AIN SHAMS UNIVERSITY
FACULTY OF COMPUTER AND INFORMATION SCIENCES

NEURAL NETWORKS PROJECT

Grapevines Species Classification Project

A. Project Overview

Grapevines are significant agricultural plants, primarily cultivated for their fruits, which are consumed fresh or processed into various products such as wine, juice, and raisins. Additionally, grapevine leaves hold importance as they are harvested annually and used for culinary purposes, particularly in Mediterranean cuisine. The species of grapevine leaves play a crucial role in determining the quality, taste, and market value of the final products.

B. Objective

The objective of this project is hosted as a Kaggle competition, to develop a deep learning-based classification system capable of accurately identifying grapevine leaf species from image data. By leveraging the power of deep learning algorithms, we aim to overcome the limitations of manual classification and provide a reliable and efficient solution for grapevine leaf species classification. This project follows a systematic approach, encompassing the following steps:

1. **Data Collection:** Obtain a dataset consisting of images of grapevine leaves belonging to different species.
2. **Preprocessing:** Apply preprocessing techniques such as resizing, normalization, and augmentation to prepare the data for training.
3. **Model Selection:** Choose appropriate deep learning CNN architectures for the classification task, considering factors such as model complexity and computational resources.
4. **Model Training:** Train the selected models using the preprocessed dataset, optimizing model parameters to minimize the classification error.
5. **Model Evaluation:** Evaluate the trained models on a separate test dataset to assess their performance in accurately classifying grapevine leaf species.
6. **Performance Analysis:** Analyze the performance of the trained models using metrics such as accuracy, precision, recall, and F1-score.
7. **Fine-tuning and Optimization:** Fine-tune the models and explore optimization techniques to further improve classification performance.
8. **Deployment:** Deploy the trained classification system for practical use, potentially integrating it into agricultural processes for automated species identification.

C. Dataset Overview

➢ Dataset Description

The dataset consists of images of grapevine leaves belonging to five distinct species: **AK, Ala-idris, Buzgulu, Dimnit, and Nazli**. Each image is captured using a special self-illuminating system, resulting in consistent lighting conditions across the dataset The dataset is divided into two sets: a training set for model training and a test set for evaluation. The training data comprises 350 samples of the grapevine leaf species for model learning and generalization. Within the training set, each class is represented by 70 samples, ensuring balanced representation across the different grapevine leaf species."

Fig.1 AK

Fig.2 Ala-idris

Fig.3 Buzuglu

Fig.4 Dimnit

Fig.5 Nazli

➢ Problem Statement

The visual similarity between grapevine leaf species poses a challenge for accurate classification. Human observers struggle to differentiate between the species, leading to inconsistencies and errors in manual classification efforts. This dataset serves as a valuable resource for training and evaluating deep learning models to address this classification challenge.

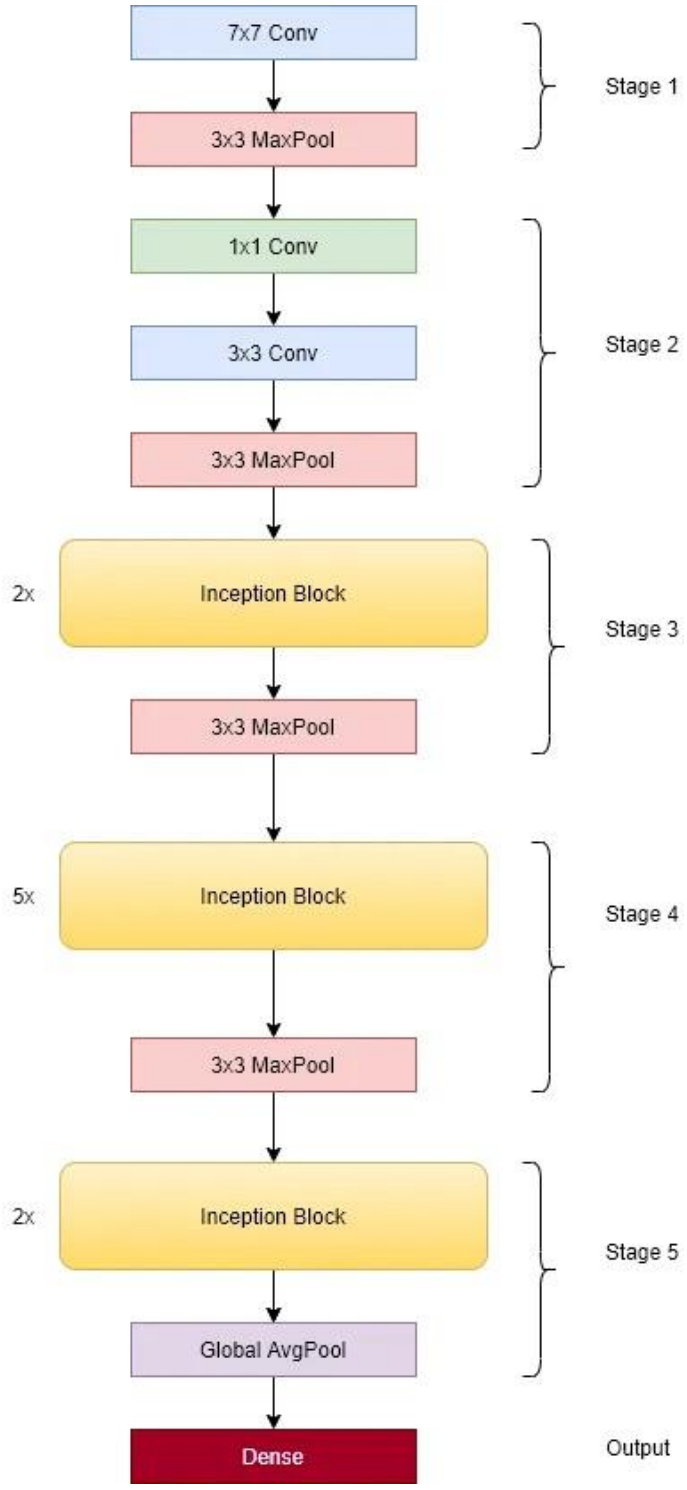
Methodology

In our exploration of deep learning architectures for grapevine leaf species classification, we initially experimented with building custom Convolutional Neural Network (CNN) architectures from scratch. However, these attempts yielded unsatisfactory results, with the models struggling to predict and classify the images correctly, resulting in low accuracy. Recognizing the limitations of our custom architectures, we transitioned to leveraging pretrained CNN architectures, which have been trained on large-scale image datasets and have demonstrated strong performance in various image classification tasks. We explored several state-of-the-art pretrained architectures, including Inception, EfficientNet, MobileNet, Xception, and ViT (Vision Transformer), adapting them to suit the requirements of our grapevine leaf species classification task.

1. Inception V3

The LeNet architecture used 5x5 convolutions, AlexNet used 3x3, 5x5, 11x11 convolutions and VGG used some other mix of 3x3, and 5x5 convolutions. But the questions that deep learning scientists were worried about were which combination of convolutions to use in different datasets to get the best results. For example, if we pick 5x5 convolutions, we end up with a fair number of parameters, there are a lot more multiplications involved, and they need a lot of parameters and are very slow, but on the other hand, it is very expressive. But if we pick 1x1 convolutions, it is much faster and does not need much memory, but maybe it does not work so well. Keeping these questions in mind, a brilliant idea was proposed in the Google LeNet paper — why not just pick them all, and stack them up in various convolutional blocks. It is also called the Inception paper, based on the movie Inception, and its famous dialogue — '**We need to go deeper**'.

1.1. Architecture



1.2. Trials

❖**Trial 1:** Image size (275x275), white background,2 dense layers (1024,512), no dropout, Average Pooling, epochs =10, batch size = 32

Train Accuracy	Val accuracy	Test Accuracy
89%	87%	80%

❖**Trial 2:** Image size (480x480), black backgroud,3 dense layers (1024,512,256), Average Pooling, dropout rate (0.2), epochs = 15, batch size = 32

Train Accuracy	Val accuracy	Test Accuracy
91%	90%	94%

❖**Trial 3:** Image size(480x480), black background,4 dense layers (1024,512 ,256,64), Average Pooling, dropout rate (0.2), epochs = 25, batch size = 64

Train Accuracy	Val accuracy	Test Accuracy
97%	93%	96%

Final Model Architecture:

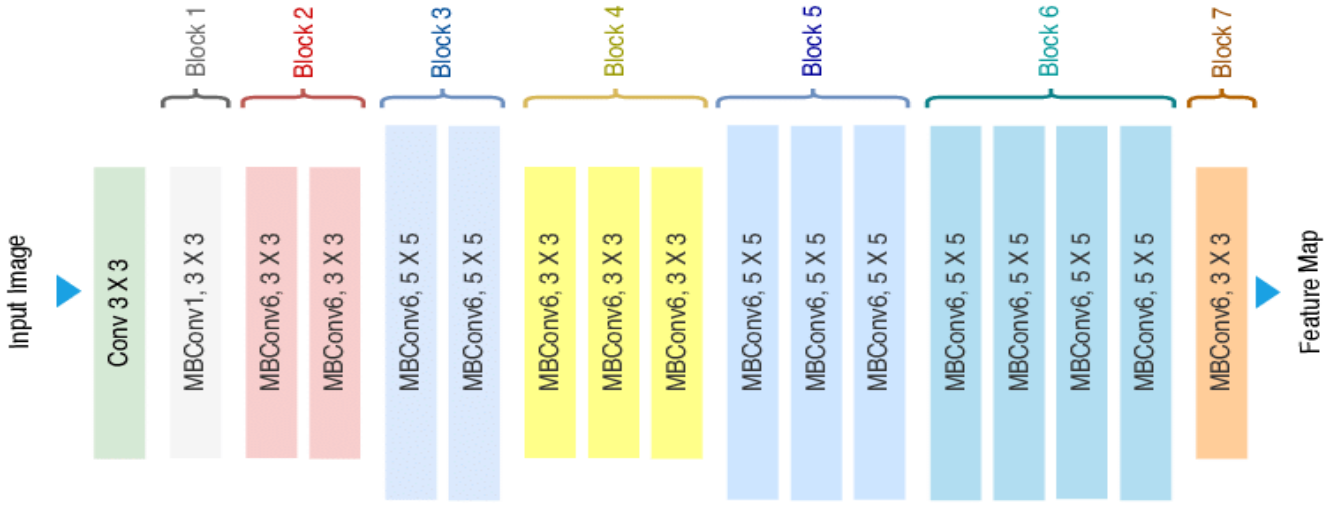
Inception V3**:** Image size(480x480), black background,4 dense layers (1024,512 ,256,64), Average Pooling, dropout rate (0.2), epochs = 25, batch size = 64

Test Accuracy **96**%

2. EfficientNet

EfficientNet is a **convolutional neural network** built upon a concept called "**Compound Scaling**." This concept addresses the longstanding trade-off between model size, accuracy, and computational efficiency. The idea behind compound scaling is to scale three essential dimensions of a neural network: width, depth, and resolution. EfficientNet comes in different variants, such as EfficientNet-B0, EfficientNet-B1, and so on, with varying scaling coefficients. Each variant represents a different trade-off between model size and accuracy, enabling users to select the appropriate model variant based on their specific requirements.

.1. Architecture



.2. Trials

❖**Trial 1:** EfficientNet **B0** Image size (400x400), white background,3 dense layers (1024,512,256), dropout rate (0.3), epochs=15, batch size = 32

Train Accuracy	Val accuracy	Test Accuracy
91%	92%	94%

❖**Trial 2:** EfficientNet **B0**, Image size (427x427), white backgroud,2 dense layers (1024 ,256), Average Pooling, dropout rate (0.2), epochs = 15, batch size = 32

Train Accuracy	Val accuracy	Test Accuracy
95%	97%	96%

❖**Trial 3:** EfficientNet **B3**, Image size(420x420), white background,2 dense layers (512 ,256), Average Pooling, dropout rate (0.3), epochs = 12, batch size = 32

Train Accuracy	Val accuracy	Test Accuracy
90%	91%	82%

❖**Trial 4:** EfficientNet **B1**, Image size(425x425), black background,3dense layers (1024,512 ,256), Average Pooling, dropout rate (0.3), epochs = 18, batch size = 32

Train Accuracy	Val accuracy	Test Accuracy
89%	90%	80%

Final Model Architecture:

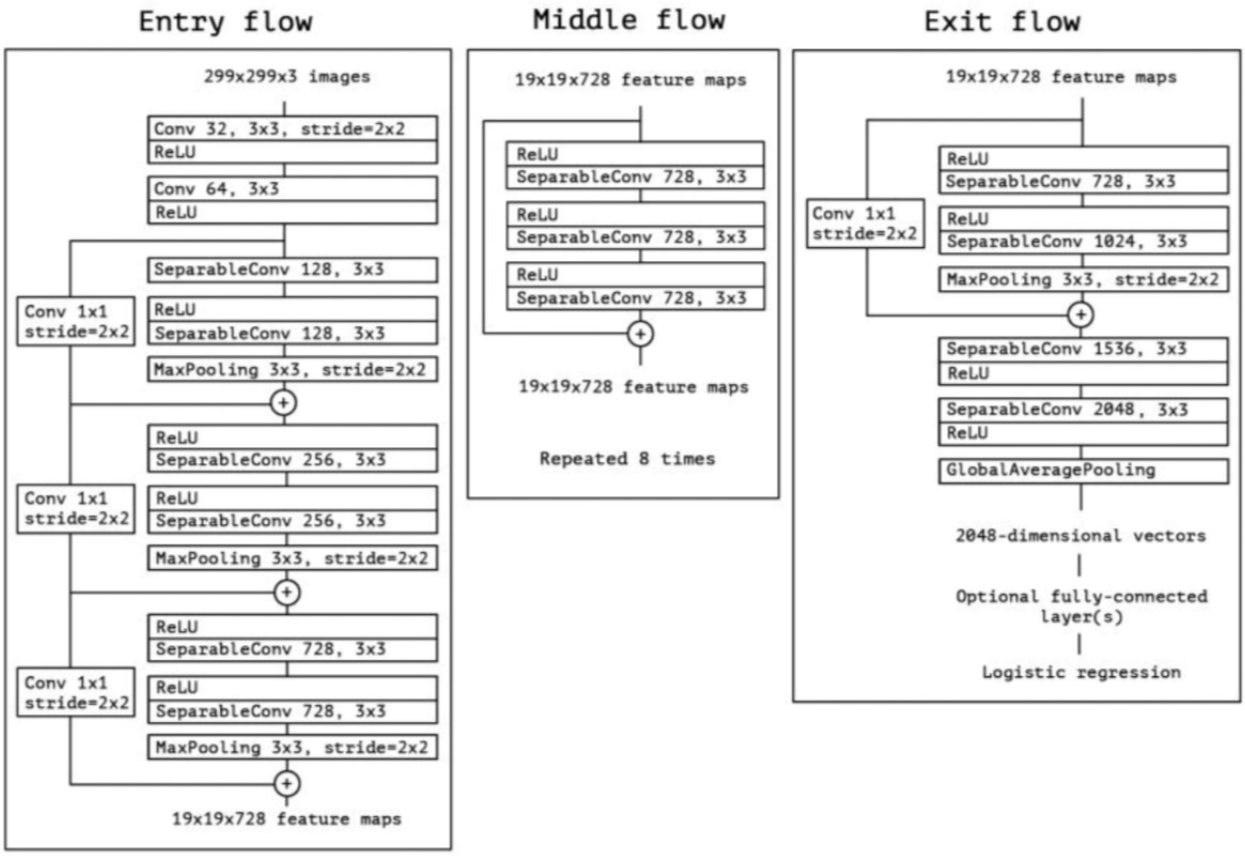
EfficientNet **B0**, Image size (427x427), 2 dense layers (1024 ,256), Average Pooling, dropout rate (0.2), epochs = 15, batch size = 32

Test Accuracy **96**%

3. Xception

Xception is a convolutional neural network architecture introduced in 2017. It's notable for its efficient use of depth-wise separable convolutions. This technique splits the convolution operation into two stages, making it computationally efficient while maintaining expressive power. Xception builds upon the Inception architecture's idea of using multiple kernel sizes in parallel, enhancing its efficiency and performance. It's widely used in various computer vision tasks such as image classification, object detection, and semantic segmentation.

.1. Architecture



.2. Trials

❖**Trial 1:** Image size (300x300), white background,2 dense layers (1024,512), dropout rate (0.4), epochs =15, batch size = 32

Train Accuracy	Val accuracy	Test Accuracy
92%	89%	82%

❖**Trial 2:** Image size(400x400), white background,3 dense layers (1024,512 ,256), dropout rate (0.2), epochs = 15, batch size = 32

Train Accuracy	Val accuracy	Test Accuracy
95%	92%	86%

Final Model Architecture:

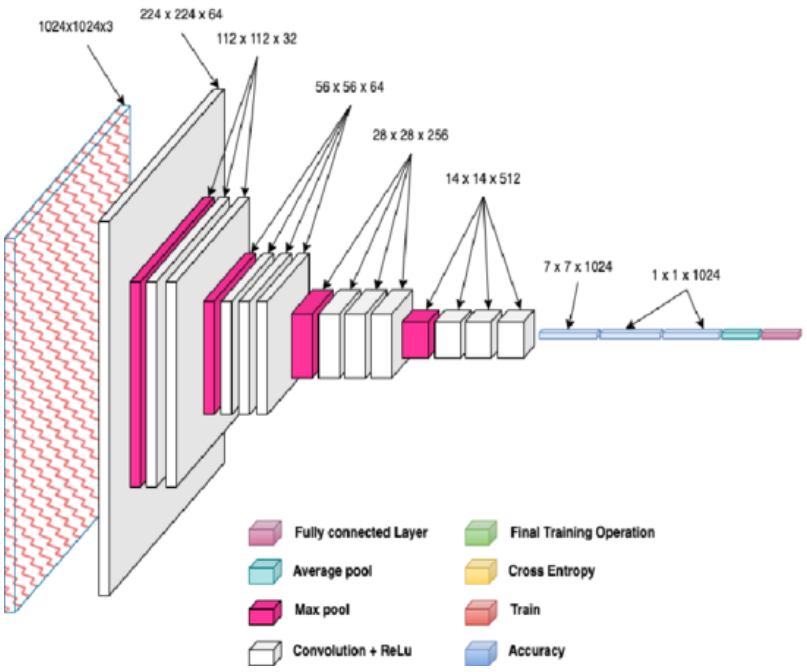
Xception, Image size (400x400), 3 dense layers (1024,512,256), white background, dropout rate (0.2), epochs = 15, batch size = 32

Test Accuracy **86**%

4. MobileNet

MobileNetV1 is a convolutional neural network designed for efficient inference on mobile and embedded devices. It utilizes depthwise separable convolutions to reduce computational cost while maintaining performance. This architecture, introduced by Google in 2017, features depthwise convolution followed by pointwise convolution to extract and combine features efficiently. MobileNetV1 incorporates techniques like batch normalization and ReLU activations for improved training stability. Its adjustable width and depth parameters offer flexibility in balancing model size, latency, and accuracy. Overall, MobileNetV1 is widely used for image classification and other computer vision tasks on resource-constrained devices.

.1. Architecture



.2. Trials

❖**Trial 1:** Image size (224x224), Black background,2 dense layers (1024,512), dropout rate (0.4), epochs =100, batch size = 32, augmentation again, RMS prop optimizer, and unfreeze and train the last 10 layers

Train Accuracy	Val accuracy	Test Accuracy
99%	99%	96%

❖**Trial 2:** Image size(224x224), Black background,2 dense layers (1024,512), dropout rate (0.4) epochs = 50, batch size = 32, augmentation again, RMS prop optimizer, Unfreeze and train the last 3 layers

Train Accuracy	Val accuracy	Test Accuracy
85%	87%	90%

❖**Trial 3:** Image size(224x224), Black background,2 dense layers (1024,512), dropout rate (0.4) epochs = 50, batchsize=32, RMS prop optimizer, Unfreeze and train the last 3 layers, using different augmentation

Train Accuracy	Val accuracy	Test Accuracy
100%	100%	100%

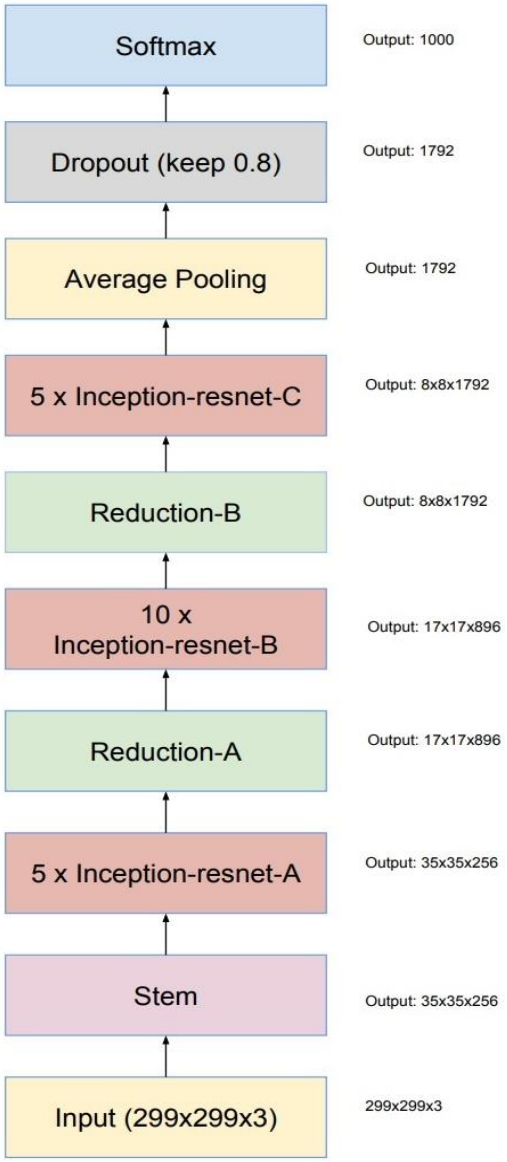
Final Model Architecture:

MobileNetV1, Image size (224x224),2 dense layers (1024,512), Black background, dropout rate (0.4), epochs = 100, batch size = 32
Test Accuracy **100**%

5. Inception ResNet

Inception-ResNet-v2 is a convolutional neural network architecture that merges the Inception and ResNet architectures, combining their strengths to achieve superior performance in various computer vision tasks, particularly image classification and object detection, it combines the depth-wise separable convolution of Inception with the residual connections of ResNet. This hybrid architecture leverages the advantages of both approaches, resulting in a network that is not only highly accurate but also computationally efficient, making it suitable for deployment on resource-constrained devices.

.1. Architecture



.2. Trials

❖**Trial 1:** Image size (224x224), white background,3 dense layers (1024,512,256), Batch Normalization, dropout rate (0.4), epochs =100,
batch size = 32,

Train Accuracy	Val accuracy	Test Accuracy
89%	70%	80%

❖ **Trial 2:** Image size(400x400), white background,3 dense layers (1024,512,256), dropout rate (0.5) epochs = 100, batch size = 32.

Train Accuracy	Val accuracy	Test Accuracy
95%	88%	88%

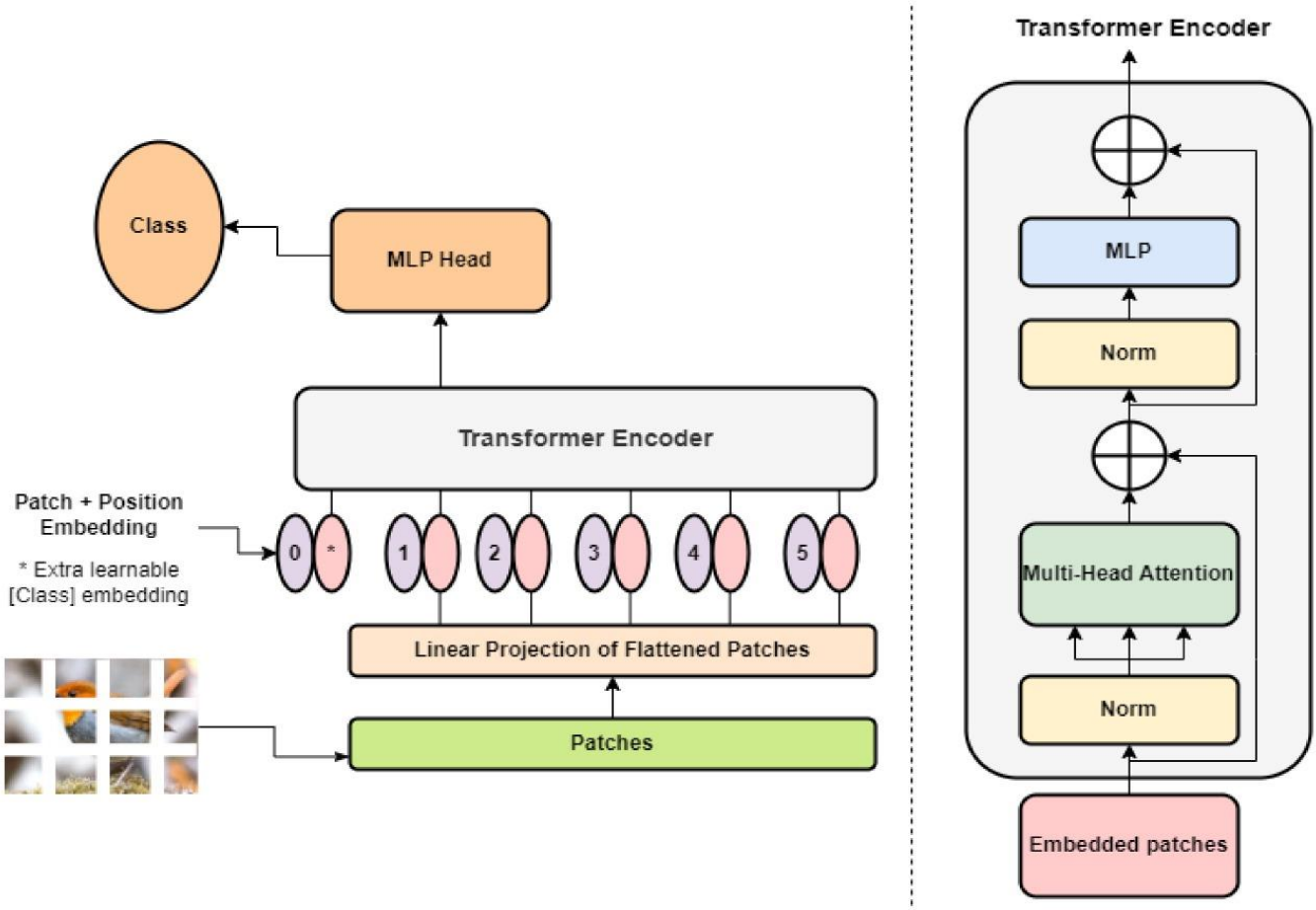
Final Model Architecture:

InceptionResNet, Image size (400x400),3 dense layers (1024,512,256), white background, dropout rate (0.4), epochs = 100, batch size = 32
Test Accuracy **88**%

6. ViT (Vision Transformer)

Transformer architectures as introduced in the “ATTENTION IS ALL YOU NEED” paper have had huge impacts in the NLP domain. But its applications in the Computer Vision domain had been limited. In 2021, a research team at Google introduced the paper “AN IMAGE IS WORTH 16X16 WORDS: TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE (2021)”, which applied the Transformer encoder architecture to the image recognition(classification) task.

.1. Architecture



.2. Trials

- **Trial 1:** Vit base 16, image size (384x384), epochs = 20

Train Accuracy	Val accuracy	Test Accuracy
91%	89%	80%

- **Trial 2:** Vit base 32, image size (384x384), epochs = 30

Train Accuracy	Val accuracy	Test Accuracy
92%	90%	82%

- **Trial 3:** Vit base 32, image size (416x416), epochs = 100

Train Accuracy	Val accuracy	Test Accuracy
95%	93%	88%

Final Model Architecture:

ViT base 32, Image size (416x416), epochs = 100

Test Accuracy **88**%

Conclusion

- Creating CNN architectures and training them from scratch lead to a very high validation loss and very low accuracy.

- Maximum Test Accuracy gained **100**% using Pre-trained MobileNet and **96**% using Inception and EfficientNetB0

- Credits:
    - Mohamed samy
    - Yomna Mohamed
    - Ammar Mohamed
    - Nadine Haitham
    - Mohamed Ashraf
    - Youssef Tamer