Democratic and Popular Republic of Algeria
Ministry of Higher Education and Scientific Research
University Mohamed Khider of Biskra

Faculty of Exact Sciences and Science of Nature and Life
Department of Computer Science

Ref: .....

Thesis Presented to Obtain the Degree of

Doctorate in Computer Science

Option: Image and Artificial Life

Entitled:

Incorporating Deep Learning and Optimization Techniques with Data Augmentation for Improved Image Analysis and Classification

Presented by:
Nouara BOUDOUH

Publicly defended on: 02/02/ 2025

In front of the Jury Committee composed of:

Mr. REZEG Khaled	Professor	President	University of Biskra
Mr. MOKHTARI Bilal	MCA	Supervisor	University of Biskra
Mm. DJEROU Leila	Professor	Examiner	University of Biskr
Mr. MELKEMI Kamal Eddine	Professor	Examiner	University of Batna2
Mr. SEBTI Foufou	Professor	Examiner	University of Sharjah, United Arab Emirates

Acknowledgement

All praise is due to God Almighty, who granted me the strength and perseverance to complete this work.

I extend my sincere thanks and appreciation to my esteemed supervisor ***Dr. Bilal Mokhtari****, for guiding me throughout this academic journey, offering valuable advice, and providing corrections that greatly contributed to the success of this work.*

I extend my thanks to all my esteemed professors for their knowledge and support, and to my friends who have been a great source of help and companionship.

Lastly, I would also like to express my deep gratitude to my husband and beloved family for their continuous support, patience, and encouragement, which gave me the strength and determination to reach this stage.

Dedication

To the soul of my father and mother.

To my husband and children.

To my sister-in-law, Akila.

To my brother, sisters, and all my dear family

thank you for your boundless love and support.

To all my friends and colleagues who have shared the difficult and joyful moments with me, my deepest gratitude.

With love,

Nouara

Abstract

Deep learning methods often face challenges due to unbalanced or non-representative data, and in many cases, data scarcity limits model effectiveness. We advocate that improving data quality can lead to significant performance enhancements. This thesis presents new methods for data augmentation. Our first method involves randomly create filters to remove certain rows and columns from the original image to generate smaller, more informative images. This method was applied to the Cats vs. Dogs dataset to train the Basic CNN and ResNet50 models, showing improved results compared to the original dataset. However, random filter generation can sometimes produce images that are too similar to the originals, reducing diversity. To address this, we developed a secondary technique incorporating a random optimization algorithm to select optimal generated images based on entropy, yielding promising results when applied to the VGG16 model. Nevertheless, image selection remains dependent on filter quality, potentially limiting diversity. Therefore, our third method employs a genetic algorithm to enhance filter generation and ensure greater diversity. Additionally, we improved the architectures of the VGG16 and VGG19 models. When applied to the Cats vs. Dogs and Chest X-ray datasets and used to train a set of seven models (VGG16, VGG19, their enhanced versions, EfficientNet-B0, Inception-V3, and Vision Transformer), we observed promising improvements in model performance compared to the second method. Since optimization techniques require considerable time and resources, we proposed an alternative method to enhance model performance without increasing data size. This approach leverages the unique capabilities of each model to extract features by merging their outputs into a unified representation used to train a single classifier. The integrated models using VGG16, VGG19, EfficientNet-B0, and Inception-V3 showed clear performance superiority compared to each model's individual performance.

Keywords: Deep Learning, Optimization, Image Analysis, Image Classification, Data Augmentation.

Résumé

Les méthodes de deep learning rencontrent souvent des défis liés à des données déséquilibrées ou peu représentatives, et dans de nombreux cas, la rareté des données limite l'efficacité des modèles. Nous proposons que l'amélioration de la qualité des données peut entraîner une hausse significative des performances. Cette thèse introduit des méthodes innovantes de l'augmentation des données. La première méthode repose sur la génération aléatoire de filtres pour supprimer certaines lignes et colonnes de l'image originale, produisant ainsi des images plus petites et informatives appliquée au dataset "Cats vs. Dogs" pour entraîner les modèles Basic CNN et ResNet50, elle a montré une amélioration par rapport à la base de données d'origine. Cependant, la génération aléatoire de filtres peut parfois produire des images trop similaires aux originales, limitant la diversité. Pour y remédier, nous avons développé une deuxième technique intégrant un algorithme d'optimisation aléatoire pour sélectionner les images optimales en se basant sur leur entropie, montrant de meilleurs résultats sur VGG16 par rapport à notre première proposition. Cependant, la dépendance de cette méthode à la qualité des filtres générés peut encore limiter la diversité. Ainsi, notre troisième méthode utilise une optimisation par algorithme génétique pour améliorer la génération de filtres et garantir une plus grande diversité. Par ailleurs, nous avons amélioré les architectures des modèles VGG16 et VGG19, et notre approche a montré des resultats significatives sur les bases de données "Cats vs. Dogs" et "Chest X-ray", en entraînant sept modèles (VGG16, VGG19, leurs versions améliorées, EfficientNet-B0, Inception-V3, et Vision Transformer). Comme l'optimisation demande du temps et des ressources, nous avons proposé une nouvelle méthode combinant les sorties de plusieurs modèles et alimentant un classificateur . Cette approche a surpassé les performances individuelles de chaque modèle, démontrant l'efficacité de nos contributions.

Mots-clés: Apprentissage Profond, Optimisation, Analyse d'Images, Classification d'Images, Augmentation des Données.

ملخص

غالباً ما تواجه أساليب التعلم العميق تحديات بسبب البيانات غير المتوازنة أو غير التمثيلية، وفي العديد من الحالات تحد ندرة البيانات من فعالية النموذج. نقترح أن تحسين جودة البيانات يمكن أن يؤدي إلى تحسين ملحوظ في الأداء. تقدم هذه الأطروحة طرقاً جديدة لزيادة البيانات. تعتمد طريقتنا الأولى على انشاء مرشحات واستخدامها لحذف خطوط وأعمدة معينة من الصورة الاصلية لتوليد صور جديدة أصغر وأكثر دلالة. تم تطبيق هذه الطريقة على مجموعة بيانات "Cats vs. Dogs" لتدريب نموذجين هما "Basic CNN" و"ResNet50". وقد أظهرت النتائج تحسناً في أداء النماذج مقارنة بقاعدة البيانات الأصلية. ومع ذلك، قد يؤدي التوليد العشوائي للمرشحات أحيانًا إلى إنتاج صور مشابهة أو قريبة جدًا من الصور الأصلية، مما يقلل من التنوع. لمعالجة هذه المسألة، قمنا بتطوير تقنية ثانية بادخال خوارزمية التحين العشوائي, والاستفادة من التكرارات المتعددة لاختيار الصورة المثلى المعززة بناءً على الإنتروبيا. وقد أظهرت هذه الطريقة، عند تطبيقها على نفس قاعدة البيانات لتدريب النموذج VGG16، نتائج واعدة متفوقة على الطريقة الأولى. ومع ذلك، يبقى اختيار الصور معتمدًا على جودة المرشحات، مما قد ينتج في كل تكرارصوراً شديدة الشبه بالأصلية ويحدّ من تنوع قاعدة البيانات. نظرًا لكون جودة الصور متعلق بجودة المرشحات، استخدمنا في طريقتنا الثالثة الخوارزمية الحينية لتحسين اختيار المرشحات وضمان تنوع أكبر. وكإضافة، قمنا بتحسين بنية نموذجين ، هما "VGG16" و"VGG19". وعند تطبيق طريقتنا على قواعد البيانات "Dogs vs. Cats" و"Chest X-ray" واستخدامها لتدريب مجموعة من سبعة نماذج (VGG16 VGG19 والنموذجين المحسنين وEfficientNet-B0 وInception-V3 وVision Transformer)، لوحظ تحسن واعد في أداء النماذج مقارنة بالطريقة الثانية. ولكون طرق التحسين تتطلب الكثير من الوقت والموارد، لذلك اقترحنا طريقة بديلة لتحسين أداء النموذج دون زيادة حجم البيانات. تستفيد هذه الطريقة من القدرات الفريدة لكل نموذج لاستخراج الميزات من خلال دمج مخرجاتها في تمثيل موحد يُستخدم لتدريب مصنف واحد. وقد أظهرت النماذج المدمجة باستخدام VGG16 VGG19 وEfficientnet-B0 وTnception-V3 تفوقًا واضحًا في الأداء مقارنة بأداء كل نموذج على حدة.

الكلمات المفتاحية: التعلّم العميق، التحسين، تحليل الصور، تصنيف الصور، وزيادة البيانات.

Contents

List of Figures

List of Tables

List of Algorithms

List of Abbreviations

ABC: Artificial Bee Colony
ACO: Ant Colony Optimization
AdaIN: Adaptive Instance Normalization
CBA: Color and Brightness Adjustments
CFFN: Coordinate Feature Fusion Network
CNN: Convolutional Neural Network
CutMix: Cutout and Mixup
DA: Data Augmentation
DE: Differential Evolution
DP: Dynamic Programming
FA: Firefly Algorithm
FC: Fully Connected
FGSM: Fast Gradient Sign Method
FN: False Negatives
FP: False Positives
GA: Genetic Algorithm
GAN: Generative Adversarial Networks
GC: Gradient Centralization
GJO: Golden Jackal Optimization
GP: Genetic Programming
GT: Geometric Transformations

IP: Integer Programming
LOA: Lion Optimization Algorithm
LP: Linear Programming
MoCo: Momentum Contrast
NANs: Neural Augmentation Networks
NAS: Neural Architecture Search
OCT: Optical Coherence Tomography
PBA: Population-Based Augmentation
PGD: Projected Gradient Descent
PSO: Particle Swarm Optimization
ResNet: Residual Networks
RGB: Red, Green, and Blue
RICAP: Random Image Cropping and Patching
RL: Reinforcement Learning
RO: Random Optimization
ROEDA: Random Optimization and Entropy-Based Data Augmentation
RS: Random Selection
SGD: Stochastic Gradient Descent
SSL: Self-supervised Learning
SwAV: Swapping Assignments between Views
TN: True Negatives
TP: True Positives
VAE: Variational Autoencoders
ViT: Vision Transformer
VGG: Visual Geometry Group
WBCs: White Blood Cells

List of Publications

International Journal Paper

- Boudouh Nouara, Bilal Mokhtari, and Sebti Foufou. Enhancing deep learning image classification using data augmentation and genetic algorithm-based optimization. International Journal of Multimedia Information Retrieval, 13(3):36, 2024.

International conference paper

- Boudouh Nouara and Mokhtari Bilal. Random pixel selection through image cropping for data augmentation and classification. In 2022 International Symposium on iNnovative Informatics of Biskra (ISNIB), pages 1–6. IEEE, 2022.

- Boudouh Nouara, and Mokhtari Bilal. "Random Optimization and Entropy-Based Data Augmentation for Image Classification and Analysis "ROEDA"." 2024 International Conference on Advances in Electrical and Communication Technologies (ICAECOT). IEEE, 2024.

- Concatenating Deep Features: An Advanced Technique for Image Classification using Ensemble Deep Learning (Presented in International Conference on Innovative and Intelligent Information Technologies (IC3IT'24),)

General Introduction

1 Problem Statement

In recent years, the field of image analysis has witnessed significant advancements, driven largely by the integration of deep learning techniques. Image analysis, which involves extracting meaningful in formation from digital images, plays a crucial role in various domains such as medical imaging, surveillance, remote sensing, and autonomous vehicles. Despite progress, challenges still exist in terms of accuracy, efficiency, and generalization, particularly when dealing with complex or high-dimensional datasets. To address these issues, researchers have increasingly turned to deep learning and optimization methods as powerful tools for enhancing image analysis.

Deep learning, a subset of machine learning based on artificial neural networks, has revolutionized the way images are processed and interpreted. Convolutional neural networks (CNNs), in particular, have demonstrated state-of-the-art performance in tasks such as object detection, image segmentation, and classification. The ability of deep learning models to automatically learn hierarchical features from raw images without the need for manual feature extraction has made them indispensable in image analysis. However, despite their success, deep learning models often require large amounts of labeled data for training and are prone to overfitting, especially in cases of insufficient data diversity. This has led to the exploration of various strategies, including data augmentation (DA) and optimization techniques, to improve model performance and robustness.

Optimization plays a fundamental role in enhancing both the efficiency and accuracy

of image analysis tasks. Optimization algorithms are employed at multiple stages, from fine-tuning the parameters of deep learning models to selecting the most relevant features for improving classification or segmentation results. Traditional optimization methods, such as gradient descent, have long been used in the training of deep neural networks. However, the rise of metaheuristic optimization techniques, including genetic algorithms, particle swarm optimization, and random optimization, has opened new avenues for solving complex image analysis problems. These algorithms leverage randomness and iterative search processes to explore a broader solution space, thereby improving the diversity and quality of the final results.

This thesis aims to explore the intersection of deep learning and optimization in the context of image analysis, with a particular focus on enhancing image classification and segmentation tasks. By combining the power of deep learning with advanced optimization techniques, we seek to improve the accuracy, efficiency, and generalization capabilities of image analysis systems. The proposed approaches will be validated through extensive experiments, demonstrating their potential to overcome existing challenges and push the boundaries of current image analysis methods.

While these developments are very encouraging, challenges are yet to be overcome regarding effective analysis similarity and classification of images. Moreover, coupled with increasing the number of images, the quality and preprocessing that goes into making image analysis models successful are critical. High-quality images with well-defined features are likely to help a model learn better, while normalization, noise reduction, and resizing are some of the preprocessing techniques helpful in standardizing input data for efficient training. Proper preprocessing ensures that the model does not pay unwanted attention to meaningless patterns, but rather meaningful ones; it also avoids variations in image quality, lightning, or noise. Therefore, balancing quantity and quality during data gathering is of prime importance for optimal model performance.

Several approaches can be employed to enhance the performance of deep learning models in image analysis. These include techniques like DA, which increases training data diversity and reduces overfitting, and transfer learning, which refines pre-existing

models for improved efficiency. Additionally, optimization methods can be used to accelerate model training, while advanced network structures enable more effective computation. Regularization techniques prevent overfitting by introducing controls during training, and ensemble learning enhances robustness by integrating the predictions from multiple models. Collectively, these strategies contribute to improved accuracy and better generalization in image analysis tasks.

Existing DA methods, while instrumental in expanding training datasets, often face significant challenges in maintaining image quality. Traditional techniques such as rotation, flipping, and scaling may create variations that fail to accurately represent real-world scenarios, resulting in less informative images for model training. These techniques can inadvertently introduce artifacts or distort essential features, ultimately compromising the overall quality of the augmented data. Since deep learning models heavily rely on high-quality labeled inputs, any inadequacy in the quality of augmented images can lead to diminished performance on unseen data.

To address these concerns, our proposed method focuses on enhancing the input of the CNN model (Images) by selectively choosing a specific set of pixels from the original image for augmentation. This approach emphasizes the retention of critical information while minimizing irrelevant details. We further enhance this pixel selection process by integrating Random Optimization (RO) and Genetic Algorithms (GA), which collaboratively improve the selection of the best-generated images based on quality and diversity. The optimization of architectures like VGG16 and VGG19 also contributes significantly to this performance boost.

The proposed method centers around the concatenation of features extracted from multiple models, which serves as an innovative strategy to enhance model performance in image analysis. By leveraging diverse architectures, this approach taps into the unique strengths of each model, allowing for a richer and more comprehensive representation of the input data.

Concatenating features from various models effectively merges different perspectives and insights gained during feature extraction, leading to a more robust understanding

of the underlying data patterns. This enriched feature representation can significantly improve the classifier's ability to differentiate between classes, especially in complex datasets where subtle variations are critical for accurate predictions.

These improvements by our proposed methods have resulted in significantly better performance measures that yield impressive performance on various image analysis tasks. The incorporation of state-of-the-art techniques, such as feature concatenation and optimization strategy, further improved the quality of generated augmented images and showed better generalization capability on unseen data. Experiments demonstrated the significant improvement in accuracy and robustness as compared to state-of-the-art approaches, which signifies the successful implementation of our methodology. These findings align with recent literature emphasizing the importance of using diverse feature sets in model performance enhancement, particularly in deep learning. Thus, our approach constitutes a significant stride in the field and promises to serve as one of the most promising avenues for future research and applications in image analysis.

2 Research Contributions

The primary contributions of this research include:

- Delivering a thorough literature review of the most pertinent image enhancement DA and optimization methods in image analysis.
- Propose a novel DA technique that involves randomly cropping rows and columns from the original image to generate augmented images.
- Enhance this technique by introducing a RO method to select the best images from the set of augmented samples.
- Further improve the initial contribution by utilizing GA to better explore the augmented image space and capture the most effective images, and Enhance the VGG16 and VGG19 architectures to improve classification accuracy, leveraging the benefits of the augmented dataset and optimization strategies.

- Enhancing model performance is achieved by concatenating the features extracted from the same images using different models, then feeding this combined feature set into a classifier.

3 Thesis Organization

The structure of the thesis is outlined as follows:

- General Introduction: outlines the research motivation and identifies the research problem. It is followed by a presentation of the thesis contributions and an overview of the thesis structure.
- Chapter 1: offers a general overview of image analysis, with a particular focus on image classification and, followed by a comprehensive review of DA techniques and existing methods, along with fundamental concepts of optimization methods and their applications in this field. Additionally, the chapter presents the datasets used for image classification tasks, details the advanced CNN architectures employed for feature extraction and classification, and discusses the various evaluation metrics used to assess model performance.
- Chapter 2: outlines the proposed DA method, which involves the random selection of rows and columns from the original image to generate new images. Additionally, it demonstrates improvements achieved through classification using two distinct models.
- Chapter 3: discusses the enhancement of the proposed method from Chapter 2 through the application of an RO method.
- Chapter 4: enhances the proposed method from Chapter 2 through the application of GA and evaluates the effectiveness of this enhancement using two datasets and seven models. Additionally, it focuses on improving the VGG16 and VGG19 architectures to further optimize performance.

- Chapter 5: presents an approach to improving model performance by concatenating the features extracted from the same images using different models, then feeding this combined feature set into a classifier.
- Conclusion and future works: concludes the thesis and outlines potential directions for future research.

Chapter 1

Fundamental Concepts and Related Work

1.1 Introduction

Image analysis is a critical area of research that applies computational methods to extract useful information from visual data. With the increasing number of digital images in all fields, such as medicine, agriculture, security, and entertainment, there has been an emerging need for strong analytical techniques. It covers various tasks of image analysis, from the simple operations of filtering and enhancement to the complex processes involving segmentation, feature extraction, and classification. In recent years, deep learning has revolutionized the field of image analysis by drastically changing the accuracy and effectiveness of classification techniques, enabling artificial neural networks to learn large datasets and make tremendous improvements in their performance.

Image classification is an important image analysis task. This task includes image classification into predefined classes according to the contents. Image classification has some important applications in object recognition, facial recognition, and scene understanding. However, in spite of its importance, image classification presents a large number of challenges. The data can give rise to several problems, such as class imbalance, which may result in model bias toward the classes that have higher representation. Furthermore, the noise in the data, such as errors or incorrect labels, may lead to false learning. The increase in dimensionality, or the number of features, further complicates the training process. One of the most important issues is the lack of data, and it affects the classification tasks immensely.

This chapter is organized as follows: Section 1.1 provides a general overview of image analysis and image classification, along with the associated challenges. In Section 1.2, we present a comprehensive review of significant areas and recent advancements in the field of image classification. Section 1.3 offers an overview of the concept and the literature on data augmentation. Following this, Section 1.4 discusses the challenges in the data augmentation domain. Section 1.5 outlines existing optimization methods, while Section 1.6 describes the datasets utilized in our study. Section 1.7 focuses on the CNNs employed, highlighting our contributions to enhancing specific models. Section 1.8 details

the evaluation metrics applied to assess the effectiveness of our methodologies. Finally, Section 1.9 concludes the chapter by summarizing the significance of the selected tools.

1.2 Image classification

Image classification is a computer vision task that identifies the class or label of an image. This normally involves machine learning algorithms or deep learning models that analyze various features of the image: color, texture, shape, patterns, among others. More specifically, it aims at the identification and classification of objects or scenes in an image into predefined classes to interpret the visual information automatically.

Some of the key challenges that an image classification faces are as follows: data imbalance, where some classes have a lot more samples than others, resulting in biased predictions; noisy data with bad quality or erroneous labels; variability among images due to differences in lighting, angles, and resolutions. In addition, the high dimensionality and complexity of objects in image data can complicate training. Overfitting is a risk, especially in the case of deep learning models, while adversarial attacks can undermine model reliability; classification tasks are further complicated by limited training data and the need for real-time processing.

Data scarcity presents a significant challenge in deep learning, often impeding model performance and generalization capabilities. To address this issue, several effective strategies have been developed, including Data Augmentation (DA)is a technique [1] such as rotation, scaling, flipping, cropping, and color adjustments can artificially increase the size and diversity of the dataset, helping the model generalize better. DA creates variations of the existing images to simulate a larger dataset. Transfer learning Utilizing pre-trained models on large datasets and fine-tuning them on smaller datasets can be highly effective. Transfer learning leverages the knowledge gained from large-scale datasets to improve performance on specific tasks with limited data [2]. Synthetic data aims to generate synthetic data using techniques such as Generative Adversarial Networks (GANs) or other data generation methods that can supplement the real dataset.

Synthetic data can help cover scenarios that may not be present in the real data [3]. Semi-supervised learning combines a small amount of labeled data with a larger amount of unlabeled data allowing the model to learn from both. Semi-supervised learning methods can help improve performance by leveraging the additional information in the unlabeled data [4]. Few-shot learning approaches are designed to train models to recognize new classes with very few examples. Techniques such as meta-learning and metric learning enable models to learn from limited data efficiently [5]. Data synthesis and augmentation including using 3D models or simulation environments, can create diverse and realistic training examples that supplement real-world data [6]. Crowdsourcing engaging the crowd to label data can help quickly expand the dataset. Crowdsourcing platforms can gather and label large volumes of images, although it requires careful quality control [7]. Applying regularization methods such as dropout [8], weight decay, or early stopping during training can help prevent overfitting and improve generalization when working with small datasets. Ensemble methods that Combine predictions from multiple models trained on the same data can improve overall performance. Ensemble methods can help mitigate the effects of limited data by leveraging the diversity of multiple models [9]. These solutions can be used individually or in combination to address the challenges posed by limited data in image analysis tasks.

1.3 Data Augmentation

Data analysis is a technique used to increase a dataset's size and diversity by creating modified versions of existing data samples. This is achieved through various transformations applied to the original data, which helps improve the robustness and generalization ability of machine learning models.

DA is applied to different types of data, such as images [10–13], audio [14, 15], and time series [16].

In image analysis, DA involves applying operations to images to produce variations that retain the original label but introduce new insights or features, Wong et al. [17] and

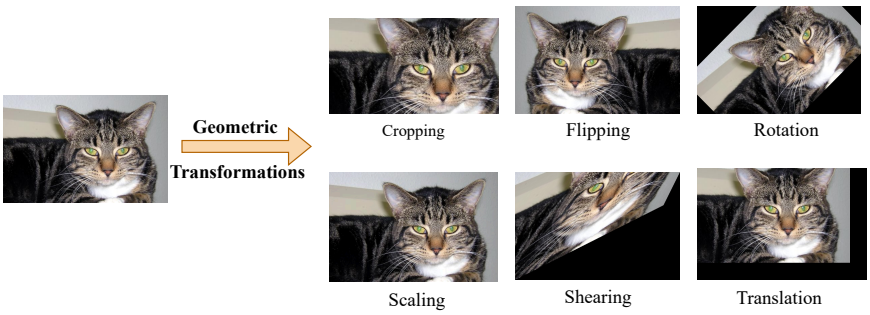


Figure 1.1: Examples of images generated by applying GT

Semenoglou et al. [18] divide the proposed methods of data augmentation into two main categories: classical and deep learning-based.

1.3.1 Classic approaches

Classic DA techniques are typically used in traditional image processing and machine learning workflows:

Geometric Transformations (GT): Geometric transformation in image processing refers to manipulating an image's spatial properties, such as:

- Rotation: Rotating images by various angles.
- Scaling: Resizing images to different scales.
- Translation: Shifting images horizontally or vertically.
- Cropping: Extracting random or centered portions of images.
- Flipping: Horizontally or vertically flipping images.
- Shearing: Applying geometric distortions to simulate perspective changes.

Figure 1.1 illustrates examples of images transformed through various geometric techniques. These transformations, such as rotation, scaling, and translation, aim to enhance the visual diversity of the dataset. By simulating different perspectives and orientations,

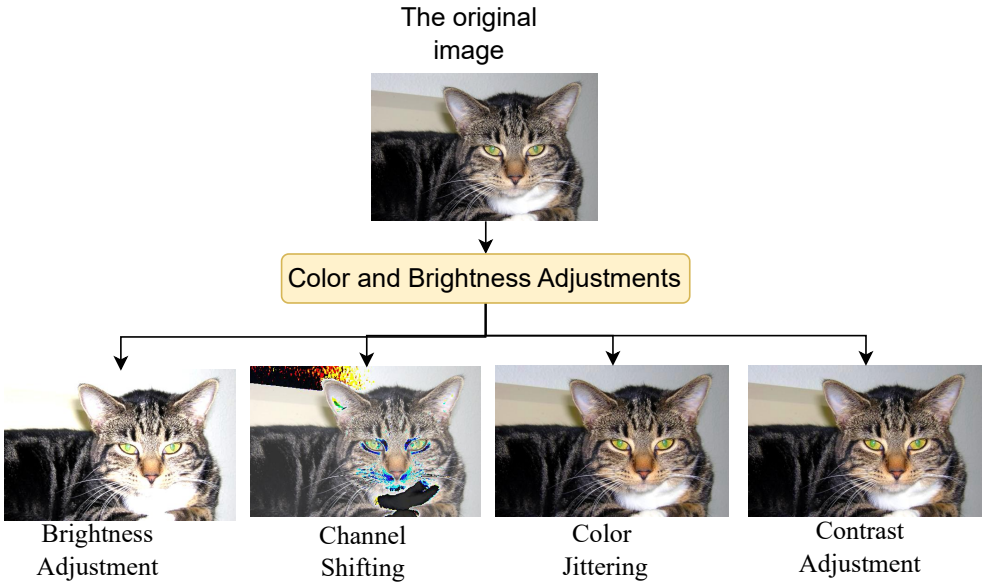


Figure 1.2: Example of images obtained by applying CBA

these modifications improve the robustness of deep learning models, enabling them to better handle variations they might encounter in real-world applications.

Color and Brightness Adjustments (CBA): CBA involves modifying an image's color balance and light intensity to enhance its visual appearance or meet specific analysis requirements.

- Brightness Adjustment: Altering the brightness of images.
- Contrast Adjustment: Changing the contrast levels of images.
- Color Jittering: Modifying color properties such as saturation, hue, and exposure.
- Channel Shifting: Adjusting the intensity of color channels.

Figure 1.2 illustrates examples of images obtained after applying various color and brightness adjustment techniques. These modifications are intended to enhance the visual diversity of the dataset, improving the robustness of deep learning models by sim-

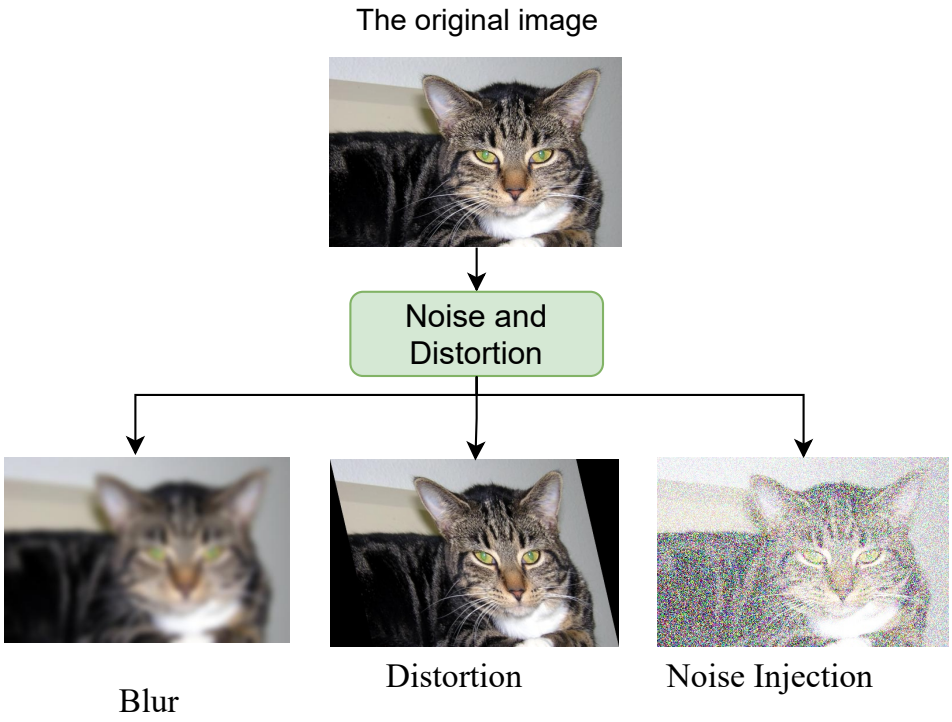


Figure 1.3: Example of Images Obtained by Applying Noise and Distortion

ulating different lighting conditions and color variations that the model might encounter during real-world applications.

Noise and Distortion: Noise and distortion can be strategically applied to create new images by introducing controlled random variations and systematic alteration, such as:

- Noise Injection: Adding random noise to images.
- Blur: Applying blurring effects to simulate out-of-focus conditions.
- Distortion: Introducing random distortions or warps to images.

Figure 1.3 displays an example of images obtained by applying noise and distortion

techniques. These adjustments are used to simulate real-world imperfections, such as random pixel variations and image degradation.

DA has attracted considerable research, following a review of the most important proposed methods. In a recent approach proposed by Paschali et al. [19], both affine and projective transformations are applied randomly on training dataset images. The technique underwent a thorough evaluation of the complex tasks of breast tumor classification from mammograms and fine-grained skin lesion classification from poor data. However, the method has some limitations, including a high computational cost due to the use of GANs, which require significant resources and lengthy training times. Additionally, the quality of the generated data can vary, potentially limiting the model's performance if the synthetic data lacks sufficient diversity. The approach is also complex to implement and heavily dependent on hyperparameter tuning, making it challenging to achieve optimal results.

Random Image Cropping and Patching (RICAP) is another DA technique proposed by Takahashi et al. [20]. This method involves randomly selecting and cropping four images from the training dataset, then patching these cropped portions together to form a new composite training image. The cropped regions from each of the four images are combined, with each segment contributing a part of the final image. During this process, the class labels are adjusted according to the area proportions of the crops, ensuring that the new training image represents a mixture of the classes from the original images. RICAP effectively increases the variability of the training samples by exposing the model to mixed samples, thus helping to improve its robustness and generalization ability. This approach is particularly useful in scenarios where the dataset is limited, as it allows the model to learn from a broader set of variations in the input data.

Elgendi et al. [21] tested how geometric augmentations used in recent papers affected the ability to detect COVID-19. Moreover, they evaluated the performance of 17 deep-learning algorithms with and without geometric augmentations. This method relies on geometric augmentations, which may not capture the full variability of medical imaging data. Moreover, it may not generalize to other medical tasks or datasets, and it requires

substantial computational resources, challenging its practical use in clinical settings.

Howard et al. [22] discuss the use of various photometric modifications, including random color jittering, which involves changing the brightness, contrast, and saturation of images. In the research of Zhang et al. [23], they introduce the Mixup augmentation technique, which involves the linear interpolation of pairs of images and their corresponding labels to generate new training examples. By blending images and labels, Mixup encourages the model to learn from a wider range of samples, thereby enhancing generalization and robustness. The paper provides theoretical analysis and empirical validation of Mixup across various deep learning tasks, including image classification and object detection. However, it lacks extensive exploration of Mixup's impact across diverse datasets or domains, thus restricting its applicability. Although Mixup demonstrates improved model performance in various tasks, its superiority over traditional augmentation methods may not be consistent. Moreover, the paper overlooks potential challenges like handling class imbalances and computational overhead during training, limiting its comprehensive evaluation.

Yen et al. [24] introduce a novel DA methodology known as CutMix. This technique amalgamates Cutout and Mixup augmentation approaches. CutMix involves the random selection of rectangular patches from two images during training and their replacement, promoting simultaneous learning from original and amalgamated regions. This method enhances feature learning and localization, thereby improving the generalization and robustness of CNNs. Experimental results across various computer vision tasks and datasets demonstrate the superiority of CutMix over other augmentation techniques in terms of accuracy and robustness. This technique offers an innovative approach to DA by blending images through overlapping patches. While this technique demonstrates improved model performance and feature localization in various computer vision tasks, it faces several limitations. These include dependency on image overlap, potential information loss during mixing, increased training complexity, and domain specificity. Addressing these limitations would enhance the applicability and robustness of CutMix across different datasets and domains.

The second group, however, includes photometric modification, the significant methods are color jittering, grayscaling, filtering, illumination perturbation, noise addition, vignetting, contrast adjustment, random erasing, etc. They change the RGB channels by changing pixel colors into new values. The color jittering technique employs a variety of manipulations, including inversion, addition, subtraction, and multiplication. Chen et al. [25] used a GridMask method involving the deletion of specific regions within the input image, introducing a grid-like pattern that obscures portions of the data. By strategically removing information in this grid-based manner, the augmentation technique aims to enhance the robustness and generalization capabilities of machine learning models, particularly in the context of image classification. This method may remove important features, introduce artificial artifacts, vary in effectiveness across different datasets and tasks, and incur significant computational overhead, posing challenges for widespread implementation.

Another approach presented by Zhong et al. [26] employs a method called Random Erasing which selects a rectangular section within an image during training and replaces its pixels with random values. This process introduces training images with varying levels of occlusion.

Many other methods have combined elements from both kinds of techniques. Among them, Kim et al. [27] proposed a method for generating new training images. Their approach includes image pre-processing steps, such as background removal and target extraction, while maintaining the original object size ratio. It also involves color perturbation, considering predefined similarities between the original and generated images, GT, and transfer learning. However, it relies on accurate initial segmentation for effective background removal and target extraction, which can be challenging in complex or varied backgrounds. While color perturbation and GT aim to enhance dataset diversity, their impact on model performance across different domains varies. Additionally, the method's computational complexity, involving multiple augmentation steps, may hinder scalability in resource-limited settings.

1.3.2 Deep learning-based DA approaches

Deep learning approaches for DA have evolved to include more sophisticated and automated methods, leveraging the power of neural networks to generate new training data or to optimize the augmentation process. Here are some notable deep learning-based DA techniques:

Generative Adversarial Networks (GANs) consist of a generator and a discriminator that collaborate to create realistic images. The generator aims to produce images that closely resemble the training data, while the discriminator works to differentiate between real and generated images. This dynamic allows GANs to augment datasets by generating synthetic samples that are similar to the original data, ultimately providing a more diverse and robust training set. Antreas et al. [28] introduce DA GANs (DAGANs). The goal of DAGANs is to address the challenge of limited data in machine learning, particularly for training deep neural networks. DAGANs use GANs to generate augmented samples for a given dataset. The key idea is to train a GAN model that, given an input sample, can generate new variations of that sample that are different yet still belong to the same class. This helps to improve the diversity of training data without explicitly collecting more real-world data. By using DAGANs, the authors demonstrate improvements in training classifiers, especially in few-shot learning scenarios where data is scarce. The proposed approach enhances the generalization capability of models by enriching the training data with more variations, which can help improve performance on unseen data.

Variational Autoencoders (VAEs) encode input images into a latent space and subsequently decode them to generate new, similar images. This approach is particularly useful for creating slightly altered versions of existing images, thereby enhancing the variability within the dataset and allowing for a more diverse set of training samples. Kingma et al. [29] introduce the concept of Variational Autoencoders (VAEs), which combine the principles of variational inference and deep learning to perform generative modeling. The authors present a novel approach for training latent variable models,

enabling efficient inference and generation of data. The VAE framework encodes input data into a lower-dimensional latent space and then decodes it to reconstruct the original data, allowing for the generation of new samples by sampling from the learned latent distribution. The work has significantly influenced the field of generative models and has applications in various domains, including image generation and semi-supervised learning.

Hou et al. [30] propose VA-GAN, a novel approach that combines Variational Autoencoders (VAEs) with GANs for visual anomaly detection. The VA-GAN framework is designed to effectively model the distribution of normal attributes in visual data while simultaneously detecting anomalies based on deviations from this distribution. By leveraging the strengths of both VAEs and GANs, VA-GAN is capable of generating realistic samples and identifying abnormal instances, making it a powerful tool for applications in surveillance, quality control, and medical imaging. The paper presents experimental results demonstrating the effectiveness of VA-GAN in improving anomaly detection performance compared to existing methods.

Neural Style Transfer: it is a technique that applies the visual style, including textures and colors, of one image to another while preserving the content of the target image. This approach facilitates the creation of images that exhibit different appearances but maintain the same underlying structure, ultimately increasing the variability within the dataset. Gatys et al. [31] introduce a neural algorithm for artistic style transfer that leverages deep CNNs. The core idea is to separate the content and style of images. The authors used a pre-trained CNN(specifically, VGG-19) to extract feature representations of both the content image and the style image. The algorithm defines a loss function that combines content loss (which measures the difference between the generated image and the content image) and style loss (which measures the difference between the generated image and the style image). Style is represented using Gram matrices that capture the correlations between different filter responses at various layers of the network.

Huang et al. [32] propose a method for arbitrary style transfer that allows for real-time applications by introducing Adaptive Instance Normalization (AdaIN). This technique modifies the content features extracted from a content image by adapting them to the statistics of the style features from a style image. The AdaIN process involves normalizing the content features and then scaling and shifting them using the mean and variance of the style features. This allows the model to effectively apply the style of any given image to the content of another in a single forward pass, significantly speeding up the style transfer process compared to previous methods that required iterative optimization.

AutoAugment: it employs reinforcement learning to identify optimal combinations of augmentation operations and their respective magnitudes. This approach automates the discovery of effective augmentation strategies, significantly reducing the time required for manual tuning and enhancing the efficiency of the DA process. Cubuk et al. [33] present AutoAugment, a framework that employs reinforcement learning to automate the discovery of effective DA strategies for deep learning models. By using a reinforcement learning controller, the method systematically learns augmentation policies from a diverse search space, including transformations like rotation and color adjustments. The study shows that models trained with these learned policies significantly outperform those using standard augmentations on benchmark datasets such as CIFAR-10 and ImageNet, enhancing model robustness and generalization. AutoAugment effectively saves time and improves performance by uncovering effective strategies that may not be readily apparent to human designers.

RandAugment: it is a simplified version of AutoAugment that reduces the search space by randomizing the selection of augmentation operations, optimizing only two parameters: the number of transformations applied and their magnitude. This approach decreases computational overhead while still providing the advantages of diverse augmentations, making it an efficient alternative for enhancing training datasets. Augment is a practical DA technique introduced by Cubuk et al [34]. This method simplifies the augmentation process by randomizing the selection of operations applied to training data, optimizing only two parameters: the number of transformations and their magni-

tudes. By doing so, RandAugment significantly reduces computational overhead while maintaining the effectiveness of diverse augmentations, achieving state-of-the-art performance on various benchmarks. The foundational concepts behind RandAugment are rooted in earlier works such as AutoAugment, which utilized reinforcement learning to discover optimal augmentation strategies.

Population-Based Augmentation (PBA) is an advanced DA technique designed to enhance the robustness and performance of machine learning models, particularly in the context of deep learning. As proposed by Lim et al. [35], PBA aims to optimize the augmentation process by treating it as a search problem, where different augmentation strategies are explored and evaluated to find the most effective combination for improving model accuracy.

Adversarial Training for DA focuses on generating adversarial examples—slight modifications of input images aimed at misleading models—and utilizing these examples to enhance the model's robustness and generalization. This technique is particularly useful in situations with limited training data or when models are susceptible to adversarial attacks. Key methods include generating adversarial examples using algorithms like the Fast Gradient Sign Method (FGSM) and Projected Gradient Descent (PGD), and incorporating these examples into the training dataset to help the model learn to differentiate between normal and adversarial samples.
The study of Madry et al. [36] focuses on enhancing machine learning models' robustness by incorporating adversarial examples into the training process. By introducing slight perturbations to the original data, this technique aims to expose models to challenging variations, which improves their ability to generalize to unseen data. The study demonstrates that integrating these adversarial examples leads to significant improvements in model performance across various tasks and datasets, ultimately bolstering defenses against adversarial attacks. For more details, you can refer to the paper itself or related works on adversarial training and DA strategies.

Goodfellow et al [37] explore the phenomenon of adversarial examples—inputs to ma-

chine learning models that have been intentionally perturbed to cause misclassification. The authors provide a theoretical framework to understand why neural networks are susceptible to these subtle alterations, proposing that adversarial examples arise from the linearity of neural network models in high-dimensional spaces. They also discuss methods to generate adversarial examples, such as the Fast Gradient Sign Method (FGSM), and emphasize the implications of these findings for the robustness of machine learning systems. It highlights the need for new training methodologies, including adversarial training, to enhance model resilience against such attacks. This work has significantly influenced subsequent research in adversarial machine learning.

Self-Supervised Learning (SSL) for DA leverages unlabeled data to train models to extract valuable features by solving pretext tasks, such as predicting image rotations or distinguishing different augmentations of the same image. This approach enables models to learn from the inherent patterns in data without requiring extensive labeled datasets, making it particularly useful in scenarios where labeled data is scarce. Key methodologies include contrastive learning frameworks like SimCLR, which maximizes the similarity between augmented views of the same data instance, and Momentum Contrast (MoCo), which enhances scalability in SSL. These techniques have shown significant success in various applications, including computer vision tasks like image classification and object detection.

SimCLR is a framework designed for contrastive learning (SimCLR) by Chen et al. [38] is a framework designed for contrastive learning that emphasizes the importance of DA in self-supervised representation learning. It operates by generating multiple augmented views of the same image and training a neural network to maximize agreement between these views while minimizing agreement with views from different images. The process utilizes a simple architecture with a projection head, making it straightforward to implement and efficient for learning useful visual features from unlabeled data (Chen et al., 2020).

Momentum Contrast (MoCo) proposed by Heet et al. [39] introduces a method for unsupervised visual representation learning by maintaining a dynamic dictionary of en-

coded features. This dictionary allows the model to contrast current input features against a larger set of previous representations, thereby improving the robustness of the learned features. The momentum encoder mechanism helps to keep the representations consistent over time, facilitating more stable learning from unlabelled data.

Swapping Assignments between Views (SwAV) proposed by Caron et al. [40] and focuses on learning visual features by contrasting cluster assignments rather than direct representations. The method employs a clustering approach, where it encourages similar images to share cluster assignments, thereby enhancing feature learning through the self-supervised framework. This allows SwAV to efficiently learn rich visual features without requiring labeled datasets, making it a powerful tool for various computer vision tasks. The cutout technique, introduced by DeVries et al. [41] is a straightforward regularization method that involves randomly masking square regions of input images during training. However, the cutout technique may obscure critical image features, potentially impacting model performance. Additionally, while it aids in regularization, it could introduce unnatural patterns that might bias learning outcomes. Its effectiveness may vary across different datasets and tasks beyond image classification, which limits its applicability. Moreover, the computational resources required for implementing Cutout on large datasets can be significant, posing practical challenges for widespread adoption.

Mixup is a DA technique proposed by Zhang et al. [42] that enhances the generalization capabilities of deep learning models, particularly in image classification tasks. The method involves creating new training samples by taking linear combinations of two randomly chosen images and their corresponding labels

CutMix is a DA technique introduced by Yun et al. [24] that enhances the training of neural networks by combining images and their labels. Instead of merely blending two images, CutMix involves cutting out a patch from one image and pasting it onto another. This is accompanied by an adjustment of the labels based on the proportion of each image in the mixed sample.

Neural Augmentation Networks (NANs) are advanced techniques in DA that utilize neural networks to generate new training samples, enhancing the diversity of

datasets. By creating augmented data that retains the semantic content of the original images while introducing variations, NANs help improve model robustness and generalization, particularly in scenarios with limited data. This approach not only addresses issues like overfitting but also enhances performance against adversarial attacks.

Xie et al. [43] introduces a framework for DA that leverages consistency training, where a model is trained to produce similar predictions for augmented versions of the same input. The authors propose using unsupervised methods to generate diverse augmentations, ultimately improving model robustness and generalization capabilities. Their approach demonstrates that consistent predictions across augmented data can significantly enhance performance, especially in tasks with limited labeled data.

AutoAugment, proposed by Cubuk et al. [33, 44], is a reinforcement learning-based method that automates the discovery of effective data augmentation (DA) policies. By searching for optimal combinations of augmentations that maximize model performance, AutoAugment alleviates the manual effort involved in data preprocessing. The method employs a reinforcement learning controller to select the best transformations and their magnitudes, optimizing techniques such as rotation and color jitter to enhance classification performance on validation sets. Experimental results demonstrate significant accuracy improvements across various benchmark datasets, including CIFAR-10, CIFAR-100, and ImageNet, underscoring the potential of automated augmentation in deep learning workflows.

Population-based Augmentation (PBA), proposed by Ho et al. [45], utilizes population-based training to dynamically adjust augmentation policies throughout the training process. This approach optimizes both the type and magnitude of augmentations, varying them over time to enhance model performance. PBA leverages evolutionary strategies to optimize augmentation schedules, adapting them to the current state of the model during training. By improving upon static augmentation strategies, PBA reduces the reliance on large validation sets typically required for policy optimization.

RandAugment, introduced by Cubuk et al. [34], simplifies the AutoAugment ap-

proach by reducing the number of search parameters involved in the augmentation process. Instead of searching for optimal augmentation policies, RandAugment randomly selects augmentation operations and focuses on optimizing just two hyperparameters: the number of operations and their magnitude. This simplification of the search space leads to a reduction in the computational cost required to find effective augmentations while still retaining performance improvements. As a result, RandAugment has demonstrated competitive results on datasets such as CIFAR-10 and ImageNet, Improving the diversity of the training dataset for better generalization of machine learning models. This is achieved through various optimization techniques, such as Genetic Algorithms, which optimize the selection and combination of augmentation techniques like rotation, flipping, scaling, or color jittering. Additionally, AutoAugment, a reinforcement learning approach, learns optimal augmentation policies to enhance model robustness. By employing these strategies, optimizing DA with AutoAugment significantly improves generalization in image classification tasks.

Genetic Algorithms have been effectively employed to optimize image augmentation techniques by evolving a population of augmentation strategies over multiple generations. These strategies encompass a range of transformations, including scaling, flipping, and brightness adjustments. The Genetic Algorithms search for augmentation parameters that yield the best classification performance, utilizing selection, mutation, and crossover to explore diverse augmentation strategies. This approach can significantly enhance performance, particularly when integrated with deep learning models, as the GA efficiently identifies augmentation policies that generalize better on the training data. The methodology is discussed in detail in [46].

1.4 Challenges and Difficulties

Data augmentation techniques enhance model performance in image analysis; however, each category has its limitations

1.4.1 Limitations of traditional approaches

Geometrical transformation (GA) methods of DA carry a lot of significance in terms of enriching machine learning models' training spaces. However, GT, such as rotation or deformation, can affect changes in meaning, particularly for domains with high positional and orientational sensitivity, such as computer vision. The consequence of this may be undesirable distortions. Some of the algorithms for transformation rely on parameters [46], [47]. Insufficient tuning of the parameters may lead to unrealistic or unnecessary DA, which results in an adverse effect on quality training. GT is resource-intensive, especially for large datasets. It leads to high costs in terms of training time and computation. The effectiveness of these methods could vary depending on data forms, such as structured data.

This loss of semantic information introduced by photometric variations in brightness, contrast, or hue can influence object interpretation in recognition applications where it may be critical. This limits generalization because models become sensitive to the variations that these modifications introduce, which negatively affect real-world performance. The models are less robust to substantial changes in initial lighting. Photometric modifications require deep domain expertise because inappropriate settings result in unrealistic or biased scenes. Complex photometric modifications carry a computational overhead, hence affecting the training speed of the models. On the other hand, this may be very useful for allowing the model to better generalize in DA, this cropping off a part of the image may provide robustness by allowing it to learn and extract the pattern when parts or regions are occluded or missing pieces, learn from the surrounding regions of a missing part. Besides, the generation of new images with different subsets of missing pixels keeps the training data variable, not dependent on specific features or specific spatial configurations of pixels

1.4.2 Limitations of deep learning-based approaches

Deep learning-based approaches to data augmentation face several limitations, including high computational costs that require substantial resources for training and inference. Their complexity can also pose challenges, necessitating specialized expertise for effective implementation. Furthermore, these methods risk overfitting, as they may produce synthetic data that fails to accurately represent real-world scenarios. Lastly, the success of deep learning-based augmentation is heavily reliant on the quality and quantity of the original training data, which can limit their effectiveness if the data is inadequate.

In addition to methods proposed to address data scarcity in classification, optimization is essential for boosting model accuracy. This process can involve enhancing image quality or creating more diverse data points through techniques like noise reduction, which improve the robustness of input features and support better classification performance. Additionally, directly optimizing model parameters, such as through hyperparameter tuning or fine-tuning layers in deep learning models, is a standard approach for increasing accuracy and robustness, particularly in deep learning applications

1.5 Optimization methods

Optimization can be defined as trying to find the best solution out of a set of feasible solutions by either maximizing or minimizing a given objective function. This is achieved by making systems or decisions as effective as possible. Optimization can take many classifications. The most common techniques in solving optimization problems include techniques such as gradient descent and evolutionary algorithms. This is a very important process in operations research, economics, engineering, and machine learning, where the procedure improves performance, trims costs, and makes improved decisions.
Optimization methods can be broadly categorized into different approaches, each suited for specific problems. Optimization methods can be broadly categorized into different approaches, each suited for specific problems. In the following subsection, we will explore

these optimization methods in detail, examining their characteristics.

1.5.1 Exact methods

Exact optimization methods are techniques designed to find the optimal solution to a problem, ensuring the best outcome according to a defined objective function. In the following, we present an overview of the most prominent methods.

Linear Programming (LP): is a mathematical optimization technique used to find the best outcome in a mathematical model whose requirements are represented by linear relationships. It involves maximizing or minimizing a linear objective function, subject to a set of linear constraints.

Integer Programming (IP): employing branch and bound and cutting planes for problems with integer constraints.

Dynamic Programming (DP): is a method for solving complex optimization problems by breaking them down into simpler overlapping subproblems. It solves each subproblem only once and stores the results, avoiding the need to recompute them.

Branch and bound: it is an algorithm for solving discrete optimization problems, especially in integer programming. It explores the solution space by dividing it into subproblems, evaluating each with bounds, and discarding suboptimal ones to avoid unnecessary searches.

Constraint Programming: which focuses on satisfying constraints rather than optimizing an objective.

1.5.2 Heuristic and metaheuristic methods

Heuristic and Metaheuristic methods are approximation techniques used to find near-optimal solutions for complex optimization problems when exact methods are impractical due to time or computational constraints. In the following, we present an overview of

the most prominent methods.

Rndom Optimization (RO): introduced by Anderson et al. [48], encompasses a variety of optimization techniques that operate without the need for gradient information. This characteristic makes these methods particularly effective for handling non-continuous or non-differentiable functions.

Genetic Algorithms (GA) introduced by Sampson [49] Mimic natural selection by generating populations of potential solutions and applying crossover, mutation, and selection to evolve better solutions.

Differential Evolution (DE) introduced by Storn [50] uses population-based optimization where differences between randomly selected pairs are added to generate new candidate solutions.

Simulated annealing: presented by Kirkpatrick et al. [51] Inspired by the annealing process in metallurgy, it explores the solution space by probabilistically accepting worse solutions initially to escape local minima.

Tabu search: is a technique proposed by Glover et al. [52] Uses memory structures to avoid revisiting recently explored solutions, enhancing the search for optimal results in complex spaces.

Swarm intelligence

- **Ant Colony Optimization (ACO)**: presented by Dorigo et al. [53] Inspired by the behavior of ants searching for paths between their colony and food sources, used for solving discrete optimization problems.
- **Artificial Bee Colony (ABC)**: Simulates the foraging behavior of bees, useful in both continuous and discrete optimization.
- **Particle Swarm Optimization (PSO)**: proposed by Kennedy et al. [54]Simulates the social behavior of birds or fish, where particles move through the solution space, influenced by their best-known positions and the group's best position.
- **Crow Search Algorithm:** it is a nature-inspired metaheuristic optimization

method based on the intelligent behavior of crows, specifically their habit of storing food in hidden places and the tendency to follow other crows to discover their food locations. It was introduced by Askarzadeh et al. [55] and has been used for solving various continuous and discrete optimization problems.

- **The Firefly Algorithm (FA):** it is a bio-inspired optimization technique introduced by Yang et al. [56], which simulates the flashing behavior of fireflies to solve complex optimization problems across various domains such as engineering, computer science, and data analysis.
- **Lion Optimization Algorithm (LOA)**: it a method proposed by Yazdani [57] is a nature-inspired metaheuristic optimization technique that simulates the social behaviors and hunting strategies of lions. Developed to solve complex optimization problems

Gradient-Based methods:

introduced by Cauchy et al. [58] iteratively adjusts parameters in the direction of the negative gradient to minimize a cost function, with variants such as Stochastic Gradient Descent (SGD), Mini-batch Gradient Descent, and Adam. Newton's Method, on the other hand, uses second-order derivatives to identify points where the gradient equals zero, providing faster convergence but requiring the computation of the Hessian matrix.

Gradient-Free Optimization

- **Genetic Programming (GP)**: Proposed by Koza et al. [59], GP is a technique that evolves programs or functions to address complex problems, making it particularly valuable in scenarios where gradients are difficult or impossible to compute.
- **Bayesian optimization:** This is a probabilistic model-based approach commonly employed for hyperparameter tuning. It utilizes a probabilistic model of the function being optimized, allowing each evaluation step to be informed by previous results and thereby improving the efficiency of the search process.

Machine learning-based optimization

- **Reinforcement Learning** (RL): Initially proposed by Watkins et al. [60], RL enables agents to learn decision-making strategies through interactions with their environment. This approach is particularly effective for optimizing complex decision-making processes across various applications.
- **Neural Architecture Search (NAS)**: Introduced by Zoph et al. [61], NAS leverages optimization techniques to automate the search for optimal neural network architectures, thereby enhancing model performance and reducing the need for manual design.

1.5.3 Convex optimization

This approach focuses on optimizing convex functions over convex sets, ensuring that any local minimum is inherently a global minimum due to the problem's structure. It is particularly suited for problems characterized by well-defined mathematical properties, making it a versatile tool across various domains. The comprehensive framework for convex optimization was extensively detailed by Boyd et al. in their foundational work [62].

1.5.4 image optimization-based classification

Image classification-based optimization leverages optimization techniques to enhance the accuracy and efficiency of classifying images into predefined categories.

Khan et al. [63] presents an innovative methodology for the classification of optical coherence tomography (OCT) images, integrating deep learning techniques with ant colony optimization (ACO). This research introduces a hybrid model that effectively combines the feature extraction prowess of deep learning networks with the optimization capabilities of ACO, resulting in enhanced classification accuracy. The findings suggest that this approach surpasses conventional image classification methods, highlighting its

considerable potential to advance diagnostic effectiveness in medical imaging.

Lee et al. [64] review various swarm intelligence algorithms, including particle swarm optimization and ant colony optimization, along with their applications in image classification and other image processing tasks. The paper discusses the performance and improvement strategies of these algorithms in the context of image segmentation, matching, and feature extraction.

Yong et al. [65] present a new optimization technique that has been developed to improve the efficiency and performance of deep neural networks during training, which has been called Gradient Centralization. GC centralizes gradients by subtracting their mean during backpropagation, which stabilizes and accelerates optimization. The authors have shown that this scheme leads to an improved convergence rate, reduced oscillation during training, and better generalization for many deep-learning models. They further validate the empirical effectiveness of GC on various applications through extensive experimentation, including image classification tasks. This work emphasizes GC as a simple yet powerful technique that easily integrates with most existing training pipelines to further raise performance.

Sadeghi et al. [66] propose a new multi-objective binary chimp optimization algorithm with the aim of elevating feature selection processes. This approach seeks to optimize the multi-objectives comprising feature relevance and classification accuracy that are very essential for model performance improvement. This approach is then applied to Synthetic Aperture Radar image classification as an example, showing clearly the elevated effectiveness of the choice of features from high-dimensional complex datasets. This work has shown a remarkable improvement in classification performance along with computational efficiency by incorporating an optimized feature set into deep learning models. They further demonstrate, through a series of extensive experiments, that the method outperforms the current feature selection methods; hence, the work represents valuable insights into remote sensing and image analysis.

Ling et al. [67] offers two new methods to enhance image detection for autonomous driving. Receptive Field Attention Convolution enriches feature capture by adjusting the

receptive field, while a triplet attention mechanism focuses on the target object, context, and background. These two together will lower the barrier of real-time detection by raising both the accuracy and efficiency. It follows that the experimental results on benchmark datasets show significant improvements compared to existing state-of-the-art methods, further guaranteeing safe and reliable autonomous vehicle operations.

1.6 Experimental Datasets

To evaluate the effectiveness of our proposed methods, we conducted experiments on two challenging datasets: the Cats vs. Dogs dataset [68] and the Chest X-ray dataset [68]. These datasets were selected for their distinct classification difficulties: differentiating between cats and dogs in the first dataset, and distinguishing between normal and pneumonia conditions in the second.

1.6.1 Cats vs. Dogs dataset

The Cats vs. Dogs dataset is a widely used benchmark in computer vision, aimed at binary image classification tasks involving cats and dogs. It comprises 24,989 images, categorized into two classes: cats and dogs. The dataset is partitioned into 19,998 images for training, 2,496 for testing, and 2,495 for validation, with a balanced representation of both classes. This dataset presents significant challenges due to the high variability within each class and the visual similarities between the two, making it an excellent test for assessing the precision of classification models.

1.6.2 Chest X-ray dataset

The Chest X-ray dataset is a critical resource for medical image analysis, consisting of 5,856 Chest X-ray images labeled into two classes: NORMAL and PNEUMONIA. It is divided into 5,216 training images (1,341 in the NORMAL class and 3,875 in the PNEUMONIA class) and 624 testing images (234 NORMAL and 390 PNEUMONIA). This

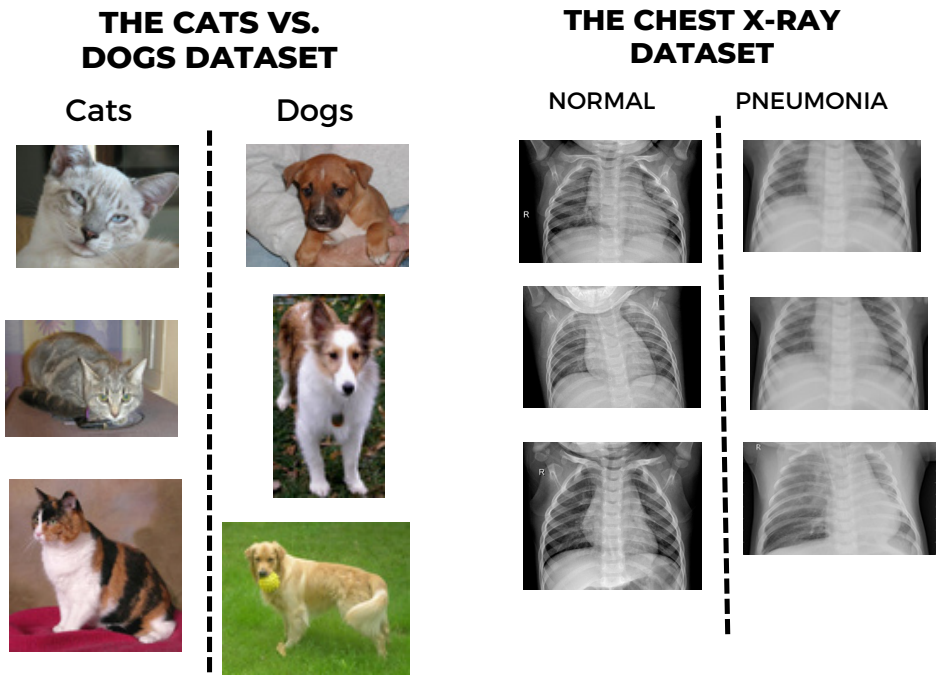


Figure 1.4: Sample images from the two datasets used.

dataset presents a substantial challenge as the visual distinctions between normal and pneumonia-infected lungs can be subtle, requiring a model with strong feature extraction capabilities. All X-ray images were obtained as part of routine clinical care and were sourced from retrospective cohorts of children aged one to five years (some examples in Figure 1.4).

1.7 Experimental Deep Learning Architectures

In our study, we employ seven prominent models: Basic CNN, ResNet-50, VGG16, VGG19, Inception-V3, Vision Transformer (ViT), and EfficientNet-B0—to train on the aforementioned datasets.

Layer (type)	Output Shape	Param #
conv2d_6 (Conv2D)	(None, 32, 32, 32)	896
dropout_6 (Dropout)	(None, 32, 32, 32)	0
conv2d_7 (Conv2D)	(None, 30, 30, 32)	9,248
max_pooling2d_3 (MaxPooling2D)	(None, 15, 15, 32)	0
flatten_3 (Flatten)	(None, 7200)	0
dense_6 (Dense)	(None, 512)	3,686,912
dropout_7 (Dropout)	(None, 512)	0
dense_7 (Dense)	(None, 2)	1,026

Total params: 3,698,082 (14.11 MB)
Trainable params: 3,698,082 (14.11 MB)
Non-trainable params: 0 (0.00 B)

Figure 1.5: Architecture of a basic CNN model.

1.7.1 Basic CNN

First, we used a simple CNN architecture designed for binary image classification with an input shape of 32×32 pixels and 3 color channels (RGB). As illustrated in Figure 1.5, the architecture includes two convolutional feature extraction layers interspersed with dropout layers to mitigate overfitting. A max-pooling layer downsamples the feature maps, reducing dimensionality while preserving essential information. The flattened layer converts the 2D feature maps into a 1D vector, followed by a fully connected layer that learns complex patterns. Finally, the output layer predicts probabilities for two classes, enabling effective binary image classification.

1.7.2 ResNet50

ResNet50, short for Residual Network with 50 layers, is a deep CNN architecture introduced by Kaiming He et al. [69]. ResNet50 is renowned for its innovative use of residual connections, which address the problem of vanishing gradients in deep networks. These residual connections allow the network to learn identity mappings, enabling the con-

struction of very deep networks without suffering from performance degradation.

1.7.3 VGG16 and VGG19

VGG16 and VGG19 [70] are well-established CNNs designed for image classification and recognition tasks. VGG16 consists of 16 weight layers, including 13 convolutional layers and 3 fully connected layers, while VGG19 is a deeper variant with 19 weight layers, comprising 16 convolutional layers and 3 fully connected layers. Both architectures follow a consistent design principle, utilizing small 3x3 convolutional filters stacked in series to increase the depth of the network. This approach allows the models to capture intricate patterns and hierarchical features from input images. After each convolutional layer, a ReLU activation function is applied to introduce non-linearity. Max-pooling layers are strategically placed between the convolutional blocks to downsample the feature maps, reducing spatial dimensions while preserving critical information.

As an additional step in our research, we enhanced the structure of the two models we utilized, namely VGG16, and VGG19 (Section 1.7). This adjustment followed numerous experiments. These enhancements significantly bolstered the models' performance, yielding better results.

1.7.4 The enhanced VGG16 and VGG19

The transition from convolutional layers to fully connected layers enables the network to synthesize the learned features into high-level representations for classification. The final softmax layer outputs class probabilities. The primary distinction between VGG16 and VGG19 is their depth—VGG19 includes three additional convolutional layers, which enable it to capture more complex and nuanced features, potentially improving classification performance, particularly in datasets with subtle visual differences. Both models are recognized for their simplicity and ability to generalize well across a range of computer vision tasks.

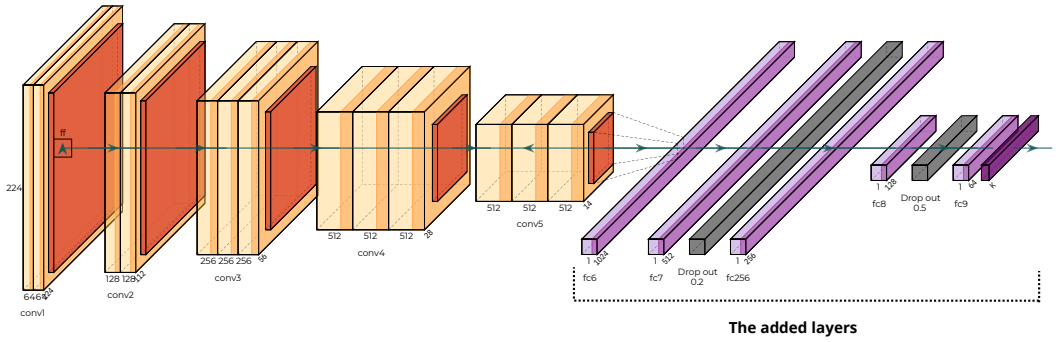


Figure 1.6: The modified architecture of the VGG16 model.

We enhanced the standard VGG16 and VGG19 architectures by incorporating additional fully connected layers to increase their capacity for feature learning. Specifically, we introduced fully connected layers with 1024 and 512 units, respectively, followed by a dropout layer with a rate of 0.2 to prevent overfitting. Afterward, another set of fully connected layers with 256 and 128 units, respectively, was added, each accompanied by a dropout layer with a rate of 0.5. Finally, a fully connected layer with 64 units was appended to the network structure, as depicted in Figures 1.6 and 1.7. This refined architecture was the result of extensive experimentation, and the configuration was selected based on its superior performance in our tests.

1.7.5 Inception-V3

Inception-V3 [71] is a CNN designed for efficient image analysis and object detection, with a focus on balancing model complexity and computational efficiency. Its architecture is built around Inception modules, which enable the network to capture features at multiple scales by applying convolutional filters of varying sizes (1x1, 3x3, and 5x5) within the same layer. This multi-scale approach allows the model to extract both fine and coarse features from the input image, promoting a richer feature representation.

A key innovation of Inception-V3 is the use of 1x1 convolutions for dimensionality reduction. By reducing the number of input channels before applying larger convolutional

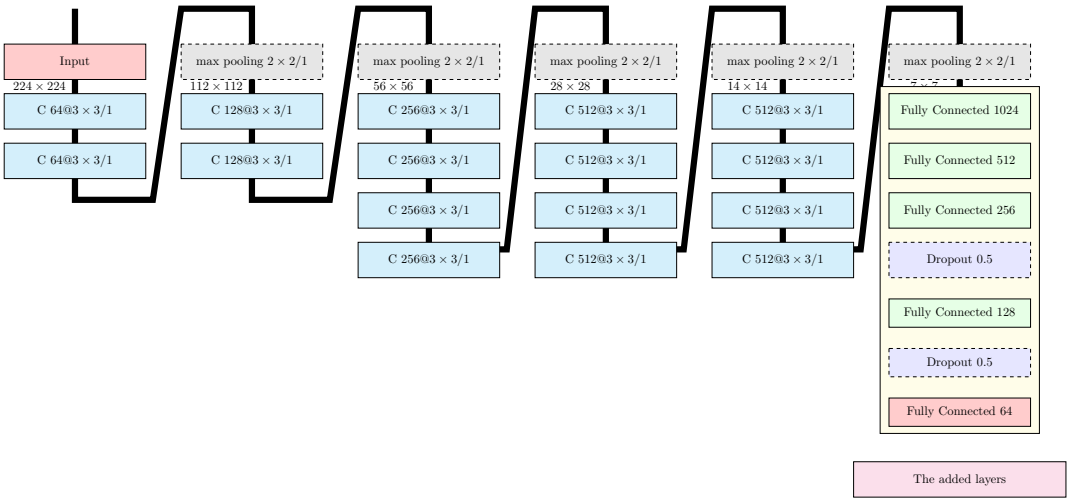


Figure 1.7: The modified architecture of the VGG19 model.

filters, the network significantly lowers its computational cost while maintaining performance. Inception-V3 consists of 48 layers, including convolutional layers, max-pooling layers, and fully connected layers, and processes input images of size 299x299 pixels.

Despite its depth and complexity, Inception-V3 is designed to be parameter-efficient, with fewer than 25 million parameters compared to the 60 million in AlexNet [72]. This efficiency allows the network to be both deeper and wider, facilitating the learning of complex patterns while keeping the model relatively lightweight. Furthermore, Inception-V3 has been extensively trained on large-scale datasets such as ImageNet, and its pre-trained weights enable effective transfer learning, making it a highly versatile and powerful model for a wide range of computer vision tasks, including image classification and object detection.

1.7.6 The Vision Transformer (ViT)

The Vision Transformer (ViT), introduced by Dosovitskiy et al. [73], represents a groundbreaking shift in computer vision by adapting transformer architecture, traditionally used in natural language processing, to image data. Unlike conventional CNNs, ViT divides images into sequences of fixed-size patches, which are then treated similarly to tokens in

a transformer model. Through self-attention mechanisms, ViT captures global dependencies between these patches, enabling it to model relationships across the entire image. This approach allows for highly effective feature extraction, with the self-attention layers capturing both local and long-range dependencies. ViT has shown competitive performance across a variety of tasks, including image classification and object detection, often surpassing traditional CNN-based architectures. Its flexibility in handling different input sizes and its strong ability to model complex spatial relationships make ViT a versatile and promising model for advancing computer vision techniques.

1.7.7 EfficientNet-B0

EfficientNet-B0, introduced by Tan et al. [74], is an exceptionally efficient CNN designed to optimize performance while minimizing computational costs. It achieves this through depthwise separable convolutions, which reduce the number of parameters and computations required without sacrificing accuracy. EfficientNet-B0 employs a compound scaling method that uniformly scales the network's width, depth, and resolution, ensuring optimal efficiency across varying computational budgets. The architecture starts with a stem convolution and progresses through several blocks of depthwise separable convolutions, totaling approximately 28 layers. The final layers include global average pooling and a fully connected classification layer that produces the final predictions. By balancing efficiency and performance, EfficientNet-B0 offers state-of-the-art results across multiple image classification tasks while remaining computationally economical, making it ideal for resource-constrained environments.

1.8 Evaluation Metrics

In deep learning, evaluating a model's performance is crucial to ensure its effectiveness and reliability. Several metrics are used to assess different aspects of the model's ability to make correct predictions. Below are short descriptions and formulas for some commonly used metrics:

	Predicted Positive	Predicted Negative
Actual Positive	TP	FN
Actual Negative	FP	TN

Table 1.1: Confusion matrix

1.8.1 Confusion matrix

The confusion matrix (Table 1.1)is a table used to describe the performance of a classification model. It compares the actual target values with those predicted by the model. The matrix is particularly useful for visualizing the performance of a model and calculating various metrics.

A typical confusion matrix for binary classification is structured as follows:

Where:

- TP = True Positives: The model correctly predicts the positive class.
- TN = True Negatives: The model correctly predicts the negative class.
- FP = False Positives: The model incorrectly predicts the positive class.
- FN = False Negatives: The model incorrectly predicts the negative class.

1.8.2 Accuracy

Accuracy quantifies the proportion of correctly classified instances over the total number of instances, serving as a key metric in classification tasks. It is defined in Equation (1.1).

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{1.1}$$

1.8.3 Error rate

The error rate indicates the proportion of misclassified instances out of the total instances, acting as the complement to accuracy. It is formally expressed in Equation

(1.2).

$$\text{Error Rate} = \frac{FP + FN}{TP + TN + FP + FN} \tag{1.2}$$

Alternatively:

$$\text{Error Rate} = 1 - \text{Accuracy} \tag{1.3}$$

1.8.4 Recall

Recall, also known as sensitivity or true positive rate, measures the proportion of actual positives that are correctly identified by the model. It is formally defined in Equation (1.4).

$$\text{Recall} = \frac{TP}{TP + FN} \tag{1.4}$$

1.8.5 F1 score

The F1 Score is the harmonic mean of precision and recall, providing a balance between them. It is particularly useful when the class distribution is imbalanced. It is defined as shown in Equation (1.5).

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{1.5}$$

Where:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{1.6}$$

1.9 Conclusion

In conclusion, this chapter has presented the fundamental concepts of image analysis, image classification and optimization, highlighting their essential role in improving model performance. We also provided an overview of existing methods, showcasing their diver-

sity and effectiveness in addressing the challenges in the field. In the upcoming chapter, we will explore a comprehensive study on DA, highlighting the proposed methods and recent advancements in this field. We will also discuss the tools utilized in our research, including datasets, models, and metrics employed to evaluate the results obtained.

Chapter 2

Random Pixel Selection through Image Cropping for Data Augmentation and Classification

2.1 Introduction

In this chapter, we introduce a novel, DA method that utilizes random filters to generate new images to add more images to the dataset, enhancing the training process. Each filter determines which pixels to retain from the original image, resulting in a new image with fewer pixels. The idea behind this method is that the most essential details in an image are frequently located in a more minor part.
The experimental results reveal that training the Basic CNN and ResNet50 separately with both versions of the Cats vs. Dogs dataset—namely, the original and augmented datasets created using the proposed approach—leads to improved performance. This enhancement is specifically due to the training of the models on the augmented dataset. This chapter is divided into four sections. Section 2.1 presents an introduction. The proposed methods are detailed in Section 2.2. Section 2.3 focuses on the experimental validation, including the challenges of the proposed method. Finally, Section 2.4 concludes the chapter by summarizing the key findings of the study.

2.2 Proposed Method

Our contribution proposed in [75] involves selecting a subset of pixels from specific rows and columns of an image to generate new images. To streamline this process during implementation, we employ a technique that crops certain rows and columns using randomly generated filters while preserving the remaining parts to form a new image. This method is influenced by two key factors that significantly impact the results: the number of selected pixels and the positions of the chosen rows and columns. The objective is to enrich the dataset by generating new images that differ from the originals, thereby diversifying the data and mitigating overfitting. While the proposed method focuses on the random selection of preserved pixels, we denote it as RS (for Random Selection).

In contrast to the random erasing DA method proposed by Zhong et al. [26], which alters pixel values without changing the image size, and the cutout method by DeVries

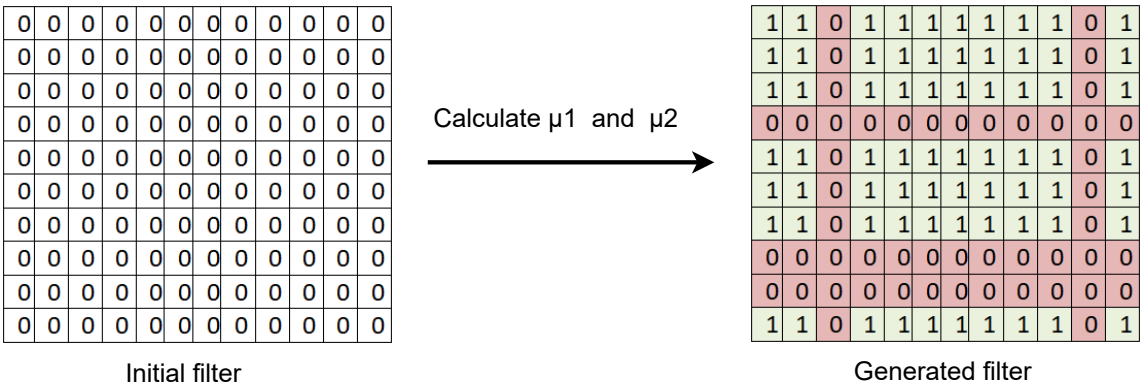


Figure 2.1: Filter generation process.

et al. [41], which masks a region of the image potentially losing crucial information, our approach selectively crops non-adjacent rows and columns. This ensures the retention of important information from various regions of the image while also creating smaller images, reducing the overall memory requirements of the augmented dataset.

By selecting specific pixels, our method identifies key regions in images (representative regions), which helps improve the classification process. For each image in the original dataset, we generate multiple filters corresponding to the number of new images to be created. Each filter is a matrix of the same dimensions as the image, initialized with zeros. We then select certain rows and columns according to equations (2.1) and (2.2), setting their pixel values to 1 (as described in Algorithm 2.1). These values of 1 represent the pixels that will be retained in the new image, while the zeros indicate the regions to be cropped (as described in Algorithm 2.2).

To create a new image, the given image is first converted into a pixel matrix, which is then processed alongside the corresponding filter. Pixels matching positions with a value of 1 in the filter are selected and preserved in the new image matrix. The final step involves converting this matrix back into an image format for storage.

As illustrated in Figure 2.1, after calculating μ_1 and μ_2, the pixels to be retained and transformed into 1 to create the new image are displayed in green, while the pixels to be removed and marked as 0 are displayed in red.

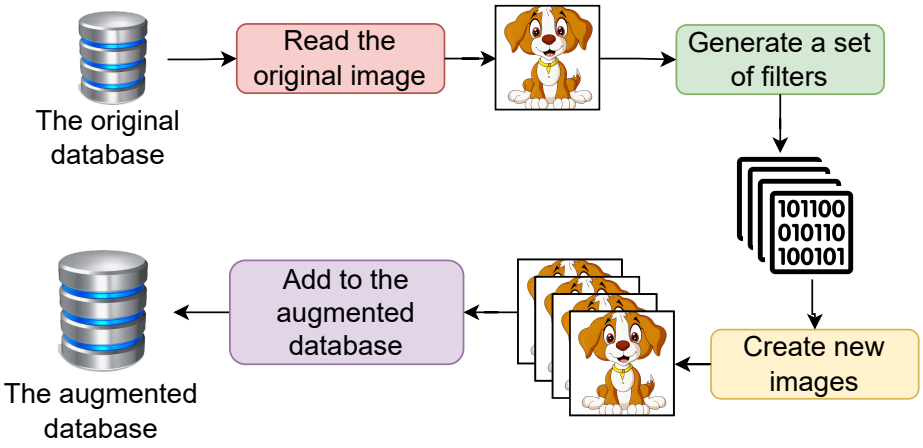


Figure 2.2: Outline of the RS Method.

Each filter generates a new image, which is then stored in the database. These newly created images, along with the original training images, will be used to train the CNN. The process of the proposed method is depicted in Figure 2.2. It is important to note that during the testing phase, only the original images are used for validation.

The number of selected pixels from the original image determines the resolution of the newly created image. We define the following terms:

- n represents the number of rows, and m the number of columns in the original image.
- L_i and C_j are the indices of the rows and columns in the original image, respectively.
- F is an $n \times m$ matrix, referred to as a filter, where all values are initially set to 1.
- μ_1 and μ_2 denote the number of rows and columns in the target image, as calculated by equations (2.1) and (2.2).
- P_1 and P_2 represent the probability of selecting a particular row or column.
- N indicates the resolution of the target image.

Therefore:

$$\mu_1 = \lceil P_1 n \rceil \tag{2.1}$$

$$\mu_2 = \lceil P_2 m \rceil \tag{2.2}$$

$$N = \mu_1 \times \mu_2 \tag{2.3}$$

Let X and Y be discrete random variables. The probability functions $f(X)$ and $f(Y)$ can be expressed as follows:

$$f(L_i) = P_1(X = L_i) \tag{2.4}$$

and:

$$f(C_j) = P_2(Y = C_j) \tag{2.5}$$

Such as:

$$\sum_{i=1}^{n} f(L_i) = 1, \sum_{i=1}^{m} f(C_i) = 1$$

Therefore:

The values of all pixels corresponding to the selected rows and columns in F are set to 1, indicating the pixels that will be preserved from the original image.

Utilizing the filter F enables the generation of a distinct new image each time from the same original image. The steps for creating a filter for a given image are illustrated in Algorithm 2.1. In Figure 2.3 an illustrative example of the proposed method demonstrates the use of two different filters to generate two new images from the same original image.

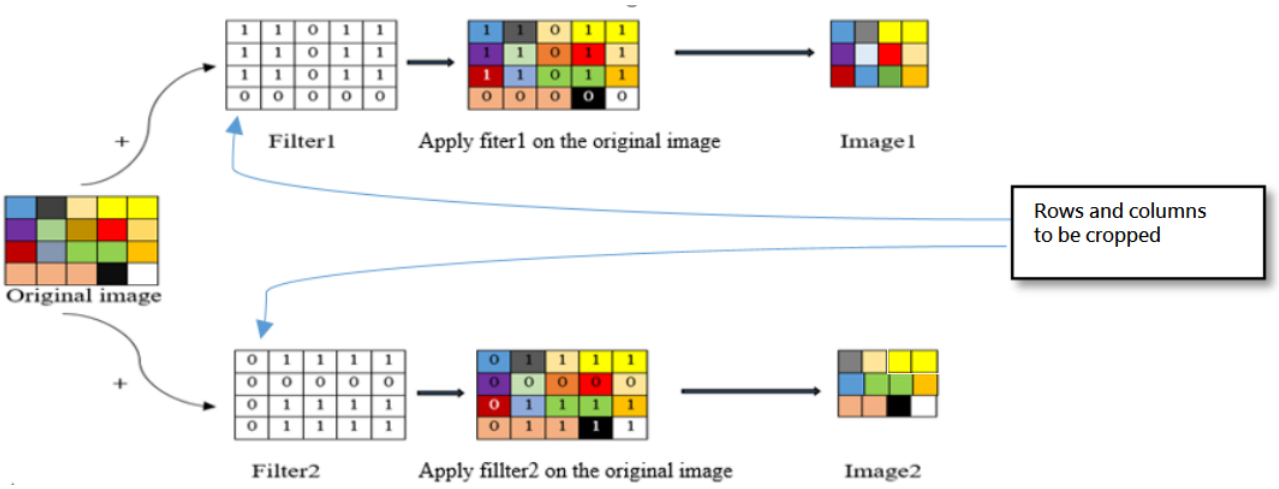


Figure 2.3: An example of the RS method using two filters

Algorithm 2.1 Creating a new filter F

Designate rows L_i and columns C_j using Equation (2.4) and (2.5)
Compute μ_1 as the number of rows in the target image
Compute μ_2 as the number of columns in the target image
Generate a random vector **VRn** containing indices of selected rows
Generate a random vector **VCn** containing indices of selected columns
for $i = 1 \rightarrow \mu_1$ **do**
  $x =$ **VRn**$[i]$ {x is the row index of the selected pixel}
  for $j = 1 \rightarrow \mu_2$ **do**
    $y =$ **VCn**$[j]$ {y is the column index of the selected pixel}
    $F[x][y] = 1$ {Select the pixel to preserve}
  end for
end for

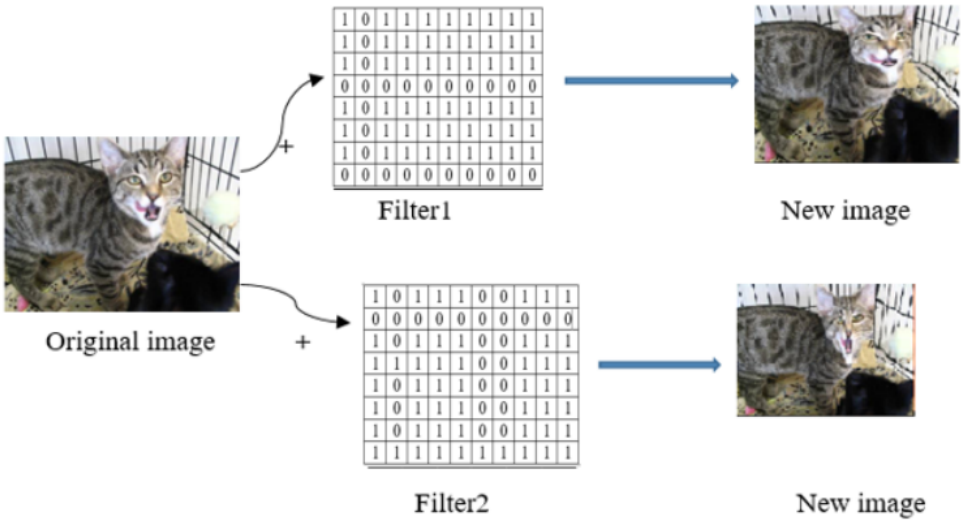


Figure 2.5: wo images with different sizes and appearances were generated using distinct filters.

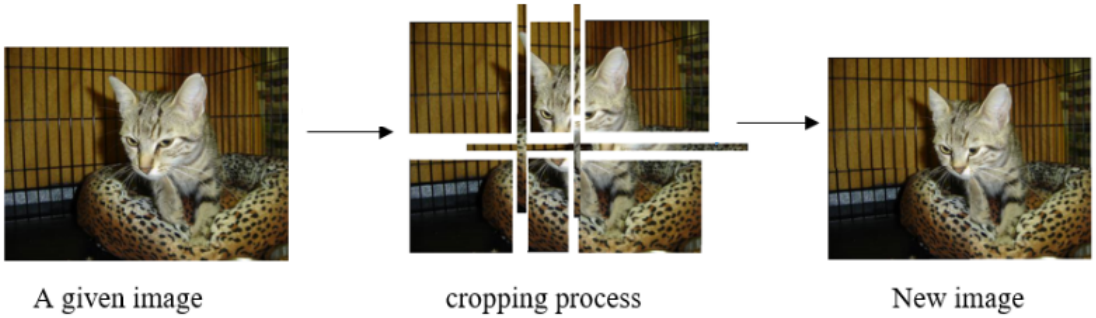


Figure 2.4: Illustration of cropping lines and columns to generate a new image.

By selecting various indices and different numbers of rows and columns using a randomly generated filter in each iteration (with each filter being unique), we can diversify the resulting images.

Algorithm 2.2 outlines the process of generating new images by applying various filters. Figure 2.4 provides an example of cropping rows and columns to create a new image. Figure 2.5 demonstrates that the newly generated images vary due to the application of different filters.

Algorithm 2.2 Generation of new images using a given filter F.

```
Open the image as an array
Retrieve the dimensions of the image, n and m
Generate a new filter using Algorithm 1
Initialize k = 0
for i = 1 to n do
  Set h = 0 /* k and h represent the indices of rows and columns in the new image, respectively.*/
  for j = 1 to m do
    if F[i][j] == 1 then
      newImage[k][h] = image[i][j] /* Retain the same pixel from the original image*/
      h = h + 1
    end if
  end for
  k = k + 1
end for
Save newImage /*Add the new image to the original dataset*/
```

2.3 Results and Discussion

In this section, we employed two versions of the dataset: the original dataset, referred to as $Orig-Db$, and the augmented dataset created using my proposed method, designated as $RS-Db$.

In this experiment, we carried out two tests using different numbers of examples from the same database, each with two distinct models. First, we employed a subset of the Kaggle Cats vs. Dogs dataset, as outlined in Subsection 1.6.1. From the original dataset of 24,000 samples, we selected a total of 10,000, which included 8,000 for training (4,000 images of cats and 4,000 images of dogs) and 2,000 for testing (1,000 images of cats and 1,000 images of dogs). This subset was utilized to train the model detailed in Subsection 1.7.1.

Table 2.1 presents the results obtained from training the Basic CNN mentioned above, comparing the performance using both the $RS-Db$ and the $Orig-Db$. The two versions

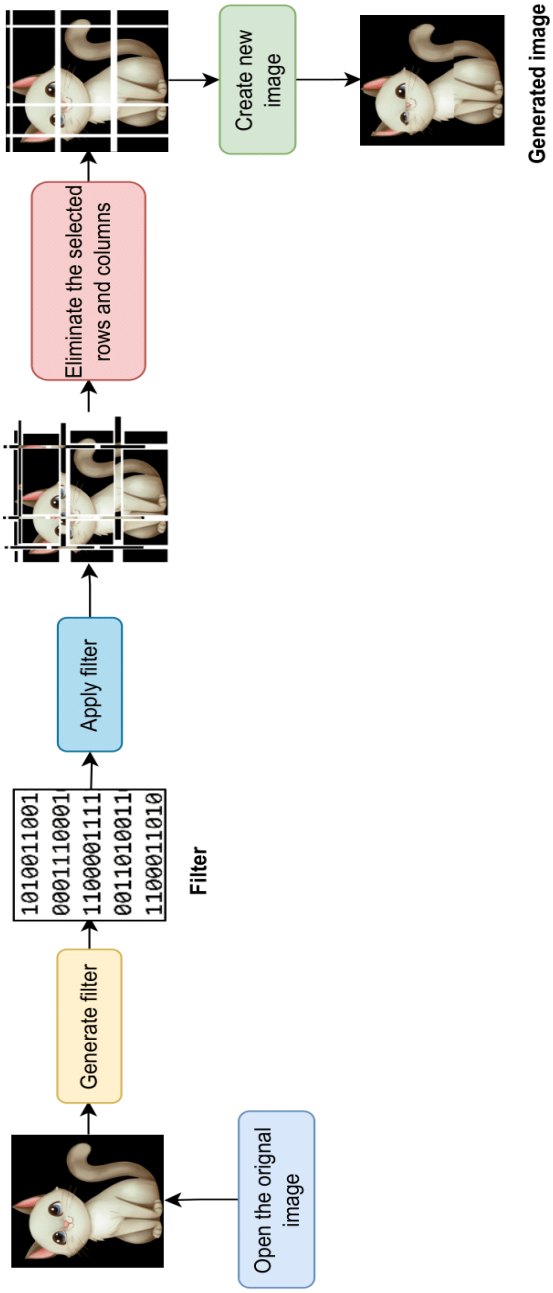


Figure 2.6: The process of generating a new image applying a filter F using RS.

Table 2.1: Results of training the Basic CNN on two versions of the Cats Vs Dogs dataset, using different numbers of examples.

Dataset	Dataset size	Accuracy	Error
$Orig - Db$	X	73.65%	1.5937
$RS - Db$	3X	75.35%	1.6027
$RS - Db$	5X	77.05%	1.6072

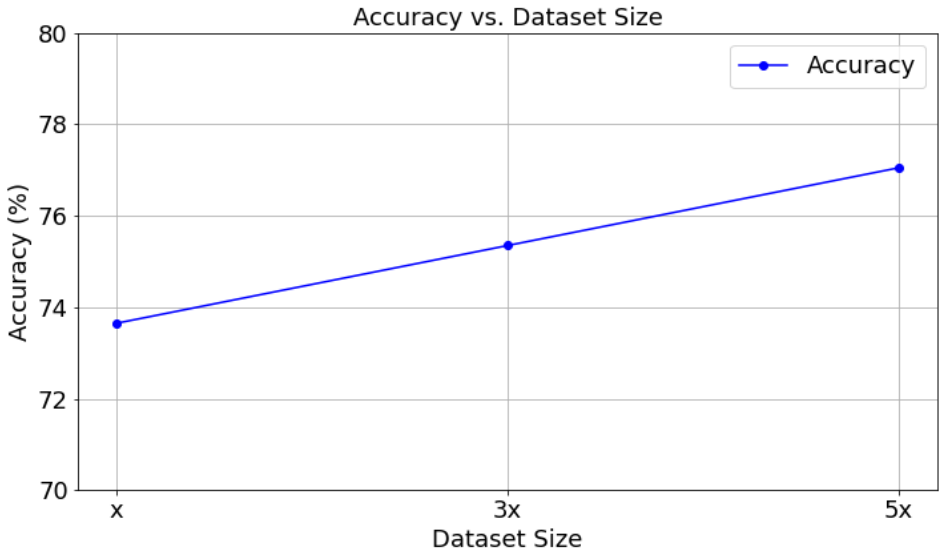


Figure 2.7: Rise in accuracy with an increase in training data size.

of the datasets utilized for training are defined as follows: $Orig - Db$ for the original dataset and $RS - Db$ for the dataset augmented using the RS method. Based on the results obtained, we observe that increasing the size of the $RS - Db$ by three times (from the original size X) led to a 1.7% increase in accuracy (from 73.65% with the $Orig - Db$to 75.35% with the RS-Db). The error increase was minimal, at just 0.0135%. In contrast, when the dataset size was equal to X, the accuracy improved by 3.4% (from 73.65% with the $Orig - Db$ to 77.05% with the $RS - Db$ when increased to five times), while the error rose by less than 0.009%. This indicates that the RS method effectively identifies the most significant areas of the images (representative regions) by selecting a subset of pixels from the original image.

As illustrated in the curves Figures 2.7 and 2.8, accuracy increases significantly with

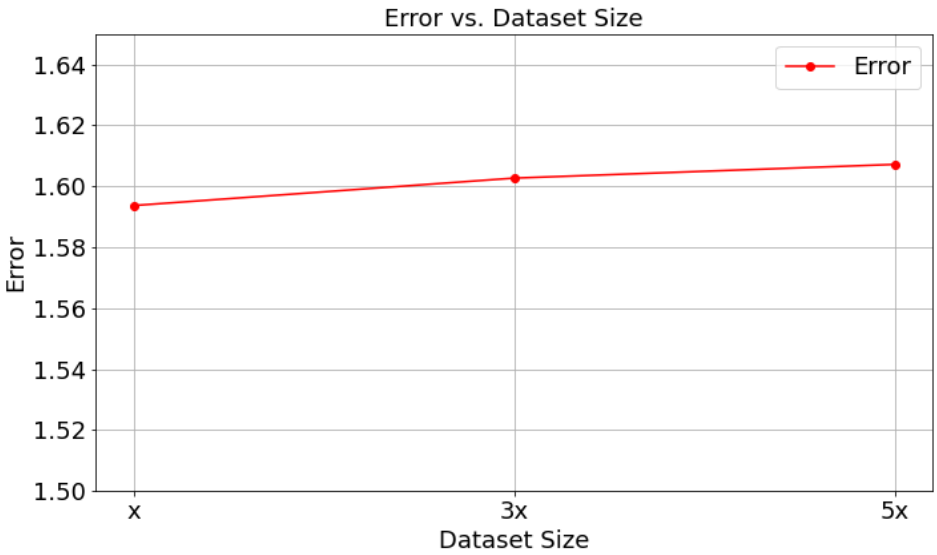


Figure 2.8: Rise in error with an increase in training data size.

Table 2.2: Accuracy obtained from training the ResNet50 model over 30 epochs using the two versions of the Cats vs. Dogs dataset.

Dataset	Dataset size	Accuracy	Error
$Orig - Db$	X	85.68%	0.6834
$RS - Db$	4X	88.93%	0.3479

the size of the augmented data, while the error rises only gradually. This demonstrates the effectiveness of the proposed method for DA. We further assessed our approach by training the ResNet-50 Convolutional Neural Network, as outlined in Subsection 1.7.2. This training utilized both the original Cats vs. Dogs dataset, described in Section 1.6 of the same chapter, and the augmented dataset created using the RS method.

Table 2.2 presents the results obtained from training the ResNet-50 using both the $Orig - Db$ and the RS-Db. The obtained results indicate that when the $RS - Db$ was increased factor of 4 (4X, X is the size of the original dataset), accuracy improved by 3.25%, rising from 85.68% with the $Orig - Db$ to 88.93% with the RS-Db. Additionally, the error decreased by 0.3355%. These results demonstrate that the RS method generates diverse images that differ from one another by selecting a subset of pixels from the original image, thereby enriching the original dataset.

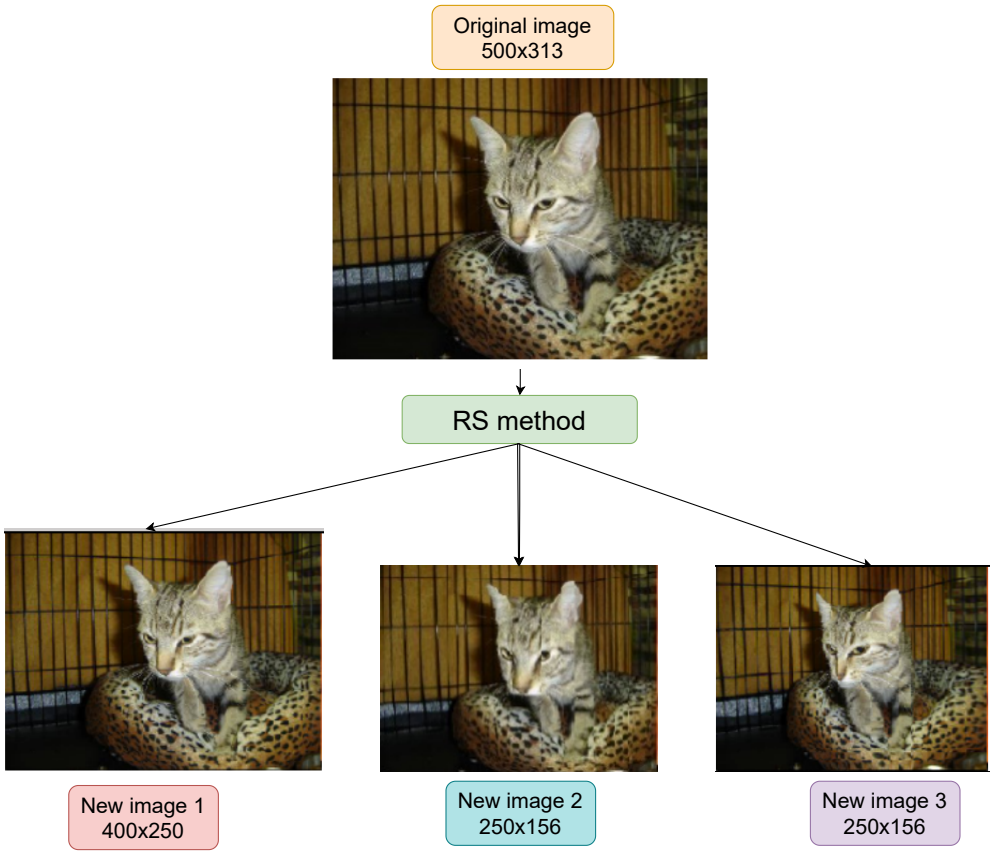


Figure 2.9: Example of newly obtained images by varying the number of selected pixels.

To demonstrate that the filter allows for image variation even when the output images have the same resolution, we applied our method to an image from the utilized dataset. Figure 2.9 displays three images generated from the original image (with a resolution of 500 x 313 pixels) using three different filters. Each image is distinct from the original and from one another. New Image 1 (with a resolution of 400 x 250 pixels) differs in resolution from the other two (New Image 2 and New Image 3), while New Image 2 and New Image 3 have the same resolution (with a resolution of 250 x 156 pixels) but differ in appearance."

Challenges of the proposed method The randomness involved in generating fil-

ters, particularly when selecting the number and positions of pixels to retain for creating new images, can result in new images that are too similar to the original. This lack of variation stems from the possibility that the random process may frequently preserve pixel arrangements that closely resemble the original image, thus limiting the potential for meaningful alterations. As a consequence, the augmented dataset may not exhibit the desired level of diversity, which is crucial for improving model robustness and reducing overfitting.

2.4 Conclusion

In this chapter, we developed a new method based on selecting a set of pixels from the original image to create a new, smaller image using filters. The experiments we conducted yielded promising results. However, the random selection of pixels in the filters can sometimes generate images that are too similar or nearly identical to the original ones, which limits the diversity of the dataset. Consequently, this reduces the potential improvement in model performance.

Chapter 3

Random Optimization and Entropy-Based DA for Image Classification and Analysis "ROEDA"

3.1 Introduction

In Chapter 2, we proposed a method based on the random generation of filters, allowing for the selection of a set of pixels within an image to create new images. However, this approach may generate images that are too similar to the original, due to the random generation of numbers and the selection of pixel positions to preserve in the filter, thus limiting the diversity of the augmented dataset. To overcome this limitation, in this chapter, we introduce RO, a class of algorithms that leverage randomness in their search mechanisms to solve complex problems. Our study focuses on selecting the most distinct image from those generated by our previous method [75], relying on entropy to quantify the content dissimilarity between the generated images and the original one, thereby enhancing the diversity and richness of the training dataset. Our approach using RO, like other optimization techniques, relies on iterative processes and a fitness function to guide the search for optimal solutions. In each iteration, a new set of potential solutions is generated and evaluated using the fitness function, which quantifies how well each solution performs according to predefined criteria. The process repeats, gradually refining the solutions over multiple iterations until the best outcome is identified. our approach utilizes the RS method to generate a set of solutions in each iteration, selecting the best one according to the fitness function. Entropy is used to measure the degree of change in the generated images, with the most altered image, relative to the original, being chosen and added to the augmented dataset.

The experimental results demonstrate that the enhancement of the RS method, through the use of RO and the entropy as a criterion for selecting the best images among the generated ones, significantly improves the model's performance.

This chapter is structured into four main sections. Section 3.1 introduces the study, outlining its objectives and scope. Section 3.2 explains the process of RO, while Section 3.4 delves into the concept of entropy. The proposed methods are thoroughly detailed in Section 3.4. Section 3.5 presents the experimental validation, emphasizing the challenges encountered during the implementation of the proposed approach. Finally, Section 3.6

summarizes the key findings and conclusions drawn from this research.

3.2 Random Optimization Method

RO techniques employing random sampling for performing a search in the solution space of hard optimization problems are usually successfully applied to high-dimensional or non-convex or combinatorial problems, where other methods fail, or where objective functions are not differentiable. Using a variety of techniques to balance the exploration-exploitation trade-off in finding optimal or near-optimal solutions, they are global optimization methods used for avoiding local minima. RO is often less computationally intensive compared to exhaustive searches, hence valuable in such fields as engineering, machine learning, and operations research.

RO techniques have emerged recently as a strong tool in image analysis, particularly in tasks related to segmentation, classification, and feature extraction. This is achieved by the use of random strategies for exploring large solution spaces, improving convergence speed and the quality of solutions.

Algorithm 3.3 RO algorithm

Initialize:
Define the objective function $f(x)$ to be optimized (minimized or maximized).
Set an initial solution x_{best} (can be random or pre-defined).
Define the number of iterations N or stopping criterion.
Define the search space.
for $i = 1$ to N **do**
  Generate a random candidate solution x_{new} within the search space.
  Evaluate the objective function $f(x_{\text{new}})$.
  if $f(x_{\text{new}}) > f(x_{\text{best}})$ **then**
    Update $x_{\text{best}} = x_{\text{new}}$.
  end if
end for

return x_{best} as the optimal solution.

Algorithm 3.3 outlines the process of the random selection method. Mohapatra et

al. [76] presents a new variant of the Golden Jackal Optimization algorithm. It would integrate random opposition-based learning into the GJO algorithm to enhance the ability of the GJO algorithm to handle hard optimization problems. Since opposition-based learning involves the evaluation of not only current solutions but also their opposites during the process of optimization, it would be expected to result in fast convergence and better solutions.

3.3 Entropy

Entropy is one of the important concepts related to information theory, thermodynamics, and in many more areas of science and engineering. In information theory, the entropy of a random variable is one that measures the uncertainty or randomness of the random variable. It denotes the amount of information contained in the message or the extent of disorder in a system. Mathematically, entropy $H(X)$ of a discrete random variable X is defined as:

$$H(x) = -\sum_{x} P(x) \cdot \log_2(P(x)) \tag{3.1}$$

where $P(x_i)$ is the probability of occurrence of each possible outcome x_i of the random variable.

Algorithm 3.4 Entropy calculation

1: **Input:** A probability distribution $P = \{p_1, p_2, \ldots, p_n\}$ where p_i is the probability of event i
2: **Output:** Entropy H
3: Initialize entropy $H = 0$
4: **for** each probability p_i in P **do**
5: **if** $p_i > 0$ **then**
6: Update entropy: $H \leftarrow H - p_i \log_2(p_i)$
7: **end if**
8: **end for**
9: **RETURN:** Entropy H

Algorithm 3.4 details the steps involved in calculating entropy, outlining the process of iterating through a probability distribution and applying the entropy formula (Equation (3.1)).

Applications include quantifying the complexity and information content of images,for-which it has become a useful metric in a wide range of applications. Entropy could be used as a means of assessing image quality, where higher entropy would be indicative of detailed and complex images, and lower entropy would suggest simplicity and uniformity. The entropy in texture analysis helps in distinguishing different textures that could assist in classification and segmentation tasks. It also finds application in feature extraction fo retaining important information with reduced dimensionality and in adaptive thresholding, where the local entropy can dynamically adapt thresholds to perform better binarization due to the changes in lighting. These applications show the versatility and importance of entropy in developing and enhancing image processing and analysis techniques.

Sparavigna et al. [77] conveys the uses of entropy concerning image analysis. It explains how the entropy measures are applied to various tasks such as segmentation, enhancement, and classification. This paper identifies the role of entropy in capturing the essential information content of images so that superior analysis and processing methods may be developed.

Espinosa et al. [78] introduces the EspEn graph (EspEn Graph for the Spatial Analysis of Entropy in Images), which is developed to show the spatial distribution of entropy values across image pixels. What is meant here is to further facilitate the understanding of entropy as an image analysis and processing concept, with a graphical graph showing patterns and structures that may not be so apparent with other methods.

3.4 Proposed Method

This research seeks to enhance the RS technique by integrating an RO method. The proposed approach facilitates the selection of the most appropriate image from a set of α

images generated from the original images using the RS method (α present the population size), with the selection criterion based on entropy to ensure optimal choices. This procedure is repeated multiple times, with β representing the number of iterations. Our goal is to increase dataset diversity, thereby improving the model's learning capability. The overall process of our newly proposed approach is illustrated in Figure 3.1.

In this study, we utilize entropy as the criterion for selecting the most suitable generated images. Shannon entropy, as illustrated by Equation (3.1) (Section 3.4, provides a quantitative measure of information disorder. After calculating the entropy for both the original and generated images, we analyze the variance between their entropy values. This variance serves as an indicator of dissimilarity between the images, with a greater difference reflecting more significant disparities in their information content. By comparing the entropy values of the two images, we can assess their divergence in terms of randomness, complexity, or information richness. A marked contrast in entropy indicates a substantial difference in the patterns or structures represented in the images. In our study, we specifically selected images that differ from the original to enhance the diversity of the augmented dataset.

The calculation of entropy for each generated image entails analyzing the RGB color channels individually and then combining the results to derive the overall entropy of the image. In the context of digital photographs, probabilities are estimated from the image histogram. The combined histogram of the three color channels creates a 3D histogram, with each axis representing the intensity values for the respective channels. The intensity values of the Red, Green, and Blue channels are denoted as $P(x, y, z)$, which represents the joint probability mass function. The entropy of an RGB image is described by Equation (3.2).

$$H(x, y, z) = -\sum_{x}\sum_{y}\sum_{z} P(x, y, z) \cdot \log_2(P(x, y, z)) \tag{3.2}$$

Here, x, y, and z represent the respective intensities of the Red, Green, and Blue channels, while $P(x, y, z)$ denotes the joint probability mass function for the three chan-

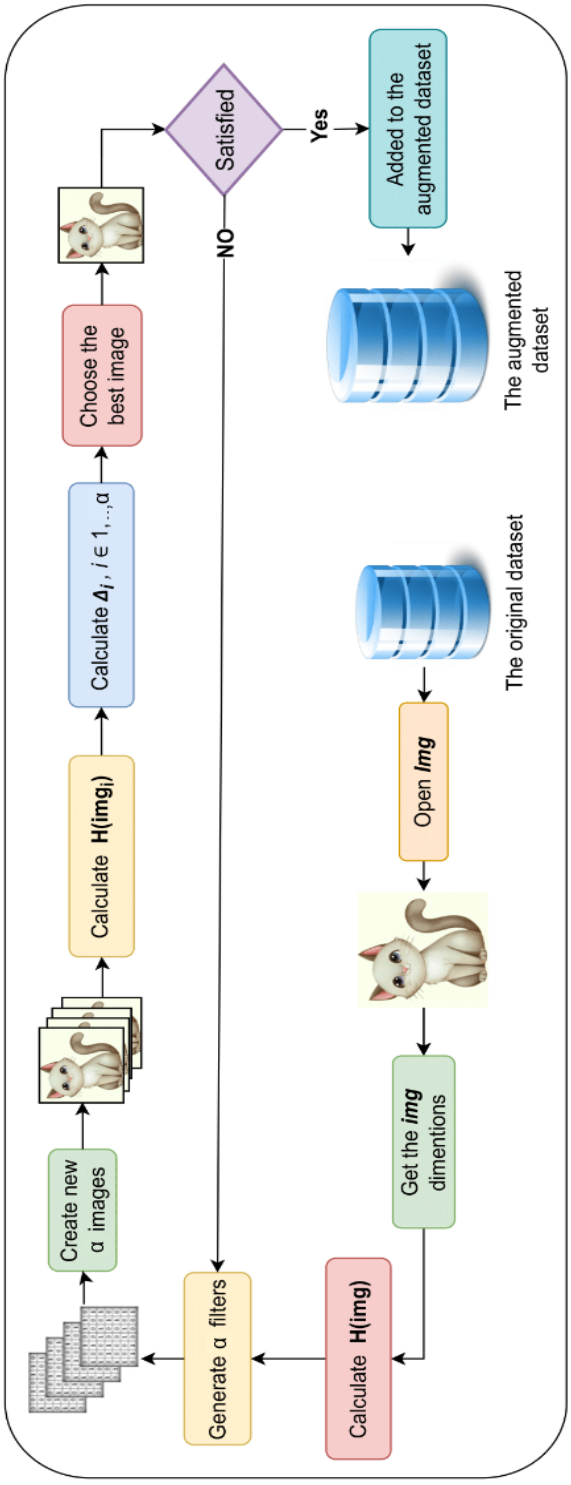


Figure 3.1: Process of ROEDA.

nels. To account for the varying resolutions of the generated images, we normalize the calculated entropy for each new image by dividing it by its corresponding resolution NR, as shown in Equation (3.3).

$$H(x, y, z) = H(x, y, z)/NR \tag{3.3}$$

$$\Delta_i = H(img) - H(img_i), \quad i = 1, \ldots, m \tag{3.4}$$

The procedural steps of our proposed method, illustrated in Figure 3.1, consist of a series of actions. The process begins by opening the original image img to extract its dimensions and calculate its entropy, denoted as $H(img)$. Next, a set of α filters F is generated using Algorithm 2.1 (Chapter 2; Section 2.2), resulting in images $img_i$for$i = 1, ..., \alpha$, created according to Algorithm 2.2 presented in Chapter 2. For each generated image img_i, its entropy is calculated, followed by the computation of Δ_i between img and img_i using Equation (3.4). The optimal image that maximizes Δ is selected from the generated set, and this process of generating filters and creating the corresponding images is repeated until the desired number of images, denoted as β, is achieved. Finally, the selected images are integrated into the augmented dataset. This entire process is performed for each image in the original dataset until all images have been augmented. An example of the resulting images produced by applying the proposed approach is shown in Figure 3.1. Algorithm 3.5 summarizes the steps of ROEDA. Figure 3.1 presents an example of images generated using the ROEDA method.

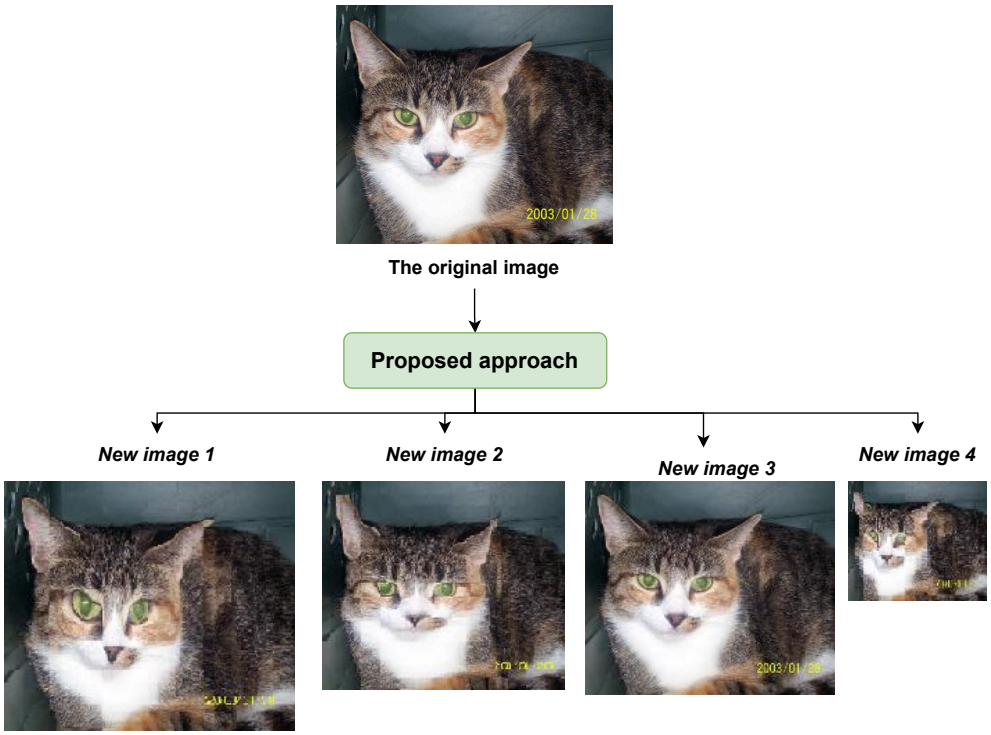


Figure 3.2: Examples of the resulting images generated using the ROEDA method.

3.5 Results and Discussion

In this work, we utilized the Kaggle Cats vs. Dogs dataset, described in Subsection 1.6.1, to train the VGG16 model, described as well in the same chapter (Subsection 1.7.3).

Table 4.5 presents a detailed comparison of VGG16's performance across various versions of the Cats vs. Dogs dataset after training for 30 epochs. The datasets analyzed include the original dataset ($Orig-Db$), the dataset augmented using the random selection method ($RS-Db$), and the dataset augmented with the proposed approach ($ROEDA-Db$).

Using the original dataset, $Orig-Db$ achieved an accuracy of 92.03%, while $RS-Db$ slightly improved this figure to 92.07%. Notably, the $ROEDA-Db$ demonstrated a more

Algorithm 3.5 Generating new images with the ROEDA method.

Input: img, α, β {β represents the number of generations; α denotes the number of images generated in each iteration.}
Open img
Calculate $H(img)$
for $j = 1$ to β **do**
  for $i = 1$ to α **do**
    Generate filter F_i {Refer to Algorithm 2.1}
    Create image img_i
    Calculate $H(img_i)$
    Calculate $\Delta_i = H(img) - H(img_i)$
  end for
  Identify the maximum value of Δ
  Select the optimal image
  Insert img_i into the augmented dataset
end for
Output: The β best generated images

Table 3.1: Comparative analysis of the VGG16 application utilizing the many upgraded versions of Cats vs Dogs.

Dataset version	**Dataset Size**	**Accuracy**
$Orig - Db$	Original Size (x)	92.03%
$RS - Db$	4x	92.07%
$ROEDA - Db$	4x	92.23 %

substantial enhancement, reaching an accuracy of 92.23%. This represents an improvement of 0.16% over $RS - Db$ and 0.20% over $Orig - Db$, highlighting the effectiveness of the proposed augmentation approach.

Tables 3.2, 3.3, and 3.4 indicate that the proposed approach exhibits superior classification performance. Furthermore, it effectively maintains a balanced class distribution while preserving the overall high performance of the classification model for the Cats vs. Dogs task.

Challenges of the Proposed methodology The proposed method using RO and entropy for DA has several limitations. Its effectiveness depends on randomness, which may lead to suboptimal filters if not adequately explored. While aiming to enhance

Table 3.2: The accuracy metrics for training VGG16 were evaluated using $Orig - Db$

Class	**Precision**	**Recall**	**F1-Score**	**Support**
Cat	0.93	0.91	0.92	1249
Dog	0.91	0.93	0.92	1247
Accuracy	0.92	-	-	2496
Macro avg	0.92	0.92	0.92	2496
Weighted avg	0.92	0.92	0.92	2496

Table 3.3: The accuracy metrics for training VGG16 were evaluated using $RS - Db$.

Class	**Precision**	**Recall**	**F1-Score**	**Support**
Cat	0.91	0.95	0.93	1249
Dog	0.95	0.90	0.93	1247
Accuracy	0.93	-	-	2496
Macro avg	0.93	0.93	0.93	2496
Weighted avg	0.93	0.93	0.93	2496

Table 3.4: The accuracy metrics for training VGG16 were evaluated using $ROEDA - Db$.

Class	**Precision**	**Recall**	**F1-Score**	**Support**
Cat	0.91	0.94	0.92	1249
Dog	0.93	0.91	0.92	1247
Accuracy	0.92	-	-	2496
Macro avg	0.92	0.92	0.92	2496
Weighted avg	0.92	0.92	0.92	2496

diversity, the quality of generated synthetic images may not match the original dataset, potentially introducing noise. There is also a risk of overfitting if certain patterns are favored too strongly. Additionally, the method can be sensitive to hyperparameters, complicating implementation, and biases in generated images may not accurately represent the original dataset's diversity. Addressing these limitations is essential for improving its effectiveness in various machine-learning scenarios.

3.6 Conclusion

This work presents a new DA technique that combines RO with entropy-based selection. The proposed method generates a diverse array of synthetic images, significantly expanding the training dataset for machine learning models, especially in computer vision applications. Experimental results demonstrate notable enhancements in classification performance, resulting in improved model accuracy and robustness. These findings underscore the practical advantages of the proposed DA approach, illustrating its potential to optimize model training through the integration of RO and entropy-driven image selection.

Chapter 4

Enhancing Deep Learning Image Classification Using Data Augmentation and Genetic Algorithm-based Optimizations

4.1 Introduction

In Chapter 2, we introduced a novel data augmentation method based on the random generation of filters to create new images. While this approach showed promise, the inherent randomness in filter generation can lead to augmented images that closely resemble the originals, limiting the dataset's diversity. To address this limitation, we enhanced the method by incorporating random optimization based on entropy to select the best images from the generated set as presented in Chapter 3. However, the augmentation's effectiveness still largely depends on the quality of the generated filters, which may result in images that are insufficiently distinct from the originals, potentially reducing the dataset's diversity and negatively impacting model performance.

In this chapter, we use GA to identify the most effective filters used for generating new augmented images. This approach enables us to enrich the dataset with diverse and representative new images. The GA initiates with a set of random filters and refines them through genetic operators, ultimately identifying the optimal filters that produce maximally distinct images. The key contributions of our approach include (i) The identification of significant pixels within the original image to be preserved. This process enhances the deep-learning model's accuracy when applied to the generated images. In contrast to the conventional method presented by Nouara et al. [79] presented in chapter 2, which randomly removes rows and columns from the original image (Subsection 4.2.1), potentially yielding both improved and degraded results, our newly proposed approach employs a GA (Subsection 4.2.2) to systematically identify images with non-contiguous deletions of rows and columns, increasing the likelihood of obtaining the best possible images. (ii) Merge more contextually meaningful images extracted from the original image to create the most significant composite (Subsection 4.2.3). This merging process is achieved by utilizing the GA crossover operator.

This chapter is structured into four sections. Section 4.1 offers an introduction to the topic. Section 4.2 provides an in-depth explanation of the proposed methods. In

Section 4.3, the focus shifts to the experimental validation, emphasizing the challenges encountered with the proposed approach. Lastly, Section 4.4 concludes the chapter by summarizing the key findings of the study.

4.2 Proposed Method

Our novel approach consists of two key stages. The first stage utilizes the RS method proposed in [75] and detailed in Chapter 2 to create a diverse set of initial filters using Algorithm 2.1. This method is designed to generate filters that capture a wide range of features from the input data. In the second stage, a GA is employed to refine these filters, selecting the optimal ones that promote a high level of diversity. This optimization process enhances the quality of the filters by prioritizing those that most effectively contribute to the variation within the dataset. Finally, synthetic images are generated using the selected filters, which are then integrated into the training dataset, thereby improving the robustness and accuracy of the machine-learning models. Algorithm 4.6 outlines the key steps involved in the GA process, detailing its sequence of operations for optimization.

Algorithm 4.6 Genetic algorithm

```
Initialize population P with random solutions
Evaluate fitness for each individual in P
while termination condition not met do
    Select parents from P based on fitness
    Perform crossover to create offspring
    Apply mutation to offspring
    Evaluate fitness of offspring
    Replace P with offspring
end while
return the best solution found
```

4.2.1 Generate filters

The selection of rows and columns to be included or excluded is accomplished through the use of filters. A filter is defined as a matrix that matches the dimensions of the original image, consisting solely of binary values: 0 and 1. Here, a value of 0 indicates the pixels to be discarded, while a value of 1 indicates the pixels to be retained from the original image. The total count of 1s in the filter determines the resolution of the new image to be generated, with this resolution being randomly predetermined before filter creation. Initially, we generate a set of filters for each original image, where each filter produces a unique new image.

The random selection of pixels to preserve, along with the stochastic positioning of the value of 1 within the filter, facilitates the generation of a diverse array of images.

To create a new image using a specific filter, we iterate through both the filter and the original image, retaining the original image pixels that correspond to the value 1 in the filter (see Algorithm 2.2 ,2, Section 2.2). The resulting image is then incorporated into the augmented dataset for training the deep learning model.

Conversely, the current approach advances the RS method by incorporating a GA to identify optimal and meaningful images from a broad range of potentially generated options. Furthermore, instead of indiscriminately removing all pixels within the selected rows or columns, this method selectively eliminates specific elements while retaining critical information contained within designated regions. Figure 4.1 illustrates the overall framework of the proposed approach.

As illustrated in Figure 4.1, our approach begins with the application of the RS method to generate an initial population of filters for each image in the original training dataset. These filters are then converted into vector form for compatibility with the GA. The primary goal of employing the GA is to identify the most optimal filters that prevent the removal of continuous segments of pixels represented by zeros. This is essential to ensure that we do not eliminate contiguous areas that may contain significant features within the image while maintaining distinctiveness among the selected filters.

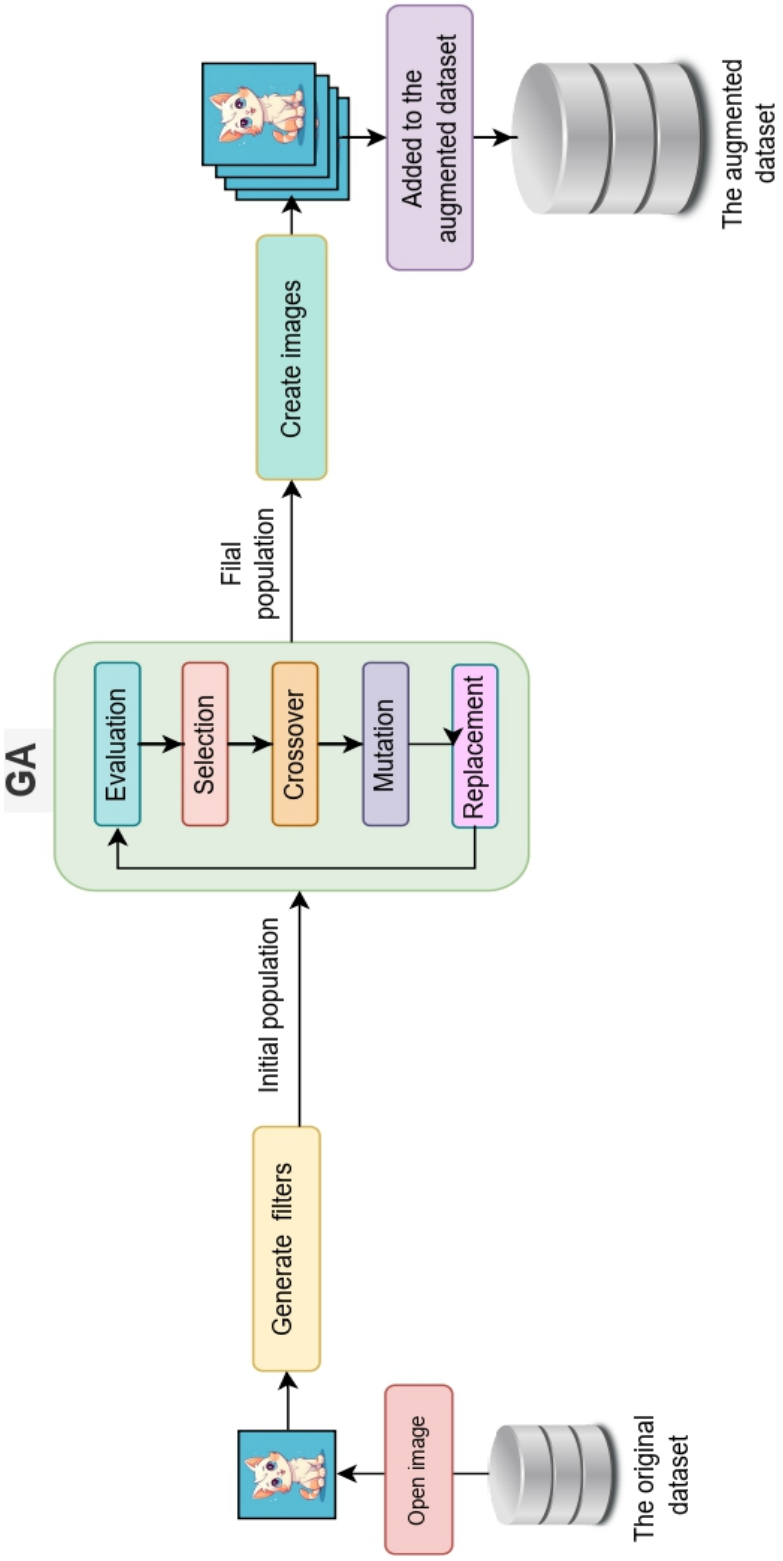


Figure 4.1: Overview of the methodology for the proposed DA technique

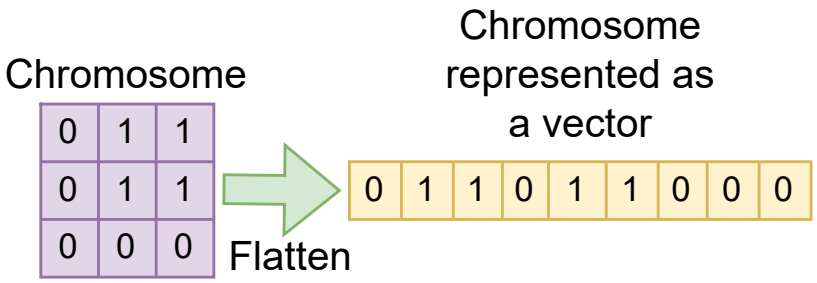


Figure 4.2: A simplified example illustrating the representation of a filter as a chromosome in vector form.

Through a series of iterative generations, we create individual images based on each filter in the final population. These newly generated images are subsequently incorporated into the original dataset, enhancing the diversity and richness of the training data for deep learning models. In the following sections, we will provide a comprehensive overview of our proposed approach.

4.2.2 Application of GA

Chromosomes

The filter, consisting of n rows and m columns, is converted into a chromosome represented as a vector of size $R = n \times m$, as illustrated in Figure 4.2.

Initial population

GA is applied to each image in the original dataset, starting with an initial population composed of a collection of filters. The creation of these filters is executed according to Algorithm 2.1 (2, Section 2.2), a key contribution of our work presented in Chapter 2. Each filter is represented as a matrix of the same dimensions as the original image to be augmented, consisting of rows and columns filled with 1 and 0. The primary objective is to selectively choose pixels from the original image to construct a refined version.

In this context, a value of 1 in the filter indicates that the corresponding pixel in the original image is preserved, while a value of 0 signifies its removal. A filter F

with dimensions $n \times m$ is generated randomly based on predetermined probabilities that designate which columns and rows will be retained (1) or eliminated (0). These probabilities determine the specific values and locations of 0 and 1 within the filter. The total count of retained pixels within the filter F defines the new resolution of the generated image, as calculated using Equations (2.3) (Chapter 2; Section 2.2).

The process of filter creation is detailed in Algorithm 2.1, as presented in Chapter 2. This algorithm specifies how the positions of the selected within the filter are determined through a random generation process, ensuring variability and diversity in the resulting filters.

Algorithm 2.1 (Chapter 2, Section 2.2) outlines the filter creation process. It describes how the filter's positions are selected through a random generation mechanism, ensuring variability and enhancing the diversity of the filters produced.

Fitness function

The selection of candidates for the subsequent stage of the GA relies on the fitness function. In our methodology, the quality of an individual is evaluated based on the total number of 1s and 0s, as well as their distribution within the filter. The quantities of 1s and 0s in the filter represent the resolution of the resulting image and play a crucial role in determining its quality. Specifically, a higher count of zeros corresponds to an increased fitness value, while a lower count of ones is associated with a decreased fitness value.

The arrangement of these values within the filter significantly impacts image quality. When the position of a 1 coincides with a cluster of pixels in the original image that conveys meaningful information, the generated image exhibits high quality. Conversely, a misalignment of these positions leads to diminished image quality, establishing a direct correlation between the placement of 1s and 0s in the filter and the resultant image quality.

To quantitatively assess the quality of each individual within a given population, we propose a novel fitness function, as outlined in Equation (4.1).

$$f(x) = \frac{\sum_{i=0}^{R-2} \mathbf{1}_A(x_i, x_{i+1})}{\theta}, \quad \theta = N(R-N) \tag{4.1}$$

Where:

$$\begin{cases} \mathbf{x}_i \quad represents\ a\ gene\ i\ of\ the\ individual\ x \\ \mathbf{N}: represents\ the\ total\ number\ of\ 1 \\ \qquad\qquad in\ the\ individual\ x \\ \mathbf{R}: represents\ the\ dimention\ of\ the\ individual\ x \\ (\mathbf{R}-\mathbf{N}): represents\ the\ total\ number\ of\ 0 \\ \qquad\qquad in\ the\ individual\ x \end{cases}$$

The function defined in Equation (4.2), denoted as f, accepts an element from the set of filters as its input and produces a positive real number ($\mathbb{R}^+$) as its output.

$$\begin{cases} f : \text{Filters} \to \mathbb{R}^+ \\ \quad x \mapsto f(x) \end{cases} \tag{4.2}$$

Significant regions within an image are formed by the juxtaposition and conjunction of pixel sets. When a generated filter contains a contiguous set of zeros at the same positions as important pixels, the removal of these pixels results in the loss of critical information in the generated image. This implies that contiguous zeros in a filter may correspond to vital areas in the original image, and their elimination compromises the integrity of the generated image. Therefore, it is crucial to select filters with fewer contiguous zeros. By minimizing the number of contiguous zeros in the chosen filter, we can effectively minimize the fitness function.

The fitness function $f(x)$, as presented in Equation (4.1), is derived from the indicator function detailed in Equation (4.3). This indicator function operates on a set E and assesses whether any element in E belongs to a subset F of E.

$$\begin{cases} \mathbf{1}_F : \mathbf{E} \to \{0, 1\} \\ \quad x \to \mathbf{1}_F(x) \end{cases} \tag{4.3}$$

The indicator function is equal to 1 inside the set F and 0 outside, as illustrated in Equation (4.3). In our research, the evaluation of an individual is done considering the number of successive zeros. The indicator function will be redefined in Equation (4.4).

$$\begin{cases} \mathbf{1}_A : \mathbf{F} \to \{0, 1\} \\ \quad (x_i, x_{i+1}) \mapsto \mathbf{1}_A(x_i, x_{i+1}) \end{cases} \tag{4.4}$$

Where

- F is the filter
- $A = \{(x_i, x_{i+1}) \mid x_i = x_{i+1} = 0, \quad i = 0, R-2\}$ denotes the set of consecutive gene pairs that have a value of 0.

The total number of 1s in the filter represents the resolution of the resulting image generated when the filter is applied. In a given population, the proposed fitness for each individual is computed using Algorithm 4.7. The indicator function assesses the occurrence of consecutive zeros in the filter. When minimizing the fitness function, a preference is given to high-resolution filters, while maximizing the function prioritizes the selection of filters with lower resolution.

Algorithm 4.7 Calculating the fitness score.

Input: Individual x
Initialize $Sum \leftarrow 0$ {Variable to accumulate the sum of contiguous zeros}
for $i = 1$ to R **do**
  $Sum \leftarrow Sum + \mathbf{1}_A(x_i)$ {Update Sum based on the indicator function}
end for
$f(x) \leftarrow \frac{Sum}{\theta}$ {Calculate fitness based on the sum of contiguous zeros}
Output: $f(x)$

To clarify the proposed fitness function, we generate two populations based on an original image with a resolution of $R = 10$. Population 1 consists of filters, all having the same new resolution of $N = 4$, as illustrated in Figure 4.3. In contrast, Population 2 includes filters with varying resolutions: two filters have a new resolution of $N = 5$, while the other two possess a resolution of $N = 6$, as shown in Figure 4.4.

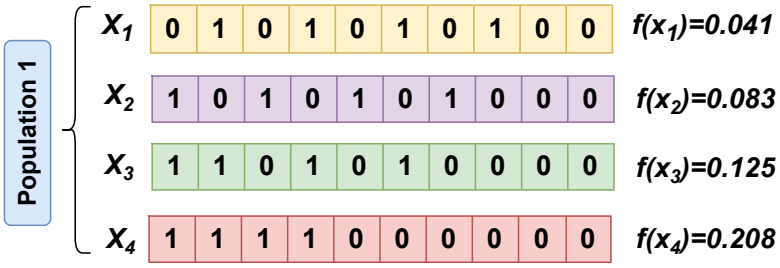


Figure 4.3: Fitness function evaluation with images of uniform resolution.

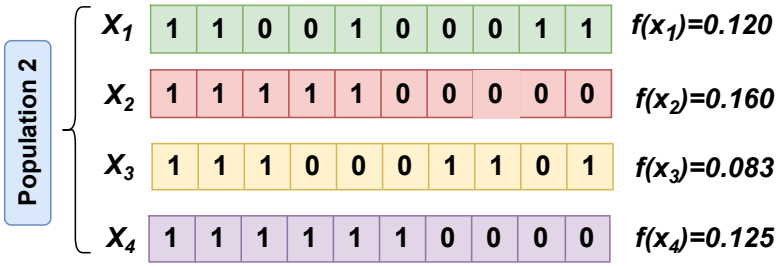


Figure 4.4: Computation of the fitness function with images of varying resolutions.

Table 4.2: Fitness evaluation for individuals with varying resolutions.

x_i	$\sum_{k=0}^{R-1} \mathbf{1}_A(x_k)$	$f(x_i)$	N	$R-N$
x_1	03	0.120	5	5
x_2	04	0.160	5	5
x_3	02	0.083	6	4
x_4	03	0.125	6	4

Table 4.1: Evaluating the fitness function for individuals sharing the same resolution within a population.

x_i	$\sum_{k=0}^{R-1} \mathbf{1}_A(x_k)$	$f(x_i)$	N	$R-N$
x_1	01	0.041	4	6
x_2	02	0.083	4	6
x_3	03	0.125	4	6
x_4	04	0.208	4	6

We computed the objective function for the two populations, with the results presented in Tables 4.1 and 4.2.

Table 4.1 illustrates that both the indicator function and the proposed function increase concurrently as the adjacency of zeros rises, and vice versa. To maintain diversity among the images generated from the original image (Resolution $R = 10$), we generate filters with varying resolutions (see Figure 4.4).

Interestingly, Table 4.2 shows that two filters, x_1 and x_4, with distinct resolutions, exhibit identical values for the indicator function: $1_A(x_1) = 1_A(x_4)$.

This raises the question of which filter to choose: the low-resolution filter x_1 or the high-resolution filter x_4? The proposed fitness function, as detailed in Equation (4.1), normalizes the indicator function using the factor $\theta = N(R-N)$. It accounts for the number of occurrences of both 1s and 0s in each filter, as these vary from one filter to another. In this case, the values of the proposed fitness function for the two filters differ, with $f(x_1) < f(x_4)$. As shown in Table 4.2, filter x_1 has a lower adjacency of zeros compared to filter x_4. Thus, filter x_1 will be selected for the next step.

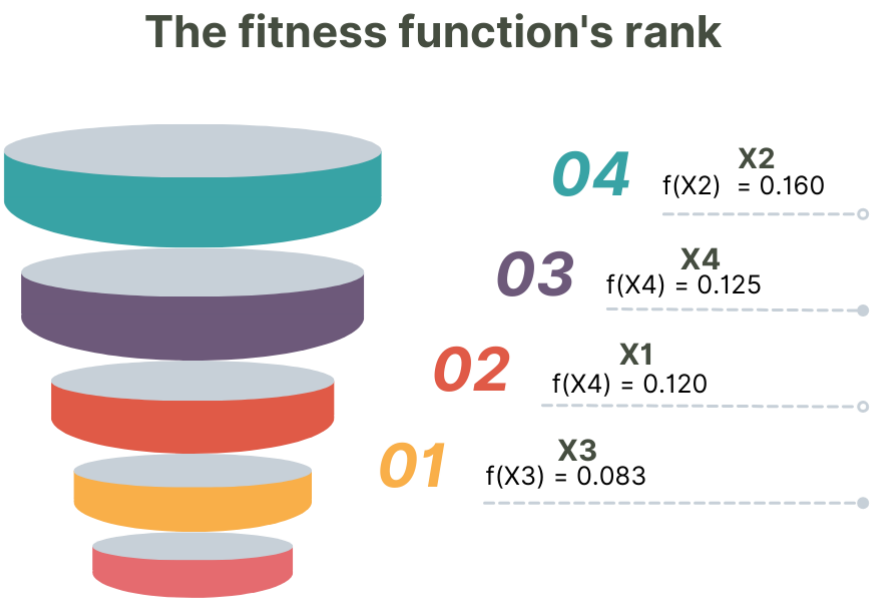


Figure 4.5: Ascending order of fitness values based on the proposed fitness function.

Selection

This step is essential as it allows us to select the fittest individuals for genetic operations. We utilized the ranking technique to identify new individuals within a specific generation (population) of GA. This technique is widely recognized for arranging individuals based on their fitness in either ascending or descending order. In our study, we chose a descending order, meaning that individuals with higher fitness values are more likely to be selected. For instance, in Figure 4.5, individuals x_1 and x_3 are selected from the population. Figure 4.5 demonstrates how the ranking technique organizes individuals based on their respective fitness values.

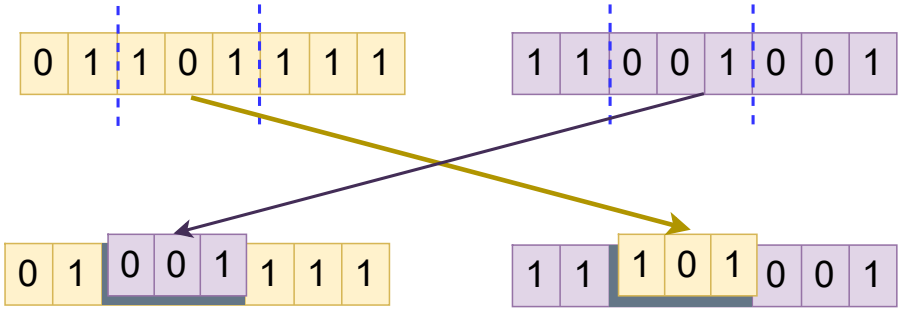


Figure 4.6: Illustration of the crossover operator in action.

Crossover

The crossover operator creates new offspring by combining the genetic information of two-parent individuals according to a specified crossover rate, defined in the range $\in [0, 1]$. In our approach, we employ a two-point crossover operator with a user-defined rate. Two random points, denoted as P_1 and P_2, are generated within the interval $[1, R]$. The procedure for executing the two-point crossover is detailed in Algorithm 4.8.

Algorithm 4.8 Two-Point crossover algorithm

Input: Parents X_1, X_2
Input: Crossover rate r
Generate two random points P_1, P_2 such that $1 \leq P_1 < P_2 \leq R$
rand $\leftarrow$ random number in $[0, 1]$
if rand $\leq r$ **then**
    $O_1 \leftarrow$ Concatenate $X_1[1 : P_1], X_2[P_1 + 1 : P_2], X_1[P_2 + 1 : R]$
    $O_2 \leftarrow$ Concatenate $X_2[1 : P_1], X_1[P_1 + 1 : P_2], X_2[P_2 + 1 : R]$
else
    $O_1 \leftarrow X_1$
    $O_2 \leftarrow X_2$
end if
Output: Offspring O_1, O_2

Figure 4.6 illustrates a simplified example of the crossover operator. This operator produces new offspring with unique resolutions. In Figure 4.7, we demonstrate the crossover of two individuals, x_1 and x_2, at two distinct points. The individuals x_1 and

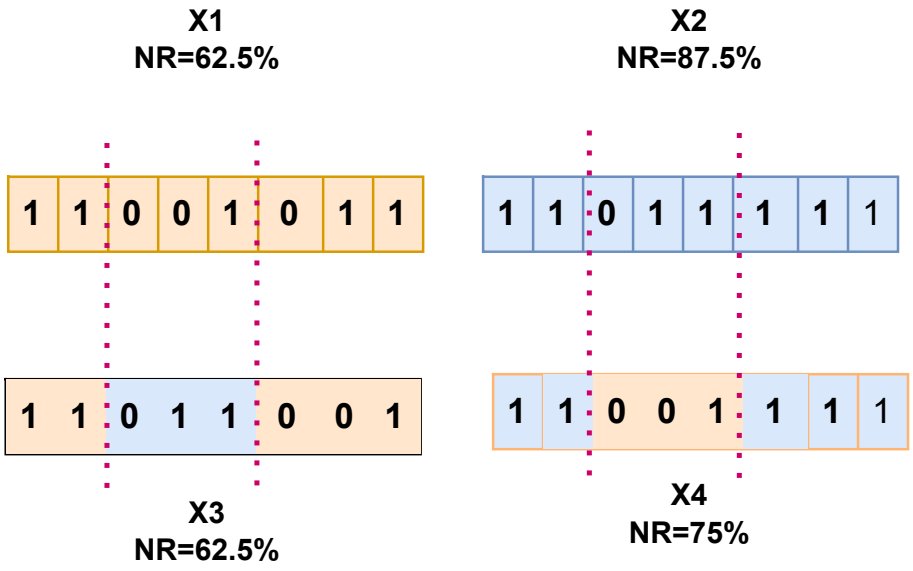


Figure 4.7: Alteration of resolution by the crossover operator.

x_2 correspond to different new resolutions of 62.5% and 87.5%, respectively, indicating the percentage of pixels retained from the original image in the generated images. The resulting offspring, x_3, maintains the same resolution of 62.5% as x_1, but features a different pixel distribution. In contrast, x_4 exhibits a completely new resolution of 57%, which differs from both x_1 and x_2. .

As we can see, this ensures diversity among the resulting individuals, guaranteeing a wide range of newly generated images that are different from each other.

Mutation operator

The mutation operator is a crucial genetic operation that introduces random alterations to one or more genes or chromosomal segments within a population, as detailed in Algorithm 4.9. This operation is essential for maintaining genetic diversity, thereby preventing the GA from becoming trapped in local optima. In our approach, we apply the mutation operator to a single gene, denoted as g where $g \in [1, R]$, altering its value. This process

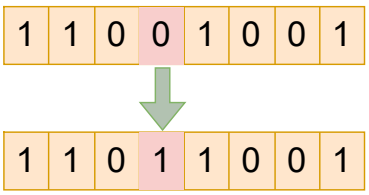


Figure 4.8: Illustration of the mutation operator's effect.

consistently generates new individuals. Figure 4.8 illustrates the effect of the mutation operator.

Algorithm 4.9 Mutation algorithm

Require: Individual X, Mutation rate r_m, Problem constraints
Ensure: Mutated individual X'
  $X' \leftarrow X$ {Initialize the mutated individual as a copy of the original}
  for each gene g in X' **do**
    $p \leftarrow$ random value in [0, 1]
    if $p \leq r_m$ **then**
      $g \leftarrow$ random value within the specified problem constraints
      Update gene g in X'
    end if
  end for
  return X' {Return the mutated individual}

Replacement

This is the final step of GA, where the newly generated population replaces the previous generation. The aforementioned steps are iteratively repeated until the stopping criterion is met.

4.2.3 Image generation

The final population consists of filters that will be utilized to generate new images. The creation of each new image corresponding to a filter in the final population involves simultaneously traversing both the filter and the original image. During this process,

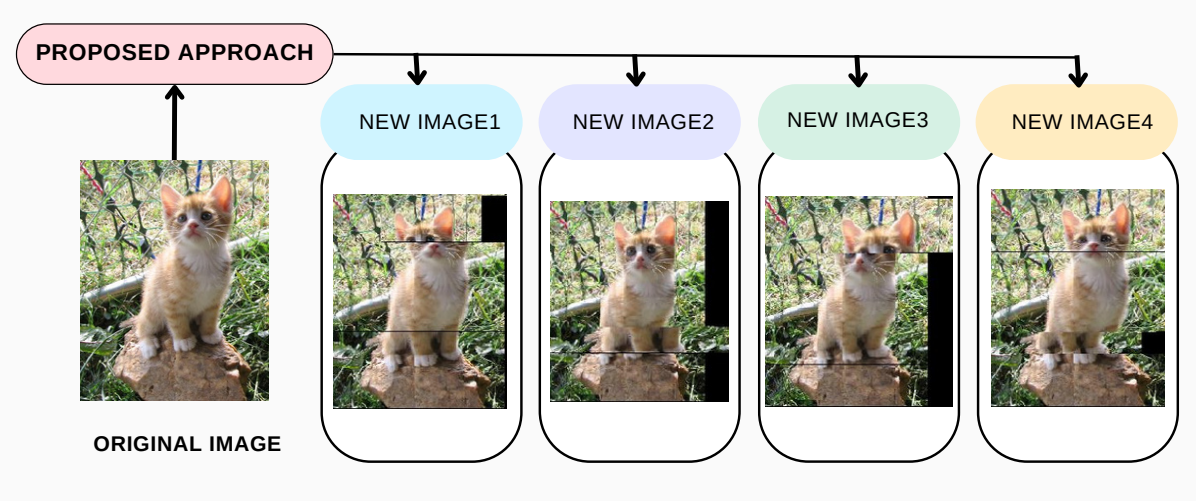


Figure 4.9: Examples of generated images utilizing the proposed approach with the Cats vs. Dogs dataset.

only the pixels of the original image that correspond to a value of 1 in the filter are preserved.

Figures 4.9 and 4.10 showcase new images generated from the final population. These images not only differ from the original but also exhibit variations in size and appearance, highlighting the diversity achieved through the proposed approach.

4.3 Results and Discussion

To demonstrate the effectiveness of our proposed method, we conducted experiments on two datasets: Cats vs. Dogs and Chest X-rays, as described in Section 1.6.

In our work, we tested and compared the training results obtained by applying the pre-trained models presented in Subsection 1.7.3, 1.7.5, 1.7.7, 1.7.6. In this work, as an additional contribution to our research, we refined the architectures of the two models we used, specifically VGG16 and VGG19, as outlined in Section 1.7.4. These adjustments were made after extensive experimentation and have significantly improved the models' performance, leading to better results.

We utilized the original datasets, the datasets generated using the method proposed in

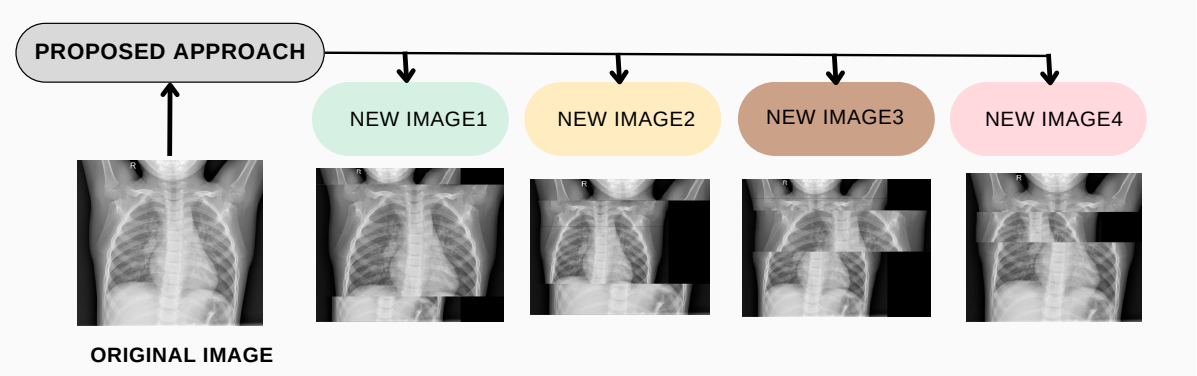


Figure 4.10: Examples of generated images utilizing the proposed approach with the Chest X-ray dataset.

Chapter 2, as well as those generated using our new approach.

The three versions of the datasets used for training are defined as follows: $Orig-Db$ represents the original dataset, $RS-Db$ corresponds to the dataset augmented with the RS method, and $PA-Db$ refers to the dataset enhanced using the proposed approach.

Table 4.3 presents a comparison of training results for three versions of the Cats vs. Dogs dataset—$Orig-Db$, $RS-Db$, and $PA-DB$—using both the original and modified VGG16 and VGG19 architectures. The table includes accuracy and test error rates for each dataset version and architecture.

The modified VGG16 and VGG19 architectures consistently outperform their original counterparts in terms of accuracy across all dataset versions. Notably, for the $PA-DB$ version, the modified VGG16 architecture achieves a 0.52% improvement in accuracy compared to the original VGG16, while the modified VGG19 architecture shows a 0.27% accuracy gain over the original VGG19. However, the test error rates of the modified VGG16 and VGG19 architectures are slightly higher than those of the original versions for the $RS-Db$ and $PA-Db$ versions.

These findings suggest that the modified VGG16 and VGG19 architectures are more effective in classifying images of cats and dogs, particularly for the $PA-Db$ version.

The impact of different DA techniques was further evaluated through additional experiments.

Table 4.3: Comparison of the results achieved through training the three versions of Cats vs. Dogs datasets: $Orig - Db$; $RS - Db$ and $PA - Db$ for each version using the original and the modified VGG16 and VGG19 architectures.

Dataset versions	**Model**	**Accuracy**	**Test errors**
$Orig - Db$	**VGG16**	92.55%	19.78%
$RS - Db$		92.59%	17.87%
$PA - Db$		92.91%	20.11%
$Orig - Db$	**Modified VGG16**	91.71%	21.36%
$RS - Db$		92.63%	21.82%
$PA - Db$		93.47%	18.16%
$Orig - Db$	**VGG19**	91.07%	21.45%
$RS - Db$		91.39%	20.42%
$PA - Db$		92.11%	21.88%
$Orig - Db$	**Modified VGG19**	90.18%	27.18%
$RS - Db$		92.30%	31.01%
$PA - Db$		92.38%	20.52%

During the experiments, a comprehensive 30-epoch training process was applied to each model—VGG16, VGG19, Inception-V3, and EfficientNet-B0—using the datasets mentioned above, while a 50-epoch training period was allocated for the ViT model. Each dataset was augmented using both the RS method and our proposed method. Subsequently, each model was trained on three distinct versions of each dataset.

The comparative analysis presented in Table 4.4 aims to evaluate the effectiveness of our proposed method in preserving essential features across various augmentation levels. This analysis compares model performance on the $RS - Db$ and $PA - Db$, focusing on accuracy and test errors. For VGG16, the $PA - Db$ dataset improves accuracy by 0.84% and reduces test errors by 3.66%. VGG19 shows a slight accuracy increase of 0.08% with $PA - Db$, alongside a significant reduction in test errors by 10.49%. Inception-V3 experiences a minor accuracy improvement of 0.12% with $PA - Db$, but a slight increase in test errors by 2.04%. The ViT model sees a notable accuracy gain of 1.25% and a minor reduction in test errors by 0.15% with $PA - Db$. EfficientNet-B0 benefits

Table 4.4: Comparison of results from training the three versions of the Cats vs. Dogs dataset—$Orig - Db$, $RS - Db$, and $PA - Db$—using the five selected models.

Dataset version	**Model**	**Accuracy**	**Test errors**
$Orig - Db$	**VGG16**	91.71%	21.36%
$RS - Db$		92.63%	21.82%
$PA - Db$		93.47 %(**+0.84**)	18.16%
$Orig - Db$	**VGG19**	90.18%	27.18%
$RS - Db$		92.30%	31.01%
$PA - Db$		92.38% (**+0.08**)	20.52%
$Orig - Db$	**Inception-V3**	97.75%	10.53%
$RS - Db$		97.91%	10.27%
$PA - Db$		98.03% (**+0.12**)	12.31%
$Orig - Db$	**ViT**	72.44%	53.99%
$RS - Db$		77.36%	47.00%
$PA - Db$		78.61% (**+1.25**)	46.85%
$Orig - Db$	**EfficientNet-B0**	96.67%	13.08%
$RS - Db$		97.92%	07.90%
$PA - Db$		98.40% (**+0.48**)	06.70%

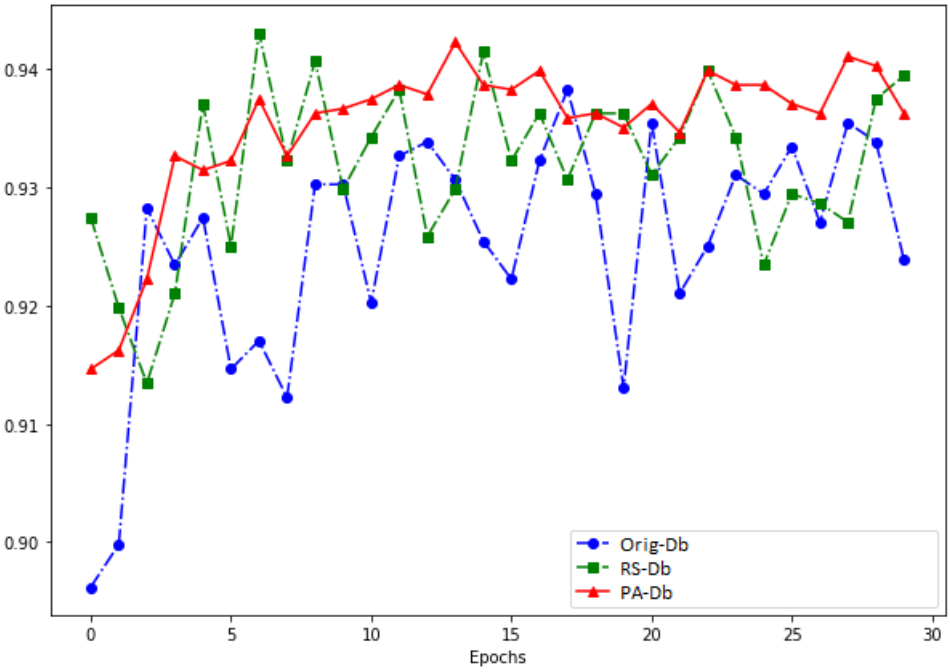


Figure 4.11: The accuracy curves for training the VGG16 model across three versions of the Cats vs. Dogs dataset: $Orig - Db$, $RS - Db$, and $PA - Db$.

from a 0.48% accuracy improvement and a 1.20% reduction in test errors with $PA - Db$. Overall, $PA - Db$ consistently enhances model accuracy and generally reduces test errors, with the most significant improvements observed in the ViT and EfficientNet-B0 models, although Inception-V3 shows a slight increase in test errors.

As illustrated in Figure 4.11, the accuracy curve reflects the performance of the machine learning model across the three versions of the Cats vs. Dogs dataset during training, highlighting the progression of accuracy over time. Notably, the curve for the $PA - Db$ dataset begins to rise from the third epoch, reaching a peak before stabilizing. This curve exhibits greater stability and balance compared to the last two curves, which can be attributed to the higher quality of data utilized during the training process.

These findings underscore the effectiveness of our augmentation strategy, indicating a significant enhancement in the model's performance. The consistent improvement in accuracy from the original dataset to the augmented versions emphasizes the positive impact of augmentation on the model's classification capabilities. Importantly, our

Table 4.5: Comparison of results obtained from applying the VGG16 model on the Cats vs. Dogs dataset and its various augmented versions.

Dataset version	**Dataset size**	**Accuracy**	**Test errors(%)**
$Orig - Db$	Original Size (x)	91.71%	21.36%
$RS - Db$	4x	92.63%	21.82%
	6x	92.95%	19.92%
$PA - Db$	4x	93.47%	18.16
	with the size (4x)	**(+0.84%)**	**(-3.66%)**
	with the size (6x)	**(+0.52%)**	**(-1.76%)**

proposed approach, represented by the $PA - Db$, achieves the highest accuracy when compared to the $RS - Db$ (RS Method), further demonstrating the superior quality of the images generated from the $PA - Db$.

The results summarized in Table 4.5 reveal significant differences among the datasets. The $Orig - Db$ shows a loss function of 21.36%, indicative of higher prediction errors. In contrast, the $RS - Db$ presents a slightly higher loss of 21.82% compared to the $Orig - Db$, suggesting an increase in errors.

In contrast, the $PA - Db$ demonstrates a lower loss function of 18.16%, outperforming both the original and randomly augmented datasets. This indicates a potential enhancement in model performance, characterized by fewer prediction errors. Moreover, the model's training with high-quality images enables it to effectively identify samples that contain significant information.

Moreover, we developed an augmented version of the $RS - Db$ dataset by increasing its size by a factor of 6 (resulting in a dataset size of 6x, assuming the original dataset size is x). We then compared this version with $PA - Db$, which was augmented by a factor of 4. Notably, $PA - Db$, augmented by a factor of 4, achieved an accuracy that is 0.52% higher than the $RS - Db$ augmented by a factor of 6, while also demonstrating a 1.76% reduction in error. This highlights the superiority of our proposed method in attaining higher accuracy with fewer examples, as it effectively selects the most informative images from the generated set.

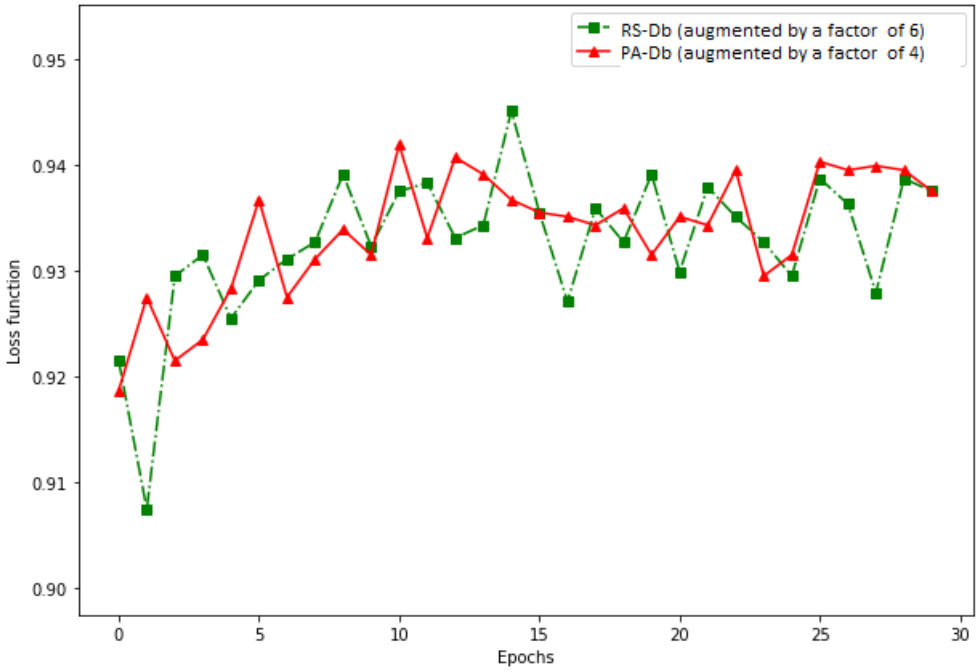


Figure 4.12: The accuracy curves illustrate the training performance of the VGG16 model on two versions of the Cats vs. Dogs dataset: $RS-Db$, which was augmented by a factor of six, and $PA-Db$, augmented by a factor of four.

Table 4.6: Analysis of the confusion matrix derived from training VGG16 with the $Orig - Db$.

	Precision	Recall	F1-Score	Support
Cat	0.88	0.96	0.92	1249
Dog	0.96	0.87	0.91	1247
Accuracy			0.92	2496
Macro Avg	0.92	0.92	0.92	2496
Weighted Avg	0.92	0.92	0.92	2496

Table 4.7: Analysis of the confusion matrix derived from training VGG16 with the $RS - Db$.

	Precision	Recall	F1-Score	Support
Cat	0.92	0.92	0.92	1249
Dog	0.92	0.92	0.92	1247
Accuracy	0.92	0.92	0.92	2496
Macro Avg	0.92	0.92	0.92	2496
Weighted Avg	0.92	0.92	0.92	2496

In Figure 4.12, the accuracy curve emphasizes the outstanding performance of $PA - Db$ augmented by a factor of four. Notably, the accuracy curve for $PA - Db$ exhibits greater stability during training compared to the $RS - Db$ augmented by a factor of six. This difference can be attributed to the superior quality of the data in $PA - DB$, further underscoring the reliability and effectiveness of our proposed method in enhancing model learning.

The results obtained from the confusion matrices, as shown in Tables 4.6, 4.7, and 4.8, consistently exhibit strong performance, with accuracy, precision, recall, and F1-Score

Table 4.8: Analysis of the confusion matrix derived from training VGG16 with the $PA - Db$.

	Precision	Recall	F1-Score	Support
Cat	0.92	0.95	0.93	1249
Dog	0.95	0.91	0.93	1247
Accuracy			0.93	2496
Macro Avg	0.93	0.93	0.93	2496
Weighted Avg	0.93	0.93	0.93	2496

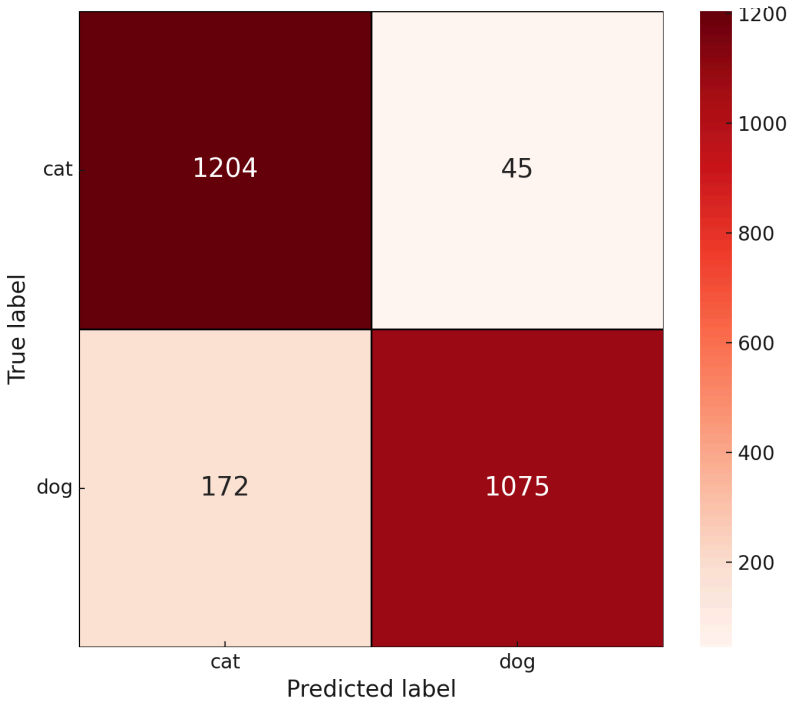


Figure 4.13: Confusion matrix generated from training VGG16 using $Orig - Db$.

values consistently ranging between 0.92% and 0.93%. Notably, Table 4.8 illustrates the performance of the proposed approach, demonstrating slightly enhanced precision, recall, and F1-Score for both Cat and Dog categories compared to the results in Tables 4.6 and 4.7.

The confusion matrices presented in Figure 4.13, Figure 4.14, and Figure 4.15 illustrate the model's classification performance across three dataset versions. Notably, the model exhibited a higher number of classification errors for both categories when trained on the $RS - Db$ compared to the $Orig - Db$. In contrast, utilizing the $PA - Db$ resulted in a reduction of classification errors for both categories relative to the other dataset versions. These errors stem from the inherent challenges posed by the substantial similarity

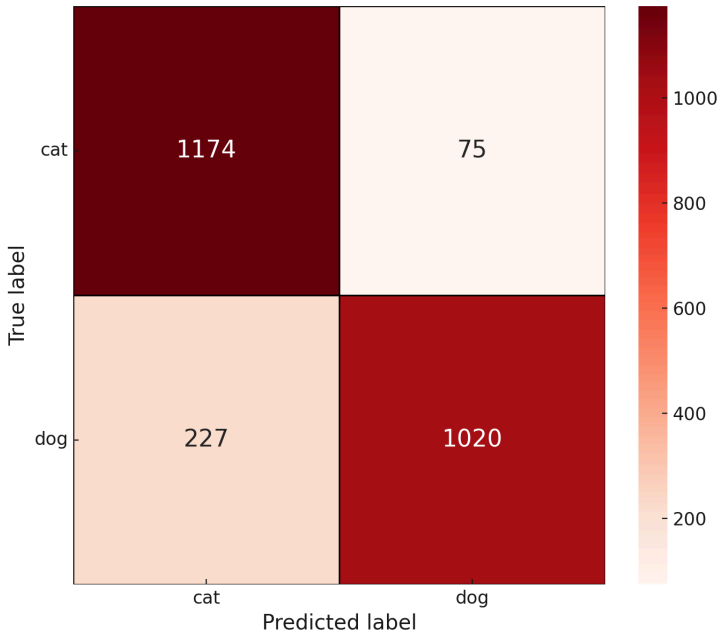


Figure 4.14: Confusion matrix generated from training VGG16 using $RS - Db$.

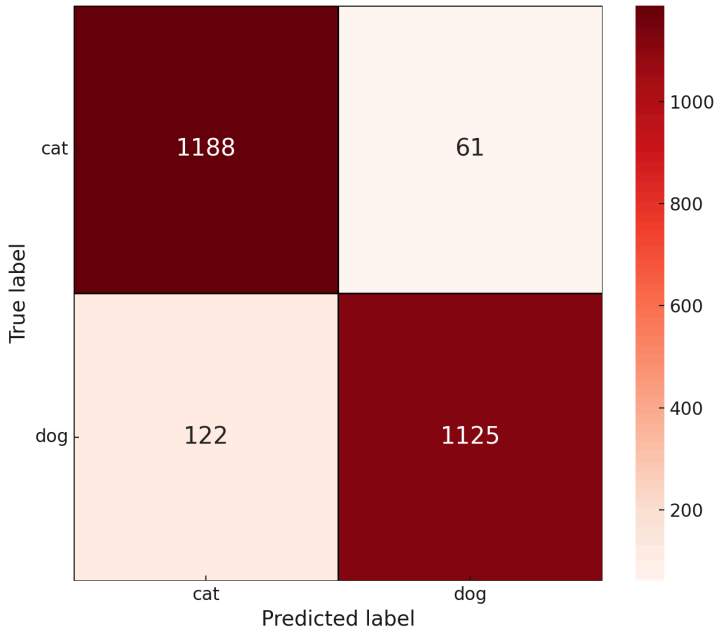
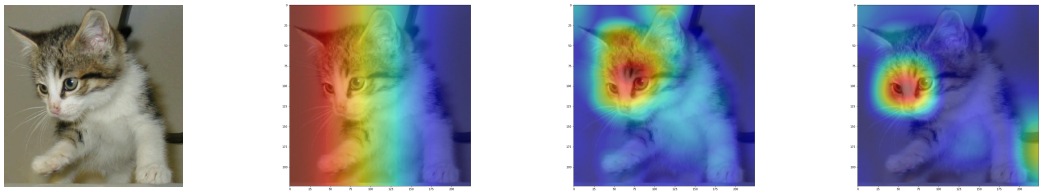


Figure 4.15: Confusion matrix generated from training VGG16 using $PA - Db$.

between the cat and dog classes.



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

Figure 4.16: Heatmaps from EfficientNet-B0 trained on three dataset versions.

In our research, we utilize heat maps to visually depict data distributions and highlight key areas of interest within images from the three versions of the dataset. Heat maps [80] are powerful tools in deep learning, particularly in CNN, as they create visual explanations for image classification predictions. A heat map effectively identifies and emphasizes critical regions within an input image that significantly contribute to predicting a specific class. By employing a color gradient (e.g., transitioning from blue for low importance to red for high importance), the heat map visually represents the intensity of activation across these regions. This method provides valuable insights into the model's specific areas of focus, greatly enhancing the interpretability of its decision-making process. The ability of heat maps to highlight influential image regions makes them indispensable for understanding and validating the reasoning behind CNN predictions.

To further demonstrate the efficacy of our method in generating significant images that identify important classification regions, Figure 4.16 displays three heat maps of the same image (shown in Figure 4.16a), each predicted by EfficientNet-B0 trained with different versions of the Cats vs. Dogs dataset. Figure 4.16b specifically shows the heat map generated by EfficientNet-B0 when trained on $Orig - Db$. This heat map features a cat with a gradient overlay running vertically across the image, transitioning from red on the left edge to violet on the right edge through orange, yellow, green, and blue. Notably, the gradient overlay does not emphasize distinctive cat features such as the

mouth or nose. The heat map in Figure 4.16c highlights significant activation around the cat's face but lacks focus on distinct features. In contrast, the model trained with $PA-Db$ demonstrates strong activation around the mouth and nose regions of the cat's face, indicating that the model focuses on these areas to identify the cat, as shown in Figure 4.16d. This suggests the high quality of the images used for training, as they contain important details.

To underscore the effectiveness of our image augmentation approach, we conducted extensive model training using the three versions of the Cats vs. Dogs datasets ($Orig-Db$, $RS-Db$, and $PA-Db$). When assessing our proposed approach, two additional models (VGG19, Inception V3, Vision Transformer, and EfficientNet-B0) were trained on the different dataset versions. The outcomes of our experiments are summarized in Table 4.9. Remarkably, leveraging the VGG19 model, our approach consistently achieves the highest accuracy of 92.38%, surpassing the randomly augmented dataset by 0.08%, along with a noteworthy reduction in error by 10.49%. Meanwhile, the Inception V3 model attains an accuracy of 98.03%, exceeding the accuracy of the randomly augmented dataset by 0.12%. The EfficientNet-B0 model demonstrates an impressive accuracy of 98.40%, reflecting a 0.48% improvement over the randomly augmented dataset and a significant error reduction of 29.51%. Finally, the Vision Transformer (ViT) model achieves an accuracy of 78.61%, improving by 1.25% compared to the randomly augmented dataset and reducing errors by 0.15%. Collectively, these findings emphasize the robustness and efficacy of our proposed image augmentation strategy.

In this study, we compared the performance of our proposed method against other established techniques. We augmented the Cats vs. Dogs dataset four times using various methods: RS [79], Cutout [41], MixUP [23], CutMix [24], geometric transformations [81], and our approach. Each version of the augmented dataset was then trained separately using the EfficientNet-B0 model for 30 epochs.

The results shown in Table 4.9 and illustrated in Figure 4.17 demonstrate that our method significantly outperforms traditional data augmentation techniques. The superior accuracy and reduced error rate achieved by our approach indicate its effectiveness

in enhancing the model's performance on the Cats vs. Dogs dataset when utilizing EfficientNet-B0. These findings suggest that the innovative strategies employed in our method yield a more robust and generalizable augmentation framework, ultimately leading to improved model training and overall performance.

Table 4.9: Comparison of our method with other approaches utilizing the Cats vs. Dogs dataset and EfficientNet-B0.

Method	**Accuracy**	**Error**
RS	97.92%	07.90%
Cutout	97.86%	08.14%
MixUp	95.99%	19.40%
CutMix	97.80%	08.10%
GT	97.46%	09.47%
Our method	**98.40%**	**06.70%**

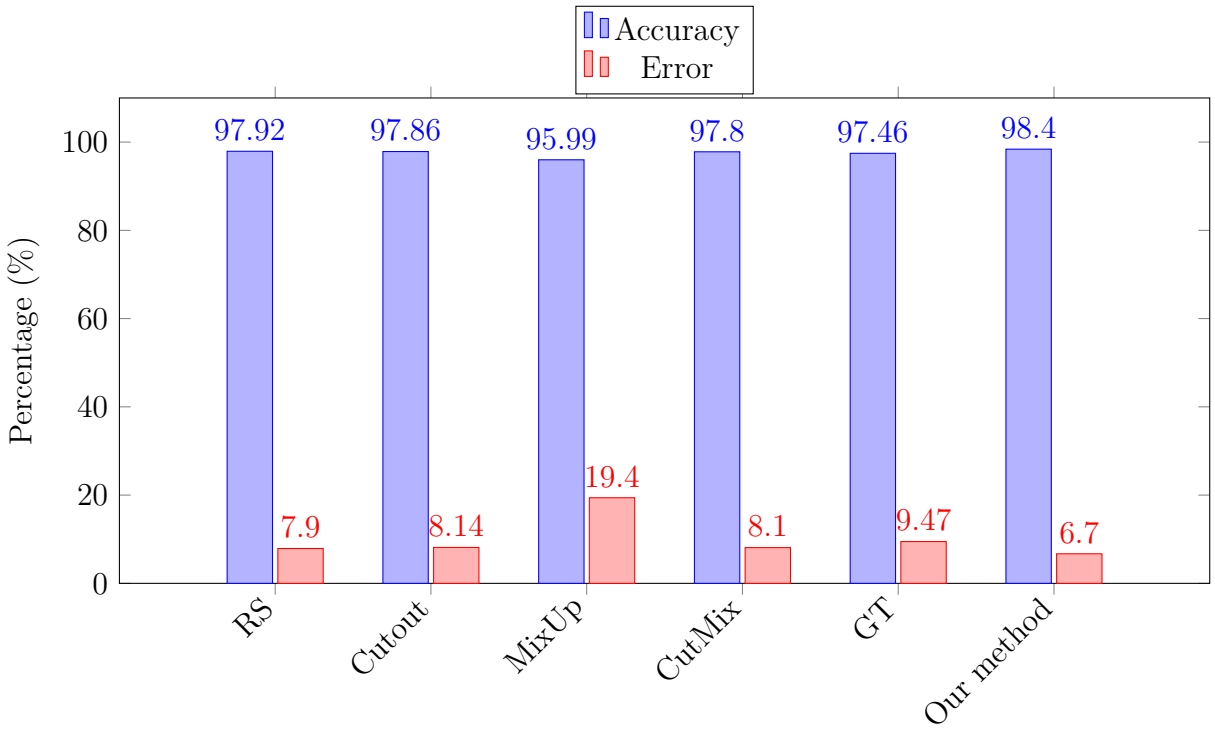


Figure 4.17: Comparison of accuracy and Error for different methods using the Cats vs Dogs dataset and EfficientNet-B0.

Table 4.10: Comparison results were achieved by training the Chest X-ray dataset's three versions: the $Orig - Db$, the $RS - Db$, and the PA Db. Each dataset is trained with various models.

Dataset version	**Model**	**Accuracy**	**Test errors**
$Orig - Db$	**VGG16**	72.44%	1.78%
$RS - Db$		75.00%	1.25%
$PA - Db$		79.81% (+**4.81**)	1.00% (**-0.25**)
$Orig - Db$	**VGG19**	73.40%	215.30%
$RS - Db$		79.17%	176.66%
$PA - Db$		82.05% (+**2.24**)	68.47% (**-108.19**)
$Orig - Db$	**Inception-V3**	89.58%	58.31%
$RS - Db$		89.74%	78.92%
$PA - Db$		90.71% (+**0.970**)	54.92% (**-24.00**)
$Orig - Db$	**EfficientNet-B0**	89.90%	50.95%
$RS - Db$		89.26%	65.88%
$PA - Db$		93.91% (+**4.65**)	36.37% (**-29.51**)
$Orig - Db$	**ViT**	70.05%	63.19%
$RS - Db$		79.48%	48.44%
$PA - Db$		80.89% (+**1.41**)	52.04% (**-03.60**)

Like the Cats vs. Dogs dataset, we applied augmentation techniques to enhance the Chest X-ray dataset, utilizing both the RS method and our proposed approach. Following this preprocessing step, we trained the below models: VGG16, VGG19, Inception V3, EfficientNet-B0, and ViT. The outcomes of these training sessions are detailed in Table 4.10, showcasing the results for each model individually.

Table 4.8 presents a comparative analysis of the results obtained from training VGG16, VGG19, and Inception-V3 on three versions of the Chest X-ray dataset: $Orig - Db$, $RS - Db$, and $PA - Db$. VGG16, when trained with the dataset augmented using the proposed method, achieved the highest accuracy of 79.81%, representing a notable improvement of 4.81% over the RS Method. Additionally, the proposed method reduced test errors to 1.00%, indicating a decrease of 0.25% compared to the RS method.

For VGG19, the proposed approach yielded an accuracy of 82.05%, surpassing both the original and $RS - Db$. Moreover, the method significantly reduced test errors by 108.19%, achieving 68.47%. The Inception-V3 model also benefited from the proposed method, attaining a higher accuracy of 90.71%, with a marginal improvement of 0.97% over the $RS - Db$. Importantly, this approach led to a 24.00% reduction in test errors, bringing them down to 54.92%.

The results for EfficientNet-B0 demonstrate that our method achieved the highest accuracy of 93.91%, exceeding both the original and $RS - Db$ models. Additionally, the proposed approach notably reduced test errors by 29.51%. With the ViT model, our method achieved an accuracy of 80.89%, showing a marginal improvement of 1.41% over $RS - Db$, along with a 3.60% reduction in test errors.

Table 4.10 illustrates that the proposed augmentation method consistently outperforms the RS method across different models, highlighting its effectiveness in enhancing model performance.

Limitations of our methodology While GA is a powerful optimization tool, it presents several challenges due to its stochastic nature, which can result in unpredictable outcomes and complicate the reproducibility of experiments. GA typically requires numerous iterations, leading to high computational costs, and demands careful tuning of genetic operators. Furthermore, they may encounter difficulties in navigating high-dimensional search spaces, increasing the risk of becoming trapped in local optima. Another limitation is that GA often requires a substantial amount of data to perform effectively, which may not always be available. Additionally, the selection of appropriate fitness functions is critical; poorly defined fitness criteria can hinder the algorithm's ability to converge on optimal solutions. Therefore, caution and thorough evaluation are essential when employing GA for data augmentation.

4.4 Conclusion

This work introduces a novel DA method that leverages GA. The results demonstrate the method's robustness and effectiveness in enhancing model learning. By integrating GA into the image generation process, we can strategically select optimal images that contain critical information. This approach optimizes both image selection and augmented data generation, resulting in significant improvements in model performance. Our method emphasizes not only the quantity but also the quality of the generated images.

Chapter 5

Enhancing Image Classification with Ensemble Deep Learning through Deep Feature Concatenation

5.1 Introduction

In the previous three chapters (Chapters 2, 3 and 4), we proposed three data augmentation methods to enhance the model's performance in image classification. In this chapter, we introduce a new deep ensemble learning method that leverages the strengths of existing models to improve classification tasks.

Ensemble deep learning is employed to address the limitations of individual models in complex tasks such as image classification. Single models may struggle with generalization, robustness, and accuracy, particularly in the face of variable data and the risk of overfitting, which can impair consistent performance across diverse scenarios By leveraging the strengths of multiple models, ensemble deep learning enhances overall performance, increases accuracy, and bolsters resilience to errors, making it a more dependable approach for challenging classification problems

Ensemble deep learning integrates multiple neural networks to enhance the overall performance and robustness of models. Various techniques have been developed for this purpose, each offering unique advantages. One widely utilized method is Bagging [82], which aims to reduce variance by combining predictions from multiple models trained on different subsets of the training data. Another prominent approach is Boosting [83], where models are trained sequentially, with each model focusing on correcting the errors of its predecessor. To make final predictions, a secondary model (the meta-model or level-1 model) combines the outputs of several base models (level-0 models) trained on the same dataset. This advanced ensemble technique is referred to as stacking or stacked generalization, which leverages the strengths of diverse models, enabling the meta-model to optimally combine their predictions [84]. Furthermore, the Bag of Tricks method incorporates multiple deep learning models using various techniques to enhance performance. Bayesian ensembles [85] offer an alternative strategy by employing Bayesian methods to sample from the posterior distribution of model parameters, resulting in a robust ensemble. Additionally, multi-head ensembles, where a single network is trained with multiple output heads on different subsets of data or tasks, enrich the landscape of

ensemble learning in deep learning.

Our method utilizes the outputs from multiple CNN models (Subsection 5.3.1). By concatenating these outputs (Subsection 5.3.2), we harness the complementary strengths and perspectives captured by different models (Subsection 5.3.3).

This chapter is organized into five sections. Section 5.1 provides a general introduction to the fundamental concepts of ensemble deep learning. Section 5.2 reviews existing approaches in this field. Section 5.3 offers a detailed explanation of the proposed methods. In Section 5.4, we focus on experimental validation, highlighting the challenges encountered with the proposed approach. Section 5.5 outlines the challenges associated with the proposed method. Finally, Section 5.6 summarizes the key findings of the study and concludes the chapter.

5.2 Related Work

Combining features from diverse models in ensemble deep learning enhances model diversity and performance [86]. This approach leverages the unique strengths of each model, leading to improved overall efficiency and robustness while maintaining the integrity of the original architectures. In this section, we review relevant research on these techniques.

Mungoli et al. [87] propose a method that integrates extracted features at multiple levels within the neural network architecture. They utilize adaptive fusion layers to combine features from various stages of the network, including early, middle, and late layers. However, this approach is hindered by several limitations, including increased computational complexity, uncertain scalability to larger datasets, a heightened risk of overfitting, and implementation challenges.

Dong et al. [88] present a method aimed at enhancing the classification accuracy of White Blood Cells (WBCs) by fusing features extracted from multiple CNN. Their process involves feature fusion across different layers or stages of the CNNs to capture both low-level and high-level information, significantly improving classification performance.

Nonetheless, this method faces limitations such as increased computational demands, reliance on high-quality and abundant data, and challenges in implementation.

Lin et al. [89] propose a feature fusion approach to improve COVID-19 detection by combining features extracted from multiple deep learning models applied to chest X-ray images. This method aims to create a more robust representation, enhancing classification accuracy compared to utilizing features from individual models. However, the effectiveness of this technique may depend on the quality and diversity of the deep learning models employed. If the models lack sufficient diversity or calibration, the fusion may not yield significant performance improvements. Additionally, this approach may require substantial computational resources and careful tuning of fusion strategies to achieve optimal results.

Liao et al. [90] introduce the Coordinate Feature Fusion Network (CFFN), which enhances fine-grained image classification by integrating features from different network layers through a multi-scale feature fusion mechanism. While CFFN improves the model's discriminative power and classification accuracy, it also increases computational complexity, resulting in longer training and inference times. Furthermore, it necessitates careful tuning of the fusion strategy and network parameters, complicating practical implementation.

Feature fusion techniques across different layers of CNNs face several challenges, including increased computational complexity and longer training and inference times. Their scalability to larger datasets is often uncertain, coupled with risks of overfitting and implementation hurdles. In contrast, our proposed method streamlines the fusion process by concentrating solely on combining the final outputs of the CNNs. This approach reduces computational demands by eliminating the need for multi-layer integration, thus circumventing some of the challenges associated with complex feature fusion. By focusing on the final outputs, we mitigate the risks related to overfitting and enhance scalability.

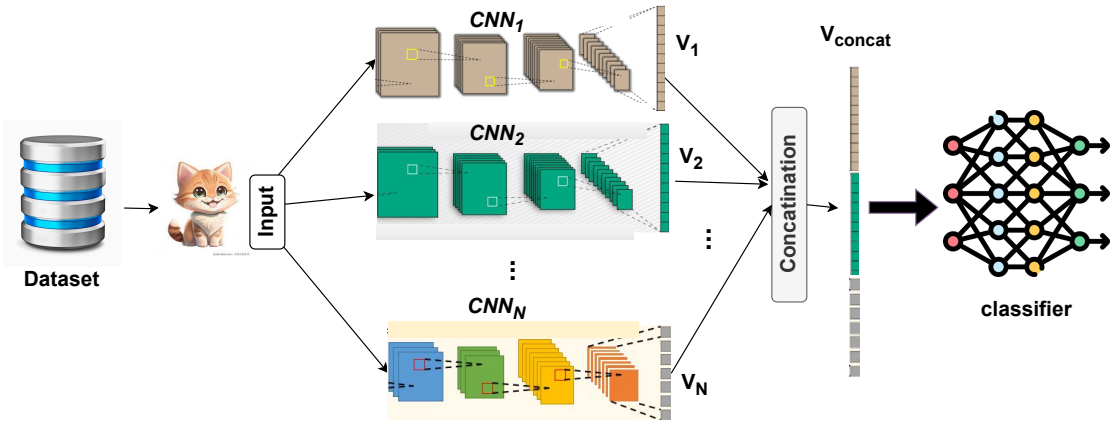


Figure 5.1: Overview of the proposed methodology.

5.3 Proposed Methodology

Initially, we extract feature vectors from multiple CNNs, each specifically designed to capture distinct information from the input data. By leveraging the strengths of these diverse CNN models, our proposed approach presents a novel method for enhancing classification performance. The extracted feature vectors are then concatenated to create a unified representation, which serves as input for a final classifier. This concatenated representation integrates the varied feature sets learned by the different models, enriching the information available for classification and enhancing both the accuracy and robustness of the system. Figure 5.1 illustrates the proposed methodology, outlining the sequential steps of the approach, which are further elaborated in the subsections below.

5.3.1 Feature extraction using CNN

Initially, we extract feature vectors from multiple CNNS, each specifically designed to capture distinct information from the input data. By leveraging the strengths of these diverse CNN models, our proposed approach presents a novel method for enhancing classification performance. The extracted feature vectors are then concatenated to create a unified representation, which serves as input for a final classifier. This concatenated rep-

resentation integrates the varied feature sets learned by the different models, enriching the information available for classification and enhancing both the accuracy and robustness of the system.

$$\mathbf{f}_{i,j} = (\mathbf{W} * \mathbf{X})_{i,j} + b \tag{5.1}$$

where $\mathbf{f}_{i,j}$ is the feature map value at position (i, j), $\mathbf{W}$ is the convolutional filter, $\mathbf{X}$ is the input image, and b is the bias.

Through the application of convolutional filters, pooling layers, and non-linear activation functions, CNN effectively extract vital features, including edges, textures, and more intricate patterns. The pooling operation, which simplifies the feature map by diminishing its spatial dimensions, is mathematically represented by Equation (5.2).

$$\mathbf{p}_{i,j} = \max_{(i,j)}(\mathbf{f}_{i',j'}) \tag{5.2}$$

where $\mathbf{p}_{i,j}$ denotes the value of the pooled feature map at position (i, j), derived from applying max pooling to a local region of the feature map $\mathbf{f}$. The term $\max_{(i,j)}(\mathbf{f}_{i',j'})$ represents the maximum value within that specified region.

The activation function, typically a Rectified Linear Unit (ReLU), is then applied to each element of the feature map, as expressed in Equation (5.3)

$$\text{ReLU}(\mathbf{f}_{i,j}) = \max(0, \mathbf{f}_{i,j}) \tag{5.3}$$

where $\text{ReLU}(\mathbf{f}_{i,j})$ represents the activated value at position (i, j) in the feature map after applying the ReLU function.

This process enables CNNs to capture a wide range of features and representations from the input image. The diversity in model architectures ensures that each model extracts distinct aspects of these features in unique ways.

To derive the final feature vector from a CNN, the process can be mathematically represented as follows: Let X be the input image, $\mathbf{X} \in \mathbb{R}^{H \times W \times C}$ where:

- H is the height of the input image,
- W is the width of the input image,
- C is the number of channels (e.g., 3 for RGB).

The CNN applies a series of convolutional layers $\mathcal{C}_i(\cdot)$ and pooling layers $\mathcal{P}_i(\cdot)$ to the input image. After L layers, the feature maps can be represented by Equation (5.4).

$$\mathbf{F} = \mathcal{P}_L(\mathcal{C}_L(\ldots \mathcal{P}_1(\mathcal{C}_1(\mathbf{X})))) \tag{5.4}$$

where:

- $\mathbf{F} \in \mathbb{R}^{H' \times W' \times D}$ is the output tensor,
- H' and W' are the spatial dimensions of the feature maps,
- D is the number of feature maps.

The output tensor $\mathbf{F}$ is flattened into a 1D vector $\mathbf{V}$ as presented by the Equation (5.5).

$$\mathbf{V} = \text{Flatten}(\mathbf{F}) \in \mathbb{R}^{H'W'D} \tag{5.5}$$

5.3.2 Concatenation of feature vectors

After employing multiple CNNs to extract features from an input image, we combine these extracted features by concatenating their corresponding flattened vectors. This concatenated vector integrates the diverse representations learned by the different models, providing a more holistic and comprehensive feature set that captures a wider range of characteristics from the input image. The resulting vector is mathematically represented by Equation (5.6).

$$\mathbf{V}_{\text{concat}} = [\mathbf{V}_1; \mathbf{V}_2; \ldots; \mathbf{V}_N] \tag{5.6}$$

where $[\mathbf{V}_1; \mathbf{V}_2; \ldots; \mathbf{V}_N]$ denotes the concatenation of the vectors $\mathbf{V}_1, \mathbf{V}_2, \ldots, \mathbf{V}_N$ using $CNN_1, CNN_2, ..., CNN_N$.

The concatenation process is detailed in Algorithm 5.10.

Algorithm 5.10 Vectors concatenation

Input: $\mathbf{V}_{\text{concat}}$, V_i, $i = 1, \ldots, N$
for $j = 1$ to length(V_i) **do**
  $\mathbf{V}_{\text{concat}}[k] \leftarrow V_i[j]$
  $k \leftarrow k + 1$
end for

5.3.3 Classification

In our approach, we employ the fully connected classifier (FC) introduced in Chapter 1; Section 1.7, Susection 1.7.4. This classifier consists of layers that effectively reduce the dimensionality of the feature space while enhancing pattern recognition, as demonstrated in Equation (5.7).

$$\mathbf{y} = \text{Classifier}(\mathbf{V}_{\text{concat}}) \tag{5.7}$$

where $\mathbf{y}$ is the classification output of the classifier.

The used classifier begins with 1024 neurons, followed by layers containing 512, 256, 128, and 64 neurons. To mitigate overfitting and promote generalization, dropout rates of 0.2 and 0.5 are applied. The final fully connected layer integrates the features into a compact representation, mapping the high-dimensional space to distinct classes for accurate predictions. An illustrative depiction of the classifier is shown in Figure 5.2.

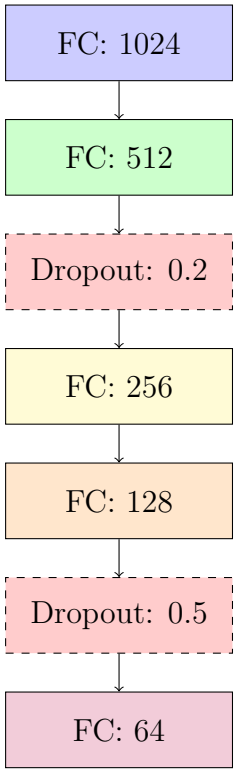


Figure 5.2: The used classifier.

Algorithm 5.11 provides a detailed outline of the proposed method.

Algorithm 5.11 Combining CNN outputs for image classification

Input: Dat: Dataset, A: Set of the used CNN models
Output: Classified image.
for each image X in Dat **do**
  for each CNN_i in A **do**
    Pass X through CNN_i to get $\mathbf{V}_i$
    Concatenate $\mathbf{V}_i$ into $\mathbf{V}_{\text{concat}}$ {using Algorithm 5.10}
  end for
end for
Classification

5.4 Experimental Results

To validate the proposed approach, we conducted extensive experiments utilizing the widely recognized Cats vs. Dogs dataset (refer to Chapter 1, Section 1.6). In this experimental phase, we selected several CNN architectures renowned for their effectiveness in various image classification tasks, including VGG16 and VGG19 (Chapter 1, Section 1.7; Subsection 1.7.3), Inception-V3 (Chapter 1, Section 1.7; Subsection 1.7.5), and EfficientNet-B0 (Chapter 1, Section 1.7; Subsection 1.7.7). In this context, the symbol '+' signifies concatenation; for example, "VGG16 + VGG19" represents the combination of the feature vectors extracted from both VGG16 and VGG19. This methodology allows us to leverage the strengths of multiple models to enhance classification performance.

Table 5.1: Comparative analysis of classification performance across various combined CNN models.

The used models	**Accuracy**	**Error**
VGG16	91.71%	21.36%
VGG19	90.18%	27.18%
VGG16 + VGG19	**93.75%**	**19.05%**
Inception-V3	97.75%	10.53%
Inception-V3 + VGG19	**97.96%**	**09.97%**
Inception-V3 +VGG16	**98.12%**	**04.96%**
Inception-V3 + VGG16 + VGG19	**98.48%**	**04.50%**
EfficientNetB0	96.67%	13.08%
EfficientNetB0 + Inception-V3	**98.32%**	**12.07%**

When comparing the results from combined models with those from individual models in Table 5.1, it is evident that the combined models significantly outperform their single counterparts. By integrating features from multiple models, these combinations achieve higher accuracy and lower error rates than any individual model.

For instance, the VGG16 + VGG19 model attains an accuracy of 93.75% and an error rate of 19.05%. This represents an improvement of +2.04% in accuracy and a reduction in error of -2.31% compared to VGG16, as well as an accuracy increase of +3.57% and a lower error rate of -8.13% relative to VGG19.

Similarly, the Inception-V3 + VGG19 model reaches an accuracy of 97.96% with an error rate of 9.97%. Here, the accuracy differs by +0.21% and the error rate decreases by -0.56% when compared to Inception-V3, while demonstrating an accuracy boost of +7.78% and a drop in error rate of -17.21% compared to VGG19.

The Inception-V3 + VGG16 model achieves an accuracy of 98.12% and an error rate of 4.96%, showing a difference of +0.37% in accuracy and a decrease of -5.57% in error rate compared to Inception-V3, alongside an improvement of +6.41% in accuracy and a reduction of -16.40% in error rate relative to VGG16.

When combining Inception-V3 with both VGG16 and VGG19, the results yield an accuracy of 98.48% and an error rate of 4.50%. This indicates an accuracy improvement of +0.73% and a reduction in error of -6.03% compared to Inception-V3, along with an increase of +6.77% in accuracy and a decrease of -16.8% in error rate compared to VGG16, and a further accuracy improvement of +8.30% with a reduction of -22.68% in error rate compared to VGG19.

Lastly, the EfficientNetB0 + Inception-V3 model achieves an accuracy of 98.32% and an error rate of 12.07%. This shows an accuracy increase of +1.65% and a decrease in error of -1.01% compared to EfficientNetB0 while reflecting an accuracy improvement of +0.57% and an increase in error rate of +1.54% when compared to Inception-V3.

These results indicate that combining models leads to substantial improvements in performance, with the combined models achieving overall higher accuracy and lower error rates. Figure 5.3 illustrates that integrating multiple models results in superior performance, with the best-performing combinations reaching an accuracy of 98.48% and reducing the error rate to 4.50%. In contrast, individual models show lower accuracy rates, with some performing significantly worse than the combined models.

5.5 Challenges of Concatenation Method

Concatenating outputs from multiple CNN models for a classifier presents several challenges: increased computational complexity, risk of overfitting, difficulties in selecting

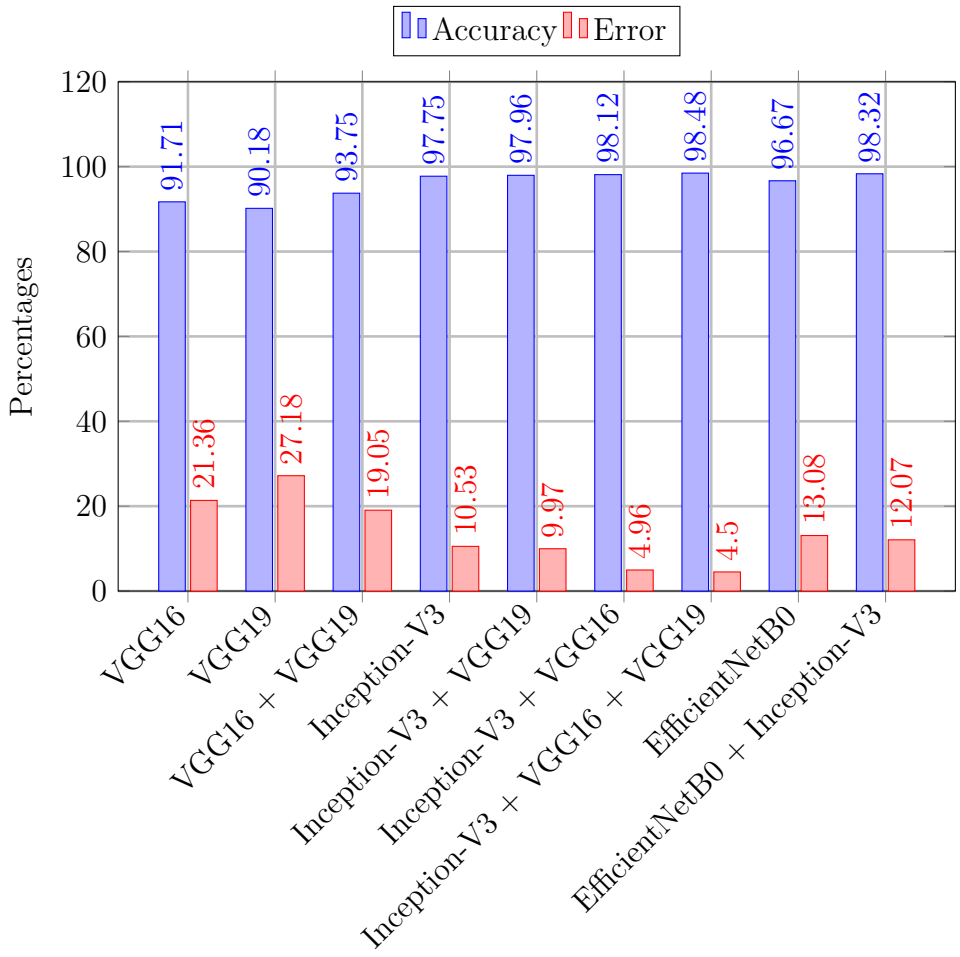


Figure 5.3: Comparison of classification performance, showing accuracy and error rates for different CNN models and their combinations.

compatible models, reduced scalability, and limited interpretability. These factors need careful management to maximize the benefits of this method.

5.6 Conclusion

Our approach of concatenating outputs from multiple CNN models has significantly enhance classification performance. Experimental results demonstrate that this method improves accuracy and reduces error rates compared to single-model approaches. The highest performance was achieved by combining more than two models, confirming the effectiveness of utilizing multiple models to improve robustness and generalization. This promising strategy paves the way for advances in image classification and has broad implications for applications in computer vision and machine learning.

Conclusion and Future works

1 Summary and key findings

In this thesis, our primary objective is to advance the field of image analysis, especially in image classification, by proposing a novel data augmentation method, followed by two enhancements using optimization techniques. We also proposed an additional method to enhance model performance by employing ensemble deep learning.

Our first contribution is based on the random selection of rows and columns from a given image to create new images. The second contribution introduces an enhancement using RO, while the third utilizes GA. Additionally, we have refined two CNN architectures used in the experimental phase. The fourth contribution involves concatenating features extracted using multiple CNN models to enhance model performance.

Building on the existing body of literature, this study investigates advanced approaches for enhancing image analysis, particularly in the domain of DA. These methods enable the generation of compact images that maintain high semantic value, thereby improving the overall effectiveness of image analysis tasks. We further leverage the architecture and features of existing models to enhance image analysis performance and improve classification accuracy.

Despite the variety of existing and proposed methods for augmenting data, they rely on making changes to the image without affecting its size. These modifications may alter the value or position of pixels, resulting in the loss of important information that plays a crucial role in the effectiveness of image analysis tasks.

The classification results demonstrated an enhancement in accuracy when training both versions of a portion of the Cats vs. Dogs dataset (the original dataset and the dataset augmented using the RS method) utilizing a basic CNN. Similar improvements were observed when training the VGG16 model on both versions of the entire Cats vs. Dogs dataset.

In our proposed technique ROEDA, we integrated RO into our initial contribution (RS), which facilitated the selection of the most effective images generated by the RS model. The results indicated a significant improvement compared to the RS method. Ad-

ditionally, when training the VGG16 model on three different versions of the dataset—the original dataset, the dataset augmented using RS, and the dataset augmented using ROEDA noticeable enhancements in classification accuracy were observed.

The use of GA to enhance the RS method has yielded promising results. To evaluate the effectiveness of this innovative approach (GA-based method), we utilized two distinct datasets: the Cats vs. Dogs dataset and the Chest X-ray dataset. For the testing phase, we employed three versions of each dataset: the original dataset, the dataset augmented using the RS method, and the dataset augmented with our proposed method. All three versions of each dataset were used to train seven CNN models: VGG16, VGG19, Inception-V3, EfficientNet-B0, ViT, and the enhanced versions of VGG16 and VGG19. The findings demonstrate that our GA-based proposal effectively captures critical areas of information, facilitating improved model learning. This enhancement significantly boosts the model's performance, particularly in classification tasks.

To demonstrate the effectiveness of the GA-based method, we trained the VGG16 model using two versions of the Cats vs. Dogs dataset. The first version was augmented from the original dataset by a factor of 6 using the RS method, while the second version was augmented with the GA-based method by a factor of 4. The results indicated that the second version, which utilized fewer examples, achieved superior performance compared to the first version. This finding substantiates that our method enhances image quality by selecting the most significant and representative information regions.

The exploitation of capabilities extracted from existing models in image classification allowed us to propose a new method aimed at enhancing the performance of the models for the same task. It is based on a technique that concatenates the features obtained from each model in one feature vector to pass to a classifier.

This approach involves selecting high-performing models in image classification, specifically VGG16, VGG19, Inception-V3, and EfficientNet-B0. We utilize each model individually to extract features from the images. These features are then concatenated and passed to a classifier. As a result, we achieve significantly better outcomes compared to the individual models.

In conclusion, our proposed methods utilize advanced data augmentation techniques, particularly by integrating optimization strategies such as RO and GA. These innovative approaches aim to generate new images that retain high semantic value while enhancing classification performance. By focusing on the most informative and representative regions of the dataset, our techniques improve image quality and ensure that essential information is preserved, ultimately increasing the effectiveness of image analysis tasks. Additionally, we propose an ensemble deep learning method to leverage the strengths of existing models in image classification by concatenating their features, which will then be used for classification.

2 Future work

In future work, we aim to expand the application of our proposed methods across diverse domains. Specifically, we will explore their utility in fields such as medical image analysis, where accurate image classification can significantly impact diagnosis and treatment outcomes, as well as in object detection for autonomous vehicles, where precision is critical. By systematically applying our augmentation strategies, we expect to enhance the performance and robustness of machine learning models in these areas. Additionally, we plan to investigate how these optimization methods can be tailored to meet the unique challenges of different domains, ensuring that the generated augmented data retains high semantic value and improves model learning. This exploration will provide valuable insights into the versatility of our approach and its potential to drive advancements in various applications of image analysis.

Furthermore, it is important to note that the proposed methods require significant resources and time for execution. To address this, we intend to utilize a distributed system to share tasks, which will enable us to gain efficiency and reduce processing time.

Bibliography

[1] J. Twinkle and S. Kaur. A survey of data augmentation techniques in image classification. *International Journal of Engineering Research and Technology*, 9(1):18–23, 2020.

[2] S. Kornblith, H. Shin, N. Dvornik, P. Goyal, K. Duh, and J. Ponce. Do imagenet pretrained models transfer better? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2661–2671, 2019.

[3] M. Frid-Adar, H. C. Shin, A. Fung, A. P. M., and et al. Synthetic data for deep learning. *Nature Machine Intelligence*, 1(3):162–168, 2018.

[4] M. Ouali, A. Ait Si Ali, I. Boughazi, N. Zaghba, and A. Khamassi. A comprehensive review on semi-supervised learning. *arXiv preprint arXiv:2004.05873*, 2020.

[5] M. Barbuto and A. de Almeida. A survey on few-shot learning: Methods and applications. *IEEE Transactions on Neural Networks and Learning Systems*, 33(10):4678–4695, 2021.

[6] A. Brock, J. Donahue, and K. Simonyan. Large scale gan training for high fidelity natural image synthesis. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.

[7] N. Sultana and I. Hussain. A survey of crowdsourcing techniques for data annotation. *International Journal of Computer Applications*, 975:8–14, 2020.

[8] P. Baldi and P. J. Sadowski. Understanding dropout. In *Advances in Neural Information Processing Systems*, volume 26, 2013.

[9] H. Yin and Y. Liu. Recent advances in ensemble learning: A review. *IEEE Transactions on Neural Networks and Learning Systems*, 31(11):4399–4413, 2020.

[10] Alhussein Fawzi, Horst Samulowitz, Deepak Turaga, and Pascal Frossard. Adaptive data augmentation for image classification. In *2016 IEEE international conference on image processing (ICIP)*, pages 3688–3692. Ieee, 2016.

[11] Agnieszka Mikołajczyk and Michał Grochowski. Data augmentation for improving deep learning in image classification problem. In *2018 international interdisciplinary PhD workshop (IIPhDW)*, pages 117–122. IEEE, 2018.

[12] Akbar Karimi, Leonardo Rossi, and Andrea Prati. Aeda: an easier data augmentation technique for text classification. *arXiv preprint arXiv:2108.13230*, 2021.

[13] Canjie Luo, Yuanzhi Zhu, Lianwen Jin, and Yongpan Wang. Learn to augment: Joint data augmentation and network optimization for text recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13746–13755, 2020.

[14] Tom Ko, Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur. Audio augmentation for speech recognition. In *16th annual conf. of the int'l speech communication association*, pages 3586–3589, 2015.

[15] Deokgyu Yun and Seung Ho Choi. Deep learning-based estimation of reverberant environment for audio data augmentation. *Sensors*, 22(2):592, 2022.

[16] Guillermo Iglesias, Edgar Talavera, Ángel González-Prieto, Alberto Mozo, and Sandra Gómez-Canaval. Data augmentation techniques in time series domain: A survey and taxonomy. *arXiv preprint arXiv:2206.13508*, 2022.

[17] Xiang Wang, Kai Wang, and Shiguo Lian. A survey on face data augmentation for the training of deep neural networks. *Neural computing and applications*, 32(19):15503–15531, 2020.

[18] Artemios-Anargyros Semenoglou, Evangelos Spiliotis, and Vassilios Assimakopoulos. Data augmentation for univariate time series forecasting with neural networks. *Pattern Recognition*, 134:109132, 2023.

[19] M. Paschali, W. Simson, A. G. Roy, M. F. Naeem, R. Göbl, C. Wachinger, and N. Navab. Data augmentation with manifold exploring geometric transformations for increased performance and robustness. *arXiv preprint arXiv:1901.04420*, 2019.

[20] R. Takahashi, T. Matsubara, and K. Uehara. Data augmentation using random image cropping and patching for deep cnns. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(9):2917–2931, 2019.

[21] Mohamed Elgendi, Muhammad Umer Nasir, Qunfeng Tang, David Smith, John-Paul Grenier, Catherine Batte, Bradley Spieler, William Donald Leslie, Carlo Menon, Richard Ribbon Fletcher, et al. The effectiveness of image augmentation in deep learning networks for detecting covid-19: A geometric transformation perspective. *Frontiers in Medicine*, 8:629134, 2021.

[22] Andrew G Howard. Some improvements on deep convolutional neural network based image classification. *arXiv preprint arXiv:1312.5402*, 2013.

[23] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

[24] Sanghi Yun, Donggeun Han, Seong Joon Oh, Sanghoon Chun, and Youngjoo Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. *arXiv preprint arXiv:1905.04899*, 2019.

[25] Pengguang Chen, Shu Liu, Hengshuang Zhao, and Jiaya Jia. Gridmask data augmentation. *arXiv preprint arXiv:2001.04086*, 2020.

[26] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang. Random erasing data augmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 13001–13008, April 2020.

[27] E. K. Kim, H. Lee, J. Y. Kim, and S. Kim. Data augmentation method by applying color perturbation of inverse psnr and geometric transformations for object recognition based on deep learning. *Applied Sciences*, 10(11):3755, 2020.

[28] Antreas Antoniou, Amos Storkey, and Harrison Edwards. Data augmentation generative adversarial networks. *arXiv preprint arXiv:1711.04340*, 2017.

[29] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2014.

[30] Linlin Hou, Ruijiao Wang, Xiaodong Zhang, Qi Zhan, and Yifan Zhang. Vagan: Attribute-aware visual anomaly detection with generative adversarial network. *arXiv preprint arXiv:1904.10706*, 2019.

[31] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. A neural algorithm of artistic style. *Journal of Vision*, 16(12):326–326, 2016.

[32] Xun Huang, Mengdan Liu, Serge Belongie, and Jan Kautz. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1510–1519. IEEE, 2017.

[33] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation policies from data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 113–123, 2019.

[34] E. D. Cubuk, B. Zoph, and Q. V. Le. Randaugment: Practical data augmentation with no training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3026–3036. IEEE, 2020.

[35] J. Lim, H. Kim, and S. Lee. Population-based augmentation: Efficient data augmentation for classification and object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1827–1836. IEEE, 2019.

[36] Aleksander Madry, Andrei Makelov, Ludwig Schmidt, Dimitris Tsipras, and Leonid VL. Towards deep learning models resistant to adversarial attacks. In *Proceedings of the 6th International Conference on Learning Representations (ICLR)*, 2018.

[37] Ian J Goodfellow, Jon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[38] Ting Chen, Simon Kornblith, Arvind Srinivasan, Joshua Tenenbaum, and Oriol Vinyals. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020.

[39] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. *arXiv preprint arXiv:1911.05722*, 2020.

[40] Mathilde Caron, Thibaut Courtade, Olivier Bachem, Olivier Hénaff, Piotr Bojanowski, Matthijs Douze, Jean Lecrerc, Hadrien Aflalo, Vincent Lemaire, Puneet Goyal, et al. Unsupervised learning of visual features by contrasting cluster assignments. *arXiv preprint arXiv:2006.09882*, 2020.

[41] T. DeVries and G. W. Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.

[42] Hongyi Zhang, Moustapha Cisse, Krishnamurthy Dvijotham, Frank Golatowski, and Quoc V. Le. Mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2018.

[43] Qizhe Xie, Zihang Dai, Gholamreza Haffari, and Junyoung Hwang. Unsupervised data augmentation for consistency training. In *Advances in Neural Information Processing Systems*, volume 33, pages 6256–6266, 2019.

[44] Ekin D. Cubuk, Barret Zoph, Dhruv Mane, V. Vasudevan, and Quoc V. Le. Autoaugment: Learning augmentation policies from data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 113–123, 2020.

[45] Daniel Ho, Xi Chen, Aravind Srinivas, Yan Duan, and Pieter Abbeel. Population based augmentation: Efficient learning of augmentation policy schedules. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, pages 2731–2741, 2019.

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

[47] Teerath Kumar, Rob Brennan, Alessandra Mileo, and Malika Bendechache. Image data augmentation approaches: A comprehensive survey and future directions. *IEEE Access*, 2024.

[48] Richard Loree Anderson. Recent advances in finding best operating conditions. *Journal of the American Statistical Association*, 48(264):789–798, 1953.

[49] Jeffrey R Sampson. Adaptation in natural and artificial systems (john h. holland), 1976.

[50] Rainer Storn and Kenneth Price. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11:341–359, 1997.

[51] Scott Kirkpatrick, C Daniel Gelatt Jr, and Mario P Vecchi. Optimization by simulated annealing. *science*, 220(4598):671–680, 1983.

[52] Fred Glover. Future paths for integer programming and links to artificial intelligence. *Computers & operations research*, 13(5):533–549, 1986.

[53] Marco Dorigo, Vittorio Maniezzo, and Alberto Colorni. Ant system: optimization by a colony of cooperating agents. *IEEE transactions on systems, man, and cybernetics, part b (cybernetics)*, 26(1):29–41, 1996.

[54] James Kennedy and Russell Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95-international conference on neural networks*, volume 4, pages 1942–1948. ieee, 1995.

[55] Alireza Askarzadeh. A novel metaheuristic method for solving constrained engineering optimization problems: crow search algorithm. *Computers & structures*, 169:1–12, 2016.

[56] Xin-She Yang. Firefly algorithm, stochastic test functions and design optimisation. *International journal of bio-inspired computation*, 2(2):78–84, 2010.

[57] Maziar Yazdani and Fariborz Jolai. Lion optimization algorithm (loa): a nature-inspired metaheuristic algorithm. *Journal of computational design and engineering*, 3(1):24–36, 2016.

[58] Augustin Cauchy et al. Méthode générale pour la résolution des systemes d'équations simultanées. *Comp. Rend. Sci. Paris*, 25(1847):536–538, 1847.

[59] John R Koza. Genetic programming as a means for programming computers by natural selection. *Statistics and computing*, 4:87–112, 1994.

[60] Christopher John Cornish Hellaby Watkins. Learning from delayed rewards. 1989.

[61] B Zoph. Neural architecture search with reinforcement learning. *arXiv preprint arXiv:1611.01578*, 2016.

[62] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[63] Awais Khan, Kuntha Pin, Ahsan Aziz, Jung Woo Han, and Yunyoung Nam. Optical coherence tomography image classification using hybrid deep learning and ant colony optimization. *Sensors*, 23(15):6706, 2023.

[64] Hyungkeuk Lee, NamKyung Lee, and Sungjin Lee. A method of deep learning model optimization for image classification on edge device. *Sensors*, 22(19):7344, 2022.

[65] Hongwei Yong, Jianqiang Huang, Xiansheng Hua, and Lei Zhang. Gradient centralization: A new optimization technique for deep neural networks. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, pages 635–652. Springer, 2020.

[66] Fatemeh Sadeghi, Ata Larijani, Omid Rostami, Diego Martín, and Parisa Hajirahimi. A novel multi-objective binary chimp optimization algorithm for optimal feature selection: Application of deep-learning-based approaches for sar image classification. *Sensors*, 23(3):1180, 2023.

[67] Zhipeng Ling, Qi Xin, Yiyu Lin, Guangze Su, and Zuwei Shui. Optimization of autonomous driving image detection based on rfaconv and triplet attention. *arXiv preprint arXiv:2407.09530*, 2024.

[68] The Kaggle official web site, 2023. Available at https://www.kaggle.com/dataset (2023-08-13).

[69] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778. IEEE, 2016.

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

[71] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going

deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[72] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60:84–90, 2012.

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

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

[75] Boudouh Nouara and Mokhtari Bilal. Random pixel selection through image cropping for data augmentation and classification. In *2022 International Symposium on iNnovative Informatics of Biskra (ISNIB)*, pages 1–6. IEEE, 2022.

[76] Sarada Mohapatra and Prabhujit Mohapatra. Fast random opposition-based learning golden jackal optimization algorithm. *Knowledge-Based Systems*, 275:110679, 2023.

[77] Amelia Carolina Sparavigna. Entropy in image analysis, 2019.

[78] Ricardo Alonso Espinosa Medina. Espen graph for the spatial analysis of entropy in images. *Entropy*, 25(1):159, 2023.

[79] Boudouh Nouara and Mokhtari Bilal. Random pixel selection through image cropping for data augmentation and classification. In *2022 International Symposium on iNnovative Informatics of Biskra (ISNIB)*, pages 1–6, 2022.

[80] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016.

[81] Francisco López de la Rosa, José L Gómez-Sirvent, Roberto Sánchez-Reolid, Rafael Morales, and Antonio Fernández-Caballero. Geometric transformation-based data augmentation on defect classification of segmented images of semiconductor materials using a resnet50 convolutional neural network. *Expert Systems with Applications*, 206:117731, 2022.

[82] Mahmoud Smaida and Serhii Yaroshchak. Bagging of convolutional neural networks for diagnostic of eye diseases. In *COLINS*, pages 715–729, 2020.

[83] Mounika Nalluri, Mounika Pentela, and Nageswara Rao Eluri. A scalable tree boosting system: Xg boost. *Int. J. Res. Stud. Sci. Eng. Technol*, 7(12):36–51, 2020.

[84] Begum Ay Ture, Akhan Akbulut, Abdul Halim Zaim, and Cagatay Catal. Stacking-based ensemble learning for remaining useful life estimation. *Soft Computing*, 28(2):1337–1349, 2024.

[85] Tong He, Zhi Zhang, Hang Zhang, Zhongyue Zhang, Junyuan Xie, and Mu Li. Bag of tricks for image classification with convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 558–567, 2019.

[86] M Tanveer, Aryan Rastogi, Vardhan Paliwal, MA Ganaie, Ashwani Kumar Malik, Javier Del Ser, and Chin-Teng Lin. Ensemble deep learning in speech signal tasks: a review. *Neurocomputing*, 550:126436, 2023.

[87] Neelesh Mungoli. Adaptive feature fusion: enhancing generalization in deep learning models. *arXiv preprint arXiv:2304.03290*, 2023.

[88] Na Dong, Qingyue Feng, Mengdie Zhai, Jianfang Chang, and Xiaoming Mai. A novel feature fusion based deep learning framework for white blood cell classification. *Journal of Ambient Intelligence and Humanized Computing*, pages 1–13, 2023.

[89] Kuo-Hsuan Lin, Nan-Han Lu, Takahide Okamoto, Yung-Hui Huang, Kuo-Ying Liu, Akari Matsushima, Che-Cheng Chang, and Tai-Been Chen. Fusion-extracted features by deep networks for improved covid-19 classification with chest x-ray radiography. In *Healthcare*, volume 11, page 1367. MDPI, 2023.

[90] Kaiyang Liao, Gang Huang, Yuanlin Zheng, Guangfeng Lin, and Congjun Cao. Coordinate feature fusion networks for fine-grained image classification. *Signal, Image and Video Processing*, 17(3):807–815, 2023.