Investigating the effects of data augmentation on the accuracy of flower classification

Project group 89 | Julius Fechner (2619892), Nourhan Gooshan (2623534), Robin Herlan (2594618), Yasmina el Yakoubi (2623760)

Abstract

Overfitting is a common problem in machine learning, where a model fits too closely to the training set, which weakens its generalization capabilities. For image classification specifically, techniques exist that augment existing training data to increase the training dataset size and allow the model to learn more diverse features. We investigate three such augmentation techniques: flipping, cropping and random erasing, and evaluate how they affect the accuracy of an image classification agent. We find that while they do increase the model accuracy, comparisons between the techniques yield inconclusive results as to which is better.

1 Introduction

The ability to process and understand images is a crucial component in understanding the world. To most humans, this ability is as easy as breathing. However, it has proven a challenge to give machines this same ability. Nevertheless, recent breakthroughs in machine learning techniques, particularly neural networks, has enabled computers to mimic our ability to process and comprehend images. An important component in machine learning methodology is data preparation, to avoid potential pitfalls such as overfitting. For image classification, an important method of data preparation is that of data augmentation. Existing images are modified in some way to create new images for the dataset. These modified images help reduce overfitting by causing a model to focus on different features of the image. In this paper, we investigate the following research question: to what extent do different data augmentation techniques affect the accuracy of an image classification model? The data augmentation techniques that will be evaluated are flipping, cropping and random erasing, and the data set that will be used is provided by the Kaggle competition ‘Petals to the metal - Flower classification on TPUs’.

1.1 Hypothesis

We believe that all data augmentation techniques should increase the model’s classification accuracy. Individually, flipping will likely increase the accuracy the least, as this transformation alters the dataset the least. Flipping should be followed by cropping, which has been proven to be a more effective augmentation technique [3]. Finally, random erasing should provide the highest increase in accuracy, as it modifies the internal sections of the image and may force the model to learn the most diverse set of features for effective classification [8].

1.2 Approach

We investigate the effect different data augmentation techniques have on a model classifying flower images. As such, we must first decide on a model to use as the baseline for our comparisons. Since the focus is on investigating data preparation techniques, we utilize transfer learning and forgo the creation of a new image classification model. We choose our model through literature analysis. After choosing a model, we evaluate the possible data augmentation techniques through literature analysis and choose 3 to compare. Before beginning the evaluation of the augmentation techniques we choose a set of optimal model and data augmentation hyperparameters through hyperparameter optimisation. Once these have been selected, we run the model four times: once without any data augmentation techniques, and once for each augmentation technique. We then discuss the results and their implications, as well as possibilities for future research.

2 Data analysis

The data we used for our project comes from the kaggle competition 'Petals to the Metal - Flower Classification on TPU'. The data provided by this competition contains images of various flowers. Figure 7, which can be seen in the appendix, shows examples of the pictures provided by the Kaggle competition. As can be noticed, not all pictures provide a clear view of the flower to classify. The elements not relevant to the flower in question are considered noise. Examples of noise include shadows, other irrelevant flowers in a picture, or backgrounds that distract from the target. The presence of noise in images is one reason why data augmentation techniques are effective. Changing the images forces the model to focus on different features, which may improve the robustness of the model.

The dataset is pre-split into a training set, a validation set and a test set. The training set contains 16465 images, the validation set has 3712 images and there are 7382 images in the test set. The data is distributed over 104 different classes; figure 2 shows the distribution of the data over these 104 categories.

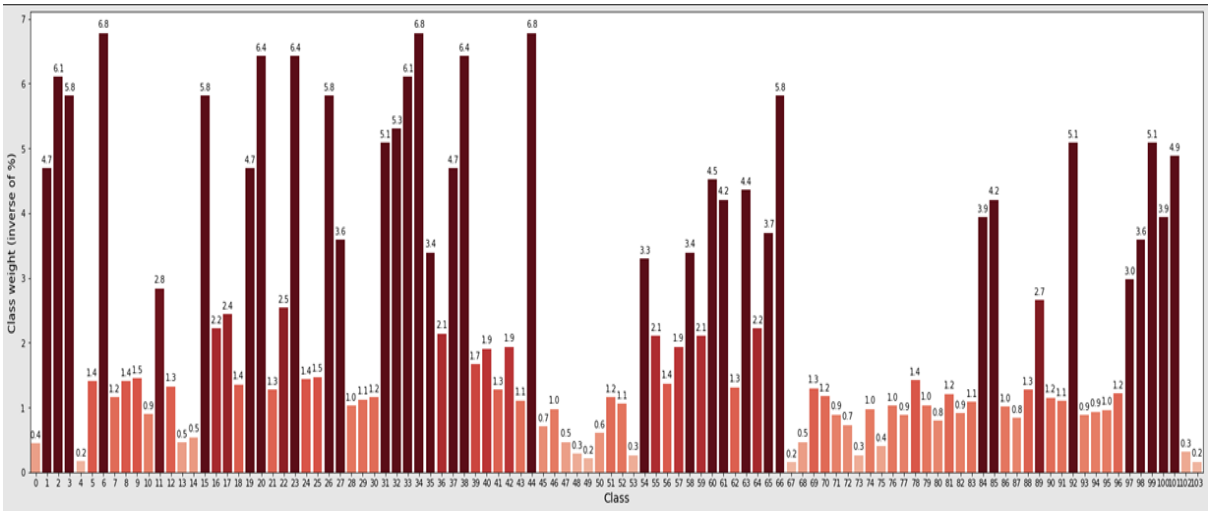

Figure 1: Data distribution over the 104 classes

3 Methodology

3.1 Choosing the model

In the domain of image classification, models are generally based on convolutional neural networks (CNNs). CNNs have the advantage of being able to perform classification tasks straight from images, which makes them one of the most popular neutral networks.[4] They consist of an input layer, several hidden layers and an output layer. The hidden layers are usually made up of convolution, pooling and fully connected layers.[14] With these layers, a CNN can take as input an image and assign to it various values such as biases and weights using the filter and matrix vector multiplication. The CNN transforms the image into a feature map through the filter, which gets passed through the remaining layers and at the output layer it is summed with the bias to form the feature output.[4] CNNs have proven to be a very effective architecture for classifying images, and most research in the field of image classification is based on CNNs. As such, we will be choosing a CNN based model.

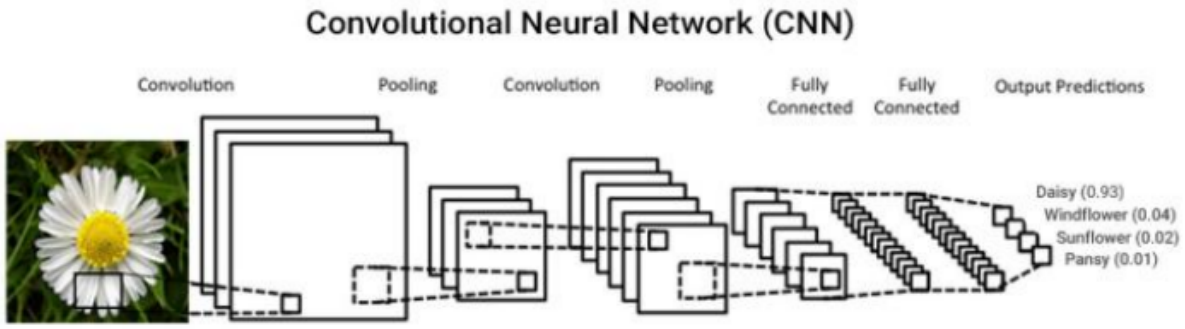

Fig 2: The typical architecture of a CNN. The square represents an example of a filter/convolutional kernel. This CNN has convolution, pooling and fully-connected layers. The output shows the different predictions that the network made based on the input image.[9]

For our research we will employ transfer learning with pre-trained models. These models were trained on a large dataset to solve a similar problem in image classification tasks. An advantage of using pre-trained models is the reduced computational cost and faster learning process since there is no need to start training from scratch. In transfer learning new layers are added to the pre-trained model. These new layers are then used for training and together with the other layers are used to solve a classification task. [13]

Amongst the most popular pre-trained models are LeNet, GoogLeNet and VGGNet. These models are commonly used in research into image classification. [9] After investigating various CNN models, we decided to use the Inception-V3 model. Research shows that the Inception-V3 model has a higher accuracy rate compared to other models such as Xception, VGG16 and OverFeat when used in flower classification based on datasets of various sizes. Inception-V3 was shown to perform classification with a higher accuracy rate for rank-1 accuracy and rank-5 accuracy.[9] This model is the third generation of Google's inception models. The Inception-V3 model is pre-trained on the Imagenet dataset, which contains over 14 million images. Compared to other models, Inception-V3 comes with many advantages,

one of which is that it is designed to perform well under strict constraints on memory and computational budget.[7] We chose Inception-V3 instead of VGG-16 or any other model because in 2014 Inception-V3 won the ImageNet Large Scale Visual Recognition Challenge (ILSVRC).[11] VGG-16 came second in this competition. One reason that could have contributed to Inception-V3's victory is that at the time, it only contained around 5 million parameters which is 12x smaller than its predecessor AlexNet.[7]

Figures 3.1 and 3.2 show the graph of the accuracy of InceptionV3 and VGG16 on the training and validation set of the flower dataset. This was made with no data augmentation. As can be seen, InceptionV3 has a ~70% accuracy when compared to the VGG16 model which performed much worse at ~20% accuracy. This further reinforces our choice in the InceptionV3 model.

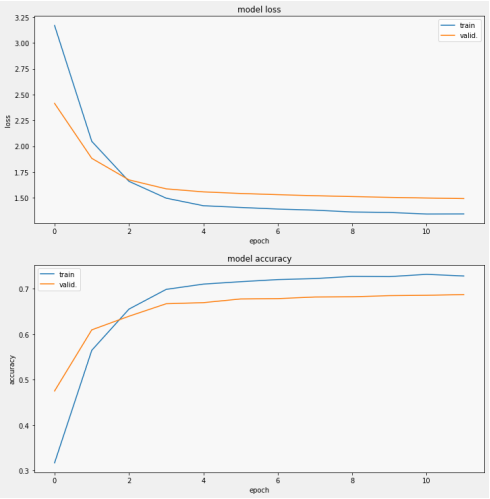

Fig 3.1: InceptionV3 graph showing the model accuracy and loss on the flower dataset using no augmentation

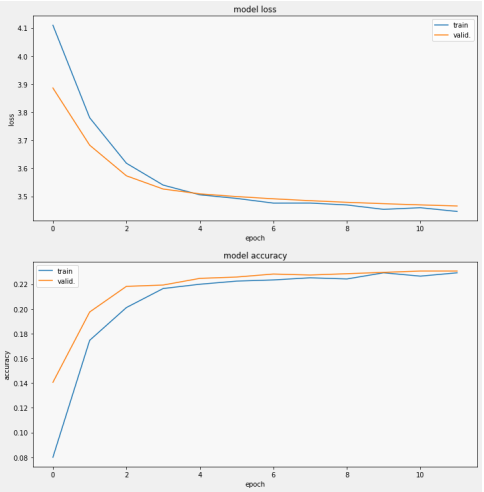

Fig 3.2: VGG16 graph showing the model accuracy and loss on the flower dataset using no augmentation

3.2 Choosing the data augmentation techniques

Data augmentation techniques can be separated into two broad categories: naive augmentation techniques that utilize simple image transformation procedures and advanced, 'black-box' techniques that use deep neural networks to create new images [10, 3]. This paper will evaluate three naive data augmentation techniques and compare their performances. Naive data augmentation techniques can be divided further into two categories: geometric or positional modification and color modification. Previous research into these techniques have found that geometric transformations generally yield better results than color modifications [3], and thus the focus of this paper will be on geometric transformations. The list of possible geometric transformations are: flipping, rotating, cropping, shearing, elastic distortions and random erasing. The techniques investigated in this paper will be flipping, cropping and random erasing. These techniques offer a good overview of the different possible transformations and their relative difficulties, with flipping being the simplest transformation, followed by cropping and then random erasing. Additionally, other research has found that these techniques offer a good range of effectiveness at increasing a models classification

accuracy [3, 8], with flipping leading to the smallest increases and random erasing offering the largest increase in model accuracy.

Flipping

Flipping is a simple data augmentation technique where the output image is the mirror-reversal of the input image along the horizontal axis. [3].

Figure 4: example of image flipping

Cropping

Cropping an image entails the removal of an outer area of the image, specified by giving the height H and width W of a new image centered around some random point in the old image. The new image, now a smaller size than the original, must be rescaled to the original size [3]. This can be done either by adding padding or resampling the image. For this paper, images have been resized with nearest sampling.

Figure 5: example of image cropping with resample based resizing

Random erasing

Random erasing is a technique that randomly removes a section of an image by replacing the pixel RGB values with 0,0,0. We remove a rectangle of size S, where S is chosen from two variables S_{low} and S_{high}. The aspect ratio of this new rectangle is a random variable R, initialized between two values R_{low} and R_{high}. Once these values are set, a random point is chosen within the image until the created rectangle fits within the image. This rectangle is then erased from the original image [8]. The pseudocode for this technique can be found in appendix A.

Figure 6: example of random erasure

3.3 Choosing the hyperparameters

There are a number of hyperparameters that must be chosen, and these can be split into two separate classes: model hyperparameters and data augmentation specific hyperparameters. The model hyperparameters are parameters that must be set for the chosen model, and these parameters will remain constant for every model that is trained and tested. The augmentation specific hyperparameters are values that must be configured for each data augmentation technique, and these will be different for each technique depending on the parameters they require. Bayesian optimization will be used to search the hyperparameter space. Bayesian optimization offers similar results to random search while on average obtaining these results with fewer trials [12]. This is because Bayesian optimization builds a probability model of the objective function and then uses this to make an informed selection of the best hyperparameters. This is beneficial as the overall hyperparameter space is large and the training of the model may take a long time, depending on the optimal number of epochs.

Model Hyperparameters

The model hyperparameters will consist of the optimal number of epochs for which the model will train, the starting learning rate, as well as various parameters for the learning rate schedule. A learning rate schedule is an algorithm that dynamically changes the learning rate of a model depending on current values; in this case, the learning rate will vary with the number of epochs. The algorithm for the learning rate scheduler can be found in appendix A.

Hyperparameter	Value
Epochs	60
Learning rate	0.001
Minimum learning rate	0.0000001
Maximum learning rate	0.005
Ramp-up epochs	13

Table 1: model hyperparameters.

Data augmentation hyperparameters

All data augmentation techniques have a variable *P* that denotes the chance of an image not being augmented. For example, *P = 0.4* denotes that an image has a 40% chance of not being augmented; inversely, an image has a 60% chance of being augmented

Flipping

For flipping, the only relevant hyperparameter is the probability of (not) being flipped.

Hyperparameter	Value
P	0.4

Table 2: flipping hyperparameters

Cropping

Along with the probability of an image being cropped, cropping has two other parameters *H* and *W*. These two parameters are the height and width coefficients, respectively, that are multiplied with the respective dimension of the original image to obtain the size of the new image. The image is then cropped around a random central point with the new dimensions to obtain a new image.

Hyperparameter	**Value**
P	0.5
H	0.6
W	0.7

Table 3: cropping hyperparameters

Random erasing

The hyperparameters for random erasing include the probability, the low and high values of the rectangle to be removed (S_l and S_h respectively), and the high value of aspect ratio of the new rectangle (R_h). The low value for the aspect ratio $R_H / 3$.

Hyperparameter	**Value**
P	0.3
S_l	0.1
S_h	0.2
R_h	0.4

Table 4: random erasure hyperparameters

4 Results and Discussion

The final results will be calculated and interpreted using the F_1-score. The F_1-score is a statistical analysis which calculates the accuracy of a given test. It is calculated using the harmonic mean of precision and recall. The F_1-score offers advantages over traditional accuracy because accuracy only looks at all correctly identified cases whereas the F_1-score also puts an emphasis on the incorrectly identified cases. This is especially important in situations where there is an imbalance in class distribution. The exact formula is shown below. The best score possible is a 1, which means that both the precision and recall have a perfect score and the worst score is a 0, which means that either the precision or recall is 0. For the case of multi-class accuracy, the final F_1-score is the aggregate of each class' F_1-score.

$$F_1 = 2 * \frac{precision*recall}{precision + recall} \quad precision = \frac{TP}{TP + FP} \quad recall = \frac{TP}{TP + FN}$$

Looking at the results, it becomes clear that there is a major limitation. Since the test set was unlabeled, we were not able to compute any other metrics that could have helped in analyzing the results. More on this in the limitation section.

Table 5 shows the results of the different data augmentation techniques tested on the testing set. For the baseline model where no data augmentation was applied, the resulting F1-score was 0.80821. This score is higher than the first baseline which was made in Fig 3.1 without any hyperparameter tuning. However, Fig 3.1 did not use the F1-score, but rather accuracy as a metric. This shows that the hyperparameter tuning had a significant impact on the models accuracy.

	F1-score
Baseline/No augmentation	0.80821
Flipping	0.81168
Cropping	0.80759
Random Erasing	0.81323

Table 5: Results of the model tested on the testing set

In our hypothesis it was stated that according to research conducted we predicted the Random Erasing data augmentation to perform best from the three augmentation techniques. While this is the case in these results, the difference between Random Erasing and Flipping is minimal. Additionally, it was predicted that Cropping would perform better than Flipping since it changes the geometry of the image in a more extreme way and hence putting more emphasis on the features of the image. This was not the case. In fact, Cropping performed worse than all augmentation techniques and the baseline. However, the difference is again minimal.

The research question that was stated in the introduction of this paper was: to what extent do different data augmentation techniques affect the accuracy of an image classification model? Based on the limited results that we were able to collect, it is unclear whether data augmentation techniques increase the accuracy of image classification models. Existing research showed that this was the case for these augmentation techniques.[1,2,3,4] However, our study was not conclusive in that regard.
Since the change in F1-score for the different techniques is so minimal, a statistical significance test could help to determine whether this is by chance. As such, one major component that could be investigated in future works could be; What is the statistical significance of the different data augmentation techniques?
Because there was only a limited amount of time to conduct this research, we were not able to determine whether combining different augmentation techniques would yield an increase in accuracy. It would make sense if this was the case since previous research showed that changing the geometry of an image improves accuracy. Perhaps the issue with our chosen approach was that the geometry wasn't changed enough. As such, future works could investigate whether there is an optimal set of data augmentation techniques that perform better than their separate parts.

5 Limitations

One limitation that we came across during our research is that since the provided test set was unlabeled, we could not produce more results other than the F1-score. This is because the F1-score is computed when the test set predictions are submitted to the kaggle competition. As such, we never had access to the labels of the test set. Being able to compute more results for the test set could have been advantageous by showing us different metrics such as standard deviation, cohen's kappa, statistical significance, etc…. This could have given us some insight into what could be improved in future research.

Another limitation that we encountered was that in order to tune and train our model we had to use TPU's. However, there is only a limited quota of TPU time available each week and we discovered that oftentimes the quota was reached quite quickly. If we had more time to fine tune our model and data augmentation parameters, we could have tried more data augmentation techniques and improved on the ones that we tested.

6 References

[1] J. Shijie, W. Ping, J. Peiyi and H. Siping, "Research on data augmentation for image classification based on convolution neural networks," 2017 Chinese Automation Congress (CAC), Jinan, China, 2017, pp. 4165-4170, doi: 10.1109/CAC.2017.8243510.

[2] Lei, Cheng & Hu, Benlin & Wang, Dong & Zhang, Shu & Chen, Zhenyu. (2019). A Preliminary Study on Data Augmentation of Deep Learning for Image Classification. Internetware '19: Proceedings of the 11th Asia-Pacific Symposium on Internetware. 1-6. doi: 10.1145/3361242.3361259.

[3] Shorten, C., Khoshgoftaar, T.M. A survey on Image Data Augmentation for Deep Learning. J Big Data 6, 60 (2019). doi: 10.1186/s40537-019-0197-0

[4] Gu, Shanqing; Pednekar, Manisha; and Slater, Robert (2019) "Improve Image Classification Using Data Augmentation and Neural
Networks," SMU Data Science Review: Vol. 2 : No. 2 , Article 1.
[5] R. Yamashita, M. Nishio, RKG Do, K. Togashi, Convolutional neural networks: an overview and application in radiology, Insights Imaging 9 (2018) 611–629, doi: 10.1007/s13244-018-0639-9.

[6] P. Wang, X. Zhang, Y. Hao, A method combining CNN and ELM for feature extraction and classification of SAR image 6134610 J. Sens. 2019 (2019), doi: 10.1155/2019/6134610.

[7] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 2016, pp. 2818-2826, doi: 10.1109/CVPR.2016.308.

[8] Zhong, Z., Zheng, L., Kang, G., Li, S. and Yang, Y. (2020) “Random Erasing Data Augmentation”, *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(07), pp. 13001-13008. doi: 10.1609/aaai.v34i07.7000.

[9] I. Gogul and V. S. Kumar, "Flower species recognition system using convolution neural networks and transfer learning," 2017 Fourth International Conference on Signal Processing, Communication and Networking (ICSCN), Chennai, India, 2017, pp. 1-6, doi: 10.1109/ICSCN.2017.8085675.

[10] A. Mikołajczyk and M. Grochowski, "Data augmentation for improving deep learning in image classification problem," 2018 International Interdisciplinary PhD Workshop (IIPhDW), Świnoujście, Poland, 2018, pp. 117-122, doi: 10.1109/IIPHDW.2018.8388338.

[11] Russakovsky, Olga, et al. "Imagenet large scale visual recognition challenge." International journal of computer vision 115.3 (2015): 211-252. doi: 10.1007/s11263-015-0816-y

[12] Victoria, A.H., Maragatham, G. Automatic tuning of hyperparameters using Bayesian optimization. Evolving Systems 12, 217–223 (2021). doi: 10.1007/s12530-020-09345-2

[13] Gao, Yuqing, and Khalid M. Mosalam. "Deep transfer learning for image‐based structural damage recognition." Computer‐Aided Civil and Infrastructure Engineering 33.9 (2018): 748-768. doi: 10.1111/mice.12363
[14]Hussain, M., Bird, J. J., & Faria, D. R. (2018, September). A study on cnn transfer learning for image classification. In UK Workshop on computational Intelligence (pp. 191-202). Springer, Cham.

Appendix A: Algorithms

Random erasing

Pseudo code for the random erasure of sections of an image as presented in [9]

Algorithm : Random Erasing Procedure

Input : Input image I;
Image size W and H;
Area of image S;
Erasing probability p;
Erasing area ratio range s_l and s_h;
Erasing aspect ratio range r_1 and r_2.

Output: Erased image I^*.

Initialization: $p_1 \leftarrow$ Rand (0, 1).

if $p_1 \geq p$ **then**
  $I^* \leftarrow I$;
  return I^*.
else
  while *True* **do**
    $S_e \leftarrow$ Rand $(s_l, s_h) \times S$;
    $r_e \leftarrow$ Rand (r_1, r_2);
    $H_e \leftarrow \sqrt{S_e \times r_e}$, $W_e \leftarrow \sqrt{\frac{S_e}{r_e}}$;
    $x_e \leftarrow$ Rand $(0, W)$, $y_e \leftarrow$ Rand $(0, H)$;
    if $x_e + W_e \leq W$ *and* $y_e + H_e \leq H$ **then**
      $I_e \leftarrow (x_e, y_e, x_e + W_e, y_e + H_e)$;
      $I(I_e) \leftarrow$ Rand (0, 255);
      $I^* \leftarrow I$;
      return I^*.
    end
  end
end

Learning rate scheduler

Algorithm : Learning rate schedule

Input : Current epoch E;
Current learning rate LR
Minimum learning rate $MinLR$
Maximum learning rate $MaxLR$
Number of ramp-up epochs R
Exponential decay ED

Output: Updated learning rate LR^*

if $E \leq R$ **then**
  $LR^* \leftarrow (MaxLR - LR) \div (R \times E + LR)$;
else
  $LR^* \leftarrow (MaxLR - MinLR) \times E^{E-R} + MinLR$;
end
Return: LR^*;

Figure 7 : Examples of the images present in the data set