complexity, frequency content, or contrast statistics, thereby reducing manual bias in donor selection and enabling a more controlled analysis of how specific texture properties influence model behavior.

Further work could also experiment with more controlled manipulation of background context, for example, by removing or standardizing backgrounds, to better isolate object-intrinsic features. More generally, the current pipeline could be extended to other datasets, object categories, and transformation types, thereby helping determine to what extent the findings of this thesis generalize beyond the selected ImageNet subset and the specific manipulations studied here.

7 Conclusion

This thesis investigated whether segmentation-based data augmentation can be used as a structured framework for analyzing the behavior of modern image classification models under controlled visual manipulations. To this end, the task-agnostic object-centric *sam2aug* pipeline was developed to systematically generate datasets that isolate visual factors such as texture, shape, context, and object scale.

The results show that this approach provides a flexible tool for studying model behavior beyond standard benchmark evaluation. In particular, the experiments confirm that ImageNet-trained models rely heavily on texture cues and demonstrate that this behavior can be systematically influenced by transformations that preserve structural information.

Across all experiments, consistent differences between model architectures were observed. Vision Transformer models exhibit greater robustness and a higher reliance on structural information compared to convolutional neural networks, indicating differences in how visual information is represented.

In addition, the experiments highlight limitations in robustness to contextual and geometric changes. Model predictions are affected by background shifts and object-scale variations, whereas parent-level evaluation shows that models often retain coarse semantic understanding even when fine-grained predictions fail.

Overall, this thesis contributes both a modular framework for object-centric data augmentation and an empirical analysis of model behavior under controlled visual manipulations. The findings provide a basis for future work to improve robustness and to better understand decision-making in modern computer vision systems.

A Appendix

This appendix provides detailed analyses that complement the main results, including class-level evaluations, texture-specific effects, and additional evaluation metrics.

A.1 Image Selection Details

A.1.1 Inpainting Results Without Mask Dilation

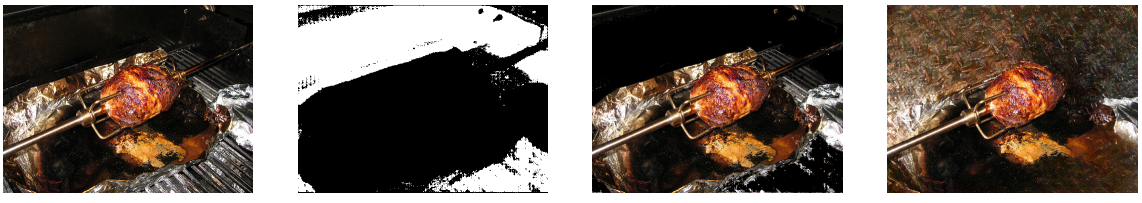
Inpainting without mask dilation can lead to unexpected results and hallucinations of a generative model that reconstructs the image using residual parts of the object (Figure 12).

(a) (b) (c)

Figure 12: Sample image with inpainting results without mask dilation: (a) segmented object, (b) background with cut-out object, (c) inpainted background without dilation of the segmented mask.

A.1.2 Samples of Excluded Classes

The class “oven” has been excluded from the experiments in this thesis due to the poor quality of the resulting segmentation masks. As illustrated in the Figure 13, the object is not fully separated from the original image, resulting in residual object fragments that leak into the background.



(a) (b) (c) (d)

Figure 13: Samples of the generated masks for images from the class “oven”: (a) original image, (b) segmentation mask, (c) background with cut-out object, (d) inpainted background.

A.1.3 Possible Object Truncation by Relocation

One of the constraints for image selection was that the bounding box intersected the image boundary in a way that suggested truncation. That helped avoid additional potential bias in model predictions, as truncated relocated parts can mislead the classification. In the Figure 14, some examples of truncated objects are shown.

Figure 14: Possible object truncation after relocation.

A.1.4 List of Selected Classes for Experiments

For the texture-shape bias experiment, we selected the following sub-classes out of 15 parent categories (Table 4). Description of the classes is obtained from the dictionary (Polhamus, 2023). For identifying a single class, the WordNet ID (WNID) was used. Due to the image selection constraints, it was not always possible to obtain 10 images per class for all classes. These are two classes that were left with 7 images each: *knife* and *bottle*.

Table 5 summarizes the subset of classes used for object relocation and rescaling experiments.

The full list of selected single images, including prediction results and bounding box area information that led to the selection, is available in the GitHub repository.

A.2 Additional Experimental Results

This chapter of the Appendix provides additional experimental results that complement the main findings presented in the core chapters of this thesis. The supplementary analyses include extended quantitative evaluations, detailed breakdowns by model architecture and class, and additional visual examples. These results are intended to provide deeper insights into model behavior under varying transformations. While not central to the main narrative, the presented results support and further substantiate the conclusions drawn in the primary analysis.

A.2.1 Shape-Texture Bias Experiment

Cross-Dataset Comparison ***Shape Bias (Top-5)*** Top-5 shape bias follows the same trend as top-1, with consistently higher values across all models and conditions.

Table 4: Selected classes and number of images for the shape-texture bias experiment

WNID	Class name	Parent category	Amount of images
n02690373	Airliner	Airplane	10
n02132136	Brown bear	Bear	10
n02835271	Tandem bicycle	Bicycle	10
n01514859	Hen	Bird	10
n02981792	Catamaran	Boat	10
n02823428	Beer bottle	Bottle	7
n03100240	Convertible	Car	10
n02123045	Tabby	Cat	10
n04099969	Rocking chair	Chair	10
n02708093	Analog clock	Clock	10
n02090622	Borzoi	Dog	10
n02504458	African elephant	Elephant	10
n03085013	Computer keyboard	Keyboard	10
n03041632	Cleaver	Knife	7
n03417042	Garbage truck	Truck	10

Table 5: Selected classes and number of images for object relocation with background shift and object rescaling experiments

WNID	Class name	Parent category	Amount of images
n02690373	Airliner	Airplane	5
n02835271	Tandem bicycle	Bicycle	5
n01514859	Hen	Bird	5
n02981792	Catamaran	Boat	5
n03100240	Convertible	Car	5
n02123045	Tabby	Cat	5
n04099969	Rocking chair	Chair	5
n02708093	Analog clock	Clock	5
n02090622	Borzoi	Dog	5
n02504458	African elephant	Elephant	5
n03417042	Garbage truck	Truck	5

Structural transformations (Texture + Edges, NST, AdaIN) substantially increase shape bias, with the highest values observed for ViT-Base in the stylized datasets (Figure 15).

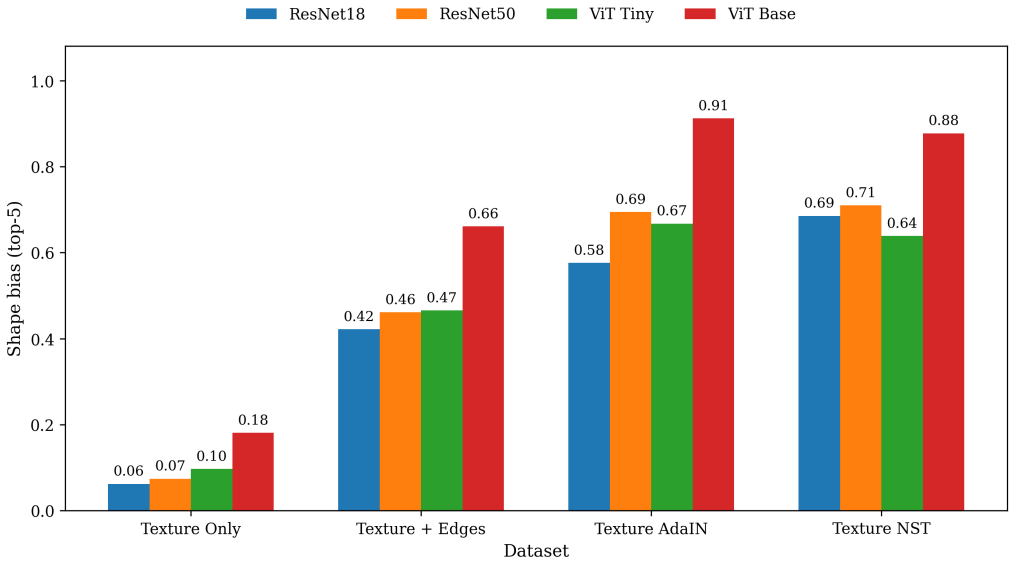


Figure 15: Shape bias computed from top-5 predictions across all dataset variants (x-axis) and different models (bar color).

Class-Level Analysis To provide a more compact and interpretable overview of model behavior, we analyze class-level shape bias aggregated across all donor textures (Figures 16 – 19). This representation summarizes how consistently each model relies on shape information for a given object class based on the model's top-1 predictions.

In the Texture Only condition, shape-consistent predictions are largely suppressed across most classes (Figure 16). Only a small number of structurally distinctive classes, such as rocking chair and catamaran, retain moderate shape bias. This confirms that replacing object appearance with texture strongly shifts model decisions toward texture cues.

Introducing structural information in the Texture + Edges condition leads to a noticeable increase in shape bias for several classes (Figure 17). In particular, classes with distinctive geometric structure, such as tandem bicycle and rocking chair, show strong recovery of shape-consistent predictions. However, variability across classes remains, indicating that edge information only partially restores shape-based recognition.

In the Texture NST condition, shape bias increases substantially across most classes (Figure 18). Many classes achieve high shape-consistent prediction rates, especially for Vision Transformer models. This suggests that optimization-based stylization preserves global structure sufficiently to support shape-based recognition, while reducing the influence of texture.

The Texture AdaIN condition exhibits a different pattern (Figure 19). While shape bias remains high for several classes, predictions are less consistent across models and classes. In particular, some classes show increased variability, reflecting the fact that feature-based stylization can alter both texture and structural cues.

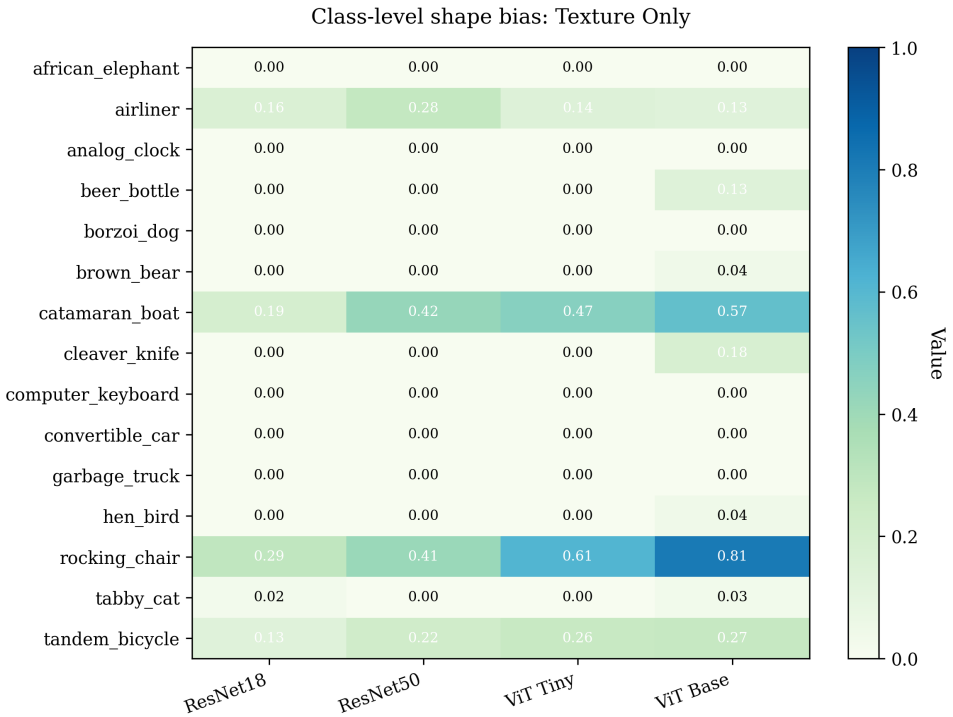


Figure 16: Class-level shape bias in the Texture Only dataset. Values represent the proportion of shape-consistent top-1 predictions aggregated across all donor textures.

Across all conditions, Vision Transformer models - especially ViT-Base - consistently achieve higher shape bias across classes compared to convolutional networks. This indicates a stronger reliance on global structure and greater robustness to texture perturbations.

Semantic Robustness: Parent-Level Metrics To complement the fine-grained evaluation, model predictions were also evaluated at the parent-category level, providing a more coarse-grained measure of semantic correctness (Table 6). We highlight in bold the highest accuracy values and the lowest error values.

Parent-level metrics follow the same trend as the class-instance shape predictions presented in Chapter 5 (Figure 10), with the lowest shape-level performance in the Texture Only condition and the highest in the Texture NST condition, indicating improved coarse semantic alignment under structurally preserved transformations. Similar to the main top-1 and top-5 results, we observe that ViT-Base achieves the highest performance across all datasets, predicting labels more often that are similar to the original ”shape” class.

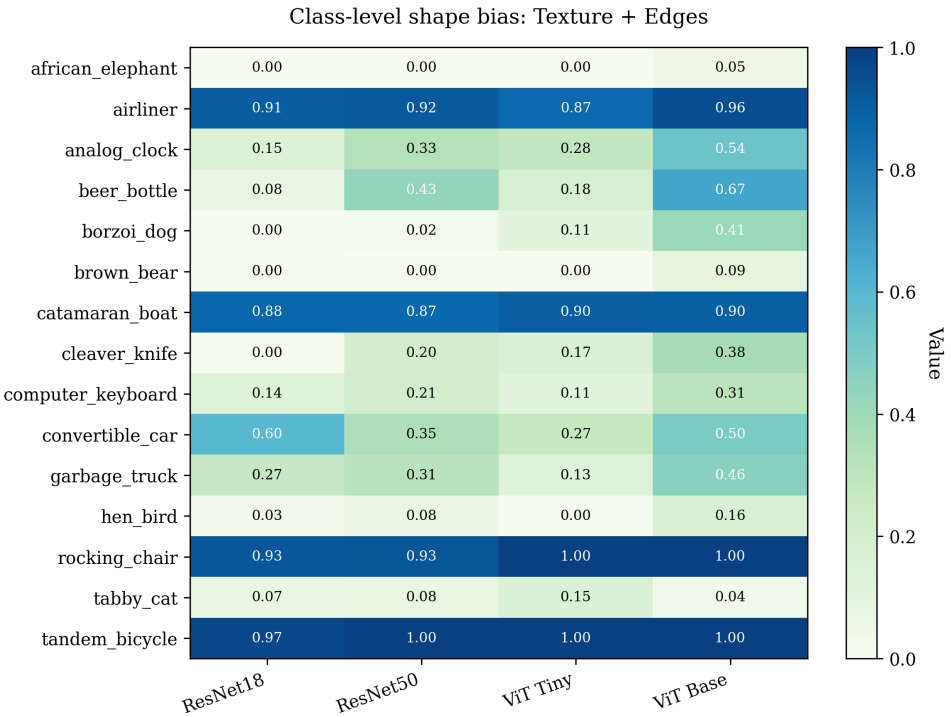


Figure 17: Class-level shape bias in the Texture + Edges dataset.

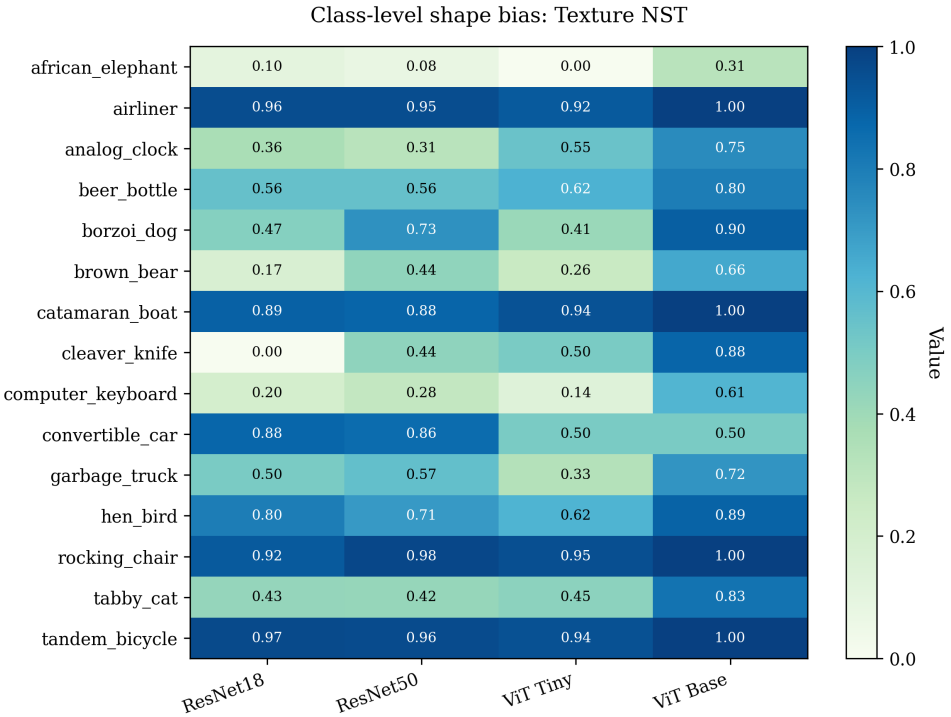


Figure 18: Class-level shape bias in the Texture NST dataset.

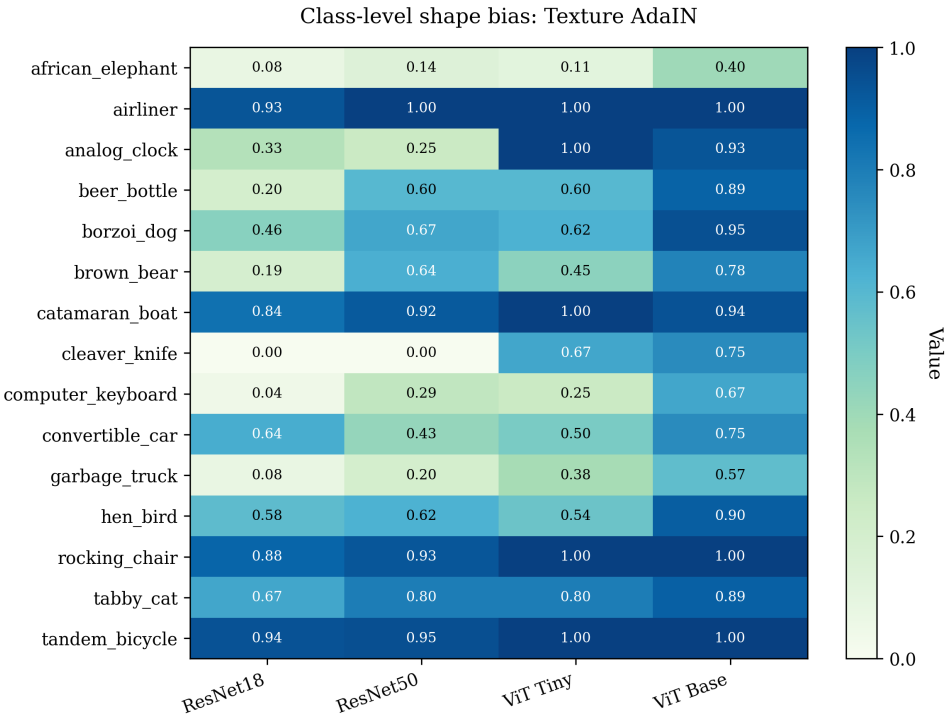


Figure 19: Class-level shape bias in the Texture AdaIN dataset.

A.2.2 Object Relocation with Background Shift Experiment

Class-Level Analysis To further analyze model behavior under contextual changes, we evaluate performance at the class level.

Class-wise accuracies were computed by grouping generated samples by object class and model, and averaging correctness indicators across all generated instances of the respective class. Since background type was not included in the grouping, the reported class-wise results aggregate performance across both evaluated relocation backgrounds.

Table 7 reports class-wise top-1 accuracy and the change in predicted probability for the correct class for the object rescaling experiment across all evaluated model architectures. Again, in bold we highlight the highest values for the accuracy metric and the lowest for the error-based metric.

Top-1 Accuracy by Class The distribution of top-1 accuracy across object classes is shown in Table 7. Performance varies substantially across classes and models. Several classes achieve consistently high accuracy across models. For example, *hen bird* and *rocking chair* exhibit high top-1 accuracy, with multiple models achieving 90–100%. In contrast, some classes exhibit lower performance. The class airliner shows strong variability across models, ranging from 40% (ResNet-18) to 100% (ViT-Base). Similarly, class *garbage truck* exhibits large differences, with ViT-Base achieving 100%, while ViT-Tiny reaches 0%. *tabby cat* class shows generally higher

Table 6: Parent-category-based evaluation metrics across shape-texture dataset variants and model architectures

Dataset Variant	Model	Parent Top-1	Parent Top-5	Parent Probability Mass
Texture Only	ResNet-18	6,10%	19,30%	4,80%
Texture Only	ViT-Tiny	7,50%	17,20%	6,60%
Texture Only	ResNet-50	7,80%	25,30%	6,80%
Texture Only	ViT-Base	**15,60%**	**33,20%**	**13,50%**
Texture + Edges	ResNet-18	28,60%	50,10%	24,40%
Texture + Edges	ViT-Tiny	26,00%	46,80%	23,30%
Texture + Edges	ResNet-50	35,60%	57,20%	31,20%
Texture + Edges	ViT-Base	**47,20%**	**75,00%**	**42,00%**
Texture AdaIN	ResNet-18	26,20%	46,20%	21,10%
Texture AdaIN	ViT-Tiny	28,30%	46,20%	23,60%
Texture AdaIN	ResNet-50	30,60%	53,60%	25,30%
Texture AdaIN	ViT-Base	**52,80%**	**79,20%**	**44,80%**
Texture NST	ResNet-18	37,80%	63,10%	31,60%
Texture NST	ViT-Tiny	34,20%	58,20%	29,30%
Texture NST	ResNet-50	42,60%	66,90%	38,40%
Texture NST	ViT-Base	**65,70%**	**86,50%**	**57,20%**

accuracy for CNN-based models, with ResNet-18 achieving 30% and 10%, respectively, while Vision Transformers have 0%.

Confidence Change Across all classes and models, $\Delta P(\text{correct})$ is predominantly negative, indicating a decrease in confidence after object relocation. The magnitude of this decrease varies substantially across classes. Some classes show relatively small decreases, such as *hen bird* and *rocking chair*, where values remain closer to zero. In contrast, larger decreases are observed for classes such as *airliner*, *garbage truck*, and *borzoi dog*, where $\Delta P(\text{correct})$ reaches values below -0.6 for several models and even -95 for ViT-Tiny for the *garbage truck* class.

Across models, ResNet-18 and ViT-Tiny generally exhibits the largest decreases in probability, while ViT-Base shows smaller reductions for many classes.

A.3 Image Rights

Texture images for the shape-texture bias experiment (Figure 5) and background images for the background shift experiment (Figure 7) were obtained from Unsplash Unsplash (2026a). All images are distributed under the Unsplash License Un-

Table 7: Class-wise Top-1 accuracy and ΔP(correct) for the object rescaling experiment across different model architectures

	ResNet-18		ViT-Tiny		ResNet-50		ViT-Base	
Class	Top-1	ΔP(correct)	Top-1	ΔP(correct)	Top-1	ΔP(correct)	Top-1	ΔP(correct)
African elephant	70.00%	-34.33%	70.00%	**-17.56%**	**100.00%**	**3.38%**	80.00%	-22.78%
Airliner	40.00%	-75.23%	10.00%	-80.63%	50.00%	-51.10%	**100.00%**	-25.36%
Analog clock	10.00%	-50.82%	80.00%	-22.02%	50.00%	-50.01%	**100.00%**	-13.89%
Borzoi dog	40.00%	-69.03%	70.00%	-59.97%	90.00%	-21.51%	80.00%	-24.78%
Catamaran boat	80.00%	-31.92%	50.00%	-36.93%	50.00%	-54.98%	70.00%	-42.08%
Convertible car	60.00%	-51.72%	40.00%	-43.42%	90.00%	-14.15%	**100.00%**	-23.62%
Garbage truck	30.00%	-75.06%	0.00%	-95.03%	80.00%	-34.16%	**100.00%**	-27.96%
Hen bird	**90.00%**	**-17.98%**	**100.00%**	-20.13%	90.00%	-8.61%	**100.00%**	-4.94%
Rocking chair	80.00%	-28.91%	90.00%	-18.01%	**100.00%**	-6.66%	**100.00%**	**-2.46%**
Tabby cat	30.00%	-35.02%	0.00%	-60.28%	30.00%	-39.73%	0.00%	-47.70%
Tandem bicycle	70.00%	-40.93%	60.00%	-53.67%	80.00%	-21.36%	90.00%	-27.06%

splash (2026b), which allows usage free of charge for both commercial and non-commercial purposes.

The individual image sources are documented in the bibliography: Figure 5 (left to right): peacock feathers Mozhvilo (2024), tiger texture Maltsev (2025), zebra texture YAICH (2018), chain mail texture 2H Media (2023), honeycomb texture Hamersmit (2021); Figure 7 background images (left to right): open-water background Giglia (2022), and water-surface background yucar studios (2016).

Bibliography

2H Media. Eine nahaufnahme eines maschendrahtzauns. `https://unsplash.com/photos/EZ2lyadsokM`, 2023. Unsplash image, published on 2023-04-14. Accessed: 2026-04-11.

Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 33(5):898–916, 2010.

Tom Burgert, Oliver Stoll, Paolo Rota, and Begüm Demir. Imagenet-trained cnns are not biased towards texture: Revisiting feature reliance through controlled suppression. *arXiv preprint arXiv:2509.20234*, 2025.

Qiang Cai, Mengxu Ma, Chen Wang, and Haisheng Li. Image neural style transfer: A review. *Computers and Electrical Engineering*, 108:108723, 2023.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, et al. Sam 3: Segment anything with concepts. *arXiv preprint arXiv:2511.16719*, 2025.

Tian Qi Chen and Mark Schmidt. Fast patch-based style transfer of arbitrary style. *arXiv preprint arXiv:1612.04337*, 2016.

Antonio Criminisi, Patrick Pérez, and Kentaro Toyama. Region filling and object removal by exemplar-based image inpainting. *IEEE Transactions on image processing*, 13(9):1200–1212, 2004.

Yingying Deng, Fan Tang, Weiming Dong, Chongyang Ma, Xingjia Pan, Lei Wang, and Changsheng Xu. Stytr2: Image style transfer with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11326–11336, 2022.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. A learned representation for artistic style. *arXiv preprint arXiv:1610.07629*, 2016.

Leon A Gatys, Alexander S Ecker, and Matthias Bethge. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*, 2015.

Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2414–2423, 2016.

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.

Elena Giglia. Hintergrundmuster. `https://unsplash.com/photos/jdwwGGTCe74`, 2022. Unsplash image, published on 2022-03-11. Accessed: 2026-04-11.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

Ante Hamersmit. Braun-weisses quadratisches muster. `https://unsplash.com/photos/gi1f13S1-64`, 2021. Unsplash image, published on 2021-07-14. Accessed: 2026-04-11.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE international conference on computer vision*, pages 1501–1510, 2017.

Yongcheng Jing, Yezhou Yang, Zunlei Feng, Jingwen Ye, Yizhou Yu, and Mingli Song. Neural style transfer: A review, 2018. URL `https://arxiv.org/abs/1705.04058`.

Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016.

Nick Kanopoulos, Nagesh Vasanthavada, and Robert L Baker. Design of an image edge detection filter using the sobel operator. *IEEE Journal of solid-state circuits*, 23(2):358–367, 1988.

Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *International journal of computer vision*, 1(4):321–331, 1988.

Kunal Kawadkar. Comparative analysis of vision transformers and convolutional neural networks for medical image classification. *arXiv preprint arXiv:2507.21156*, 2025.

Pum Jun Kim, Seung-Ah Lee, Seongho Park, Dongyoon Han, and Jaejun Yoo. On the reliability of cue conflict and beyond, 2026. URL `https://arxiv.org/abs/2603.10834`.

Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9404–9413, 2019.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4015–4026, 2023.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

Teerath Kumar, Rob Brennan, Alessandra Mileo, and Malika Bendechache. Image data augmentation approaches: A comprehensive survey and future directions. *Ieee Access*, 12:187536–187571, 2024.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521 (7553):436–444, 2015.

Yanghao Li, Naiyan Wang, Jiaying Liu, and Xiaodi Hou. Demystifying neural style transfer. *arXiv preprint arXiv:1701.01036*, 2017a.

Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. Universal style transfer via feature transforms. *Advances in neural information processing systems*, 30, 2017b.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. Exploring the limits of weakly supervised pretraining. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

Anatoly Maltsev. Nahaufnahme von tigerfell mit orangen und schwarzen streifen. `https://unsplash.com/photos/Zlh7phQ_-hg`, 2025. Unsplash image, published on 2025-11-23. Accessed: 2026-04-11.

George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

Shervin Minaee, Yuri Boykov, Fatih Porikli, Antonio Plaza, Nasser Kehtarnavaz, and Demetri Terzopoulos. Image segmentation using deep learning: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(7):3523–3542, 2021.

Elena Mozhvilo. Eine nahaufnahme der federn eines pfaus. `https://unsplash.com/photos/lESq5T4NApY`, 2024. Unsplash image, published on 2024-01-27. Accessed: 2026-04-11.

Muhammad Muzammal Naseer, Kanchana Ranasinghe, Salman H Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Intriguing properties of vision transformers. *Advances in Neural Information Processing Systems*, 34: 23296–23308, 2021.

Humza Naveed, Saeed Anwar, Munawar Hayat, Kashif Javed, and Ajmal Mian. Survey: Image mixing and deleting for data augmentation. *Engineering applications of artificial intelligence*, 131:107791, 2024.

Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2536–2544, 2016.

Luis Perez and Jason Wang. The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint arXiv:1712.04621*, 2017.

Aaron Polhamus. Github gist: ¡insert title¿, 2023. URL `https://gist.github.com/aaronpolhamus/964a4411c0906315deb9f4a3723aac57`. Accessed: 2026-04-11.

Weize Quan, Jiaxi Chen, Yanli Liu, Dong-Ming Yan, and Peter Wonka. Deep learning-based image and video inpainting: A survey. *International Journal of Computer Vision*, 132(7):2367–2400, 2024.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. Sam 2: Segment anything in images and videos, 2024. URL `https://arxiv.org/abs/2408.00714`.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of big data*, 6(1):1–48, 2019.

Jamie Shotton, John Winn, Carsten Rother, and Antonio Criminisi. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *European conference on computer vision*, pages 1–15. Springer, 2006.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Farhana Sultana, Abu Sufian, and Paramartha Dutta. Evolution of image segmentation using deep convolutional neural network: A survey. *Knowledge-Based Systems*, 201:106062, 2020.

Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky. Resolution-robust large mask inpainting with fourier convolutions, 2021. URL `https://arxiv.org/abs/2109.07161`.

Satoshi Takahashi, Yusuke Sakaguchi, Nobuji Kouno, Ken Takasawa, Kenichi Ishizu, Yu Akagi, Rina Aoyama, Naoki Teraya, Amina Bolatkan, Norio Shinkai, et al. Comparison of vision transformers and convolutional neural networks in medical image analysis: a systematic review. *Journal of Medical Systems*, 48(1):84, 2024.

Alexa R. Tartaglini, Wai Keen Vong, and Brenden M. Lake. A developmentally-inspired examination of shape versus texture bias in machines, 2022. URL `https://arxiv.org/abs/2202.08340`.

Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6924–6932, 2017.

Unsplash. Unsplash: Free high-resolution photos, 2026a. URL `https://unsplash.com`. Accessed: 2026-04-11.

Unsplash. Unsplash license, 2026b. URL `https://unsplash.com/license`. Accessed: 2026-03-14.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Zishan Xu, Xiaofeng Zhang, Wei Chen, Minda Yao, Jueting Liu, Tingting Xu, and Zehua Wang. A review of image inpainting methods based on deep learning. *Applied sciences*, 13(20):11189, 2023.

Stephane YAICH. Schwarz-weisses zebrafell. `https://unsplash.com/photos/-QXDI1COGMA`, 2018. Unsplash image, published on 2018-10-17. Accessed: 2026-04-11.

yucar studios. Blaues meerwasser während des tages. `https://unsplash.com/photos/cfR-V1QuEKw`, 2016. Unsplash image, published on 2016-03-05. Accessed: 2026-04-11.

Yuxin Zhang, Nisha Huang, Fan Tang, Haibin Huang, Chongyang Ma, Weiming Dong, and Changsheng Xu. Inversion-based style transfer with diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10146–10156, 2023.

Declaration of Authorship

Ich erkläre hiermit gemäß §9 Abs. 12 APO, dass ich die vorstehende Abschlussarbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Des Weiteren erkläre ich, dass die digitale Fassung der gedruckten Ausfertigung der Abschlussarbeit ausnahmslos in Inhalt und Wortlaut entspricht und zur Kenntnis genommen wurde, dass diese digitale Fassung einer durch Software unterstützten, anonymisierten Prüfung auf Plagiate unterzogen werden kann.

Bamberg, 13.04.2026

Place, Date

Signature