UKRAINIAN CATHOLIC UNIVERSITY

BACHELOR THESIS

Research of Data Augmentation Approaches for Enhancing Classification Model Performance

Author:
Bohdan VEY

Supervisor:
Oles DOBOSEVYCH

A thesis submitted in fulfillment of the requirements for the degree of Bachelor of Science

in the

Department of Computer Sciences and Information Technologies
Faculty of Applied Sciences

Lviv 2023

Declaration of Authorship

I, Bohdan VEY, declare that this thesis titled, "Research of Data Augmentation Approaches for Enhancing Classification Model Performance" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: _____

Date: _____

"Learn to say 'no' to the good so you can say 'yes' to the best."

John C. Maxwell

UKRAINIAN CATHOLIC UNIVERSITY

Faculty of Applied Sciences

Bachelor of Science

Research of Data Augmentation Approaches for Enhancing Classification Model Performance

by Bohdan VEY

Abstract

After a significant improvement in the computational powers of modern computers, the models became larger, and their accuracy increased. However, due to a high amount of parameters, modern neural networks also need much bigger datasets for efficient usage. Augmentation partly solves this problem, but the most up-to-date augmentation still doesn't change the image patterns. We propose a new way of augmentation by using inpainting models to change the image's nature. Then we compare model performance by using traditional augmentation and GANAugmentation. The second part of this study will use Test Time Augmentation(TTA) to improve model performance for data which come from another source.

Acknowledgements

I want to express my gratitude to everyone around me during these four years of study. I also want to thank my supervisor Oles DOBOSEVYCH , who always helped me with the idea and answered my questions.

Contents

List of Figures

List of Tables

List of Abbreviations

TTA **T**est **T**ime **A**ugmentation
NN **N**eural **N**etwork

To my family and friends

Chapter 1

Introduction

Neural networks became a very important part of our everyday lives and led to improvement in many fields: healthcare, autonomous driving and manufacturing, etc. One of the main fields that have grown significantly during the last years, after the development of Alexnet Krizhevsky, Sutskever, and Hinton, 2012 became Computer Vision. Using convolution layers improves the model's performance and allows the use of Computer Vision models for most image processing problems.

Due to the significant spread of computer vision technologies around different fields, the exponential growth of model parameter number problem of data lack became considerable. Despite this, the problem is solved by increasing the dataset size, which needs a lot of money, or by using augmentation. However, the augmentation technique has not undergone significant changes in the last ten years and still uses primitive methods that cannot significantly affect the image. As shown in Figure 1.1, Canny edge detection shows similar results for each image even after providing different augmentation methods.

FIGURE 1.1: Edge detection after different types of augmentation

This could lead to the problem described at Besse et al., 2018, where model performance dropped after the change of the dataset due to the incorrectly learned features and the badly created dataset in which the model focused more on the background than on the main object it had to recognize.

We decided that this problem could be solved by using augmentation, which will significantly change the image by influencing certain regions of the image using various methods: blur, using inpainting models, as well as complete removal of the region.

In Chapter 2, we describe methods to detect regions, which we should modify, and techniques and models used to perform augmentation. We also analyze other methods of augmentation that have become popular nowadays.

In Chapter 3, we will describe our approach to augmenting images and provide an overview of the framework. In Chapter 4, we will provide detailed results of

our experiments, including changes in model performance and the time needed to augment one image. In the last Chapter 5.1, we will summarize all results and decide which methods could potentially be used.

Chapter 2

Related works

2.1 GAN Augmentation: Augmenting Training Data using Generative Adversarial Networks

Using GAN for augmentation has long been considered one of the options to improve the results of models, especially when we talk about fields where it is challenging to increase the dataset size quickly. One of the research made in this area was released by Bowles et al., 2018. In this paper, they synthetically generate data using PGGAN Karras et al., 2018 to artificially increase the dataset size. This

During the experiment, they train different models for medical image segmentation. The first task was to train a GAN model that would be able to generate medical images, which will be difficult to distinguish from actual photos taken on expensive equipment. After that, a new image segmentation model was trained on a combination of real-world data and images that were generated on a pre-trained GAN model.

In this paper, they performed a large number of experiments with different initial data. So they wanted to determine the optimal number of images needed for training the GAN model, as well as find the relationship between the number of generated images in our training dataset and the model's results on real data.

The results of an experiment show that in case, the size of the training dataset, which was used for both to train GAN model and the segmentation model is small enough to get optimal results; we need only 50% of additional data; while in the case out dataset size increases significantly we could double initial dataset size.

However, the fact that we need to generate fewer photos for the smaller size of the initial dataset shows that GAN images are not ideal. There is a slight data shift between synthetic and natural images, even though GANs were trained to decrease this difference. So, in our work, we will try to adopt a number of modified samples and the percentage of corrupted images.

2.2 Model explainability evaluation

2.2.1 Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization

Along with the improvement of results and the exponential growth of the number of model parameters, modern neural networks have faced the problem of needing more explanations for their results. While machine learning approaches like decision trees or linear regression could correctly explain why they show these results, the most decision of deep neural networks for a long time remained unclear even to those who developed the model. One possible solution to this problem was Class

Activation Map(CAM). Using CAM could highlight parts of the image that are the most important for predicting a specific class. One of the best and oldest methods to calculate CAM results is Grad-CAMSelvaraju et al., 2019, which uses gradients to improve the explanation results.

Grad CAM calculates scores by using the last hidden layer of the model. The first step is to calculate class feature weights by using gradients of the feature map for each class respectively. Then it calculates the score for a specific region by using the weighted average for feature values.

$$L^c_{\text{Grad-CAM}} = ReLU \underbrace{\left(\sum_k \alpha^c_k A^k \right)}_{\text{linear combination}}$$

The ReLU function is used; without it, localization maps sometimes highlight more than just the desired class and perform worse at localization.

During the last few years, many new techniques have arisen(Draelos and Carin, 2021 Chattopadhyay et al., 2017 Fu et al., 2020), but most of them don't appear to use a new method but rather to improve gradient-based approaches given by authors of GradCAM.

For example, HIResCAM Draelos and Carin, 2021 provides a technique to calculate CAM without averaging over all layers but instead calculating weights for a specific region. This approach shows better results for IoU Score with human-labelled data and looks more focused on a classified object.

Methods based on other techniques to calculate important regions show much worse results. Recent research Adebayo et al., 2020 shows that their results are less dependent on a value of a model neuron. In figure 2.1, the authors decide to randomize the last few layers of a model and compare its outcomes depending on the number of corrupted layers for different approaches. The results show that most techniques not based on a gradient method don't affect layer randomization. Moreover, their results are often similar to edge detectors, which shows that their outcome is more affected by image than by model parameters.

Our Grad CAM usage focuses on its ability to determine the most crucial region. We will use this algorithm to detect areas where our model could change without affecting objects in an image.

2.2.2 ROAR and ROAD explainability evaluation

One of the important parts that we decided to consider in our paper is explainability. Understanding how the model works can help us develop augmentation that could automatically adjust to the model to help it train better in a given data space. A combination of different explainability technologies can show us weaknesses in different parts of model training. For example, the concentration of GAN on the wrong object may indicate an error in the dataset, namely an imbalanced dataset. But most of the recent techniques in model explainability still need human intervention. The technique proposed by Mohseni, Block, and Ragan, 2020 offers to calculate model explainability by using human-grounded local explanations as ground truth. This approach can accurately determine whether the model focuses on the correct regions and significantly reduce the probability of an error in the dataset. However, it

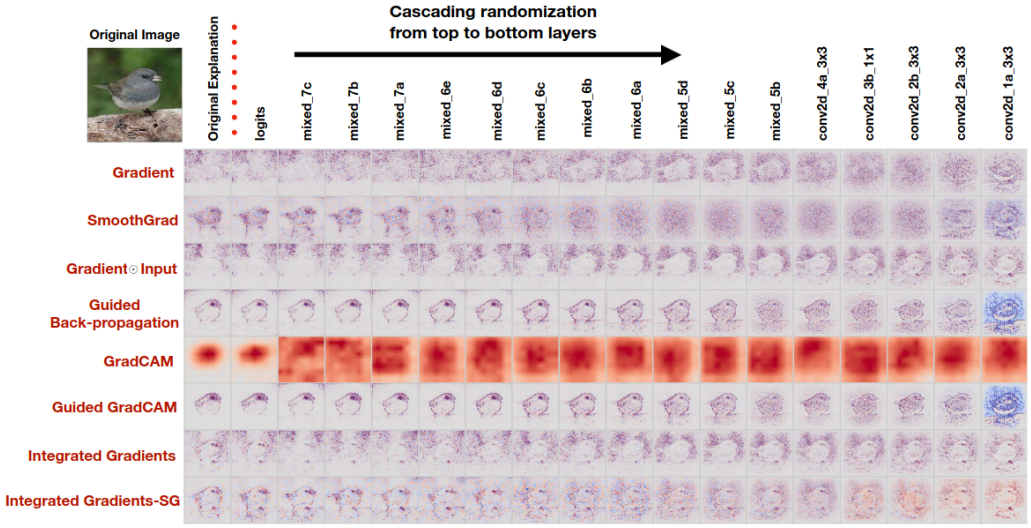


FIGURE 2.1: Change of activation for different approaches depend on layer randomization

also requires much more human resources because it requires additional pixel-wise analysis of each photo.

The second approach that could be used to evaluate dataset correctness was designed by Hooker et al., 2019. The idea presented in this paper was built around calculating the change of confidence after corrupting some per cent of the image. However, the first generation of this approach requires continual retraining of the model, as it uses removing of specific regions, which leads to data shift and to incorrect work of the model. The modification of the algorithm was presented by Rong et al., 2022 and doesn't require retraining of the model. Their research shows that the blurring image technique proposed by the author doesn't affect image patterns, and the model, which was trained on uncorrupted data, could still be used on new images.

The corruption technique given by an author could be used as one of the new augmentation techniques. By blurring the region, which was decided by a model, as important, we could focus its attention on the other patterns and decrease overfitting on train data.

2.3 High-Resolution Image Synthesis with Latent Diffusion Models

Image synthesis is another field which got huge improvements nowadays. GAN models have occupied a large part of this field in recent years and are still one of the most popular solutions in image generation. However, using of diffusion models became famous after releasing of the paper developed by Rombach et al., 2022a. The approach given by authors got massive success in many fields, including text-to-image, super-resolution, and image inpainting.

The idea of diffusion models is quite different from the method proposed by GAN. In fact, diffusion is a probabilistic model designed to learn the distribution of data space by gradually denoising images affected by normal noise.

$$L_{dm} = \mathbb{E}_{x,e\ \mathcal{N}(0,1),t}\left[\|\epsilon - \epsilon_\theta(x_t, t)\|_2^2\right]$$

The model trained in this way learns to generate images from a certain distribution, receiving as input a set of random data, which the model considers to be noisy initial images.

One of the main advantages of diffusion models, which has been helpful in my work, is the ability to paint certain regions of the image without harming the regions that need to be left intact. The inpainting technology used in stable diffusion is one of the most modern and trained on real-world data, making using this model possible in my work 2.2.

FIGURE 2.2: Example of work for stable diffusion inpainting algorithm

2.4 Test-Time augmentation

2.4.1 AugNet: Dynamic Test-Time Augmentation via Differentiable Functions

Another technique that is used during inference to improve model performance is called Test Time Augmentation(TTA). This method proposes to use different types of augmentation in the original image and then take the result as the average or maximum value proposed by a model. TTA could improve your model performance by 2% or 3%. However, this method still does not use in most of the state-of-the-art solutions, as usage of TTA could lead to significant growth in computation, as you need to run your model not only on one image but on an augmented version. The solution to this problem was presented by Enomoto, Busto, and Eda, 2023. The authors of this article suggested using neural networks to determine the best method of image augmentation. They trained a network that determined the best parameters for image augmentation. Augmentation parameters should be chosen in such a way that the main model thinks that the input image comes from the same dataset on which the model was trained. As most of the augmentation techniques, like rotation, have a continuous magnitude parameter from 0 to 360 degrees, each parameter,

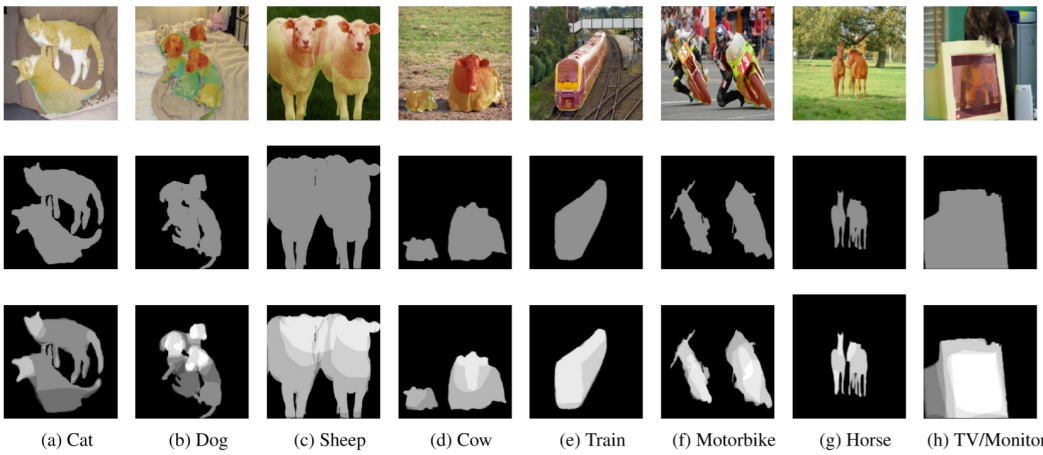
which could be used for augmentation, will first be discretized, so each augmentation will have a finite number of possible parameters, and the model could now be trained on this parameters.

The result shows that adding preprocessing to your model by finding the best augmentation could significantly improve your model performance in case your model was trained on data which came from another source and doesn't affect your model if your data source doesn't change.

2.5 Dataset

2.5.1 Pvoc Attenttion Dataset

This dataset Mohseni, Block, and Ragan, 2020 was created in 2020 and is a reduced version of the Pvoc dataset. It consists of 20 classes, among which there are popular and easily recognizable objects: boat, train, and person. The main advantage of this dataset is that there are 20 photos in each class, together with the human-attention maps2.3 marked by at least 3 people. This map shows the extent to which each pixel is related to a given category, and unlike a normal segmentation map, the values for each class can take different values, not just 0 or 1. This dataset gives us about 400 photos with clear boundaries of the main objects in the scene and can be used to train our augmentations.



(a) Cat (b) Dog (c) Sheep (d) Cow (e) Train (f) Motorbike (g) Horse (h) TV/Monitor

FIGURE 2.3: Example of human-attention map given in the Pvoc Attention Dataset

2.5.2 Gender and Scene recognition dataset

Another dataset Gao et al., 2022 that I decided to use in my work was presented together with the visual explanation framework. This dataset consists of two parts. Each part has a minimum of 1000 photos with two different classes. The task on each dataset is to recognize one of two classes. However, the advantage of this dataset is the attention map, which indicates how much each pixel affects the person asked to say what is depicted in the picture. Unlike the previous dataset, the values in this can be both positive and negative, where negative will mean that this region, on the contrary, tells the user that another class is depicted in the picture. This distribution allows for a more accurate examination of the model. Still, in my case, I will use

the map data to identify regions whose variables can potentially affect the correct outcome of the model, making these regions forbidden to change.

Chapter 3

Method

In our study, we used different types of augmentation to compare the results of the models: TTA, augmentation using stable diffusion, as well, as conventional methods of augmentation made using the kornia.

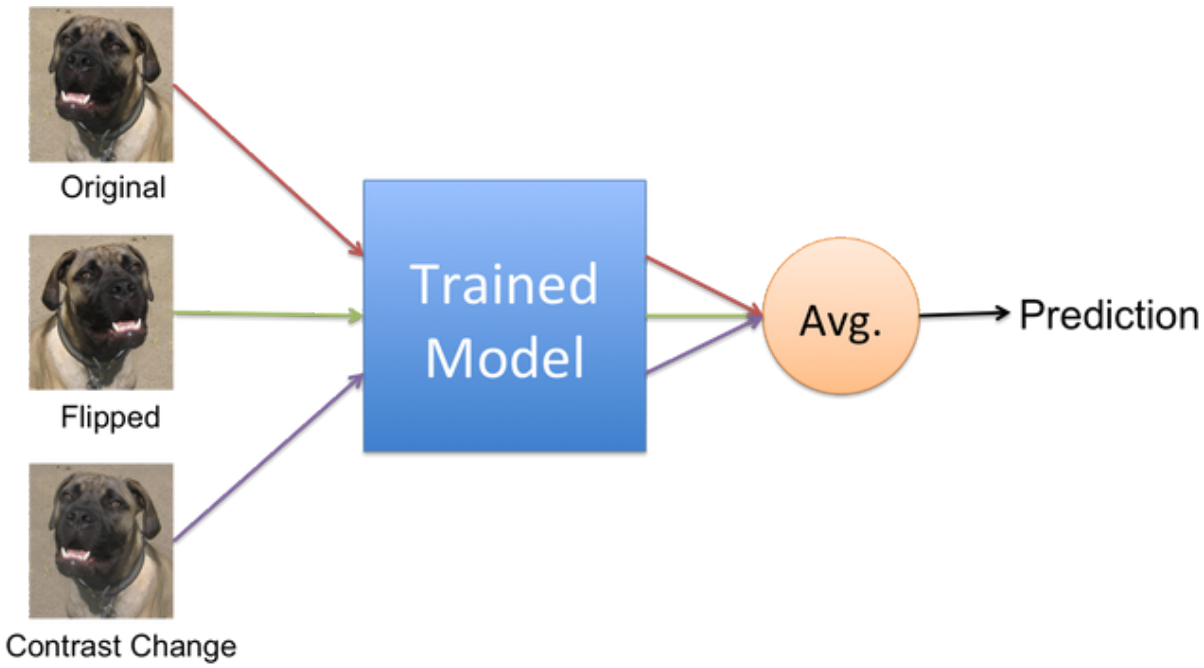


FIGURE 3.1: The example of TTA work

During our experiment, two models: mobilenet Howard et al., 2017, efficientnet Tan and Le, 2020,, were trained on special explainability datasets described in the related works section. These datasets were used in connection with the fact that they have ground-truth predictions regarding the location of the object, which can be used in the case of training on stable diffusion augmentation.

All models were trained with the same parameters, and the only difference between each method is augmentation used to improve model performance.

3.1 Test Time Augmentation

The essence of this experiment is to check the effect of TTA Shanmugam et al., 2021 on the results of the model. This experiment will be conducted in two stages, the first will consist of trained models without any augmentation during training, and the second will include augmentation using the kornia method. During testing, we will use the framework written by https://github.com/qubvel/ttach to performe

TTA3.1, which allows us to parallelize the model during testing. After that, we will record the results of the model, as well as the time spent on training and testing, and determine whether this method is useful in real projects.

3.2 Kornia Augmentation

This method consists in using common and popular methods of augmentation. During the experiment, we will use different augmentation methods, both old and new ones written by us personally and published in kornia E. Riba and Bradski, 2020, to improve the results of the model. During testing, we will determine whether different augmentations affect the results of the models, as well as determine the total time required to train the model with and without augmentation

3.2.1 Random Rain Augmentation

FIGURE 3.2: Example of Random Rain Augmentation

Random Rain Augmentation is created to simulate rain conditions on an image. To do this, we generate raindrops randomly split across the whole picture using our algorithm modified to run on the GPU 3.3. To improve the speed performance of the augmentation each drop consists of the same size and shape and can easily be generated in parallel across the entire image. Performence tests showed that using the GPU, the modified algorithm was able to improve the speed from 5.89 seconds to 0.81 compared to the method written on albumentation Buslaev et al., 2020 in case of running on 1000 examples.

3.2.2 Random Snow Augmentation

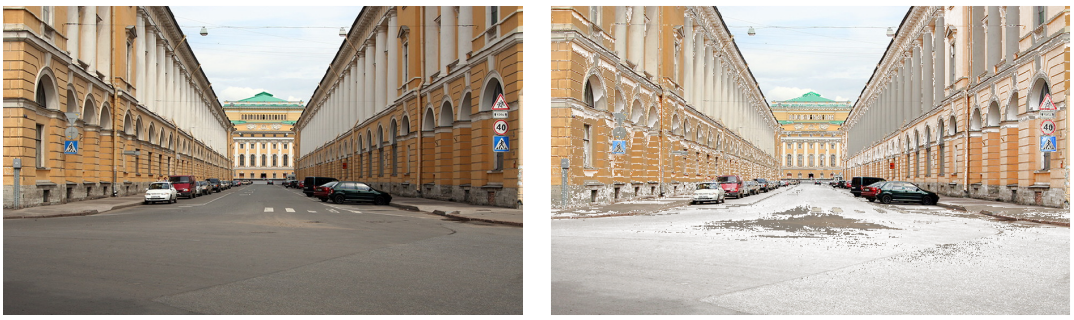


FIGURE 3.3: Example of Random Snow Augmentation

Random Snow Augmentation simulates weather conditions, particularly generating snow across the image. In our case, this augmentation finds the darkest regions in the image and multiplies their value, simulating the fall of white snow over certain areas. The idea behind this method is that in most outdoor photos, the darkest regions are often responsible for the ground or road, so generating a white blanket over these regions results in the simulation of snow.

3.3 Stable diffusion augmentation

The following method consists of corrupting some percent of the image and then trying to restore the original photo by using a stable diffusion inpainting algorithm. In our case, we use two different pre-trained models from Rombach et al., 2022b to achieve more considerable variability. We also decide not to retrain our diffusion model on data from the dataset. This step is taken because our algorithm is calculated on a small number of images, and we will often not have enough data to properly train our stable-diffusion model, as it requires a large number of images for training. Also, with the help of this method, we were able to avoid adding an additional step to the model, which would consist in retraining a heavy diffusion model, which is challenging to do on small servers with a small amount of video memory and requires a lot of time to train a new model.

We also decide to use two different methods to determine which regions we want to corrupt. In the first case, we will use the GradCAM algorithm to detect areas which are the least important for our model performance. Corrupting this region should allow our model to focus even more on the main image. The model trained in such a way will not focus on the object's background, as our background will be changed after each iteration.

The second approach is to delete the most valuable part of the image. By corrupting these parts, we could achieve better variability for the model. The model trained in such a way will have more training data of the main object, which helps the model to distinguish different things better.

For the correct algorithm process, we decided to remove 30-40% of the image in the case of augmentation of regions with a lower impact on the model and from 10 to 20% if the regions have a more significant impact on the model performance. The difference in the augmentation area is necessary because the main object usually occupies a smaller area. However, we cannot remove the entire object because then the image will be distorted too much, and the diffusion model will not be able to reproduce the original image as much as possible.

In order for the model to initially be able to identify objects better, we decided to give it the opportunity to learn the initial patterns by training it without using augmentation for the first 20 epochs, and the model itself at the beginning of training was already pre-trained on Imagenet Deng et al., 2009 data.

However, the data created in this way could contain data shifts on the main part of the image and will train on unrealistic photos, which could lead to a drop in performance. On the other hand, increasing the dataset in such a way could lead to a better model generalization.

The next problem that could arise is not the ability of the diffusion model to determine what was depicted in the picture from the beginning and also inability to enter a prompt by the user, which is necessary for the correct operation of the model. To avoid this problem, we decided to create an automatic prompt for each

image that looks like: "A photo of {class name}". In this case, to generate prompt, we should save each class name inside the Dataset to create the correct input.

Chapter 4

Experiments

4.1 PVOC dataset

4.1.1 Setup

In the initial experiment, we decide to use the PVOC dataset Mohseni, Block, and Ragan, 2020. The dataset was created for object detection and localization, so it often has more than one class in the image. In order for the model to work with a more significant number of objects, we decided to use Sigmoid 4.1.1 as the final activation function.

$$h_\theta(x) = \frac{1}{1 + e^{-}\theta^T x}$$

To decide that prediction was correct, we will use a threshold value equal to 0.5.

As we could have more than 1 class on the image, we could not use the original GradCAM to calculate region importance. In order to modify our algorithm, we decided to use a weighted average to calculate the importance value of the region for each class. In order to speed up the algorithm and improve the accuracy of identifying essential regions, a threshold size of 0.5 was also applied, and the class whose confidence score was less than 0.5 did not participate in the calculation of the final score 4.1.1. The final answer will be normalized so the biggest value will be equal to one and lowest to zero.

$$F(x) = \sum_{x=0}^{19} \begin{cases} 0, & \text{if } h_\theta(x) < 0.5, \\ h_\theta(x) * GradCAM(x), & \text{if } h_\theta(x) > 0.5 \end{cases}$$

Since we also decided that our final activation function would be a sigmoid rather than a softmax, using the cross-entropy loss could lead to an error. Instead, we decided to think of the task as defining 20 independent classes, where each output value would indicate the presence or absence of a particular object in the image. Thus we will use Binary Cross Entropy Loss. However, since our loss may be more significant than necessary for training, we will reduce our learning rate by ten times compared to the value we will use during the training of the other models.

Another important point for training is a correctly chosen prompt. For the problem of multilabel classification, using the prompt that we use in our model will be incorrect, because we do not know which class to specify. For this, it was decided to make a hierarchy system and in the case of finding several objects in the photo, the one with the highest confidence in our initial model is selected.

Due to this use of the diffusion model, there may be a problem with the disappearance of certain classes. This means that there should be a specific object behind the ground truth data in our photo, but after using augmentation, this object will

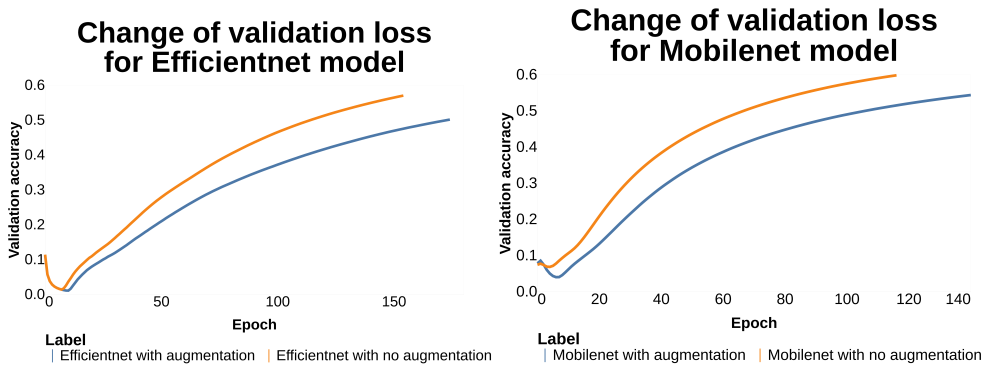
disappear. In our case, it was decided to ignore this problem since it is impossible to determine which class will disappear in advance.

(A) Correct image (B) Transformed unimportant part (C) Transformed important part

FIGURE 4.1: Example of transformation using stable-diffusion in case the correct class is bottle

In order to simulate the limited size of the dataset, it was decided to reduce the size of the training dataset to 500 photos. In contrast, the size of the validation dataset remained unchanged. To increase the speed of the dataset, augmented photos were generated before the start of the main training session. To do this, a small model was first trained for 20 epochs, and this model was used as an example to identify essential regions. For each image, one sub-image of each type was produced: the image with a changed important region and the image with a change of background. An example of generated images can be seen in the Figure 4.1. If an essential subpart of the image is affected, a slight change in the bottle texture can be seen, while if less important regions are affected, text appears near the bottle, which does not affect the main object.

4.1.2 Results



(A) Validation accuracy for efficientnet pvoc dataset (B) Validation accuracy for mobilenet pvoc dataset

FIGURE 4.2: Validation accuracy for Pvoc dataset

The training results showed that augmentation in the case of training on the PVOC dataset worsens the model performance. These results could also be because some classes were disappearing during testing 4.3. In this Example, we can see how after using the augmentation, the ship in the background has completely disappeared, and only the tree class remains. However, in training, the labels will say

that a tree and a boat, which have already been erased from the image, are both in the photo. This caused problems with labels during training, and our model could not work correctly. From the Figure 4.2, we can see that for the Efficientnet, the accuracy drop is more than 7%, while the model needs more time for training, which may indicate the inefficiency of this method for the PVOC dataset. The Mobilenet model also showed a significant decrease in accuracy when using augmentation. However, when we check whether the class with the highest probability is in the photo, the result increased from 0.40 to 0.44 when using augmentation on the Efficientnet model. These results show that this method can lead to inaccuracies in the case of using multi-label classification due to label shifts during work with augmentation. However, the results of a single-class classification can be better.

(A) Correct image (B) Transformed important part

FIGURE 4.3: Example of the disappearance of the boat after using the diffusion augmentation

4.2 Scene Classification dataset

4.2.1 Setup

The SceneClassification dataset Gao et al., 2022 was developed to train a model to recognise whether a photo was taken in an urban position or its natural environment. The original dataset contains 1750 photos of each type for training. However, we decided to reduce this number to 500 in order to simulate the situation when we have a limited number of photos.

The final layer of the model was an output of size 2, where each value told how likely it was to be a given class. Since only 1 class can be depicted in this dataset on one photo, we decided to use Softmax 4.2.1 as the final activation function. This solution will guarantee that the sum of all classes at the output will be exactly one and that we will be able to use GradCAM to detect important regions.

$$\sigma(y_i) = \left(\frac{e^{y_i}}{\sum_j e^{y_j}} \right) j = 1, ..., n$$

We decide to use this dataset as it is also an excellent example of using our traditional augmentation method. To test this, during our work, we also wrote several methods for weather augmentation, including Random Rain Augmentation and Random Snow Augmentation 3.2. This dataset can be considered an excellent example to test the performance of our augmentation methods since all the photos in it are taken outdoors 4.4, so weather augmentation may be necessary under these conditions.

Unfortunately, due to the poorly constructed dataset, we cannot test if our weather augmentations affect the model output when we experience a change in weather. However, we decided that these augmentation methods are still necessary in the case of training on this dataset. After all, most of the photos in it contain sunny weather, so we will have a date shift when working on actual data because the weather is not always sunny.

Despite the problems mentioned above, we will still be able to check the operation of our diffusion augmentation in combination with weather augmentation, which is also important in our research.

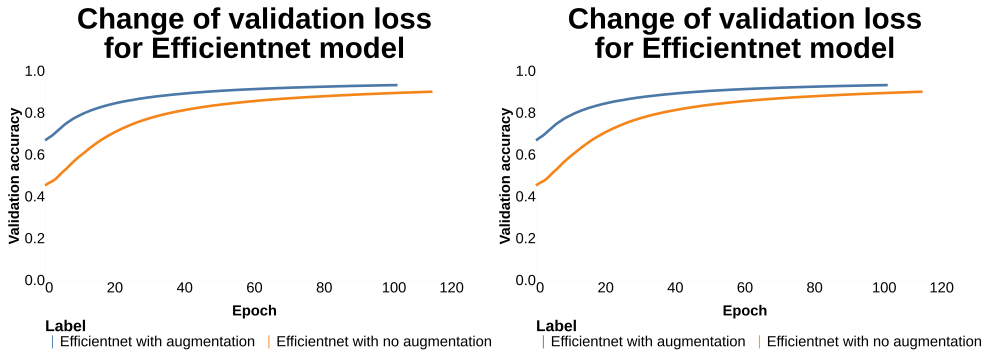
(A) Correct image (B) Rainy image (C) Snowy image

FIGURE 4.4: Example of the weather augmentation on Scene Dataset

To improve the work of diffusion augmentation, we choose to change the method of writing prompts. Instead of writing "A photo of {class name}", we decided to clarify what exactly is shown in the picture. In this dataset, each photo in the title contains a name of an object depicted in the picture. For example, an image of a bridge will be named bridge-{photo_id}. By knowing this fact, we can create a prompt more precisely regarding a link to a photo by specifying that the picture will feature a specific object. This modification will allow our algorithm to work more accurately, and the generated images will have better quality.

However, regardless of the prompt change, we will still train our model as a binary classifier that needs to tell whether an image is urban or natural.

We decided to reduce the size of the training dataset. In order to do this, we also used the photo's name because this is how we could divide our dataset into equal parts. In this case, we decided that each specific object in the dataset should occur at most 40 times, given the fact that each class contains several different objects. For example, the urban class could include a bridge or a house. We decided that this number of photos would be enough to train our model minimally and that our model would only have the opportunity to train on a small amount of data. The final size of the dataset was about 1000 photos, where each class contained exactly 500 images.



(A) Validation accuracy for Efficientnet scene dataset

(B) Validation accuracy for Mobilenet scene dataset

FIGURE 4.5: Validation accuracy for scene dataset

4.2.2 Results

In this case, our results showed that using synthetic data to increase the dataset's size can reduce the time needed to train the optimal model and improve the model's accuracy on the validation dataset that did not undergo changes during training.

From the training results 4.5, the accuracy of the model with augmentation increased by 2% for each model compared to the training results without diffusion augmentation. Thus, in the case of training the Mobilenet model, the accuracy increased from 92.5% to 94.4%, which may indicate the effectiveness of using this method under such parameters.

Using this method, we could also reduce the number of epochs needed to train the model by 20, compared to a model that did not use synthetic images for training. For this, we used a callback called EarlyStopping. This method is used to reduce the time and resources spent on training and avoid overfitting. In our case, during training, we set the automatic shutdown of the model in case the loss for the validation dataset does not decrease during the last 10 epochs. So we saw that our new model needs 10% less time to achieve optimal results on the new dataset.

On the other hand, the generation of new images itself also takes a long time. It took approximately 8 hours to generate the 2 synthetic pictures from each of the 3500 photos in the initial training dataset before we reduced its size. While training 20 epochs using the Nvidia A100 video card took only 1 hour, which makes it quite challenging to say precisely which of the methods is faster.

4.3 Gender Classification dataset

4.3.1 Setup

The last dataset we decided to work with is the gender classification dataset Gao et al., 2022. The main task in this dataset is to recognize whether a man or a woman is in the photo. Despite its initial simplicity, this dataset has many pitfalls that can complicate model training. So, in this dataset, there are many photos where a person's face is not clearly visible due to a mask or other reasons 4.6. This problem significantly complicates the work of the model, which worsens its results. In addition, the different distributions of photos: some photos are taken during sports competitions that require the person to wear specific clothes, while others may be a simple selfie taken with a phone camera, which can also affect the model result.

FIGURE 4.6: Example of images in gender classification dataset

Another reason why we decided to use this particular dataset is because of an explicit data shift when working with the diffusion model. The stable diffusion model, without pertaining, works poorly on human photos and almost never can clearly generate a face 4.7. This drawback was used to analyze the model's performance if the generated data differs from real-life data.

We also decided to reduce the size of the dataset to 600 photos, consisting of 300 photos of men and 300 photos of women. Due to the fact that all the photos in the same class are similar to each other, unlike the photos from the Scene Classification Dataset, this number of images is more than enough to train the model, which should show good results.

All other parameters during training remained unchanged, and no other methods were used for training this model, except for those prescribed in the Framework 3.

(A) Correct image (B) Transformed important part

FIGURE 4.7: Example of bad augmentation for gender dataset

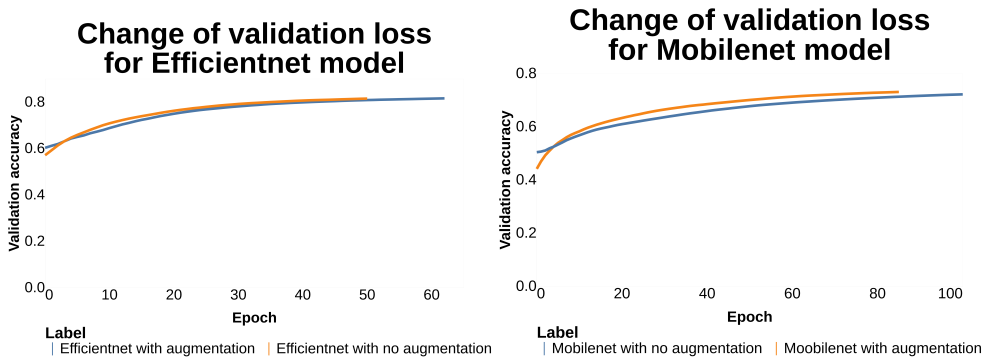
4.3.2 Results

The results showed that despite the significant data shift, which occurred due to mixing the original dataset of 1200 generated photos using Stable diffusion, the accuracy of the model increased by an average of 1% for different models4.8.

Such improvement cannot guarantee the effectiveness of this method. So it can occur both due to minor changes in the DataLoader and due to the effect of augmentation.

In addition, we can again see a significant reduction in the number of epochs required to train the model using the new model. Thus, when using Mobilenet augmentation, the model needed 15% less time to achieve the best results than the model trained without augmentation.

In general, the performance of the diffusion model in this dataset is not significant, and its use under these conditions requires much more effort than the potential benefit that this algorithm can bring to improve accuracy.



(A) Validation accuracy for Efficientnet scene dataset

(B) Validation accuracy for Mobilenet scene dataset

FIGURE 4.8: Validation accuracy for scene dataset

Chapter 5

Conclusion and Future work

5.1 Conclusion

Model	Pvoc Classification Dataset	Scene Classification Dataset	Gender Classification Dataset
Mobilenet with no augmentation	0.5990	0.925	0.7218
Efficientnet with no augmentation	0.5708	0.930	0.81
Mobilenet with augmentation	0.5428	0.944	0.7310
Efficientnet with augmentation	0.5021	0.935	0.82

TABLE 5.1: Results of model performance

In summary, synthetic data generation technology can increase the model's efficiency. Still, for this, many conditions are necessary, which are required for the correct operation of the algorithm.

An example can be that among the three datasets on which we tested this technique, we were able to clearly improve the results of the model in only one of them.

The general augmentation results on different datasets can be seen in **??**.

In fact, the potential problem we can face while generating new data is the disappearance of the labels, which we could see while working with the Pvoc Dataset. As well as the poor operation of the Stable Diffusion model, which generates incorrect and unrealistic data while working with the Gender Classification Dataset.

Despite the drawbacks, the model's results can be significantly improved if our newly created dataset contains well-generated data. Thus, in the case of working with the SceneClassificationDataset, augmentation showed a significant impact on the performance of the model and was able to reduce the probability of an error by 25%.

In conclusion, we can say that in the case of a large number of real images, this technology will not be able to improve the results. However, provided that the size of our input dataset is quite small, the use of generative augmentation can significantly improve the model results.

In addition, unlike the method of full image generation, the inpainting technique does not require user intervention, because all generated data can be automatically labeled.

5.2 Future work

One of the possible developments of this method could be the adaptation of its work on segmentation models. Using correctly generated data, we will be able to expand the total size of the segmentation dataset, without user intervention.

Thus knowing the exact location of all objects, we can develop our inpainting method so that we do not have a change of ground truth labs after the augmentation work.

This method requires additional research and significant improvement of the modern inpainting method because, during work, it must leave all objects in their places while entirely changing the appearance of the photo itself.

Bibliography

Adebayo, Julius et al. (2020). *Sanity Checks for Saliency Maps*. arXiv: 1810.03292 [cs.CV].

Besse, Philippe et al. (Nov. 2018). "Can Everyday AI be Ethical? Machine Learning Algorithm Fairness (english version)". In: DOI: 10.13140/RG.2.2.22973.31207.

Bowles, Christopher et al. (2018). *GAN Augmentation: Augmenting Training Data using Generative Adversarial Networks*. arXiv: 1810.10863 [cs.CV].

Buslaev, Alexander et al. (2020). "Albumentations: Fast and Flexible Image Augmentations". In: *Information* 11.2. ISSN: 2078-2489. DOI: 10.3390/info11020125. URL: https://www.mdpi.com/2078-2489/11/2/125.

Chattopadhyay, Aditya et al. (2017). "Grad-CAM++: Generalized Gradient-based Visual Explanations for Deep Convolutional Networks". In: *arXiv preprint arXiv:1710.11063*.

Deng, Jia et al. (2009). "Imagenet: A large-scale hierarchical image database". In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee, pp. 248–255.

Draelos, Rachel Lea and Lawrence Carin (2021). *Use HiResCAM instead of Grad-CAM for faithful explanations of convolutional neural networks*. arXiv: 2011.08891 [eess.IV].

E. Riba D. Mishkin, D. Ponsa E. Rublee and G. Bradski (2020). "Kornia: an Open Source Differentiable Computer Vision Library for PyTorch". In: *Winter Conference on Applications of Computer Vision*. URL: https://arxiv.org/pdf/1910.02190.pdf.

Enomoto, Shohei, Monikka Roslianna Busto, and Takeharu Eda (2023). *Dynamic Test-Time Augmentation via Differentiable Functions*. arXiv: 2212.04681 [cs.CV].

Fu, Ruigang et al. (2020). *Axiom-based Grad-CAM: Towards Accurate Visualization and Explanation of CNNs*. arXiv: 2008.02312 [cs.CV].

Gao, Yuyang et al. (2022). "RES: A Robust Framework for Guiding Visual Explanation". In: *Proceedings of the 28th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM.

Hooker, Sara et al. (2019). *A Benchmark for Interpretability Methods in Deep Neural Networks*. arXiv: 1806.10758 [cs.LG].

Howard, Andrew G. et al. (2017). *MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications*. arXiv: 1704.04861 [cs.CV].

Karras, Tero et al. (2018). *Progressive Growing of GANs for Improved Quality, Stability, and Variation*. arXiv: 1710.10196 [cs.NE].

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton (2012). "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems 25*. Ed. by F. Pereira et al. Curran Associates, Inc., pp. 1097–1105. URL: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf.

Mohseni, Sina, Jeremy E. Block, and Eric D. Ragan (2020). *A Human-Grounded Evaluation Benchmark for Local Explanations of Machine Learning*. arXiv: 1801.05075 [cs.HC].

Rombach, Robin et al. (2022a). *High-Resolution Image Synthesis with Latent Diffusion Models*. arXiv: 2112.10752 [cs.CV].

Rombach, Robin et al. (2022b). "High-Resolution Image Synthesis With Latent Diffusion Models". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10684–10695.

Rong, Yao et al. (2022). *A Consistent and Efficient Evaluation Strategy for Attribution Methods*. arXiv: 2202.00449 [cs.CV].

Selvaraju, Ramprasaath R. et al. (2019). "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization". In: *International Journal of Computer Vision* 128.2, pp. 336–359. DOI: 10.1007/s11263-019-01228-7. URL: https://doi.org/10.1007%2Fs11263-019-01228-7.

Shanmugam, Divya et al. (2021). *Better Aggregation in Test-Time Augmentation*. arXiv: 2011.11156 [cs.CV].

Tan, Mingxing and Quoc V. Le (2020). *EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks*. arXiv: 1905.11946 [cs.LG].