VILNIUS UNIVERSITY
FACULTY OF MATHEMATICS AND INFORMATICS
SOFTWARE ENGINEERING

„Cassava“ lapų ligų atpažinimas pasitelkiant kompiuterinę regą

Cassava Leaf Disease Recognition using Computer Vision

Bachelors work

By: Aivaras Petrikas

Supervisor: asist. dr. Vytautas Valaitis

Reviewer: asist. dr. Linas Litvinas

Vilnius – 2023

The author is thankful for the high performance computing resources provided by the Information Technology Research Center of Vilnius University.

Abstract

This paper explores the development and refinement of innovative deep-learning architectures for image classification tasks of classifying cassava leaf diseases. The backbone of this research is the EfficientNetB0 model, one of the most widely-utilized Convolutional Neural Networks developed by Google. Various methods to address class imbalance issues were explored, including the use of sigmoid focal loss, class weights, and over/under-sampling techniques. The most promising results were observed in the over/under-sampling approach. To enhance performance, the number of MBConv blocks within EfficientNetB0 was reduced, leading to improved classification outcomes and a lighter model, requiring less memory and resources to operate. The SE attention module was replaced with the ECA module, successfully reducing the model size and boosting its accuracy, thereby advancing the model's efficiency. Hybrid architectures were also investigated, with a transformer encoder connected to the EfficientNetB0 output, where 7x7 feature maps were utilized for classification. Although a slight decrease in accuracy was recorded with the cassava leaf disease dataset, a performance improvement was observed when applied to the AID dataset.

Keywords: Cassava leaves, classification, EfficientNet, ECA module, transformer, class imbalance

Santrauka

Šiame straipsnyje nagrinėjamas novatoriškų giliojo mokymosi architektūrų kūrimas ir tobulinimas siekiant spręsti manijokų lapų ligų klasifikavimo užduotį. Šio tyrimo pagrindas – EfficientNetB0 modelis, vienas iš plačiausiai naudojamų konvoliucinių neuroninių tinklų, kurį sukūrė „Google“. Išnagrinėti įvairūs metodai klasių disbalanso problemoms spręsti, įskaitant sigmoidinio židinio nuostolių, klasių svorių bei padidintos ir sumažintos imties metodų naudojimą. Geriausius klasifikavimo rezultatus pasiekė padidintos ir sumažintos imties metodas. Siekiant padidinti našumą, EfficientNetB0 modelyje buvo sumažintas MBConv blokų skaičius, kas pagerino klasifikavimo rezultatus, o modelis tapo lengvesnis, ir jam veikti reikia mažiau atminties ir išteklių. SE dėmesio modulis buvo pakeistas ECA moduliu, kas leido sėkmingai sumažinti modelio dydį ir padidinti jo tikslumą, taip padidinant modelio efektyvumą. Taip pat buvo ištirta hibridinė architektūra, prie EfficientNetB0 išvesties prijungus transformerių neuroninio tinklo koduotoją, kai klasifikavimui buvo naudojami 7x7 požymių žemėlapiai. Nors naudojant manijokų lapų ligų duomenų rinkinį buvo užfiksuotas nedidelis tikslumo sumažėjimas, tačiau, taikant hibridinį modelį AID duomenų rinkiniui, buvo pastebėtas našumo pagerėjimas.

Raktiniai žodžiai: Manijokų lapai, klasifikavimas, EfficientNet, ECA modulis, transformerių neuroninis tinklas, klasių disbalansas

CONTENTS

INTRODUCTION 6
Aim 6
Objectives 6

1. NEURAL NETWORKS 8
1.1. Multilayer perceptron 8
1.2. Convolutional Neural Networks 9

2. TRAINING THE MODEL TO CLASSIFY CASSAVA DISEASES 11
2.1. Data preparation for training 11
2.2. Knowledge transfer 11
2.3. Classifier setup 11
2.4. EfficientNet architecture 12

3. SOLVING CASSAVA LEAF DISEASE DATASET IMBALANCE 13
3.1. Cassava leaf disease dataset 13
3.2. Class weights 13
3.3. Oversampling and Undersampling 14
3.4. Focal loss function 14
3.5. Results 15

4. MODIFYING EFFICIENTNET 18
4.1. Reducing MBConv module count 18
4.2. SE block replacement 18
4.3. Results 20

5. COMBINING EFFICIENTNET WITH VIT 21
5.1. Transformers and ViTs 21
5.2. EfficientNet and ViT unimodal 21
5.3. Benefits and shortcomings of connecting EfficientNetB0 and ViT 23

6. RESULTS AND CONCLUSIONS 26
6.1. Results 26
6.2. Conclusions 26

REFERENCES 28

Introduction

Cassava is the second largest carbohydrate source in Africa and is one the most important sources of food which can sustain harsh climatic conditions. At least 80 % of households in sub-Saharan Africa grow these roots, but a variety of spreading diseases are devastating the crop. It is important to detect these diseases early to save the harvest. To do this, African officials fund agri-cultural experts to go to farmers and visually diagnose Cassava leaves, but this is expensive and inefficient. Computer vision can be used to solve this problem, but it is difficult because farmers do not have professional equipment and can only scan photos of the leaves with poor quality mobile phone cameras [MME+21]. Other challenges in classifying plant leaves that modern computer vision models face include different lighting of the photographs, similarities in leaf morphology, hard-to-see signs of leaf diseases, and differences in the background of the photographs [Dan21]. Computer vision algorithms could help to identify these diseases and thus help to preserve the Cassava crop. Although modern solutions have shown to have a respectable classification, the devices of African farmers that would scan the leaves may face difficulties with large neural networks and may show insufficient performance. One of the state-of-the-art approaches to solving the image classification problem is EfficientNet convolutional neural networks developed by Google. However, there are improvements and adjustments that can be made to the network so as to improve classification accuracy or make it more applicable for usage with outdated hand-held devices.

Aim

This thesis investigates modifications of the contemporary neural network EfficientNetB0 that can improve its classification accuracy as well as reduce its size and reduce its computational cost without sacrificing significant classification performance.

Objectives

1. To solve class imbalance using oversampling and undersampling with the combination of image augmentation techniques.
2. Reducing the number of parameters in the EfficientNetB0 model by removing excessive inverted residual blocks.
3. Replacing Squeeze-and-excitation blocks with Efficient Channel Attention blocks in the EfficientNetB0 to make the model more efficient.

4. Connect transformer encoder to the output of EfficientNetB0 and make it classify Cassava leaf diseases.

1. Neural networks

1.1. Multilayer perceptron

A neural network is a computational model consisting of basic processing entities that mimic the behavior of neurons in the biological world. These artificial neurons interact by transmitting signals via a network of connections characterized by weights and biases.

On the other hand, a perceptron refers to a compact architecture of a neural network, comprising a solitary neuron split into input and output layers. The role of the input layer is to accept inputs, while the output layer is structured to produce the output.

The principle of the Perceptron is as follows:

1. The perceptron receives inputs and uses them for reading.
2. If the result of the reading is positive, the perceptron provides a positive output.
3. If the result of the reading is negative, the perceptron provides a negative output.
4. An output value equal to a value is obtained by adding the sum of the products of the neuron's inputs and the weights of the corresponding inputs as an argument to an activation function (e.g. sigmoid, threshold...).

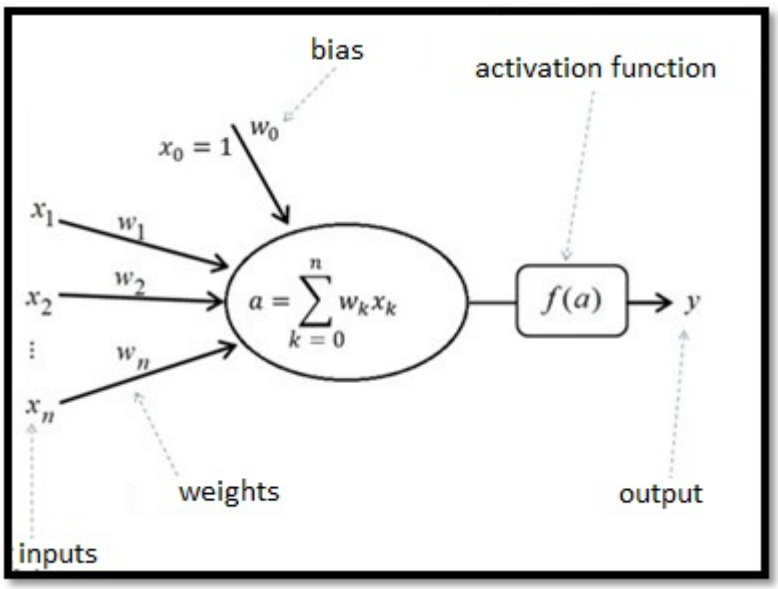

Fig. 1. Perceptron output formula

A multilayer perceptron is constructed by connecting these neurons into more complex networks with multiple layers. A multi-layer perceptron can be used to perform a wider range of tasks, perform more complex calculations and produce classifiers because, unlike a perceptron, it can have more than one output.

A neural network has to be trained, in other words, configured so that the desired set of inputs feeds the desired set of outputs. This requires a data set and an optimisation algorithm. During training, the neural network is given the inputs of the dataset and the correct output. The optimisation algorithm modifies the weights and biases so that the neural network returns as close as possible to the correct output. The difference between the values guessed by the network and the real values is measured in terms of loss, and the objective of the optimisation algorithm is to minimise the loss. This process is repeated over more than one the network training phase called an epoch, where the model is given all the data in the training dataset [KS96].

1.2. Convolutional Neural Networks

A convolutional neural network (CNN) represents a specific variety of neural networks, tailored to function with grid-oriented structures, such as images. The initial algorithm recognizing symbols in images was created by Yann LeCun, a French computer scientist, which spurred the employment of CNNs for tasks like image classification, object recognition, among others that involve images.

Convolutional networks are characterized by a one-way flow of information from the input layer through to the output layer. They are structured with convolutional layers, pooling layers, and fully connected layers [RW22].

- The role of convolutional layers in a neural network is to extract distinguishing characteristics from the input image. Compact filters, characterized by particular weightings, are employed to fulfill this task. By performing convolution operations, these filters derive feature maps, enabling the network to identify various features, edges, shapes, and so on, of the object within the image.
- Subsequently, this information is channeled into a pooling layer, which condenses the feature arrays into arrays of lower dimensions. This process involves calculating either the maximum or the average of the data within the neighboring array.
- Finally, a fully connected layer of neurons takes over, further manipulating the data and consolidating it into an output layer.

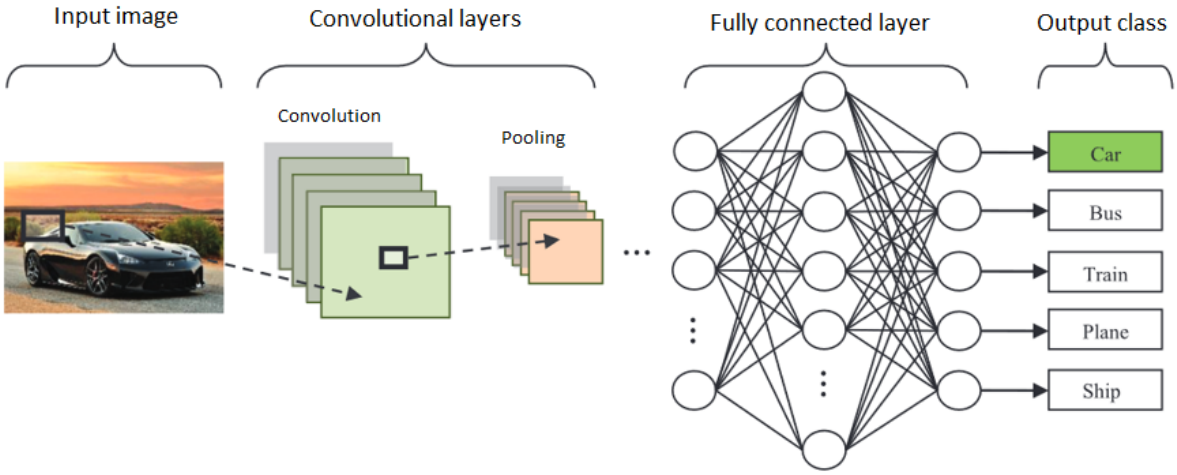

Fig. 2. Convolutional neural network

2. Training the model to classify cassava diseases

2.1. Data preparation for training

At first, a test set is created what contains 200 images of each class while the rest of the images are set to be used for training. The training and validation sets are divided by a ratio of 0.85 and 0.15. To feed the images into the model training process, the ImageLoader component of the Tensorflow library was used to detect various augmentations when loading images from the training dataset: random zooms, flips, rotations, etc. This process prevents the model from overfitting [WP17]. To speed up the training time, the images have been reduced to 350 x 350 pixels.

2.2. Knowledge transfer

Knowledge transfer is a machine learning technique where the knowledge accumulated by a trained model for one task is transferred and used to perform another task. Training a large neural network with large datasets takes a lot of time and resources, so knowledge transfer weights and biases can reduce training time and often provide satisfactory classification results. Knowledge transfer has been applied to the development of the Cassava leaf classifiers. Before training the EfficientNetB0 models on the Cassava leaf dataset, the ImageNet weights were loaded, which these models acquired after a long training process on the ImageNet dataset consisting of 14 million images.

2.3. Classifier setup

As a backbone for further experiments I used one of the awarded submissions of Cassava leaf disease classification competition. The code performs a cassava leaf disease classification task using a TensorFlow and Keras-based deep learning model. It employs an EfficientNetB0 model, pre-trained on ImageNet, excluding the top layers and including custom layers for the specific task. The model is compiled using the Adam optimizer, categorical cross-entropy loss (with label smoothing), and tracks categorical accuracy. Training includes model checkpointing, and learning rate reduction based on validation loss. The model is trained for 10 epochs with a batch size of 15 [HG21].

2.4. EfficientNet architecture

EfficientNet comprises a set of convolutional neural network designs tailored to deliver top-notch accuracy, using fewer parameters and consuming less computational power compared to its predecessors. The design hinges on a compound scaling approach, which simultaneously adjusts the depth, width, and resolution of the network.

The structure of EfficientNet begins with an introductory convolutional layer, succeeded by several blocks. Each of these blocks consists of an array of convolutional layers, batch normalization, and activation functions. These blocks are interconnected through skip connections, enhancing gradient flow and making the training process more efficient.

A distinctive feature of EfficientNet is the implementation of "squeeze-and-excitation" (SE) blocks. These SE blocks dynamically adjust channel-wise feature reactions by determining attention weights, enabling the network to concentrate on significant features and downplay irrelevant ones.

In conclusion, EfficientNet has demonstrated remarkable effectiveness in diverse computer vision applications, demanding less resources compared to other top-tier models. [KPP21]

3. Solving Cassava Leaf Disease dataset imbalance

3.1. Cassava leaf disease dataset

The dataset contains 21,367 photos of Cassava leaves sent by people in Uganda as part of a survey by the National Crops Resources Research Institute. The format of the photos reflects real data that would have been collected from farmers' farms using mobile devices. The set contains 5 classes: Cassava Bacterial Blight (0), Cassava Brown Streak Disease (1), Cassava Green Mottle (2), Cassava Mosaic Disease (3), Healthy (4). Out of 21,367 images, one class has barely more than 1000 images were taken for Cassava Bacterial Blight while the Cassava Mosaic Disease has over 13000 images.

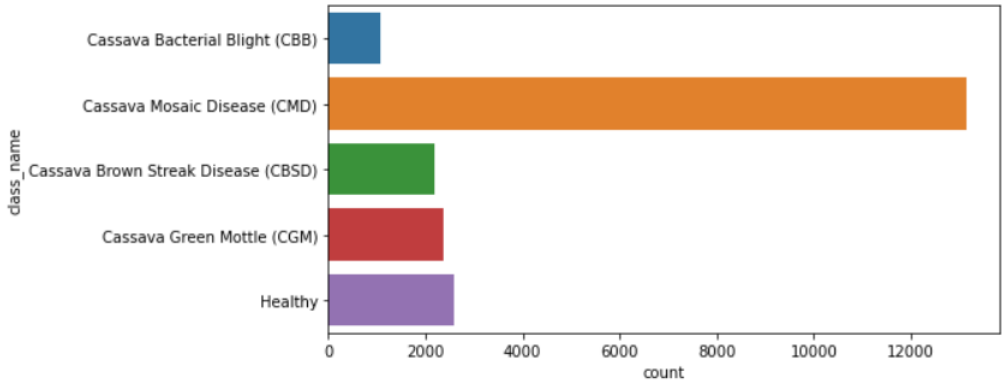

Fig. 3. Cassava leaf disease dataset distribution

Class imbalance is a big problem in machine learning because it refers to an unequal distribution of classes within a dataset, which is encountered mostly in classification tasks where the distribution of classes or labels in a given dataset is not uniform. This problem affects the performance of models as they tend to lean more towards the majority class and eliminate the minority class, resulting in poor performance in evaluation metrics such as precision, recall, F1-score, and ROC score [HG09].

3.2. Class weights

Class weighting is a technique used to balance the training procedure by changing the weight that each training sample has when computing the training loss. In this study, class weighting was introduced, and the weight of each sample was related to its class. The weights of a class are usually tied to its proportion of image samples in the dataset [DAG20]. In this experiment, the class_weight parameter is used in the "fit" function for a model in Keras, a popular deep-learning library in Python. The class_weight parameter is used to specify weights for different classes

during model training. The class_weight parameter accepts a dictionary where each key-value pair corresponds to a class and its associated weight. The keys are typically class indices, and the values are the associated weights. Class weights are used to modify the loss function during training. During each update, the loss for each sample is multiplied by the class weight of that sample's class. This means that if you assign a higher weight to the minority class, the model will "pay more attention" to examples from that class - it will treat errors on these examples as more important, and consequently adjust itself more to reduce these errors.

3.3. Oversampling and Undersampling

Oversampling and undersampling are two widely adopted resampling techniques used to handle the problem of class imbalance in machine learning. Class imbalance refers to an unequal distribution of classes within a dataset, which is encountered mostly in classification tasks where the distribution of classes or labels in a given dataset is not uniform. Undersampling is the process of decreasing the amount of majority target instances or samples while oversampling can be performed by increasing the amount of minority class instances or samples by producing new instances or repeating some instances [HG09]. This imbalance shall be solved by reducing the number of Cassava Mosaic Disease pictures in the training process, however, the number of other classes could be increased using image augmentations. Techniques like Image cropping, flipping, rotation, color transformation, and noise injection can be used to create datasets without gathering more original images [PGB20]. Therefore, in this experiment, undersampling and oversampling were from the "imblearn" library one after the other, respectively.

3.4. Focal loss function

The Focal Loss function was conceived to combat the class imbalance issue common in object detection, characterized by a surplus of straightforward negative examples relative to challenging positive ones. The conventional cross-entropy loss assigns equal importance to all examples, which can result in the model being dominated by the numerous easy negatives, thus not paying adequate attention to the hard positives.

To counteract this, the Focal Loss function incorporates a modulating factor into the traditional cross-entropy loss that reduces the influence of easy examples and emphasizes difficult ones. Specifically, it includes a factor of (1 - pt) which is powered to the index equal to gamma in the cross-entropy loss. The pt denotes the predicted probability of the correct class and gamma is an adjustable parameter determining the extent to which easy examples are down-weighted.

When gamma is zero, the Focal Loss reverts back to the standard cross-entropy loss.

By concentrating more on difficult examples, the Focal Loss aids in enhancing detection accuracy for objects that are either scarce or challenging to detect. [LGG+17]

Sigmoid Focal Loss, a novel loss function, was used for Cassava Leaf Disease Classification to address class imbalance issues.

3.5. Results

Table 1. Model Performance Metrics for Different Strategies

Strategy	Accuracy	Precision	Recall	F1-Score
Original	74,80%	77,95%	74,80%	74,58%
Over/Under Sampling	74,50%	75,79%	74,50%	74,26%
Focal Loss	68,10%	74,01%	68,10%	67,81%
Class Weights	73,50%	73,62%	73,50%	73,36%

Table 2. Model Accuracies for Different Strategies

Strategy	CBB	CBSD	CGM	CMD	Healthy
Over/Under Sampling	82.50%	75.00%	64.50%	91.00%	59.50%
Focal Loss	57.00%	49.50%	62.50%	98.00%	73.50%
Original	50.00%	78.00%	69.50%	95.00%	81.50%
Class Weights	76.50%	76.00%	71.00%	85.50%	58.50%

With the original strategy, accuracy, precision, recall, and F1-score were relatively high, with values of 74.80%, 77.96%, 74.80%, and 74.58% respectively. When the over/under sampling strategy was applied, there was a slight decrease in all metrics - accuracy was 74.50%, precision was 75.79%, recall was 74.50%, and the F1-score was 74.26%. The focal loss strategy had a more significant decrease, with accuracy, precision, recall, and F1-score dropping to 68.10%, 74.01%, 68.10%, and 67.81% respectively. The final strategy, using class weights, demonstrated improved performance compared to the focal loss strategy but was still lower than the original and over/under sampling strategies. The metrics for the class weights strategy were 73.50% for accuracy, 73.62% for precision, 73.50% for recall, and 73.36% for the F1-score. These results indicate that handling class imbalance is a nuanced task and the choice of strategy can have significant impact on model performance.

Fig. 4. Convolution matrixes

Upon analyzing the confusion matrices resulting from the application of over/under sampling strategies, we can observe a consistent pattern of confusion between the classes of "Healthy" and "Cassava Bacterial Blight". This suggests that these sampling techniques are causing these two classes to become intertwined in the feature space, an unintended consequence that negatively affects model accuracy. Despite efforts to adjust the class balance through these strategies, this remains a significant challenge, underscoring the difficulty in distinguishing these classes. A visual inspection of the images confirms these findings, revealing strong similarities between the healthy and Cassava Bacterial Blight samples that may constitute a case of class overlap. This overlap could be an inherent characteristic of the dataset, implying that further improvements in class separation might be unattainable. The implication of these observations is that we may have reached the limit of what can be achieved with our current approach, employing alternative strategies or acceptance of the inherent limitations presented by this form of class overlap.

Healthy **CBB**

Fig. 5. Healthy vs CBB

4. Modifying EfficientNet

4.1. Reducing MBConv module count

The EfficientNet models, built upon a core structure of inverted residual blocks or MBConv modules, offer impressive accuracy on large datasets. However, these models may not offer the same efficacy when applied to small datasets, such as the Cassava leaf dataset that contains just over 20,000 images. Reducing the number of MBConv modules can offer a viable solution in such scenarios, as it can considerably reduce model complexity and computational effort. The principle behind this adjustment is to prevent overfitting, a common issue when high-capacity models like EfficientNet are applied to limited datasets [LCX+22]. By reducing the MBConv module count, not only are we avoiding the risk of overfitting, but also potentially improving the model's accuracy. Furthermore, this approach also aids in reducing the overall model size, an essential aspect for systems with limited computational resources or for applications demanding low-latency predictions. Therefore, model customization, specifically reducing the number of MBConv modules, can enhance performance on small datasets while providing the added benefits of reduced model size and improved computational efficiency. Two types of tactics for reducing the MBConv modules were selected: "T1" (9 modules) and "T2" (6 modules).

Table 3. Stage Specifications

Stage	Operator	Resolution	Channel	Repeats (Og)	Repeats (T1)	Repeats (T2)
1	MBConv6, k3 x 3	112 x 112	16	1	1	1
2	MBConv6, k3 x 3	112 x 112	24	2	2	1
3	MBConv6, k5 x 5	56 x 56	40	2	2	1
4	MBConv6, k3 x 3	28 x 28	80	3	2	1
5	MBConv6, k5 x 5	14 x 14	112	3	1	1
6	MBConv6, k5 x 5	14 x 14	192	4	0	0
7	MBConv6, k3 x 3	7 x 7	320	1	1	1

"Og" in the table stands for the original EfficientNetB0 layout.

4.2. SE block replacement

The incorporation of channel attention has significantly improved convolutional neural network performance, leading to the development of intricate attention modules. One popular approach is the Squeeze-and-excitation (SE) module, introduced in 2017. However, dimensionality

reduction affects prediction performance and efficiency. To address this, the Efficient Channel Attention (ECA) module was introduced, enhancing performance with fewer parameters. By replacing SE with ECA parameter count can be reduced.

The SE module applies global average pooling and two fully connected layers to learn channel-wise dependencies. However, this approach diminishes detailed information during pooling and increases computational expense. On the other hand, the ECA module employs a 1D convolution with a kernel size of 1, capturing local cross-channel interactions without shrinking feature map dimensions, thereby retaining intricate details.

The ECA module offers several benefits over SE. It introduces the reduction of dimensionality and trims down the parameters, ensuring the preservation of critical information which in turn boosts the performance of predictions. Moreover, it utilizes fewer parameters in comparison to the fully connected layers in SE, leading to a model architecture that is more efficient.

To sum it up, while the SE module enjoys widespread usage, the ECA module presents a superior alternative in terms of both performance and efficiency. The choice between the two largely hinges on the requirements of the application and the balance between computational complexity and accuracy [HL21].

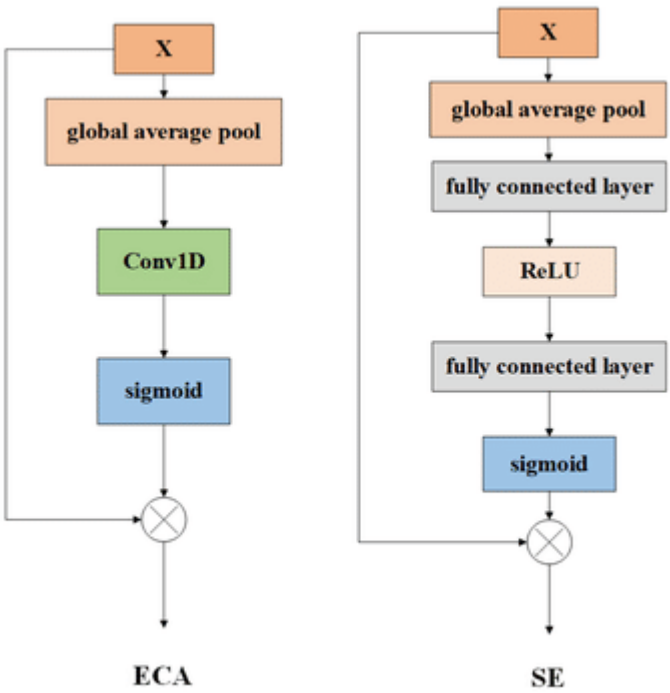

Fig. 6. ECA vs SE

4.3. Results

Table 4. Model Specifications and Performance Metrics

Model	Parameters	Model Size	Accuracy	Precision	Recall	F1-Score
T1	1714288	20.01 MB	75,10%	75,72%	75,10%	75,27%
ECA	3742316	43.27 MB	75,00%	75,92%	75,00%	75,13%
T2	1565644	18.17 MB	74,10%	75,16%	74,10%	74,26%
Original	4378792	50.68 MB	74,50%	75,78%	74,50%	74,25%
ECA + T1	1547856	18.04 MB	74,60%	75,38%	74,60%	74,70%

5. Combining EfficientNet with ViT

5.1. Transformers and ViTs

The Transformer model, a unique type of neural network design, was introduced by Vaswani and his team in 2017, specifically tailored for tasks related to natural language processing such as machine translation. It is constructed on the foundation of an encoder-decoder framework, a standard setup for tasks that require translation of sequences.

What sets the transformer model apart from preceding encoder-decoder constructs is its use of self-attention mechanisms rather than recurrent or convolutional layers. Self-attention enables the model to concentrate on various segments of the input sequence while forming each part of the output sequence, bypassing the restrictions imposed by the sequential character of recurrent or convolutional layers.

The Transformer model incorporates residual connections and layer normalization to reinforce training stability and accelerate the process. Furthermore, it employs a method known as multi-head attention which lets the model focus on multiple areas of the input sequence concurrently.

Finally, the Transformer model has proven its ability to deliver cutting-edge results across a multitude of tasks in the realm of natural language processing, encompassing machine translation, language modeling, and text generation [LCF18].

5.2. EfficientNet and ViT unimodal

A new method of combining EfficientNet and visual transformer network was proposed and the merged model achieved a notable 91% accuracy on the brain tumor dataset, surpassing traditional models. The renowned ResNet 152 model achieved 85%, while standalone ViT reached 89% accuracy. Evidently, the novel combination of EfficientNet and ViT outperformed these models, promising improved classification systems for similar datasets in the future [KBT+22]. There was another study where EfficientNet and Vision Transformers are used together in a mixed convolutional-transformer network to detect deep-fakes. The combination showed decent result when detecting deep-fake videos, recording F1 score very close to the state-of-the-art methods on the Deep- Fake Detection Challenge (DFDC) [CG21].

Specifically, EfficientNet is used as a feature extractor to extract visual features from the input video frames, while Vision Transformers are used to obtain an informative global description for the downstream task. The combination of these two models allows for more accurate detection

of deep-fakes by leveraging the strengths of both models.

EfficientNet is a family of convolutional neural networks that have been optimized for both accuracy and efficiency. These networks use a compound scaling method to balance model size and computational cost, resulting in highly efficient models that can achieve state-of-the-art performance on various computer vision tasks.

Vision Transformers, on the other hand, are based on the transformer architecture commonly used in natural language processing tasks. They use self-attention mechanisms to capture long-range dependencies between different parts of an image, allowing them to better understand the context and relationships between different objects in an image [CG21]. The same strategy was applied to building a model for Cassava leaf disease classification. The architecture for the unimodal can be seen in the figure below.

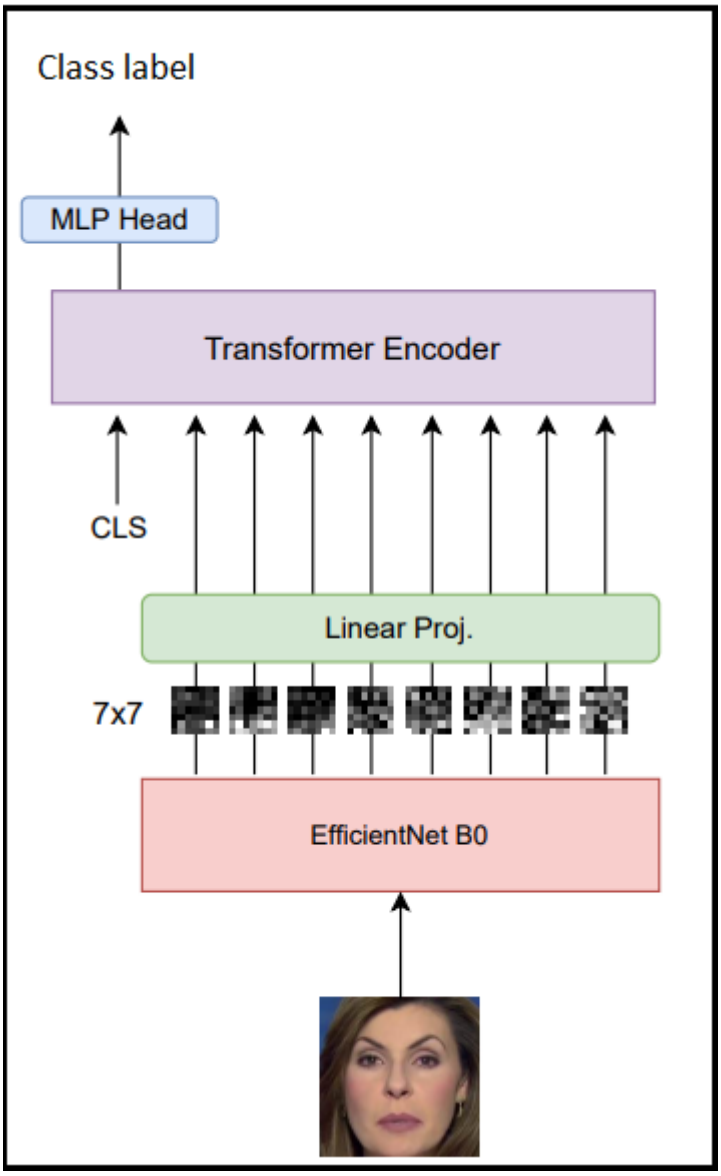

Fig. 7. EfficientNet and ViT hybrid model

The transformer encoder in this model is a stack of 4 transformer layers. Each transformer layer comprises two parts: a multi-head self-attention mechanism and a feed-forward neural network. The multi-head attention has 6 heads and a dimension per head of 64, which makes up for a total input dimension of 512. This input dimension is the same for the feed-forward network, which has a hidden layer dimension of 1024. Both the attention mechanism and the feed-forward network are wrapped in a Layer Normalization layer, and they use residual connections to add the input of the layer to its output. Dropout of rate 0.3 is applied for regularization. The outputs of these layers are then summed with the original input, following the standard practice in residual connections. This process is performed for each transformer layer in the encoder.

Table 5. EfficientNetB0 + ViT configuration data and results

Heads	Depth	Is over/under sampling used	Is EfficientNet Trained	Accuracy
3	2	No	No	43.3%
4	3	No	No	47.7%
6	4	No	No	41.2%
4	3	Yes	No	58.4%
4	3	Yes	Yes	65.1%

The accuracy results did not match the classification results of standalone EfficientNetB0 when classifying cassava leaf diseases.

5.3. Benefits and shortcomings of connecting EfficientNetB0 and ViT

EfficientNets, renowned for their exceptional balance of efficiency and precision, truly shine when employed in image classification tasks. These models work optimally, minimizing the computational and memory resources required, while achieving high levels of accuracy, making them invaluable in tasks such as snake species identification.

However, CNN models don't encode and capture the relations at the pixel level within the input image. Conversely, Vision Transformers excel in their capacity to process spatial information - an essential aspect when working with image data. They also display a significant ability in capturing long-range dependencies within images, a task often challenging for many models [LFM21].

The application of Vision Transformer (ViT) and EfficientNet for cassava leaf disease classification demonstrated underwhelming results due to the particular nature of these plant diseases. These illnesses manifest their characteristics through localized irregularities on the leaf

surfaces such as spots and blotches where the relevant information is predominantly concentrated in particular regions rather than being distributed across the whole image. This local feature orientation renders the global context, which ViT and EfficientNet prioritize, less significant. In contrast, a standalone Convolutional Neural Network (CNN) with its ability to focus more on local features and textures of images has been shown to deliver superior performance. Moreover, the unimodal nature of ViT + EfficientNet results in an overly complex model that is prone to overfitting the data, further undermining its efficacy in accurately classifying cassava leaf diseases.

Cassava Green Mottle **Cassava Bacterial Blight**

Fig. 8. Cassava leafs

When using the same setup of Vision Transformer (ViT) and EfficientNet, better results were found when classifying aerial images from the AID dataset. In these images, understanding the full picture - the 'global context' - is crucial to recognize whether the image shows a bridge, an airport, or a city. These models, which consider all parts of the image, are well-suited for this task because they can use the whole image to make their decision, rather than focusing on small, localized changes.

Fig. 9. AID Dataset

Table 6. Model Performance on AID dataset

Model Name	Accuracy
EfficientNetB0	96.5%
EfficientNetB0 + ViT	98.4%

6. Results and conclusions

6.1. Results

1. Approaches to address class imbalance: Various strategies including sigmoid focal loss, class weights, and over/under-sampling were experimented with in attempts to address class imbalance. Among these, the over/under sampling was found to yield the best results. This strategy was evidently successful when applied to the unimodal of transformer encoder and EfficientNetB0.
2. Improved performance with reduced MBConv modules: When the number of MBConv modules was reduced in the EfficientNetB0 architecture, it lead to a model which is 2.5 times smaller in memory and number of parameters as well as showed improvements in accuracy jumping from 74,5% to 75,1%.
3. Replacement of SE with ECA modules in EfficientNetB0: The model size was further reduced and its accuracy was even enhanced by the substitution of the SE attention module with an ECA module. The number of parameters decreased by 15% while the accuracy increased from 74,5% to 75%.
4. Integration of Transformer Encoder with EfficientNetB0: A Transformer Encoder was connected to the output of EfficientNetB0, wherein the Transformer utilized 7x7 feature maps for classification. Although this setup exhibited a slightly lower accuracy with the Cassava leaf disease dataset, a higher accuracy was shown with the AID dataset compared to using standalone EfficientNetB0.

6.2. Conclusions

1. Oversampling and undersampling efficacy: The technique of oversampling and undersampling has been demonstrated to be an effective method to tackle the issue of class imbalance in image classification tasks. By modifying the distribution of the training dataset, models can be better equipped to handle less represented classes and thereby improve their overall performance.
2. Reduced number of MBConv blocks in EfficientNetB0: The findings of this research suggest that the model's accuracy on smaller datasets can be improved by reducing the number of MBConv blocks in the EfficientNetB0 model. This indicates that overcomplication of the model structure could lead to overfitting and consequently degrade the model's performance on less abundant datasets.

3. Successful size reduction of EfficientNetB0 with ECA Module: Experiments in this research showed that the size of the EfficientNetB0 model can be decreased without sacrificing its performance by replacing the SE attention module with the ECA module on a smaller dataset. This leads to a more efficient model, with less computational demand and similar, if not improved, performance.
4. A hybrid model creation with EfficientNet and Transformer Encoder: A unimodal was created by fusing the EfficientNet with a transformer encoder that exhibited improved classification accuracy, particularly in scenarios where global relations between pixels are crucial.

References

[CG21] D. Coccomini, D. Gragnaniello. Combining EfficientNet and Vision Transformers for Video Deepfake Detection. 2021. Available also from: https://arxiv.org/abs/2106.05624.

[DAG20] D. Duarte, U. Andriolo, G. Gonçalves. Addressing the Class Imbalance Problem in the Automatic Image Classification of Coastal Litter from Orthophotos Derived from UAS Imagery. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. 2020, volume 5, pp. 439–446. Available from: https://doi.org/10.5194/isprs-annals-V-3-2020-439-2020.

[Dan21] S. Dane. *Plant Pathology 2021 - FGVC8* [https://kaggle.com/competitions/plant-pathology-2021-fgvc8]. 2021.

[HG09] H. He, E. A. Garcia. Learning from Imbalanced Data. *IEEE Transactions on Knowledge and Data Engineering*. 2009, volume 21, number 9, pp. 1263–1284.

[HG21] A. HG. *Cassava Disease - Keras TF EfficentNet [90]*. 2021. Available also from: https://www.kaggle.com/code/andreshg/cassava-disease-keras-tf-efficentnet-90/input. Kaggle Notebook Submission.

[HL21] M.-L. Huang, Y.-C. Liao. *Stacking Ensemble and ECA-EfficientNetV2 Convolutional Neural Networks on Classification of Multiple Chest Diseases Including COVID-19* [https://www.sciencedirect.com/science/article/pii/S1076633222006328]. 2021.

[KBT+22] R. Kadri, B. Bouaziz, M. Tmar1, F. Gargouri1. Multimodal deep learning based on the combination of EfficientNetV2 and ViT for Alzheimer's disease early diagnosis enhanced by SAGAN data augmentation. *International Journal of Computer Information Systems and Industrial Management Applications*. 2022, volume 2150-7988, number 3, pp. 313–325.

[KPP21] S. Kalvankar, H. Pandit, P. Parwate. Galaxy Morphology Classification using EfficientNet Architectures. *Monthly Notices of the Royal Astronomical Society*. 2021, volume 000, pp. 1–13.

[KS96] B. Krose, P. van der Smagt. *An introduction to Neural Networks*. Amsterdam, the Netherlands: The University of Amsterdam, 1996.

[LCF18] S. M. Lakew, M. Cettolo, M. Federico. A Comparison of Transformer and Recurrent Neural Networks on Multilingual Neural Machine Translation. In: *Proceedings of the 27th International Conference on Computational Linguistics*. 2018, pp. 4052–4062.

[LCX+22] C.-Y. Luo, S.-Y. Cheng, H. Xu, P. Li. *Human behaviour recognition model based on improved EfficientNET* [https://www.sciencedirect.com/science/article/pii/S187705092200045X]. 2022.

[LFM21] B. Lior, Friedrich, C. M. EfficientNets and Vision Transformers for Snake Species Identification Using Image and Location Information. *arXiv preprint arXiv:2108.09016*. 2021.

[LGG+17] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár. Focal loss for dense object detection. *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2980–2988. Available also from: https://openaccess.thecvf.com/content_ICCV_2017/papers/Lin_Focal_Loss_for_ICCV_2017_paper.pdf.

[MME+21] E. Mwebaze, J. Mostipak, J. Elliott, S. Dane. *Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review* [https://kaggle.com/competitions/cassava-leaf-disease-classification]. 2021.

[PGB20] J. A. Pandian, G. Geetharamani, A. B. *Data Augmentation On Plant Leaf Disease Image Dataset Using Image Manipulation and Deep Learning Techniques* [https://ieeexplore.ieee.org/abstract/document/8971580]. 2020.

[RW22] W. Rawat, Z. Wang. *Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review* [https://ieeexplore.ieee.org/abstract/document/8016501]. 2022.

[WP17] J. Wang, L. Pere. *The Effectiveness of Data Augmentation in Image Classification using Deep Learning* [http://vision.stanford.edu/teaching/cs231n/reports/2017/pdfs/300.pdf]. 2017.