

University of Louisville

ThinkIR: The University of Louisville's Institutional Repository

Electronic Theses and Dissertations

5-2024

Attention guided data augmentation for improving the classification performance of vision transformers.

Nada Baili
University of Louisville

Follow this and additional works at: <https://ir.library.louisville.edu/etd>



Part of the [Other Computer Engineering Commons](#)

Recommended Citation

Baili, Nada, "Attention guided data augmentation for improving the classification performance of vision transformers." (2024). *Electronic Theses and Dissertations*. Paper 4327.
<https://doi.org/10.18297/etd/4327>

This Doctoral Dissertation is brought to you for free and open access by ThinkIR: The University of Louisville's Institutional Repository. It has been accepted for inclusion in Electronic Theses and Dissertations by an authorized administrator of ThinkIR: The University of Louisville's Institutional Repository. This title appears here courtesy of the author, who has retained all other copyrights. For more information, please contact thinkir@louisville.edu.

ATTENTION GUIDED DATA AUGMENTATION FOR IMPROVING THE
CLASSIFICATION PERFORMANCE OF VISION TRANSFORMERS

By

Nada Baili
M.Sc., Computer Science and Engineering
University of Louisville, KY, USA

A Dissertation
Submitted to the Faculty of the
J.B. Speed School of Engineering
in Partial Fulfillment of the Requirements
for the Degree of

Doctor of Philosophy in Computer Science and Engineering

Department of Computer Science and Engineering
University of Louisville
Louisville, Kentucky

May 2024

Copyright 2024 by Nada Baili

All rights reserved

ATTENTION GUIDED DATA AUGMENTATION FOR IMPROVING THE
CLASSIFICATION PERFORMANCE OF VISION TRANSFORMERS

By

Nada Baili
M.Sc., Computer Science and Engineering
University of Louisville, KY, USA

A Dissertation Approved on
April 23, 2024

By the Following Dissertation Committee:

Hichem Frigui, Ph.D., Dissertation Director

Olfa Nasraoui, Ph.D.

Andrew David Karem, Ph.D.

Sabur Hassan Baidya, Ph.D.

Tamer Inanc, Ph.D.

ACKNOWLEDGEMENTS

I would like to take this opportunity to acknowledge all those who have supported me during the journey of completing my PhD dissertation.

First, I would like to thank my advisor Prof. Hichem Frigui, for his guidance and support throughout my research. His mentorship has been valuable in shaping my research and career aspirations.

I also extend my sincere appreciation to my committee members, Prof. Olfa Nasraoui, Prof. Andrew David Karem, Prof. Sabur Hassan Baidya, and Prof. Tamer Inanc, for their constructive feedback and insightful comments. I appreciate their time, and effort in making this dissertation possible.

I would like to express my gratitude to all those who have played a part in my academic and personal growth. To my colleagues, mentors, and teachers who have challenged and inspired me, thank you for sharing your knowledge and expertise.

To all my dear friends. To Syrine Benammou, Fadoua Khmaissia, and Sahar Sinene Mehdoui. I am deeply grateful for your support and encouragement. Your friendship has been a source of joy and inspiration, and has made the difficult moments more bearable. Thank you for the fun times that have made my PhD journey more enjoyable and memorable.

Finally, I would like to extend my heartfelt appreciation to my family. To my parents, who have always believed in me and supported me in every decision I made. To Nawel, Aymen, Assaad, and Nesrine, who have always been my source of inspiration. Last but not least, to my husband and life partner, Khalil, and to my precious son, Yusuf. Your unconditional love and support have been a driving force in my life. I am forever indebted to you.

ABSTRACT

ATTENTION GUIDED DATA AUGMENTATION FOR IMPROVING THE CLASSIFICATION PERFORMANCE OF VISION TRANSFORMERS

Nada Baili

April 23, 2024

For over a decade, Deep Neural Networks (DNNs) have been rapidly progressing and achieving great success, forming a robust foundation of state of the art machine learning algorithms that impacted various domains. The advances in data acquisition and processing have undeniably played a major role in these breakthroughs. Data is a crucial component in building successful DNNs, as it enables machine learning models to optimize complex architectures, necessary to perform certain difficult tasks. However, acquiring large-scale data sets is not enough to learn robust models with generalizable features. Instead, an ideal training set should be diverse enough and contain enough variations within each class for the model to learn the most optimal decision boundaries.

The poor performance of a machine learning model can often be traced back to the existence of under-represented regions in the feature space of the training data. These sparse regions can prevent the model from capturing the large intra-class variations. Data augmentation is a common technique that has been used to inflate training datasets with new samples, as an attempt to improve the model performance. However, these techniques usually focus on expanding the data in size and do not necessarily aim to cover the under-represented regions of the feature space.

This dissertation presents a novel Attention-guided Data Augmentation technique for Vision Transformers, called ADA-ViT. Our method is tailored to be specifically applied to Transformer-based vision models. These models are considered the state of the art learning strategy in almost all computer vision applications, and they have gained more interest in recent research than their classic counterparts, e.g. convolution-based networks.

Our proposed data augmentation method aims to improve the diversity of the training set by selecting informative samples with respect to their potential contributions of improving the

model performance. We leverage the attention scores computed within the transformer model to get an insight on the image regions that caused the misclassification. The identified image regions form misclassification concepts that explain the model limitations in a given class. These learned concepts indicate the presence of under-represented regions in the training dataset that contributed to the misclassifications. We leverage this information to guide our data augmentation process by identifying new samples and using them to augment the training data in an effort to improve the coverage of the identified under-represented regions. We achieve this by designing a utility function to rank and select new samples from secondary image repositories based on their similarity to the extracted misclassification concepts.

ADA-ViT aims beyond increasing the data in size. It focuses on improving the diversity of the training set by finding and covering under-represented regions in the feature space of the training data. To the best of our knowledge, no prior work has considered this aspect for the case of Vision Transformer models. The advantage of our approach is that it leverages available noisy web data repositories for augmentation, thus alleviating the need for large labeled data. This is because ADA-ViT uses a ranking system that can filter out noisy and irrelevant samples.

We evaluate our data augmentation technique on two computer vision applications, and using multiple scenarios. We conduct extensive experiments and analysis to demonstrate the problem of under-represented regions in the training feature space and show the effectiveness of our method in addressing this issue. We also compare our method, using benchmark datasets, to baseline models trained using the available labeled data only, and using the augmented labeled data and state-of-the-art data augmentation methods. We show that our proposed augmentation consistently improves the results. We also perform an in-depth analysis to justify the observed improvements.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS		iii
ABSTRACT		iv
LIST OF TABLES		ix
LIST OF FIGURES		xi
1	INTRODUCTION	1
1.1	Problem Statement	2
1.2	Overview of the Proposed Solution	3
1.3	Thesis Contributions	4
1.4	Document Organization	5
2	BACKGROUND AND RELATED WORKS	6
2.1	Data Augmentation	6
2.1.1	Data Augmentation without External Data Repositories	6
2.1.2	Data Augmentation with External Data Repositories	7
2.1.3	Data Augmentation for the Under-Represented Regions	8
2.2	Vision Transformers	9
2.2.1	Limitations of Convolution Neural Networks	10
2.2.2	Overview of the Vision Transformer Architecture	10
2.2.3	Self-Attention	12
2.2.4	Multi-Head Attention	13
2.3	Automatic Target Recognition	14
3	ATTENTION-GUIDED DATA AUGMENTATION FOR IMPROVING	
	THE CLASSIFICATION PERFORMANCE OF VISION TRANSFORMERS 16	
3.1	Overview of ADA-ViT algorithm	17
3.2	Similarity Function ($\varphi(\cdot)$)	19
3.3	Label Regularization Score ($\alpha(\cdot)$)	20
3.4	Under-representation Score ($\beta(\cdot)$)	20
3.5	Degree of match ($\Delta(\cdot)$)	21
3.5.1	Identification of Misclassification Concepts	22
3.5.2	Computation of Δ	23
3.6	Vision Transformers with Adaptive Data Augmentation	24

3.6.1	On the Convergence of our Learning Framework	26
3.6.2	Justification of the choice of the Vision Transformer Model	26
4	APPLICATION 1: OBJECT IDENTIFICATION IN RGB IMAGES	28
4.1	Experimental Setup	29
4.1.1	Datasets	29
4.1.2	Baseline Model	32
4.1.3	Training Parameters	33
4.2	Performance Analysis	33
4.2.1	Performance of a Baseline ViT without ADA-ViT Augmentation	34
4.2.2	Performance of a Baseline ViT with ADA-ViT Augmentation	37
4.2.3	Analysis of Misclassified Samples	38
4.3	Illustration of ADA-ViT	42
4.4	Class Representation using Multiple Prototypes	44
4.5	Stopping Criteria for ADA-ViT Iterations	47
4.6	Ablation study	51
4.6.1	Justification of the ADA-ViT Scoring Function Design	51
4.6.2	Importance of Guiding Data Augmentation by ADA-ViT scoring	53
4.6.3	Impact of the Size of Selected New Samples	56
4.7	Comparison with other State-Of-The-Art Data Augmentation Techniques	56
4.8	Chapter Summary	60
5	APPLICATION 2: AUTOMATIC TARGET RECOGNITION FROM	
	INFRARED IMAGES	62
5.1	Data Preparation	63
5.1.1	YOLO Algorithm Overview	63
5.1.2	Original Training Set	64
5.1.3	External Image Repository	64
5.2	Experimental Analysis	66
5.2.1	Performance of a Baseline ViT without ADA-ViT Augmentation	67
5.2.2	Performance of a Baseline ViT with ADA-ViT Augmentation	69
5.3	Importance of Guiding Data Augmentation by ADA-ViT scoring	72
5.4	Comparison with state-of-the-art Data Augmentation Techniques	74
5.5	Chapter Summary	77
6	CONCLUSIONS AND POTENTIAL FUTURE WORKS	78
6.1	Conclusions	78

6.2 Potential Future Work	80
REFERENCES	81
CURRICULUM VITAE	87

LIST OF TABLES

4.1	Data partition for CUB, CUB-Families, and TinyImageNet	31
4.2	Intra-class variance and inter-class similarity measures for CUB, CUB-Families, and TinyImageNet.	31
4.3	Accuracy performance of baseline models on the test set.	34
4.4	Accuracy performance of baseline models with and without ADA-ViT augmentation.	37
4.5	Number of images added by ADA-ViT for the different datasets.	37
4.6	Classification accuracy when different methods are used to represent the training data of each class.	46
4.7	Size of selected samples by ADA-ViT for augmentation for each iteration. We report the results for the ViT-B model.	48
4.8	Evolution of the misclassified samples in the test set of CUB dataset across five iterations. We mark by * the optimal number of iterations.	49
4.9	Evolution of the misclassified samples in the test set of CUB-Families dataset across five iterations. We mark by * the optimal number of iterations.	49
4.10	Evolution of the misclassified samples in the test set of TinyImageNet dataset across five iterations. We mark by * the optimal number of iterations.	50
4.11	Ablation study: Comparison of classification accuracies of different ADAViT variants on the three selection datasets.	52
4.12	IOU between the selected images by the four different ADAViT variants for CUB dataset.	53
4.13	Random vs. Confidence-based vs. ADA-ViT-guided Augmentation.	55
4.14	Comparison of the classification accuracies on the three selected datasets. We run ADAViT for 3 iterations. For the CNN baseline, we only report the accuracy of BRACE and we do not run other augmentation methods with the CNN baseline as this is outside the scope of our research.	59
4.15	Number of images added by each data augmentation method that requires external datasets for augmentation.	60
5.1	Number of samples per class and per data partition in the FLIR ADAS dataset.	65
5.2	Number of generated detections per class by YOLO for the Brno dataset.	66
5.3	Accuracy results of baseline ATR systems trained on the original dataset only.	67

5.4	Accuracy results of the ATR system finetuned on the original dataset and the ADA-ViT augmentations.	69
5.5	Number of added samples by ADA-ViT after three iterations.	71
5.6	Comparison of the accuracy of the ViT with different data selection methods. . . .	74
5.7	Comparison of classification accuracies of different data augmentation techniques on FLIR ADAS dataset. We run ADAViT for 3 iterations. For the CNN baseline, we only report the accuracy of BRACE and we do not run other augmentation methods with the CNN baseline as this is outside the scope of our research.	75
5.8	Number of images added by each data augmentation method for FLIR ADAS. . . .	76

LIST OF FIGURES

2.1	An overview of the ViT architecture [1].	11
2.2	Representative examples of attention from the output token to the input space [1].	12
2.3	(a): Architecture of a transformer encoder. (b): Architecture of a multi-head self-attention block. (c): Components of a single self-attention head featuring the scaled-dot product attention.	13
3.1	Proposed learning approach that integrates data augmentation in the learning process.	17
3.2	Examples of selected samples by ADAViT for CUB dataset. In the third column, we highlight the patches responsible for the misclassification of the validation image. In the fourth column, we highlight the patches on the new image that increased its corresponding utility score the most, leading to its selection for augmentation.	18
3.3	Examples of label regularization cases. The score α is highest when the web label is correct (examples (1) and (2)), and is lowest when the web label is incorrect (examples (3) and (4)).	21
3.4	(a): An image of a female <i>Painted Bunting</i> misclassified as an <i>Orange Crowned Warbler</i> . (b): The heatmap of the attention map superimposed on the original image. (c): Typical images of <i>Painted Bunting</i> in the training set (mostly images of the male bird). (d): Example images of <i>Orange Crowned Warbler</i> in the training set.	23
3.5	Overview of the proposed ADA-ViT framework. For simplicity, we assume there is only one validation image from c_i misclassified as c_j . N is the number of image patches, D is the ViT’s embedding size, Z_t^c is the set of training feature vectors representative of class c , and [CLS] designates the classification token. Arrows in <i>red</i> describe the algorithm flow for the validation data, while arrows in <i>blue</i> describe the algorithm flow for the new web sample. This figure is best viewed in color.	25
4.1	Sample images from the web dataset used for CUB and CUB-Families.	32
4.2	Sample images from the web dataset used for TinyImageNet.	32
4.3	Samples of correctly classified validation images and their 5 NN images from the training set using the baseline model for CUB. We indicate the true class label and the distance of the kNN above each image.	35
4.4	Samples of misclassified validation images and their 5 NN images from the training set using the baseline model for CUB. We indicate the true class label and the distance of the kNN above each image.	36

4.5	Category 1 misclassifications and their 5 nearest neighbors from the training set. The first column shows test examples from Category 1. The remaining five columns show the 5 NNs from the training set using embeddings generated by (a) the baseline model, and (b) the first iteration of ADA-ViT augmentation.	39
4.6	Category 2 misclassifications and their 5 nearest neighbors from the training set. The first column shows test examples from Category 2. The remaining five columns show the 5 NNs from the training set using the embeddings generated by (a) the baseline model, and (b) the first iteration of ADA-ViT augmentation.	40
4.7	Category 3 misclassifications and their 5 nearest neighbors from the training set. The first column shows test examples from Category 3. The remaining five columns show the 5 NNs from the training set using the embeddings generated by (a) the baseline model, and (b) the first iteration of ADA-ViT augmentation.	41
4.8	TSNE analysis of the four selected classes (C) from CUB-Families: <i>Cuculidae</i> (c1), <i>Alcidae</i> (c2), <i>Mimidae</i> (c3), <i>Fringilidae</i> (c4), and <i>Podicipedidae</i> (c5). In the legend, T represents the training data of C only, \bar{T} is the training data for all classes except C , $V\text{-correct}$ is the correctly classified validation data of C , $V\text{-incorrect}$ is the incorrectly classified validation data of C , and New indicates the added samples for C by ADA-ViT. This figure is best viewed in color.	45
4.9	Comparison of different methods to represent the class <i>Tyrannidae</i> from CUB-Families and the class <i>Wooden Spoon</i> from TinyImageNet. This figure shows the TSNE projection of the training samples of the studied classes, as well as the vector corresponding to the mean of feature vectors of all training samples, the medoids of the clusters obtained by each clustering algorithm, and a subset of the added new samples for these classes.	47
4.10	Examples of selected images for CUB dataset by $ADAViT$, $ADAViT^{-\beta}$, and $ADAViT^{-\Delta}$. We display the misclassification concepts on the misclassified validation sample, as well as the patches on the new selected image that match these concepts, as identified by the Δ term in $ADAViT$ and $ADAViT^{-\beta}$	53
4.11	Examples of selected images by the Random Augmentation and Guided Augmentation by ADA-ViT scoring.	55
4.12	Evolution of the performance of the model trained on different sizes of ADA-ViT augmentations on (a) CUB dataset, (b) CUB-Families and (c) Tiny ImageNet.	57
5.1	Example images from the FLIR ADAS dataset [2].	65
5.2	Generated detections by the finetuned YOLOv5x model from Brno dataset.	66

5.3	Confusion matrix of the baseline ViT-B model trained on FLIR ADAS before augmentation and evaluated on the test set. Note that the class <i>bus</i> is absent from the test set.	68
5.4	Examples of misclassified test images and their nearest neighbors using the baseline model for FLIR ADAS. The first column displays test images misclassified by the baseline. The remaining columns show the 5 nearest neighbors from the training set. We display the true label and the distance above each nearest neighbor.	69
5.5	2-D TSNE analysis of the training, misclassified validation, and new samples.	70
5.6	Confusion matrix of the ATR system finetuned on the original dataset and the ADA-ViT augmentations, and evaluated on the test set. Note that the class <i>bus</i> is absent from the test set.	71
5.7	Corrected test samples by ADA-ViT from Figure 5.4, and their nearest neighbors from the training set using the model finetuned on ADA-ViT augmentations. The first column displays test images correctly classified by the finetuned model. The remaining columns show the 5 nearest neighbors from the training set. We display the true label and the distance above each nearest neighbor. The added images by ADA-ViT are marked by green boxes.	72

LIST OF ALGORITHMS

3.1 Attention-Guided Data Augmentation for Vision Transformers	19
--	----

CHAPTER 1

INTRODUCTION

Deep Neural Networks (DNNs) have demonstrated outstanding performances in several computer vision tasks, including object classification [3, 4], object detection [5, 6], etc. proving to be essential in our modern world and impacting a wide range of domains from automated driving [7, 8], to medical devices [9, 10], and even safety systems [11]. It is undeniable that a significant part of the success of DNNs is owed to the tremendous efforts by the research community to build robust neural network architectures [12, 13], and develop efficient training techniques. However, the increasing complexity of DNN systems is directly correlated with the need for larger data sets to train these machine learning models.

The need for huge amounts of data has been even more exacerbated with the recent emergence of Transformer models. These networks revolutionized most machine learning fields, such as natural language processing [14], and computer vision [1]. Because of their particular deep learning structure that is distinct from the classic convolution-based computer vision models, Vision Transformers (ViT) lack some of the inductive biases inherently found in Convolutional Neural Networks (CNNs), such as translation equivariance and locality. Therefore, ViT models do not generalize well when trained on insufficient amounts of data. The absence of inductive bias allowed Transformer models to significantly exceed state of the art on several computer vision tasks [1, 15–17], since they are not bound by strict assumptions and are able to explore deeper patterns and more complex features. However, this success comes at the expense of requiring larger data sets for training.

Acquiring large-scale labeled data sets is a tedious and expensive process. It may even be infeasible in certain fields, such as healthcare [18]. Meanwhile, there is an abundance of data on the web that can be easily retrieved through web scraping that uses bots to extract data online. However, this type of data hasn't gained much interest, despite its tremendous size and lower costs, because of its weak and uncertain annotations and noise. Therefore, there is a high need for a mechanism that enables automatic sample selection and filtering of these web data sets, since manually cleaning them is costly and tedious, to unlock their high potential and solve the issue of data shortage.

Acquiring large-scale data sets is not necessarily sufficient to train successful DNNs. Beside the size, an ideal training set should be diverse and contain enough variations to cover the different patterns exhibited within each class. Failure to collect a representative training set results in the presence of under-represented regions in the feature space of the training data. Therefore, a model

trained on such incomplete training sets will not generalize well to unseen test data that fall in these gaps of the feature space.

In this dissertation, we address the problem of improving the performance of vision transformer models when the available training data is either small in size or incomplete in representativeness. Based on the fact that the success of machine learning systems depends crucially on the quality of the data, and not just their size, we focus on studying the shortcomings of the trained ViT model by uncovering the presence of under-represented regions in the feature space of the training data. We design a data augmentation algorithm that carefully selects candidate samples from secondary weakly-annotated image repositories to cover these under-represented regions and increase the size of the training data with only relevant samples. Our goal is to bridge the gap in performance of ViT models by improving the diversity and size of the training data while leveraging available weakly-annotated data sets, without the need for meticulous labeling and data cleaning.

1.1 Problem Statement

The quality of the training dataset, measured in size and diversity, plays a major role in determining the performance of a machine learning model. Ideally, a machine learning model should be exposed to a diverse training set that covers the variance imposed by the task in hand, to generalize well to unseen data during inference. However, this is not usually the case, as training data may not be sufficiently comprehensive and informative to train robust models.

Data augmentation is an intuitive way to circumvent this limitation, by expanding the data with new samples that can boost the diversity and coverage of each class in a classification scenario. Several research efforts adopted the direction of synthetically creating new samples from the existing training set by applying geometric transformations, color space augmentations, noise injection, ...etc. These techniques constitute the foundation of data augmentation and have been widely used. They have shown to significantly improve the performance of machine learning models, while being easy to implement and apply without the need for the cost of manually acquiring more labeled data. However, these methods are constrained to exploring local neighborhoods of the available data samples and cannot target specific sparse regions of the feature space. Therefore, they are unable to significantly expand the diversity and coverage of the training dataset.

Another way to augment data is to make use of external image repositories obtained from the web. Although they are large in scale and relatively cheap to acquire, these web datasets are usually weakly annotated and can contain out-of-distribution images. If used directly without filtering, these datasets are more likely to have a negative impact on the model's performance. There are some research efforts [19, 20] that aimed to develop unsupervised preprocessing to filter these web datasets prior to using them for augmentation. However, these methods focused more on expanding the data in size regardless of its representativeness. Adding images with information already present

in the current training dataset, can lead to model overfitting, or unnecessarily calls for more complex architectures to process the large data load.

1.2 Overview of the Proposed Solution

Most existing data augmentation techniques are designed to increase the size of training sets and do not consider its diversity and class coverage capacities. With the increasing demands for more data and the difficulty to acquire high-quality labeled samples for training, it is important to shift the attention from the size of training data to their representativeness quality and variance coverage. This calls for efforts to dig deep in the learning of trained machine learning models to identify under-represented regions in the training sets and reveal the missing data patterns from the available training set that prevented the model from generalizing to unseen data and learning the optimal decision boundary. Then, new samples that display features characteristic of the identified under-represented regions can be selected from existing weakly labeled datasets for augmentation, after applying unsupervised filtering methods. This new approach of selective diversity-aware data augmentation alleviates the need for costly data labeling and opens up possibilities to use available weakly-annotated data repositories.

In this work, we propose a comprehensive and fully attention-based data augmentation framework to guide the process of sample selection from external image repositories. Our goal is to select relevant samples for augmentation that aim to boost the diversity and class coverage of training sets. Our proposed method is specifically tailored to be applied to ViT models. This is because Transformers have been widely used recently, particularly for computer vision tasks, and they have been the ultimate choice of deep neural network architectures for new research and experiments.

To improve the diversity of data sets, we investigate the shortcomings of ViT models trained with limited data sets. This task is achieved using the built-in attention mechanism that makes ViT models white boxes. We show that the attention scores, computed within ViT models, can provide visual insights that can explain the confusion between certain classes. These visual explanations consist of misclassification concepts that are caused by the existence of under-represented regions in the feature space of the training data. We leverage these misclassification concepts to guide our search for relevant samples from external image repositories, by computing the degree of match between them and the retrieved concepts. We carefully design a utility function that assigns a relevance score to each new sample in these image repositories. The computed utility score takes into consideration two main factors:

- Whether the new sample falls in the under-represented regions of the training data.
- Whether the new sample displays the extracted concepts that contributed to the misclassification.

Our proposed solution addresses the current limitations of existing data augmentation techniques, by considering the capacity of training sets to cover the variance of the task, instead of simply expanding its size with almost duplicate information. Our data augmentation technique enables the use of large noisy image repositories without the need for prior filtering or cleaning, which alleviates the need for carefully annotated data sets for augmentation. Additionally, our proposed method selects new samples that aim to cover the under-represented regions in the training dataset, increase its representativeness, and correct the current model’s decision boundaries. We show that our method can outperform existing data augmentation methods while adding the least number of samples.

1.3 Thesis Contributions

This thesis makes the following contributions:

1. We propose a data augmentation technique that aims beyond expanding the data in size, but instead focuses on improving their diversity by selecting informative samples with respect to their potential contributions of improving the model performance. To the best of our knowledge, no prior work has ever considered this aspect for the case of Vision Transformer models.
2. We explore the explainability potential of vision transformers and propose a novel way to utilize the attention scores for the purpose of revealing the existence of under-represented regions in the feature space of the training data.
3. Our approach explores the computed attention scores within the transformer model to get an insight on the image regions that contributed to the misclassification. These extracted image regions form misclassification concepts that explain the model limitations in a given class. We design a utility function to rank and select new samples from online image repositories based on their similarity to the extracted concepts.
4. Our developed method leverages available noisy data repositories for augmentation, thus alleviating the need for accurate (and tedious) data labeling.
5. Our proposed framework is standalone and does not require any external machine learning algorithms. It only relies on the built-in attention mechanism of the ViT model. This makes our method intuitive, simple, easy to use, and applicable to any vision transformer-based architecture that computes attention maps.
6. We conduct extensive experiments to demonstrate the problem of under-represented regions and its impact on the model performance. We also justify the formulation of the proposed utility function through ablation studies and cluster analysis.

We evaluate our approach on two different applications. The first one involves object identification in standard RGB imagery. For this application, the available online image repositories used for augmentation are noisy. The second application consists of Automatic Target Recognition (ATR) using infrared (IR) imagery. The available IR data used for augmentation is unlabeled. For this scenario, we leverage automatic detectors to generate weakly annotated datasets suitable for augmentation. For both applications, we show that the proposed scheme improves the classification performance in terms of both accuracy and robustness compared to the baseline model trained using only the available data, without augmentation.

1.4 Document Organization

In the following chapters, we start by reviewing related work in the data augmentation area and outline relevant specifics regarding vision transformers in Chapter 2. In Chapter 3, we present our proposed data augmentation method. In Chapters 4 and 5, we report a comprehensive experimental evaluation of our method designed to demonstrate its effectiveness for the RGB and Infrared applications, respectively. Finally, in Chapter 6, we summarize our findings and discuss potential future research directions.

CHAPTER 2

BACKGROUND AND RELATED WORKS

In this chapter, we provide background material that is relevant to our research. We start with a review of the commonly used data augmentation techniques. Next, we provide an overview of the Vision Transformer architecture, where we particularly focus on the key components that will be useful to build our method. Finally, we provide a brief overview of automatic target recognition, which is one of the main applications that our research aims to solve.

2.1 Data Augmentation

As deep neural networks grew larger in the last decades, there often was not enough available data to train them, especially considering that some part of the overall dataset should be spared for validation. Data augmentation is a common technique used to improve the performance of machine learning models, mainly in cases of overfitting where the training dataset is not sufficiently representative to capture the variance of the problem. The idea consists of expanding the training set in size by generating new samples.

Since their introduction, data augmentation techniques evolved and different methods have been proposed. These methods vary from simply perturbing the existing data to create new samples, synthetically generating data using generative models, or tapping into external data repositories to expand the training set. Recently, few methods focused on the quality of the augmented data to ensure its diversity.

2.1.1 Data Augmentation without External Data Repositories

Most common data augmentation techniques exploit the available dataset to create additional samples. This includes making minor perturbations to the dataset [21], such as random geometric transformation (e.g. horizontal/vertical flip, rotation, and shear), random color space transformation (e.g. brightness, contrast, saturation, and hue), or noise injection (e.g. gaussian blur). Cutout [22] and Hide-And-Seek [23] randomly mask out square regions of input during training, while Cutmix [24] replaces the removed regions with a patch from another image. Mixup [25] and Snapmix [26] generate a weighted combination of random image pairs from the training data.

These data augmentation techniques tend to apply transformations to random regions of the image, which can introduce unwanted variance, such as background noises. The work in [27]

investigates this problem and proposes an attention-based image cropping technique to crop and resize only the relevant image parts. In the same context, AutoAugmentation [28] creates a search space of data augmentation policies and automatically designs a specific policy so as to maximize the model performance. Maxdrop [29] aims to remove the maximally activated features to encourage the network to consider the less prominent features.

Despite their wide use and success in improving the model accuracy, these approaches do not take into consideration what kind of features the model has already learnt in their data augmentation scheme. Moreover, they are limited in the way that they are constrained to the current samples' neighborhood, and do not focus on specific regions of the feature space that require additional samples. Therefore, they may not lead to a significant improvement in the diversity of the dataset.

Another way to augment the data is by generating synthetic samples using deep learning algorithms, such as Generative Adversarial Networks [30] or Variational AutoEncoders [31]. Most approaches [32–34] rely on generative models to augment datasets for image classification tasks. These methods are similar in their goal to the methods mentioned earlier that generate synthetic copies. Some methods attempt to create realistic replica of few random samples. Others focus on replicating only the samples that were hard to classify by the model. Few other methods attempt to generate new samples with specific features or properties that are lacking from the current training set.

The biggest barrier for the data augmentation techniques that are based on generative models is the requirement for large data and tedious parameter tuning to train these generative models and create realistic, high-quality, and unfuzzy images for augmentation. Moreover, these methods are also bound to investigate the immediate vicinity of the input data if they only attempt to replicate their training samples.

2.1.2 Data Augmentation with External Data Repositories

Since acquiring large labeled data is difficult and costly, an interest has shifted towards online image repositories. These web datasets offer inexpensive large-scale sources of images retrieved from the internet. Despite their convenience in overcoming the shortage in data, these web datasets are often weakly-annotated and can contain out-of-distribution images. This is because constructing fine-grained image datasets typically requires domain-specific expert knowledge, which is not always available for crowd-sourcing platform annotators. The noise in web training can severely degrade the model performance. Therefore, it is prominent to establish a preprocessing or a selection system to efficiently filter and use these web datasets for the training of machine learning models.

Some research works [19, 20] investigated this direction and proposed training frameworks to learn directly from web images. For example, [19] proposed an approach focused on overcoming label noise and data bias. To this end, the authors trained two models to identify and select in-

distribution images and re-label samples with noisy labels. In the same context, [20] explored a different approach to learn from web data by transferring knowledge from existing strongly supervised datasets to weakly supervised web images. The authors took advantage of sophisticated object recognition algorithms to select relevant samples using detailed annotations from object bounding boxes and part landmarks.

These methods are able to expand the search for new samples beyond the local neighborhood of the existing training set. However, they do not guarantee the selection of relevant samples that will improve the data diversity and cover the under-represented regions.

2.1.3 Data Augmentation for the Under-Represented Regions

Few data augmentation methods [35,36], that explicitly aim to improve the diversity of data sets, were developed to address the issue of under-represented regions. For example, [35] presents a framework for augmenting the training set with only synthetically generated copies of new misclassified examples, rather than modified images coming from the original training set. These misclassified examples represent counter examples that hold features the model hasn't learnt yet. Information about the identified counterexamples are collected in error tables, that can provide explanations about the model's vulnerabilities and find recurring patterns leading to misclassification. Therefore, these error tables can be used to generate counter examples for augmentation.

The method in [35] has only been evaluated on CNN architectures on the case study of object detection. Additionally, the technique calls for an image generator, which can burden the overall algorithm complexity and requires tuning efforts to get realistic images.

Another work [36] proposes a data augmentation method, called BRACE, which uses concept-based explanations for DNN decisions to guide the data augmentation process and add informative samples based on their relevance and their capacity to cover the under-represented regions in the training set. The authors extract model explanations, expressed in different forms, such as saliency maps or linguistic explanations generated by post-hoc methods (e.g. GradCam [37]) or interpretable models (e.g. Comprehensive CNN [38]). The main idea is to utilize the retrieved explanations to analyze the model's performance and reveal the causes leading to the misclassification. Then, BRACE compares object parts of new samples, extracted from pretrained object detectors (e.g. RCNN [39]), to the extracted misclassification concepts and selects the samples that have the highest match to augment the training data.

Similar to the work in [35], BRACE has only been evaluated on CNN architectures. Additionally, the authors in [36] relied on post-hoc explanations methods, specifically GradCam [37], for the purpose of generating fine-grained explanations to find visual cues justifying the misclassifications. Their methods also calls for pretrained object detectors. We argue that using external fixed machine learning algorithms, that are agnostic to the task and data in hand, not only increases the

method’s complexity, but can also introduce a significant margin of error.

In contrast to all approaches mentioned above, our work is self-contained and does not require any additional explanation algorithms, as it relies entirely on the attention map computed within the transformer model. Additionally, our work represents the pioneering effort in addressing the matter of under-represented regions within vision transformers.

2.2 Vision Transformers

Up until recently, learning tasks involving textual data relied primarily on attention mechanisms integrated with Recurrent Neural Networks (RNN) [40] as the predominant architecture. However, with the first emergence of Transformer models in [41], the attention of the scientific community shifted towards the widely adopted Transformer models that we see today, thus supplanting RNN models like long short-term memory [42].

A Transformer is a deep learning model that utilizes the self-attention mechanism, assigning varying importance to different segments of the input data. This model type is becoming increasingly favored for addressing Natural Language Processing challenges [43–45], paving the way for more crucial and interesting research, such as Large Language models (e.g. GPT by OpenAI [46] and BERT by Google [47]). The success of transformers resides in their ability to encode long-range dependencies between all tokens of the same input, making it possible to capture global and local interactions leading to the formation of an input representation with contextual relevance.

Inspired by the work in [41] that first presented the transformer in the context of machine translation, Dosovitskiy et al. proposed the Vision Transformer (ViT) [1], a concept that restructures the conventional transformer to process visual data effectively. This adaptation can be fully justified as upon closer examination, the fields of NLP and Computer Vision (CV) reveal several notable high-level similarities. First, as sentences are sequential, an image can also be considered as a sequence of smaller image patches. Additionally, as the meaning of a word can often be fully understood only by relating it to the other words in the sentence, it may be argued that individual image parts need to be contextualized with the broader image in order to be fully disambiguated. Consequently, it is reasonable to anticipate that the long-range self-attention models used in NLP would prove highly effective in modeling visual data.

Similar to Transformers outperforming classic sequential models for text data, ViT demonstrated its effectiveness over Convolution Neural Networks (CNNs), for several visual tasks, such as image classification [48], image captioning [49], image segmentation [50], object detection [51], autonomous driving [52], ...etc. Moreover, ViTs have been applied to generative modeling and multi-model tasks [53], including visual grounding [54], visual-question answering [55], and visual reasoning [56]. The emergence of Vision Transformers has also provided an important foundation for developing larger vision models [15–17,51] that made significant contributions to the state of the

art.

2.2.1 Limitations of Convolution Neural Networks

Given the increasingly widespread use of transformer architecture, having a comprehensive understanding of the fundamental disparities between ViTs and CNNs is crucial. The major difference is that CNNs rely mainly on their inherent inductive biases, which are local connectivity and translation equivariance [57]. These knowledge priors help escalate the network learning on relatively small datasets, but can quickly saturate the learning with larger amounts of data in CNNs. ViTs lack the majority of the CNN’s inductive biases, and are, therefore, able to learn more rules and better-quality intermediate representations with more data.

Additionally, ViTs are able to retain more spatial information than certain CNN architecture, such as Residual Neural Networks [58]. Since the convolution operation uses receptive fields with a fixed size, convolution kernels can only capture short-range spatial information that cannot model dependencies beyond the initial receptive field. Transformers, on the other hand, are able to capture long-range dependencies both locally and globally, by looking at all spatial locations and modeling dependencies between all of them, much beyond the receptive field of convolution filters.

Finally, despite the significant advances in hardware and the common use of GPUs in training neural networks, CNNs remain costly in terms of computation, especially when applied to high-resolution images. A complexity analysis in [1] showed that transformers ensure faster training and inference compared to CNNs.

2.2.2 Overview of the Vision Transformer Architecture

An overview of the ViT model is shown in Figure 2.1. ViTs represent images as sequences. A sequence can be created from an image $x \in \mathbb{R}^{H \times W \times C}$, where (H, W) is the spatial resolution of the original image and C is the number of channels, by first splitting the image into smaller patches of fixed size (P, P) . Then, every patch is flattened into a single vector by concatenating the channels of all pixels in a patch. This results in a sequence of patches $x_p \in \mathbb{R}^{N \times (P^2 \cdot C)}$, where $N = HW/P^2$ is the resulting number of patches, which also serves as the effective input sequence length for the Transformer. The ViT uses a constant latent vector size D through all its layer. Therefore, flattened image patches are linearly projected into D dimensions to form patch embeddings, using a learnable embedding matrix E (Equation (2.1)).

As shown in Figure 2.1 and Equation (2.1), positional encodings are added to the patch embeddings to retain positional information, by means of a learnable position embedding matrix E_{pos} . To help with the classification, the authors in [1], inspired by the original BERT paper [47], prepended an extra learnable embedding, called classification token and denoted as $[CLASS]$, to the sequence of embedded patches. This added token serves as a representation of the global image.

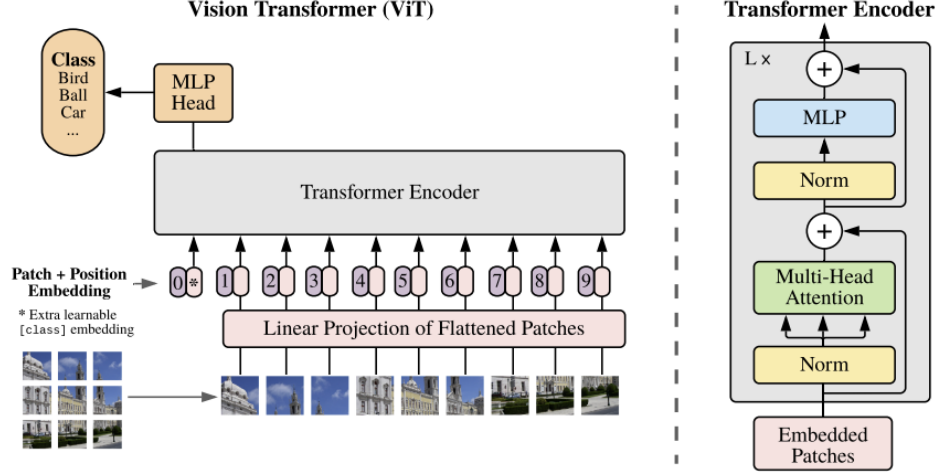


Figure 2.1: An overview of the ViT architecture [1].

The resulting sequence of $(N + 1)$ embedding vectors is fed the transformer encoder.

The encoder block is identical to the original transformer proposed in [41]. It consists of alternating layers of Layer Normalization (LN), Multi-head Self-Attention Network (MSA), and Multi-Layer Perceptrons (MLP) blocks. The LN block keeps the training process stable and adjusts the model’s weights to the variations among the training images. The MSA block is responsible for generating attention maps from the given embedded visual tokens. These attention maps help the network focus on the most critical regions in the image. MLP is a two-layer classification network with GELU activation. The final MLP block appended on top of the model is the classification head, which is attached to the $[CLASS]$ token to predict the image class.

To summarize, let L be the number of layers in the transformer encoder, E a learnable embedding matrix for the linear projection of patches, and E_{pos} a learnable embedding matrix for the positional encoding of patches. Given an input X split into N patches x_p , the transformer performs the following steps:

$$z_0 = [x_{CLASS}; x_p^1 E; x_p^2 E; \dots; x_p^N E] + E_{pos}, \quad E \in \mathbb{R}^{(P^2 \cdot C) \times D}, E_{pos} \in \mathbb{R}^{(N+1) \times D} \quad (2.1)$$

$$z'_l = MSA(LN(z_{l-1})) + z_{l-1}, \quad l = 1 \dots L \quad (2.2)$$

$$z_l = MLP(LN(z'_l)) + z'_l, \quad l = 1 \dots L \quad (2.3)$$

$$y = MLP(z_L^{[CLASS]}) \quad (2.4)$$

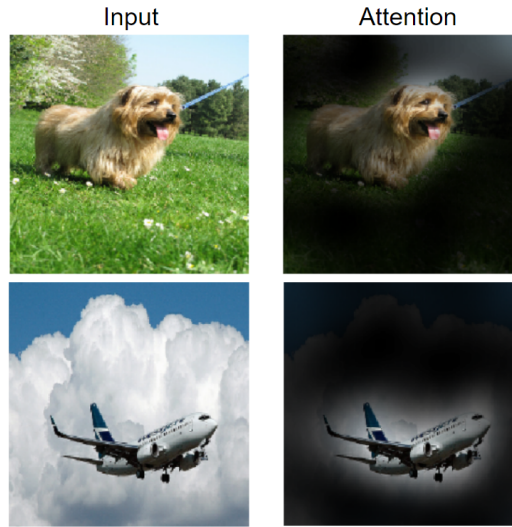


Figure 2.2: Representative examples of attention from the output token to the input space [1].

Finally, since ViTs lack some of the inductive biases of CNNs and can benefit more from training on large amounts of data, a common practice has been to pretrain the ViT model on huge datasets of images. Then, the pretrained models can be finetuned on downstream datasets for image classification.

2.2.3 Self-Attention

Self-attention [41] is the most important building block of the transformer network. The main goal of the self-attention module is to generate an input representation that captures global and local dependencies between all tokens of the input. In practice, a self attention module takes in N input tokens and returns N outputs, which represent new embeddings of the inputs. These embeddings are the result of a self-attention mechanism that allowed the inputs to interact with each other ("self") and capture their inter-dependencies ("attention") to generate an embedding for each token that encodes its contextual relevance. For example, in Figure 2.2 we show how self-attention enables the ViT model to attend to image regions that are semantically relevant to the classification task.

Scaled Dot-Product Attention

The attention mechanism used in the Transformer uses three variables to represent each input X : Query (Q), Key (K), and Value (V), computed as follows:

$$Q = XW_Q; K = XW_K; V = XW_V$$

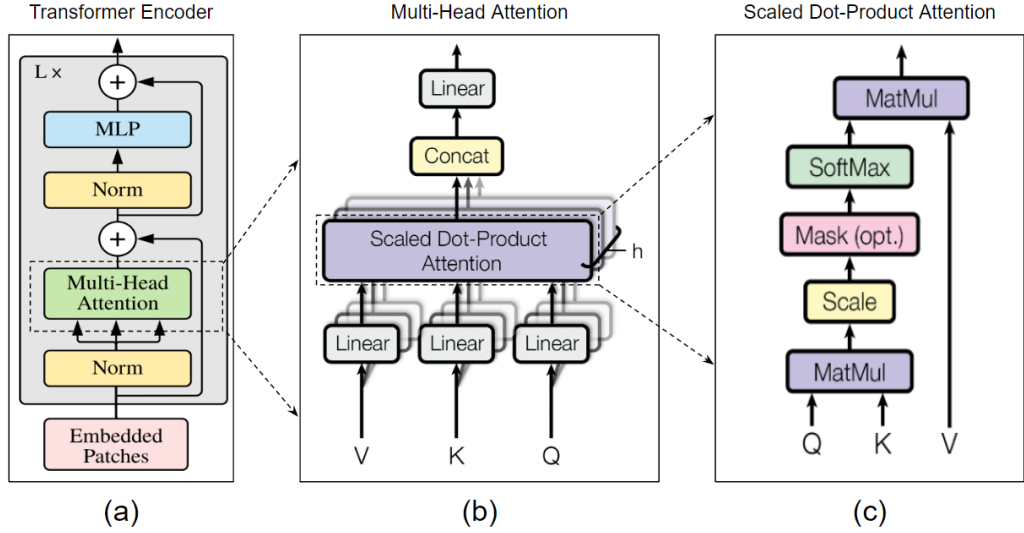


Figure 2.3: (a): Architecture of a transformer encoder. (b): Architecture of a multi-head self-attention block. (c): Components of a single self-attention head featuring the scaled-dot product attention.

Where the input is $X \in \mathbb{R}^{N \times D}$, and the projections are learnable parameter matrices $W_Q \in \mathbb{R}^{D \times d_q}$, $W_K \in \mathbb{R}^{D \times d_k}$, $W_V \in \mathbb{R}^{D \times d_v}$. N is the number of input tokens, D is the constant latent vector size used by the model through all its layers, and d_q , d_k and d_v are the sizes of W_Q , W_K , W_V , respectively. Therefore, $Q \in \mathbb{R}^{N \times d_q}$, $K \in \mathbb{R}^{N \times d_k}$, and $V \in \mathbb{R}^{N \times d_v}$.

Formally, the attention function calculates the association (attention weight) between the Query token and the Key token and multiplies the Value associated with each Key. This computation is based on a Scaled-Dot Product [41] operation (Figure 2.3, (c)). The matrix of outputs is computed as follows:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (2.5)$$

In Equation (2.5), $softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)$ is an $N \times N$ matrix called *Attention Map*, and represents relevance scores of each token with respect to all other tokens. The attention output is computed as a weighted sum of the values, where the weight assigned to each value is taken from the attention map. Therefore, the final attention output is a new embedding of the input that captures the relevant interactions between tokens, necessary to learn a given task.

2.2.4 Multi-Head Attention

It has been shown [41] that, instead of performing one single attention, it was more beneficial to have h scaled dot-product attention heads in order to capture different patterns in the dataset. Defining the Q, K, and V calculation as a single head, the multi-head attention mechanism simply uses different projection matrices W_Q , W_K , W_V for each head. The attention function is performed

in parallel for each of the h heads. Then, the resulting values are concatenated and once again projected (Figure 2.3, (b)). The multi-head attention can be formulated as follows:

$$MultiHead(Q, K, V) = Concat(head_1, \dots, head_h)W^O$$

where

$$head_i = Attention(Q_i, K_i, V_i),$$

and $W^O \in \mathbb{R}^{h \cdot d_v \times D}$ is a learnable parameter matrix.

The intuition behind multi-head self-attention is to allow the model to better capture positional information by having each head attend to different parts of the sequence each time. This means that each head will also capture different contextual information by uniquely correlating tokens. Consequently, the combination of the heads can generate a more robust representation of the input.

2.3 Automatic Target Recognition

Over the last few years, a strong interest has been raised around deep learning-based approaches to develop Automatic Target Recognition (ATR) systems using infrared images. Various techniques have been proposed to address the challenges associated with this task, including feature extraction, transfer learning, and data augmentation.

Most early work [59] adopted a model-based learning approach that relies on engineered feature extraction followed by classification. For instance, the authors in [60] used Scale Invariant Feature Transform (SIFT) [61] features, while [62] used Histogram of Oriented Gradients (HOG) [63] features followed by a Support Vector Machine (SVM) [64] classifier. These methods often rely on domain-specific knowledge and are not optimized for end-to-end training, limiting their performance on complex ATR tasks. More recently, deep learning-based methods have shown promise for ATR from infrared images. For instance, Nasrabadi et al. [65] proposed a framework with two deep Convolution Neural Networks (CNNs) to both localize the target and recognize its class while rejecting false alarms. Another recent study [66] proposed a fully-connected CNN and was shown to outperform more complex state-of-the-art CNN architectures when trained on a synthetically generated IR dataset. In another related approach [67], the authors proposed a multistage ATR system that performs target detection by localizing the hot spots, and target identification using a CNN. In [68], an ensemble method, which uses multiple classifiers in a tree-structured framework, was proposed. To address the problem of infrared variation, an IR variation reduction block CNN (IVO-CNN) was proposed in [69]. Transfer learning has also been explored as a means to address the lack of labeled data for infrared ATR. For example, Hu et al. [70] used a pre-trained CNN on the ImageNet dataset for feature extraction from infrared images. Wang et al. [71] proposed a transfer learning approach

for ATR from IR images. They used a pre-trained CNN on a large-scale RGB image classification dataset and fine-tuned it on the IR image dataset. They also used a novel region-based attention mechanism to highlight discriminative regions in the IR images. Despite their promising potentials, such approaches are limited by the availability of suitable pre-trained models and may not generalize well to diverse ATR tasks.

Data augmentation is another popular technique for addressing the low-data regime problem in ATR. For instance, Zhang et al. [72] proposed a method to generate synthetic infrared images by applying geometric and photometric transformations to existing labeled data. Likewise, Zheng et al. [73] proposed a data augmentation approach for ATR from IR images that generates synthetic images by applying geometric transformations and adding noise. Similarly, Li et al. [74] proposed an augmented training approach that generates synthetic IR images with random backgrounds, thermal signatures, and orientations. However, such augmentation techniques are often limited by the diversity of the original labeled dataset and may not capture the full range of variability in the target objects.

Recently, few efforts explored transformer models for ATR systems with IR images. For example, Zhao et al. [75] proposed a few-shot ATR system based on an instance-aware transformer that exploits the power of all instances to build a more robust input representation. On the other hand, Ethan et al. [76] focused on target detection and proposed an Edge IR Vision Transformer (EIR-ViT) for automatic target detection utilizing infrared images, that is lightweight and operates on the edge for easier deployability.

Despite these efforts, there remain significant challenges in developing effective ATR methods for infrared images. These include the lack of color information, difficulty in acquiring high-resolution labeled data, and sensitivity to environmental conditions. These factors contribute to the creation of under-representative training sets with large gaps in their feature space that inhibit the development of robust and generalizable ATR systems.

In this research, we build an ATR system based on a ViT model. We address the issue of under-represented regions in the feature space of the IR training sets by applying our proposed data augmentation method to add relevant samples that increase the class coverage in the available training set. We show that the model trained on the augmented set yields better performance and the obtained ATR system is more robust to the variance of the data.

CHAPTER 3

ATTENTION-GUIDED DATA AUGMENTATION FOR IMPROVING THE CLASSIFICATION PERFORMANCE OF VISION TRANSFORMERS

In this chapter, we present our Attention-Guided Data Augmentation (ADA-ViT) method, designed and developed to improve the performance of Vision Transformer (ViT) models. Our work addresses the particular problem of training with datasets that are not sufficiently representative to cover the large intra-class variations, thus leading to under-represented regions in their feature space. The presence of gaps in the training feature space leads the model to learn sub-optimal decision boundaries and prevents it from generalizing well to unseen data.

The standard way of training machine learning models consists mainly of offline training and online inference on unseen test data. As illustrated in Figure 3.1, we propose a different learning pipeline that extends the standard approach, by adding an augmentation block within the offline training loop. The added component is automatically integrated in the training process and does not need manual setting.

Our augmentation framework leverages feedback from the model performance on a held-out validation set to understand the current model vulnerabilities and reveal the presence of under-represented regions in training feature space. Then, we use this information about the model limitations to guide the search in external image repositories for new samples that can cover the identified gaps in the training set, and potentially improve its performance. Finally, we finetune the trained model, for few epochs, on the new training set, consisting of the initial dataset and the new selected samples. We repeat these steps for a certain number of iterations, until no significant improvement is observed.

Our work is based on two main assumptions. First, we assume that the model performance on the validation set is an accurate estimation of the true performance on the unseen test set [77]. In other words, we assume that the validation and test sets have been extracted similarly and thus, are drawn from the same distribution. Second, we assume that we have access to external image repositories, that are large in scale and diverse. To guarantee these criteria, we typically utilize web datasets obtained from scraping websites for data extraction. Although this method of obtaining data for augmentation is cheap, fast, and guarantees diversity, these web datasets are poorly annotated and contain noisy and out-of-distribution images. ADA-ViT is able to handle noisy datasets since it selects new samples for augmentation based on a ranking system using a utility function

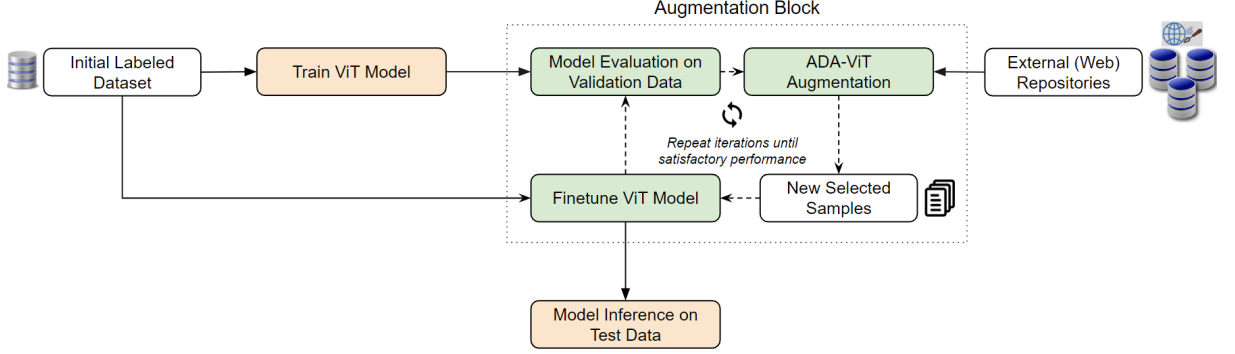


Figure 3.1: Proposed learning approach that integrates data augmentation in the learning process.

that quantifies the relevance of the new sample and its potential contribution to improving the data diversity and model performance. More specifically, ADA-ViT ranks new samples by computing and aggregating three major scores: a label regularization score α , an under-representation score β , and a degree of match Δ . In the following sections, we describe in details these scores and their importance on the overall utility score to select relevant candidate samples for augmentation.

3.1 Overview of ADA-ViT algorithm

Starting from an initially trained ViT model T , ADA-ViT adds new samples to a given class c from the pool of online candidate images by considering two main factors:

- Whether the new sample x falls in the under-represented region of the feature space of class c that caused the misclassification as class \bar{c} by T . This characteristic is described by the under-representation score $\beta(x, c, \bar{c})$ (Section 3.4).
- Whether the new sample x displays similar visual features to $S_{c \rightarrow \bar{c}}$, which is the set of concepts that led to the misclassification of class c as \bar{c} , extracted from the validation set. This characteristic is described by the degree of match $\Delta(S_{c \rightarrow \bar{c}}, x)$ (Section 3.5).

Both $\beta(\cdot)$ and $\Delta(\cdot)$ scores are aggregated to form a utility function as follows:

$$utility(x) = \alpha(x, c) \times \sum_{\bar{c} \in \bar{C}} [\beta(x, c, \bar{c}) \times \Delta(S_{c \rightarrow \bar{c}}, x)]. \quad (3.1)$$

In Equation (3.1), $\alpha(\cdot)$ is a label regularization score that intends to penalize new samples with noisy labels in the external web repository (Section 3.3).

In each class c , the new samples are ranked based on the utility function defined in Equation (3.1), and the top $N(c)$ samples are selected to augment that class. $N(c)$ is computed using:

$$N(c) = \frac{\sum_{x \in D_c^v} \delta(T(x) \neq c)}{|D_c^v|} \times |D_c^t|, \quad (3.2)$$










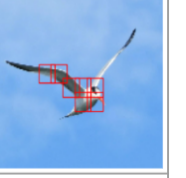
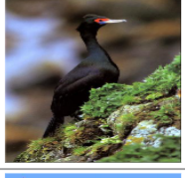


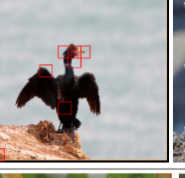






Class	Sample training image	Misclassified validation image	Top 3 added web images		
<i>Mangroove_Cuckoo</i>					
<i>Elegant_Tern</i>					
<i>Red_Faced_Cormorant</i>					
<i>Red_eyed_Vireo</i>					

Figure 3.2: Examples of selected samples by ADAViT for CUB dataset. In the third column, we highlight the patches responsible for the misclassification of the validation image. In the fourth column, we highlight the patches on the new image that increased its corresponding utility score the most, leading to its selection for augmentation.

where D_c^v is the validation subset for class c , D_c^t is the training subset for class c , and δ is a function that returns 1 if its argument is true, otherwise it returns 0. $N(c)$ is computed to be proportional to the ratio of misclassification in class c . Intuitively, we want the augmentation to be larger for the classes with higher misclassification rates.

The proposed ADA-ViT approach is summarized in Algorithm 3.1. In Figure 3.2, we show examples of selected web samples by ADAViT for augmentation. As designed, the selected samples (those with top utility scores) are similar to the misclassified images from the validation set. This is because ADA-ViT considers the validation set an indicator of the true performance of the model on the unseen data. Therefore, it aims to augment the training data with new images that share common features with the hard samples from the validation set that have been previously misclassified by the model.

Algorithm 3.1 Attention-Guided Data Augmentation for Vision Transformers

Input Trained ViT model T , Original training dataset D^t , Class labels C
Output Finetuned ViT model T'

```
1:  $T' \leftarrow T$  ▷ initialize the new weights
2: for iter  $\in [0, max\_iterations]$  do
3:    $D' \leftarrow D^t$  ▷ initialize the new training dataset
4:   for  $c \in C$  do
5:      $N(c) \leftarrow$  compute the number of images to add
6:      $X_c^{new} \leftarrow$  Subset of new images from class  $c$ 
7:     for  $x \in X_c^{new}$  do
8:       Compute  $\alpha(x, c)$ 
9:       for  $\bar{c} \in \bar{C}$  do
10:        Compute  $\beta(x, c, \bar{c})$ 
11:        Compute  $S_{c \rightarrow \bar{c}}$ 
12:        Compute  $\Delta(S_{c \rightarrow \bar{c}}, x)$ 
13:      end for
14:      Compute  $utility(x)$ 
15:    end for
16:     $D_{aug}^{new} \leftarrow N(c)$  samples with highest utility scores
17:     $D' \leftarrow D' \cup D_{aug}^{new}$ 
18:  end for
19:   $T' \leftarrow$  finetune  $T'$  on  $D'$ 
20: end for
```

3.2 Similarity Function ($\varphi(\cdot)$)

In the computation of the under-representation score β (Section 3.4) and the label regularization score α (Section 3.3), we need to quantify the degree to which a new sample displays common features with the training data from a class. To do so, we design a similarity function that can reliably quantify the shared features while taking into consideration the issue of high intra-class variability that results in under-represented regions in the training feature space.

First, we use clustering to categorize the different training data points of a class c into compact and homogenous groups. At this stage, any clustering algorithm can be used. Then, we represent the class c with the medoids of its clusters. The medoid of a cluster is selected to represent each cluster. It corresponds to a sample within the cluster that is considered the most representative one. This ensures that a new sample will be compared to a realistic image.

Finally, we compute the cosine similarity between the feature of a new sample x and the feature of each cluster medoid, and consider the maximum similarity. Formally, we compute the similarity function, φ , between a new sample x and the training data of a given class c as follows:

$$\varphi(z_x^{new}, Z_c^t) = \max_{k \in K} Sim_{cos}(z_x^{new}, z_{medoid,k}^t). \quad (3.3)$$

In Equation (3.3), K characterizes the obtained clusters from the training data, z_x^{new} is the feature vector of the new sample x , and $z_{medoid,k}^t$ is the feature vector of the medoid of cluster k .

3.3 Label Regularization Score ($\alpha(\cdot)$)

In our work, we assume we have access to large external image repositories, on which we perform sample selection for augmentation. Web datasets are relatively easy and cheap to acquire, while being large and diverse. However, web data repositories are typically noisy and weakly-annotated. Therefore, it is possible to have new samples that are incorrectly labeled. In the case of datasets with fine-grained classes where the inter-class variation is low, samples with noisy labels may be wrongfully selected to augment the incorrect class, which worsens the model confusion. We address this issue of label noise in the external web datasets by introducing a penalty term, $\alpha(x, c)$, defined for a new web sample x with web label c , using:

$$\alpha(x, c) = \frac{\varphi(z_x^{new}, Z_c^t)}{\max_{c_i \in C} \{\varphi(z_x^{new}, Z_{c_i}^t)\}}. \quad (3.4)$$

In Equation (3.4), $\varphi(\cdot)$ is the similarity function defined in Section 3.2, z_x^{new} is the feature vector of x , and Z_c^t is the set of feature vectors of all training examples from class c .

$\alpha(x, c)$ compares the similarity between x and the available training data of class c to its similarity to all other classes. If x is more similar to the training data of a different class $c_i \neq c$ than the training data of class c , then it is possible that x has been incorrectly labeled in the web repository, and the image x will be penalized by $0 \leq \alpha \ll 1$. In this case, the utility score of the new sample is low, decreasing its chances of being selected for augmentation. On the other hand, if x was correctly labeled as class c , then its similarity with the training data of class c should be maximum (or close to the maximum), resulting into higher values of α ($\alpha \approx 1$), and thus little to no penalty. In this case, the utility score will remain almost unaffected for this particular new sample. Figure 3.3 displays the penalty term of few web images. As it can be seen, α is close to 1 when the web label and the true label are the same.

3.4 Under-representation Score ($\beta(\cdot)$)

The first part of the sample selection process adopted by ADA-ViT consists in finding new data points that fall in the under-represented regions of the feature space. We define an under-represented region as a region in the training feature space occupied by some test samples that share global visual similarities with the training data from the same class, and yet the model misclassifies them with high confidence. Based on this definition, we formulate the under-representation score, β , for a new sample x and for a given class c , as follows:

$$\beta(x, c, \bar{c}) = \varphi(z_x^{new}, Z_c^t) \times P(T(x) = \bar{c}|x). \quad (3.5)$$

In Equation (3.5), $\varphi(\cdot)$ designates the similarity function defined in Equation (3.3), z_x^{new} the feature vector of x , and Z_c^t the set of feature vectors of all training samples from class c . ADA-ViT represents

	Sample image	Web Label	True Label	Similarity with Web Label	Maximum Similarity	α
(1)		Black Footed Albatross	Black Footed Albatross	0.7	0.7 (Black Footed Albatross)	1.0
(2)		Indigo Bunting	Indigo Bunting	0.67	0.7 (Blue Grosbeak)	0.96
(3)		Northern Fulmar	Gray Kingbird	0.01	0.6 (Gray Kingbird)	0.02
(4)		Crested Auklet	N/A (Class absent from the dataset)	0.1	0.5 (Clark Nutcracker)	0.2

Figure 3.3: Examples of label regularization cases. The score α is highest when the web label is correct (examples (1) and (2)), and is lowest when the web label is incorrect (examples (3) and (4)).

an input image by extracting the embedding of the corresponding classification ($[CLS]$) token from the last self-attention layer (refer to Section 2.2.2, Equation (2.3), since this added token is intended to learn a representation for the global input image. Finally, $P(T(x) = \bar{c}|x)$ is the prediction probability of the model T in the wrong class \bar{c}

The first component in Equation (3.5) has a penalty effect on out-of-distribution samples, while the second component aims to assign lower scores to samples that have already been correctly classified by the model, and thus will not add relevant information in the new augmented training set. Therefore, a high β score means that the new sample x shares global common features with the training data from a given class, and yet it was misclassified by the model T with a high confidence. These samples fall in the under-represented regions of the training data, because they display a set of rules that the current model T hasn't been able to learn yet due to a shortage in such examples in the current training set.

3.5 Degree of match ($\Delta(\cdot)$)

The second part of the sample selection process adopted by ADA-ViT consists of finding new samples that display specific fine-grained visual features, similar to those found in samples from the validation set that were misclassified by the current model T . At this stage, we dive deep into the

model’s learning, thanks to the attention mechanism of vision transformers, and try to understand its vulnerabilities by studying its performance on the validation set, assuming that this analysis provides an accurate estimation of the model’s behaviour on the unseen test set.

First, we investigate the model limitations on the validation set by studying the misclassified samples. We extract a set of concepts that justify the model confusion between classes (Section 3.5.1). Then, we search in the external image repositories for new samples displaying features that match the identified misclassification concepts. We compute the score Δ to quantify the degree of match between the new samples and the misclassification concepts (Section 3.5.2).

3.5.1 Identification of Misclassification Concepts

Let $S_{c \rightarrow \bar{c}}$ be the set of visual features extracted from the validation images of class c misclassified as class \bar{c} . $S_{c \rightarrow \bar{c}}$ represents a set of concepts that justify the model confusion of class c as \bar{c} . To form $S_{c \rightarrow \bar{c}}$, we need to retrieve visual explanations of the model’s prediction decisions. To do so, ADA-ViT relies only on the attention map, already computed within the transformer model, by considering the attention weights computed at the self-attention layer as relevancy scores for the individual image patches.

Figure 3.4 motivates our use for the attention weights to retrieve visual explanations of the model decisions. In this figure, we display an example of a female *Painted Bunting* bird that has been misclassified as an *Orange Crowned Warbler* bird. Female birds of the *Painted Bunting* specie are characterized by their uniform green-yellow color [78], while the males are multicolored (blue head, green back and red belly). *Orange Crowned Warbler* are typically green-yellow [78]. The image regions with the highest attention scores are focused on the bird’s green-yellow body (Figure 3.4, b). A scan of the training dataset shows that there is only one image of a female Painted Bunting, while the majority of the training samples for that particular class were images of the male bird. Therefore, the attention mechanism revealed that this image of a bird was misclassified because it displayed atypical visual features, according to the available training data, for that particular class. We can conclude that the self-attention component can provide concrete and direct visual insight about the image patches that are crucial to the final model decision. Consequently, it is possible to acquire visual explanations as to what might have caused misclassifications by simply looking at the attention weights.

We form $S_{c \rightarrow \bar{c}}$ by proceeding as follows. Let $X_{c \rightarrow \bar{c}}^v$ be the set of validation images from class c misclassified as \bar{c} . For each image $x \in X_{c \rightarrow \bar{c}}^v$, we retrieve its corresponding attention maps computed at all of the h attention heads and all of the L layers in the transformer model (refer to Equation (2.5)). Then, we combine the multiple attention maps using Attention Rollout [79], which is a technique that computes maps of the attention from the output token to the input space. Briefly, Attention Rollout works by taking the maximum attention weights of the ViT across all heads

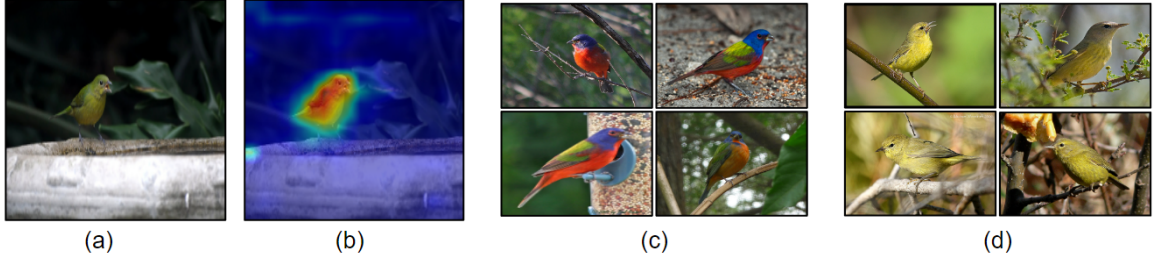


Figure 3.4: (a): An image of a female *Painted Bunting* misclassified as an *Orange Crowned Warbler*. (b): The heatmap of the attention map superimposed on the original image. (c): Typical images of *Painted Bunting* in the training set (mostly images of the male bird). (d): Example images of *Orange Crowned Warbler* in the training set.

and then recursively multiplying the weight matrices of all layers. This accounts for the mixing of attention across tokens through all layers. Therefore, for each image x , we obtain a single attention map, $A = \{a_{i,j}\}_{1 \leq i,j \leq N}$, that captures the total attention flow between the N image patches across the different transformer layers.

Next, we select the most important image patches, p , with attention scores $a_p > Q_q$, with Q_q corresponding to the q^{th} quantile computed on the total attention scores in A . Finally, $S_{c \rightarrow \bar{c}}$ is formed by retrieving the features z_x^p corresponding to the identified relevant patches p for image x . Formally, $S_{c \rightarrow \bar{c}}$ is defined as follows:

$$S_{c \rightarrow \bar{c}} = \cup_{x \in X_{c \rightarrow \bar{c}}^v} \{z_x^p | a_p > Q_q\} \quad (3.6)$$

Among the advantages of using patch-level explanations to form $S_{c \rightarrow \bar{c}}$ is the possibility to control the granularity of the extracted concepts. Depending on the patch size used in the ViT model, it is possible to acquire fine-grained or coarse model explanations that justify the class confusion. Moreover, by setting the quantile threshold Q_q to higher values, the misclassification concepts become more relevant and concise.

3.5.2 Computation of Δ

At this stage, we have defined $S_{c \rightarrow \bar{c}}$ for each class c and \bar{c} to encompass the set of semantic concepts responsible for the misclassification of class c as \bar{c} . Next, we select new images from external image repositories, for each class c that have visual features similar to the concepts extracted in $S_{c \rightarrow \bar{c}}$. That is, we augment the dataset with images that exhibit certain visual characteristics which previously confused the model into learning the wrong rules.

We compute the degree of match between each concept $s \in S_{c \rightarrow \bar{c}}$ and each new image $x \in X_c^{new}$, where X_c^{new} is the set of new images from class c . Since the concepts from $S_{c \rightarrow \bar{c}}$ describe individual image patches with high importance, we adopt a similar approach on the new samples by representing a new image with a set of patches that have the highest attention scores. We do this

to facilitate the matching between the concepts in $S_{c \rightarrow \bar{c}}$ and the new images, by making it a patch-to-patch comparison instead of patch-to-image. Additionally, by selecting only few image regions for comparison, instead of using all the image patches, we significantly speed-up and simplify the matching computation, by disregarding irrelevant image regions that the model has not considered while making its decision.

Similar to what was described in Section 3.5.1, we retrieve the attention maps corresponding to a given new image, and join them using Attention Rollout [79]. Then, we identify the most important patches in the new image that have attention scores above a certain threshold. We note that we relax the quantile threshold for new images compared to the one we used to form $S_{c \rightarrow \bar{c}}$. This is because, unlike the misclassification concepts which should be as fine-grained as possible to locate minor details causing the model confusion, we would like to select as many patches as possible for new images, to compare with the misclassification concepts and have potentially higher matches.

For each new image x , we compute the degree of match as follows:

$$\Delta(S_{c \rightarrow \bar{c}}, x) = \frac{1}{|S_{c \rightarrow \bar{c}}|} \sum_{s \in S_{c \rightarrow \bar{c}}} w_s \quad (3.7)$$

where,

$$w_s = \max_{z_p \in Z_x^{new}} (-\log[1 - \text{sim}_{\cos}(s, z_p)] + \epsilon) \quad (3.8)$$

In Equation (3.8), $|\cdot|$ denotes the cardinality function that returns the number of elements in $S_{c \rightarrow \bar{c}}$, sim_{\cos} is the cosine similarity function, $Z_x^{new} \in \mathbb{R}^{P \times D}$ is the set of feature vectors of the P identified important image patches in the new image x , and ϵ is a constant parameter set to a low value to avoid undefined values for the logarithm.

For each new image $x \in X_c^{new}$, we compute w_s to find the highest degree of match between some region (image patch) in x and a concept $s \in S_{c \rightarrow \bar{c}}$. A large value of w_s indicates the presence of the concept s in x . Therefore, the new sample x is likely to be selected. As suggested in [36], we also use a negative log-likelihood to favor samples that have few matching concepts with high w_s scores over samples that have several matching concepts with lower w_s scores.

Figure 3.5 presents a comprehensive overview of the proposed ADA-ViT framework that summarizes the different steps to compute a utility score for a new sample.

3.6 Vision Transformers with Adaptive Data Augmentation

In the final step of our proposed approach, we finetune the weights of the initially trained ViT model on the new augmented training set. The new training set consists of the selected new samples from the external image repositories, ranked based on their utility scores, as well as the original training set used to train the initial model. We use the validation set for hyperparameter tuning and model selection during the original training and the subsequent model finetuning. We

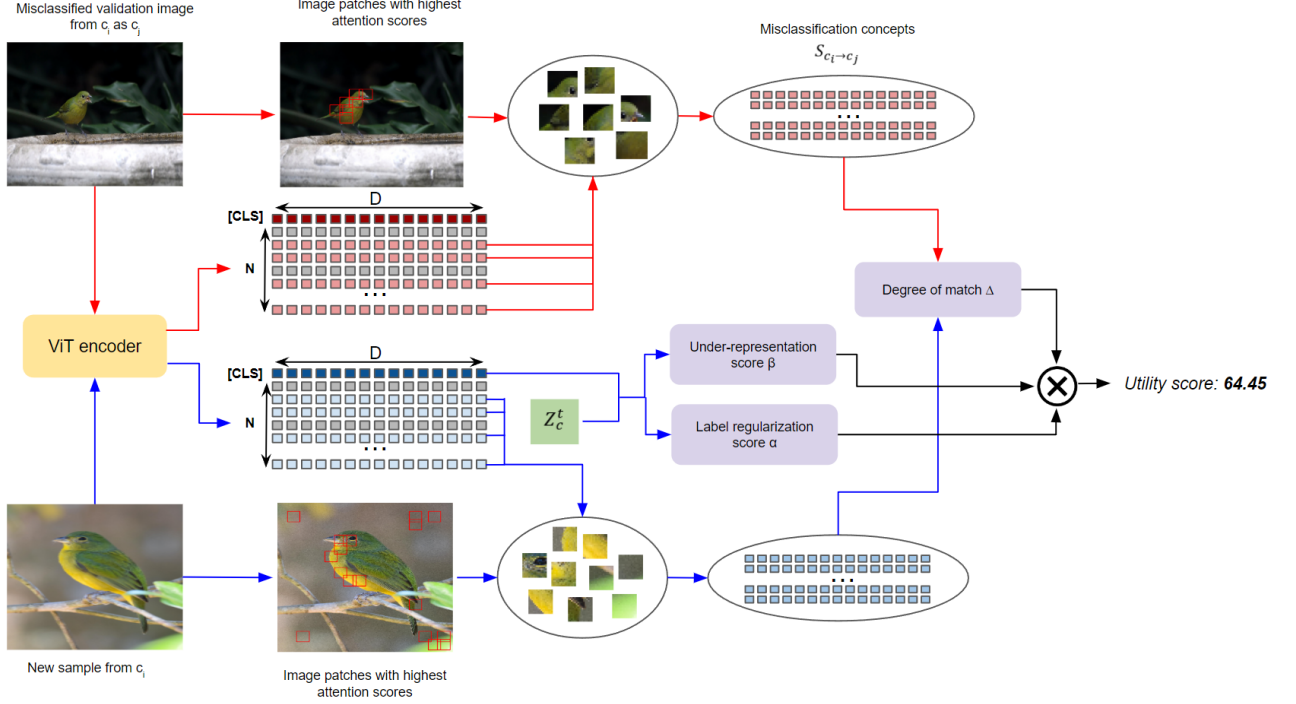


Figure 3.5: Overview of the proposed ADA-ViT framework. For simplicity, we assume there is only one validation image from c_i misclassified as c_j . N is the number of image patches, D is the ViT’s embedding size, Z_c^t is the set of training feature vectors representative of class c , and [CLS] designates the classification token. Arrows in *red* describe the algorithm flow for the validation data, while arrows in *blue* describe the algorithm flow for the new web sample. This figure is best viewed in color.

iterate this process until we achieve the maximum number of iterations or when the model accuracy plateaus, meaning that the model performance is not improving anymore.

It is necessary to include the original training data in the new augmented set for several reasons. First, we only select a very small set of images that is not sufficient to train or even fine-tune a model. Second, adjusting the weights of the trained model on the new selected samples only can lead to overfitting, since we add only few samples from the external dataset in each iteration, and these samples display specific features that were lacking from the original training set, which is not enough to train a ViT model and can seriously degrade the original model learning. Finally, excluding the original training set can lead the model to forget the already learnt features and significantly change the decision boundary. Instead, the goal is to only adjust the weights to account for certain features and data patterns that were previously absent in the training set.

There are certain training tricks that we used in our approach to control the model weight adjustment during finetuning, as to prevent overfitting or severe decision boundary shifts. First, we only finetuned the model for few epochs, since the original model has already converged. Second, we used pretrained weight decay [80], which is a regularization technique proposed for BERT models that penalizes strong changes of the finetuned weights from the original pretrained weights. This

technique works better than conventional weight decay in the case of transformer models and helps stabilize the finetuning. Finally, we applied attention dropout [81] which is a regularization technique for transformer models, that works by randomly dropping out elements from the attention map. This has the effect of excluding the corresponding features from the attention calculation, which can help to regularize the attention mechanism and prevent overfitting.

3.6.1 On the Convergence of our Learning Framework

To ensure the convergence of our proposed approach, two conditions must be satisfied [1, 82]. First, it is important to verify that there is no data distribution mismatch between the original training set and the external data repository used for augmentation [82]. This means that the proposed approach should not select noisy or out-of-distribution samples. Otherwise, the model’s weights greatly shift during finetuning, and the initial convergence point is lost for a sub-optimal decision boundary. To address this point, ADA-ViT employs a utility function that takes into consideration the similarity of the new samples with the original training data. Additionally, we assign penalty scores to noisy samples that push the utility score to decrease and reduce the likelihood of selecting such samples.

The second condition assumes that the ViT model is capable of improving its performance as we feed it more data [1]. Unlike Convolution Neural Networks, whose performance tends to plateau at a certain point regardless of the amount of additional data they get, Transformer models, in general, are able to learn more features with more data, thanks to their lack of some inductive biases that constrain the learning of CNNs. Moreover, in the particular context of our work, we select relevant and informative samples to augment the training data, compared to traditional data augmentation techniques. Therefore, the ViT model is almost always guaranteed to be exposed to new data patterns and information, which encourages more learning and makes the performance less likely to quickly plateau.

To summarise, under the above-mentioned conditions, the iterative process of adding new data from external image repositories and retraining the model is expected to improve the performance, as the model is exposed to more examples and learns to generalize better. The convergence of the approach is expected when we retrieve the majority of relevant samples from the external data repositories. In this case, the performance improvement is no longer significant.

3.6.2 Justification of the choice of the Vision Transformer Model

Our work is specifically tailored to attention-based models. This is because ViTs have gained significant interest and adoption in the field of computer vision. They have been successful in competing with or even outperforming traditional convolutional neural networks (CNNs) in many image-related tasks. They have also shown potential in handling various image resolutions, offering

a more scalable approach compared to CNNs. Therefore, it is crucial to direct any current or future research efforts towards further improving the performance of ViTs.

With their ability to process images as sequences of patches, Vision Transformers have enabled researchers to explore novel ways of understanding and interpreting visual data. Extracting explanation-based concepts constitutes the core idea of the ADA-ViT algorithm. This is rendered possible and relatively simple thanks to the attention-mechanism in ViTs. Unlike CNN models that require post-hoc explanation methods, ViT models almost display a white-box behaviour, as simply projecting the attention weights to the input space provides an insightful look on the important image regions that led to the model decision. This explainability aspect of ViT models plays a major role in making our approach self-sufficient and easy to apply and adapt to any attention-based model.

CHAPTER 4

APPLICATION 1: OBJECT IDENTIFICATION IN RGB IMAGES

In this chapter, we present the experimental results of our proposed data augmentation method, ADA-ViT, to solve the task of object classification in RGB images. This task has always gained the majority of interest from the computer vision research community due to its significant impact on various fields. The State-Of-The-Art (SOTA) in image classification has been rapidly evolving and achieving more success everyday. With the recent emergence of attention-based models, Vision Transformers (ViT) surpassed the SOTA in terms of both efficiency and accuracy. While data augmentation techniques have undeniably contributed to these breakthroughs and proven to improve the performance of machine learning models, no efforts, to date, have been specifically dedicated to Transformer models. Additionally, the current data augmentation techniques focus on expanding the data in size or proposing advanced techniques to clean noisy external image repositories for augmentation purposes. However, very few works addressed the issue of the quality of training sets and aimed to improve its diversity with selective data augmentation. Specifically, the problem of identifying under-represented regions in the training feature space and solving it with guided data augmentation has not gained much interest, despite its importance, especially not for ViT models. Our work is relevant because it falls within the scope of addressing under-representative training sets with data augmentation for the particular case of Vision Transformers.

We evaluate our approach on three RGB benchmark datasets: CUB [83], CUB-Families [84], and Tiny-ImageNet [85]. We carefully select these datasets as they differ by size and class granularity. Our goal is to show that our proposed method works well on both small and large-scale datasets, with fine-grained classes where objects are hardly distinguished and the data is usually limited, or coarser classes where the issue of under-representative training sets is more obvious and the intra-class variance is high.

We conduct extensive analysis to demonstrate the effectiveness of our approach and its broad applicability. We demonstrate the issue of under-represented regions and showcase how ADA-ViT solves the problem by filling in the gaps in the feature space with the selected new samples. We also perform an ablation study to highlight the impact of each component in our proposed utility score function. Finally, we show that our method can significantly improve the performance of a ViT model, compared to other data augmentation techniques, while adding the least number of samples during the augmentation. Our primary objective is not to surpass the current state-of-the-

art models on the studied datasets, but instead to showcase the potential of our proposed approach by improving the performance over the baseline models trained without augmentation and the models trained with other data augmentation techniques that do not consider the under-represented regions of the training feature space.

We design our experiments to address the following research questions:

- **RQ1:** What is the impact of using ADA-ViT augmentation on the performance of a baseline ViT model?
- **RQ2:** How can we demonstrate the problem of under-represented regions in the training feature space and how can ADA-ViT address this issue?
- **RQ3:** What is the impact of using clustering for class representation?
- **RQ4:** How can we determine the optimal number of iterations for the ADA-ViT algorithm?
- **RQ5:** What is the impact of each component in the utility score function on the model performance?
- **RQ6:** What is the impact of using ADA-ViT utility score function on guiding sample selection?
- **RQ7:** What is the impact of the number of selected samples for augmentation on the classification performance?
- **RQ8:** How well does ADA-ViT perform compared to other state-of-the-art data augmentation techniques?

In the following sections, we start by stating our experimental setting, namely the evaluation datasets, the used baseline ViT , and training parameters. Finally, we present the results for the aforementioned research questions.

4.1 Experimental Setup

4.1.1 Datasets

To evaluate our proposed approach, we carry out experiments on three benchmark datasets:

- Caltech-UCSD Birds-200-2011 (CUB) [83]. This dataset is the most widely-used benchmark for fine-grained visual categorization tasks. It contains a total of 11,788 images of 200 sub-categories of birds. The images are of high resolution, with an average size of approximately 480x640 pixels. The CUB dataset is particularly challenging because the classes differ by only minor details, and the available data to learn from is limited. This is reflected in a low intra-class variance, high inter-class similarity (Table 4.1), and a low number of training samples per

class (Table 4.2). Therefore, we are expected to leverage the scarce training resources to, first, identify fine differences to distinguish between classes, and then identify the features that are absent from the training data and that are necessary for a better model generalization.

- CUB-Families [84]. This dataset groups the 200 species of birds in CUB into 37 families, according to the ornithological systematics [86]. It contains the same number of 11,788 total images as CUB. Although we are no longer dealing with fine-grained classes and the inter-class similarity is lower (Table 4.2), this dataset presents a different set of challenges. As the classes get coarser, the dataset becomes more imbalanced (Table 4.1) and the intra-class variance increases (Table 4.2), which enhances the likelihood of having under-represented regions in the training feature space. To motivate our approach and demonstrate its effectiveness, we further highlight the issue of under-representative training sets for this particular dataset by manually removing 47 bird species belonging to classes across 23 families from the training set [87]. No species were removed from the validation or the test sets. This procedure aims to intentionally create under-represented regions in the training feature space and serves to showcase the gravity of this issue and how ADA-ViT is able to address it.
- TinyImageNet [85]. This dataset is a subset of the ImageNet dataset [88]. It contains 110,000 images of 200 classes downsized to 64×64 images. Unlike the previous two datasets, TinyImageNet is relatively larger in scale and the classes cover a broader range of objects. Although, in such cases, there is usually enough data to train robust models (Table 4.2), the problem of under-represented regions is still present. This is because when datasets describe various independent objects, there are usually numerous different scenarios describing the objects, which can be difficult to capture them all in the initial training set, resulting in a high intra-class variance (Table 4.2).

Table 4.1 presents the data partition we used for each of the three datasets. We also compute the intra-class variance and inter-class similarity for the different datasets and show them in Table 4.2. We obtain the intra-class variance by computing the covariance matrix of the features extracted from a trained model and corresponding to each class of each dataset, summing the covariances over the first dimension, then averaging them over the second dimension, and, finally, we report the mean over the different classes. We obtain the inter-class similarity by representing each class with their cluster medoids, computing the cosine similarity between the medoids of different classes and taking the maximum similarity, and, finally, averaging the obtained similarities over the classes.

We build CUB-Families with under-represented classes using the following steps. The original CUB Families dataset has 5,994 training images. From this set, we removed images belonging to the bird species in [87], while excluding certain bird families (*Pacific Loon* and *Waxing* families)

TABLE 4.1

Data partition for CUB, CUB-Families, and TinyImageNet

Dataset	Classes	Total	Train	Train Samples/Class	Validation	Test
CUB	200	11,788	3,994	10-20	2,000	5,794
CUB-Families	37	11,788	4,585	30-539	2,275	4,928
TinyImageNet	200	110,000	80,000	400	20,000	10,000

TABLE 4.2

Intra-class variance and inter-class similarity measures for CUB, CUB-Families, and TinyImageNet.

Dataset	Intra-class variance	Inter-class similarity
CUB	66.61	162.05
CUB-Families	405.89	29.01
TinyImageNet	1332.25	16.25

which have a smaller number of images compared to other classes. With this step, 1,409 images were removed from the training set, resulting in 4,585 training images. We create the validation set with the images removed from the training set in the previous step, and a set of randomly selected test images. From each of the 37 classes in the test set, we randomly select 20% of the images and add them to the validation set. In this step, we did not select images from the bird species that we removed from the training set. This resulted in 2,275 validation images and 4,928 test images.

External Image Repositories

For both CUB and CUB-Families, we use the online image repository provided in [89], to select new images for augmentation. This web data covers the same classes of CUB dataset, and contains 18,388 total images. For TinyImageNet, we use the online image repository provided in [36], which has 38,618 total images and covers 181 of the 200 classes of TinyImageNet. Both image repositories are noisy and include out-of-distribution samples. We display sample images from both datasets in Figure 4.1 and Figure 4.2. For example, the *Black Footed Albatross* class contains noisy images of text documents about birds.

These collected web datasets sometimes include duplicates or samples that are very similar to the ones in the original training dataset. Adding such examples does not contribute to improving the classification accuracy. Hence, before selecting the new data for augmentation, we remove the duplicates and the images that are similar to existing training data. We consider two images to be similar if the cosine similarity between their feature vectors is higher than 0.99.

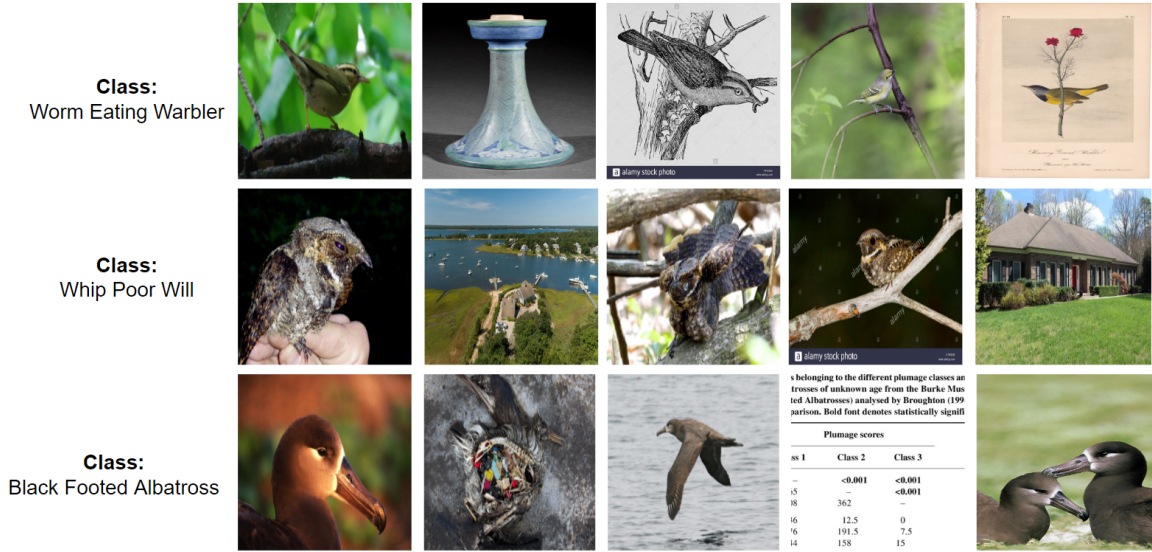


Figure 4.1: Sample images from the web dataset used for CUB and CUB-Families.

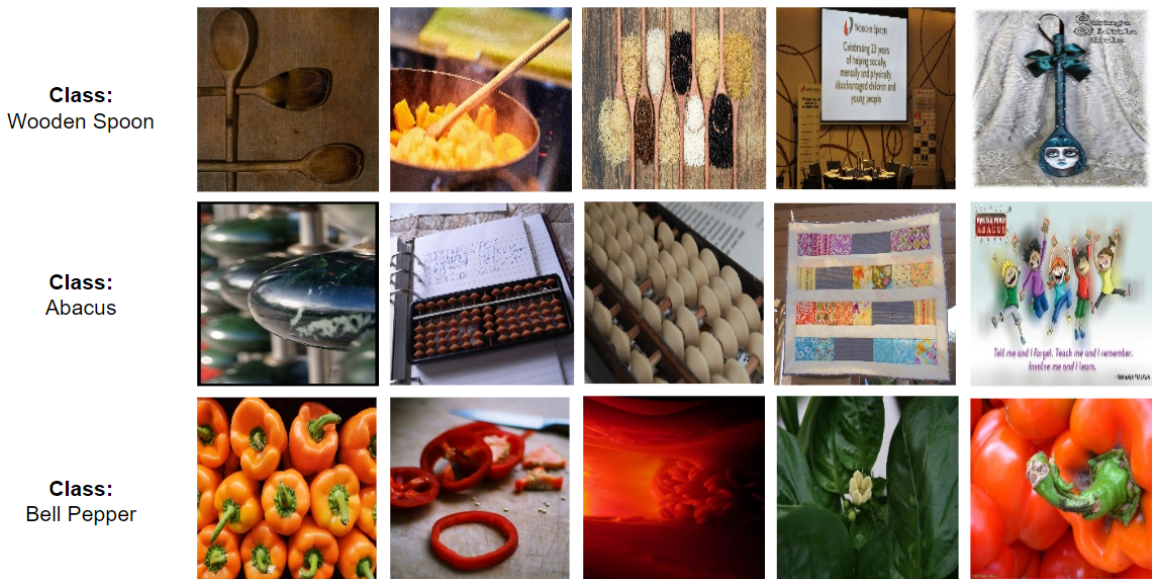


Figure 4.2: Sample images from the web dataset used for TinyImageNet.

4.1.2 Baseline Model

We conduct our experiments on the basic Vision Transformer model that has been initially introduced in [1]. This is the first ViT architecture that has been proposed as an adaptation of the Transformer model to the computer vision field. Even though there are several efforts [15–17, 90] that built on the original ViT and proposed improved versions of this model outperforming it, we still chose to experiment with the vanilla ViT for several reasons. First, we would like to faithfully evaluate the effectiveness of our data augmentation method, independently of any other outside model improvements that were designed to boost the performance. Also, we believe that our

proposed method is applicable to any attention-based transformer architecture, as long as the model computes attention scores to divided image patches. Therefore, it is not necessary to experiment with all ViT variants. Instead, it is sufficient to prove that the method works on the basic model version.

4.1.3 Training Parameters

In our experiments, we utilize two variants of vision transformers that differ in their architectural complexity: base (ViT-B) and large (ViT-L), with embedding dimensions of 768 and 1024, respectively. We set the image size to 224×224 for all datasets. We divide the input images into overlapping patches with a patch size of 16 and a sliding window of 12, as proposed in [90]. We noticed that increasing the number of input patches using a sliding window helps to improve the performance of the baseline ViT. This is especially true for the case of fine-grained datasets, such as CUB, where we found that the original split method, which cuts the images into non-overlapping patches, harms the local neighboring structures especially when discriminative regions are split.

All models are initialized with pretrained weights from ImageNet21K [88]. We tune the quantile Q_p and set it to 0.98 for validation samples (refer to Equation (3.6)), while we relax it for the new web samples (refer to Equation (3.8)) by setting it to 0.95. The baseline transformers, as well as the models trained with other data augmentation techniques, are trained with SGD optimizer and a cosine-annealing scheduler. We keep the same optimizer and scheduler to fine-tune the ViT model on the augmented data. During finetuning, we employ training strategies to prevent overfitting, such as applying attention dropout [81] and pretrained weight decay [80]. We replicate each experiment 5 times and report the mean and standard deviation of the accuracy.

Finally, our code is implemented using PyTorch, and all experiments are run on NVIDIA Tesla V100 GPUs.

4.2 Performance Analysis

RQ1: What is the impact of using ADA-ViT augmentation on the performance of a baseline ViT model?

In this experiment, We compare the performance of a baseline ViT (Section 4.2.1), trained using the existing training data only, against the performance of our approach (Section 4.2.2), where the model is trained with augmentations generated by ADA-ViT. We provide a detailed analysis of the experimental results (Section 4.2.3), by analysing the misclassifications and highlighting the performance gains achieved by our proposed data augmentation.

4.2.1 Performance of a Baseline ViT without ADA-ViT Augmentation

First, we present the accuracy results of our baseline ViT model trained on the original training set without augmentation. Then, we analyze the performance by investigating the correct and incorrect classifications.

We train the baseline ViT using the available training sets of CUB, CUB-Families and TinyImageNet. The model is validated using the respective validation sets of the studied datasets. After obtaining the optimal performance on the validation set, we evaluate the model on the test set of the considered datasets. The average testing accuracies across five runs for all datasets are presented in Table 4.3.

TABLE 4.3

Accuracy performance of baseline models on the test set.

Dataset	ViT-B	ViT-L
CUB	89%	90.25%
CUB-Families	93.7%	95.2%
Tiny ImageNet	89.9%	93.5%

As aforementioned, beside selecting the best performing model checkpoint, our approach uses a hold-out validation set to guide the data augmentation process. We leverage the validation set to identify under-represented regions in the feature space and select new samples that can cover these gaps. We assume that the validation set is drawn from the same distribution as the testing set. Thus, by analyzing the model’s errors on the validation set and understanding their causes, we can better design our data augmentation and training strategies to address these challenges and improve the overall performance of our classifier.

For a more in-depth analysis, we focus on CUB dataset, but we note that similar trends were observed for CUB-Families and TinyImageNet. In Figure 4.3, we display four sample images that were correctly classified by the baseline. In Figure 4.4 we display four sample images, that were misclassified by the baseline. We focus on four classes from the most confused ones. For each figure, we show, in the first column, a sample image from the validation set that has been either correctly classified (Figure 4.3) or misclassified (Figure 4.4). In the remaining 5 columns, we display the 5 Nearest Neighbors (NN) from the training set to the sample images of the first column, along with their distances.

For the first example, the baseline model correctly classifies an image of the bird pictured flying over the sea (Figure 4.3, first row), but it misclassifies the image of the same bird pictured sitting on the ground (Figure 4.4, first row). It is also worth noting that the misclassified image shows a partial view of the bird with a focus on its head. We inspect the training set and find that this bird is usually presented flying or swimming in the sea with a complete view on its entire body.



Figure 4.3: Samples of correctly classified validation images and their 5 NN images from the training set using the baseline model for CUB. We indicate the true class label and the distance of the kNN above each image.

Therefore, the baseline model was confused about this atypical image of a bird from Figure 4.4, since it is different from the samples of the correct class and it is more similar to samples from the class of the closest neighbor. In the second example, the baseline model succeeds to predict the correct class of an image showing a close view of the bird and displaying clear features (Figure 4.3, second row). However, the same bird is pictured in a different image (Figure 4.4, second row) from a more distant angle and under darker lighting conditions that effaced the distinguishing features of this bird, leading to misclassification. Similarly, the bird in the third row of Figure 4.4 was misclassified because the corresponding image only shows the back of the bird, which prevents the appearance of any distinguishing features on the face. The correctly classified sample in the third row of Figure 4.3 shows that this bird has distinctive facial features, without which it is difficult to identify its correct specie. Finally, the last example (Figure 4.4, fourth row) was misclassified by the baseline mainly

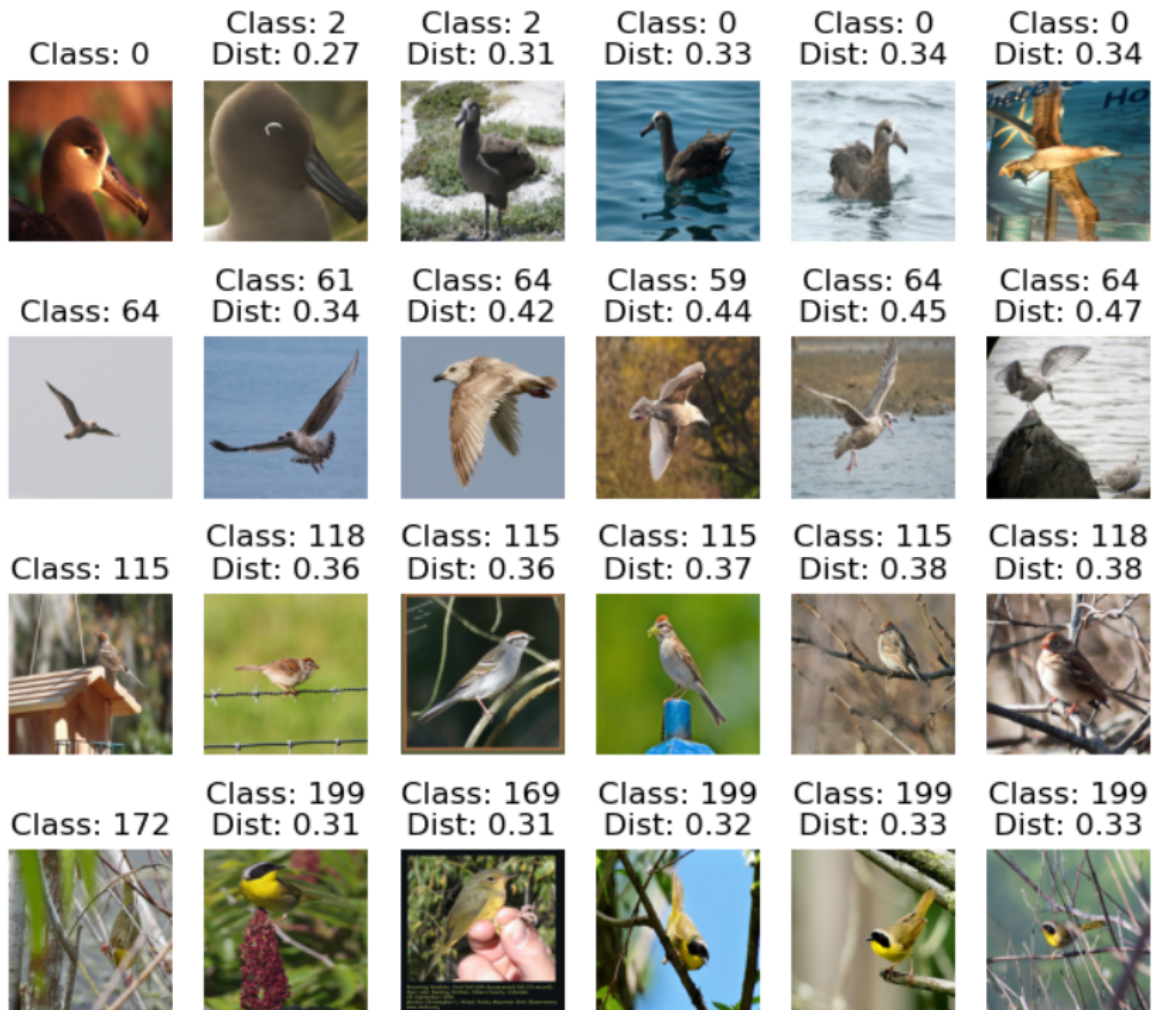


Figure 4.4: Samples of misclassified validation images and their 5 NN images from the training set using the baseline model for CUB. We indicate the true class label and the distance of the kNN above each image.

because of occlusion from the tree branches, and the particular pose of the bird that blocked the view on important features, like the color of the belly and face.

Generally, we notice that the baseline model struggles with images containing significant occlusion, atypical bird poses, different backgrounds, or when there are few similar images from the training data. It tends to perform better with more typical representations of the bird classes. Another interesting observation is that the 5 nearest neighbors of all the images that were classified correctly have distances less than 0.3. On the other hand, even the closest images to the misclassified samples have distances larger than 0.3. This may indicate that the main reason for misclassifying these samples is because they are under-represented in the training set.

4.2.2 Performance of a Baseline ViT with ADA-ViT Augmentation

In this section, we present the results of the previous baseline ViT model after finetuning it using a combination of the new selected samples by ADA-ViT and the original training set. ADA-ViT selects new samples from external web datasets that can enhance the under-represented classes in the original training set and cover the sparse gaps in the training feature space. We run ADA-ViT for three iterations and, each time, we finetune the model on the augmented set and the original data. We present, in Table 4.4, the average testing accuracies across five runs for CUB, CUB-Families and TinyImageNet. We also show the number of added samples by ADA-ViT for each dataset, in Table 4.5

Dataset	CUB		CUB-Families		TinyImageNet	
	ViT-B	ViT-L	ViT-B	ViT-L	ViT-B	ViT-L
Original dataset	89%	90.25%	93.7%	95.2%	89.9%	93.5%
ADA-ViT	91.05%	91.8%	97.45%	97.8%	91%	93.9%

TABLE 4.4

Accuracy performance of baseline models with and without ADA-ViT augmentation.

Dataset	ViT-B	ViT-L
CUB	2,293	494
CUB-Families	1,030	299
TinyImageNet	9,567	1920

TABLE 4.5

Number of images added by ADA-ViT for the different datasets.

We can clearly see that using ADA-ViT augmentation yields significant performance gains over the baseline model trained without augmentation. We also note that the number of added samples is correlated with the performance of the model on the original dataset. For example, ViT-L usually requires less augmentation than ViT-B since the baseline ViT-L outperforms the baseline ViT-B. That is, the lower the performance of the baseline model is on the original training set, the more samples ADA-ViT selects for augmentation. This shows that ADA-ViT addresses the specific limitations of the model and aims to correct its misclassifications with carefully selected samples that can bridge the performance gap.

The results indicate that the generated augmentations improve the model’s ability to generalize better to unseen data. ADA-ViT sample selection strategy proves to be advantageous because it generates augmentations that enables the model to learn better representations of the data. By enhancing the under-represented classes and including challenging samples, the model becomes more

robust and improves its class representation and generalization capabilities to achieve better accuracy.

4.2.3 Analysis of Misclassified Samples

In this section, we analyze and compare the misclassified samples by the baseline model trained without augmentation and/or the baseline model trained with ADA-ViT augmentation. We focus our study on CUB dataset since it is a small dataset and the classes are fine-grained, which allow for a more detailed analysis. We also focus on the first iteration of ADA-ViT since that is where we see the most considerable improvement in the performance

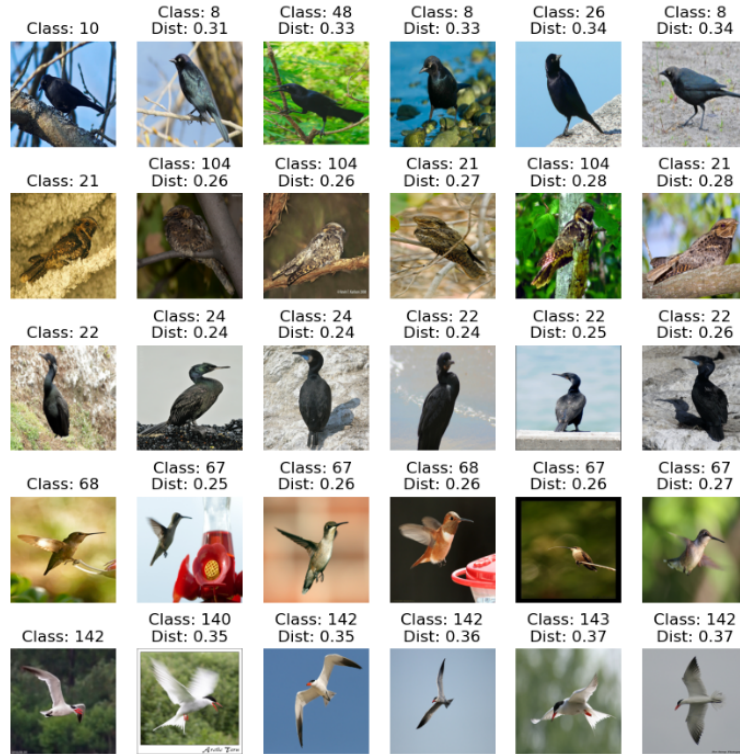
We consider three categories of misclassifications:

- **Category 1:** samples misclassified by the baseline, and corrected after the first iteration of ADA-ViT.
- **Category 2:** Newly introduced misclassified samples after the first iteration of ADA-ViT.
- **Category 3:** samples misclassified by the baseline, and remained misclassified after the first iteration of ADA-ViT.

For each of the three aforementioned categories, we display test samples from five different classes of CUB. For each test sample, we display the five nearest neighbors from the training set, along with their respective distances, using embeddings generated by the baseline model trained before and after ADA-ViT augmentation. By inspecting the neighborhood of each prediction, we can deduce the impact of the generated augmentations on the classifier’s decisions. We note that the training set for the baseline model initially includes the original training set only, while the finetuned model uses both the augmented samples and the original data.

In each of the three figures (Figure 4.5, Figure 4.6 and Figure 4.7), we display the results of the KNN analysis for samples from each of the three categories of misclassifications (Category 1, Category 2 and Category 3), respectively. In all figures, the first column shows test examples from a given category, while the subsequent five columns show their 5 nearest neighbors from the training set using the embeddings generated by (a) the baseline model, and (b) the first iteration of ADA-ViT augmentation. We also display the classes and the distances of the 5 nearest neighbors above each neighbor image.

Figure 4.5 shows examples of misclassified samples by the baseline model that were corrected after one iteration of ADA-ViT (Category 1). In Figure 4.5a, we observe that the baseline model matches the selected samples with relatively distanced images from the wrong classes. As shown in Figure 4.5b, these five misclassified test samples are corrected after one iteration of ADA-ViT and are mapped to training images from the same class with smaller distances. This indicates that ADA-ViT has helped to learn a better model that provides more meaningful and informative



(a) Baseline Embedding.



(b) ADA-ViT (iter. 1) Embedding.

Figure 4.5: Category 1 misclassifications and their 5 nearest neighbors from the training set. The first column shows test examples from Category 1. The remaining five columns show the 5 NNs from the training set using embeddings generated by (a) the baseline model, and (b) the first iteration of ADA-ViT augmentation.



(a) Baseline Embedding.



(b) ADA-ViT (iter. 1) Embedding.

Figure 4.6: Category 2 misclassifications and their 5 nearest neighbors from the training set. The first column shows test examples from Category 2. The remaining five columns show the 5 NNs from the training set using the embeddings generated by (a) the baseline model, and (b) the first iteration of ADA-ViT augmentation.



(a) Baseline Embedding.



(b) ADA-ViT (iter. 1) Embedding.

Figure 4.7: Category 3 misclassifications and their 5 nearest neighbors from the training set. The first column shows test examples from Category 3. The remaining five columns show the 5 NNs from the training set using the embeddings generated by (a) the baseline model, and (b) the first iteration of ADA-ViT augmentation.

representations of the samples. Figure 4.5b also reveals that several new images selected by ADA-ViT for augmentation, marked using green boxes, are located in the direct neighborhood of the now correctly classified samples. This indicates that ADA-ViT improved the model learning by adding similar images to the hard samples that were previously misclassified. This allows the model to better capture the underlying data distribution and refine the mapping of the classes by matching them to images that display more similar features.

Figure 4.6 shows examples of correctly classified samples by the baseline that became misclassified after one iteration of ADA-ViT (Category 2). In Figure 4.6a, we notice that the first nearest neighbor to the correctly classified samples is not always from the same class as the test sample. Additionally, few nearest neighbors are actually from the correct class. This indicates that, even though these test samples are correctly classified by the baseline model, the model is probably not strongly confident about the prediction and its decision is susceptible to change. Figure 4.6b shows that the new selected samples by ADA-ViT are usually located in the direct neighborhood of the now misclassified samples. These new samples describe classes that are highly similar to the class of the test sample, and they display images of birds in similar poses and backgrounds as the test samples. This suggests that the augmentations may have introduced additional confusion to the model that led to the misclassification.

Finally, Figure 4.7 shows examples of misclassified samples by the baseline that remained misclassified after one iteration of ADA-ViT (Category 3). We notice that the test samples present challenging features, such as uncommon or distant views of the birds, or severe occlusion that hide distinctive features. In Figure 4.7b, we notice that fewer added samples by ADA-ViT are in the direct vicinity of the test samples. Meanwhile, samples from the original dataset that caused the baseline mistakes, appear more as nearest neighbors. Interestingly, we can see improvements in the mapping of few classes. For instance, in the first test example (class 136), there is a new added image from the correct class that appears among the nearest neighbors. In the second example (class 81), the first four nearest neighbors are from the same correct class and the first three nearest neighbors are new images. Similarly, in the fourth test sample (class 22), two neighbors are newly added images from the correct class. We predict that the model will be able to improve its learning of these test samples and it will correctly classify them in the next few iterations of ADA-ViT.

4.3 Illustration of ADA-ViT

RQ2: How can we demonstrate the problem of under-represented regions in the training feature space and how can ADA-ViT address this issue?

In this experiment, our goal is to, first, demonstrate the problem of under-represented regions in the feature space of the training data. Then, we show how ADA-ViT is able to address this issue

by filling in these gaps with new samples. We focus our analysis on CUB-Families, since this dataset has among the highest intra-class variances and a severely imbalanced training set, compared to CUB and TinyImageNet. Therefore, it suffers the most from the under-represented regions in its training feature space. Additionally, as explained in Section 4.1.1, we further aggravate the problem of under-representative training data by manually removing some species from specific bird families. This results in the creation of gaps in the feature space of the training data.

First, we train a baseline vision transformer on CUB-Families without any augmentation. We select five bird family classes that were affected the most by the class removal and scored the lowest accuracies to conduct our analysis on them. The selected classes are *Cuculidae*, *Alcidae*, *Mimidae*, *Fringilidae*, and *Podicipedidae*. We use the trained model to extract the learnt features corresponding to the images of the studied classes. Specifically, we extract features from the last linear layer that precedes the classification layer. The retrieved features have a dimension of 768 using the ViT Base model. We use TSNE [91] as a tool to reduce the high dimensionality of the obtained features and be able to visualize them by mapping each image to a point on a two-dimensional space.

In Figure 5.5a, we display a region of the TSNE projection of the features corresponding to the training and validation samples generated by the trained ViT for the studied bird families. Note that most misclassifications are occurring in the regions of the feature space where there is little to no training examples. This observation confirms that the model does not generalize well when there are gaps in the feature space of the training data. Therefore, we hypothesize that if we target these sparse regions and fill them with new training samples, the model’s performance improves due to better class coverage.

Next, we run ADA-ViT for one iteration to select new images from the web data to augment the studied classes. In Figure 5.5b, we add the TSNE projection of the new selected images to the feature space of Figure 5.5a. As it can be seen, for each class, the added images overlap with the misclassified validation samples, and occupy the regions where the previous model errors occurred. This demonstrates that ADA-ViT aims to fill in the gaps in the feature space and augments the training data with new samples that can cover under-represented regions responsible for the misclassifications.

To quantify the results, we scan the set of new added images after one iteration. ADA-ViT selected a total of 362 new samples. 60% of the added samples come from the 47 species that were initially removed from the training data (out of 200 original species). This means that an image of class $c \in C^*$, where C^* is the set of classes describing the 47 removed species, is three times more likely to be selected by ADA-ViT than any other sample from the remaining species (153 species). Moreover, in only the first iteration, ADA-ViT added images that covered 64% of the removed species. After running ADA-ViT for three iterations, we verify that 73% of the removed species have been covered by new images. We check the remaining 27% and confirm that these classes were previously

confused with other species that have either not been removed, or have been already covered by new images. Therefore, the model was able to learn their corresponding bird families without seeing actual images of these uncovered species.

4.4 Class Representation using Multiple Prototypes

RQ3: What is the impact of using clustering for class representation?

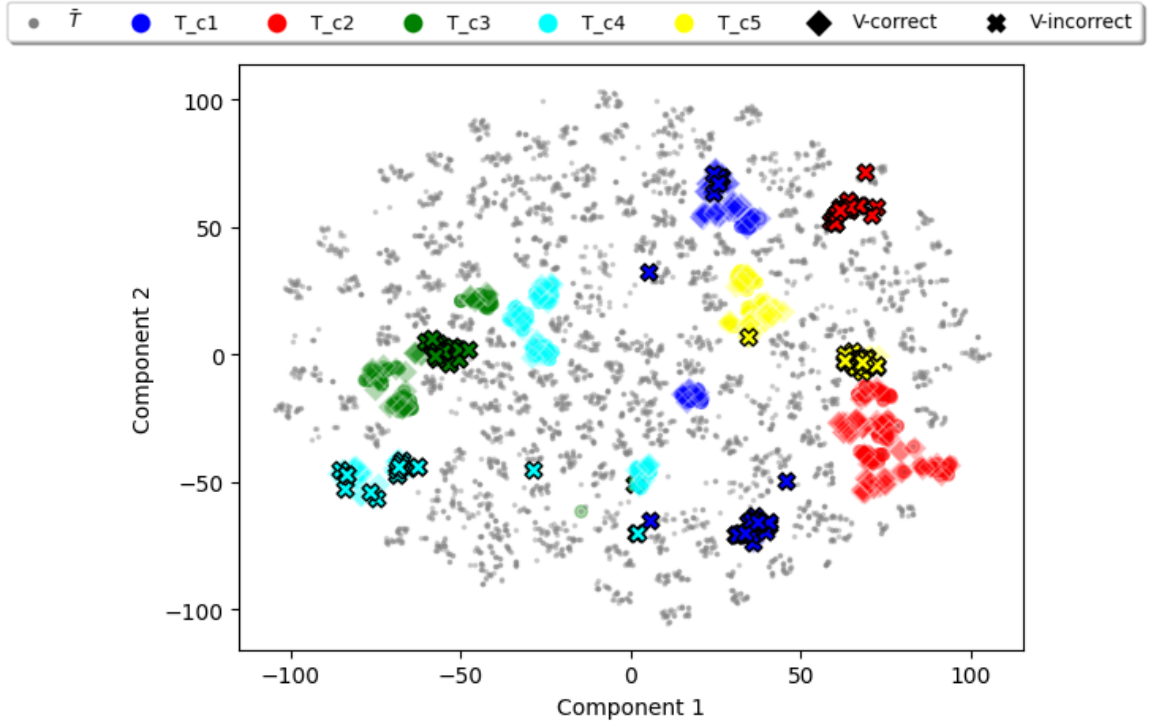
The mathematic formulation of the under-representation score β (Section 3.4) and the label regularization score α (Section 3.3) calls for a method to quantify the degree to which a new sample displays common features with the training data of a given class. To this end, we need to generate prototypes to represent the training data of each class.

One way to go about this problem is to represent the data with a single prototype that corresponds to the mean feature vector of all training images from the same class. However, we argue that this method is not effective as it assumes that the training data is unimodal, and thus can be represented by a single data point. There are several cases where this assumption does not hold true. For example, when classes are not compact and there are gaps in the feature space of the training data, the average of features may fall in the gap area and, consequently, would not reliably represent the class. This is why we believe that using multiple prototypes should yield better and more accurate class representation.

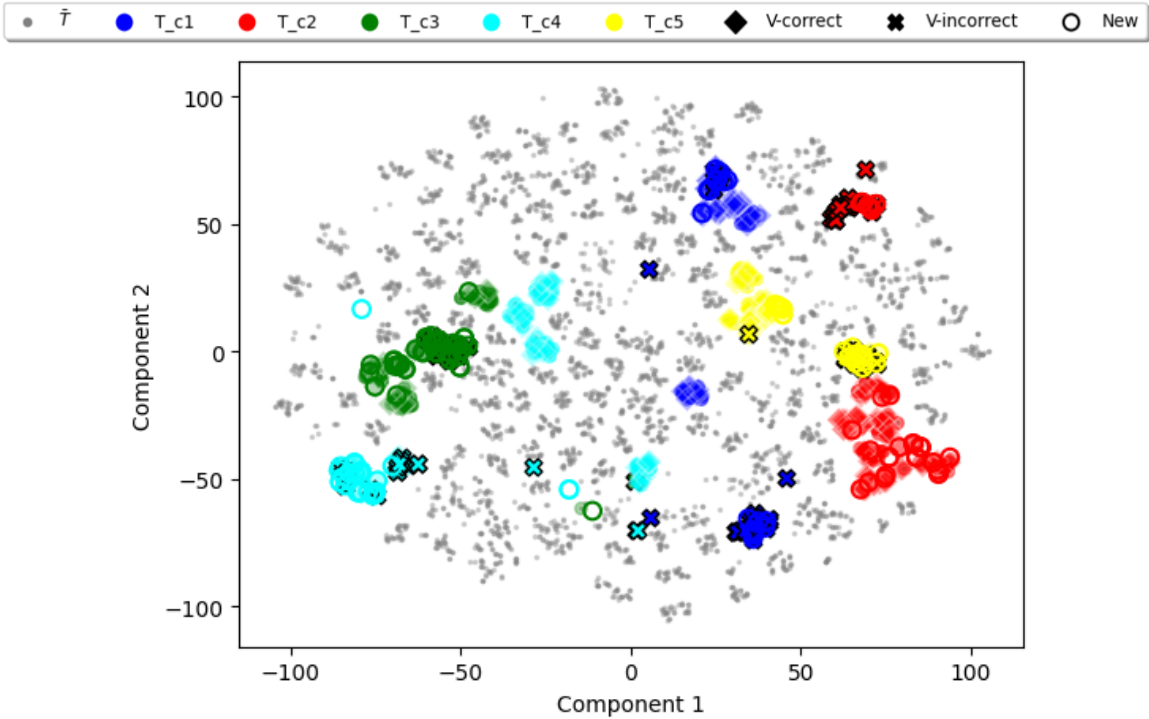
We proposed an alternative solution that represents classes using multiple prototypes captured by clustering the data. Each cluster is represented by its medoid. Then, we designed a similarity function φ (Section 3.2) that computes the cosine similarity between the feature of a new sample and the feature of each cluster medoid.

In this experiment, we highlight the advantage of using multiple prototypes to represent the data of a class. We compare three different methods for class representation:

1. Compute the mean of the feature vectors of all training data of each class, as was adopted by [36]. We refer to this approach as *All-Mean*.
2. Use K-medoids algorithm to group the training samples of each class into K clusters and represent each class with its corresponding cluster medoid. We set $K = 5$ for CUB and CUB-Families, and $K = 10$ for Tiny ImageNet. We refer to this approach as *K-Med*.
3. Use hierarchical agglomerative clustering with a minimum distance threshold, to favor more compact clusters, and represent each class with its corresponding cluster medoid. We set the minimum distance threshold to 0.25 for CUB and CUB-Families, and 0.55 for Tiny ImageNet. We refer to this approach as *AGG*.



(a) TSNE projection of the training and validation samples.



(b) TSNE projection of the training, validation, and new samples.

Figure 4.8: TSNE analysis of the four selected classes (C) from CUB-Families: *Cuculidae* (c1), *Alcidae* (c2), *Mimidae* (c3), *Fringilidae* (c4), and *Podicipedidae* (c5). In the legend, T represents the training data of C only, \bar{T} is the training data for all classes except C , V -correct is the correctly classified validation data of C , V -incorrect is the incorrectly classified validation data of C , and New indicates the added samples for C by ADA-ViT. This figure is best viewed in color.

TABLE 4.6

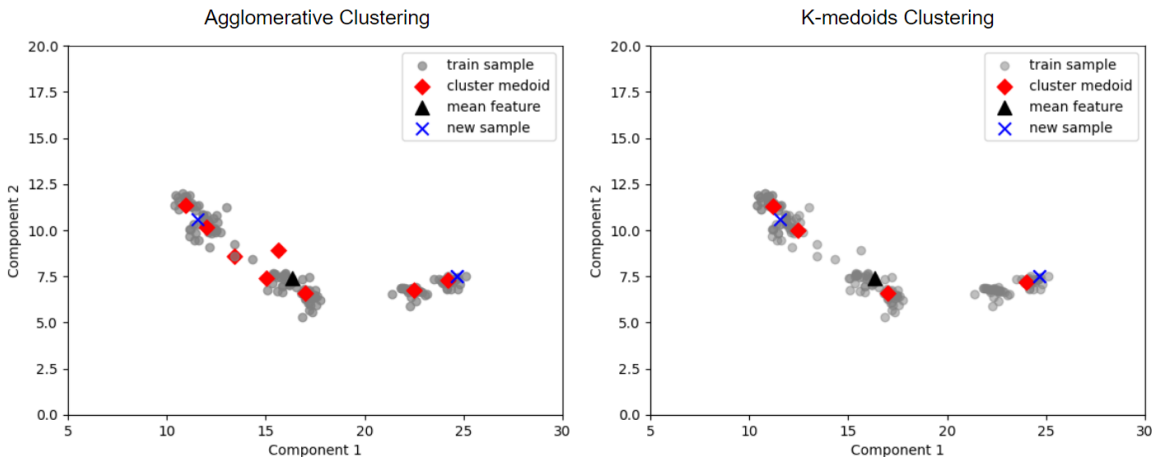
Classification accuracy when different methods are used to represent the training data of each class.

Dataset	Method	All-Mean	K-Med	AGG
CUB	Original	89%	89%	89%
	Iteration#3	90%	90.95%	91.05%
CUB-Families	Original	93.7%	93.7%	93.7%
	Iteration#3	97%	97.3%	97.45%
Tiny ImageNet	Original	89.9%	89.9%	89.9%
	Iteration#3	90.45%	90.6%	91%

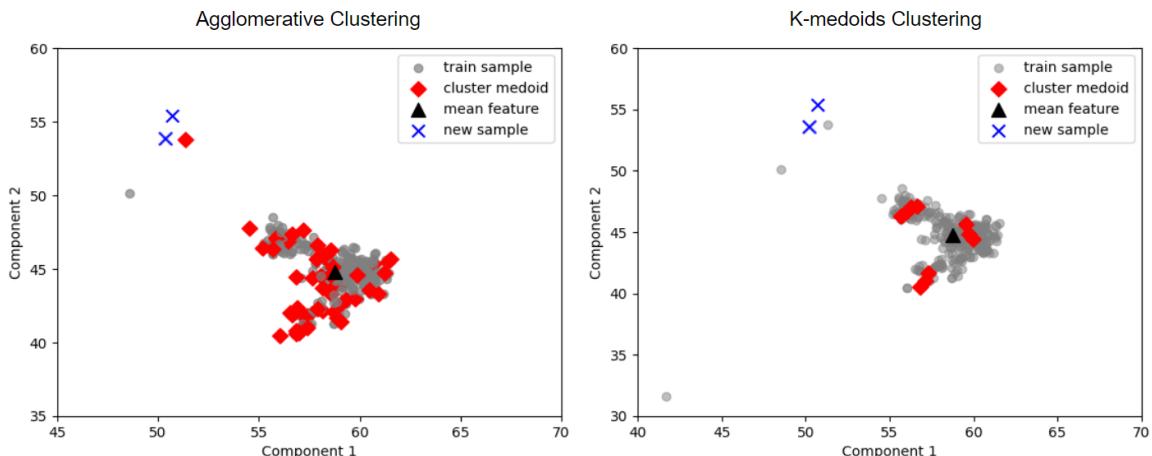
We train a baseline ViT model. Then, we run the ADAViT algorithm for three iterations using each of the three methods, and finetune the baseline model on the new augmented set. In Table 4.6, we display the obtained accuracy results for the different trained models. We note that, in all datasets, the clustering-based methods consistently reach higher accuracy than taking the mean of all features. This gap in performance is even more obvious for CUB-Families, where there are significant under-represented regions in the training feature space. This observation is further supported in Figure 4.9a which shows the TSNE projection of the training data of the class *Tyrannidae* from CUB-Families. As it can be seen, there are gaps in the training feature space due to the high intra-class variance. The mean of the feature vectors of all training samples may not represent all the variations of the target class reliably. Instead, it was biased toward the most dense area of data points. Consequently, the new web sample, which was projected in the less dense region, fell far away from the mean feature vector, resulting in a lower similarity, and thus a lower value of the under-representation score β . Meanwhile, both K-medoids and the agglomerative clustering succeeded to generate medoids that cover the majority of the space occupied by the training samples. Therefore, there is a medoid close enough to the new sample to obtain a high β score.

Finally, we notice from Table 4.6 that the agglomerative clustering outperforms K-medoids. This is because the agglomerative clustering is a bottom-up approach that starts with fine-grained clusters and iteratively merges them based on similarity. By specifying a minimum cluster distance threshold instead of fixing the number of clusters, we allow the formation of compact clusters with more representative medoids. On the other hand, the K-medoids tends to create balanced groups of samples. This means that the few distant samples that occupy the under-represented regions of the feature space will not be able to have their independent clusters, and will be assigned to nearby groups instead. This leads to the formation of sparse clusters with higher intra-cluster distance. This observation can be confirmed in Figure 4.9b which displays the TSNE projection of the class *Wooden Spoon* from Tiny ImageNet. The new samples are projected close to a single training data point that indicates the presence of an under-represented region in the feature space. Unlike the K-medoids, the agglomerative clustering assigned a cluster to this training data point which allowed

the new samples to have a higher similarity with the corresponding cluster medoid.



(a) TSNE projection of the class *Tyrannidae* from CUB-Families.



(b) TSNE projection of the class *Wooden Spoon* from Tiny ImageNet.

Figure 4.9: Comparison of different methods to represent the class *Tyrannidae* from CUB-Families and the class *Wooden Spoon* from TinyImageNet. This figure shows the TSNE projection of the training samples of the studied classes, as well as the vector corresponding to the mean of feature vectors of all training samples, the medoids of the clusters obtained by each clustering algorithm, and a subset of the added new samples for these classes.

4.5 Stopping Criteria for ADA-ViT Iterations

RQ4: How can we determine the optimal number of iterations for the ADA-ViT algorithm?

The previous experiments have shown that selecting new data for augmentation using ADA-ViT approach can improve the results. Thus, in theory, this process can be repeated to boost the performance even further. To this end, we proposed a learning pipeline where ADA-ViT can run

iteratively. In each iteration, we obtain a new model as a result of finetuning the model from the previous iteration on new selected samples that aim to fix the previous errors.

This iterative approach has several advantages. First, it enables the ViT model to utilize a larger and more diverse augmented set, which can improve the model’s performance. This is because we accumulate the augmented data throughout the iterations. Moreover, by inspecting the model performance after each iteration, we can keep on identifying more under-represented regions, that may have been missed in previous iterations or may have appeared in newer iterations.

In this experiment, our goal is to determine the optimal number of iterations of ADA-ViT that allows us to obtain the best model performance while limiting the overall training time. To do so, we need to analyze the model performance after each iteration, by keeping track of its errors and observing the evolution of the model’s correct and incorrect classifications throughout the different iterations. As a rule of thumb, it is best to end the iterations of ADA-ViT when we start introducing more errors than we are correcting the previous ones. Alternatively, a stopping criteria can be designed to accommodate specific requirements imposed by certain applications. For instance, it may be optimal to stop the iterations when we reach a satisfactory performance on certain predefined important targets, even if we confuse more the predefined less important targets.

Iteration	CUB	CUB-Families	TinyImageNet
Iter. 1	550	394	3,087
Iter. 2	492	219	2,183
Iter. 3	435	158	1,694
Iter. 4	420	134	1,601
Iter. 5	418	125	1,002

TABLE 4.7

Size of selected samples by ADA-ViT for augmentation for each iteration. We report the results for the ViT-B model.

Table 4.7 summarizes the size of the selected samples by ADA-ViT used to augment the training data and finetune the ViT model in each iteration. We notice that the size of the selected samples decreases as the number of ADA-ViT iterations increases. This is because ADA-ViT is designed to gradually identify and cover the under-represented regions of the training feature space and, thus, progressively improve the performance of the model. In the first iterations, under-represented regions are more obvious and ADA-ViT is able to identify a relatively large set of errors that need to be corrected in the next iterations. As the model is refined in the subsequent iterations, there are less errors to investigate. Consequently, ADA-ViT selects fewer samples for augmentation, since the model has already learnt most of the difficult patterns in the data.

Tables 4.8, 4.9, and 4.10 show the evolution of different categories of misclassifications for CUB, CUB-Families and TinyImageNet across five iterations of the proposed ADA-ViT algorithm.

Iteration	T_{misc}	$T_{misc \rightarrow corr}$	$T_{misc \rightarrow misc}$	$T_{corr \rightarrow misc}$
Baseline	652	-	-	-
ADA-ViT (Iter. 1)	607	112	540	67
ADA-ViT (Iter. 2)	591	54	553	38
*ADA-ViT (Iter. 3)	583	21	570	13
ADA-ViT (Iter. 4)	585	3	580	5
ADA-ViT (Iter. 5)	584	4	581	3

TABLE 4.8

Evolution of the misclassified samples in the test set of CUB dataset across five iterations. We mark by * the optimal number of iterations.

Iteration	T_{misc}	$T_{misc \rightarrow corr}$	$T_{misc \rightarrow misc}$	$T_{corr \rightarrow misc}$
Baseline	4,598	-	-	-
ADA-ViT (Iter. 1)	169	4,434	164	5
ADA-ViT (Iter. 2)	148	30	139	9
ADA-ViT (Iter. 3)	136	22	126	10
*ADA-ViT (Iter. 4)	129	17	119	10
ADA-ViT (Iter. 5)	128	12	117	11

TABLE 4.9

Evolution of the misclassified samples in the test set of CUB-Families dataset across five iterations. We mark by * the optimal number of iterations.

In these tables, we quantify the following categories of misclassifications:

- T_{misc} : the total number of misclassifications in the test set.
- $T_{misc \rightarrow corr}$: The number of misclassified samples in the previous iteration that were corrected in the current iteration.
- $T_{misc \rightarrow misc}$: the number of misclassified samples in the previous iteration that remained misclassified in the current iteration.
- $T_{corr \rightarrow misc}$: the number of correctly classified samples in the previous iteration that became misclassified in the current iteration.

The first row of each table shows the total number of misclassifications using the baseline model trained without augmentation. The subsequent rows show the number of misclassifications in the different categories we previously defined, across five iterations of ADA-ViT, using the models finetuned on the augmented data that we generate after each iteration.

We observe that the total number of misclassifications, T_{misc} , of the baseline is higher for CUB-Families, followed by TinyImageNet, and finally CUB dataset. This can be justified by the fact that CUB-Families suffers the most from under-represented regions in its training feature space, mainly because the intra-class variance is highest for this dataset, and we further highlighted the

Iteration	T_{misc}	$T_{misc \rightarrow corr}$	$T_{misc \rightarrow misc}$	$T_{corr \rightarrow misc}$
Baseline	1008	-	-	-
ADA-ViT (Iter. 1)	956	119	889	67
ADA-ViT (Iter. 2)	929	54	902	27
*ADA-ViT (Iter. 3)	914	34	895	19
ADA-ViT (Iter. 4)	915	19	895	20
ADA-ViT (Iter. 5)	915	35	880	35

TABLE 4.10

Evolution of the misclassified samples in the test set of TinyImageNet dataset across five iterations. We mark by * the optimal number of iterations.

gaps in the feature space by manually removing some species from specific classes in the training set. As for TinyImageNet, this dataset is challenging and it is larger in size than CUB and CUB-Families. Nevertheless, ADA-ViT still managed to improve the performance of the model for the three datasets, which demonstrates the advantage of the iterative process of ADA-ViT in refining the model’s learning and improving its generalization capabilities.

Furthermore, we notice that the total number of misclassifications, T_{misc} , decreases after each ADA-ViT iteration. This suggests that ADA-ViT is effectively correcting previous errors and improving the overall model’s performance. This observation can be further confirmed by looking at the number of corrected misclassifications, $T_{misc \rightarrow corr}$. During the first three iterations, the model is correcting way more errors than it is introducing, and the total number of corrected misclassifications is consistently increasing across the different iterations. This further proves that the selected new samples for augmentation by ADA-ViT is effectively addressing the model’s weaknesses. We also note that $T_{misc \rightarrow corr}$ is highest during the first few iterations, where there are more obvious errors to correct. This is when ADA-ViT is able to best identify under-represented regions in the feature space. This observation is consistent with the findings from Table 4.7, where we saw the size of the selected samples decreasing as the iterations progress.

The size of the newly introduced misclassifications, quantified by $T_{corr \rightarrow misc}$, is significantly smaller than the size of corrected samples, $T_{misc \rightarrow corr}$. We also observe that $T_{corr \rightarrow misc}$ generally decreases as the iterations progress, indicating that the new selected samples are relevant and do not introduce significant noise to the training set. The introduced misclassifications are likely due to augmentations that may have introduced additional challenges to the model and increased its confusion about specific classes.

We notice that the number of misclassified samples that have not been corrected in the next iteration, quantified by $T_{misc \rightarrow misc}$, is almost consistent across different iterations for all datasets, except for CUB-Families where we see $T_{misc \rightarrow misc}$ decreasing especially during the first three iterations. For CUB and TinyImageNet, these uncorrected misclassifications probably characterize challenging samples displaying atypical features that are not covered by the web datasets used for

augmentation. Further boosting the diversity of the web datasets to include similar challenging samples may help to correct this category of errors. As for CUB-Families, $T_{misc \rightarrow misc}$ is seen to decrease after each iteration, indicating that the model is progressively addressing errors of the previous iterations. Since the under-represented regions are more numerous in CUB-Families than the other datasets, ADA-ViT required several iterations to fully address all the weaknesses of the model and efficiently cover most gaps in the feature space.

Our findings from this experiment have revealed that the rate of improvement starts to decrease after few iterations. The improvement rate can be characterized by two main factors. First, we look at T_{misc} and ensure that it is significantly decreasing from the previous iteration. Second, we inspect $T_{misc \rightarrow corr}$ and $T_{corr \rightarrow misc}$ and verify that we are not introducing more errors than we are correcting. Based, on these two criteria, we decide to stop the iterations of ADA-ViT. For CUB and TinyImageNet, we find it is best to stop at the third iteration since T_{misc} is not improving after the third iteration and $T_{misc \rightarrow corr}$ is almost equal to $T_{corr \rightarrow misc}$. As for CUB-Families, this dataset requires more iterations. Therefore, it is best to stop at the fourth iteration.

4.6 Ablation study

4.6.1 Justification of the ADA-ViT Scoring Function Design

RQ5: What is the impact of each component in the utility score function on the model performance?

In this experiment, we examine the effect of each component of the utility score function (defined in Equation (3.1)) independently, by implementing three variants of ADA-ViT:

- $ADAViT^{-\alpha}$: we remove the label regularization term α and keep the under-representation score β and the degree of match Δ .
- $ADAViT^{-\beta}$: we remove the under-representation score β and keep the label regularization term α and the degree of match Δ .
- $ADAViT^{-\Delta}$: we remove the degree of match Δ and keep the label regularization term α and the under-representation score β .

Table 4.11 shows the accuracy results of the models trained with the three implemented ADA-ViT variants, in addition to the original ADA-ViT with all components. We observe that the highest accuracy is achieved with α , β and Δ all present in the utility function. In particular, using both β and Δ to compute the utility score proves to be most beneficial, compared to using one without the other. Moreover, we observe that β has the highest impact on the utility score as its absence degrades the performance of the model the most, followed by Δ , and finally α .

We further confirm these findings by calculating the overlap between the images selected by the four different ADAViT variants. We measure the overlap by computing the Intersection Over Union (IOU) between the selected samples, and we display the results in Table 4.12. We notice that removing β from the utility score decreases the overlap with the original algorithm the most with $IOU(ADAViT, ADAViT^{-\beta}) = 19\%$, followed by Δ with $IOU(ADAViT, ADAViT^{-\Delta}) = 73\%$ and α with $IOU(ADAViT, ADAViT^{-\alpha}) = 87\%$. We also find that the overlap between $ADAViT^{-\beta}$ and $ADAViT^{-\Delta}$ is only 17%, which indicates that β and Δ tend to select different samples.

We interpret these results as follows. The term β represents the under-representation score that is responsible for selecting hard samples that are similar to what the model previously misclassified. Therefore, samples with higher β scores hold information that the initial model hasn't learned yet due to the lack of such examples in the training set, which explains why β alone has the largest impact on the utility score. On the other hand, Δ serves to further finetune the samples selected by β in order to favor misclassified examples with specific concepts that led to the misclassification. However, if we use Δ alone, these misclassification concepts can be identified in various samples, that are not necessarily under-represented. For instance, Figure 4.10 shows examples of selected samples by $ADAViT^{-\beta}$ and $ADAViT^{-\Delta}$ for some classes of CUB dataset. In this figure, we see that $ADAViT^{-\Delta}$ tends to select samples that are not necessarily under-represented (samples correctly classified by the initial model) simply because they displayed similar features to the identified misclassification concepts. On the other hand, β selects under-represented samples that do not necessarily resemble the misclassified validation samples (do not display the misclassification concepts). Therefore, it is important to use both β and Δ jointly to select both hard samples that display specific features that led to model confusion in the previous iteration. These are the most relevant samples capable of filling in the gaps in the initial training set. Finally, we see that removing α impacts the model performance the least. This is mainly because α is used to address the occasional issue of label noise in the secondary image dataset and does not intervene in the selection of relevant samples that populate the under-represented regions of the feature space.

TABLE 4.11

Ablation study: Comparison of classification accuracies of different ADAViT variants on the three selection datasets.

Dataset	Method	ADAViT	$ADAViT^{-\alpha}$	$ADAViT^{-\beta}$	$ADAViT^{-\Delta}$
CUB	Original	89%	89%	89%	89%
	Iteration#3	91.05%	90.3%	90.05%	90.1%
CUB-Families	Original	93.7%	93.7%	93.7%	93.7%
	Iteration#3	97.45%	97.05%	96.5%	96.7%
Tiny ImageNet	Original	89.9%	89.9%	89.9%	89.9%
	Iteration#3	91%	90.5%	90%	90%

TABLE 4.12

IOU between the selected images by the four different ADAViT variants for CUB dataset.

Method	$ADAViT^{-\alpha}$	$ADAViT^{-\beta}$	$ADAViT^{-\Delta}$
ADAViT	0.87	0.19	0.73
$ADAViT^{-\alpha}$	1.0	0.18	0.69
$ADAViT^{-\beta}$	0.18	1.0	0.17

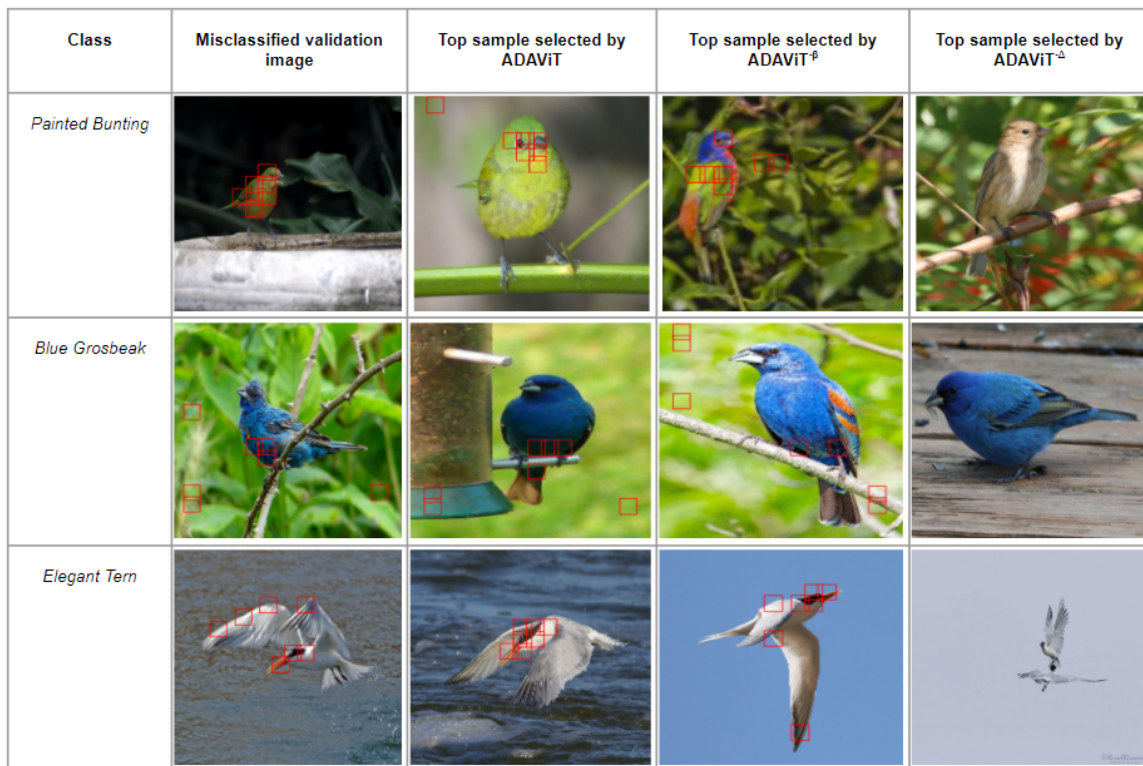


Figure 4.10: Examples of selected images for CUB dataset by $ADAViT$, $ADAViT^{-\beta}$, and $ADAViT^{-\Delta}$. We display the misclassification concepts on the misclassified validation sample, as well as the patches on the new selected image that match these concepts, as identified by the Δ term in $ADAViT$ and $ADAViT^{-\beta}$

4.6.2 Importance of Guiding Data Augmentation by ADA-ViT scoring

RQ6: What is the impact of using ADA-ViT utility score function on guiding sample selection?

We investigate the importance of guiding the selected augmentations from the external web datasets using ADA-ViT utility score function. We compare three scenarios of selecting samples from web datasets to use as additional augmentation for the training data. In each scenario, we augment the training set using the same number of samples. We only vary the selection criteria:

Random sampling vs. Confidence-based sampling vs. ADA-ViT-guided sampling. We describe the experiments as follows:

- **Experiment 1:** ViT baseline trained with the original training data without augmentation.
- **Experiment 2:** Trained ViT baseline finetuned on the original training data, in addition to a subset from the web data of size N (Equation (3.2), Section 3.1) selected by applying ADA-ViT scoring strategy.
- **Experiment 3:** Trained ViT baseline finetuned on the original training data, in addition to a random subset from the web data with the same size N as in Experiment#2.
- **Experiment 4:** Trained ViT baseline finetuned on the original training data, in addition to a subset from the web data corresponding to under-performing samples, with the same size N as in Experiment#2. An under-performing sample can be either a misclassified sample or a correctly classified sample with low confidence. For the latter case, we set a threshold on the predicted confidences of correctly classified samples. This threshold correspond to the lower outlier boundary, calculated using: $Q_1 - 1.5 \times IQR$, with Q_1 being the lower quartile and IQR the interquartile range. The outlier boundary sets a statistical fence for a data distribution, beyond which a data point is considered an outlier. In this context, an under-represented sample can be viewed as an outlier, since there are not enough data points from the training set that share similar features with it. In this experiment, we test all web samples using the trained model from the previous iteration. Then, we randomly select N images from the identified under-performing samples.

Table 4.13 shows the classification accuracies on CUB, CUB-Families and TinyImageNet for the four different settings, described above. We see that all augmentations succeeded to improve the classification accuracy. However, using the ADA-ViT scoring function to rank and select samples gives the best improvement, followed by the random selection, and, finally, the confidence-based selection.

We display in Figure 4.11 examples of the selected samples using random, confidence-based, and ADA-ViT-guided augmentations. We notice that the images selected by the random strategy do not necessarily relate to the misclassified validation samples. They can be easy to classify, which will not introduce new and relevant information to the model. They can also be noisy samples that may introduce further confusion to the model. The confidence-based strategy yields the worst classification results for CUB and CUB-Families, as shown in Table 4.13. While this strategy encourages the model to learn from challenging regions of the feature space by augmenting the data with misclassified samples, this method fails when the external image repository is noisy. This is the case for CUB and CUB-Families, where most of the added images are out-of-distribution samples,

as shown in Figure 4.11. The external image repository used for Tiny ImageNet is less noisy, and thus, the confidence-based strategy is seen to outperform random selection for this dataset. On the other hand, we see that ADA-ViT selects relevant samples that resemble the misclassified validation image and does not pick up any noise.

Dataset	CUB		CUB-Families		Tiny ImageNet	
Baseline model	ViT-B	ViT-L	ViT-B	ViT-L	ViT-B	ViT-L
Original dataset	89%	90.25%	93.7%	95.2%	89.9%	93.5%
Random	89.8%	90.55%	96.3%	96.1%	90.1%	93.6%
Confidence	89.2% %	90.5 %	95.3 %	96.6 %	90.35 %	93.85%
ADA-ViT	91.05%	91.8%	97.45%	97.8%	91%	93.9%

TABLE 4.13

Random vs. Confidence-based vs. ADA-ViT-guided Augmentation.







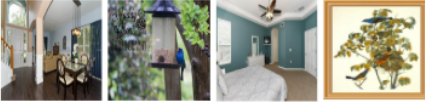
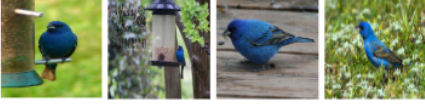
Class	Misclassified validation image	Sample selection method	Examples of selected images
Wooden Spoon (Tiny ImageNet)		Random	
		Confidence	
		ADA-ViT	
Blue Grosbeak (CUB)		Random	
		Confidence	
		ADA-ViT	

Figure 4.11: Examples of selected images by the Random Augmentation and Guided Augmentation by ADA-ViT scoring.

4.6.3 Impact of the Size of Selected New Samples

RQ7: What is the impact of the number of selected samples for augmentation on the classification performance?

As explained in Section 3.1, we rank the new samples based on the computed utility scores and select the top $N(c)$ to augment each class c using Equation (3.2). This number is computed as a proportional quantity to the ratio of misclassification in class c so that the augmentation is larger for the classes with higher misclassification rates.

In this experiment, we study the impact of the size of the augmentation on the classification performance. In other words, we vary the number of selected samples by taking fractions and multiples of the number N , and we analyze the performance of the final model each time. In Figure 4.12, we report the testing accuracies of the classifier when we add $N/2$, N , $2 \times N$, $4 \times N$, $8 \times N$, $16 \times N$, or $32 \times N$ samples for augmentation. We report the results for the three datasets CUB, CUB-Families and Tiny ImageNet.

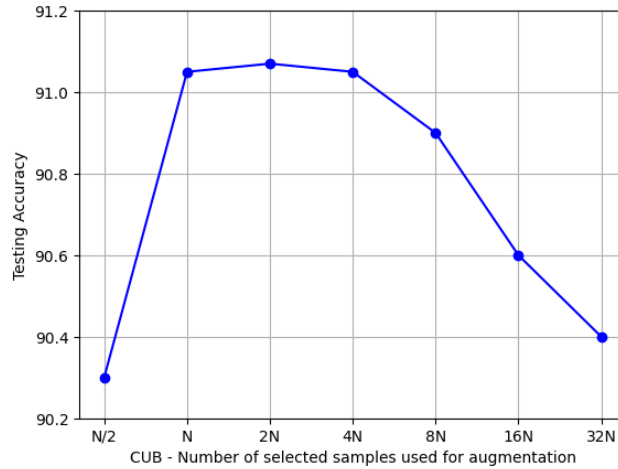
We notice that, for all datasets, the optimal number of images to select for augmentation is N , as this number yields the best performance while limiting the complexity of the training by adding fewer training samples. As we add more images, we see that the classification performance drops, mainly for CUB and CUB-Families whose external image repositories used for augmentation are noisy. This is because we start adding out-of-distribution samples that have low rankings, which can add further confusion and leads the model to learn sub-optimal decision boundaries. If we select fewer samples (less than N), the under-represented regions of the feature space are not fully covered by new samples, and, consequently, the model does not reach the maximum performance.

4.7 Comparison with other State-Of-The-Art Data Augmentation Techniques

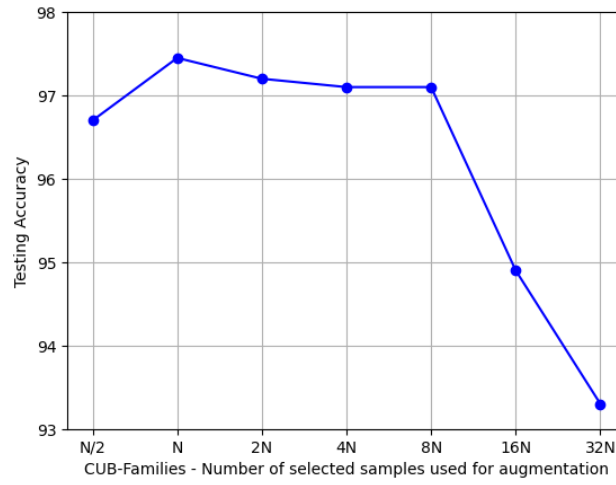
RQ8: How well does ADA-ViT perform compared to a baseline ViT and other state-of-the-art data augmentation techniques?

To illustrate the advantage of ADA-ViT, we evaluate its performance on CUB, CUB-Families and TinyImageNet against three state-of-the-art data augmentation methods: a combination of techniques that generate augmentations from the current training set [24, 25, 28], a Meta-Set based method [19], and BRACE [36]. For this comparison, the four models are provided with the same input data which is split into training, validation and test subsets. We also feed them the same web datasets, in case the method performs augmentation from external image repositories.

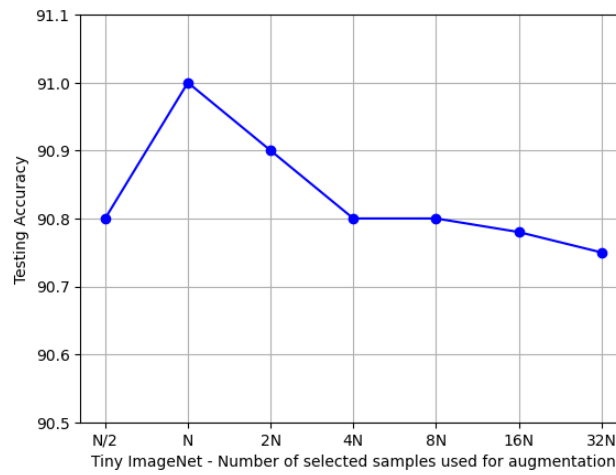
- **Local** [24, 25, 28]: We use a combination of data augmentation techniques that generate new samples by applying transformations to the original training set, thus the *Local* notation. These



(a)



(b)



(c)

Figure 4.12: Evolution of the performance of the model trained on different sizes of ADA-ViT augmentations on (a) CUB dataset, (b) CUB-Families and (c) Tiny ImageNet.

augmentations are typically done in-place during each epoch or mini-batch of training. We use CutMix [24], Mixup [25], and AutoAugment [28] augmentations. CutMix randomly removes parts from an image and replaces them with patches from another image. Mixup creates new samples by generating a weighted combination of random image pairs from the training data. Finally, AutoAugment is an automated approach that searches for the best transformation policy among several augmentation operations, such as translation, rotation, or shearing, and the probabilities and magnitudes with which the functions are applied. These augmentation methods achieved great success on several RGB benchmark datasets, such as CIFAR-100 [92], SVHN [93], and ImageNet [88]. We use the validation set to tune the parameters of these techniques on the studied datasets.

- **Meta-Set** [19]: This method requires the access to external image repositories for augmentation. Specifically, this data augmentation technique proposes a framework that can learn directly from web datasets. To address the noise in the web training sets, it learns two networks to distinguish in- and out-of distribution samples, and to correct the labels of in-distribution noisy data, guided by a small amount of clean meta-set. The goal is to alleviate the harmful effects caused by out-of-distribution noise and properly exploit all of the in-distribution samples for training. This method is relevant in our comparison because it operates in an opposite way as ADA-ViT. While our method aims to select only the relevant samples, the Meta-Set approach focuses on filtering out noisy samples and adding all the in-distribution images to the training set, without further selection.
- **BRACE** [36]: This method also uses external image repositories. It is the most relevant work to ADA-ViT since it addresses the issue of under-represented regions in the training feature space. Similar to our work, BRACE uses a utility function to rank the new samples based on their relevance and their potential contribution to improving the model performance. This method is only applicable to CNN-based models, because it leverages concept-based model explanations extracted from post-hoc explanation methods, such as GradCam [37], or extracted from special CNN architectures that are interpretable, such as Comprehensible CNNs [38]. In our comparison, we implement BRACE with GradCam, as this option yielded the best results according to the paper [36]. We use a CNN backbone of ResNet-200 [3] as baseline, which has almost the same number of parameters as ViT-Base.

Table 4.14 shows the classification accuracies of the baseline models trained with ADA-ViT, as well as the performance of the models trained with the other compared data augmentation techniques. We report our results using a ViT base (ViT-B) and ViT large (ViT-L) transformer baselines, on the three selected datasets. We observe that ADA-ViT augmentation consistently outperforms other methods for all datasets. In particular, ADAViT improves the baseline performance by approx-

Dataset	CUB			CUB-Families			Tiny ImageNet		
Baseline model	ViT-B	ViT-L	CNN	ViT-B	ViT-L	CNN	ViT-B	ViT-L	CNN
Original dataset	89%	90.25%	83.4%	93.7%	95.2%	88%	89.9%	93.5%	87.6%
Local [24, 25, 28]	88.75%	90.1%	N/A	93.5%	95.9%	N/A	90%	93.2%	N/A
Meta-Set [19]	89.9%	90.8%	N/A	97.3%*	97.6%*	N/A	89.7%	93.2%	N/A
BRACE [36]	N/A	N/A	84.75%	N/A	N/A	89.1%	N/A	N/A	87.9%
ADA-ViT	91.05%	91.8%	N/A	97.45%	97.8%	N/A	91%	93.9%	N/A

* ADA-ViT is not significantly better than a given baseline method with 95% confidence.

TABLE 4.14

Comparison of the classification accuracies on the three selected datasets. We run ADAViT for 3 iterations. For the CNN baseline, we only report the accuracy of BRACE and we do not run other augmentation methods with the CNN baseline as this is outside the scope of our research.

imately 2% on CUB dataset. Our method shows to be most advantageous on CUB-Families, where the model performance improved, on average, by more than 3%. This is because CUB-Families suffers from significant gaps in the feature space of its training data and ADAViT adds samples that aim to cover these under-represented regions. However, for TinyImageNet, ADAViT improved the performance by around 1%, on average. We interpret this result by considering the fact that the performance on the original dataset is already high enough, and TinyImageNet is a large and comprehensive data where the issue of class under-representation may not be as obvious as in the other datasets. Finally, we note that ViT-B tends to achieve better performance improvement than the larger model ViT-L, which is mainly due to differences in the initial model performance (before augmentation). This finding is expected, since smaller models are less able to learn from the data than their larger counterparts with their complex architectures. However, we want to highlight the fact that, even though ViT-L is a complex model, it still benefited from our augmentation.

The *Local* augmentation failed to improve the performance of the baseline model, compared to the other approaches, as this method is constrained to the local neighborhood of the existing training set. Therefore, the created samples can not recover the missing features from the training set and complete the under-represented regions in the training feature space. In particular, in the CUB-Families dataset, where there are explicit gaps in the training set, the *Local* augmentation was unable to recover the removed species, resulting in no significant improvement in the accuracy of the model.

As shown in Table 4.15, the Meta-Set approach adds the highest number of images, compared to the other methods. For CUB dataset, even though it adds 5 times more images than ADA-ViT, the improvements of Meta-Set are still falling short. For CUB-Families, this method has a significantly similar performance to ADA-ViT. However, it adds almost 10 times more images than our approach. This indicates that selecting fewer samples that target under-represented regions only is sufficient to improve the performance of the classifier, without unnecessarily increasing the task complexity. Finally, Meta-Set displays scalability limitations as its performance decreased significantly when

trained on Tiny ImageNet, which is a larger-scale data compared to the other datasets.

Dataset	CUB	CUB-Families	Tiny ImageNet
Meta	11,492	11,492	24,136
BRACE	3,991	3,298	13,124
ADA-ViT	2,293	1,030	9,567

TABLE 4.15

Number of images added by each data augmentation method that requires external datasets for augmentation.

The BRACE method, which is a CNN-based approach, is behind the other data augmentation techniques, mainly because the CNN baseline scored lower than the transformer baseline. This observation serves to confirm the findings of recent studies [1, 16, 41, 47] that highlighted the advanced learning capabilities of attention-based models and showed that transformers are inherently more robust than CNNs. Nevertheless, BRACE still managed to increase the accuracy of the baseline CNN for all datasets, which highlights the importance of selective data augmentation. In particular, it achieved 1.4% improvement for CUB dataset and 0.3% for Tiny ImageNet. For CUB-Families, the dataset with the least representative training set, BRACE increased the accuracy by only 1.1%, compared to 3.5% for ADA-ViT (ViT-B). This gap in performance can be explained by two main factors. First, as discussed in Section 4.4, this can be an indicator of a flaw in the BRACE utility score, which represents classes by a single data point that corresponds to the mean of features of all training samples, as opposed to ADA-ViT which employs clustering to generate more robust class representations. This issue becomes more highlighted for the case of datasets with high intra-class variation, such as CUB-Families. Second, the under-performance of BRACE, compared to ADA-ViT, can reflect the advantage of using the attention weights learnt inside the transformer itself to identify concepts that led to the misclassification, over post-hoc explanation methods, such as GradCam, or pretrained object detectors, such as RCNN, which are agnostic to the task and dataset in hand.

4.8 Chapter Summary

In this chapter, we discussed the experimental results of our proposed data augmentation framework evaluated on three RGB benchmark datasets. These datasets vary in both size and class granularity. First, we conducted an in-depth analysis of the issue of under-represented regions in the training feature space and showcased its harmful impact on the model performance. Then, we illustrated how ADA-ViT operates to address this issue by adding new samples that can cover these sparse regions of the feature space. We carried out extensive experiments to justify the design of our data augmentation framework. Our ablation study showed that the three scores α , β and δ contribute differently to the sample selection process and we concluded that it is best to include them

all in the utility function to obtain the maximum model performance. The optimal number of ADA-ViT iterations is found to be 3 for CUB and Tiny ImageNet, and 4 iterations for datasets with more severe under-represented regions, like CUB-Families. We also varied the size of the selected samples and found that the computed number N yields the highest performance while limiting the overall training complexity. We conducted an experiment to highlight the benefits of using clustering to represent the data of a class with multiple prototypes. Another experiment showed the advantage of using ADA-ViT to guide the sample selection from secondary web datasets over other data selection methods, such as random sampling or selecting under-performing samples based on the prediction confidence. Finally, we compared the performance of our proposed approach with other state-of-the-art data augmentation techniques. The purpose of this experiment is to highlight the importance of considering under-represented regions in the training data when applying data augmentation. Additionally, we showed that our method achieves the highest performance improvements while adding the least number of samples. This proves that expanding training sets in size only, without considering the diversity of samples, does not necessarily lead to more robust models that can generalize better. This is why ADA-ViT is specifically designed to enhance the quality and the representativeness of the training data.

CHAPTER 5

APPLICATION 2: AUTOMATIC TARGET RECOGNITION FROM INFRARED IMAGES

In this chapter, we propose a new strategy to improve Automatic Target Recognition (ATR) from infrared (IR) images by leveraging our proposed data augmentation technique. We show that ADA-ViT can be used to identify and incorporate few additional relevant samples that bridge the performance gap and lead to a more accurate and robust ATR system.

Automatic Target Recognition (ATR) from infrared images is an important task in computer vision with many practical applications in security, emergency services, automotive, environment, and other fields [94]. Infrared images offer fundamental advantages over regular imaging solutions, such as their ability to perform well in low-light and low-visibility situations. This is critical for outdoor applications where light and visibility can vary significantly. However, infrared sensors can be sensitive to meteorological conditions and sensor calibration. This results in having the same target appearing differently in various instances, leading to high intra-class variability. Therefore, it is important to collect large and diverse IR datasets that cover the broad variance of the underlying data distribution to build robust and high performing ATR systems. This requirement usually limits the accuracy of existing methods for ATR applications, since it is challenging to acquire large IR datasets due to the high cost of collecting and labeling the data. Consequently, IR datasets used to train ATR systems frequently suffer from severe under-represented regions in their feature spaces, as a result from the inability to acquire sufficient diverse samples that can cover the class variance.

Data Augmentation has been used as a solution [21] to circumvent the issue of limited IR data. Some proposed methods [95] create new samples from the existing training data by applying different kinds of geometric and intensity transformations. Other generative methods [66] explored the direction of expanding the data with synthetic samples using features from the existing training set. While both approaches succeeded to increase the size of the IR training set, they are constrained to exploring local neighborhoods of the current data samples and cannot significantly expand the diversity and coverage of the training dataset.

In this chapter, we adapt our proposed data augmentation approach to overcome the issue of limited IR datasets available for training and augmentation. We incorporate ADA-ViT along with Vision Transformer models to improve the robustness of ATR models in challenging environments. Our approach offers several advantages over existing methods. First, it can effectively leverage the

limited IR labeled data available for augmentation by selecting only the relevant samples that can improve the model performance, thus reducing the need for expensive and time-consuming data collection and labeling. Second, it can improve the model’s robustness to different environmental conditions, by pushing the model to learn from the most challenging regions of the feature space and selecting samples that are capable of covering the under-represented regions.

We evaluate our approach on an annotated infrared benchmark dataset. We also leverage non-annotated IR datasets for augmentation, and make use of automatic detectors to generate weak annotations. We conduct extensive analysis to demonstrate the effectiveness of our approach and its particular applicability for an infrared scenario. We demonstrate the issue of under-represented regions in IR data and showcase how ADA-ViT solves the problem by filling in the gaps in the feature space with the selected new samples. Finally, we show that our method can significantly improve the performance of an ATR system, compared to other data augmentation techniques, as well as other data selection strategies.

We design our experiments to investigate the following research questions:

- **RQ1:** What is the impact of using ADA-ViT augmentation on the performance of an ATR system?
- **RQ2:** What is the impact of using ADA-ViT utility score function on guiding sample selection?
- **RQ3:** How well does ADA-ViT perform compared to state-of-the-art data augmentation techniques?

5.1 Data Preparation

5.1.1 YOLO Algorithm Overview

The YOLO object detector is a popular deep learning-based approach for object detection in images [5]. Unlike traditional object detection approaches that use region proposals and post-processing steps, YOLO uses a single neural network to predict both the object locations and class probabilities directly from the image pixels. YOLOv5x [96] is an improved version of the traditional YOLO model with a simpler design but better performances.

The YOLO network divides the input image into a grid of cells and predicts the bounding boxes for each object based on the cell locations [5]. Each cell predicts a fixed number of bounding boxes, and each bounding box is defined by five values: the (x, y) coordinates of the box center, the box width w , the box height h , and the confidence score for the box. The confidence score measures how confident the network is that the bounding box contains an object. To predict the class probabilities for each bounding box, the YOLO network uses a softmax activation function applied to the output of a convolutional layer [5].

The class probabilities are conditioned on the presence of an object in the bounding box and the class label of the object. To assign confidences to the outputs, YOLO uses a combination of the box confidence and the class probabilities [5]. The box confidence is defined as the product of the objectness score and the Intersection over Union (IoU) between the predicted box and the ground-truth box. The objectness score measures the likelihood that an object is present in the bounding box, while the IoU measures the overlap between the predicted box and the ground-truth box [5]. The class confidence is the product of the box confidence and the class probability for the predicted class. In this research, we leverage the YOLO confidence score and the IoU threshold for each detected bounding box in order to generate weakly-annotated augmentations.

5.1.2 Original Training Set

To evaluate our proposed approach, we carry out our experiments on an annotated infrared benchmark dataset: FLIR ADAS v2 [2]. The dataset was acquired via a thermal and visible camera pair mounted on a vehicle. It captures traffic footage from various locations in the world, mainly England, France and the US. The dataset provides fully annotated thermal image frames, describing different targets, captured at different times of the day and different weather conditions. The annotations include the bounding box coordinates of the detected targets, the time of the day the video was captured, and the degree of target occlusion.

There is a total of 15 different categories of objects included in this dataset. In our experiments, we focus on 8 classes that have sufficient training data samples: *person*, *bike*, *car*, *motor*, *bus*, *truck*, *light*, and *sign*. Figure 5.1 shows example images from each category. Certain targets appear far away from the camera, and thus are only captured by few pixels, which is not sufficient for robust training and reliable evaluation. Therefore, we filter the dataset to keep only targets with a bounding box area larger than 500.

We use three data partitions to train and evaluate our models: a training set, a validation set to select the best model checkpoint and perform the misclassification analysis for ADA-ViT, and a held-out test set for final model evaluation. We summarize the number of samples in each class and each data partition in Table 5.1. The FLIR ADAS dataset is highly imbalanced, with *car* and *person* being the majority classes, and *truck*, *light*, and *motor* the minority classes. We address this aspect of the dataset in our training settings by using a weighted cross-entropy loss. We also note that the class *bus* is absent from the test set.

5.1.3 External Image Repository

We use an external IR dataset for augmentation, called Brno Urban dataset [97]. This dataset also captures traffic footage from various road sceneries and under different weather conditions. However, it is not annotated and the ground-truth bounding boxes are not provided for our targets

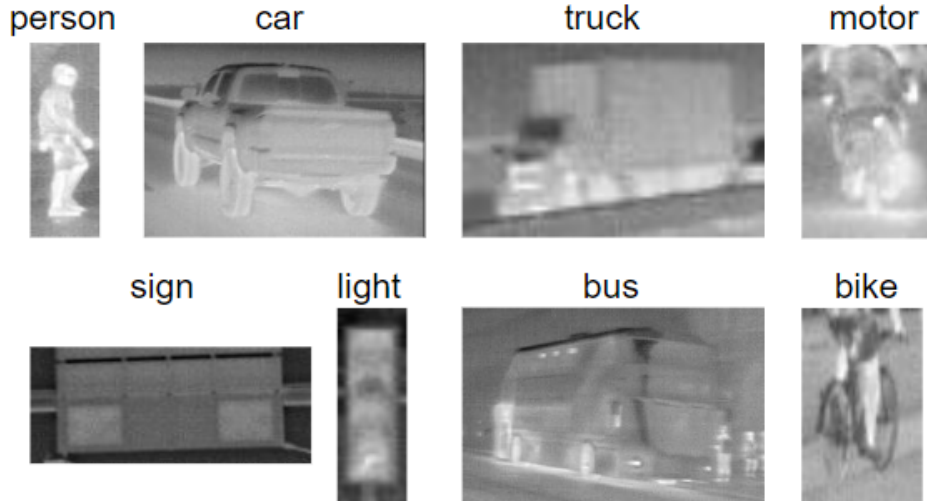


Figure 5.1: Example images from the FLIR ADAS dataset [2].

Classes	Training	Validation	Test
Total	52,930	4,927	11,647
car	31,513	3,254	7,535
person	12,432	1,059	1,684
bike	3,105	108	39
sign	2,021	248	252
bus	1,721	100	0
light	848	89	15
motor	730	42	1,425
truck	560	27	697

TABLE 5.1

Number of samples per class and per data partition in the FLIR ADAS dataset.

of interest. Since ADA-ViT requires the access to at least a weakly labeled image repository for augmentation, we leverage pretrained multiclass automatic detectors to localize and extract targets from the frames of this dataset. By doing so, we create an annotated external IR dataset that we use for sample selection and augmentation.

We use a YOLOv5x [96] model with pretrained weights on the COCO dataset [98], which is a RGB dataset that includes our targets of interest. We build an IR automatic detector by finetuning YOLOv5x on the training data of FLIR ADAS, since we have access to the annotated bounding boxes. We train the detector to localize and recognize the 8 targets from FLIR ADAS. Then, we evaluate the finetuned detector on the test set. During inference, our goal is to maximize the number of detections. Therefore, we set the confidence score threshold and the IoU threshold to low values of 0.5 and 0.45, respectively. This ensures the creation of a large IR dataset suitable for augmentation. However, setting low values to the confidence score and IoU can generate noisy and imprecise detections, or detections with incorrect labels.

The best checkpoint achieves a mean Average Precision (mAP) @ IoU=0.5 of 70%. We note that the performance of the IR detector could be further improved with a larger training set, meticulous hyperparameter tuning, or higher parameter thresholds during inference. However, obtaining the optimal detector performance is not primordial in the scope of this work. Moreover, the sub-optimal performance of the IR detector will be useful, in the context of this work, to demonstrate the effectiveness of our method in dealing with noise in the external image repository, which alleviates the need for perfectly labeled IR datasets used for augmentation. For example, Figure 5.2 shows the output of the detector which can mostly localize true targets with great precision, but it can occasionally generate false or imprecise detections, or detect true targets with the wrong labels. Thanks to the utility score function that ranks new samples based on their relevance, ADA-ViT should be able to filter out noisy detections by assigning low utility scores. Table 5.2 summarizes the number of obtained detections in each class.

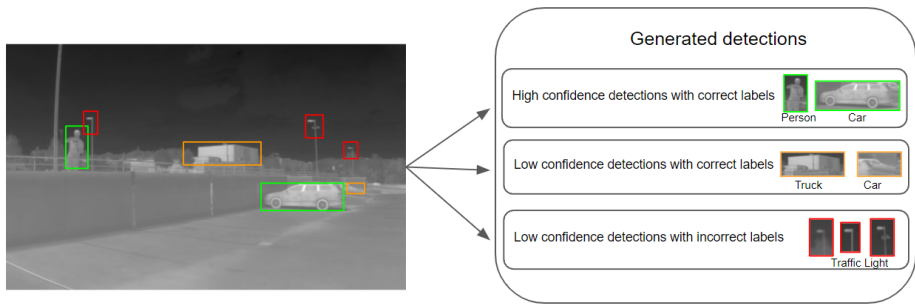


Figure 5.2: Generated detections by the finetuned YOLOv5x model from Brno dataset.

Classes	Number of samples
Total	15,407
car	10,487
person	2,531
bike	38
sign	52
bus	286
light	701
motor	24
truck	1288

TABLE 5.2

Number of generated detections per class by YOLO for the Brno dataset.

5.2 Experimental Analysis

RQ1: What is the impact of using ADA-ViT augmentation on the performance of an ATR system?

In this experiment, We compare the performance of a baseline ATR system (Section 5.2.1), trained using the existing training data only, against the performance of our approach (Section 5.2.2), where the ATR model is finetuned on the augmentations generated by ADA-ViT. We provide an in-depth analysis on the current model’s limitations and the performance gains achieved by our method.

5.2.1 Performance of a Baseline ViT without ADA-ViT Augmentation

We train our baseline ViT using the available training set of FLIR ADAS and evaluate it on the validation set at each epoch. After obtaining the optimal performance on the validation set, we evaluate the model on the held-out test set. In Table 5.3, we report the average testing accuracies across five runs.

Dataset	ViT-B	ViT-L
FLIR ADAS	94.2%	95.9%

TABLE 5.3

Accuracy results of baseline ATR systems trained on the original dataset only.

Figure 5.3 shows the confusion matrix of the baseline ATR model trained with the original data of FLIR ADAS only using ViT-B. We observe that some classes have higher misclassification rates, indicating that these classes might be more challenging to recognize. For instance, the classes *motor*, *truck* and *bike* have the highest misclassification rates. They are mostly confused with other classes that may be visually similar or share some common features.

In Figure 5.4 we display three sample images that were misclassified by the baseline, corresponding to the three classes with the highest misclassification rates: *motor*, *truck* and *bike*. Each row corresponds to a specific class. The first column displays test images that have been misclassified by the baseline. The remaining 5 columns show the 5 nearest neighbors to the misclassified test samples, taken from the training set. We also display the true label and the distance above each nearest neighbor.

We notice that all distances of the nearest neighbors are relatively large. This indicates that the test samples are located in sparse regions of the feature space. The first example shows an image of a *truck* partly occluded by a *car*, which led the model to confuse the *truck* with a *car*. The misclassification is further supported by the nearest neighbors of this test sample, which are all images of large cars that resemble trucks, such as vans and pickups. The second example is an image of a *bike* that has been misclassified as a *motor*. Due to the intrinsic properties of IR sensors, the bike appears dark and blends in with the background of the image, leaving the person riding

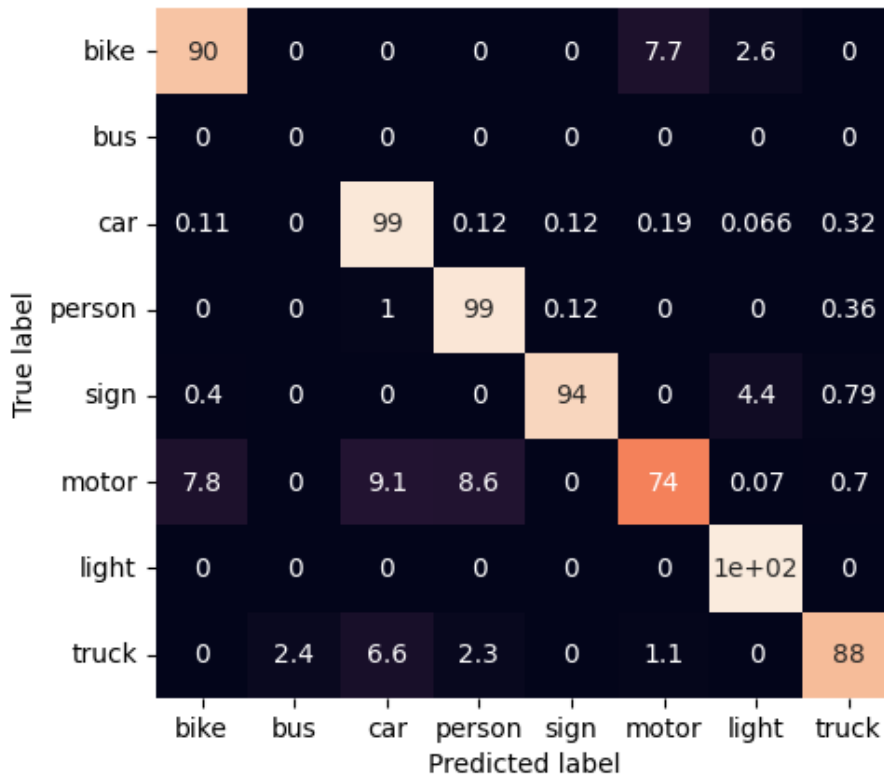


Figure 5.3: Confusion matrix of the baseline ViT-B model trained on FLIR ADAS before augmentation and evaluated on the test set. Note that the class *bus* is absent from the test set.

the bike to be more distinguished. Therefore, all the nearest neighbor images featured people riding motors. Finally, the last example displays an image of a *motor* along with parts of a vehicle. The nearest neighbors consistently show cropped images of cars, which led the model to confuse the *motor* as a *car*. All these misclassified test samples commonly display atypical features that are under-represented in the training set, thus preventing the model from predicting their correct classes. Our approach aims to guide the data augmentation process to feed images that contain similar features to these misclassified test samples, the model can eventually generalize better.

In Figure 5.5a, we display the 2D TSNE projection of the features corresponding to the training samples generated by the trained baseline model without augmentation. Overall, the classes appear to be separable. However, we notice the presence of gaps in the feature space that indicate under-represented regions responsible for misclassifications. In Figure 5.5b, we display the 2D TSNE projection of the features corresponding to the misclassified validation samples in the same feature space as Figure 5.5a. We see that most misclassifications are occurring in the regions of the feature space where there is little to no training examples. This observation confirms that the model does not generalize well when there are gaps in the feature space of the training data. Therefore, if we target these sparse regions and fill them with new training samples, the model would eventually be

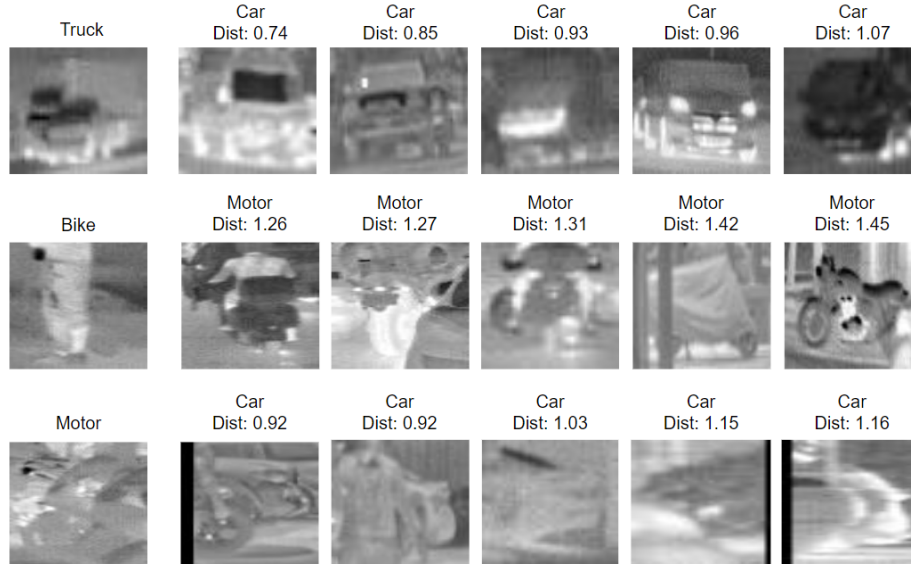


Figure 5.4: Examples of misclassified test images and their nearest neighbors using the baseline model for FLIR ADAS. The first column displays test images misclassified by the baseline. The remaining columns show the 5 nearest neighbors from the training set. We display the true label and the distance above each nearest neighbor.

able to perform better due to better class coverage.

5.2.2 Performance of a Baseline ViT with ADA-ViT Augmentation

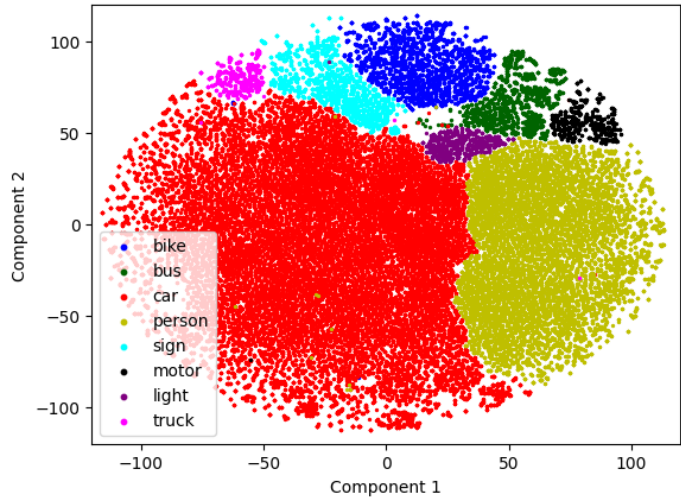
In this section, we present the results of the previous baseline ATR systems finetuned on the new selected samples by ADA-ViT, on top of the original training set. ADA-ViT selects new samples from the external IR image repository that can enhance the under-represented classes in the original training set and cover the sparse gaps in the training feature space. We run ADA-ViT for three iterations and, each time, we finetune the model on the augmented set and the original data. We present, in Table 5.4, the average testing accuracies across five runs for the FLIR ADAS dataset. We show the number of added samples by ADA-ViT in Table 5.5 for both ViT-B and ViT-L model architectures. We also show the new confusion matrix obtained from the finetuned ATR system on the ADA-ViT augmentation, in Figure 5.6.

Dataset	ViT-B	ViT-L
Original Dataset	94.2%	95.9%
ADA-ViT	96.4%	97.2%

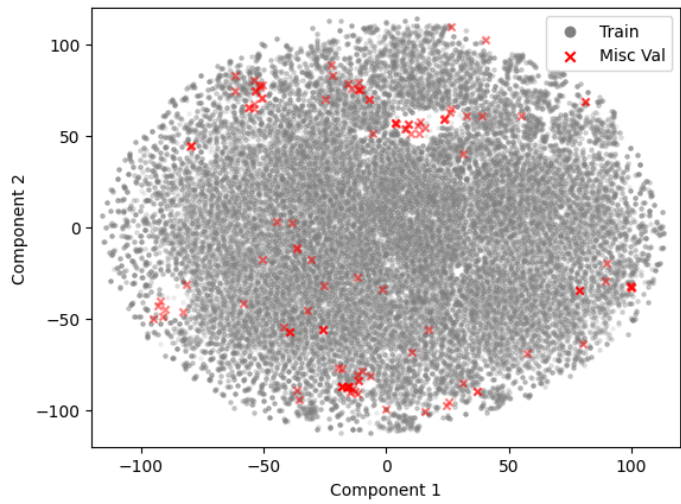
TABLE 5.4

Accuracy results of the ATR system finetuned on the original dataset and the ADA-ViT augmentations.

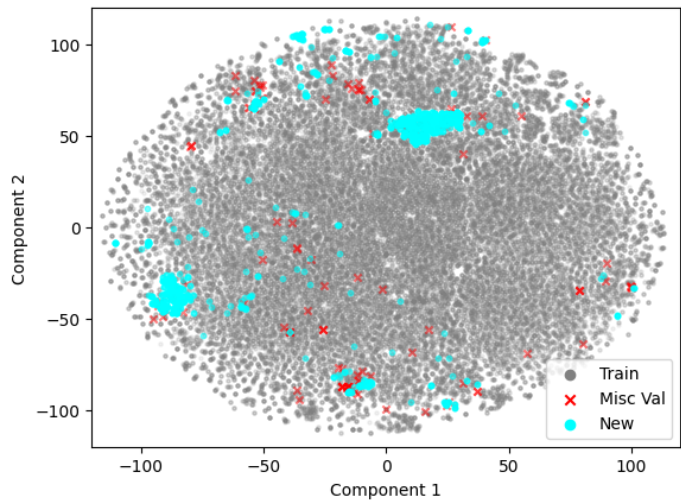
ADA-ViT augmentation yields significant performance gains over the baseline model trained



(a) TSNE projection of the training samples color-coded by class.



(b) TSNE projection of the training and misclassified validation samples.



(c) TSNE projection of the training, misclassified validation and new samples.

Figure 5.5: 2-D TSNE analysis of the training, misclassified validation, and new samples.

Model	Number of Added Samples
ViT-B	3,470
ViT-L	2,866

TABLE 5.5

Number of added samples by ADA-ViT after three iterations.

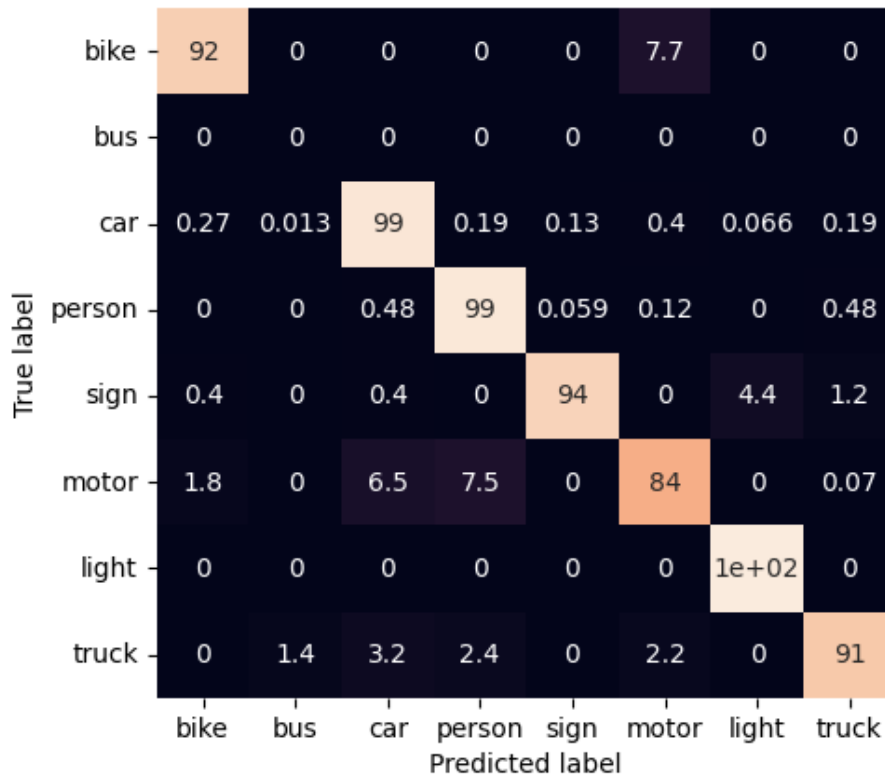


Figure 5.6: Confusion matrix of the ATR system finetuned on the original dataset and the ADA-ViT augmentations, and evaluated on the test set. Note that the class *bus* is absent from the test set.

without augmentation. Particularly, the ATR system is improved by more than 2% using ViT-B, and more than 1% using the larger model ViT-L. Moreover, we notice improvements in the accuracy of the classes that used to have high misclassification rates by the baseline model. This shows that ADA-ViT addresses the specific limitations of the model and aims to correct its misclassifications with carefully selected samples that can bridge the performance gap.

In Figure 5.7, we display the three test images from Figure 5.4 that were misclassified by the baseline. These images are now correctly classified by the finetuned model. Each row in Figure 5.7 corresponds to a class. The first column displays the test image, while the remaining columns show the 5 nearest neighbors to the test sample, taken from the training set. We notice that most of the nearest neighbors, marked by green boxes, are new images selected by ADA-ViT from the external

image repository. This indicates that the generated augmentations improved the model’s ability to generalize better to challenging data. The test samples are mapped to training images that display similar visual features. For example, in the first row, ADA-ViT adds a new sample that displays an occluded truck, similar to the test sample. We also notice an improvement in the data mapping shown by the smaller distances of the nearest neighbors, which proves that ADA-ViT is generating augmentations that enable the model to learn better representations of the data. By enhancing the under-represented classes and including challenging samples in the feature space, the model becomes more robust and improves its class representation and generalization capabilities to achieve a better accuracy.

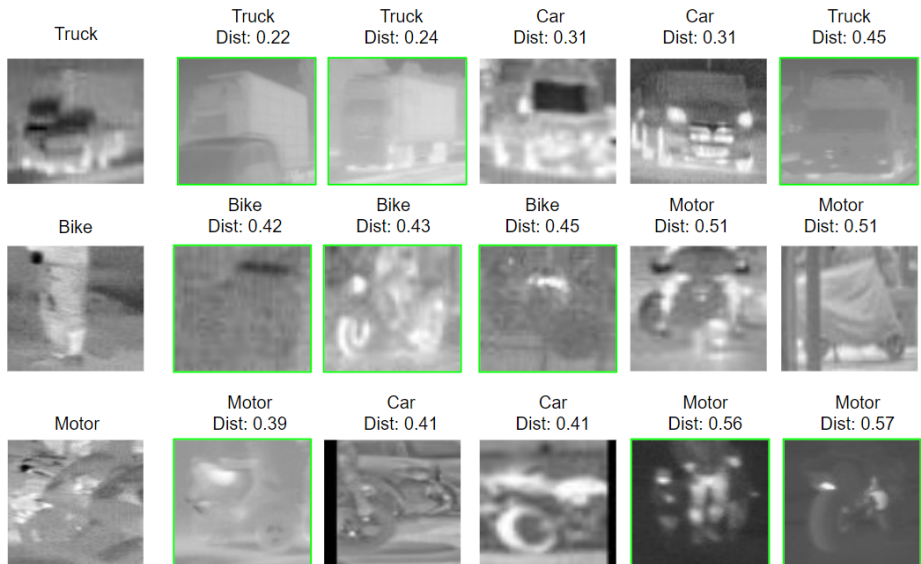


Figure 5.7: Corrected test samples by ADA-ViT from Figure 5.4, and their nearest neighbors from the training set using the model finetuned on ADA-ViT augmentations. The first column displays test images correctly classified by the finetuned model. The remaining columns show the 5 nearest neighbors from the training set. We display the true label and the distance above each nearest neighbor. The added images by ADA-ViT are marked by green boxes.

In Figure 5.5c, we add the TSNE projection of the new selected images by ADA-ViT for augmentation in the same feature space as Figure 5.5b. The added images are seen to overlap with the misclassified validation samples, and occupy the regions where the previous model errors occurred. This demonstrates that ADA-ViT aims to fill in the gaps in the feature space and augments the training data with new samples that can cover under represented regions responsible for the misclassifications.

5.3 Importance of Guiding Data Augmentation by ADA-ViT scoring

RQ2: What is the impact of using ADA-ViT utility score function on guiding sample selection?

Under this task, we investigate the importance of guiding the selected IR augmentations from the external image datasets using ADA-ViT utility score function. We compare three scenarios of selecting samples to use as additional augmentation for the training data. In each scenario, we augment the training set using the same number of samples. We only vary the selection criteria: Random sampling vs. Confidence-based sampling vs. ADA-ViT-guided sampling. We describe the experiments as follows:

- **Experiment 1:** ViT baseline trained with the original training data without augmentation.
- **Experiment 2:** Trained ViT baseline finetuned on the original training data, in addition to a subset from the external data of size N (Equation 3.2, Section 3.1) selected by applying ADA-ViT scoring strategy.
- **Experiment 3:** Trained ViT baseline finetuned on the original training data, in addition to a random subset from the external data with the same size N as Experiment#2.
- **Experiment 4:** Trained ViT baseline finetuned on the original training data, in addition to a subset from the external data corresponding to under-performing samples, with the same size N as Experiment#2. An under-performing sample can be either a misclassified sample or a correctly classified sample with low confidence. For the latter case, we set a threshold on the predicted confidences of correctly classified samples. This threshold corresponds to the lower outlier boundary, calculated using: $Q_1 - 1.5 \times IQR$, with Q_1 being the lower quartile and IQR the interquartile range. The outlier boundary sets a statistical fence for a data distribution, beyond which a data point is considered an outlier. In this context, an under-represented sample can be viewed as an outlier, since there are not enough data points from the training set that share similar features with it.

Table 5.6 shows the classification accuracies on FLIR ADAS dataset for the four different settings, described above. We see that all augmentations succeeded to improve the classification accuracy. However, using the ADA-ViT scoring function to rank and select samples gives the best improvement, followed by the confidence-based selection, and, finally, the random selection. The random sampling tends to select either easy samples that are already correctly classified and will not add any relevant information to the model, or it may select noisy samples with incorrect labels, leading to further confusion to the model. The confidence-based sampling succeeds to lead the model to learn from the challenging regions of the feature space by adding hard samples. However, the selected images do not necessarily come from the under-represented regions, as they may be outliers describing noisy samples. The ADA-ViT scoring and sample selection strategy enables the model to learn from challenging samples that can cover the under-represented regions of the feature space, while ensuring that noisy samples are not being selected.

Dataset	FLIR ADAS	
Baseline model	ViT-B	ViT-L
Original dataset	94.2%	95.9%
Random	94.7%	96.3%
Confidence	95.3 %	96.4 %
ADA-ViT	96.4%	97.2%

TABLE 5.6

Comparison of the accuracy of the ViT with different data selection methods.

5.4 Comparison with state-of-the-art Data Augmentation Techniques

RQ3: How well does ADA-ViT perform compared to state-of-the-art data augmentation techniques?

To illustrate the advantage of ADA-ViT, we evaluate its performance on the FLIR ADAS dataset against three state-of-the-art data augmentation methods: a combination of techniques that generate augmentations from the current training set [24, 25, 28], a Meta-Set based method [19], and BRACE [36]. For this comparison, the four models are provided with the same input data which is split into training, validation and test subsets. We also feed them the same external IR datasets, in case the method performs augmentation from external image repositories. All of the considered data augmentation techniques reported great success on several RGB benchmark datasets. However, their effectiveness on an infrared application has not been previously explored and requires further investigation. In this experiment, we study and analyze their performance for an infrared scenario, and compare them to our proposed data augmentation technique. We use the validation set to tune the parameters of these techniques on the studied dataset.

- **Local** [24, 25, 28]: We use a combination of data augmentation techniques that generate new samples by applying transformations to the original training set, thus the *Local* notation. These augmentations are typically done in-place during each epoch or mini-batch of training. We use CutMix [24], Mixup [25], and AutoAugment [28] augmentations. CutMix randomly removes parts from an image and replaces them with patches from another image. Mixup creates new samples by generating a weighted combination of random image pairs from the training data. Finally, AutoAugment is an automated approach that searches for the best transformation policy among several augmentation operations, such as translation, rotation, or shearing, and the probabilities and magnitudes with which the functions are applied.
- **Meta-Set** [19]: This method requires the access to external image repositories for augmentation. Specifically, this data augmentation technique proposes a framework that can learn directly from web datasets. To address the noise in the web training sets, it learns two networks

to distinguish in- and out-of distribution samples, and to correct the labels of in-distribution noisy data, guided by a small amount of clean meta-set. The goal is to alleviate the harmful effects caused by out-of-distribution noise and properly exploit all of the in-distribution samples for training. This method is relevant in our comparison because it operates in an opposite way as ADA-ViT. While our method aims to select only the relevant samples, the Meta-Set approach focuses on filtering out noisy samples and adding all the in-distribution images to the training set, without further selection.

- **BRACE** [36]: This method also uses external image repositories. It is the most relevant work to ADA-ViT since it addresses the issue of under-represented regions in the training feature space. Similar to our work, BRACE uses a utility function to rank the new samples based on their relevance and their potential contribution to improving the model performance. This method is only applicable to CNN-based models, because it leverages concept-based model explanations extracted from post-hoc explanation methods, such as GradCam [37], or extracted from special CNN architectures that are interpretable, such as Comprehensible CNNs [38]. In our comparison, we implement BRACE with GradCam, as this option yielded the best results according to the paper [36]. We use a CNN backbone of ResNet-200 [3] as baseline, which has almost the same number of parameters as ViT-Base.

Dataset	FLIR ADAS		
	ViT-B	ViT-L	CNN
Original dataset	94.2%	95.9%	91.5%
Local [24, 25, 28]	94.8%	96.2%	N/A
Meta-Set [19]	95.3%	96.25%	N/A
BRACE [36]	N/A	N/A	93.1%
ADA-ViT	96.4%	97.2%	N/A

TABLE 5.7

Comparison of classification accuracies of different data augmentation techniques on FLIR ADAS dataset. We run ADA-ViT for 3 iterations. For the CNN baseline, we only report the accuracy of BRACE and we do not run other augmentation methods with the CNN baseline as this is outside the scope of our research.

Table 5.7 shows the classification accuracies of the ATR system trained with ADA-ViT augmentation, as well as its performance when trained with other data augmentation techniques. We report our results using a ViT base (ViT-B) and ViT large (ViT-L) transformer baselines, on FLIR ADAS.

We observe that ADA-ViT augmentation consistently outperforms other methods. In particular, our method improves the baseline performance by more than 2% using the smaller model ViT-B, and more than 1% using the larger model ViT-L. We note that ViT-B tends to achieve

better performance improvement than the larger model ViT-L, which is mainly due to differences in the initial model performance (before augmentation). This finding is expected, since smaller models are less able to learn from the data than their larger counterparts with their complex architectures. However, we want to highlight the fact that, even though ViT-L is a large model with a high initial performance, it still benefited from our augmentation.

The *Local* augmentation failed to improve the performance of the baseline model, compared to the other approaches, as this method is constrained to the local neighborhood of the existing training set. Therefore, the created samples cannot recover the missing features from the training set and complete the under-represented regions in the training feature space.

As shown in Table 5.8, the Meta-Set approach adds the highest number of images, compared to the other methods. Even though it adds 5 times more images than ADA-ViT, the improvements of Meta-Set are still falling short. This indicates that selecting fewer samples that target under-represented regions only is sufficient to improve the performance of the classifier, without unnecessarily increasing the task complexity.

Augmentation Method	Number of added images
Local	9,324
Meta	15,407
Brace	6,119
ADA-ViT	3,470

TABLE 5.8

Number of images added by each data augmentation method for FLIR ADAS.

The BRACE method, which is a CNN-based approach, is behind the other data augmentation techniques, mainly because the CNN baseline scored lower than the transformer baseline. This observation serves to confirm the findings of recent studies [1,16,41,47] that highlighted the advanced learning capabilities of attention-based models and showed that transformers are inherently more robust than CNNs. Nevertheless, BRACE still managed to increase the accuracy of the baseline CNN by around 2%, which highlights the importance of selective data augmentation. This gap in performance can be explained by two main factors. First, as discussed in Section 4.4, this can be an indicator of a flaw in the BRACE utility score, which represents classes by a single data point that corresponds to the mean of features of all training samples, as opposed to ADA-ViT which employs clustering to generate more robust class representations. This issue becomes more highlighted for the case of datasets with high intra-class variation, such as IR data. Second, the under-performance of BRACE, compared to ADA-ViT, can reflect the advantage of using the attention weights learnt inside the transformer itself to identify concepts that led to the misclassification, over post-hoc explanation methods, such as GradCam, or pretrained object detectors, such as RCNN, which are agnostic to the task and dataset in hand.

5.5 Chapter Summary

In this chapter, we discussed the experimental results of our proposed data augmentation framework evaluated on an infrared application. We conducted our experiments on an infrared benchmark dataset that describes various targets captured at different times of the day and different weather conditions. This led to the creation of an under-representative training set that suffers from sparse gaps in the feature space. First, we conducted an in-depth analysis of the issue of under-represented regions in the training feature space of an infrared dataset, and showcased its harmful impact on the model performance. Then, we illustrated how ADA-ViT operates to address this issue in an IR scenario, by adding new samples that can cover these sparse regions of the feature space. We also carried out an experiment to demonstrate the advantage of using ADA-ViT to guide the sample selection from external IR datasets over random data sampling or confidence-based augmentations. Finally, we compared the performance of our proposed approach with other carefully selected state-of-the-art data augmentation techniques. The purpose of this experiment is to highlight the importance of considering under-represented regions in the training data of IR datasets when applying data augmentation. We showed that our method achieves the highest performance improvements while adding the least number of samples, compared to other techniques that have been evaluated on RGB datasets only. This proves that ADA-ViT is able to train robust ATR systems that can generalize well to the challenging IR test data, while requiring fewer samples for training.

CHAPTER 6

CONCLUSIONS AND POTENTIAL FUTURE WORKS

6.1 Conclusions

In this thesis, we addressed the challenge of improving the performance of Vision Transformer models trained with under-representative training sets by proposing a data augmentation technique, called ADA-ViT, that aims to expand the data with relevant samples, capable of improving the diversity and class coverage of training sets. Our method leverages the attention mechanism of transformer models to understand the model vulnerabilities and learning limitations. We also make novel use of the validation set to analyze the model performance and extract visual model explanations that justify the misclassifications. This is because, assuming that the validation and test sets are drawn from the same distribution, the model performance on the validation set can be a good indicator for the performance on the unseen test set.

We call for external image repositories to search for new samples that display similar visual features to the extracted misclassification concepts. These external repositories can be noisy and weakly labeled. In other words, they can be labeled automatically with several incorrect labels. Our search for candidate samples is guided by a utility score function that we carefully designed to rank the new samples based on their relevance and their potential contribution to improving the data diversity, and eventually the model performance. Our proposed utility function takes into consideration the degree to which a new sample falls in the under-represented regions of the feature space of the training data, as well as the degree to which it matches the features of the hard samples extracted from the validation set. The utility score function also considers the cases where the external data repository is noisy, and applies a penalty score to out-of-distribution samples or in-distribution samples with noisy labels.

One of the key contributions of this work is the investigation of the problem of under-represented regions in the training feature space for the case of Vision Transformers. While the research on data augmentation techniques has been ongoing for a long time, few works prioritized the aspect of data diversity over data size expansion. In particular, Vision Transformers have been excluded from this research, even though they have achieved significant success and are increasingly being adopted in various contexts. Our work aims to bridge this gap and include Vision Transformers in the currently active research areas.

Moreover, the potential of the attention mechanism has not been fully exploited yet for the

purpose of model interpretability. Our method proposes a novel way of utilizing attention weights to explain the model decision and reveal potential learning limitations. Finally, our data augmentation framework circumvents the issue of acquiring large-scale labeled data for augmentation. Instead, it learns directly from large web datasets that are easier and cheaper to acquire. Regarding the sample and label noise that characterize web datasets, we do not employ complicated mechanisms to clean these data repositories. Instead, we adopt a different approach based on sample selection to only retrieve the relevant images that can remediate the model vulnerabilities and cover the under-represented regions in the training feature space, without the hassle of anomaly detection or label noise correction.

We applied our proposed strategy to an application involving RGB benchmark datasets, that varied in size and class granularity. We showed that our method is able to improve the data diversity and cover the gaps in the training feature space with new samples, in the case of small and larger datasets, with fine-grained or coarser classes. We conducted experiments to demonstrate and visualize the issue of under-representative training sets and showcased how our proposed data augmentation technique is able to add samples that occupy the sparse empty regions of the feature space. We also presented an in-depth analysis of the proposed utility score function to justify its design and show its importance to guide the sample selection process.

We evaluated our method on three RGB datasets: CUB, CUB-Families and TinyImageNet. Our results show that ADA-ViT is able to significantly improve the accuracy of a baseline model trained without augmentation by more than 3% on CUB-Families, around 2% on CUB, and 1% on TinyImageNet. We also compared our method with various state-of-the-art data augmentation techniques that do not consider the issue of under-represented regions of the feature space in their augmentation process. Our results reveal that ADA-ViT significantly outperforms these compared methods while being the least complex and adding fewer images for augmentation.

In addition to object recognition in standard RGB images, we applied the proposed strategy to Automatic Target Recognition (ATR) using infrared imagery. Our method alleviates the need for large labeled secondary IR datasets for augmentation. Instead, we leverage automatic detectors to generate weakly annotated datasets that are diverse enough to include samples capable of covering the under-represented regions of the training set. While maximizing the number of detections to ensure the diversity of the secondary image repository, the automatic detector may generate false detections or true targets with noisy labels in the process. Our method is able to handle the noise in the secondary datasets using the utility score function that we designed to assign lower ranking to out-of-distribution samples and penalize misdetections. By selecting only the relevant samples that are capable of enhancing the class diversity and covering the sparse gaps in the training set, the model is guided to learn from the challenging regions of the feature space, thus demonstrating enhanced robustness and accuracy of ATR models in challenging environments compared to existing

methods. We evaluated our method on the FLIR ADAS dataset, which contains targets captured at different environmental conditions. Our data augmentation technique is seen to improve the performance of a baseline model by over 2% while requiring the least number of added samples. This proves that the research in data augmentation should focus more on data quality and class representativeness over size expansion.

6.2 Potential Future Work

Our proposed data augmentation technique constitutes an effort, among many, to develop more robust and accurate machine learning systems in the face of limited data and an increasing demand for high-performance models in various applications. The promising results obtained in this dissertation open several avenues for future research. Potential directions for extending this work include:

- Extending the use of ADA-ViT to semi-supervised models, where we can leverage large unlabeled datasets for augmentation.
- Experimenting with more sophisticated Vision Transformer models that utilize more advanced self attention-based architectures, such as TransFG [90].
- Extending ADA-ViT to a RGB-Infrared fusion algorithm by learning and adding samples from both data types.
- Applying the proposed data augmentation strategy to other domains and tasks, such as natural language processing, medical imaging, or audio signal processing, to assess their effectiveness and adaptability in different contexts

REFERENCES

- [1] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby, “An image is worth 16x16 words: Transformers for image recognition at scale,” 10 2020.
- [2] “Flir. free flir thermal dataset for algorithm training.,” .
- [3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, “Deep residual learning for image recognition,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.
- [4] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger, “Densely connected convolutional networks,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2261–2269.
- [5] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi, “You only look once: Unified, real-time object detection,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788.
- [6] Mingxing Tan, Ruoming Pang, and Quoc V. Le, “Efficientdet: Scalable and efficient object detection,” in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 10778–10787.
- [7] Hao Shao, Letian Wang, Ruobing Chen, Steven L. Waslander, Hongsheng Li, and Yu Liu, “Reasonnet: End-to-end driving with temporal and global reasoning,” in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 13723–13733.
- [8] Lei Ke, Shichao Li, Yanan Sun, Yu-Wing Tai, and Chi-Keung Tang, *GSNet: Joint Vehicle Pose and Shape Reconstruction with Geometrical and Scene-Aware Supervision*, pp. 515–532, 11 2020.
- [9] Yiqiang Chen, Wang Lu, Xin Qin, Jindong Wang, and Xing Xie, “Metafed: Federated learning among federations with cyclic knowledge distillation for personalized healthcare,” *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–12, 2023.
- [10] Chaoyi Wu, Xiaoman Zhang, Ya Zhang, Yanfeng Wang, and Weidi Xie, “Pmc-llama: Further finetuning llama on medical papers,” 04 2023.
- [11] Nada Baili, Mahdi Moalla, Hichem Frigui, and Andrew Karem, “Multistage approach for automatic target detection and recognition in infrared imagery using deep learning,” *Journal of Applied Remote Sensing*, vol. 16, 11 2022.
- [12] Zewen Li, Fan Liu, Wenjie Yang, Shouheng Peng, and Jun Zhou, “A survey of convolutional neural networks: Analysis, applications, and prospects,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 12, pp. 6999–7019, 2022.
- [13] Wim De Mulder, Steven Bethard, and Marie-Francine Moens, “A survey on the application of recurrent neural networks to statistical language modeling,” *Computer Speech Language*, vol. 30, 01 2014.
- [14] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, ukasz Kaiser, and Illia Polosukhin, “Attention is all you need,” in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2017, NIPS’17, p. 6000–6010, Curran Associates Inc.

- [15] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo, “Swin transformer: Hierarchical vision transformer using shifted windows,” in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 9992–10002.
- [16] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jegou, Julien Mairal, Piotr Bojanowski, and Armand Joulin, “Emerging properties in self-supervised vision transformers,” in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 9630–9640.
- [17] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou, “Training data-efficient image transformers distillation through attention,” 12 2020.
- [18] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta, “Revisiting unreasonable effectiveness of data in deep learning era,” 07 2017.
- [19] Chuanyi Zhang, Yazhou Yao, Xiangbo Shu, Zechao Li, Zhenmin Tang, and Qi Wu, “Data-driven meta-set based fine-grained visual recognition,” in *Proceedings of the 28th ACM International Conference on Multimedia*, New York, NY, USA, 2020, MM ’20, p. 2372–2381, Association for Computing Machinery.
- [20] Zhe Xu, Shaoli Huang, Ya Zhang, and Dacheng Tao, “Webly-supervised fine-grained visual categorization via deep domain adaptation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 5, pp. 1100–1113, 2018.
- [21] Khoshgoftaar Shorten and Connor Taghi M., “A survey on image data augmentation for deep learning,” *Journal of Big Data*, vol. 6, 2019.
- [22] Terrance DeVries and Graham Taylor, “Improved regularization of convolutional neural networks with cutout,” 08 2017.
- [23] Krishna Singh, Hao Yu, Aron Sarmasi, Gautam Pradeep, and Yong Lee, “Hide-and-seek: A data augmentation technique for weakly-supervised localization and beyond,” 11 2018.
- [24] Sangdoon Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo, “Cutmix: Regularization strategy to train strong classifiers with localizable features,” *CoRR*, vol. abs/1905.04899, 2019.
- [25] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz, “mixup: Beyond empirical risk minimization,” in *International Conference on Learning Representations*, 2018.
- [26] Shaoli Huang, Xinchao Wang, and Dacheng Tao, “Snapmix: Semantically proportional mixing for augmenting fine-grained data,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 2, pp. 1628–1636, May 2021.
- [27] Tao Hu and Honggang Qi, “See better before looking closer: Weakly supervised data augmentation network for fine-grained visual classification,” *ArXiv*, vol. abs/1901.09891, 2019.
- [28] Ekin Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc Le, “Autoaugment: Learning augmentation policies from data,” 05 2018.
- [29] Sungheon Park and Nojun Kwak, “Analysis on the dropout effect in convolutional neural networks,” 03 2017, pp. 189–204.
- [30] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio, “Generative adversarial networks,” *Communications of the ACM*, vol. 63, pp. 139 – 144, 2014.
- [31] Romain Lopez, Pierre Boyeau, Nir Yosef, Michael I. Jordan, and Jeffrey Regier, “Auto-encoding variational bayes,” 2020.
- [32] Antreas Antoniou, Amos Storkey, and Harrison Edwards, “Data augmentation generative adversarial networks,” 11 2017.
- [33] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila, “Training generative adversarial networks with limited data,” *ArXiv*, vol. abs/2006.06676, 2020.

- [34] Luis Perez and Jason Wang, “The effectiveness of data augmentation in image classification using deep learning,” *ArXiv*, vol. abs/1712.04621, 2017.
- [35] Tommaso Dreossi, Shromona Ghosh, Xiangyu Yue, Kurt Keutzer, Alberto Sangiovanni-Vincentelli, and Sanjit Seshia, “Counterexample-guided data augmentation,” 05 2018.
- [36] Sandareka Wickramanayake, Wynne Hsu, and Mong Li Lee, “Explanation-based data augmentation for image classification,” in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, Eds. 2021, vol. 34, pp. 20929–20940, Curran Associates, Inc.
- [37] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra, “Grad-cam: Visual explanations from deep networks via gradient-based localization,” in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 618–626.
- [38] Sandareka Wickramanayake, Wynne Hsu, and Mong Lee, “Comprehensible convolutional neural networks via guided concept learning,” 07 2021, pp. 1–8.
- [39] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” in *Advances in Neural Information Processing Systems*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, Eds. 2015, vol. 28, Curran Associates, Inc.
- [40] Yao Qin, Dongjin Song, Haifeng Chen, Wei Cheng, Guofei Jiang, and Garrison Cottrell, “A dual-stage attention-based recurrent neural network for time series prediction,” *arXiv preprint arXiv:1704.02971*, 2017.
- [41] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin, “Attention is all you need,” 2017.
- [42] Sepp Hochreiter and Jürgen Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, pp. 1735–80, 12 1997.
- [43] Shaowei Yao and Xiaojun Wan, “Multimodal transformer for multimodal machine translation,” in *Proceedings of the 58th annual meeting of the association for computational linguistics*, 2020, pp. 4346–4350.
- [44] Usman Naseem, Imran Razzak, Katarzyna Musial, and Muhammad Imran, “Transformer based deep intelligent contextual embedding for twitter sentiment analysis,” *Future Generation Computer Systems*, vol. 113, pp. 58–69, 2020.
- [45] Taihua Shao, Yupu Guo, Honghui Chen, and Zepeng Hao, “Transformer-based neural network for answer selection in question answering,” *IEEE Access*, vol. 7, pp. 26146–26156, 2019.
- [46] Alec Radford and Karthik Narasimhan, “Improving language understanding by generative pre-training,” 2018.
- [47] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” 2018.
- [48] Chun-Fu Richard Chen, Quanfu Fan, and Rameswar Panda, “Crossvit: Cross-attention multi-scale vision transformer for image classification,” in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 357–366.
- [49] Luowei Zhou, Hamid Palangi, Lei Zhang, Houdong Hu, Jason Corso, and Jianfeng Gao, “Unified vision-language pre-training for image captioning and vqa,” in *Proceedings of the AAAI conference on artificial intelligence*, 2020, vol. 34, pp. 13041–13049.
- [50] Zhao Yang, Jiaqi Wang, Yansong Tang, Kai Chen, Hengshuang Zhao, and Philip HS Torr, “Lavt: Language-aware vision transformer for referring image segmentation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 18155–18165.

- [51] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko, “End-to-end object detection with transformers,” in *European conference on computer vision*. Springer, 2020, pp. 213–229.
- [52] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai, “Bevformer: Learning bird’s-eye-view representation from multi-camera images via spatiotemporal transformers,” in *European conference on computer vision*. Springer, 2022, pp. 1–18.
- [53] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi, “Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models,” *arXiv preprint arXiv:2301.12597*, 2023.
- [54] Jiajun Deng, Zhengyuan Yang, Daqing Liu, Tianlang Chen, Wengang Zhou, Yanyong Zhang, Houqiang Li, and Wanli Ouyang, “Transvg++: End-to-end visual grounding with language conditioned vision transformer,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.
- [55] Haiyang Tang, “Vision question answering system based on roberta and vit model,” in *2022 International Conference on Image Processing, Computer Vision and Machine Learning (ICIP-CML)*. IEEE, 2022, pp. 258–261.
- [56] Xiaojian Ma, Weili Nie, Zhiding Yu, Huaizu Jiang, Chaowei Xiao, Yuke Zhu, Song-Chun Zhu, and Anima Anandkumar, “Relvit: Concept-guided vision transformer for visual relational reasoning,” *arXiv preprint arXiv:2204.11167*, 2022.
- [57] Yun-Hao Cao and Jianxin Wu, “A random cnn sees objects: One inductive bias of cnn and its applications,” in *Proceedings Of The AAAI Conference On Artificial Intelligence*, 2022, vol. 36, pp. 194–202.
- [58] Xiangning Chen, Cho-Jui Hsieh, and Boqing Gong, “When vision transformers outperform resnets without pre-training or strong data augmentations,” *arXiv preprint arXiv:2106.01548*, 2021.
- [59] B Li, R Chellappa, Q Zheng, S Der, N Nasrabadi, L Chan, and L Wang, “Experimental evaluation of flir atr approaches—a comparative study,” *Computer Vision and Image Understanding*, vol. 84, no. 1, pp. 5–24, 2001.
- [60] Greer J Gray, Nabil Aouf, Mark A Richardson, Brian Butters, Roy Walmsley, and Edgar Nicholls, “Feature-based target recognition in infrared images for future unmanned aerial vehicles,” *Journal of Battlefield Technology*, vol. 14, no. 2, pp. 27–36, 2011.
- [61] David G. Lowe, “Distinctive image features from scale-invariant keypoints,” *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004.
- [62] Zhi-Guo Cao, Xuan Zhang, and Wenwu Wang, “Forward-looking infrared target recognition based on histograms of oriented gradients,” pp. 27–, 11 2011.
- [63] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*, 2005, vol. 1, pp. 886–893 vol. 1.
- [64] Theodoros Evgeniou and Massimiliano Pontil, “Support vector machines: Theory and applications,” 01 2001, vol. 2049, pp. 249–257.
- [65] N. M. Nasrabadi, “Deeptarget: An automatic target recognition using deep convolutional neural networks,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 55, no. 6, pp. 2687–2697, 2019.
- [66] Antoine d’Acremont, Ronan Fablet, Alexandre Baussard, and Guillaume Quin, “Cnn-based target recognition and identification for infrared imaging in defense systems,” *Sensors*, vol. 19, no. 9, pp. 2040, Apr 2019.

- [67] Iain Rodger, Barry Connor, and Neil M. Robertson, “Classifying objects in LWIR imagery via CNNs,” in *Electro-Optical and Infrared Systems: Technology and Applications XIII*, David A. Huckridge, Reinhard Ebert, and Stephen T. Lee, Eds. International Society for Optics and Photonics, 2016, vol. 9987, pp. 152 – 165, SPIE.
- [68] E. Gundogdu, A. Koç, and A. A. Alatan, “Object classification in infrared images using deep representations,” in *2016 IEEE International Conference on Image Processing (ICIP)*, 2016, pp. 1066–1070.
- [69] S. Kim, W. Song, and S. Kim, “Infrared variation optimized deep convolutional neural network for robust automatic ground target recognition,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2017, pp. 195–202.
- [70] Xiaodong Cheng Xiaoping Liu Xiaohui Hu, Yufeng Huang and Huan Zhao, “Transfer learning-based automatic target recognition in sar images,” in *IEEE Transactions on Geoscience and Remote Sensing*, 2018, vol. 56, p. 3187–3200.
- [71] Ruiming Liu, Yanhong Lu, Chenglong Gong, and Yang Liu, “Infrared point target detection with improved template matching,” *Infrared Physics Technology*, vol. 55, no. 4, pp. 380–387, 2012.
- [72] Bin Wang, Guorui Ma, Haigang Sui, Yongxian Zhang, Haiming Zhang, and Yuan Zhou, “Few-shot object detection in remote sensing imagery via fuse context dependencies and global features,” *Remote Sensing*, vol. 15, no. 14, 2023.
- [73] Nikesh Devkota and Byung Wook Kim, “Deep learning-based small target detection for satellite-ground free space optical communications,” *Electronics*, vol. 12, no. 22, pp. 4701, 2023.
- [74] Lili Zhang, Xiuhui Wang, Qifu Bao, Bo Jia, Xuesheng Li, and Yaru Wang, “Infrared fault classification based on the siamese network,” *Applied Sciences*, vol. 13, no. 20, 2023.
- [75] Xin Zhao, Xiaoling Lv, Jinlei Cai, Jiayi Guo, Yueting Zhang, Xiaolan Qiu, and Yirong Wu, “Few-shot sar-atr based on instance-aware transformer,” *Remote Sensing*, vol. 14, no. 8, 2022.
- [76] Ethan R. Adams, Arthur C. Depoian II, Aidan G. Kurz, Colleen P. Bailey, and Parthasarathy Guturu, “Automatic target detection utilizing an edge IR vision transformer (EIR-ViT),” in *Automatic Target Recognition XXXIII*, Riad I. Hammoud, Timothy L. Overman, and Abhijit Mahalanobis, Eds. International Society for Optics and Photonics, 2023, vol. 12521, p. 125210T, SPIE.
- [77] Mark J. van der Laan and Sandrine Dudoit, “Unified cross-validation methodology for selection among estimators and a general cross-validated adaptive epsilon-net estimator: Finite sample oracle inequalities and examples,” 2003.
- [78] “Online bird guide,” <https://www.allaboutbirds.org/>, Accessed: 2022-11-02.
- [79] Samira Abnar and Willem H. Zuidema, “Quantifying attention flow in transformers,” *CoRR*, vol. abs/2005.00928, 2020.
- [80] Tianyi Zhang, Felix Wu, Arzoo Katiyar, Kilian Q Weinberger, and Yoav Artzi, “Revisiting few-sample bert fine-tuning,” *arXiv preprint arXiv:2006.05987*, 2020.
- [81] Zehui Lin, Pengfei Liu, Luyao Huang, Junkun Chen, Xipeng Qiu, and Xuanjing Huang, “Dropattention: A regularization method for fully-connected self-attention networks,” *ArXiv*, vol. abs/1907.11065, 2019.
- [82] Mohammad Saeed Masiha, Amin Gohari, Mohammad Hossein Yassaee, and Mohammad Reza Aref, “Learning under distribution mismatch and model misspecification,” in *2021 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2021, pp. 2912–2917.
- [83] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, “Caltch-uscdb birds-200-2011 dataset,” 2011.

- [84] Tianshui Chen, Wenxi Wu, Yuefang Gao, Le Dong, Xiaonan Luo, and Liang Lin, “Fine-grained representation learning and recognition by exploiting hierarchical semantic embedding,” in *Proceedings of the 26th ACM International Conference on Multimedia*, New York, NY, USA, 2018, MM ’18, p. 2023–2031, Association for Computing Machinery.
- [85] Ya Le and Xuan S. Yang, “Tiny imagenet visual recognition challenge,” 2015.
- [86] “Species limits and taxonomy in birds,” <https://americanornithology.org/species-limits-and-taxonomy-in-birds/>, Accessed: 2022-10-18.
- [87] Yongqin Xian, Christoph Lampert, Bernt Schiele, and Zeynep Akata, “Zero-shot learning - a comprehensive evaluation of the good, the bad and the ugly,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PP, 07 2017.
- [88] Tal Ridnik, Emanuel Ben-Baruch, Asaf Noy, and Lihi Zelnik-Manor, “Imagenet-21k pretraining for the masses,” 2021.
- [89] Zhe Xu, Shaoli Huang, Ya Zhang, and Dacheng Tao, “Webly-supervised fine-grained visual categorization via deep domain adaptation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 5, pp. 1100–1113, 2018.
- [90] Ju He, Jie-Neng Chen, Shuai Liu, Adam Kortylewski, Cheng Yang, Yutong Bai, and Changhu Wang, “Transfg: A transformer architecture for fine-grained recognition,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022, vol. 36, pp. 852–860.
- [91] “t-sne clearly explained,” <https://towardsdatascience.com/t-sne-clearly-explained-d84c537f53a>, Accessed: 2024-04-03.
- [92] Alex Krizhevsky, “Learning multiple layers of features from tiny images,” 2009.
- [93] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Ng, “Reading digits in natural images with unsupervised feature learning,” *NIPS*, 01 2011.
- [94] Seok Yoon, Taek Lyul Song, and Tae Kim, “Automatic target recognition and tracking in forward-looking infrared image sequences with a complex background,” *International Journal of Control, Automation and Systems*, vol. 11, 02 2013.
- [95] Nada Baili, Mahdi Moalla, Hichem Frigui, and Andrew D. Karem, “Multistage approach for automatic target detection and recognition in infrared imagery using deep learning,” *Journal of Applied Remote Sensing*, vol. 16, no. 4, pp. 048505, 2022.
- [96] Glenn Jocher, Alex Stoken, Jirka Borovec, NanoCode012, ChristopherSTAN, Liu Changyu, Laughing, tkianai, Adam Hogan, lorenzomamma, yxNONG, AlexWang1900, Laurentiu Diaconu, Marc, wanghaoyang0106, ml5ah, Doug, Francisco Ingham, Frederik, Guillhen, Hatovix, Jake Poznanski, Jiacong Fang, Lijun Yu, changyu98, Mingyu Wang, Naman Gupta, Osama Akhtar, PetrDvoracek, and Prashant Rai, “ultralytics/yolov5: v3.1 - Bug Fixes and Performance Improvements,” Oct. 2020.
- [97] Adam Ligocki, Ales Jelinek, and Ludek Zalud, “Brno urban dataset-the new data for self-driving agents and mapping tasks,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 3284–3290.
- [98] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick, “Microsoft coco: Common objects in context,” in *Computer Vision – ECCV 2014*, David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, Eds., Cham, 2014, pp. 740–755, Springer International Publishing.

CURRICULUM VITAE

NAME: Nada Baili

ADDRESS: Computer Science & Engineering Department
Speed School of Engineering
University of Louisville
Louisville, KY 40292

EDUCATION:

M.S., Computer Science
December 2020
University of Louisville, *Louisville, Kentucky*

B.Eng., Polytechnic Engineering
June 2019

Tunisia Polytechnic School, *Tunis, Tunisia*

RESEARCH AND WORK EXPERIENCE:

1. Graduate Research Assistant, **University of Louisville**, August 2023 - May 2024
2. Applied Scientist Intern, **Amazon**, May 2023 - August 2023
3. Graduate Research Assistant, **University of Louisville**, August 2022 - May 2023
4. Applied Scientist Intern, **Amazon**, May 2022 - August 2022
5. Graduate Research Assistant, **University of Louisville**, February 2019 - May 2022
6. Software Engineer, **EURA NOVA**, June 2018 - August 2018

PUBLICATIONS:

1. Baili, N. and Frigui, H. "ADA-ViT: Attention-Guided Data Augmentation for Vision Transformers". 2023 IEEE International Conference on Image Processing (ICIP), 385-389.
2. Baili, N. and Frigui, H. "Identifying Non-Targets in Automatic Target Recognition Systems with Regularized Margin Entropy Loss". arXiv preprint (2023).

3. Baili, N., Moalla, M., Frigui, H. and Kareem, AD. "Multistage approach for automatic target detection and recognition in infrared imagery using deep learning". Journal of Applied Remote Sensing 16 (4), 048505-048505.

HONORS AND AWARDS:

1. CECS Arthur M. Riehl Award, 2022
2. PHI KAPPA PHI Honor Society membership, 2022
3. Third Prize in Best Poster Presentation, CMD-IT/ACM Tapia, 2021
4. CSE Master of Science Award, 2020
5. Study of the U.S. Institutes (SUSI) for student leaders scholarship, 2017
6. Tunisian National Scholarship for Engineering Studies, 2016