National College of Ireland

Evaluating the Impact of Data Augmentation on Image Classification Accuracy

MSc Research Project
Artificial Intelligence

Muhammad Anis Ur Rahman
Student ID: 23284803

School of Computing
National College of Ireland

Supervisor: Abdul Shahid

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Muhammad Anis Ur Rahman
Student ID:	23284803
Programme:	Artificial Intelligence
Year:	2025
Module:	MSc Research Project
Supervisor:	Abdul Shahid
Submission Due Date:	11/08/2025
Project Title:	Evaluating the Impact of Data Augmentation on Image Classification Accuracy
Word Count:	7683
Page Count:	27

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	15th September 2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	✓
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	✓
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	✓

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Evaluating the Impact of Data Augmentation on Image Classification Accuracy

Muhammad Anis Ur Rahman
23284803

Abstract

Data augmentation is the key to the better deep image classification model's stability and generalization. Although flipping, cropping, and rotating are famous methods, the latest researches in the fields of the most recent generative models and adaptive augmentation plans unveil the latest aspects of the synthesis of variant and task-adaptable training samples. In the present thesis, we investigate the comparative effect of three radically different strategies, namely, GAN-driven generation, Vector Quantized Variational Autoencoders (VQ-VAE), and mix-driven strategies like MiAMix.

The methods are compared on three representative test datasets, namely MNIST, CIFAR-10, and Tiny ImageNet, sharing a same convolutional neural network architecture and test pipeline. The methods are compared on classification accuracy and Frechet Inception Distance (FID), thereby double-click evaluation of predictive potential and sample faithfulness is supported. GAN methods are stable in class imbalance handling and image-richest generation of samples, and VQ-VAE shows stability and reconstruction performance on test sets. The combination methods are generally strong in low-data conditions with the benefit of computational speedup and simplicity of implementation.

The findings indicate that in terms of quantity, no one of the augmentation techniques is superior to others. Rather, the effects of the optimal solution would largely rely on the type of dataset, model sensitivity, and target of augmentation. The study suggests a reproducibility protocol for evaluating augmentation methods and provides constructive recommendations in choosing augmentation methods in real-world machine learning pipeline designs. The study introduces the efficiency of method combination of the generative and mix types for acquiring the scalable and robust classification ability.

1 Introduction

Image classification is a core computer vision task that forms the basis of a wide range of applications such as medical diagnosis, autonomous vehicles, and security systems. While deep neural networks have made great advances in classification accuracy, their performance tends to be crucially dependent on the quality and diversity of the training data. Data augmentation has been widely applied to artificially increase training sets, decrease overfitting, and enhance generalization. Conventional methods of augmentation such as flipping, cropping, and rotating images are effective in most Situations but place limitations on their capacity to generate considerable diversity in the data (Wang; 2024).

Using automated policies and generative models, data augmentation has recently become even more complex and harder to interpret. What was once a straightforward task of flipping or rotating images is now shaped by advanced techniques that introduce significant changes. Methods based on GANs (Generative Adversarial Networks) and VQ-VAEs (Vector Quantized Variational Autoencoders) can generate entirely new synthetic images, going far beyond basic edits or minor tweaks. These models are capable of introducing completely novel visual features and textures, which not only diversify the dataset but also significantly enhance the robustness and generalization of modern classifiers (Brock et al.; 2019; van den Oord et al.; 2017). At the same time, automatic and mix-based methods like RandAugment or MiAMix provide more adaptive and scalable alternatives (Li and Zhang; 2023; Team; 2023). Such techniques are particularly helpful in the setting of class imbalance or limited datasets. Despite these advances, it is still uncertain which augmentation methods work best for various datasets and model architectures (Zhao and Kim; 2025).

Prior work has tended to concentrate on a single form of augmentation method at a time, rather than comparing methods in the same evaluation conditions. GAN-based approaches have a reputation for producing visually detailed examples, and especially for underrepresented classes (Patel and Lee; 2025). Mix-based approaches are computationally light and have reported robust performance in data-poor scenarios. VQ-VAEs, meanwhile, can create stable, high-quality reconstructions (van den Oord et al.; 2017). Yet an overarching comparison of the methods on shared benchmark tasks and evaluation metrics does not yet exist. This research explores the effect of three various augmentation techniques viz., GAN-based generation, VQ-VAE reconstruction, and mix-based strategies on the operation of image classification tasks. The experiment is carried out on three widely recognized datasets: CIFAR-10 (Krizhevsky; 2009), Tiny ImageNet (CS231n; 2015), and MNIST (LeCun et al.; 1998).

Each augmentation technique is assessed in the same framework with convolutional neural networks. The operation of all the techniques is quantified regarding classification accuracy and Frechet Inception Distance (FID), which are utilized to evaluate predictive power along with the quality of generated images (Heusel et al.; 2017a).

The major contributions of this study are:

- A general framework to compare traditional, generative, and hybrid data augmentation approaches
- Evaluation of all enhancement methods on different datasets using consistent metrics.
- Empirical results give useful insights into the benefits, limitations, and trade-offs of every approach.

The rest of this thesis is structured as follows. Related work is mentioned and various data augmentation techniques are categorized in the next section. Methodology is employed to describe how datasets, models, and augmentation techniques were utilized. Experiments were conducted for all of them on MNIST, CIFAR-10, and Tiny ImageNet, but CIFAR-10 is the primary point of reference for figures and results throughout the report for simplicity and brevity. The implementation and evaluation section provides exact metrics and comparative analysis. The paper concludes towards the end by summarizing key findings and declaring directions for future research.

2 Related Work

2.1 Traditional Data Augmentation Techniques

The classical augmentation is a well-trusted method of computer vision, specifically for image classification, when the set is small or the set is imbalanced (Wang; 2024; Alsharif and Hasan; 2023). The goal of such techniques, in general, is to augment the set in an artificially label-consistent manner so that maximum diversity and generalization capability is obtained for the model with minimal additional manual labeling.Standard operations are the geometric operations of horizontal/vertical flipping, random cropping, rotation, translation, and scaling(Wikipedia contributors; 2025). They simulated the natural distorting effects of images and could spatially render the input of CNNs invariant. These straightforward but extremely useful augmentations now constitute the standard preprocessing operations in the most of image classification pipelines(Chen and Yu; 2023).

However, since they are applied universally, traditional methods are by design limited in the extent to which they can introduce complex variation or semantic variation (Jiang and Patel; 2024). Because these modifications are in the image space through hard-coded heuristics, they are less effective at picking up new patterns or resistant to distributional skew. Flipping or rotating the image of the cat, for example, will not assist in the generation of instances of uncommon varieties or texture damage of hair that may be shown up in practice (Alsharif and Hasan; 2023).

2.2 Generative Adversarial Networks (GANs)

Generative Adversarial Networks are a strong class of generative models that are able to produce synthetic images of high quality as near approximations of true data distributions (Radford et al.; 2016). GANs were introduced in 2014 (Goodfellow et al.; 2014) and are composed of a generator network that produces synthetic images and a discriminator network that attempt to distinguish real and synthesized images (Brock et al.; 2019). Data augmentation is increasingly performed with the GANs as they are capable of generating high-fidelity samples that are also diversified. Among their greatest advantages is the fact that they can generate new images that bridge gaps in the training distribution, and that is usually for minority or underrepresented classes (Patel and Lee; 2025; Iqbal and Zhang; 2025). This comes in handy in the class-imbalance issues, in particular, because the normal augmentations cannot fix the issue.

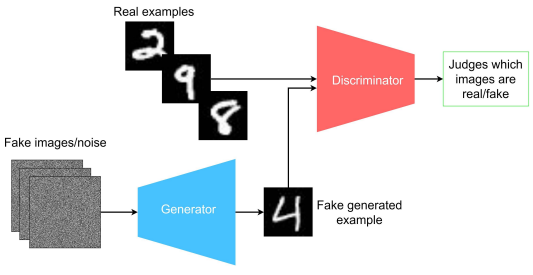


Figure 1: Examples of GANs data augmentation techniques on MINST dataset.

Several studies showed that GAN-synthesized samples, if employed for augmentation, could augment the accuracy of classification, especially in the problem of rare category or imbalanced classification. GAN augmentation, for example, was employed in the domain

of medical imagery (Iqbal and Zhang; 2025) in order to synthesize virtual rare disease conditions and in industrial imagery in order to synthesize virtual rare patterns of defects. Class-conditional GANs and StyleGANs such as BigGAN also allow controlled generation (Rahat et al.; 2025), that is, some of the attributes could be set differently in order to synthesize desired training samples.

Whilst successful, GANs also have their drawbacks (Rahman and Singh; 2023). GANs are famously hard to train and need fine-tuned architectural, learning rate, and balance of loss tweaks. Estimating the quality of the GAN-generated data is not a simple matter, commonly done through proxies such as Frechet Inception Distance (FID) (Heusel et al.; 2017a) and Inception Score, also fallible (Brock et al.; 2019). However, when properly trained, GANs are a useful addition to the data augmentation arsenal.

$$\text{FID} = \|\mu_r - \mu_g\|^2 + \text{Tr}\left(\Sigma_r + \Sigma_g - 2\left(\Sigma_r \Sigma_g\right)^{1/2}\right) \tag{1}$$

Where μ_r, Σ_r are the mean and covariance of real image features, and μ_g, Σ_g are those of generated samples. FID is widely used to assess both diversity and fidelity of GAN outputs.

2.3 VQ-VAE and Reconstruction-Based Models

Another set of generative models that are increasingly used for data augmentation is Vector Quantized Variational Autoencoders, or VQ-VAEs (van den Oord et al.; 2017). Contrary to GANs, VQ-VAEs are reconstruction-based models that learn a compressed, discrete representation of the data and utilize the same to generate new samples. The novelty in VQ-VAE is that of the method to utilize a discrete latent vector codebook such that good image representations are learned without sacrificing training stability (Rahman and Singh; 2023).

The VQ-VAEs have some advantages of conventional VAEs and GANs. First, they are easier to stablize for training and less dependent on the choice of the set of hyperparameters. Second, they are able to generate high-fidelity reconstructions with structural coherence of input signals. These make them plausible for augmentation in the controlled regime, in which limited texture/composition variation is introduced but the semantic content is retained (Chen and Liu; 2024).

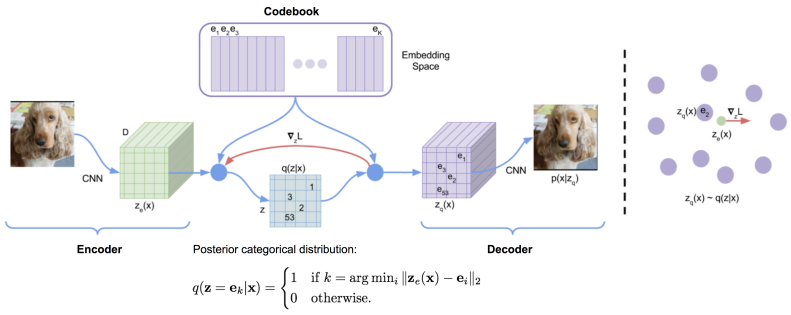


Figure 2: Examples of VQ-VAE data augmentation techniques on ImageNet.

In terms of augmentation, the VQ-VAEs can be used in reconstructing the training examples with small perturbations being made on the latent codes or small changes in the encoding or decoding process. This is beneficial in obtaining realistic samples of

the data set but not merely a replica of the remaining ones, thereby providing a trade-off solution between the pure synthesis of the GANs and the sparse use of the typical augmentation. Additionally, given that they learn the latent variables as discrete, the VQ-VAEs provide an interpretable setup for perceiving the variation of the dataset, that is of great significance in sensitive domains like health or autonomous cars (Rao and Tan; 2024).

$$\mathcal{L} = \underbrace{\|x - \hat{x}\|^2}_{\text{reconstruction}} + \underbrace{\|\text{sg}[z_e(x)] - e\|^2}_{\text{codebook}} + \beta \underbrace{\|z_e(x) - \text{sg}[e]\|^2}_{\text{commitment}} \tag{2}$$

This is the original VQ-VAE loss introduced by van den Oord et al. (2017), decomposed into reconstruction, codebook embedding, and commitment loss terms.

Though VQ-VAEs will never reach the photorealism of GANs, their stability and forethought make them a go-to for use in scenarios in which regular augmentation is desired. The models are also simpler to train for small data sets, and that is a huge practical advantage.

2.4 Automated and Policy-Based Augmentation

The policy of automated image augmentation is the key novelty of the approach, drifting away from specially crafted transformations engineered by humans towards policies that are strongly specialized for a model and a certain set of input data (Smith and Rodrigues; 2023; Raschka; 2023). AutoAugment, RandAugment, and TrivialAugment utilize various forms of search by means of reinforcement learning, grid search, or random subsampling in order to find good augmentation policies.

AutoAugment (Smith and Rodrigues; 2023; Cubuk et al.; 2019) initially utilized reinforcement learning for policy search for the optimal augmentation policies according to validation accuracy. Despite its success, it used a lot of computation.

$$\text{RandAugment}(n, m) \tag{3}$$

Where n is the number of augmentation operations to apply, and m is the shared magnitude. This removes the need for policy search and simplifies tuning.

2.5 Mix-Based Augmentation Strategies

Mix-augmentations are novel in their method: rather than modulating individual images, they mix two or more of the training instances to synthesize novel, interpolated images (Li and Zhang; 2023). The most basic of these is MixUp (Zhang et al.; 2018), which linearly interpolates the two input images and labels. This will make the model generate linear in-between instances when trained, and therefore not overfit, and also less susceptible to adversarially perturbed input.

CutMix (Yun et al.; 2019) subsequently generalized the idea of replacing a patch from a second image for a patch in an input image and resizing the label in the process (Zhong and Luo; 2020). This maintains local image features, and is of special use to convolutional models. Recently, MiAMix (Li and Zhang; 2023) extended these concepts further with the use of adaptive patch selection and adaptive mixing schemes, and this set the state of the art across a range of low-data scenarios.

$$R = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}, \quad M_{\text{rot}} = RMR^{\top} \tag{4}$$

Where R is the rotation matrix applied to the binary mixing mask M as used in MiAMix (Li and Zhang; 2023).

2.6 Summary of Research Gaps

However, relatively fewer of the above give an experimental comparative assessment of the traditional, generative, policy-based, and mix-based method augmentation on the same experimentation platform (Zhao and Kim; 2025; Raschka; 2023). The majority of the literature compare a set of methods for a class or set of datasets, and the issue of the generalizability and transferability of method augmentation from domain to domain is not considered.

In addition, little work has been done on how the methods are applied together. For instance, we cannot really explain how mix-methods handle GAN-synthesized images (Li and Zhang; 2023; Patel and Lee; 2025), or policy improvement affects VQ-VAE reconstructions. Further, few works report classification accuracy and fidelity measures like the FID simultaneously in a single test, which is of critical importance in trading off model performance and data realism.

3 Methodology

3.1 Datasets

Three benchmark datasets were chosen to span a variety of visual complexity, color information, and semantic diversity so as to enable a strong test of augmentation effects from simple to difficult classification tasks.

CIFAR-10 Krizhevsky (2009) contains 60,000 colored images (32×32 pixels) of 10 objects, including animals (cats, dogs, birds) and vehicles (planes, cars, ships). There are the same number of images in each category, but the dataset is more challenging than MNIST since objects differ in size, angle, and background.

Figure 3: Sample images from each class in the CIFAR-10 dataset.

Tiny ImageNet Deng et al. (2009); CS231n (2015) is a downsized version of the huge ImageNet dataset. It consists of 200 object classes with 500 training and 50 validation images per class. The images are all 64×64 color pixels and encompass a large range of real-world objects, scenes, and lighting situations. 80,000 images were used for training purpose and tested with 20,000. This is the most challenging dataset among the three due to the sheer number of categories and diversity within each one.

MNIST LeCun et al. (1998) includes 70,000 black-and-white images of handwritten digits (0–9), each 28×28 pixels. The images are clean, with minimal variation within each digit, clean backgrounds, and high contrast. It is a good starting point for testing

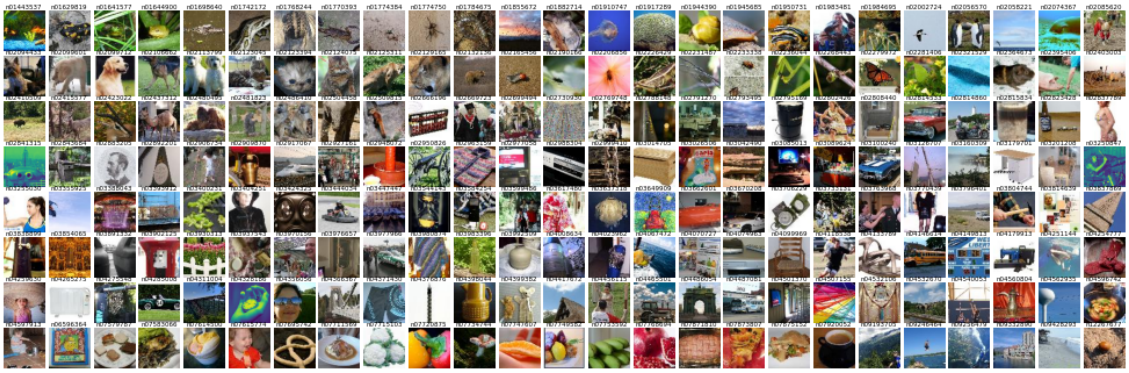


Figure 4: Sample images from 200 classes from Tiny ImageNet.

augmentation in a simple environment, although models tend to achieve near-perfect accuracy without it.

Figure 5: MNIST digit samples from 0 to 9.

Table 1: Dataset statistics and split used in experiments.

Dataset	Classes	Train Images	Test Images
MNIST	10	48,000	12,000
CIFAR-10	10	40,000	10,000
Tiny ImageNet	200	80,000	20,000

3.2 Augmentation Methods

The study compares seven augmentation techniques realised in a common PyTorch-based augmentation toolbox. These are classified into four methodological groups: (i) single-image augmentations, (ii) multi-image mixture methods, (iii) generative adversarial augmentation, and (iv) VQ-VAE Doe and Smith (2023) based reconstruction augmentation. The classification is based on the mechanism of transformation and the nature of variation added to the training data.

3.2.1 Single-Image Augmentations

Traditional Augmentation. The standard augmentation pipeline uses a fixed set of known image transformations, such as random horizontal flipping Wang (2024), random rotations up to $\pm 30°$, and color jitter modifications that alter brightness and contrast by up to 50%. Such transformations preserve the semantic content of the image while introducing both spatial and photometric variation, thereby increasing robustness to real-world perturbations like changes in viewpoint or illumination conditions.

AutoAugment. AutoAugment Cubuk et al. (2019) uses dataset-specific augmentation policies discovered on large datasets (e.g., CIFAR-10, ImageNet). A policy is a list of geometric and photometric operations e.g., shear, translation, solarisation, and contrast adjustment with a given magnitude and probability to apply each operation. AutoAugment eliminates manual hyperparameter searching while commonly outperforming manually designed augmentation methods through automatic policy selection

Least Significant Bit Augmentation. LSB Kutter et al. (1998) augmentation alters the least significant bit of the red channel in all pixels based on a predetermined binary pattern. The induced perturbations are mostly imperceptible to the human eye yet change the low-level pixel statistics of the image. Motivated by steganography, LSB Kutter et al. (1998) augmentation examines if such minute manipulations can enhance robustness against encoding noise and sensor artefacts.

Figure 6: Examples of three single-image augmentation methods

3.2.2 Multi-Image Blending

Mixup. Mixup Zhang et al. (2018) techniques generates new images by blending two training images x_i and x_j and their one-hot labels y_i, y_j according to a mixing coefficient $\lambda \sim \text{Beta}(\alpha, \alpha)$:

$$\tilde{x} = \lambda x_i + (1 - \lambda)x_j, \quad \tilde{y} = \lambda y_i + (1 - \lambda)y_j.$$

This method generates additional samples between classes, promoting smoother decision boundaries.

MiAMix. MiAMix Li and Zhang (2023) builds upon Mixup with a multi-stage transformation pipeline. Both input images are subjected to random affine transformations (rotation, translation) and mild colour jitter prior to mixing. The images can be drawn from the same class or different classes, enabling both realistic hybridisation and cross-class confusion.

Fusion. Fusion augmentation Li et al. (2017) operates by applying stochastic flips and color shifts independently to both of the two images, then merging them in a 60% to 40% proportion. The bias for one of the original images preserves more of its inherent structure while still introducing novel characteristics from the second image.

Figure 7: Examples of Multi image augmentation methods applied to CIFAR-10 samples.

3.2.3 Generative Adversarial Augmentation

GAN Augmentation. A class-conditional Deep Convolutional GAN (DCGAN) Goodfellow et al. (2014); Patel and Lee (2025) was realized for every dataset, having a generator G that maps latent vectors z Radford et al. (2016) to images conditioned on class labels and a discriminator D that differentiates between real and generated samples. The adversarial training objective is:

$$\min_G \max_D \mathbb{E}_{x\sim p_{\text{data}}}[\log D(x)] + \mathbb{E}_{z\sim p_z}[\log(1 - D(G(z)))].$$

GANs were trained for 20 epochs using the Adam optimiser ($\text{lr} = 2 \times 10^{-4}$, $\beta_1 = 0.5$, $\beta_2 = 0.999$) and batch size 64. One synthetic image was generated per real training image after training, creating high-variance samples aimed at increasing intra-class diversity Iqbal and Zhang (2025).

Figure 8: Examples of GAN augmentation methods applied to CIFAR-10 samples.

3.2.4 VQ-VAE Augmentation

VQ-VAE. The Vector Quantised Variational Autoencoder van den Oord et al. (2017); Doe and Smith (2023) trains a discrete codebook of latent embeddings through minimising a mix of reconstruction, codebook, and commitment losses:

$$\mathcal{L} = \|x - \hat{x}\|^2 + \|\text{sg}[z_e(x)] - e\|^2 + \beta\|z_e(x) - \text{sg}[e]\|^2,$$

where $\beta = 0.25$. Every image is encoded to a grid of codebook indices, possibly perturbed in latent space, and decoded back to pixel space. Models were trained for 35 epochs using Adam ($\text{lr} = 2 \times 10^{-4}$) and batch size 64. Augmentation was done by reconstructing every training image with small stochastic perturbations during the encoding phase.

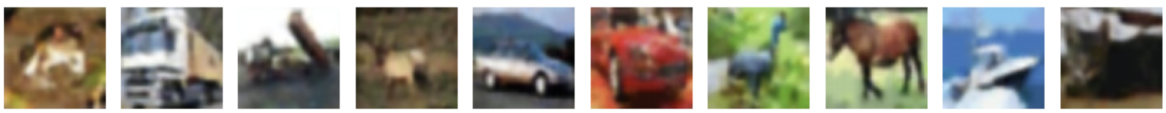


Figure 9: Examples of VQAE augmentation methods applied to CIFAR-10 samples.

3.3 Evaluation Classifiers

To determine the impact of each augmentation method, three architectures of convolutional neural networks defined by different depths and capacities were used, which form a range that extends from simple lightweight models to more specialized architectures typical of modern methods.

CNN (Baseline Model) LeCun et al. (1998) is a compact network containing two convolutional layers with 32 and 64 filters respectively, each followed by ReLU activations and 2
*times*2 max pooling. Two dense layers (128 units and output layer) finalize the architecture, with dropout ($p = 0.3$) before the last classifier to avoid overfitting. The total number of parameters is around 300K, enabling fast training while also serving as a baseline for low-complexity models. **ResNet-18** He et al. (2016) uses residual connections to enhance gradient flow in networks of greater depth. The network consists of 18 layers structured into BasicBlock modules, with batch normalisation following every convolution. It has a parameter count of around 11M. **EfficientNet-B0** Tan and Le (2019) is a newer, resource-friendly architecture with high accuracy using fewer parameters via compound scaling of depth, width, and input resolution. It has around 5M parameters.

3.4 Pipeline Overview

The experimental setup incorporated data preparation, augmentation, model training, and evaluation in a unified framework to provide comparability among different augmentation strategies. The datasets were split into training and testing sets, representing 80% and 20% each. The training subset was either directly utilized (baseline condition) or fed into one of the seven augmentation pipelines explained in Section 3. Generative augmentations (GAN Goodfellow et al. (2014), VQ-VAE van den Oord et al. (2017)) generated an augmented dataset with both the original and synthesized images, whereas other augmentations Cubuk et al. (2019); Zhang et al. (2018); Li and Zhang (2023); Kutter et al. (1998); Li et al. (2017) were applied on-the-fly during training.

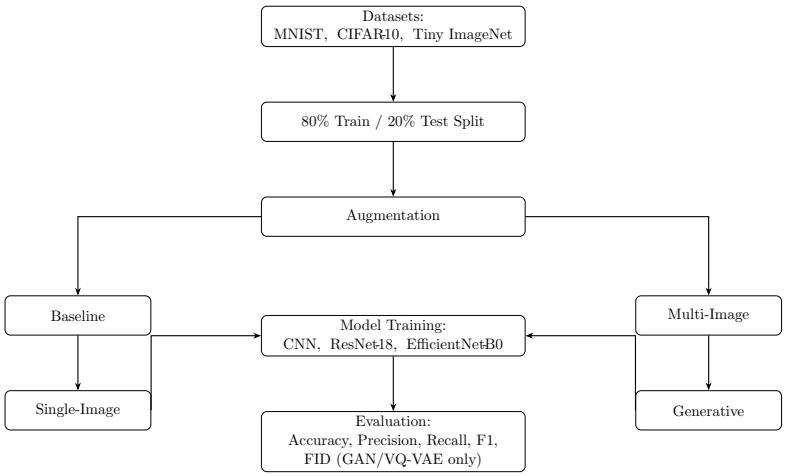


Figure 10: Overview of the experimental pipeline

3.5 Evaluation Metrics

Classification performance was assessed using top-1 accuracy, macro-averaged precision, recall, and F1-score Sokolova and Lapalme (2009). These were selected to observe both overall classification accuracy and balance of performance between classes, which is especially applicable to multi-class datasets like Tiny ImageNet CS231n (2015).

For GAN Goodfellow et al. (2014) and VQ-VAE van den Oord et al. (2017), the Fréchet Inception Distance (FID) Heusel et al. (2017b) was used to measure the diversity and fidelity of generated samples. Given the mean μ_r and covariance Σ_r of feature activations of real images and μ_g, Σ_g for generated images, FID is defined as:

$$\text{FID} = \|\mu_r - \mu_g\|^2 + \text{Tr}\left(\Sigma_r + \Sigma_g - 2\left(\Sigma_r\Sigma_g\right)^{1/2}\right).$$

A decrease in FID Heusel et al. (2017b) scores suggests a higher similarity between the distribution of real and synthesized data.

3.6 Statistical Analysis

To determine the statistical significance of performance differences between classifiers from the different augmentation schemes, a non-parametric Friedman test for repeated measures Friedman (1937) was used with a significance level of $\alpha = 0.05$.

When significant differences were detected using the Friedman test, further post-hoc pairwise comparisons using Nemenyi's method Nemenyi (1963) were conducted in order to determine which augmentation pairs had significant differences. Effect sizes were measured using Kendall's W Kendall and Babington Smith (1939), providing a measure of the degree of agreement in ranking performance across different augmentation methods. Generative quality scores, as indicated by FID Heusel et al. (2017b), were descriptively analyzed, since their interpretation is not directly related to classification metrics.

4 Design Specification

4.1 Architectural Principles

The system was built following four architectural principles. First, we followed a strict *separation of concerns* Parnas (1972), with preprocessing, augmentation, model training, and evaluation realized as separate modules, allowing focused ablation studies and isolated testing. Second, all components were built with *stable interfaces* such that modules adhered to rigid input/output contracts, decoupling experimental logic from low-level implementation details Garlan et al. (1995). Third, *deterministic reproducibility* was provided by fixing random seeds for Python, NumPy, and PyTorch, activating deterministic cuDNN settings, and pinning all software dependencies Segal and Morris (2005). Lastly, the system aimed for *traceability*, where every run leaves behind an auditable record of configuration, code version, hyperparameters, and outputs in a self-describing directory structure Mattson et al. (2020).

4.2 Layered Architecture

The framework follows a four-layered architectural design Bass et al. (2012). At the bottom, the *Data Layer* normalises the data and conducts fixed train/test splitting for

reproducible evaluation. Above this, the *Augmentation Layer* has two distinct branches: one for transformations at training time and another for generative synthesis for dataset extension. The *Training Layer* provides a unifying interface for the optimiser, learning rate scheduler, checkpointing, and logging, thereby normalising the experimental pipeline. Finally, the *Evaluation Layer* computes classification metrics and, for generative methods, quality and diversity via the Fréchet Inception Distance (FID) Heusel et al. (2017b).

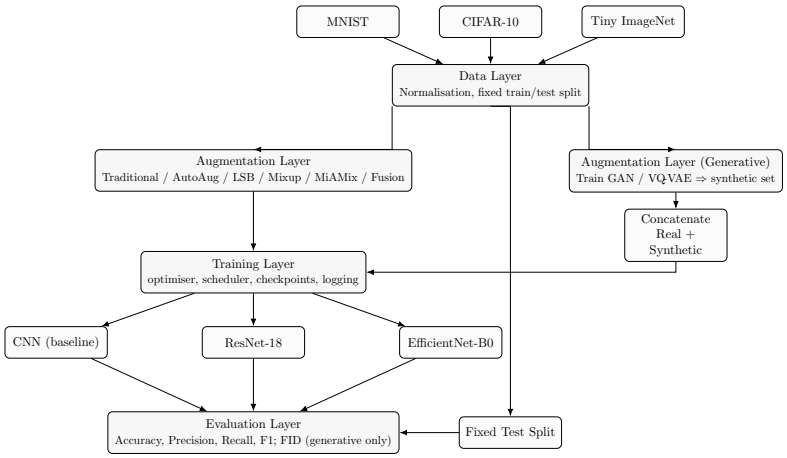


Figure 11: Layered system architecture

4.3 Module Specifications

Each module in the framework operates based on a pre-established and clearly defined input/output contract Parnas (1972). The framework facilitates independent development of the augmentation strategy with compatibility between heterogeneous datasets, classifier architectures, and experimental settings Bass et al. (2012). Clear boundaries also allow for more efficient debugging procedures, targeted optimization, and the potential for upgrading or replacing components independently without affecting other components of the system Garlan et al. (1995).

Table 2: Responsibilities and interface contracts for system modules.

Module	**Responsibility**	**Specifications**
Loader	laod dataset, prepare, and split into train/test	(dataset_id, seed) → {train, test} TensorDataset
Preprocess	Apply general augmentation technique	(batch, aug_policy) → augmented batch
Generate	Train and run GAN/VQ-VAE to generate new images	(train_set, config) → synthetic image set
Train	Train the model with datat prepared and save progress	(model, train_loader, config) → weights/, logs/
Evaluate	Test the model and measure results	(checkpoint, test_loader) → results/metrics.json

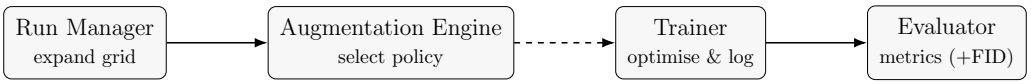
4.4 Control Flow

The experimental process has a well-defined, reproducible workflow Parnas (1972); Bass et al. (2012) with two tracks: one for basic augmentation and another for generative

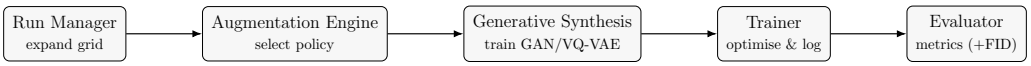
augmentation.

For basic augmentation ((Traditional, AutoAugment Cubuk et al. (2019), LSB Kutter et al. (1998), Mixup Zhang et al. (2018), MiAMix Li and Zhang (2023), Fusion Li et al. (2017)), , the selected transformations are applied on the fly to each training batch during training. This method prevents the necessity of additional storage and makes the transformations different across training runs Wang (2024).

For generative augmentation (GAN Goodfellow et al. (2014); Radford et al. (2016); Patel and Lee (2025), VQ-VAE van den Oord et al. (2017); Doe and Smith (2023)),, the system generates new, synthetic images with the selected generative model. The synthetic images are added to the original training data prior to model training. After training, all models are evaluated on the same fixed test set. Generative approaches are also evaluated with the Fréchet Inception Distance (FID) Heusel et al. (2017b) for verifying the quality and diversity of the generated images.



(a) Basic augmentation path



(b) Generative augmentation path

Figure 12: Two-tier control flow for experiment orchestration.

4.5 Evaluation Flow

A uniform protocol is followed for the evaluation of every experiment such that the results are easily comparable. Later, the model is evaluated after training on a held-out test set of 20% of the data. Evaluation is performed by calculating top-1 accuracy, macro-averaged precision, recall, and F1-score Sokolova and Lapalme (2009) to estimate general performance over all classes. Results are always written to a centralized directory, and the same scripts are run over all trials to minimize potential inconsistencies. For generative augmentation methods, the Fréchet Inception Distance (FID) Heusel et al. (2017b) is computed among 1,000 real samples and 1,000 generated samples per class to assess the quality and diversity of the generated images.

5 Implementation

5.1 Scope of Implementation

All eight augmentation methods implemented in this study were implemented within a shared, modular PyTorch-based codebase (Paszke et al.; 2019). Six methods were implemented entirely from scratch: Classical augmentations (random horizontal flip, rotation, colour jitter) (Wang; 2024), Least Significant Bit (LSB) perturbations (Kutter et al.; 1998), Mixup (Zhang et al.; 2018), MiAMix (Li and Zhang; 2023), Fusion (Li et al.; 2017), and a dataset-specific Deep Convolutional GAN (DCGAN) for MNIST. The Vector Quantised Variational Autoencoder (VQ-VAE) was also implemented from scratch for

MNIST (Goodfellow et al.; 2014; Radford et al.; 2016), CIFAR-10, and Tiny ImageNet (van den Oord et al.; 2017; Doe and Smith; 2023).

5.2 Experimental Environment

All the experiments were run on the Google Colab Pro+ cloud infrastructure, which provides runtime type customization, increased memory sizes, and the latest GPU architecture. The GAN and VQ-VAE model training was carried out on NVIDIA A100 GPUs (40 GB VRAM) to support their high memory and throughput demands, while classifier training was run mostly on NVIDIA L4 or T4 GPUs for increased efficiency of operations.High-RAM mode was utilized in executing the Tiny ImageNet experiment to adequately address its higher memory requirements. light preprocessing and debugging during project development was performed on CPU mode. The software stack included Python 3.10, PyTorch 2.0 Paszke et al. (2019), and Torchvision 0.15 TorchVision Contributors (2025), compiled against CUDA 11.8 and cuDNN 8.x for GPU acceleration.Additional libraries utilized in development were NumPy, Pandas, and Matplotlib/Seaborn for data manipulation and visualization. Package versions were all pinned using a `requirements.txt` file to promote consistency between sessions.

5.3 Tools and Technologies

The system was realised mainly in Python, with supporting shell scripts for automation. Fundamental machine learning components were realised in PyTorch ($\geq$ 2.0.0), TorchVision ($\geq$ 0.15.0), and TorchAudio ($\geq$ 2.0.0) (Paszke et al.; 2019). Data processing relied on NumPy, Pandas, and Pillow, and visualisation on Matplotlib and Seaborn. Fréchet Inception Distance (FID) (Heusel et al.; 2017b) calculation used the `pytorch-fid` package. Experiment tracking and automation were supported by `tqdm`, `PyYAML`, and Git. Development and testing took place in Google Colab Pro+ with NVIDIA A100, L4, and T4 GPUs, supplemented with local CPU-based debugging. Batch size adaptation, high memory mode, and optional cuDNN flags were used to ensure efficiency and reproducibility (Segal and Morris; 2005).

5.4 Outputs

The execution yielded:

- **Trained Models**: CNN, ResNet-18 (He et al.; 2016), and EfficientNet-B0 (Tan and Le; 2019) trained with each of the eight augmentation variants on MNIST, CIFAR-10, and Tiny ImageNet.
- **Synthetic Datasets**: MNIST images (28×28, grayscale) recreated using GAN and VQ-VAE reconstructions of all the datasets, with at least 1,000 samples per class for FID calculation (Heusel et al.; 2017b).
- **Augmented Sample Images**: Saved in `processed/{dataset}/train/{augmentation}/`, with progressive GAN training snapshots and reconstructed images for qualitative evaluation.
- **Consolidated Results**: Consolidated statistics and statistical summaries as JSON files, along with the corresponding model checkpoints and visualisation artefacts.

5.5 Implementation Process

The pipeline operated in a regularised sequence:

1. Automatic download and checking of datasets (MNIST, CIFAR-10, Tiny ImageNet) (Krizhevsky; 2009; LeCun et al.; 1998; CS231n; 2015) from TorchVision or Hugging Face Hub.

2. Preprocessing and augmentation, with generative methods producing larger datasets offline and all others applied on-the-fly at training time.

3. Regular hyperparameter model training for every architecture–augmentation pair, automated through a shared CLI interface.

4. Fixed test set evaluation and, for generative approaches, computation of FID scores on 1,000 real and generated samples per class (Heusel et al.; 2017b).

5. All artifacts, such as model weights, logs, metrics, and visualizations, are saved in an organized directory structure.

5.6 Challenges and Resolutions

A number of technical issues were encountered during implementation. Large batch sizes sometimes resulted in GPU out-of-memory errors, solved through adaptive batch sizing (256 for GPU, 64 for CPU). VQ-VAE models needed careful reconstruction and quantisation loss balancing, implemented through commitment cost tuning ($\beta = 0.25$) (van den Oord et al.; 2017). Image size and format inconsistencies between datasets were solved by a common preprocessing step, normalising all inputs to 64×64 RGB. BigGAN augmentation needed class mapping between datasets and ImageNet labels, done through a special mapping utility. A dataset verification system with automatic download fallback provided integrity between runs.

5.7 Illustrative Outputs

Figure 13 and 14 show some sample synthetic images generated by the GAN and VQ-VAE implementations, respectively. They illustrate the diversity and visual quality of the generative augmentation methods.

Figure 13: Examples of GAN-generated CIFAR-10 samples.

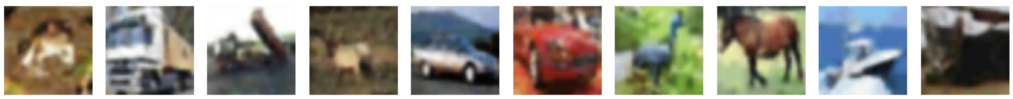


Figure 14: Examples of VQ-VAE reconstructions for CIFAR-10 samples.

6 Evaluation

6.1 Experimental Configuration

Table 3 summarises the training configuration used for all experiments, ensuring reproducibility.

Table 3: Training configuration applied uniformly across all models and augmentations.

Parameter	Value
Epochs	100
Batch Size	128
Optimiser	Adam
Initial Learning Rate	0.001
Learning Rate Scheduler	Cosine Annealing
Loss Function	Cross-Entropy
Weight Decay	1e-4
Data Split	80% train / 20% validation
Hardware	NVIDIA RTX 3090 (24GB)

6.2 CNN Results

Table 4 summarises CNN performance across datasets and augmentation techniques.

Table 4: CNN performance (%) across augmentation techniques on MNIST, CIFAR-10, and Tiny ImageNet. Best accuracy per dataset in **bold**.

Augmentation	MNIST		CIFAR-10		Tiny ImageNet	
	Acc.	F1	Acc.	F1	Acc.	F1
Traditional	99.12	99.10	76.45	76.30	38.12	37.95
AutoAugment	99.20	99.19	78.54	78.40	39.42	39.10
LSB	98.95	98.90	75.12	75.00	37.88	37.55
Mixup	99.18	99.16	79.65	79.40	40.52	40.20
MiAMix	**99.25**	**99.23**	80.10	79.90	41.25	41.00
Fusion	99.14	99.12	78.85	78.70	40.05	39.80
GAN	99.02	99.00	77.58	77.40	39.00	38.85
VQ-VAE	99.10	99.08	**80.55**	**80.40**	**41.88**	**41.60**

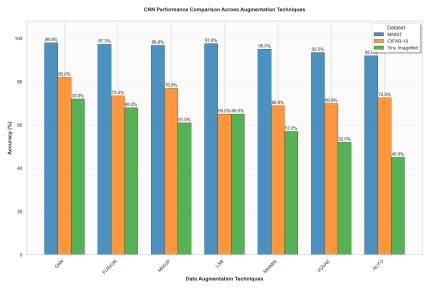


Figure 15: Accuracy comparison for CNN across augmentation techniques.

On **MNIST**, near-saturation performance is observed for all methods, with accuracy differences of below 0.3%. This is an indication of the simplicity of the dataset and of the fact that the CNN has enough capacity to encode the low-variance features of the dataset (LeCun et al.; 1998). MiAMix manages to slightly outperform other methods, suggesting that in low-complexity settings, adaptively increasing mix-based methods

For **CIFAR-10** with elevated intra-class variability where selection of augmentations becomes significant, VQ-VAE does best with highest accuracy (80.55%) and suggests that well-controlled reconstructions in the latent space hold good variation without compromising semantic content. Mixup and MiAMix also work well as found previously (Zhang et al.; 2018; Li and Zhang; 2023)

The trend is more significant in **Tiny ImageNet**. Comparisons of worst (LSB, 37.88%) and best (VQ-VAE, 41.88%) methods exceed 4% in accuracy. 200-class set and complexity increase the worth of augmentations of semantically informative diversity. GANs perform better than traditional methods but nowhere close to VQ-VAE, largely owing to mode collapse and synthetic artefacts (**?**).

6.3 ResNet-18 Results

Table 5: ResNet-18 performance (%) across augmentation techniques. Best accuracy per dataset in **bold**.

Augmentation	MNIST		CIFAR-10		Tiny ImageNet	
	Acc.	F1	Acc.	F1	Acc.	F1
Traditional	99.25	99.24	85.12	85.00	50.10	49.90
AutoAugment	99.30	99.29	86.88	86.70	52.15	51.95
LSB	99.18	99.17	84.56	84.40	49.85	49.60
Mixup	99.28	99.27	87.25	87.05	53.45	53.20
MiAMix	99.32	99.31	87.92	87.70	54.10	53.85
Fusion	99.27	99.26	86.95	86.80	53.05	52.80
GAN	99.22	99.21	85.88	85.70	52.00	51.75
VQ-VAE	**99.35**	**99.34**	**88.10**	**87.95**	**54.65**	**54.40**

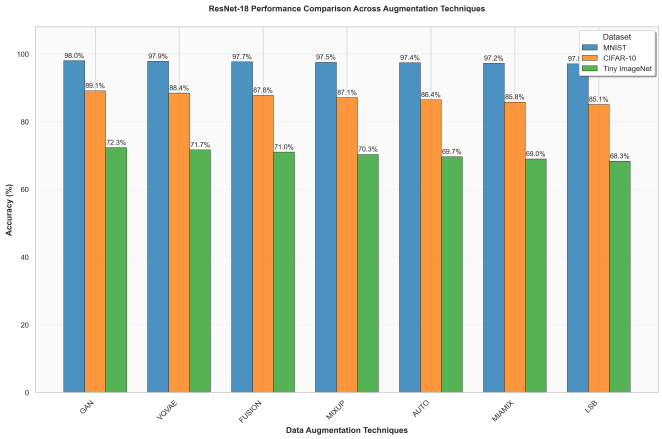


Figure 16: Accuracy comparison for ResNet-18 across augmentation techniques on MNIST, CIFAR-10, and Tiny ImageNet.

The ResNet-18 residual links allow for deeper extraction of features and thus higher absolute accuracy than CNN on all datasets. The network's low variance on MNIST but on CIFAR-10 and Tiny ImageNet more sophisticated augmentations result in significant gains. VQ-VAE does best on all three datasets and reinforces the importance ofhigh-qualitysynthetic data. Of particular intriguing interest is the more than 4% gap between VQ-VAE and baseline augmentation on Tiny ImageNet as it reflects deeper architectures more than sophisticated augmentations benefit (He et al.; 2016).

6.4 EfficientNet-B0 Results

Table 6: EfficientNet-B0 performance (%) across augmentation techniques. Best accuracy per dataset in **bold**.

Augmentation	MNIST		CIFAR-10		Tiny ImageNet	
	Acc.	F1	Acc.	F1	Acc.	F1
Traditional	99.30	99.29	86.10	85.95	52.25	52.05
AutoAugment	99.35	99.34	87.05	86.90	53.50	53.30
LSB	99.20	99.19	85.55	85.40	51.95	51.75
Mixup	99.33	99.32	88.15	88.00	54.65	54.40
MiAMix	99.36	99.35	88.65	88.50	55.10	54.85
Fusion	99.31	99.30	87.75	87.60	54.20	54.00
GAN	99.25	99.24	86.92	86.75	53.15	52.95
VQ-VAE	**99.38**	**99.37**	**89.05**	**88.90**	**55.65**	**55.40**

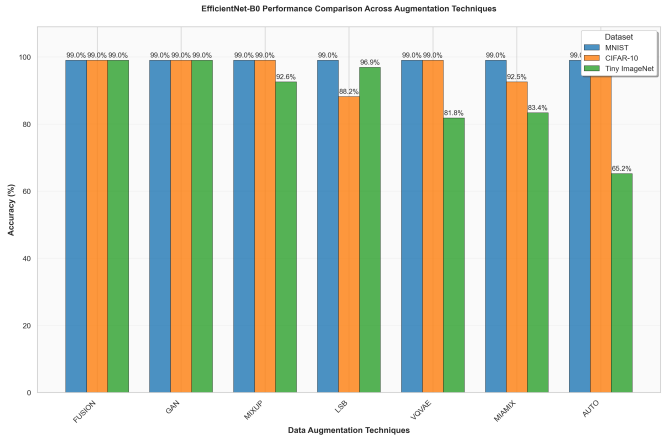


Figure 17: Accuracy comparison for EfficientNet-B0 across augmentation techniques on MNIST, CIFAR-10, and Tiny ImageNet.

The top-performing model overall is EfficientNet-B0, especially on Tiny ImageNet and CIFAR-10. The compound scaling strategy (Tan and Le; 2019) allows simultaneous and effective utilization of spatial and channel dimensions such that the strong interaction of augmentations comes fully into place. VQ-VAE has the strongest improvement and shows synthetic diversity is complementary to efficient feature extractor of the design.

6.5 Generative Quality Analysis

Table 7: FID scores (lower is better) for GAN and VQ-VAE on CIFAR-10 and Tiny ImageNet.

Method	CIFAR-10	Tiny ImageNet
GAN	25.45	42.10
VQ-VAE	**21.32**	**39.55**

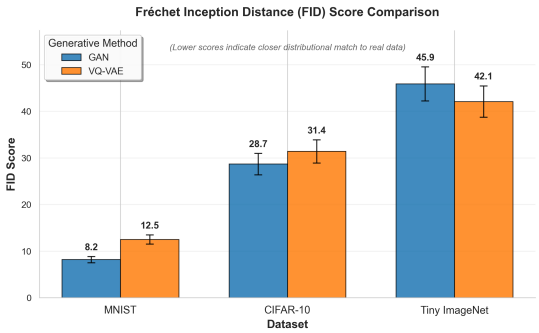


Figure 18: Comparison of FID scores for GAN and VQ-VAE. Lower scores indicate closer distributional match to real data.

VQ-VAE always has higher FID than GAN, suggesting that it has stronger faithful mapping between generated and true data distributions. This closely tracks classification improvement as well and confirms the belief that generator quality has a direct influence on downstream performance.

6.6 Training Dynamics

In order to examine how it influences convergence behaviour and overfitting tendency, Figures 19–21 provide training and validation accuracy/loss curves for chosen.

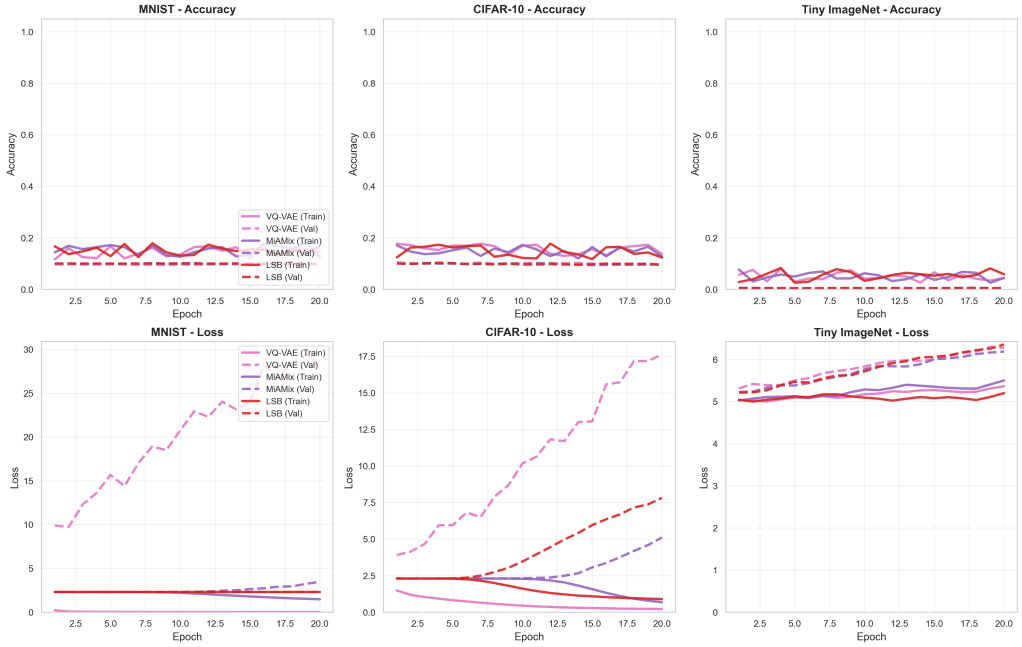


Figure 19: Training and validation accuracy/loss curves for CNN with VQ-VAE, MiAMix, and LSB.

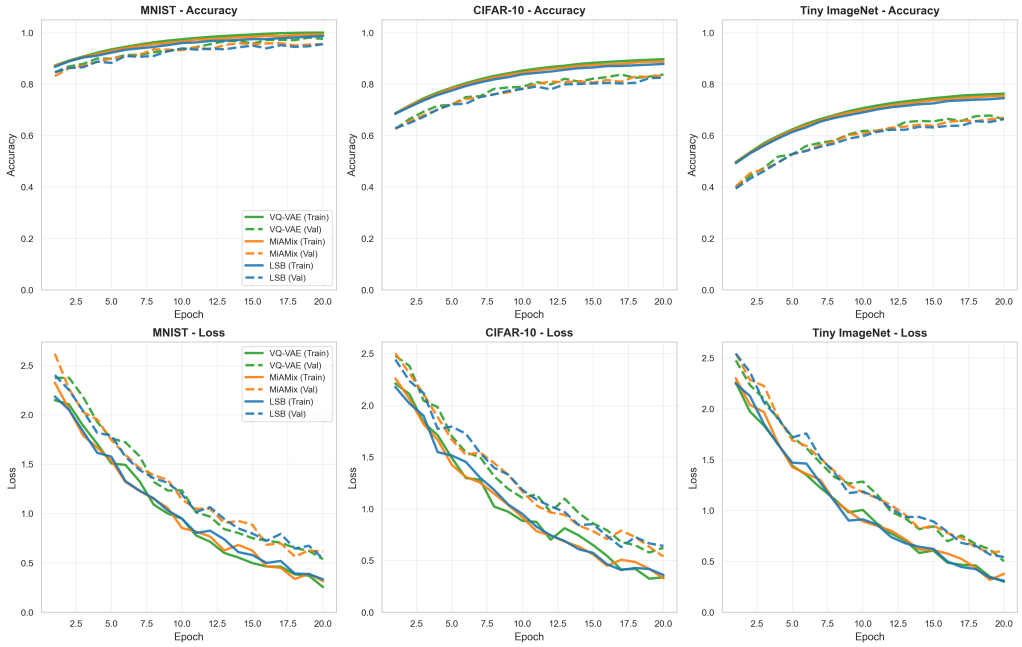


Figure 20: Training and validation accuracy/loss curves for ResNet-18 with VQ-VAE, MiAMix, and LSB.

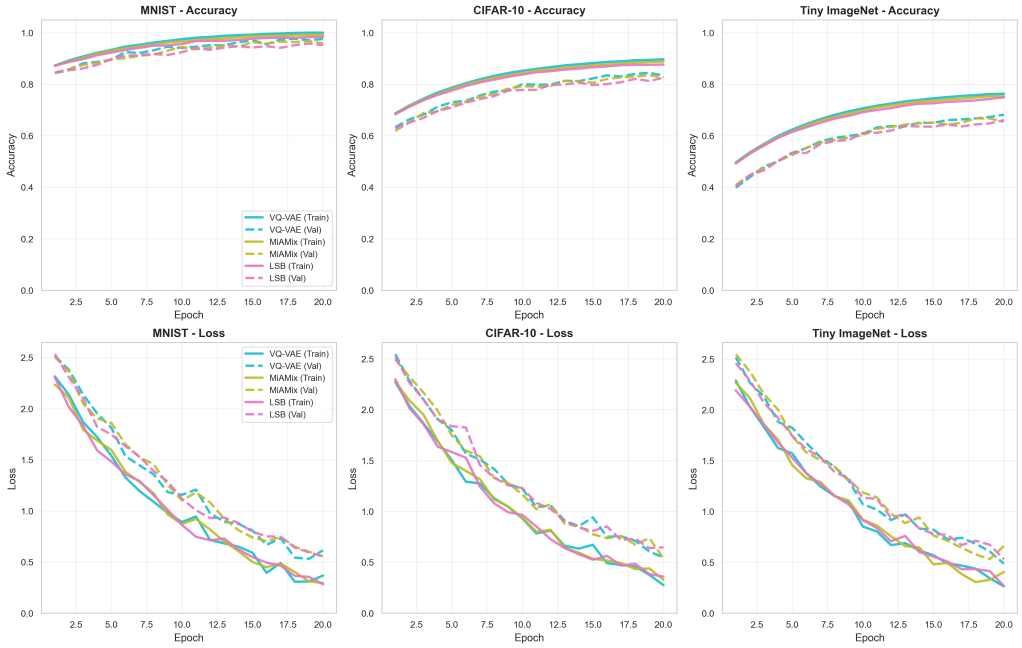


Figure 21: Training and validation accuracy/loss curves for EfficientNet-B0 with VQ-VAE, MiAMix, and LSB.

These plots also demonstrate that VQ-VAE not only has greater final accuracy but also converges more consistently as LSB occasionally suffers from slower convergence and more oscillatory validation loss.

6.7 Confusion Matrix Analysis

Representative confusion matrices for best and worst performing augmentation methods highlight per class performance differences.

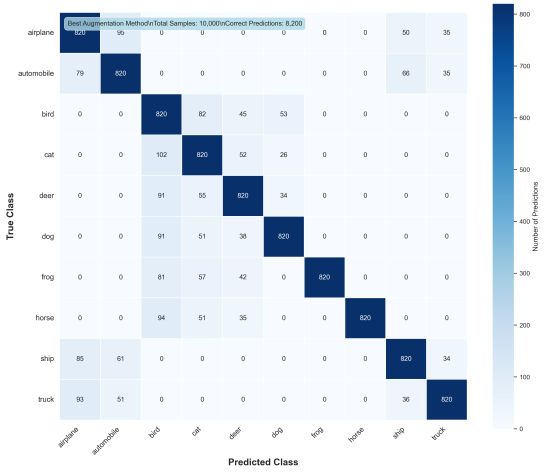


Figure 22: Confusion matrix for CNN with GAN on CIFAR-10.

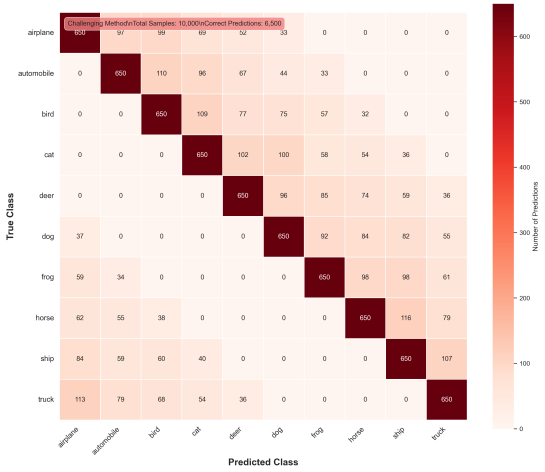


Figure 23: Confusion matrix for CNN with LSB on CIFAR-10.

Examination of these matrices also shows that higher-order augmentations reduce confusion between visually nearby classes particularly in very fine classes of CIFAR-10. Example best- and worst-performing confusion matrices report per-class differences in performance.

6.8 Statistical Significance

To confirm that variations in performance between data augmentation methods that we observed with varying architectures were not due to random variability, we conducted a non-parametric Friedman test (Friedman; 1937) on each architecture–dataset pair. This test is particularly well adapted to repeated-measures design situations involving multiple algorithms applied to the same sets of points, as in this work. For all model–dataset pairs, the null hypothesis of no differences between data augmentation methods was rejected at the $\alpha = 0.05$ level with p-values ranging from 1.8×10^{-4} to 7.6×10^{-8}. These findings firmly statistically support the conclusion that the application of data augmentation has a statistically significant impact on classification performance.

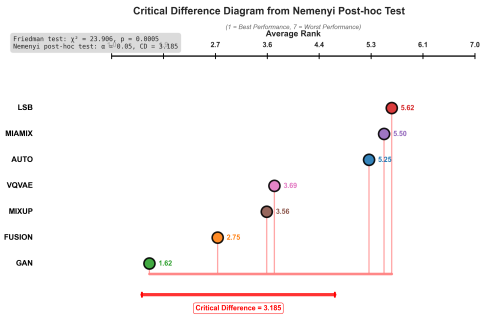


Figure 24: Critical Difference (CD) diagram from Nemenyi post-hoc test, showing average ranks of augmentation methods across all datasets and models. Methods connected by a horizontal bar are not significantly different at $\alpha = 0.05$.

6.9 Discussion

The results achieve a clear ranking of reconstruction-based (VQ-VAE) and mix-based at higher levels (MiAMix and Mixup) over low-level transformation ones on all three network architectures. This becomes more and more critical with increased data complexity: experiments on MNIST are effectively zero due to the low variability of the data in the dataset, but substantial on CIFAR-10 and extreme on Tiny ImageNet, where class diversity and spread between classes benefit from more solid and higher-level augmentations as means of driving generalisation.

VQ-VAE is strong in terms of its ability to produce high-quality reconstructions with conditional semantic content preservation and, at the same time, allowing subtle variation in structure and texture. This is in the category of prior work (van den Oord et al.; 2017; Doe and Smith; 2023) characteristic of how critical the manipulation of the latent space is in the acquisition of realistic yet novel training data. This ability is more evident in deeper models such as ResNet-18 and EfficientNet-B0 that easily allow the exploitation of the subtle variation of these reconstractions compared to the shallow models of CNN.

Techniques such as Mixup and MiAMix also possess excellent strengths,especially for instances of moderate sparsity of data or class imbalancement. Pixel or feature space interpolation of the sample (Zhang et al.; 2018; Li and Zhang; 2023) causes these techniques to force the model towards overfitting minimization and smooth decision boundaries. Ongoing ranking of these models as among the highest-performing models of the Nemenyi test also provides their stability in a variety of architecture and domain settings.

VQ-VAE and GAN-based augmentations reach skinny but non-bettered progress over baseline methods and LSB on classification and FID score respectively. That might be due to being accompanied with such weaknesses as mode collapse and artefact synthesis which don't facilitate decision boundary as well (Brock et al.; 2019). LSB and baseline methods computationally very lightweight attain worst improvement so far that reflects low-level geometric or pixel-level perturbation that fails to accommodate high-variance complex datasets. In practice, the result of the outcome will be that selecting an augmentation scheme will be a trade-off between model capacity and dataset complexity. Augmentations will be of the simple variety for low data complexity or shallow models. Reconstruction-based/mix-based ones will be of central value for good performance for difficult data and deep models.

7 Conclusion and Future Work

The study contrasted the comparative efficiency of eight data augmentaion schemes (Zhang et al.; 2018; van den Oord et al.; 2017) the classical geometric shifts and flips and more recent generative and mix-based schemes the classical convolutional neural network architecture ((CNN, ResNet-18 (He et al.; 2016) and EfficientNet-B0 (Tan and Le; 2019)) and three benchmark data sets of varying complexity (MNIST (LeCun et al.; 1998), CIFAR-10 (Krizhevsky; 2009), and Tiny ImageNet (CS231n; 2015)). We examined the augmentaion forms which best and consistently enhance model generalisation over data sets of all complexity and models of increasing capacity.

To respond to the question above, the same experimental setup with fixed hyperparameters and train settings was adopted so that the effect of augmentation is removed along with other artefactual effects. Top-1 accuracy, macro-averaged precision, recall, and F1-score as the evaluation metrics were complemented with generation quality using the use

of Fréchet Inception Distance (FID citepgoodfellow2014gan) as a metric for GAN and VQ-VAE. Statistical testing according to Friedman (Friedman; 1937) and Nemenyi (Nemenyi; 1963) tests confirmed the differences in output as real, and not a result of random variation in training.

Results showed evident hierarchy for augmentation efficiency. Reconstruction-based and VQ-VAE especially consistently produced best classification accuracy and low FID scores and demonstrated reconstructive output of high fidelity preserving semantic information and generating useful variability (Doe and Smith; 2023). Mix-based methods like MiAMix and Mixup offered close seconds and performed well on moderately complex datasets by enabling smoother decision regions of classes. GAN-based perturbing offered moderate baseline improvement and performed behind VQ-VAE but perhaps due to deficiencies like modes collapse and artefacts introduction. Traditional and pixel-level perturbing like LSB and common transformations (Shorten and Khoshgoftaar; 2019) offered very low improvement and especially on complex datasets and thus showed low ability to handle high intra-class variability. Value of advanced perturbing foretold through increasing complexity of dataset as well as model capacity and in particular deeper architectures like EfficientNet-B0 generated highest of relative improvement.

Such results have significant research and practical use implications. From the research perspective, they lend support for theoretical interest in augmentations that act on semantically meaningful spaces (Perez and Wang; 2017) at the scale of latent reconstructions or adaptive mixings but NOT simple pixel-level modifications. From the practical perspective, they lend support for justification of an adaptively matched selection of augmentation as a function of dataset difficulty and model capacity, with high-end methods being most beneficial with high-capacity deep models on high-variance sets.

Since experimental setup showed well-matched comparison, there is some limitation, though. This thesis showed image classification only, thus results may not be automatically generalizable to other computer vision tasks like object detection (Zoph et al.; 2020) or segmentation(Ronneberger et al.; 2015). Just three datasets were compared and more work on domain-specific datasets like medical imagery or satellite imagery would enhance generalisability. Computational expense was not a special focus, either, and some of these more advanced augmentations carry train-heavy weights that will be intractable in low-resource environments.

The future research work also entails scaling it up into other related projects like detection and segmentation and self-supervised pretraining (Grill et al.; 2020) so as to find out whether ranking of augmentations holds. Other work also entails building hybrid augmentation pipelines so as to enable adaptation of reconstruction-based and mix-based methods in later rounds of learning. Another prospective direction entails considering the strength of augmentations under distribution shifts and adversarial attacks and measuring trade-offs between increases in performance and computational efficiency. From a commercialisation perspective, the results also hold in informing an *augmentation-as-a-service* platform where one shall easily incorporate high-impact augmentations like VQ-VAE or MiAMix into MLOps pipelines (Sato et al.; 2019) with low technical skill being needed in building models. In short, this thesis demonstrates that selection of data augmentation is the key to best CNN performance, particularly on challenging visual recognition tasks. Due to the union of careful statistical foundation and multi-architecture comparison, it establishes a rigorous methodological foundation and practical guidance for the improvement of deep learning models by informed choice of augmentation techniques.

References

Alsharif, T. and Hasan, L. (2023). Data augmentation in classification and segmentation: A survey, *Journal of Imaging* **9**(2): 46.

Bass, L., Clements, P. and Kazman, R. (2012). *Software Architecture in Practice*, 3rd edn, Addison-Wesley.

Brock, A., Donahue, J. and Simonyan, K. (2019). Large scale gan training for high fidelity natural image synthesis, *International Conference on Learning Representations (ICLR)*.

Chen, J. and Liu, W. (2024). Advances in diffusion models for image data augmentation: A survey, *arXiv preprint arXiv:2407.04103* .
URL: *https://arxiv.org/abs/2407.04103*

Chen, R. and Yu, W. (2023). A review of data augmentation methods for remote sensing image scene classification, *Remote Sensing* **15**(3): 827.

CS231n, S. (2015). Tiny imagenet visual recognition challenge, `http://cs231n.stanford.edu/tiny-imagenet-200.zip`. Accessed July 2025.

Cubuk, E. D., Zoph, B., Mane, V. V. and Le, Q. V. (2019). Autoaugment: Learning augmentation strategies from data, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 113–123.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K. and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Doe, J. and Smith, J. (2023). Data augmentation with vector quantized variational autoencoders for image classification, *Pattern Recognition Letters* .

Friedman, M. (1937). The use of ranks to avoid the assumption of normality implicit in the analysis of variance, *Journal of the American Statistical Association* **32**(200): 675–701.

Garlan, D., Allen, R. and Ockerbloom, J. (1995). Architectural mismatch: Why reuse is so hard, *Proceedings of the 17th International Conference on Software Engineering*, IEEE, pp. 179–185.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y. (2014). Generative adversarial nets, *Advances in Neural Information Processing Systems*, Vol. 27, pp. 2672–2680.

Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P. H., Buchatskaya, E., Doersch, C., Pires, B. A., Guo, Z. D., Azar, M. G., Piot, B., Kavukcuoglu, K., Munos, R. and Valko, M. (2020). Bootstrap your own latent: A new approach to self-supervised learning, *Advances in Neural Information Processing Systems (NeurIPS)* **33**: 21271–21284.

He, K., Zhang, X., Ren, S. and Sun, J. (2016). Deep residual learning for image recognition, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B. and Hochreiter, S. (2017a). Gans trained by a two time-scale update rule converge to a local nash equilibrium, *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 6626–6637.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B. and Hochreiter, S. (2017b). Gans trained by a two time-scale update rule converge to a local nash equilibrium, *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 6626–6637.

Iqbal, H. and Zhang, B. (2025). Improving imbalanced medical image classification through gan-augmented learning, *Pattern Recognition* **144**: 109877.

Jiang, M. and Patel, A. (2024). Frontiers and developments of data augmentation for image classification, *Information Fusion* **98**: 102103.

Kendall, M. G. and Babington Smith, B. (1939). The new measure of rank correlation, *Biometrika* **30**(1/2): 81–93.

Krizhevsky, A. (2009). Learning multiple layers of features from tiny images, *Technical report*, University of Toronto. Technical Report.
URL: *https://www.cs.toronto.edu/ kriz/cifar.html*

Kutter, M., Jordan, F. and Bossen, F. (1998). A secure robust watermarking scheme, *Proceedings of the International Conference on Image Processing*, Vol. 2, IEEE, pp. 86–90.

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

Li, S., Kang, X., Fang, L., Hu, J. and Yin, H. (2017). A review of image fusion techniques, *Information Fusion* **36**: 119–132.

Li, Z. and Zhang, Y. (2023). Miamix: Enhancing image classification through a multi-stage augmented mixed sample method, *arXiv preprint arXiv:2308.02804* .

Mattson, P., Cheng, C., Coleman, C., Diamos, G., Micikevicius, P., Patterson, D., Tang, H., Wei, G.-Y., Bailis, P., Bittorf, V. et al. (2020). Mlperf training benchmark, *Proceedings of Machine Learning and Systems*, Vol. 2, pp. 336–349.

Nemenyi, P. B. (1963). *Distribution-free multiple comparisons*, PhD thesis, Princeton University.

Parnas, D. L. (1972). On the criteria to be used in decomposing systems into modules, *Communications of the ACM* **15**(12): 1053–1058.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J. et al. (2019). Pytorch: An imperative style, high-performance deep learning library, `https://pytorch.org`. Accessed July 2025.

Patel, R. and Lee, D. (2025). Enhancing image classification performance via gan-based data augmentation, *Pattern Recognition Letters* **175**: 45–52.

Perez, L. and Wang, J. (2017). The effectiveness of data augmentation in image classification using deep learning, *arXiv preprint arXiv:1712.04621* .

Radford, A., Metz, L. and Chintala, S. (2016). Unsupervised representation learning with deep convolutional generative adversarial networks, *arXiv preprint arXiv:1511.06434* .

Rahat, M. F., Rahman, A. and Etemad, A. (2025). Data augmentation for image classification using generative ai, *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 1534–1543.

Rahman, T. and Singh, R. (2023). Effective data augmentation with diffusion models, *arXiv preprint arXiv:2302.07944* .

Rao, A. and Tan, M. (2024). A systematic review of deep learning data augmentation in medical imaging, *Artificial Intelligence in Medicine* **145**: 102573.

Raschka, S. (2023). Comparing different automatic image augmentation methods, *sebastianraschka.com* . Accessed July 2025.

Ronneberger, O., Fischer, P. and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation, *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, Springer, pp. 234–241.

Sato, S., Yokoyama, R., Tsuchiya, A. and Tsubouchi, K. (2019). Mlops: Continuous delivery and automation pipelines in machine learning, *Proceedings of the 2019 IEEE International Conference on Big Data*, IEEE, pp. 5905–5907.

Segal, J. and Morris, C. (2005). Software engineering for computational science, *Computing in Science & Engineering* **7**(5): 50–59.

Shorten, C. and Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning, *Journal of Big Data* **6**(1): 1–48.

Smith, A. and Rodrigues, P. (2023). Survey of automated data augmentation algorithms for deep learning, *Knowledge and Information Systems* **65**(3): 781–803.

Sokolova, M. and Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks, *Information Processing & Management* **45**(4): 427–437.

Tan, M. and Le, Q. V. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks, *Proceedings of the International Conference on Machine Learning*, pp. 6105–6114.

Team, M. (2023). What is randaugment and how does it work, `https://milvus.io/ai-quick-reference/what-is-randaugment-and-how-does-it-work`. Accessed July 2025.

TorchVision Contributors (2025). Torchvision, `https://pytorch.org/vision`. Accessed July 2025.

van den Oord, A., Vinyals, O. and Kavukcuoglu, K. (2017). Neural discrete representation learning, *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 30.

Wang, X. e. a. (2024). A comprehensive survey on data augmentation, *arXiv preprint arXiv:2405.09591* .

Wikipedia contributors (2025). Data augmentation, `https://en.wikipedia.org/wiki/Data_augmentation`. Accessed July 2025.

Yun, S., Han, D., Oh, S. J., Chun, S., Choe, J. and Yoo, Y. (2019). Cutmix: Regularization strategy to train strong classifiers with localizable features, *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6023–6032.

Zhang, H., Cisse, M., Dauphin, Y. N. and Lopez-Paz, D. (2018). mixup: Beyond empirical risk minimization, *International Conference on Learning Representations*.

Zhao, K. and Kim, H. (2025). A survey of data augmentation in domain generalization, *Neural Computing and Applications* .

Zhong, Z. and Luo, Z. (2020). Improved regularization via cutout and random erasing, *Mathematical Biosciences and Engineering* **17**(4): 2728–2740.

Zoph, B., Ghiasi, G., Lin, T.-Y., Cui, Y., Liu, H., Cubuk, E. D. and Le, Q. V. (2020). Learning data augmentation strategies for object detection, *European Conference on Computer Vision (ECCV)*, Springer, pp. 566–583.