Computer Vision
Assignment 02

Data Augmentation on Cassava

Melchor Lafuente
Iker Jauregui

Contents

1 Introduction

The aim of this project is to analyze the impact of different data augmentation techniques have in the performance.

2 Experimental Setup

2.1 Dataset Splitting

The dataset was divided into training, validation, and test sets using the predefined splits stated at the metadata file *cassava_split.csv*, but the imbalance problem was corrected by random downsampling the CMD class for Train and Val sets (Test set was kept unmodified).

Set	**CBB samples**	**CBSD samples**	**CGM samples**	**CMD samples**	**Healthy samples**
Train	761	1532	1670	2000	1804
Val	163	328	358	400	387
Test	163	328	358	1974	387

Table 1: Dataset splits used during the experiments.

2.2 Callbacks and LR Scheduler

During the experiments, two main callbacks were used: **EarlyStopping** and **ModelCheckpoint**. For batch size selection, an initial **BatchSizeFinder** was used and it suggested a value of 2048 samples. That size allocated almost the whole memory of the GPU (50GB) so we finally used a batch size of 1024. Finally, instead of using a LearningRateFinder, we decided to employ a learning rate scheduler. We decided to use the **ReduceLROnPlateau** scheduler, as it could lead the models to obtain the best possible validation losses.

2.3 Model Architecture and Evaluation Metric

As the main purpose of this study is to analyze the obtained results after applying different data augmentation techniques, we decided to fix the model architecture to a **ResNet18**.

For the evaluation metric, we selected the **F1 Score (macro)**, as we wanted to measure the model's performance over the unbalanced test set.

3 Baseline

As baseline for this project, we took the best model from the previous assignment (*Assignment 01: Transfer Learning with Cassava*). It is a ResNet18 with pretrained weights (ResNet18_Weights.IMAGENET1K_V1) and it was fine-tunned with gradual unfreezing (per blocks) technique. It achieved a F1 (macro) score of **0.6423**.

4 Experiments

For the next experiments, the baseline's model architecture and training method was fixed while different data augmentation techniques were applied over the training data. On the validation and test splits only ImageNet transforms were used.

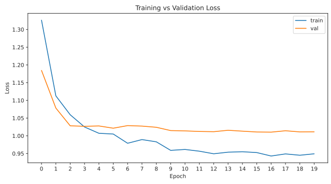
4.1 Experiment 1: Geometric Augmentations

In this first experiment we tested the performance of geometric transformations (Table 2). If we look to the loss curves (Figure 1) we can see how the model still overfits to the trainig data. However, it achieved a F1 score (macro) of **0.6893** (0.047 points more respect to the baseline model).

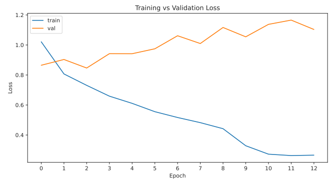
Transformation	**Parameters**	**Probability**
Resize	`size=256, interpolation=BILINEAR`	100%
RandomHorizontalFlip	—	10%
RandomVerticalFlip	—	10%
RandomResizedCrop	`size=256, scale=(0.5, 1.0), ratio=(0.75, 1.33)`	20%
RandomRotation	`degrees=(-90, 90)`	20%
RandomAffine	`degrees=(0, 0), translate=(0.25, 0.25)`	20%
CenterCrop	`size=256`	100%
ToTensor	—	100%
Normalize	`mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225]`	100%

Table 2: Configuration of the geometric transformations used in the experiment 1.

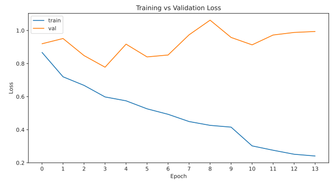
Results:



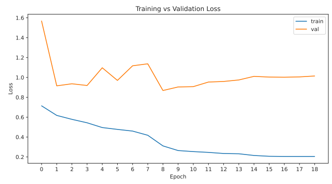
(a) Frozen until block #1



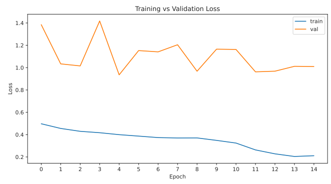
(b) Frozen until block #2



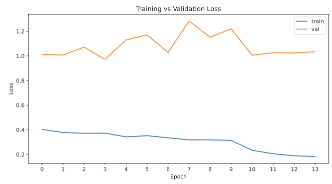
(c) Frozen until block #3



(d) Frozen until block #4



(e) Frozen until block #5



(f) Completely unfrozen

Figure 1: Loss curves for the model fine-tunned with gradual unfreezing using geometric data augmentation.

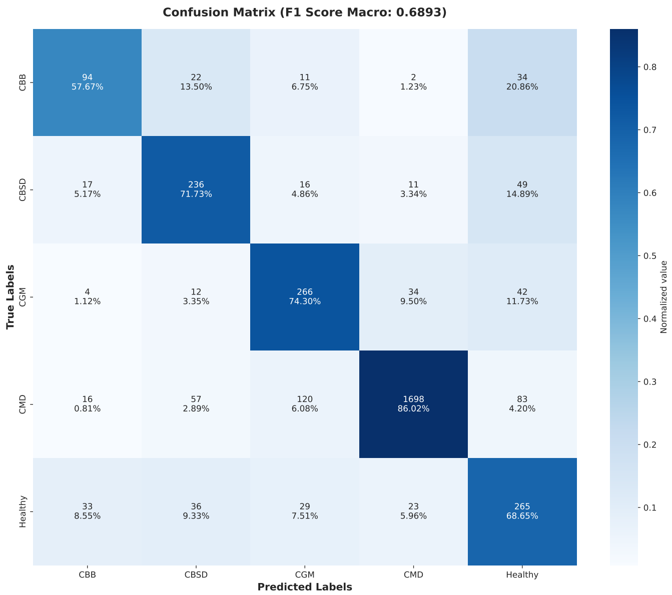
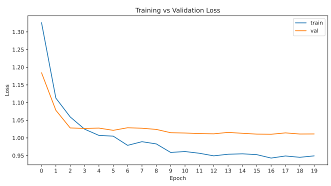


Figure 2: Confusion matrix over Test set for the fine-tunned with gradual unfreezing using geometric data augmentation.

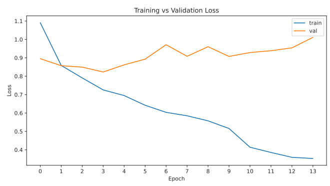
4.2 Experiment 1 bis: Geometric Augmentations with Dropout

As we have seen in the previous experiment, even if we apply data augmentation, the model still overfits. In order to try to solve this problem, the previous experiment was repeated but using dropout regularization technique on each unfrozen block (all probabilities set to 0.2). As we can see in the loss curves (Figure 3) the problem persists. However, looking at the performance over Test set (Figure 4), we can see that the model achieved a F1 score (macro) of **0.6952** (0.0529 points more respect to the baseline model and 0.0059 points more respect to the previous experiment). So, we can say that dropout improved generalization.

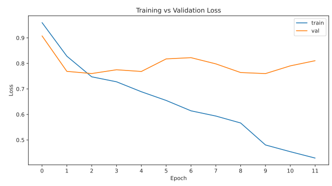
Results:



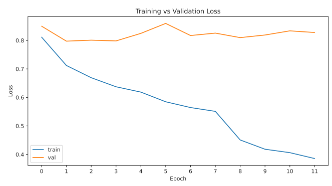
(a) Frozen until block #1



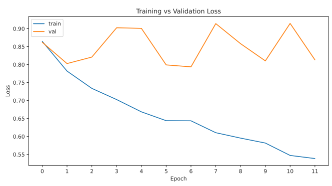
(b) Frozen until block #2



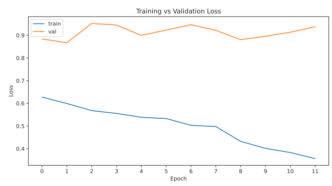
(c) Frozen until block #3



(d) Frozen until block #4



(e) Frozen until block #5



(f) Completely unfrozen

Figure 3: Loss curves for the model fine-tunned with gradual unfreezing using geometric data augmentation and dropout regularization.

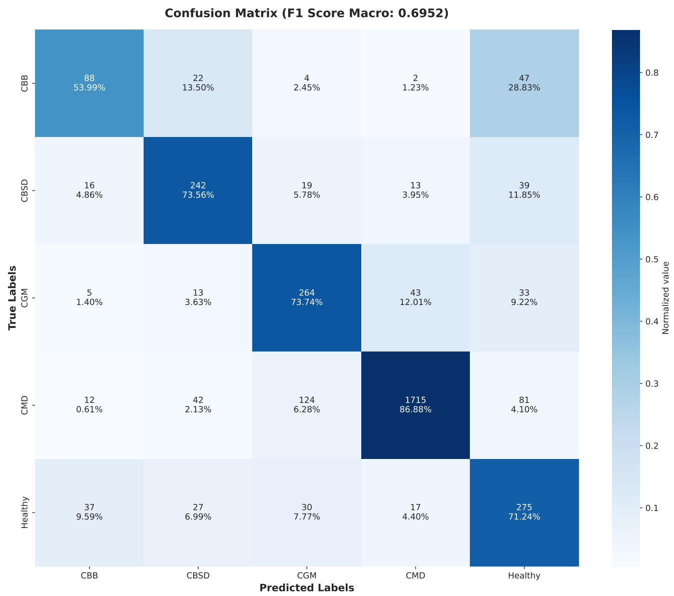


Figure 4: Confusion matrix over Test set for the fine-tunned with gradual unfreezing using geometric data augmentation and dropout regularization.

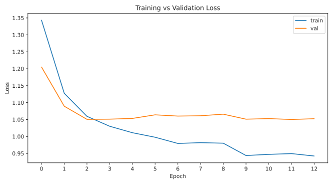
4.3 Experiment 2: Photometric Augmentations with Dropout

In the previous experiment we saw that dropout demonstrated to be useful, so in this experiment we applied the same dropout configuration. However, this time we wanted to analyze how photometric transformations would affect to the model's performance (Table 3). If we look to the loss curves (Figure 5) we can see that again the model overfits to the trainig data. This time it achieved a F1 score (macro) of **0.5858** (0.0565 points less respect to the baseline model), maybe because it overfitted even more than in previous experiments.

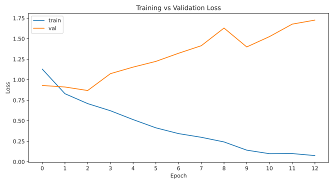
Transformation	**Parameters**	**Probability**
Resize	`size=256, interpolation=BILINEAR`	100%
ColorJitter	`brightness=0.25, contrast=0.25, saturation=0.25, hue=0.025`	50%
GaussianBlur	`kernel_size=(3,3), sigma=(0.1, 2.0)`	10%
CenterCrop	`size=256`	100%
ToTensor	—	100%
GaussianNoise	`mean=0.0, std=0.1`	10%
Normalize	`mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225]`	100%

Table 3: Configuration of the photometric transformations used in the experiment 2.

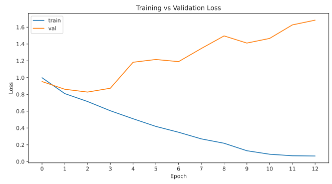
Results:



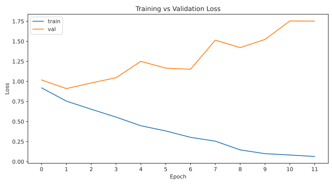
(a) Frozen until block #1



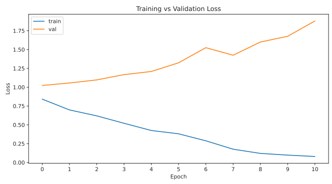
(b) Frozen until block #2



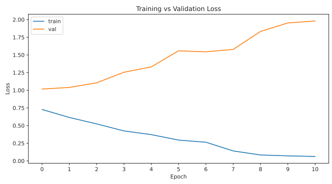
(c) Frozen until block #3



(d) Frozen until block #4



(e) Frozen until block #5



(f) Completely unfrozen

Figure 5: Loss curves for the model fine-tunned with gradual unfreezing using photometric data augmentation and dropout regularization.

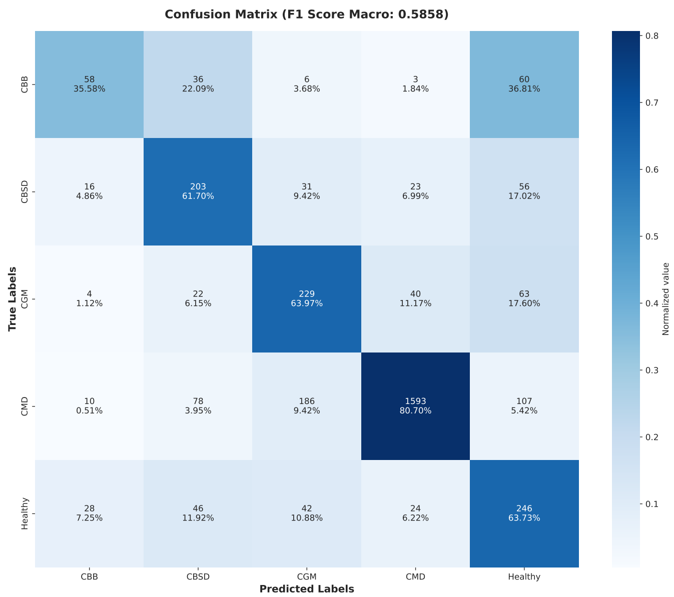


Figure 6: Confusion matrix over Test set for the fine-tunned with gradual unfreezing using photometric data augmentation and dropout regularization.

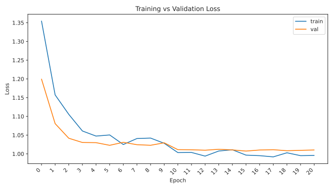
4.4 Experiment 3: Geometric and Photometric Augmentations with Dropout

We have seen that by far the model is overfitting, so in this experiments we apply both geometric and photometric transformations (Table 4) to see if this can help the model to generalize better. Unfortunately, the loss curves show that the model still overfits (Figure 7). However, we notice a slight improvement over the F1 score obtained over the Test set: **0.7017** (0.0594 points more respect to the baseline model).

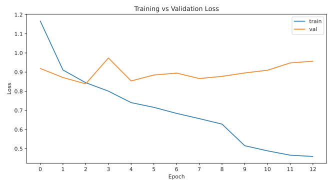
Transformation	**Parameters**	**Probability**
Resize	`size=256, interpolation=BILINEAR`	100%
ColorJitter	`brightness=0.25, contrast=0.25, saturation=0.25, hue=0.025`	50%
GaussianBlur	`kernel_size=(3,3), sigma=(0.1, 2.0)`	10%
RandomHorizontalFlip	—	10%
RandomVerticalFlip	—	10%
RandomResizedCrop	`size=256, scale=(0.5, 1.0), ratio=(0.75, 1.33)`	20%
RandomRotation	`degrees=(-90, 90)`	20%
RandomAffine	`degrees=(0, 0), translate=(0.25, 0.25)`	20%
CenterCrop	`size=256`	100%
ToTensor	—	100%
GaussianNoise	`mean=0.0, std=0.1`	10%
Normalize	`mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225]`	100%

Table 4: Configuration of the geometric and photometric transformations used in the experiment 3.

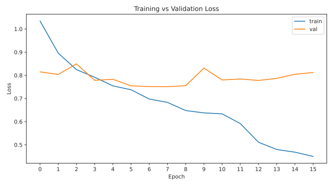
Results:



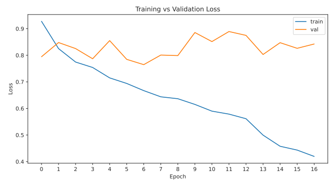
(a) Frozen until block #1



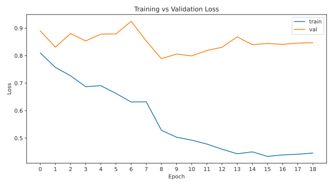
(b) Frozen until block #2



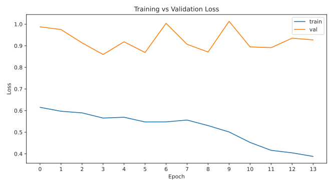
(c) Frozen until block #3



(d) Frozen until block #4



(e) Frozen until block #5



(f) Completely unfrozen

Figure 7: Loss curves for the model fine-tunned with gradual unfreezing using geometric and photometric data augmentation and dropout regularization.

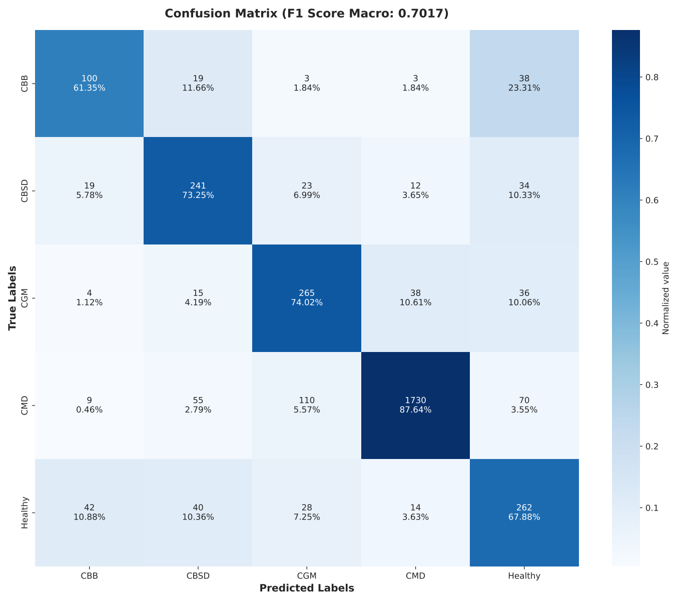


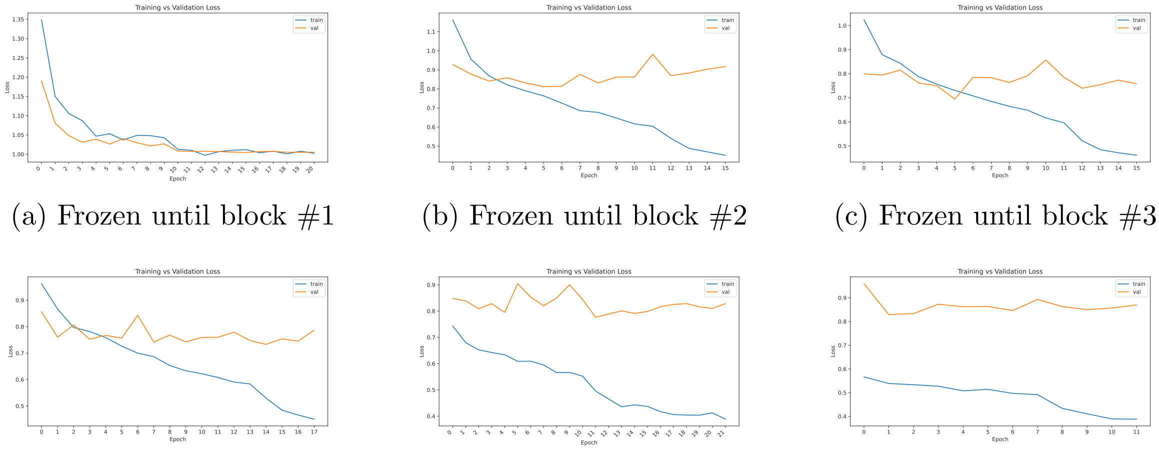
Figure 8: Confusion matrix over Test set for the fine-tunned with gradual unfreezing using geometric and photometric data augmentation and dropout regularization.

4.5 Experiment 3 bis: Previous Experiment Fixed

At this point we noticed that we weren't using exactly the same ImageNet base transforms for the Train split, because we were applying a CenterCrop of size 256x256 (while the pre-trained weights were achieved by using a size of 224x224). So, we fixed the **CenterCrop size to 224x224**. Moreover, Christian Ayala (teacher of the subject) mentioned that there was an implementation error on the gradual unfreezing technique. We fixed it too. So, basically, this experiment's configuration is the same as the previous one, but with the recently mentioned problems solved.

This time the model achieveD a F1 score of **0.7050**, which is a minimal improvement from the previous experiment (0.0033 points more).

Results:



(a) Frozen until block #1 (b) Frozen until block #2 (c) Frozen until block #3

(d) Frozen until block #4 (e) Frozen until block #5 (f) Completely unfrozen

Figure 9: Loss curves for the previous experiment's model (Experiment 3) with the correct implementation of gradual unfreezing technique and a CenterCrop of size 224x244.

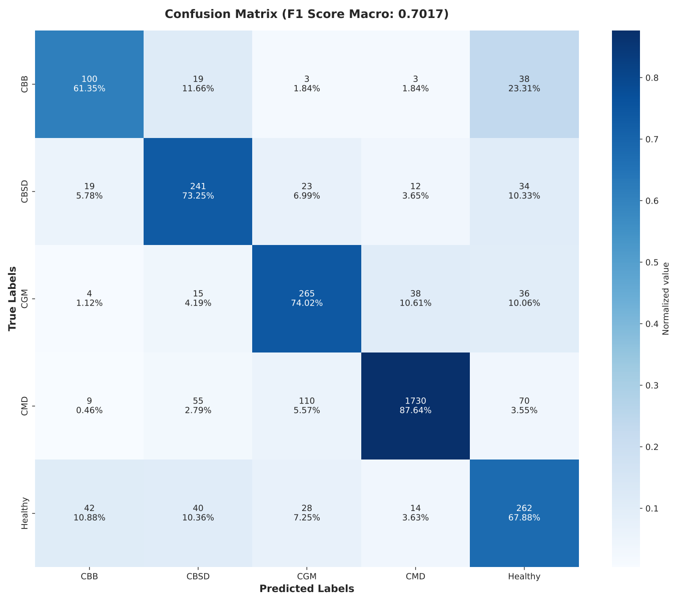


Figure 10: Confusion matrix over Test set for the previous experiment's model (Experiment 3) with the correct implementation of gradual unfreezing technique and a CenterCrop of size 224x244.

5 Test-time Augmentation

After all the experiments were performed, we wanted to apply test-time augmentation (with *ttach* library) in order to see if the best model's performance could be improved just by modifying its evaluation method. The applied transformations and their parameters can be seen in the following table (Table 5):

Transformation	**Parameters**
HorizontalFlip	—
VerticalFlip	—
Rotate90	`angles=[0, 90, 180, 270]`
Scale	`scales=[0.95, 1.0, 1.05]`
Multiply	`factors=[0.95, 1.0, 1.05]`

Table 5: Configuration of the TTA augmentations.

As we can see in the confusion matrix (Figure 11), the use of TTA achieved a F1 score of **0.7261** (0.0211 point more than the best model's performance).

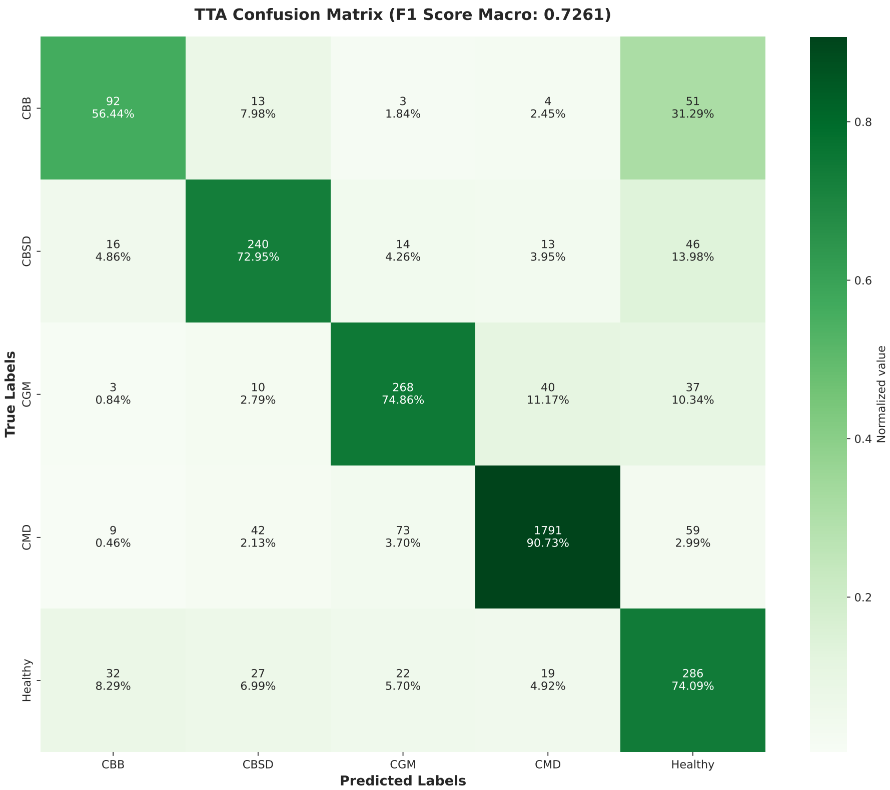


Figure 11: Confusion matrix over Test set after applying TTA.

6 Conclusions

In this study we have seen that data augmentation demonstrated to be really useful for the Cassava disease classification task. The baseline model achieved an F1 score of **0.6423**, which was improved to **0.7050** by applying geometric and photometric augmentations combined with dropout regularization.

We have seen that different augmentation strategies give different results. Geometric transformations (rotations, flips, affine transformations) worked better than photometric ones (color jitter, blur, noise). This is probably because cassava disease symptoms appear as structural patterns and lesions that don't change with geometric transformations, while photometric changes may modify the color variations that are important for disease identification. The combination of both types of augmentations with dropout achieved the best results, which means that using diverse augmentation strategies helps the model to generalize better.

Test-time augmentation improved the best model's performance even more, reaching an F1 score of **0.7261**. This shows that making predictions over augmented versions of test samples can boost the classification accuracy without additional training.

However, we noticed that all models still overfitted during training. This means that data augmentation alone, even combined with dropout, wasn't enough to fully solve the overfitting problem.

7 Future Work

In order to solve the overfitting problem observed in all experiments, we should explore additional regularization techniques such as:

- Implementing advanced data augmentation techniques like mixup or cutmix.
- Experimenting with different dropout rates or applying other regularization techniques.

Finally, we could implement our own test-time augmentation strategy using the same geometric and photometric transformations applied during training. This could give better results than the generic TTA approach we used, as it would be specifically designed for the augmentations the model learned during training.