Deep Learning-Based Transfer Learning for Classification of Cassava Disease

Ademir G. da Costa Junior, Fábio S. da Silva, Ricardo Rios

Universidade do Estado do Amazonas (UEA)

November 20, 2024

Introduction

Introduction

- Cassava is the staple food for more than 800 million people in developing countries.

Introduction

- Cassava is the staple food for more than 800 million people in developing countries.
- In 30 years, world production of this tuber has more than doubled, with Brazil consolidating its position as a giant in the sector.

Introduction

- Cassava is the staple food for more than 800 million people in developing countries.
- In 30 years, world production of this tuber has more than doubled, with Brazil consolidating its position as a giant in the sector.
- Cassava has become a pillar of food security in vast regions of Africa, Asia and Central America, standing out for its resistance and hardiness.

Problem Statement

Problem Statement

- Like other crops that feed the world, cassava faces a constant challenge: pests and diseases that can decimate entire plantations.

Problem Statement

- Like other crops that feed the world, cassava faces a constant challenge: pests and diseases that can decimate entire plantations.
- Artificial Intelligence (AI) is emerging as a powerful tool in the field, revolutionizing the agricultural sector, especially in the early and accurate detection of diseases.

Problem Statement

- Like other crops that feed the world, cassava faces a constant challenge: pests and diseases that can decimate entire plantations.
- Artificial Intelligence (AI) is emerging as a powerful tool in the field, revolutionizing the agricultural sector, especially in the early and accurate detection of diseases.
- Hypothesis: It is possible to train an effective Transfer Learning-based Convolutional Neural Network (CNN) model for this task.

Dataset

Dataset

- Dataset *Cassava Leaf Disease Classification* available on the Kaggle platform.

Dataset

- Dataset *Cassava Leaf Disease Classification* available on the Kaggle platform.
- 21,367 labeled images collected in a survey in Uganda.

Dataset

- Dataset *Cassava Leaf Disease Classification* available on the Kaggle platform.
- 21,367 labeled images collected in a survey in Uganda.
- Annotated by experts from *National Crops Resources Research Institute* (NaCRRI), Makerere University's AI lab.

Dataset

- Dataset *Cassava Leaf Disease Classification* available on the Kaggle platform.
- 21,367 labeled images collected in a survey in Uganda.
- Annotated by experts from *National Crops Resources Research Institute* (NaCRRI), Makerere University's AI lab.
- Divided into five classes, four pathologies and the healthy category.

Class Distribution

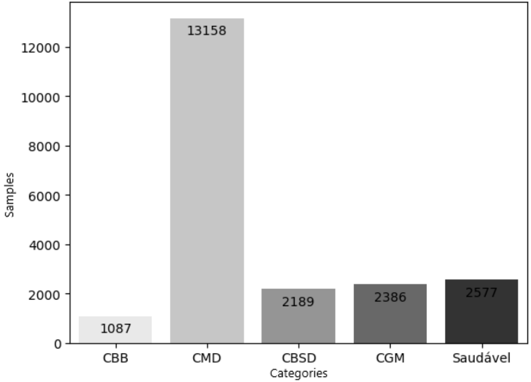


Figure 1: Graph of occurrences of each category in the data set.

Sample images

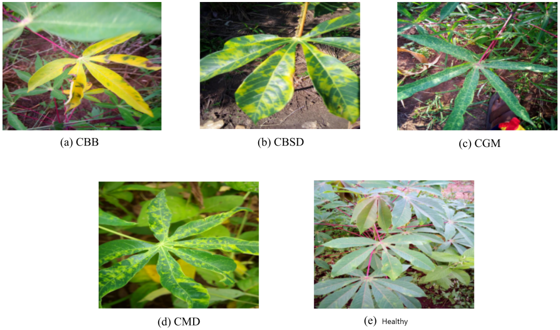


Figure 2: Images randomly selected from the dataset.

Subsets

Subsets

- The data set was divided into three subpartitions in a stratified manner.

Subsets

- The data set was divided into three subpartitions in a stratified manner.
- 70% for training.

Subsets

- The data set was divided into three subpartitions in a stratified manner.
- 70% for training.
- 10% for validation.

Subsets

- The data set was divided into three subpartitions in a stratified manner.
- 70% for training.
- 10% for validation.
- 20% for testing.

Classification Task

Classification Task

- Select CNN models for image classification, based on ImageNet results and recent publications.

Classification Task

- Select CNN models for image classification, based on ImageNet results and recent publications.
- Adjust the models for the task and classification into five classes using Transfer Learning.

Classification Task

- Select CNN models for image classification, based on ImageNet results and recent publications.
- Adjust the models for the task and classification into five classes using Transfer Learning.
- Evaluate the models in general and by class.

Selected models

Selected models

- EfficientNet B3

Selected models

- EfficientNet B3
- InceptionV3

Selected models

- EfficientNet B3
- InceptionV3
- ResNet50

Selected models

- EfficientNet B3
- InceptionV3
- ResNet50
- VGG16

Evaluation metrics

Evaluation metrics

- Accuracy.

Evaluation metrics

- Accuracy.
- Precision.

Evaluation metrics

- Accuracy.
- Precision.
- Recall.

Evaluation metrics

- Accuracy.
- Precision.
- Recall.
- F-Score (F_1).

Programming Environment

Programming Environment

- Python 3.12.2

Programming Environment

- Python 3.12.2
- Pytorch 2.3.0

Model configurations and hyperparameters

Model configurations and hyperparameters

- Unique hyperparameter profile.

Model configurations and hyperparameters

- Unique hyperparameter profile.
- 50 training epochs.

Model configurations and hyperparameters

- Unique hyperparameter profile.
- 50 training epochs.
- Adam optimizer:

Model configurations and hyperparameters

- Unique hyperparameter profile.
- 50 training epochs.
- Adam optimizer:
  - Initial learning rate: 10^{-3}

Model configurations and hyperparameters

- Unique hyperparameter profile.
- 50 training epochs.
- Adam optimizer:
  - Initial learning rate: 10^{-3}
- Early stop:

Model configurations and hyperparameters

- Unique hyperparameter profile.
- 50 training epochs.
- Adam optimizer:
  - Initial learning rate: 10^{-3}
- Early stop:
  - Patience: 5 epochs

Model configurations and hyperparameters

- Unique hyperparameter profile.
- 50 training epochs.
- Adam optimizer:
  - Initial learning rate: 10^{-3}
- Early stop:
  - Patience: 5 epochs
- Learning rate scheduler

Model configurations and hyperparameters

- Unique hyperparameter profile.
- 50 training epochs.
- Adam optimizer:
  - Initial learning rate: 10^{-3}
- Early stop:
  - Patience: 5 epochs
- Learning rate scheduler
  - ReduceLROnPlateau

Model configurations and hyperparameters

- Unique hyperparameter profile.
- 50 training epochs.
- Adam optimizer:
  - Initial learning rate: 10^{-3}
- Early stop:
  - Patience: 5 epochs
- Learning rate scheduler
  - ReduceLROnPlateau
  - Factor: 0.1

Model configurations and hyperparameters

- Unique hyperparameter profile.
- 50 training epochs.
- Adam optimizer:
  - Initial learning rate: 10^{-3}
- Early stop:
  - Patience: 5 epochs
- Learning rate scheduler
  - ReduceLROnPlateau
  - Factor: 0.1
  - Patience: 2 epochs

Model configurations and hyperparameters

- Unique hyperparameter profile.
- 50 training epochs.
- Adam optimizer:
  - Initial learning rate: 10^{-3}
- Early stop:
  - Patience: 5 epochs
- Learning rate scheduler
  - ReduceLROnPlateau
  - Factor: 0.1
  - Patience: 2 epochs
- Batch size: 32

Pré-processamento

Pré-processamento

- The images have been resized as required for each model.

Pré-processamento

- The images have been resized as required for each model.
- A centralized crop is applied to the images.

Pré-processamento

- The images have been resized as required for each model.
- A centralized crop is applied to the images.
- Normalization of pixel values to the interval [0, 1], based on the mean and standard deviation of the ImageNet set.

Pré-processamento

- The images have been resized as required for each model.
- A centralized crop is applied to the images.
- Normalization of pixel values to the interval [0, 1], based on the mean and standard deviation of the ImageNet set.
- Data Augmentation was used in the training set.

Data Augmentation

Data Augmentation

- It increases the variety and quantity of data available for training. This is done by applying transformations and variations to existing images, creating new versions of this data.

Data Augmentation

- It increases the variety and quantity of data available for training. This is done by applying transformations and variations to existing images, creating new versions of this data.
- Rotation, random cropping, horizontal *flip* and brightness changes were used on the training images.

Results

Table 1: Results table

Models	Parameters (M)	Accuracy	Precision	Recall	F_1
EfficientNet-B3	12	0.877	0.878	0.877	0.877
InceptionV3	23.8	0.866	0.867	0.866	0.866
ResNet50	25.6	0.863	0.862	0.863	0.862
VGG16	138	0.615	0.378	0.615	0.468

Discussion

Discussion

- The *F1-Score* metric was considered for the main comparison of performance between models because it offers a balance between Precision and Revocation.

Discussion

- The *F1-Score* metric was considered for the main comparison of performance between models because it offers a balance between Precision and Revocation.
- EfficientNet-B3 obtained the best results, with an F_1 of 0.877, indicating great performance, even on an unbalanced dataset.

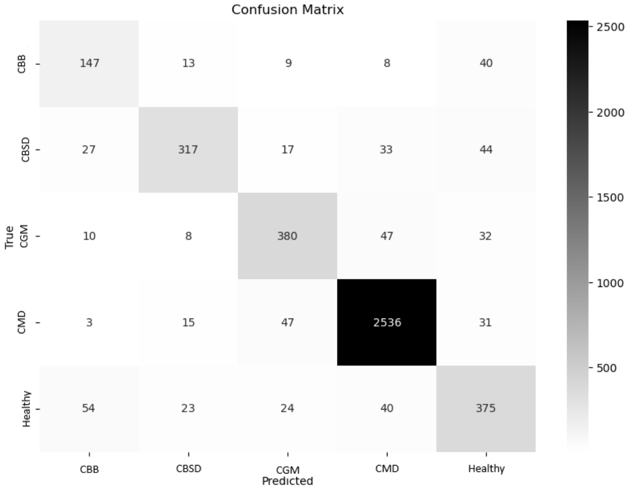
Discussion

- The *F1-Score* metric was considered for the main comparison of performance between models because it offers a balance between Precision and Revocation.
- EfficientNet-B3 obtained the best results, with an F_1 of 0.877, indicating great performance, even on an unbalanced dataset.
- InceptionV3 and ResNet50 presented results considered satisfactory, but were inferior to EffiecientNet-B3, with F_1 of 0.866 and 0.863, respectively.

Discussion

- The *F1-Score* metric was considered for the main comparison of performance between models because it offers a balance between Precision and Revocation.
- EfficientNet-B3 obtained the best results, with an F_1 of 0.877, indicating great performance, even on an unbalanced dataset.
- InceptionV3 and ResNet50 presented results considered satisfactory, but were inferior to EffiecientNet-B3, with F_1 of 0.866 and 0.863, respectively.
- VGG16 struggled to generalize correctly on such an unbalanced dataset. All predictions tended to concentrate on the majority class. It is therefore considered a degenerate network.

EfficientNet-B3 Confusion Matrix



EfficientNet-B3 Classification Report

Table 2: Classification Report

class	precision	recall	f1-score	support
CBB	0.610	0.677	0.642	217
CBSD	0.843	0.724	0.779	438
CGM	0.797	0.797	0.797	477
CMD	0.952	0.964	0.958	2632
Healthy	0.718	0.727	0.723	516

Conclusion

Conclusion

- It is possible to train an efficient model based on Deep Learning using Transfer Learning for the classification of cassava pathologies.

Conclusion

- It is possible to train an efficient model based on Deep Learning using Transfer Learning for the classification of cassava pathologies.
- **Most efficient model: EfficientNet-B3** with results considered satisfactory for the pathology classification task.

Conclusion

- It is possible to train an efficient model based on Deep Learning using Transfer Learning for the classification of cassava pathologies.
- **Most efficient model: EfficientNet-B3** with results considered satisfactory for the pathology classification task.
- Problem: VGG16 was unable to generalize the problem, perhaps it could be improved by varying the hyperparameters.

Future Work

Future Work

- Transformer Models to classify pathologies. For example: *CoCa* or *ViT* (Vision Transformer).

Future Work

- Transformer Models to classify pathologies. For example: *CoCa* or *ViT* (Vision Transformer).
- Hyperparameter adjustment, especially for VGG16:

Future Work

- Transformer Models to classify pathologies. For example: *CoCa* or *ViT* (Vision Transformer).
- Hyperparameter adjustment, especially for VGG16:
  - Optimizer

Future Work

- Transformer Models to classify pathologies. For example: *CoCa* or *ViT* (Vision Transformer).
- Hyperparameter adjustment, especially for VGG16:
  - Optimizer
  - Initial Learning Rate

Future Work

- Transformer Models to classify pathologies. For example: *CoCa* or *ViT* (Vision Transformer).
- Hyperparameter adjustment, especially for VGG16:
  - Optimizer
  - Initial Learning Rate
  - Other relevant parameters

Q&A