Eberhard Karls Universität Tübingen
Mathematisch-Naturwissenschaftliche Fakultät
Wilhelm-Schickard-Institut für Informatik

Bachelor Thesis Computer Science

Exploring the Fréchet Inception Distance: Mitigating Inception-v3´s Small-Sample Bias and Utilizing Custom Models for Comparing Image Distributions

Nina Schumacher

13.03.2025

Supervisor

Prof. Dr. Philipp Hennig
Department of Computer Science
University of Tübingen

Schumacher, Nina:
Exploring the Fréchet Inception Distance: Mitigating Inception-v3´s Small-Sample Bias and Utilizing Custom Models for Comparing Image Distributions
Bachelor Thesis Computer Science
Eberhard Karls Universität Tübingen
Period: 13.11.2024-13.03.2025

Abstract

This bachelor‘s thesis delves into the Fréchet Inception Distance (FID) metric and critically evaluates its limits. Based on experiments with different custom models this work explores whether a Fréchet Distance based on these can be a valuable alternative to FID for datasets that are very different to ImageNet. Additionally, in this study it is investigated what impact small sample sizes have on FID determination and how one can prevent the negative effects.
The experiments showed that indeed it can make sense to train a custom model for very specific datasets. Then the embedding space of the model can be used to implement a custom Fréchet Distance. This metric can be an opportunity to achieve reliable results and opens up a path for future research when working with niche data.
It was also found that the downsizing of covariance matrices through reducing the feature dimension can mitigate the bias associated with small sample sizes.
Overall this study demonstrates that while FID is a good metric most of the time, one should be aware of its limitations, make use of it with caution and where applicable create a custom Fréchet Distance for evaluation as introduced in this paper.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Consider the scenario of an art gallery with two seemingly identical art collections – one is a collection of Van Gogh's greatest paintings, the other consists only of AI-generated replicas. While they appear visually similar, subtle differences exist. How can these differences be quantified?
This is where the concept of distance comes into play – not the physical kind, but a mathematical abstraction that helps us deal with huge, often vague areas of data.

Distances are the unseen but crucial part of machine learning and data science. They allow us to compare, contrast, and categorize. They tell us how similar two probability densities in a high-dimensional space are. But distances are more than just numbers; they are the bridges between theory and practice, the tools that let us evaluate how well a model performs, how close a generated image is to a real one, or how similar two pieces of music are. Yet, not all distances are created equal. Some are simple, like the Euclidean distance which students may be familiar from their geometry classes. Others are more advanced, designed to capture the small details of complex data. But here's the catch: the choice of distance metric isn't just a technical decision – it's a philosophical one. It affects how we define similarity, how we judge quality, and how we understand the world through the lens of AI.

The Fréchet Inception Distance (FID) has emerged as the prevailing metric in the evaluation of generative models, particularly in the domain of image synthesis, thereby establishing itself as the gold standard. At its core, FID measures the distance between two distributions of images – real and generated – by utilizing the embedding spaces of neural networks; it uses the Inception Network, a creation of Google, trained on the ImageNet dataset. The Inception Network is known for being a leading force in modern AI – a black box we trust to tell us how real our generated images are. However, the validity of Inception must be considered, because it is the foundation of FID. But what if there are biases, limitations, or blind spots?

This is the point at which the narrative becomes more complex. FID is state-of-the-art, widely adopted, and often used without question. But as with any tool, blind reliance can lead to oversights due to its dependence on the environment in which it was trained. And yet, we use it as the backbone of FID, trusting it to judge the quality of images in domains it may never have seen before. This raises the question of whether the reliance on the Inception Network introduces biases that affect the validity of FID as a metric. This makes FID more than just a technical measurement. It is more about deciding how we use them.

That is why in this thesis, I will explore the domain of distances, with a focus on the emergence of FID and a critical examination of the assumptions that underpin it. The central question guiding this inquiry is: In what ways might a reliable distance measurement be rendered unreliable? How does FID work, and why has it become so popular? And perhaps most importantly, what are we missing when we take it for granted? I hope that this thesis does not only help to understand FID better but also starts a conversation about not just which tools we should use, but also about how we use them.

Before delving into the critical discussion, it is essential to consider the theoretical underpinnings of FID and Inception-v3. An important part is the identification of the limitations of the Fréchet Inception Distance. Then, the potential for FID to be converted into a more interpretable metric in certain scenarios is discussed in Chapters 3 to 5.
Chapter 3 is about the architecture of custom models and their performance in classification on CIFAR-6. In the fourth chapter, I defined a custom Fréchet Distance which uses the embedding spaces of the custom models. This custom distance is then used to quantify whether it can be useful to train custom models for very specific datasets in order to perform reliable similarity calculations.
The fifth chapter then deals with whether the reduction of the feature dimension in Inception-v3 can lead to a more precise determination of the FID for small sample sizes.
In the following discussion, the example of FID is used to provide a deeper understanding of the application of metrics, but also to highlight the need for a critical approach.

Chapter 2

Fundamentals of FID and InceptionV3

In order to understand the scope of the Fréchet Inception Distance (FID), it is essential to consider its mathematical and algorithmic foundations. FID does not compare individual images directly, but rather it compares the latent representations induced by neural networks. This methodology is based on deep neural networks. The Inception-v3 Network, which was trained on the ImageNet database and serves as a universal evaluation standard for image similarity, plays a central role in this. But how exactly does this comparison work? What are the underlying assumptions of FID? And which theoretical concepts from statistics and machine vision form its basis? This section systematically explains the mathematical structure of FID, its implementation, and its connection to the Inception-v3 network.
In the first section, I will discuss the feature extractor of the Fréchet Inception Distance – Inception-v3. Here, both the architecture and the comparison with other networks will be looked at. Then I will explain the mathematical background of FID and how FID scores are evaluated. Finally, I will indicate the biases and limitations of FID.

2.1 The Inception-v3 Network as a Feature Extractor

Before analyzing FID in detail, it is essential to understand the Inception-v3 network on which the metric is based. Introduced by Szegedy et al. (2016) Inception-v3 is a Convolutional Neural Network (CNN) developed by Google in 2015 to push the boundaries of image classification on the ImageNet dataset. It is the third iteration of the Inception series and aachieved state-of-the-art results at that time utilizing a convolutional-based architecture. In this section, we look at its architecture and its training basis. Thereby it is hoped to understand its ability to extract high-dimensional image features-a feature that makes it the basis for FID.

2.1.1 Basic Architecture

As proposed by Szegedy et al. (2016) the Inception-v3 architecture (see Table 2.1) is structured into three phases:

Stem Block: Reduces input resolution aggressively by using stacked convolutions.

- Input: $299 \times 299 \times 3 \rightarrow$ Output: $35 \times 35 \times 192$

Inception Blocks:

- $3\times$ Inception: Processes 35×35 feature maps with parallel 1×1, 3×3 and 5×5 factorized convolutions
- $5\times$ Inception: Operates on 17×17 maps using asymmetric $1 \times 7 + 7 \times 1$ convolutions
- $2\times$ Inception: Final 8×8 blocks with expanded filter banks for high-dimensional representations

Classifier: Global average pooling followed by a softmax layer.

Type	Patch Size/Stride or Remarks	Input size
conv	$3 \times 3/2$	$299 \times 299 \times 3$
conv	$3 \times 3/1$	$149 \times 149 \times 32$
conv padded	$3 \times 3/1$	$147 \times 147 \times 32$
pool	$3 \times 3/2$	$147 \times 147 \times 64$
conv	$3 \times 3/1$	$73 \times 73 \times 64$
conv	$3 \times 3/2$	$71 \times 71 \times 80$
conv	$3 \times 3/1$	$35 \times 35 \times 192$
3×Inception	see 'Inception Blocks' in 2.1.1	$35 \times 35 \times 288$
5×Inception	see 'Inception Blocks' in 2.1.1	$17 \times 17 \times 768$
2×Inception	see 'Inception Blocks' in 2.1.1	$8 \times 8 \times 1280$
pool	8×8	$8 \times 8 \times 2048$
linear	logits	$1 \times 1 \times 2048$
softmax	classifier	$1 \times 1 \times 1000$

Table 2.1: Inception-v3 architecture adapted from Szegedy et al. (2016, p. 6)

2.1.2 Architectural Innovations

The architecture of Inception-v3 takes advantage of factorization into smaller convolutions, spatial factorization into asymmetric convolutions and the use of Auxiliary Classifiers (for stabilization and regularization during training) in order to become more efficient than its predecessors (Szegedy et al., 2016). The reduction of grid sizes is also an architectural innovation, which will be explained in more detail:

To address the representational bottlenecks induced by pooling, parallel reduction paths are used (Szegedy et al., 2016).
As mentioned by Szegedy et al. (2016) traditional CNNs often reduce the spatial resolution of feature maps by simple pooling (e.g. max-pooling) or convolutions with Stride 2. However, these methods lead to two critical problems:

- Representational bottlenecks:
  If the resolution of e.g. $d \times d$ to $\frac{d}{2} \times \frac{d}{2}$ does not increase the number of filters k, an information bottleneck occurs. The reduced dimension $(\frac{d}{2})^2 \cdot k$ can lose critical details, which reduces the expressive power of the network.

- High computational costs:
  To avoid bottlenecks, a convolution with double the number of filters ($2k$) could be used before pooling. However, this is extremely computationally intensive:
  $$FLOPs = 2d^2k^2,$$
  $$\textit{i.e. } 2.1 \times 10^9 \textit{ operations for } d = 35, k = 288$$

To solve these problems parallel paths are used in Inception-v3.

- Path 1 (Convolution):
  - A 3×3 convolution with stride 2 and half the number of filters (k).
  - Compresses the feature map to $\frac{d}{2} \times \frac{d}{2}$ and learns spatial patterns.
  - Example: Reduces $35 \times 35 \times 288$ to $17 \times 17 \times 288$.

- Path 2 (Pooling):
  - A 3×3 max pooling layer with stride 2.
  - Preserves the most important activations of the original feature map.
  - Cost: Practically negligible, as no parameters are trained.

- Concatenation:
  - The outputs of both paths ($17 \times 17 \times 288 + 17 \times 17 \times 288$) are combined in $17 \times 17 \times 768$.
  - This results in the number of filters being doubled, while the spatial resolution is halved - without any loss of information.

The application of these parallel paths eliminates the bottleneck, due to the combination of convolution and pooling, preserving the information density.
It is just as efficient, as the expensive convolution is only performed with half the number of filters (k) and max-pooling is computationally almost free of charge. In the end it is more flexible than naive downsampling (Szegedy et al., 2016).

2.1.3 Training

The relevant features as well as the setting for the training of Inception-v3 are listed below (Szegedy et al., 2016).

- **Batch Normalization**: Normalizes the output of each layer to stabilize the training. This means that the network learns faster and is less susceptible to poor initialization.
- **Optimizer**: RMSProp, an algorithm that automatically adjusts the learning rate. One advantage of this optimizer is that it learns faster in areas with small gradients and slows down with large gradients (Kingma & Ba, 2017).
- **Label Smoothing**: Instead of i.e. '100% cat', the label is smoothed to '90% cat', 10% evenly distributed to other classes". This prevents the network from becoming too arrogant (overconfidence), which leads to overfitting.
- **Gradient clipping** introduced by Pascanu et al. (2013) limits the size of the gradients to a maximum value (e.g., 2.0). This is used to prevent the training from becoming unstable due to extremely large gradients.
- **Training data and hardware**
  - Data: ImageNet (1.2 million images, 1000 classes).
  - Batch size: 32 images per GPU, distributed over 50 GPUs.
  - Epochs: 100 runs through the dataset.

2.1.4 Inception-v3 in Comparison

In comparison to ResNet-50 (He et al., 2015), Inception-v3 (Szegedy et al., 2016) is similarly efficient: With an alike Top-1 accuracy (78.8% vs. 76.0% Top-1 accuracy, see Table 2.2) Inception-v3 needs 5.7 billion FLOPs, while ResNet-50 needs about 4.1 billion. Modern models, on the other hand, are more accurate, but also more expensive: EfficientNet-B7 (Tan & Le, 2020) achieves a high Top-1 accuracy, just like Deit-B (Touvron et al., 2021), but EfficientNet-B7 has almost 2.8 times as many parameters as Inception-v3 (66 million parameters) and DeiT-B even almost 3.6 times as many (approx. 86 million parameters).
Inception-v3 is a compromise: It combines moderate accuracy with acceptable computational effort.

Model	**Top-1 Accuracy**
Inception-v3	78.8 %
ResNet-50	76.0 %
EfficientNet-B7	84.3 %
DeiT-B	85.2 %

Table 2.2: Performance Comparison on ImageNet due to each Top-1 Accuracy.

2.2 The Fréchet Inception Distance (FID): Calculation and Interpretation

The Fréchet Distance, which was first established by Dowson and Landau (1982) and then as Fréchet Inception Distance initially used by Heusel et al. (2017) as a metric 'for comparing the results of GANs' (Heusel et al., 2017, p. 11) uses the feature representations from Inception-v3 to measure the statistical distance between real and generated image distributions. The reason for comparing embedding spaces is because Inception has learned useful abstractions. A foundational assumption of FID is that the feature embeddings extracted from both real and generated images follow multivariate Gaussian distributions (Heusel et al., 2017). This assumption is critical because it enables the use of the Fréchet distance to quantify the similarity between distributions (Jayasumana et al., 2024). Below, we detail the mathematical framework, interpret the score, and analyze the reliability of FID as a measure of image quality.

To determine the FID score, first select a pre-trained Inception Network (e.g. Inception-v3). Then both the real and the generated images are run separately through the model. Afterwards the feature representations of the images are extracted – the features are usually taken from the last pooling layer (before

classification) of the model, although earlier layers can also be used for extraction (Lucic et al., 2018). The mean vectors and the covariance matrices are formed from the feature vectors for both image sets (real and generated) in order to calculate the FID score. This is then calculated as (Heusel et al., 2017):

$$FID(\mu_r, \Sigma_r, \mu_g, \Sigma_g) = \|\mu_r - \mu_g\|_2^2 + Tr\big(\Sigma_r + \Sigma_g - 2(\Sigma_r\Sigma_g)^{\frac{1}{2}}\big)$$

While μ_r corresponds to the mean values of the real set and μ_g to the mean values of the generated set, Σ_r represents the covariance matrix of the features of the real data and Σ_g the covariance matrix of the features of the generated data (Kynkäänniemi et al., 2023).
If we now look at the first part $\|\mu_r - \mu_g\|_2^2 = \Sigma_{i=1}^{n}(\mu_{r,i} - \mu_{g,i})^2$, it follows that the Euclidean norm measures the distance between the centers of the two distributions in n-dimensional space.
The second part is divided into $Tr(\Sigma_r)$ and $Tr(\Sigma_g)$, which is the sum of the variances (that is, the total variance) of the real and generated data and $-2(\Sigma_r\Sigma_g)^{\frac{1}{2}}$, which is a verification term, because for a symmetrical, positive semidefinite matrix A is $A^{\frac{1}{2}}$ the unique symmetric, positive semidefinite matrix which has the property

$$A^{\frac{1}{2}}A^{\frac{1}{2}} = A\ .$$

So, this results in $Tr(\Sigma_r + \Sigma_g - 2(\Sigma_r\Sigma_g)^{\frac{1}{2}})$ and if Σ_r and Σ_g are very similar, the term $(\Sigma_r\Sigma_g)^{\frac{1}{2}}$ has almost the same value as Σ_r (resp. Σ_g), so that

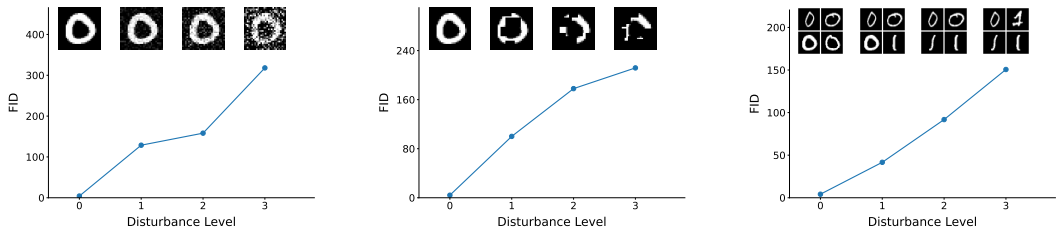
$$Tr(\Sigma_r + \Sigma_g - 2(\Sigma_r\Sigma_g)^{\frac{1}{2}}) \approx 0 \text{ applies.}$$

Overall, the distance of the mean vectors thus determines whether the distribution of generated data has the same central properties on average as the distribution of the real data while the second term compares the scatter and internal correlations of the distributions. It records whether the generated data corresponds to the real data in terms of diversity and structure. FID considers both the position (mean values) and the shape (covariances) of the distributions.

In this context, the evaluation of the FID score is as stated in Heusel et al. (2017):

- A low FID value indicates that the statistical distributions (mean values and covariance matrices) of the real and generated images are very similar. This means that the generated images are very close to the real images.
- A FID value of 0 would be ideal and means that the distributions match exactly, i.e. the real and generated images are statistically identical.
- A high FID value indicates greater differences between the distributions, which indicates that the generated images differ significantly from the real ones.

In Fig. 2.1 it can be seen how FID performs on an original set of 0s from MNIST in similarity comparison with different disturbances in the second dataset. Here, it can be seen that the stronger the disturbance (the more the second dataset deviates from the original set), the higher the FID.



(a) Gaussian Noise **(b)** Overlaying Rectangles **(c)** Set of 0s infiltrated by 1s

Figure 2.1: Changes in FID Values Depending on the Intensity of Disturbance

The widespread use of FID as a benchmark for generative models is due to its superiority over previous metrics, as noted in Heusel et al. (2017). In their paper introducing FID, the authors pointed out critical limitations of the Inception Score (IS), which evaluates generated images in isolation by measuring the entropy of the class-label distributions predicted by Inception-v3 (Salimans et al., 2016). While IS rewards diversity and discriminability, it does not directly compare the generated samples with the real data (Heusel et al., 2017).
Systematic experiments have shown that FID correlates strongly with human perception of image quality: In user studies, lower FID scores corresponded with higher perceptual ratings of image realism and variety. FID reflects human perception of image quality and diversity that is 'more consistent [...] than the Inception Score' (Heusel et al., 2017, p. 6).
In contrast to IS, FID considers the feature space of an earlier layer of the Inception Network instead of the final classification output. This makes it more robust to subtle distortions such as noise, blurring, or artificially generated patterns (Heusel et al., 2017; Lucic et al., 2018).
This results in a more comprehensive evaluation with FID than with IS.

2.3 Critical Evaluation of FID

The past shows that distances are more than just absolute values that measure distances:
From antiquity until the 16th century, the geocentric world view was assumed. The earth was at the center of the universe, surrounded by concentric, crystalline spheres to which the moon, sun, planets and fixed stars were attached. Each sphere would have a defined distance from the earth, calculated using geometric models with epicycles (auxiliary circles). In order to explain planetary movements, Ptolemy had to introduce a large number of epicycles (Hanslmeier, 2002) – a complex construct that provided precise predictions but was physically absurd. The distances between spheres were pure fiction (Ptolemy, 1999, pp. 419–421).
Although there was no empirical basis for this assumption, the geocentric model was closely interwoven with religion and philosophy. Questioning the Earth as the center of creation was considered heresy (Gebhardt & Kiesel, 2013).
The paradigm shift to heliocentrism – introduced by Copernicus – was not only scientifically but also culturally shattering. It dethroned the Earth from the center of the cosmos and triggered an identity crisis in philosophy and theology. 'The world we have lost [...] has been dismantled and replaced by something quite different in the transformation we often roughly call disenchantment' (Taylor, 2007, p. 61)
The distances of the spheres were also replaced by Kepler's laws and Newton's gravitation - a completely new understanding of space and cosmic order.
These past episodes demonstrate: Distances are never neutral, they always reflect the assumptions of those who measure them.
We must be aware of this conclusion, even if we are only measuring the similarities of the feature distributions of images. FID is not absolute either, which is why we are now taking a closer look at its limitations and challenges.

2.3.1 Limitations and Challenges of FID

Dependence on Embedding Network

FID is based on feature embeddings of an Inception-v3 model pre-trained on ImageNet. These features are closely related to the ImageNet classifications, as the pre-logits are only one affine transformation away from the logits .
As a result, FID prioritizes image regions that are relevant to ImageNet classes, even if they are irrelevant to the target dataset. For example, in Flickr-Faces-HQ (FFHQ), FID focuses on accessories such as 'ties' or 'seat belts', not on facial details (Kynkäänniemi et al., 2023).
This is also evidenced by the fact that FID sensitivity is more strongly influenced by adding noise in these ImageNet-relevant regions (Kynkäänniemi et al., 2023).

Another problem that arises from the embedding network is: Models that use ImageNet pre-trained components (e.g. discriminator) involuntarily reproduce ImageNet features, which artificially improves FID. This means that for GANs, for example Projected FastGAN ('uses an ImageNet pre-trained EfficientNet' (Kynkäänniemi et al., 2023, p. 9)) provides lower quality results despite comparable FID to StyleGAN2.
This is also shown by Sajjadi et al. (2018) who demonstrate that 'extracted high-level features may also be very sensitive to class-dependent image features and not necessarily for general image quality' (Sajjadi et al., 2018, p. 14).
This implicates that models that produce 'ImageNet-like' images (e.g. with correct class features) are overestimated, even if they are stylistically unrealistic.

This dependency on the embedding network can result in strong distortions in the FID evaluation.

Violation of Gaussian Assumption

FID is based on the assumption that Inception embeddings (penultimate layer of Inception-v3) are multivariate normally distributed. However, this assumption cannot be made without restriction, because it is not always given that the embeddings are multivariate normally distributed.
This can induce in misleading results, as FID approximates the actual distribution of Inception embeddings by a normal distribution, it underestimates or overestimates the distance between real and generated data.
In a synthetic experiment by Jayasumana et al. (2024), FID falsely remained at zero for non-normal distributions, even though the datasets were highly divergent. This leads to models with fundamentally different quality being classified as equivalent.
Limits of FID are as well presented in the distance evaluation on dog classes (D) versus non-dog classes ($\sim D$) by Sajjadi et al. (2018). It turns out that 'FID may be comparably insufficient in recognizing dissimilarities between distributions that have clear distinctions (whether across individual classes or in a one vs. all scenario)'(Sajjadi et al., 2018, p. 14).

Sample Size Effect

Extremely large samples are required for reliable estimates of FID. The reason for this is covariance estimation: When using Inception calculating FID requires estimating a 2048×2048 covariance matrix with 4 million entries. This requires a large number of images, as otherwise an estimate from limited data leads to errors in interpretability (Chong & Forsyth, 2020; Jayasumana et al., 2024).

2.3.2 Managing Scope and Interpretability

When using FID, it is important to be aware of the mentioned limitations because they cannot always be eliminated. In instances where limitations cannot be excluded, FID does not provide intuitively interpretable values. Instead, it provides an abstract scale value that is difficult to translate into concrete differences in quality.

Consequently, it is necessary to create a setting that is as suitable as possible. For example when using small datasets one has to make sure that they are at least of equal sample size to be able to carry out relative comparisons fo FID.
In addition, other complementary metrics, such as Maximum Mean Discrepancy (MMD) or Sliced Wasserstein Distance, should be used to understand and validate the results of FID (Bischoff et al., 2024).

Chapter 3

Custom Models

Now the essentials are in place to understand how FID works, what requirements it needs, and also where the limitations of FID lie.
It is at this point that a brief return to the art gallery is made. Two paintings of completely different styles hang on another wall, illuminated by a flickering neon sign. The light from the sign, which has been calibrated to mimic sunlight, casts unnatural colours onto the canvases. The authenticity of these images is subject to the judgement of a critic, yet this judgement is contingent upon the distorted colours caused by the glare. Consequently, both images appear very much alike to him.
This predicament confronts the FID: its view of the world is filtered through Inception-v3, a lens polished by the 1.2 million examples of *reality* in ImageNet.
But in this context it is about comparing the distributions of two sets of images rather than two images.
Therefore the reliability of FID measurements depends on the embedding network. As Inception is trained on ImageNet, the reliability of the Fréchet Inception Distance is questionable for satellite images with more than three RGB channels, for example.
In order to verify whether this fragility can be counteracted, custom *lenses* (embedding networks) are trained and then used to compute the custom Fréchet instance. The determination of the Fréchet instance through both the default and custom lenses is expected to facilitate the investigation of the efficacy of custom lenses in specific niche domains not included in the ImageNet dataset.

In this chapter, the architecture and performance of three custom models will be examined. The objective is to investigate whether there is a discrepancy between the models with regard to the custom Fréchet Distance.
The dataset under consideration is characterized by the provision of reliable FID values, coupled with a straightforward processing nature. The fundamental objective is to demonstrate a proof of concept that can then be applied to niche datasets (see Chapter 4).

3.1 Data

I used a CIFAR-6 dataset to train all three of the custom models. This means that I have excluded 4 classes from the original CIFAR-10 training dataset at the beginning.
In this case, the classes 'Cat', 'Automobile', 'Frog', 'Horse', 'Ship' and 'Deer' were used for training, while the model never saw images of airplanes, birds, dogs and trucks during training.
The purpose of this division is that the performance regarding to the Fréchet Distance in Chapter 4 can then be evaluated better and more comprehensively. In addition, 1000 images of each of the 6 classes used were not taken into account for training, as they are needed later for the evaluation (since the CIFAR-10 test set only consists of 1000 images per class).
This means that 24,000 images and an additional 4000 augmented images were used for training the models. In Figure 3.1, this corresponds to the upper left (salmon-colored) area.

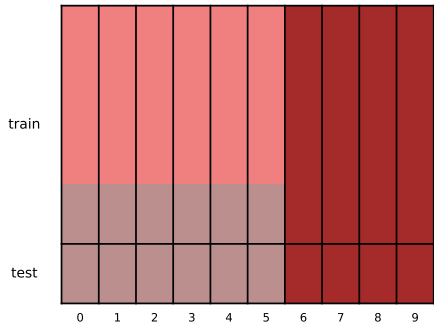


Figure 3.1: Partitioning of the dataset

3.2 Architecture

The three custom models are a simple model (simple Custom Model), an overfitting model (overfitting Custom Model), and finally a model that performed best on the training and test data (best Custom Model). It is important to note that the naming of the models is based on how their basic architecture fits into the small CIFAR-6 dataset. Even the best Custom Model is not objectively the best there is, but in this context it was the best performing of the three models.
Furthermore, all three models have the same latent dimension (128), which is smaller than Inception-v3's (2048), in order to ensure comparability between the custom models.

3.2.1 Simple Custom Model

The architecture of the simple Custom Model turns out to be a very simple, fully-connected network structure (see Table 3.1). The input image ($32 \times 32 \times 3$) is first flattened and then passed through two linear layers ($3072 \rightarrow 128$ and $128 \rightarrow 64$) with ReLU activations. Subsequently, a latent representation (here via a linear layer from 64 to 128) is extracted and used as input for the final classification ($128 \rightarrow 6$). The name 'simple Custom Model' results from the basic, straightforward structure without convolutive feature extraction – ideal as a baseline model to see how well a simple approach works with CIFAR-6.

Type	Patch Size/Stride or Remarks	Input Size	#Parameters
Linear	$32 \times 32 \times 3 \rightarrow 128$	$32 \times 32 \times 3$	~393.000
Linear	$128 \rightarrow 64$	128	~8.000
Linear	$64 \rightarrow 128$	64	~8.000
Linear	$128 \rightarrow 6$	128	~770

Table 3.1: Simple Custom Model's Architecture

3.2.2 Overfitting Custom Model

The overfitting Custom Model is a deeper, convolutional network that borrows heavily from ResNet (see Table 3.2). Starting with a 3×3 conv layer ($3 \rightarrow 32$) and subsequent batch normalization, the image is passed through four successive layers, each consisting of two basic blocks, which are explained below. Each of these layers gradually increases the number of channels (from 32 to 256) and reduces the spatial resolution by striding. Global average pooling is followed by a latent layer ($256 \rightarrow 128$) and a classic linear layer for final classification ($128 \rightarrow 6$).

Due to its high capacity, the model may tend to overfit on small datasets (such as CIFAR-6). For this reason, it is called overfitting Custom Model and is intended to investigate the effect of overfitting.

BasicBlock
The BasicBlock used in the overfitting Custom Model aims to facilitate deep network learning by employing a residual learning approach that supports gradient descent. First, a Conv2d layer with a 3×3 kernel is applied, where the stride is chosen as the passed value (1 or 2 depending on the position in the network) and a padding of 1 is used to maintain or reduce the spatial resolution in a controlled way. Since no bias is used – due to the subsequent batch normalization – this layer transforms the input channels into the desired number of output channels. Afterwards, a BatchNorm2d layer normalizes the activations of the first convolution, which helps to stabilize the training, and a ReLU activation introduces the necessary

nonlinearity to allow the network to learn complex patterns. This is followed by a second convolution in which another Conv2d layer with a 3×3 kernel, a stride of 1 and padding of 1 ared used to further process the feature maps without changing the spatial dimension – again, no bias is used. A further BatchNorm2d layer then standardizes the output of this convolution. To implement the residual connection, the system first checks whether the input has the same form as the output; if this is the case (i.e. if Stride equals 1 and the number of channels matches), the input is used directly as a shortcut. If, on the other hand, an adjustment is required because either the stride is not equal to 1 or the number of channels does not match, the input is projected using a 1×1 convolution with the corresponding stride and then batch-normalized so that the dimensions can be adjusted to those of the main segment. Finally, the output of the second batchNorm and the shortcut are added, and a new ReLU activation is performed to generate the final output of the block.

Type	**Patch Size/Stride or Remarks**	**Input Size**	**#Parameters**
Conv2d	$3 \times 3/1$	$32 \times 32 \times 3$	~864
BatchNorm2d	-	$32 \times 32 \times 32$	~64
BasicBlock (Layer 1, Block 1)	$3 \times 3/1$	$32 \times 32 \times 32$	~19.000
BasicBlock (Layer 1, Block 2)	$3 \times 3/1$	$32 \times 32 \times 32$	~19,000
BasicBlock (Layer 2, Block 1)	$3 \times 3/2$	$32 \times 32 \times 32$	~58.000
BasicBlock (Layer 2, Block 2)	$3 \times 3/1$	$16 \times 16 \times 64$	~74.000
BasicBlock (Layer 3, Block 1)	$3 \times 3/2$	$16 \times 16 \times 64$	~230.000
BasicBlock (Layer 3, Block 2)	$3 \times 3/1$	$8 \times 8 \times 128$	~295.000
BasicBlock (Layer 4, Block 1)	$3 \times 3/2$	$8 \times 8 \times 128$	~919.000
BasicBlock (Layer 4, Block 2)	$3 \times 3/1$	$4 \times 4 \times 256$	~1.181.000
AvgPool2d	$4 \times 4/1$	$4 \times 4 \times 256$	0
Linear (Latent)	$256 \rightarrow 128$	256	~33.000
Linear (Classifier)	$128 \rightarrow 6$	128	~770

Table 3.2: Overfitting Custom Model's Architecture

3.2.3 Best Custom Model

In its architecture, the best Custom Model is a compact ResNet-like model that is based on ~BasicBlock and additionally uses dropout for regularization (see Table 3.3). The model starts with a 3×3 conv layer ($3 \rightarrow 16$) and BatchNorm, followed by three residual blocks (Layer1, Layer2, Layer3). In Layer2 and Layer3, the spatial resolution is reduced by downsampling and the number of channels is doubled at the same time ($16 \rightarrow 32, 32 \rightarrow 64$). After adaptive average pooling, the features are flattened and passed on to a final classifier ($128 \rightarrow 6$) via a latent layer ($64 \rightarrow 128$). The integration of Dropout into ~BasicBlock also ensures improved generalization compared to the other two models. This is why it is called the best Custom Model.

Type	Patch Size/Stride or Remarks	Input Size	#Parameters
Conv2d	$3 \times 3/1$	$32 \times 32 \times 3$	~430
BatchNorm2d	-	$32 \times 32 \times 16$	~30
~BasicBlock (Layer 1, Block 1)	$3 \times 3/1$	$32 \times 32 \times 16$	~4.600
~BasicBlock (Layer 1, Block 2)	$3 \times 3/1$	$32 \times 32 \times 16$	~4.600
~BasicBlock (Layer 2, Block 1)	$3 \times 3/2$	$32 \times 32 \times 16$	~14.500
~BasicBlock (Layer 2, Block 2)	$3 \times 3/1$	$16 \times 16 \times 32$	~18.500
~BasicBlock (Layer 3, Block 1)	$3 \times 3/2$	$16 \times 16 \times 32$	~58.000
~BasicBlock (Layer 3, Block 2)	$3 \times 3/1$	$8 \times 8 \times 64$	~74.000
AdaptiveAvgPool2d	$1 \times 1/1$	$8 \times 8 \times 64$	0
Linear (Latent)	$64 \rightarrow 128$	64	~8.000
Linear (Classifier)	$128 \rightarrow 6$	128	~770

Table 3.3: Best Custom Model's Architecture

~BasicBlock
Based on the already existing description of BasicBlock, which uses two consecutive 3×3 convolutions with subsequent batch normalization and ReLU activation as well as an internally implemented shortcut connection, the differences of ~BasicBlock can be specified as follows: In ~BasicBlock, a dropout with a rate of 0.3 is applied immediately after the first ReLU activation, a step missing in BasicBlock. In addition, the shortcut in ~BasicBlock is not generated internally by a conditional check, but is passed externally as a parameter. This external transfer makes it possible to explicitly define the downsample path if, for example, the number of channels or spatial dimensions needs to be adjusted. Apart from these differences, ~BasicBlock otherwise follows the same structure as BasicBlock, whereby the second convolution with batch normalization and the final ReLU activation are implemented identically. Thus, the use of dropout and the flexible, external handling of the shortcut are the essential, but precisely separated modifications that distinguish ~BasicBlock from BasicBlock.

Recap The architectures show that the simple Custom Model has a very basic structure, while the overfitting Custom Model is large and deep with more than

2,800,000 parameters.
The best Custom Model, on the other hand, has just around 180,000 parameters and is therefore a more refined, more efficient model, which I expect to have better generalization properties on CIFAR-6 than the other two.
Whether the models behave as expected (based on the architecture) will be investigated in the next sections by examining the training and testing on CIFAR-6.

3.3 Training

For the training of the models, each the simple and the overfitting Custom Model were trained for 20 epochs, while the best Custom Model was trained for 30 epochs. With a learning rate of 0.002, all models were trained using ADAM (Kingma & Ba, 2017) as optimizer.

3.4 Performance

To measure the performance of the models, I have looked at the Top-1 accuracy, the Top-2 accuracy and the loss, which can be seen in Figure 1.
The overfitting Custom Model performs as expected, achieving very high values > 0.95 for the two train accuracies, while the best Custom Model and the simple Custom Model achieve very similar values, specifically 0.75 for the simple model for the Top-1 accuracy and 0.87 for the Top-2 accuracy, while the best Custom Model performs 0.01 worse for each. The loss on the training set is lowest for the overfitting model at 0.11, and the other two models are equal at 0.66.
The extremely good values for the overfitting Model in training indicate exactly that it overfits the training data.
However, the important part lies in the evaluation of the three models on the test data. Here it is layed out how good the models perform in generalization. The simple model performs worst in all categories, in particular the high loss of 1.41 is more than twice as high as the training loss. The simple Model also performs worse on the test data than the training data in terms of accuracies.
Although the overfitting performs better than the simple Custom Model on the test data, the loss is seven times higher, and the accuracies are also lower. The best Custom Model outperforms all models. It generalizes better, as all values on the test data are superior to those on the training data. The loss on the test data is 0.36 and the Top-1 accuracy is 0.88.
As the characteristics indicated by the model names suggest, the results obtained accurately reflect the behavior: The simple Custom Model is the worst performing model according to the values, while the overfitting Custom Model is far too adapted to the training data, but the best Custom Model shows the highest ability to abstract and also the best absolute values for the test set.

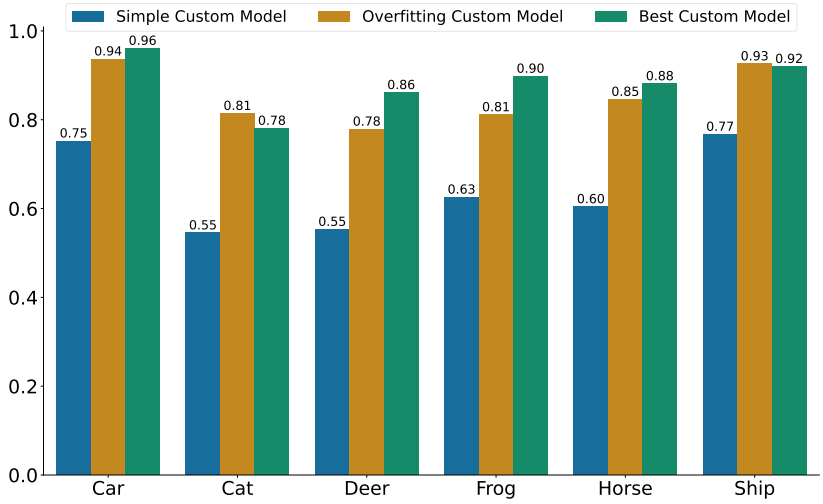
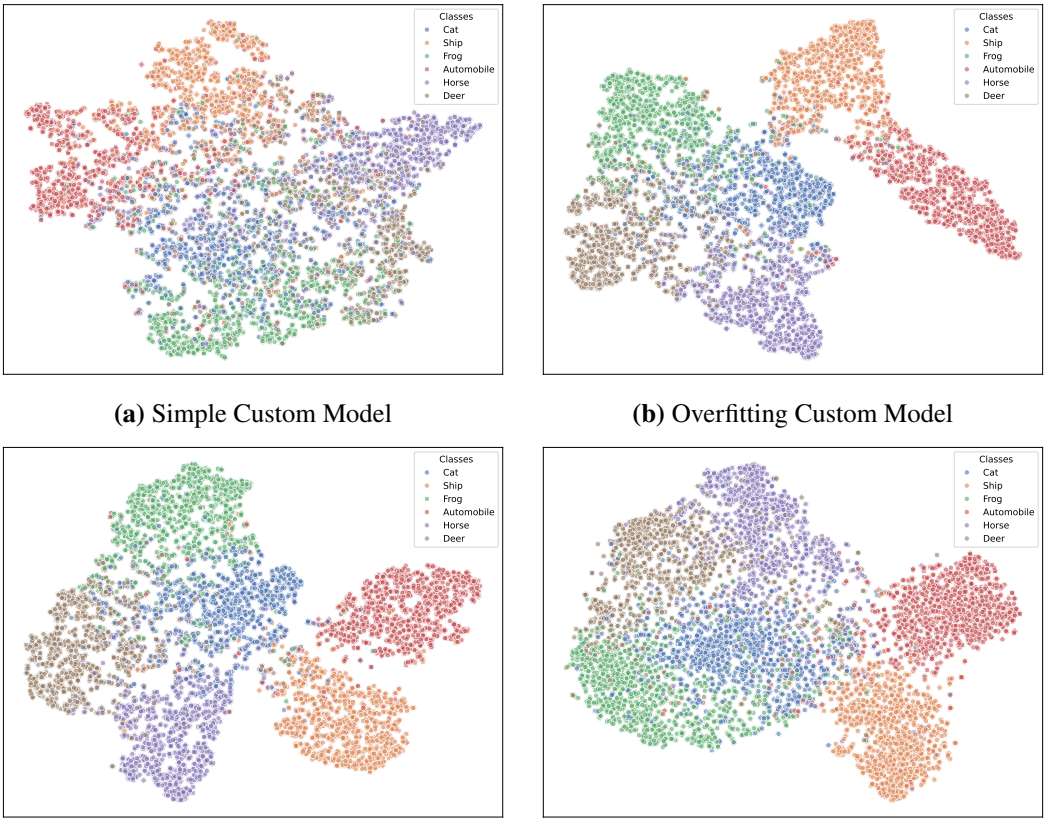


Figure 3.2: Class Specific Top-1 Accuracy

In addition to the Top-1 and Top-2 accuracies, the class-specific Top-1 accuracies were also determined on the test data to see whether and to what extent the three models behave differently in terms of performance on the individual classes as well as in the distinctions between the classes.
Figure 3.2 shows that the overfitting Custom Model can classify the classes Cat and Ship most accurately compared to the other two models, but the best Custom Model performs better for all other classes. The simple Custom Model consistently performs the worst here with an accuracy of below 0.80 for every class. But it is noteworthy that the animal classes ('Cat', 'Deer', 'Frog', and 'Horse') only achieve an accuracy between 0.55 and 0.63.
It seems remarkable that the non-animal images of the classes 'Ship' and 'Automobile' are classified most precisely by all of the Custom Models, since the overfitting model and the best Custom Model achieve accuracies greater than 0.92. In total images of cars are classified most reliably with 0.96 and 0.94 respectively.

It can be recognized in Figure 3.3, which shows the t-SNE representation (van der Maaten & Hinton, 2008) of the test data points, i.e. the interrelationships between the classes are displayed by the distances between the points.
This representation makes it possible to determine which classes of which model are most likely to be mistaken one for another, or which classes are similar for the respective model. It can be seen that the non-animal classes 'Ship' and 'Automobile' are most explicitly separated for the simple Custom Model (see Fig. 3.3a), the overfitting Custom Model (see Fig.3.3b) and the best Custom Model (see Fig. 3.3c)). However, the separation is more distinct for the overfitting and best Custom Model than for the simple Custom Model. Therefore, these classes are classified more clearly.
It can be observed that the danger of confusion within the class classification is high

for the simple Custom Model. This is shown by the almost nonexistent separation of the different classes, especially for the classes of animals.
Although the overfitting and the best Custom Model do indeed achieve a stronger separation of the animal classes, the best Custom Model is still most capable of distinguishing the 'Horse' class from the other animal classes (better than the overfitting model).
Furthermore, the latent spaces demonstrate that the 'Cat' class occupies a central position. Consequently, this class will possess the maximum average similarity with all other classes. It exhibits the widest potential for a false negative. Conversely, the 'Frog' and 'Horse' classes invariably have a very large distance and hence separation for all models, so that they are very unlikely to be swapped in labeling.
In addition, the latent space of Inception-v3 for the CIFAR-6 dataset is also plotted. It can be seen that Inception-v3 can separate the classes in principle (and better than the simple Custom Model), and again a clear separation of animal and non-animal classes is evident. Compared to overfitting and the best Custom Model, the scatter is greater and the separation less clear.



(a) Simple Custom Model

(b) Overfitting Custom Model

(c) Best Custom Model

(d) Inception-v3

Figure 3.3: t-SNE Visualization of Custom Models and Inception-v3

The way in which the models deal with the classification of five random test images is visualized in Table 3.4. You can notice that the simple Custom Model is by far the least reliable with only one correct classification. It is also visible that it has quite a strong misclassification even for the middle image (automobile), and the second highest accuracy is also wrong. For the other misclassifications, it is clear that the assignments are not entirely unambiguous, as the class-specific Top-1 accuracy values are lower than 0.5 as demonstrated in 3.2.
It is obvious that both the overfitting Custom Model and the optimal Custom Model accurately classify all test images. However, it is also apparent that the overfitting Custom Model is highly data-adapted, as evidenced by the Top-1 accuracies, which are all greater than 0.99.

		Simple Custom Model	**Overfitting Custom Model**	**Best Custom Model**
	Label: Top-1 ACC	Deer: 0.4132	Frog: 0.9989	Frog: 0.7790
	Label: Top-2 ACC	Frog: 0.3333	Cat: 0.0010	Cat: 0.1620
	Label: Top-1 ACC	Frog: 0.4748	Deer: 0.9995	Deer: 0.9996
	Label: Top-2 ACC	Deer: 0.4303	Frog: 0.0005	Horse: 0.0003
	Label: Top-1 ACC	Horse: 0.9197	Car: 1.0000	Car: 0.8613
	Label: Top-2 ACC	Deer: 0.0347	Horse: 0.0000	Horse: 0.1006
	Label: Top-1 ACC	Horse: 1.0000	Horse: 1.0000	Horse: 1.0000
	Label: Top-2 ACC	Deer: 0.0000	Deer: 0.0000	Deer: 0.0000
	Label: Top-1 ACC	Car: 0.4993	Ship: 0.9978	Ship: 0.9999
	Label: Top-2 ACC	Ship: 0.4986	Car: 0.0022	Car: 0.0001

Table 3.4: Example Classification of Custom Models

Chapter 4

Experiments with Custom Models

In the previous chapter, the structure and performance of the custom models on CIFAR-6 were examined. As stated earlier, the custom models will now be tested in terms of Fréchet Distance in order to be able to make statements about how accurately the Fréchet Distance can be determined, using the custom model's embedding spaces on CIFAR-10. For this purpose, the Fréchet distance calculated by using the embedding spaces of the custom models is presented first. To assess the utility of training custom models for domains where Inception fails (does not provide a reliable FID), such as satellite images, this investigation is performed as a proof of concept. This means that instead of using generated images from niche domains, CIFAR-10 is used as the data. Since the custom models were only trained on 6 of the 10 classes, the other 4 classes are unknown data for the models.
In total, there are 3 different subsets of CIFAR-10 (see Figure 3.1): First, the data on which the custom models was trained (6 classes), which will be referred to as 'Home Ground', since the models are familiar with both the images and the classes (see the salmon-colored area at the top left of Figure 3.1). The second subset contains the classes that the custom models know from training, but do not recognize the specific images from training (see the beige-brown area at the bottom left of Figure 3.1). This subset is called 'Borderland' because the models are not as familiar with this area as they are with Home Ground, but they are not completely ignorant either. What remains are the four classes and their associated images that the models have not seen at all (see the rusty red area to the right of Figure 3.1). This subset is therefore referred to as 'Uncharted Territory'.

It is explored how the Fréchet Distance behaves based on the embedding spaces of the custom models in Home Ground, Borderland and Uncharted Territory, and whether there are differences in the way the custom Fréchet Distance acts in different areas and for different classes.

4.1 The Custom Fréchet Distance

The FID is called Fréchet Inception Distance because it results from the Fréchet Distance between the distributions of the features of the Inception network.
Since I want to analyze how custom models perform for specific domains in this chapter, the Fréchet Distance must also be implemented for the embedding spaces of the custom models.
By using the custom models as feature extractors, this is achieved. The features of the two compared sets are extracted, afterwards these feature vectors are used to determine the mean vectors and the covariance matrices of the these sets. Both are then used to calculate the Fréchet Distance. The main difference in the calculation itself is that the matrix size differs greatly from Inception. The custom models use 128×128 sized covariance matrices, while the Inception model (as described in 2.3.1) uses 2048×2048 sized matrices. In practical terms, this means that the custom matrices contain 16.384 entries and the inception matrices 4.194.304 entries.
As the feature extractor for calculating the Fréchet Distance differs in the following due to the use of different models, the following notations are introduced:
$FC_{sm}D$ for the Fréchet Distance using the embedding space of the simple Custom Model, $FC_{om}D$ for the Fréchet Distance using the embedding space of the overfitting Custom Model and $FC_{bm}D$ for the Fréchet Distance using the embedding space of the best Custom Model. FID still refers to the Fréchet Distance determined using Inception-v3.

4.2 Requirements and Methods

The CIFAR-10 dataset is used, of which the custom models know only 6 classes from training, as described in 3.1. Since the samples of the entire dataset are limited (50.000 training images and 10.000 test images) and the test set contains only 1000 images per class, another 1000 images per class from the train set were not used for training but as additional test data. This ensures that each 1000 different images can be compared when both the two sets contain images of the same class.
In order to analyze the performance of custom models in the following sections, with a particular focus on the extent to which they can be useful for domains not covered by ImageNet, the following aspects must be considered:

1. How well do custom models perform on Home Ground and Borderland? How do they perform compared to Inception-v3? It is possible to conclude from this whether custom models can be considered successful for specific data they were trained on.

2. How well do custom models perform on Uncharted Territory? That is, on the 4 classes that they do not know from training but is similar to training data. In

addition, custom models are also tested on data which differs significantly from CIFAR-10. This could be a measure of how Inception performs on data that is not included in ImageNet.

3. Are there any differences for the custom Fréchet Distance between the embedding spaces of the custom models?

It must be recognized that the absolute values of the FID and FCDs are not significant for the performance evaluation, given that only 1000 samples each are used for calculation. This is due to the violation of the sample size condition (see 2.3.1). Furthermore, it is not possible to unconditionally assume that the distributions of the features are Gaussian (see 2.3.1).
Therefore, it is necessary to examine the relationship between the FID scores in order to draw conclusions. This implies that the utilization of human rationality becomes essential in order to be able to judge whether, for example, a deer is more like a horse or a car.

4.3 Performance on Home Ground, Borderland and Uncharted Territory

First, the basic performance between Home Ground, Borderland, and Uncharted Territory is analyzed, with more than one class for each dataset.
Figure 4.1 shows, as expected, that the FCDs of the comparison between different Home Ground images (same classes) is very accurate for the custom models as it is very low. $FC_{bm}D$ is the smallest at 0.8. But the Fréchet Distance is the largest of all the custom models when the embedding space of the simple Custom Model is used, it is not as close to zero.
It is immediately evident that the FID is 43.4, despite the expectation of a score close to zero. This outlier when comparing equal classes is consistently present (see 4.2 and 4). This phenomenon can rapidly result in misinterpretations if not evaluated within the appropriate context. It is therefore necessary to keep this pathological behavior in mind, its causes will be discussed later in Chapter 5.

Figure 4.1: Performance of $FCDs$ and FID on Home Ground and Borderland
Experiment A: Comparison of Different Images from the Same (all) Classes in Home Ground; **Experiment B**: Comparison of Classes 'Car', 'Cat', 'Frog' with 'Deer', 'Horse', 'Ship' on Borderland; **Experiment C**: Comparison of Classes 'Car', 'Cat', and 'Frog' with 'Deer', 'Horse', and 'Ship' on Home Ground

To test whether the custom models have a correspondingly higher FID with different data, three different classes each from Home Ground (Experiment B in 4.1) and Borderland (Experiment C in 4.1) are compared.
As demonstrated in 4.1, it is apparent that all the three FCDs achieve significantly high scores, demonstrating that the basic functionality of the custom Fréchet Distances is working, as similar comparisons of data achieve near-zero scores, while comparisons of different data achieve higher scores. However, it can also be seen that all custom models are slightly too adapted to the training data since it is expected that, as with the FID, the custom Fréchet Distances when comparing different Home Ground classes and when comparing different Borderland classes will be nearly identical (since the datasets being compared contain the same classes). But the FCDs demonstrate a higher degree of similarity within image comparison in Home Ground than in Borderland. But this difference is very small.

A comparison of the Borderland and Home Ground classes with the classes from Uncharted Territory reveals that the Fréchet Distances for both comparisons are almost identical (see 4.2). Accordingly, the FCDs evaluate the Borderland and Home Ground images similarly to the Uncharted Territory images. Consequently, the FCDs can process images of the same classes that were not present in the training data in the same way as the actual training data in terms of comparisons with unknown data.

Now exclusively basic comparisons in Uncharted Territory will be considered. Comparatively, the performance of Inception-v3 is also examined. Since Inception-v3 was trained on ImageNet, which contains images of these 4 classes, it is expected that it should perform accurately.

Figure 4.2: Performance of $FCDs$ and FID when Comparing Home Ground and Borderland with Uncharted Territory
Experiment D: Comparison of Different Images from the Same (all) Classes in Borderland; **Experiment E**: Comparison of all the Classes in Home Ground with all the Classes in Uncharted Territory; **Experiment F**: Comparison of all the Classes in Borderland with all the Classes in Uncharted Territory;

As illustrated in Figure 3, all of the FCDs can recognize that the feature distributions are the same when comparing the identical classes. It is further consistent that the classes 'Airlplane+Bird' and 'Dog+Truck' are more similar (Experiment H) than 'Airplane+Truck' and 'Bird+Dog' (Experiment I). This could be explained by the fact that first a set of an animal class + a non-animal class is compared with a set of the same type. And secondly two non-animal classes are compared with two animal classes. This finding suggests that custom models can transfer this gained knowledge of animal and non-animal classes to unknown data, as indicated by the custom Fréchet Distances.

It can be observed that although the absolute values of the FID of the custom models are very different (which is due to their architectures), the relationships to each other are always very similar.

4.4 Class Specific Performance

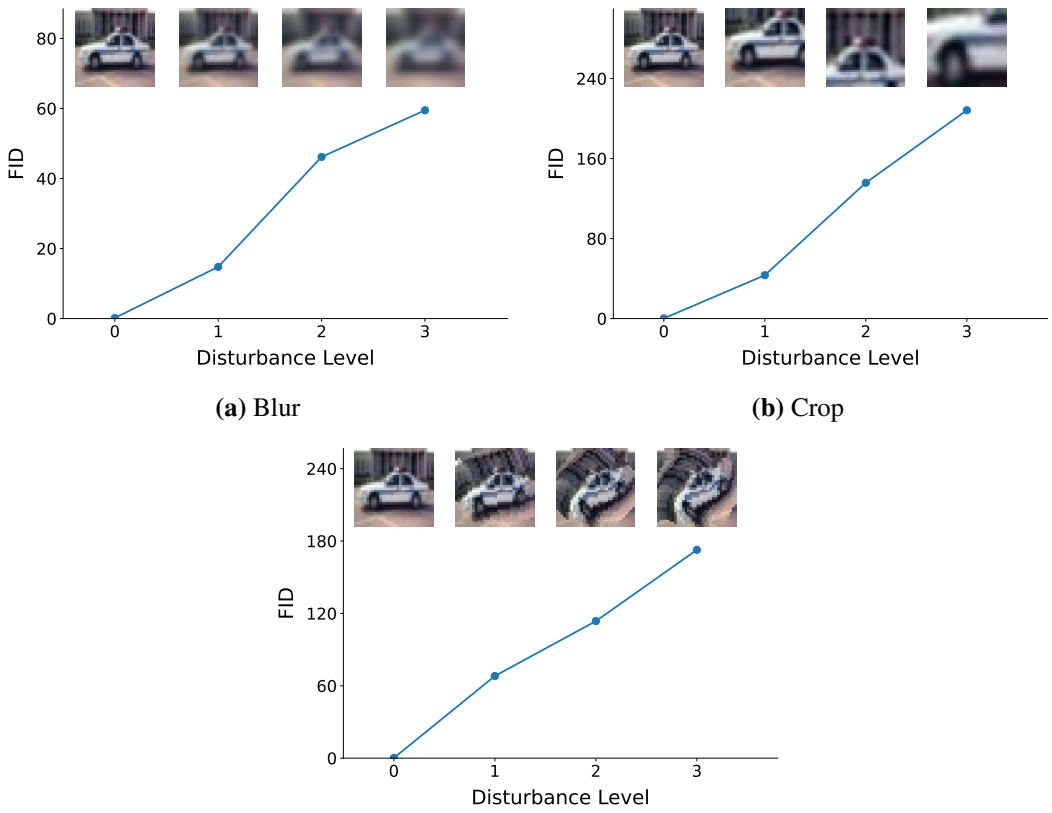
The next step is to focus on the individual classes. Initially, all classes of CIFAR-10 are to be considered. For this purpose, each FCD and FID between all classes has been determined. However, it is important to note that the four classes 'Airplane', 'Bird', 'Dog' and 'Truck' from Uncharted Territory are not known to the custom models.
The observation that all classes exhibit a high degree of similarity to themselves is indicated by the low scores of all FCDs (see F.1). Consequently, the analysis is focused on the comparisons between different classes.
A striking observation is that all FCDs are unable to accommodate the 'Bird' class.

In practice, this signifies that the 'Bird' class always achieves the highest or second-highest similarity, i.e. the smallest or second-smallest FCD, in particular for the non-animal classes. But that is not correct, because 'Deer' and 'Frog' do not have the highest similarity to 'Bird', but to other non-flying animals (see 2). It also becomes evident that the 'Bird' class is the least easy and most ambiguous to classify, especially for the custom models. This may provide a reason for the unreliable FCDs. Inception-v3 also encounters challenges with the 'Bird' class, though not to the same extent: 'Bird' attains the second smallest FID for 'Deer', 'Frog' and 'Cat'(see F.1).
Examining only the Home Ground classes, it can be seen that the Fréchet Distances, which use the embedding spaces of the custom models, evaluate all class similarities well. This means that the FCDs clearly identify the difference between animal and non-animal classes (high FCDs) as mentioned earlier. However, the differences between animal classes and between non-animal classes and different animal classes are less relevant, as it is quite difficult for humans to tell whether cars and horses or cars and deers are more similar.
These findings can also be perceived in the classes of Uncharted Territory, which means that the FCDs can transfer these fundamental differences to images that the custom models (whose embedding space is used to determine the FCDs) do not even know. It should be noted, that images from Uncharted Territory still have some similarities to those from Home Ground (same domain), and therefore in a later section it will be examined how the custom Fréchet Distances perform on data, which is very different from CIFAR-10.
It is also observable that the FCDs and FID are always very high when comparing the classes 'Horse' and 'Frog' and that the class 'Bird' cannot be reliably judged at all.

4.5 Performance on Augmented Data

I have also tested how reliably the FCDs can be determined with augmented images. For this purpose, different transformations than the ones used for the training of the custom models were applied.
The transformations are Blur, Crop and Swirl (as in Heusel et al., 2017). The cropping for example randomly selects a 24×24, 18×18 or 12×12 pixel section of the image, depending on the level, and then scales it back to 32×32 pixels.
Again, only 1000 samples - from the 'Automobile' class in Borderland – were used to ensure comparability of the FCD values. The distortions were increased in three steps. It is expected that the higher the level, the higher the Fréchet Distance for all models. Figure 4.3 shows how $FC_{bm}D$ behaves for the three distortion methods with the embedding space of the best Custom Model. For all three distortions, and for all levels within a distortion, the custom Fréchet Distances give comprehensible values. While for no distortion a value close to zero is obtained, for $FC_{bm}D$ the value in-

creases almost continuously with the strength of the distortion. Only $FC_{sm}D$, which uses the embedding space of the simple custom model, does not increase so rapidly with blur and distortion up to level 2, but from level 2 to level 3 there is a much more abrupt increase (see 5). The first two levels of blur and distortion are therefore judged to be more similar to the original data by $FC_{sm}D$ than by $FC_{bm}D$ and $FC_{om}D$.
The highest values for the FCDs using the three custom models were obtained at level 3 for the cropped images, which means that these images were judged to be the most dissimilar to the original images. In contrast, the highest FID was obtained at level 3 for the swirled images (see 7).
In summary, all of the custom Fréchet Distances using the custom model's embedding spaces perform well when dealing with distortion on images whose classes the custom models know from training.



(a) Blur

(b) Crop

(c) Swirl

Figure 4.3: Changes in $FC_{bm}D$ values depending on the intensity of the disturbance

4.6 Performance on Fashion-MNIST

In this section the performance on data not related to CIFAR-10 is now being considered. It is anticipated that this will demonstrate the impact of the embedding space on the estimation of the Fréchet Distance on data, which greatly differs from the training data.
Therefore the Fashion-MNIST dataset is utilized, given its significant differences from CIFAR-10 and smaller image sizes (28x28). This ensures that no information is lost during the preprocessing of data for custom models, as they do not require downsizing but rather upscaling.
The FID is employed as a benchmark for the FCDs, as Inception-v3 is trained on ImageNet and, in contrast to the custom models, can more effectively classify this type of image content (see 8d). This should also enable FID to achieve accurate scores.
In comparison to Inception-v3, the custom models demonstrate a higher inability to recognize any items of clothing, a finding that is reflected in the wide spread of all classes in the latent space of all three models (see 8). At most, one can easily guess that the shoe classes and the bag class are slightly separated.
To this end, all ten classes were compared on an individual basis, with all Fréchet Distances between the individual classes (including the comparison of each class with itself) being determined. The results show that, as expected, Inception-v3 provides explainable and meaningful FIDs for all 55 comparisons (see 8).
The subsequent analysis will focus on the custom models. The analysis of these models reveals that the embedding space of the best Custom Model performs the worst, as at least three clear false relations are permitted by the $FC_{bm}D$. In the case of the overfitting Custom Model's embedding space, two relevant false relations are observed, while in the simple Custom Model's embedding space, a single false relation is identified within a single class.
Specifically, the relevant false relations appear in the following classes/comparisons:
The class 'Ankle boot' achieves an almost equal $FC_{bm}D$ with the embedding space of the best Custom Model compared to the classes 'Sandal', 'Pullover' and 'Sneaker'. However, it is evident that the 'Pullover' class actually has less similarity to the 'Ankle boot' class than the other two shoe classes, since they are shoes.
Also, the 'Ankle boot' class is not correctly evaluated by using the overfitting and the simple Custom Model's embedding space in terms of $FC_{sm}D/FC_{om}D$, since using the simple Custom Model's embedding space, the 'Shirt' and 'Trouser' classes are both most similar to the 'Ankle boot' class (except for the 'Ankle boot' with itself). The remaining footwear categories attain a higher $FC_{sm}D$ with ankle boots. The $FC_{om}D$ determined with overfitting Custom Model's embedding space rates the ankle boots and coats as the most similar. Only sandals demonstrate a similar $FC_{om}D$, as do shirts and bags. Sneakers are only rated more similar to ankle boots as Dresses and Trousers.
The 'Ankle boot' class presents challenges for all custom models.

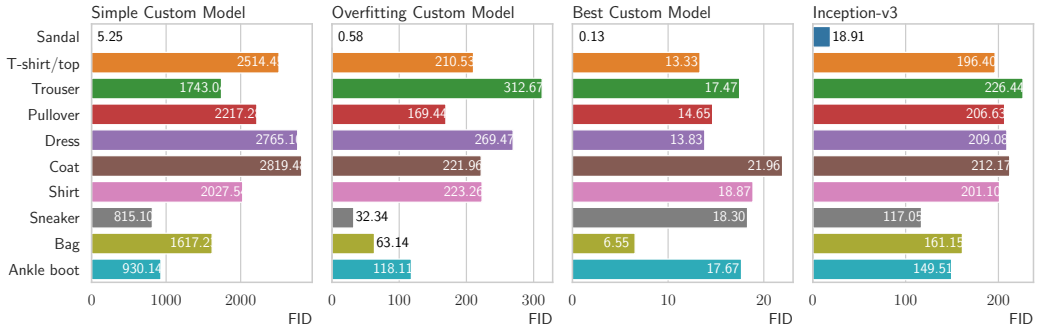


Figure 4.4: Class-Specific Performance of $FCDs$ and FID on 'Sandal' Class

In a further comparison, the class 'Dress' achieves the second lowest and lowest FCDs compared to the class Trousers and vice versa for the embedding space of the overfitting and the best Custom Model. This is not correct either, as the embedding space of the simple Custom Model and also of Inception-v3 show that the dress has the highest similarity to all other upper body garments and that the trousers, at least with the Inception model, have no really similar garment.
Finally, the embedding space of the best Custom Model does not allow a valid $FC_{bm}D$ to be calculated for the 'Sandal' class. The lowest $FC_{bm}D$ of 6.55 is obtained between the sandals and the bags, while the $FC_{bm}D$ with the other shoes are more than twice as high. Using the other two custom models as feature extractors, the smallest $FC_{sm}D/FC_{om}D$ is obtained with the 'Sneaker' class in this comparison, which also makes sense.

Overall, it can be seen that the most unreliable Fréchet Distance was obtained using the embedding space of the best Custom Model, while using the simple and overfitting Custom Model to calculate the FCD performed better. Of the three custom models, the use of the simple Custom Model as a feature extractor produces the fewest serious errors.
One possible explanation for this is that the simple Custom Model did not learn the CIFAR-6 data as well as the other two custom models (see B). This results, for example, in an $FC_{sm}D$ value that is not close to zero when comparing equal classes (see 4.1). Although it is not as large as the FID value, it is much larger than the $FC_{bm}D/FC_{om}D$. Accordingly, the simple Custom Model is not as strongly adapted to the CIFAR-6 data. It therefore does not perform particularly well on any data, but better than the models specialized on CIFAR-6.
But overall it is shown that a reliable Fréchet Distance cannot be determined using the custom models, especially not with well-trained models on certain data. The custom models are trained on CIFAR-6 whereas Inception-v3, on the other hand, does what it is supposed to do by being trained on ImageNet, which of course

contains images of garments. Therefore, it is clear that using the custom model's embedding space does not perform well on unknown image domains, which argues that it could be transferred that Inception-v3 cannot be used reliably to determine a Fréchet Distance for data that differs greatly from the ImageNet content.

Chapter 5

FID's Limitations on Sample Size

In the previous chapter, it became apparent that FID cannot determine values close to 0 at a sample size of 1000 when the same classes are compared (see 4.1, 4.2 and C). These values do not indicate that the distributions of the two datasets would be similar if they were compared. But they are, as the FCDs were able to determine correctly.
Accordingly, it is easy to draw false conclusions from these results. As the FCDs can handle this setup correctly, the problem is not the Fréchet Distance itself, but Inception, or rather the feature dimension of Inception and its dependence of this on the sample size. Because the Fréchet Inception Distance, which is based on averages and covariances, breaks down under the weight of small samples. And so the covariance matrices, which are necessary for the calculation of FID, are of a large size and cannot be sufficiently estimated with a small sample size, which means that FID is no longer reliable.
The subsequent chapter will address the question of wether and how this limitation can be addressed.

5.1 How FID handles Small Sample Size

Figure 4.1 shows that the value for experiment A for Inception V3 is 43.36, while it is almost 0 for all other models. It is also expected that the value should be close to 0, as two datasets from the same data-pool (images are not identical for both sets) are compared.
It is easy to assume that this is due to the number of only 1000 samples used per dataset, since the custom models use lower feature dimensions (128) than Inception-v3 (2048). However, Figure 5.1 shows how significant the difference for different sample sizes in terms of FID scores actually is.
Again, two datasets from the same pool of CIFAR-10 images were compared. Therefore a FID value of 0 is expected, which is almost achieved with 20,000 samples. But if we were to compare only 5000 samples each and not put them into context,

we would quickly come to the erroneous conclusion that the datasets and their distributions are not similar, as the FID value is 9.65.
Is it possible, however, to be able to keep the misleading FID values in check a little? This is what will be addressed in the upcoming sections.

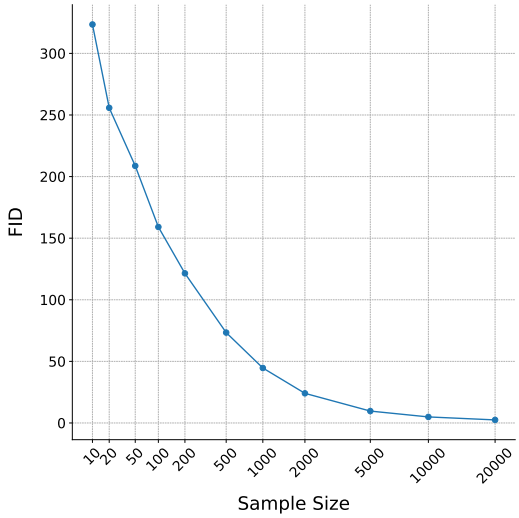


Figure 5.1: Behavior of FID Depending on Sample Size

5.2 Reduction Methods

To address the problem of unreliable FIDs with small sample sizes, it is necessary to take a closer look at its causation. As mentioned in 2.3.1 this is related to the estimation of the covariance matrices. In the Inception-v3 model, these are 2048×2048, i.e. they have more than 4 million entries that must be filled in order to obtain a valid FID.
However, if there are not enough samples available, the matrix can only be poorly estimated. In addition, one can also ask whether the poor performance is due to the fact that the Gaussian distributions of the features are not given but this will not be addressed here.
For the following, let us assume that $n = 1000$ samples and $p = 2048$ features. As mentioned above, the covariance matrix is therefore 2048×2048 high-dimensional and underdetermined. The estimation of covariance is unstable since $n \ll p$ and many eigenvalues are zero or noisy.
Since the FID is calculated as in 2.2, this means that the two covariance matrices with $n < p$ for each Σ_r and Σ_g are singular. As they are not linearly independent and their rank can be at most $min(n-1, p) = min(999, 2048) = 999$, because Σ_r and Σ_g are calculated as $\frac{1}{n-1} X^T X$, which also means that both covariance matrices have at least $p - (n-1) = p - n + 1$, so in our case $2048 - 1000 + 1 = 1049$ eigenvalues

that are zero (Horn & Johnson, 1985).
Thus, for both matrices, there exists a vector $v \neq 0$ such that:

$$\Sigma_r v = 0 \text{ and } \Sigma_g v = 0$$

The product of these two singular matrices Σ_r and Σ_g is hence also zero, because:

$$(\Sigma_r \Sigma_g)v = \Sigma_r(\Sigma_g v) = 0$$

Since $\Sigma_r \Sigma_g$ again has at least one zero eigenvector, $v \neq 0$ exists with $v^T(\Sigma_r \Sigma_g)v = 0$. In the formula for the Fréchet Inception Distance, the root is taken from the matrix product $\Sigma_r \Sigma_g$, which is usually done by eigenvalue decomposition $\Sigma_r \Sigma_g = U D U^T$. Here, D is the diagonal matrix with the eigenvalues. Therefore, with zero eigenvalues, the problem arises that $0^{\frac{1}{2}}$ is not clearly defined and becomes numerically unstable. This leads to problems such as division by zero or numerical errors in calculations, especially when there are many zero eigenvalues(Strang, 2014).

If p is reduced to 128, the following applies for p_{red}: $n > p_{red}$ and the covariance matrix 128×128 for 1000 samples can be estimated more stably.
The stable estimation leads to similar matrices, even if the reduction methods select slightly different features. This is clarified in more detail later on. The reduction is tested by mean pooling, Principle Component Analysis (PCA) and random choice:

- Mean pooling reduces 2048 features to 128 by averaging consecutive groups of 16 features.
- The reduction by PCA is performed as follows: Firstly, each characteristic (column) is adjusted by the mean value:

  $$X_{centered} = X - \bar{X}$$

  Then the covariance matrix $\Sigma_{original}$ (shape: $(2048, 2048)$) of the centered data is calculated as

  $$\Sigma_{original} = \frac{1}{n-1} X_{centered}^T X_{centered}$$

  This is followed by the decomposition into the eigenvalues, where $\lambda_1, \lambda_2, ..., \lambda_{2048}$ represents the variance along the main diagonal. The eigenvectors $v_1, v_2, ..., v_{2048}$ define the direction of the main components. Then these are sorted in descending order according to their eigenvalues, whereby the first 128 eigenvectors $v_1, v_2, ..., v_{128}$ are selected to form the projection matrix W of size $(2048, 128)$. Lastly, the centered data is then projected into the lower-dimensional space:

  $$X_{reduced} = X_{centered} \cdot W$$

In the end, the reduced features with $(1000, 128)$ shape are obtained. This ends up being a diagonal matrix whose diagonal contains the eigenvalues $\lambda_1, \lambda_2, ..., \lambda_{128}$ of the original data, while all non-diagonal elements are 0, since the principal components are not correlated if the covariance matrix from these reduced features is calculated.

$$\Sigma_{diagonal} = \begin{pmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_{128} \end{pmatrix}$$

- The random feature selection randomly chooses 128 unique indices from the 2048 available columns. Only columns with the selected indices are extracted from the features. Both datasets use the same indices to obtain comparable subsets of characteristics.

Figure 5.2 shows how the three reduction methods differ: In PCA, the feature maps were always reduced that apply $n > p$, where p always corresponds to a power of 2. For example, the features were reduced to 1024 dimensions for 2000 samples and to 64 features for 100 samples. There is no significant effect on the FID score when the feature's dimension is further downsampled with PCA.
Using these FID scores as a guide, it was tested how similar values can be obtained with mean pooling and random choice. In the end, $\sqrt{n}$ was rounded up to the nearest power of 2. This means that the features were reduced to 64 dimensions for 2000 samples and 16 dimensions for 100 samples. As you can see in figure 5.2, there are barely any differences in performance for the different reduction types.

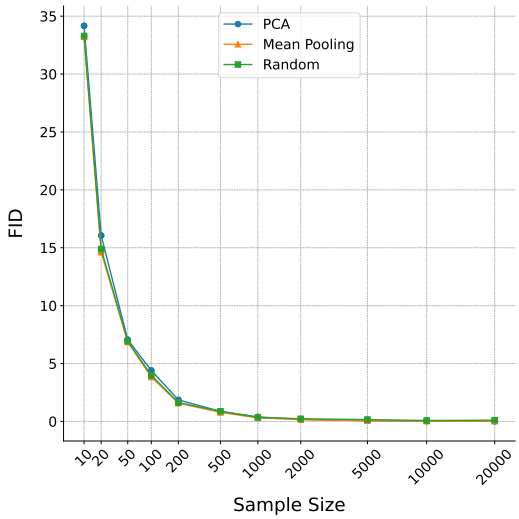


Figure 5.2: Different Reduction Methods

On the one hand, this could be due to the *flat* eigenvalue distribution of the covariance matrix; i.e., if the variance in the data is homogeneously distributed over many features (flat eigenvalue distribution), many features contribute similarly to the total variance. Specifically, in PCA this means that the first 128 principal components do not capture a dominant proportion of variance, as there is no clear *top 128* structure. In mean pooling it also ensures that variance proportions are obtained that are similar to PCA if no features dominate.
This means that the covariance matrices are structurally similar after reduction, since no method offers significantly better or worse variance preservation.
Furthermore, Inception-v3 generates redundant and highly correlated features. Many of the 2048 features encode similar visual patterns (e.g. edges, textures) since its features (from the last layer before classification) typically have high redundancy. In addition, global correlations exist, this means that features are often globally correlated across the image. Mean Pooling for example can destroy local pattern or correlations but the global structure can be maintained.
Although upscaling from 32×32 to 299×299 can create artifacts, Inception-v3 still extracts stable high-dimensional patterns because it has been trained on large images. The consequence is that each reduction method (mean pooling, PCA, random) accesses the same underlying correlation patterns, even if it selects different features. These global correlations are obtained as stated earlier by pooling the mean features by averaging the correlated features into groups (preserving the group correlations). PCA projects the correlations to the principal components that maximize linear correlations, and for random selection the correlation structure is partially preserved if randomly correlated features are selected. However, there is no guarantee that these features are informative or representative. There is a risk of selecting redundant or unimportant features.

In the following a visualization using a real-world example which explains all three methods is presented:
Imagine being tasked to describe a forest with 2048 trees, but that the available space for description is limited to 128 keywords. When using mean pooling, one is able to describe groups of trees ('10 fir trees in the north', '5 birch trees in the south').
In contrast, PCA enables you to list the most common tree species ('fir, birch, oak'). By random chance, one might name 128 random trees, which occur mainly as firs and birches.
The result is that all descriptions emphasize 'fir and birch' because they are dominant. The differences in methodology are hardly significant.

But it is evident that each of these reductions filters out noise and focuses on dominant patterns that are captured similarly regardless of the method.

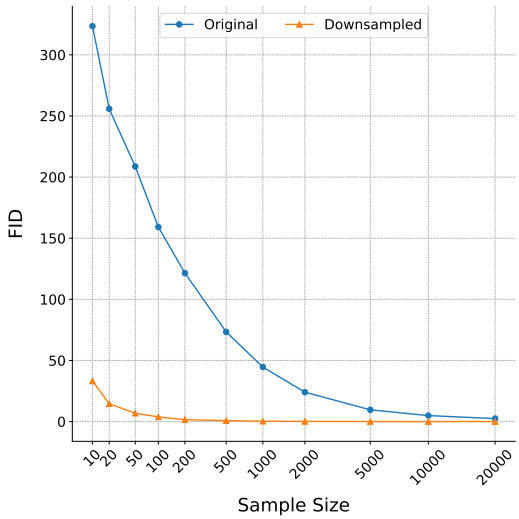


Figure 5.3: Comparison of FID Scores of Original and Reduced Feature Dimensions

The performance of the reduction for very similar datasets can be seen in 5.3. Due to the reduction, valid values for this data comparison are already recognizable with 500 samples (FID=0.86).
How the FID behaves with reduced feature dimensions and different datasets and to what extent correlations between results can still be explained is examined in the next section.

5.3 Probing with Various Data

That the reduction of the feature dimensions works well with an expected value of 0 has just been discussed. But now the general validity needs to be checked. Consequently it has to be confirmed if the reduced versions behave similarly to the original.
It is evident that definitive statements concerning absolute FID values cannot be made without fulfilling all requirements (Gaussian, sample size, etc.). Still, if 1000 samples are used consistently, as in the previous chapter, it should be possible to make statements such as whether frogs are more similar to cars or horses.

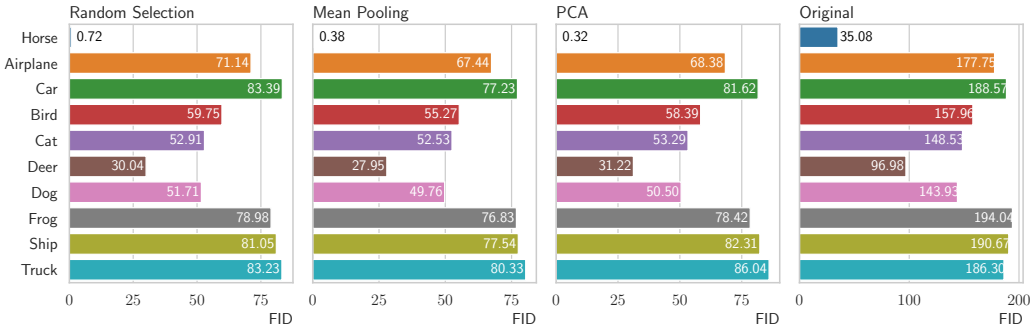


Figure 5.4: Fréchet Distances of the 'Horse' Class in Comparison with All Other Classes Using the Embedding Space of Inception-v3, with and without Feature Dimensionality Reduction

Therefore the experimental setup given in Chapter 4 will be used (including comparisons of multiple classes as well as individual comparisons of all ten classes of CIFAR-10), but the focus will now be on the performance of the reduced Fréchet Distance compared to the original one.
It is expected that a reduction in the FID from 30-40 to almost 0 for images from the same classes will result in a likely reduction in the FID when comparing images from different classes. Despite this, the FID score ratios should remain similar in relation to each other.

And that is exactly what happens – both by looking at the behavior of the FID on the comparisons of specific individual classes (F.2) and by comparing sets consisting of several classes (4).
In order to compare individual classes, the 'Horse' class is examined as an example in Figure 5.4. It becomes clear that although the absolute values for the FID decrease with the downsampled dimension of the feature vectors (and thus of the covariance matrices), the relationships of the results remain almost identical to the original FID. The most similar class remains constant for all FID calculations. Only for values that are close to each other in the original FID, it is possible that small, irrelevant shifts occur due to the dimension reduction of the feature vectors, e.g. in the class 'Horse' (see Figure 5.4):
For the original FID, the following relation applies in comparison with 'Frog', 'Ship' and 'Truck':

$$\text{FID of 'Frog'} > \text{FID of 'Ship'} > \text{FID of 'Truck'},$$

but for the FID with reduced covariance matrices,

$$\text{FID of 'Truck'} > \text{FID of 'Ship'} > \text{FID of 'Frog'} \text{ applies.}$$

But this is negligible because the main message is with the 'Horse' class, that all three of them have almost the same dissimilarity to the class 'Horse'.

The reduction-modified FIDs also perform well when comparing multiple classes. Specifically, this means that although the absolute FID scores are smaller, the relation is identical to that of the original FID, while the outlier in the comparison of identical classes is eliminated (see 4).
And this precisely was the starting point for these experiments: the behavior of the FID when comparing images of the same class. In this instance, FID remains almost constant at 0. Thus, by reducing the feature vectors and thus the covariance matrices, it may be possible to avoid the pathology of similar classes that occurs with small sample sizes.
The remaining relations of the values of FID are ensured to not change significantly as a result of the reduction, at least to the extent of this analysis.

Chapter 6

Discussion and Outlook

The present work has dealt intensively with the topic of distance measurement in the context of generative models. The investigation of a paradox underlies the entire work: how can a seemingly reliable measuring instrument become unreliable?
Through a analysis of the foundations, limitations and adaptations of FID, it was determined that its reliability is not intrinsic but dependent - on the assumptions embedded in its architecture, the data on which it is trained, and the contexts in which it is applied.
The code for the custom models, the custom Fréchet Distance as well as the downsampling methods is available at: https://github.com/41nj/Thesis-FID-2025

In the first part of this thesis, the theoretical foundations of FID and the underlying Inception-v3 network were presented in detail. It was demonstrated that FID provides an effective method to measure differences between real and generated images by mapping image distributions into a high-dimensional feature space. But the critical analysis also revealed significant limitations: the strong dependence on the Inception architecture trained on ImageNet introduces inherent biases that turn out to be particularly problematic in domains very different from ImageNet content, as stated by Kynkäänniemi et al. (2023).
The implementation of customized models and the use of their embedding spaces to compute the Fréchet Distance showed on the one hand that it would be possible to adapt the models to specific application areas, since they were able to give reliable results on the known data and on the other hand, that the Fréchet Distance depends on the underlying feature extractor, i.e. the training data of the model.
The latter was demonstrated by training the custom models on CIFAR-6, but then calculating the custom Fréchet Distances on CIFAR-10. For this data, except for the class 'Bird', comprehensible results were achieved. This is likely due to the fact that the four classes not known to the custom models generally show similarities to the CIFAR-6 data. But overall for the data the models were trained on, the FCDs were valid. This shows that training and then using custom models to calculate the Fréchet Distance can be a useful method for specific data when FID cannot provide

reliable results on it.

When calculating the custom Fréchet Distance on Fashion-MNIST, i.e. on data that is very dissimilar to the CIFAR-6/-10 data, no reliable results could be obtained. This is attributed to the significant differences between this data and the CIFAR data, and therefore the custom models on Fashion-MNIST did not perform well in the classification. It can be deduced that if models perform poorly on data, the results obtained using the Fréchet distance on the embedding spaces of these models should be treated with caution.
The Fashion-MNIST case shows, firstly, that custom models can work as feature extractors only for a specific domain (the one on which they were trained) and, secondly, that when transferred to a larger context, the FID determination could be restricted – namely to the ImageNet domains. These findings demonstrate the potential value of training custom models and utilizing their embedding space to calculate Fréchet Distance, provided that the data does not overlap with images from ImageNet.

The second part of this work addressed the limitation of FID due to small sample sizes (Chong & Forsyth, 2020; Jayasumana et al., 2024).
An approach to reduce the feature dimension was presented which improves the stability and power of FID, especially for small samples, by avoiding the singularity of the covariance matrices. Consequently, the high FID scores that are frequently observed when comparing images with high similarity can be significantly reduced without compromising the quality of the FID. In practice, it could be demonstrated that the reduction did lead to smaller absolute FID scores, but the relationships between the scores – in this case, the relationships between different classes of CIFAR-10 – remained stable.
However, it is important to note that this reduction in feature dimension requires further validation to ensure its efficacy.

Consequently, future research should perform larger tests with this simple implementable reduction to confirm or deny a general, robust applicability.
In addition, other potential contributors to the more valid results should be excluded in order to verify that the results depend only on the reduction, e.g. the role of distribution.
Subsequent comparisons of the results with CMMD introduced by Jayasumana et al., 2024 or $\overline{FID}_{\infty}$ introduced by Chong and Forsyth, 2020 may then be made. It is important to note that both Jayasumana et al. (2024) and Chong and Forsyth (2020) propose an alternative to the original FID, exactly for the reason that this work is being done – the limitations of FID.
Jayasumana et al. (2024) introduce the metric CMMD (utilizes CLIP embeddings and the Maximum Mean Discrepancy), which 'is an unbiased estimator that does not make any assumptions on the probability distribution of the embeddings and is

sample efficient' (Jayasumana et al., 2024, p. 1). In order 'to obtain an effectively bias-free estimate of scores computed with an infinite number of samples' (Chong & Forsyth, 2020, p. 1), it is demonstrated how the score can be extrapolated, which Chong and Forsyth called $\overline{FID}_\infty$.
For the element of this work that deals with using the embedding spaces of custom models to compute the Fréchet Distance, more detailed analyses can and should be done. As this component is conceptualized as a proof of concept, its validation is necessary to determine the extent to which it makes sense to train custom models for domains where FID fails.
One potential approach to validate this would be to e.g. train a custom model on satellite images, given that such data is not part of ImageNet and exhibits significant differences from average images due to its different channel structure (non-RGB). The next step would be to use the embedding space of this model to determine and analyze the Fréchet Distance. It is only then that the proof of concept can be confirmed by a concrete application.

It would also be interesting to test the application of the Fréchet Distance using custom models and the reduced FID with generated images. Because in this work only very simple data was used for the implementations and the Fréchet Distance was determined only for different class constellation comparisons. This means that statements about these cases can be made, but it is not possible to state the impact on similarities between real and truly generated data.
Due to the lack of time and resources, it was not possible to test the effects of custom distances in the training of GAN's. It would be interesting to find out whether the use of custom Fréchet Distances can lead to realistic images for those areas not covered by ImageNet.

In conclusion, it is essential to remember that metrics such as the Fréchet Inception Distance, although convincing in their mathematical conclusiveness, should not be misunderstood as universal quality indicators. As previously mentioned in the introduction, distances are not just abstract numerical values, but rather an essential bridge between theory and practice, demonstrating the context-dependent characteristics of our measurement approaches.
In this thesis I have demonstrated that FID, despite its widespread use and theoretical underpinning, is only as reliable as the assumptions and models on which it is based. I have also clarified that the reliance on pre-trained networks that have been optimized on specific datasets – such as CIFAR-6 – underlines that without contextual validation and a conscious handling of the underlying assumptions, distortions and misinterpretations cannot be eliminated.
This finding enforces a critical discourse that addresses the general application of distance measures and goes beyond the FID. In addition to the existing models being further developed and validated, it is also important to evaluate alternative approaches in order to ensure that the quality content is evaluated on a diverse and

reliable basis.

This work challenges the assumption of absolutness of any established metric, but also to maintain a critical eye – an essential step in the age of artificial intelligence, which teaches us to constantly question the limitations and context of our measurement tools.

References

Bischoff, S., Darcher, A., Deistler, M., Gao, R., Gerken, F., Gloeckler, M., Haxel, L., Kapoor, J., Lappalainen, J. K., Macke, J. H., Moss, G., Pals, M., Pei, F., Rapp, R., Sağtekin, A. E., Schröder, C., Schulz, A., Stefanidi, Z., Toyota, S., … Vetter, J. (2024). A practical guide to sample-based statistical distances for evaluating generative models in science.

Chong, M. J., & Forsyth, D. (2020). Effectively unbiased fid and inception score and where to find them.

Dowson, D. C., & Landau, B. V. (1982). The fréchet distance between multivariate normal distributions. *Journal of Multivariate Analysis*, *12*(3), 450–455.

Gebhardt, H., & Kiesel, H. (2013). *Weltbilder*. Springer Berlin Heidelberg.

Hanslmeier, A. (2002). *Einführung in astronomie und astrophysik* (Vol. 2). Springer.

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep residual learning for image recognition.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., & Hochreiter, S. (2017). Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, *30*.

Horn, R. A., & Johnson, C. R. (1985). *Matrix analysis*. Cambridge University Press.

Jayasumana, S., Ramalingam, S., Veit, A., Glasner, D., Chakrabarti, A., & Kumar, S. (2024). Rethinking fid: Towards a better evaluation metric for image generation.

Kingma, D. P., & Ba, J. (2017). Adam: A method for stochastic optimization.

Kynkäänniemi, T., Karras, T., Aittala, M., Aila, T., & Lehtinen, J. (2023). The role of imagenet classes in fréchet inception distance.

Lucic, M., Kurach, K., Michalski, M., Gelly, S., & Bousquet, O. (2018). Are gans created equal? a large-scale study.

Pascanu, R., Mikolov, T., & Bengio, Y. (2013). On the difficulty of training recurrent neural networks.

Ptolemy. (1999). *Ptolemy's almagest* (G. Toomer, Ed.). Princeton University Press. https://doi.org/doi:10.1515/9780691213361

Sajjadi, M. S. M., Bachem, O., Lucic, M., Bousquet, O., & Gelly, S. (2018). Assessing generative models via precision and recall.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., & Chen, X. (2016). Improved techniques for training gans.

Strang, G. (2014). *Linear algebra and its applications*. Elsevier Science.

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

Tan, M., & Le, Q. V. (2020). Efficientnet: Rethinking model scaling for convolutional neural networks.

Taylor, C. (2007). *A secular age*. Harvard University Press.

Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., & Jégou, H. (2021). Training data-efficient image transformers & distillation through attention.

van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-sne. *Journal of Machine Learning Research*, *9*(86), 2579–2605.

Appendix

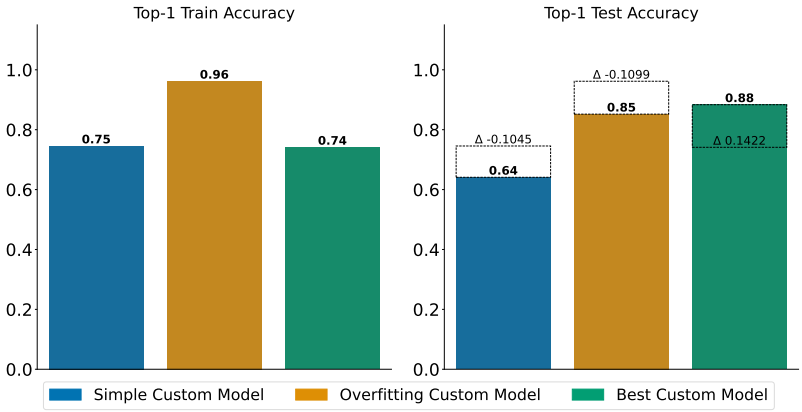
Background for Inception-v3's Comparison with other Models

ResNet-50: A CNN that connects layers with direct skip connections so that the network can be trained very deeply without the learning ability collapsing. In addition, narrow 1×1 convolutional filters ensure that computing power is saved before larger 3×3 filters are applied (He et al., 2015).

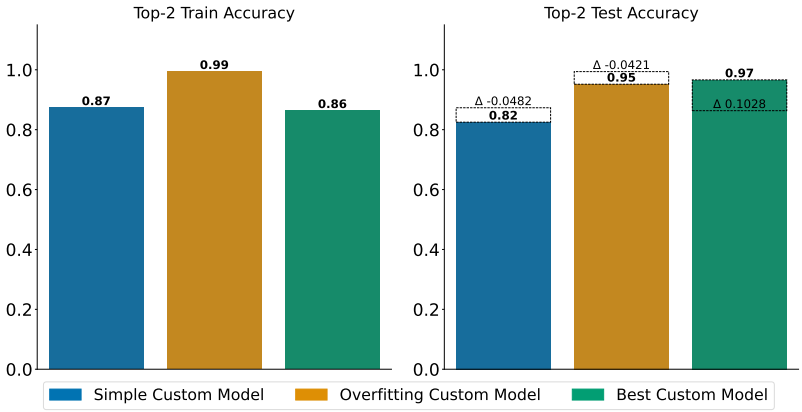
DeiT-B: A vision transformer with distillation which splits the image into small squares (e.g. 16×16 pixels) and analyzes their relationship to each other with a kind of attention mechanism (self-attention). It also learns from an already trained CNN model (teacher) to become good even with little data(Touvron et al., 2021).

EfficentNet-B7: A scaled CNN with MBConv blocks, which scales network depth, width and image resolution evenly - not too much, not too little - for optimal efficiency. The MBConv blocks are like reverse sandwich layers: First it expands the channels (e.g. $1 \rightarrow 6$), then it applies light filters and reduces them again ($6 \rightarrow 1$) to save computing power (Tan & Le, 2020).

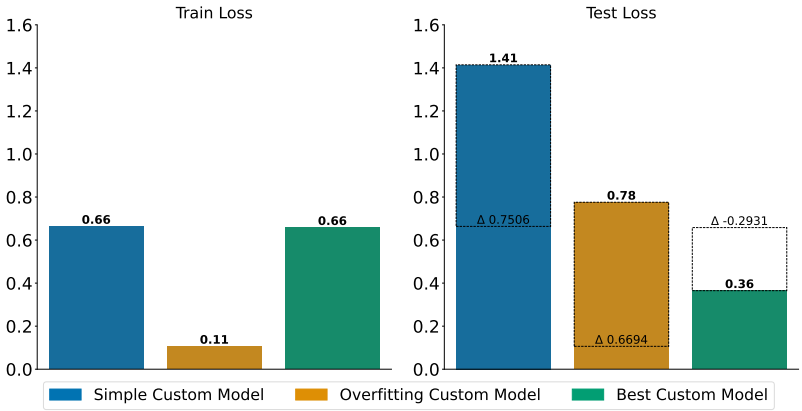
Custom Model's Performance in CIFAR-6/-10 Classification



(a) Top-1 Accuracy

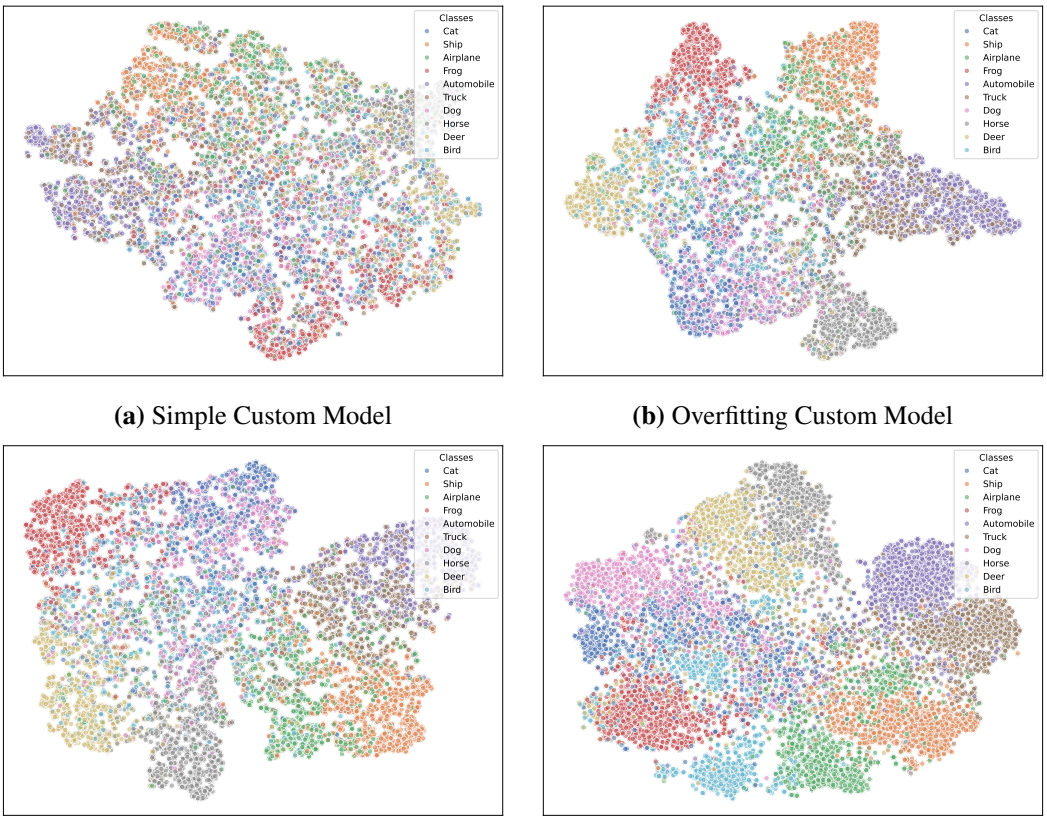


(b) Top-2 Accuracy



(c) Loss

Figure 1: Evaluation and Comparison of the Custom Model's Performance in Classification of CIFAR-6



(a) Simple Custom Model

(b) Overfitting Custom Model

(c) Best Custom Model

(d) Inception-v3

Figure 2: t-SNE Visualization of Latent Spaces from Different Models on CIFAR-10

Experiments on CIFAR-10

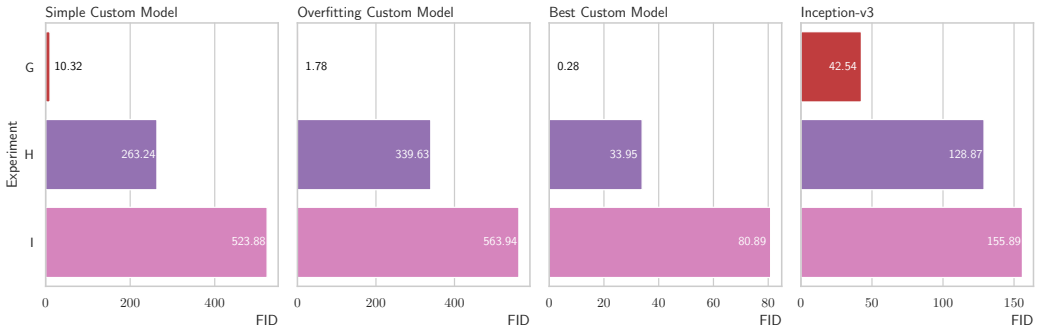
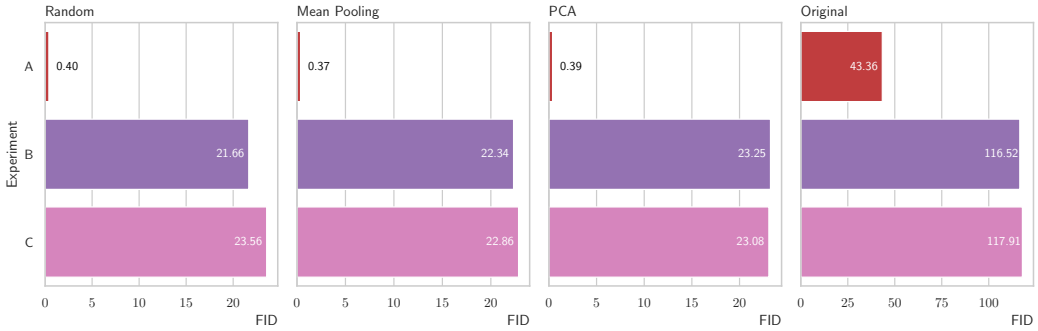
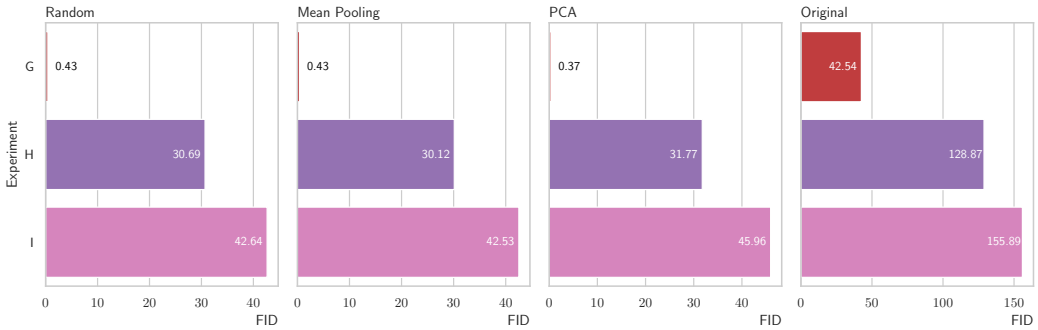


Figure 3: Fréchet Distances on Home Ground, Borderland and Uncharted Territory using the Embedding Space of the Custom Models and of Inception-v3
Experiment G: Experiment D: Comparison of Different Images from the Same (all) Classes in Uncharted Territory; **Experiment H**: Comparison of Classes 'Airplane'+'Bird' with 'Dog'+'Truck' on Uncharted Territory; **Experiment I**: Comparison of Classes 'Airplane' +'Truck' with 'Bird' +'Dog' on Uncharted Territory;



(a) Experiment A: Comparison of Different Images from the Same (all) Classes in Home Ground; **Experiment B**: Comparison of Classes 'Car', 'Cat', and 'Frog' with 'Deer', 'Horse', and 'Ship' on Borderland; **Experiment C**: Comparison of Classes 'Car', 'Cat', 'Frog' with 'Deer', 'Horse', 'Ship' on Home Ground

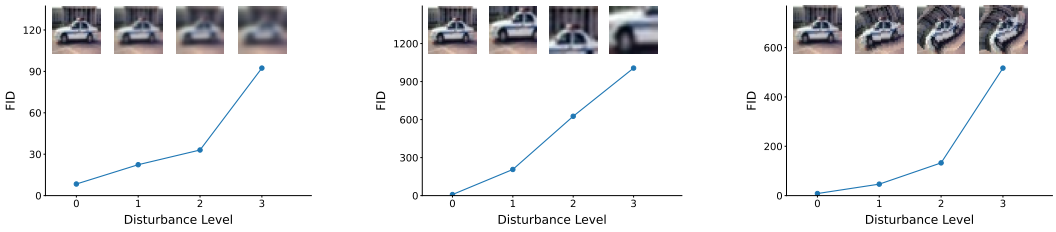
(b) Experiment D: Comparison of Different Images from the Same (all) Classes in Borderland; **Experiment E**: Comparison of all the Classes in Home Ground with all the Classes in Uncharted Territory; **Experiment F**: Comparison of all the Classes in Borderland with all the Classes in Uncharted Territory;



(c) Experiment G: Experiment D: Comparison of Different Images from the Same (all) Classes in Uncharted Territory; **Experiment H**: Comparison of Classes 'Airplane'+'Bird' with 'Dog'+'Truck' on Uncharted Territory; **Experiment I**: Comparison of Classes 'Airplane' +'Truck' with 'Bird' +'Dog' on Uncharted Territory;

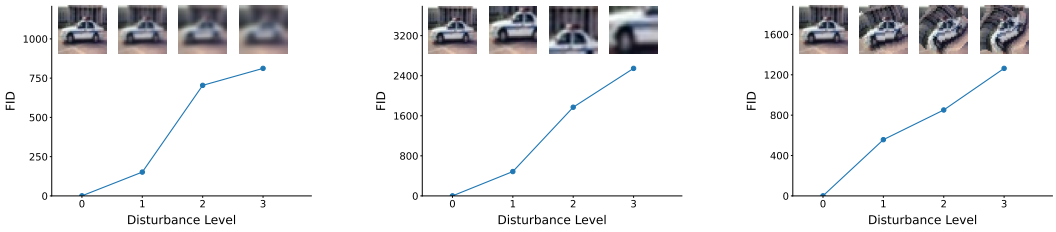
Figure 4: Fréchet Distances on Home Ground, Borderland and Uncharted Territory using the Embedding Space of Inception-v3, with and without reduced Feature Dimensionality

Fréchet Distances on Augmented Images



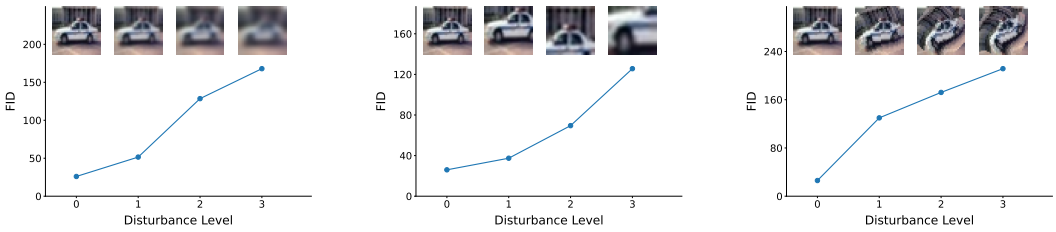
(a) Blur **(b)** Crop **(c)** Swirl

Figure 5: Changes in $FC_{sm}D$ Depending on the Intensity of the Disturbance



(a) Blur **(b)** Crop **(c)** Swirl

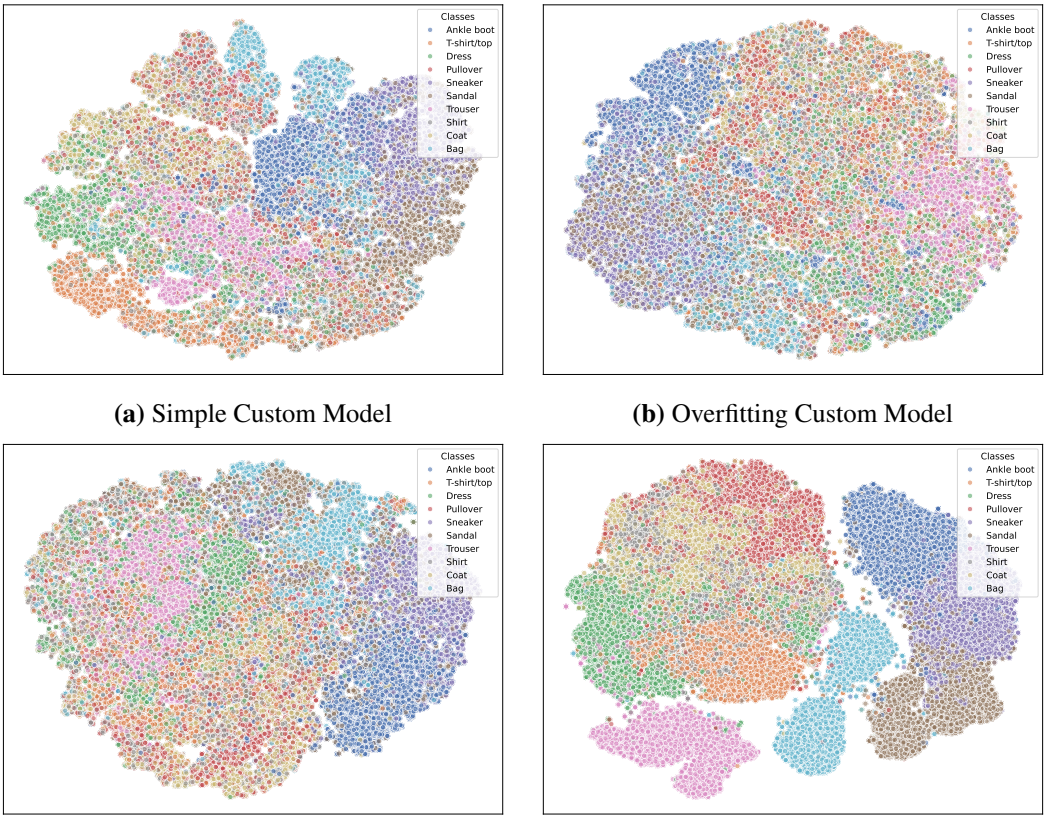
Figure 6: Changes in $FC_{om}D$ Depending on the Intensity of the Disturbance



(a) Blur **(b)** Crop **(c)** Swirl

Figure 7: Changes in FID Depending on the Intensity of the Disturbance

Latent Spaces of Models on Fashion-MNIST



(a) Simple Custom Model

(b) Overfitting Custom Model

(c) Best Custom Model

(d) Inception-v3

Figure 8: t-SNE Visualization of Latent Spaces from Different Models on Fashion-MNIST

Fréchet Distances on CIFAR-10

F.1 Custom Models

	Airplane	**Car**	**Bird**	**Cat**	**Deer**	**Dog**	**Frog**	**Horse**	**Ship**	**Truck**
Airplane	8.61	1680.36	371.46	752.25	464.52	703.33	1150.12	1685.88	353.15	364.31
Car	1680.36	12.98	2009.98	2209.71	2228.84	2268.58	2372.89	3333.85	2166.45	584.93
Bird	371.46	2009.98	7.77	229.75	105.50	229.92	461.27	1261.15	1336.31	620.14
Cat	752.25	2209.71	229.75	9.03	534.12	28.63	463.13	1657.96	1574.00	792.86
Deer	464.52	2228.84	105.50	534.12	4.77	582.04	621.43	1488.09	1477.35	857.09
Dog	703.33	2268.58	229.92	28.63	582.04	4.52	583.15	1383.69	1488.54	910.67
Frog	1150.12	2372.89	461.27	463.13	621.43	583.15	7.54	2430.55	2127.35	1086.77
Horse	1685.88	3333.85	1261.15	1657.96	1488.09	1383.69	2430.55	13.93	2925.73	1782.28
Ship	353.15	2166.45	1336.31	1574.00	1477.35	1488.54	2127.35	2925.73	9.50	1153.33
Truck	364.31	584.93	620.14	792.86	857.09	910.67	1086.77	1782.28	1153.33	10.81

Table 1: Fréchet Distances between CIFAR-10 Classes using the Embedding Space of the Simple Custom Model

	Airplane	**Car**	**Bird**	**Cat**	**Deer**	**Dog**	**Frog**	**Horse**	**Ship**	**Truck**
Airplane	0.38	1878.54	178.86	412.98	485.07	374.35	563.84	971.83	265.44	534.10
Car	1878.54	1.24	2454.13	2610.35	2793.77	2543.67	2720.03	2653.34	2208.26	365.49
Bird	178.86	2454.13	1.39	157.34	222.18	120.68	311.17	783.74	745.21	1042.47
Cat	412.98	2610.35	157.34	0.94	617.83	34.85	557.78	961.17	1103.13	1060.49
Deer	485.07	2793.77	222.18	617.83	1.03	533.20	758.15	905.04	1145.30	1368.31
Dog	374.35	2543.67	120.68	34.85	533.20	0.44	562.49	647.07	1099.85	1004.87
Frog	563.84	2720.03	311.17	557.78	758.15	562.49	0.53	1572.96	1107.57	1344.90
Horse	971.83	2653.34	783.74	961.17	905.04	647.07	1572.96	0.62	1656.28	1215.24
Ship	265.44	2208.26	745.21	1103.13	1145.30	1099.85	1107.57	1656.28	0.31	1065.37
Truck	534.10	365.49	1042.47	1060.49	1368.31	1004.87	1344.90	1215.24	1065.37	1.01

Table 2: Fréchet Distances between CIFAR-10 Classes using the Embedding Space of the Overfitting Custom Model

	Airplane	**Car**	**Bird**	**Cat**	**Deer**	**Dog**	**Frog**	**Horse**	**Ship**	**Truck**
Airplane	0.34	179.47	44.54	111.81	95.55	91.10	162.87	157.69	46.21	51.77
Car	179.47	0.33	282.12	342.32	394.91	315.97	337.56	409.94	205.70	61.42
Bird	44.54	282.12	0.22	36.24	49.63	21.77	58.13	131.64	162.33	106.05
Cat	111.81	342.32	36.24	0.25	125.51	6.12	83.08	187.07	241.10	145.45
Deer	95.55	394.91	49.63	125.51	0.25	94.24	170.42	140.87	247.90	206.14
Dog	91.10	315.97	21.77	6.12	94.24	0.19	86.14	123.72	233.42	130.83
Frog	162.87	337.56	58.13	83.08	170.42	86.14	0.15	323.80	283.06	198.60
Horse	157.69	409.94	131.64	187.07	140.87	123.72	323.80	0.13	314.25	191.12
Ship	46.21	205.70	162.33	241.10	247.90	233.42	283.06	314.25	0.23	96.95
Truck	51.77	61.42	106.05	145.45	206.14	130.83	198.60	191.12	96.95	0.22

Table 3: Fréchet Distances between CIFAR-10 Classes using the Embedding Space of the Best CUstom Model

	Airplane	**Car**	**Bird**	**Cat**	**Deer**	**Dog**	**Frog**	**Horse**	**Ship**	**Truck**
Airplane	35.81	143.29	139.71	162.27	161.53	178.39	187.58	177.75	104.25	152.05
Car	143.29	26.71	187.34	171.60	187.86	178.32	209.76	188.57	131.93	78.27
Bird	139.71	187.34	42.20	119.55	112.73	135.53	120.44	157.96	180.13	207.33
Cat	162.27	171.60	119.55	44.47	128.10	90.55	113.74	148.53	179.76	184.11
Deer	161.53	187.86	112.73	128.10	35.61	139.14	126.52	96.98	183.39	199.08
Dog	178.39	178.32	135.53	90.55	139.14	38.68	150.71	143.93	197.85	193.39
Frog	187.58	209.76	120.44	113.74	126.52	150.71	39.57	194.04	210.65	225.04
Horse	177.75	188.57	157.96	148.53	96.98	143.93	194.04	35.08	190.67	186.30
Ship	104.25	131.93	180.13	179.76	183.39	197.85	210.65	190.67	31.43	120.67
Truck	152.05	78.27	207.33	184.11	199.08	193.39	225.04	186.30	120.67	24.04

Table 4: Fréchet Distances between CIFAR-10 Classes using the Embedding Space of Inception-v3

F.2 FID with Reduced Feature Dimensionality

	Airplane	**Car**	**Bird**	**Cat**	**Deer**	**Dog**	**Frog**	**Horse**	**Ship**	**Truck**
Airplane	0.44	56.01	44.69	52.68	61.00	58.03	71.18	68.38	31.91	63.38
Car	56.01	0.26	77.49	68.48	82.61	69.10	100.07	81.62	51.88	29.51
Bird	44.69	77.49	0.39	28.90	32.39	32.73	32.95	58.39	69.43	92.71
Cat	52.68	68.48	28.90	0.43	40.87	16.50	31.38	53.29	69.42	80.33
Deer	61.00	82.61	32.39	40.87	0.41	43.91	42.68	31.22	74.55	89.01
Dog	58.03	69.10	32.73	16.50	43.91	0.41	48.15	50.50	70.98	79.70
Frog	71.18	100.07	32.95	31.38	42.68	48.15	0.40	78.42	89.22	107.43
Horse	68.38	81.62	58.39	53.29	31.22	50.50	78.42	0.32	82.31	86.04
Ship	31.91	51.88	69.43	69.42	74.55	70.98	89.22	82.31	0.34	52.41
Truck	63.38	29.51	92.71	80.33	89.01	79.70	107.43	86.04	52.41	0.27

Table 5: Fréchet Distances between CIFAR-10 Classes using the Embedding Space of Inception-v3, with Feature Dimensionality Reduced via PCA

	Airplane	**Car**	**Bird**	**Cat**	**Deer**	**Dog**	**Frog**	**Horse**	**Ship**	**Truck**
Airplane	0.38	53.42	45.01	53.04	59.33	59.81	73.75	67.44	31.70	64.59
Car	53.42	0.24	76.33	63.65	81.02	66.08	96.98	77.23	50.92	26.73
Bird	45.01	76.33	0.45	28.62	32.83	33.98	34.27	55.27	67.73	86.84
Cat	53.04	63.65	28.62	0.46	41.29	16.76	34.24	52.53	65.26	72.33
Deer	59.33	81.02	32.83	41.29	0.32	43.59	41.54	27.95	73.73	86.05
Dog	59.81	66.08	33.98	16.76	43.59	0.36	45.95	49.76	70.94	76.55
Frog	73.75	96.98	34.27	34.24	41.54	45.95	0.29	76.83	91.57	104.95
Horse	67.44	77.23	55.27	52.53	27.95	49.76	76.83	0.38	77.54	80.33
Ship	31.70	50.92	67.73	65.26	73.73	70.94	91.57	77.54	0.30	49.02
Truck	64.59	26.73	86.84	72.33	86.05	76.55	104.95	80.33	49.02	0.33

Table 6: Fréchet Distances between CIFAR-10 Classes using the Embedding Space of Inception-v3, with Feature Dimensionality Reduced by Mean Pooling

	Airplane	**Car**	**Bird**	**Cat**	**Deer**	**Dog**	**Frog**	**Horse**	**Ship**	**Truck**
Airplane	0.92	56.98	46.49	55.38	61.35	62.04	73.99	71.14	32.16	65.37
Car	56.98	0.62	78.79	66.91	83.69	68.05	99.65	83.39	52.73	27.37
Bird	46.49	78.79	0.80	30.57	36.18	34.87	34.65	59.75	70.40	88.97
Cat	55.38	66.91	30.57	0.88	40.63	17.54	33.83	52.91	68.89	76.59
Deer	61.35	83.69	36.18	40.63	0.79	47.53	42.83	30.04	78.98	88.92
Dog	62.04	68.05	34.87	17.54	47.53	0.81	47.32	51.71	74.04	77.31
Frog	73.99	99.65	34.65	33.83	42.83	47.32	0.78	78.98	91.35	108.61
Horse	71.14	83.39	59.75	52.91	30.04	51.71	78.98	0.72	81.05	83.23
Ship	32.16	52.73	70.40	68.89	78.98	74.04	91.35	81.05	0.72	51.54
Truck	65.37	27.37	88.97	76.59	88.92	77.31	108.61	83.23	51.54	0.45

Table 7: Fréchet Distances between CIFAR-10 Classes using the Embedding Space of Inception-v3, with Feature Dimensionality Reduced by Random Selection

Fréchet Distances on Fashion-MNIST

	T-shirt/top	Trouser	Pullover	Dress	Coat	Sandal	Shirt	Sneaker	Bag	Ankle boot
T-shirt/top	17.91	178.69	120.77	92.96	109.29	196.40	68.92	221.60	155.40	181.64
Trouser	178.69	13.53	167.29	137.58	171.42	226.44	176.51	266.15	203.10	238.53
Pullover	120.77	167.29	16.66	123.45	56.99	206.63	77.23	222.72	138.73	205.22
Dress	92.96	137.58	123.45	18.67	85.59	209.08	74.74	235.94	174.82	194.42
Coat	109.29	171.42	56.99	85.59	16.83	212.17	51.91	215.54	145.91	189.23
Sandal	196.40	226.44	206.63	209.08	212.17	18.91	201.10	117.05	161.15	149.51
Shirt	68.92	176.51	77.23	74.74	51.91	201.10	20.33	219.62	149.65	182.47
Sneaker	221.60	266.15	222.72	235.94	215.54	117.05	219.62	13.85	184.04	79.91
Bag	155.40	203.10	138.73	174.82	145.91	161.15	149.65	184.04	20.74	183.10
Ankle boot	181.64	238.53	205.22	194.42	189.23	149.51	182.47	79.91	183.10	14.33

Table 8: Fréchet Distances between Fashion-MNIST Classes using the Embedding Space of Inception-v3

	T-shirt/top	Trouser	Pullover	Dress	Coat	Sandal	Shirt	Sneaker	Bag	Ankle boot
T-shirt/top	0.09	6.60	4.29	4.28	7.07	13.33	2.90	42.71	16.93	28.98
Trouser	6.60	0.05	12.52	3.21	13.37	17.47	7.37	48.12	24.85	33.14
Pullover	4.29	12.52	0.08	10.39	2.30	14.65	4.11	37.07	18.77	17.51
Dress	4.28	3.21	10.39	0.06	10.68	13.83	5.16	41.06	23.57	33.68
Coat	7.07	13.37	2.30	10.68	0.15	21.96	5.22	41.21	29.25	21.29
Sandal	13.33	17.47	14.65	13.83	21.96	0.13	18.87	18.30	6.55	17.67
Shirt	2.90	7.37	4.11	5.16	5.22	18.87	0.13	52.17	23.11	30.70
Sneaker	42.71	48.12	37.07	41.06	41.21	18.30	52.17	0.09	28.49	18.95
Bag	16.93	24.85	18.77	23.57	29.25	6.55	23.11	28.49	0.12	29.08
Ankle boot	28.98	33.14	17.51	33.68	21.29	17.67	30.70	18.95	29.08	0.06

Table 9: Fréchet Distances between Fashion-MNIST Classes using the Embedding Space of the Best Custom Model

	T-shirt/top	Trouser	Pullover	Dress	Coat	Sandal	Shirt	Sneaker	Bag	Ankle boot
T-shirt/top	0.88	65.94	19.32	37.92	11.16	210.53	8.28	332.59	85.39	121.88
Trouser	65.94	0.45	108.60	30.25	67.86	312.67	75.03	417.16	197.92	156.63
Pullover	19.32	108.60	0.50	65.14	20.55	169.44	19.61	262.10	70.79	131.81
Dress	37.92	30.25	65.14	0.34	37.62	269.47	28.40	394.36	122.66	165.37
Coat	11.16	67.86	20.55	37.62	0.45	221.96	5.17	316.34	107.66	106.95
Sandal	210.53	312.67	169.44	269.47	221.96	0.58	223.26	32.34	63.14	118.11
Shirt	8.28	75.03	19.61	28.40	5.17	223.26	1.18	311.44	76.81	116.14
Sneaker	332.59	417.16	262.10	394.36	316.34	32.34	311.44	0.55	125.54	146.15
Bag	85.39	197.92	70.79	122.66	107.66	63.14	76.81	125.54	0.62	116.25
Ankle boot	121.88	156.63	131.81	165.37	106.95	118.11	116.14	146.15	116.25	0.43

Table 10: Fréchet Distances between Fashion-MNIST Classes using the Embedding Space of the Overfitting Custom Model

	T-shirt/top	Trouser	Pullover	Dress	Coat	Sandal	Shirt	Sneaker	Bag	Ankle boot
T-shirt/top	8.72	577.41	595.92	287.99	716.33	2514.45	375.09	4938.30	1695.27	1588.95
Trouser	577.41	3.18	434.95	352.32	476.83	1743.04	320.06	3611.52	1637.82	850.36
Pullover	595.92	434.95	2.58	414.28	210.09	2217.28	81.20	3998.92	917.41	999.11
Dress	287.99	352.32	414.28	3.95	228.51	2765.10	272.92	4712.34	1788.79	1220.44
Coat	716.33	476.83	210.09	228.51	3.74	2819.48	226.02	4571.30	1340.32	1130.74
Sandal	2514.45	1743.04	2217.28	2765.10	2819.48	5.25	2027.54	815.10	1617.23	930.14
Shirt	375.09	320.06	81.20	272.92	226.02	2027.54	5.59	3916.46	952.36	852.68
Sneaker	4938.30	3611.52	3998.92	4712.34	4571.30	815.10	3916.46	1.65	2182.93	1788.39
Bag	1695.27	1637.82	917.41	1788.79	1340.32	1617.23	952.36	2182.93	5.24	923.03
Ankle boot	1588.95	850.36	999.11	1220.44	1130.74	930.14	852.68	1788.39	923.03	2.78

Table 11: Fréchet Distances between Fashion-MNIST Classes using the Embedding Space of the Simple Custom Model

Visual Comparison of CIFAR-10 and Fashion-MNIST

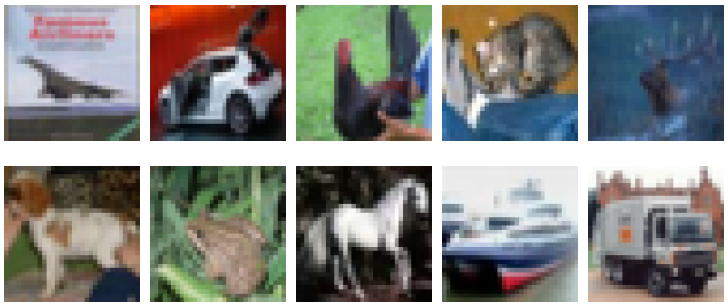


Figure 9: CIFAR-10

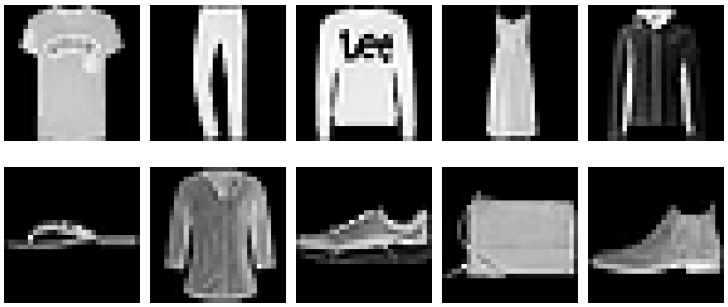


Figure 10: Fashion-MNIST

Selbständigkeitserklärung

Hiermit erkläre ich, dass ich diese schriftliche Abschlussarbeit selbständig verfasst habe, keine anderen als die angegebenen Hilfsmittel und Quellen benutzt habe und alle wörtlich oder sinngemäß aus anderen Werken übernommenen Aussagen als solche gekennzeichnet habe. Des Weiteren erkläre ich, dass die Arbeit weder vollständig noch in wesentlichen Teilen Gegenstand eines anderen Prüfungsverfahrens gewesen ist.

12.03.2025, Tübingen
Datum, Ort

M. Schu
Unterschrift