VIETNAM GENERAL CONFEDERATION OF LABOUR

**TON DUC THANG UNIVERSITY**

**FACULTY OF INFORMATION TECHNOLOGY**



**PROJECT**

**DEEP LEARNING**

FINAL PROJECT

*Supervisor*: **PhD. LE ANH CUONG**

*Authors*: **LE HUYNH HUYEN TRANG – 520C0156**

**NGUYEN NGOC TU – 520K0231**

**HO CHI MINH CITY, 2023**

VIETNAM GENERAL CONFEDERATION OF LABOUR

**TON DUC THANG UNIVERSITY**

**FACULTY OF INFORMATION TECHNOLOGY**



**PROJECT**

**DEEP LEARNING**

**FINAL PROJECT**

*Supervisor*: **PhD. LE ANH CUONG**

*Authors*: **LE HUYNH HUYEN TRANG – 520C0156**

**NGUYEN NGOC TU – 520K0231**

**HO CHI MINH CITY, 2023**

# ACKNOWLEDGEMENTS

We are really grateful because we managed to complete our report assignment about Deep Learning within the time given. We sincerely thank our lecturer, Mr. Le Anh Cuong for the guidance and encouragement in finishing this assignment and also teaching us in this course.

# PROJECT IS COMPLETED

# AT TON DUC THANG UNIVERSITY

We hereby declare that this is our own project and is under the guidance of Mr. Le Anh Cuong. The research contents and results in this topic are honest and have not been published in any publication before. The data in the tables for analysis, comments and evaluation are collected by the author himself from different sources, clearly stated in the reference section.

In addition, the project also uses a number of comments, assessments as well as data of other authors, other agencies and organizations, with citations and source annotations.

If we find any fraud, we will take full responsibility for the content of our project. Ton Duc Thang University is not related to copyright and copyright violations caused by us (if any).

*Ho Chi Minh City, 8th May , 2023*

*Authors*

*(signature and full name)*

*Le Huynh Huyen Trang*

*Nguyen Ngoc Tu*

# ABSTRACT

# CONTENTS

[**ACKNOWLEDGEMENTS 1**](#_heading=h.gjdgxs)

[**PROJECT IS COMPLETED 2**](#_heading=h.sipven46dy7o)

[**AT TON DUC THANG UNIVERSITY 2**](#_heading=h.u8qs59iz3vw0)

[**ABSTRACT 3**](#_heading=h.30j0zll)

[**CONTENTS 4**](#_heading=h.1fob9te)

[**LIST OF SYMBOLS AND ABBREVIATIONS 5**](#_heading=h.vv7sk4z20de6)

[**LIST OF TABLES, FIGURES, GRAPHS 6**](#_heading=h.3znysh7)

[**CHAPTER 1: PROBLEM DESCRIPTION 6**](#_heading=h.iai70ft8g5yj)

[**CHAPTER 2: THEORETICAL BASIS 9**](#_heading=h.ml4do87ncwbt)

[2.1. Introduction 9](#_heading=h.lwja13lsdsy3)

[2.1.1. Introduction of AI 9](#_heading=h.g01ab5h1y1cw)

[2.1.2. Classification problem 9](#_heading=h.w13k9rve6kwx)

[2.2. Convolutional Neural Network (CNN) 10](#_heading=h.l95ruwhunluq)

[2.2.1. Background 11](#_heading=h.2et92p0)

[2.2.2. Convolutional Neural Network 12](#_heading=h.tyjcwt)

[2.3. Mobilenet V2 16](#_heading=h.cayela419007)

[2.3.1. Introduction of MobileNet 16](#_heading=h.rz1gtkfhl35k)

[2.3.2. Mobilenet V2 18](#_heading=h.j31x69mievop)

[2.4. Vision Transformer (ViT) 21](#_heading=h.n6ghdp34jvow)

[**CHAPTER 3: EXPERIMENTS 25**](#_heading=h.v7v8bac6gxoe)

[3.1. Dataset 26](#_heading=h.akhga65d9xsv)

[3.2. Experiments, results and comparisons 27](#_heading=h.u8b9q4pmafj0)

[**REFERENCES 30**](#_heading=h.c0omz4z1ns7p)

# LIST OF SYMBOLS AND ABBREVIATIONS

# LIST OF TABLES, FIGURES, GRAPHS

**LIST OF FIGURES**

[Fig 1: MRI scans of normal brain (left) and cancer (right) 4](#_heading=h.b21qov8s8w9f)

[Fig 2: AI applications 6](#_heading=h.sc1hsp9dx8ge)

[Fig 3: Examples of classification problems 7](#_heading=h.xqkcnf9nfelq)

[Fig 4: Convolution operation 8](#_heading=h.ixxw4yj0qd1p)

[Fig 5: Convolution operation with stride = 1 and padding =1 9](#_heading=h.svupb3rt7zbz)

[Fig 6: CNN recognize the handwriting of MNIST dataset 10](#_heading=h.iep4gfa9m1l5)

[Fig 7: Convolution Layer 11](#_heading=h.cztall78j2md)

[Fig 8: Max Pooling Layer 11](#_heading=h.yzwfxdkygbh3)

[Fig 9: Fully Connected Layer 12](#_heading=h.xfc3nf1ris2c)

[Fig 10: CNN architecture 13](#_heading=h.lblg2n619v0p)

[Fig 11: Depthwise Separable Convolution block 14](#_heading=h.pb45xwf4ehuc)

[Fig 12: MobileNet V2 architecture 16](#_heading=h.cetnb5mzxrwp)

[Fig 13: Transformer architecture 19](#_heading=h.x7ulpzuxuomk)

[Fig 14: Vision Transformer architecture 20](#_heading=h.ablab3rqbhj2)

[Fig 15: Dataset on Kaggle 23](#_heading=h.um0fm9p2047v)

[Fig 16: training process of ViT (left) and MobileNetv2 (right) models 25](#_heading=h.s7qbmdno04t0)

**LIST OF TABLES, GRAPHS**

# CHAPTER 1: PROBLEM DESCRIPTION

The problem of brain tumor classification is an important problem in the medical field, it is given with the purpose of detecting tumors in the patient's brain through imaging impurities of the brain. This problem requires accuracy and high sensitivity to make accurate conclusions about the patient's health status.

Brain cancer is one of the most dangerous diseases to human health, so early detection and accurate diagnosis are very important for early and quick treatment. However, diagnosing brain cancer through brain impurities is a difficult job, requiring the expertise and experience of medical professionals.

Therefore, the application of machine learning models to the brain tumor classification problem is a new and effective approach in medicine. These models can learn from image data to classify tumor types with high accuracy. The use of machine learning models can help medical professionals make faster and more accurate diagnostic decisions, leading to earlier and more effective treatment for patients.

A close-up of the brain

Description automatically generated with low confidence

#### Fig 1: MRI scans of normal brain (left) and cancer (right)

The dataset used is a dataset on brain tumor detection published on Kaggle [[1]](#footnote-0) with 3000 MRI [[2]](#footnote-1)  images of the brain with different sizes and resolutions divided into 2 categories: normal brain and brain containing tumor. To solve this problem, we will use two models: Vision Transformer and MobileNet V2. Vision Transformer is a model designed for image processing using the Transformer architecture, which has been successful in natural language processing problems. Meanwhile, MobileNet V2 is a model designed for fast and efficient processing of image problems.

The steps involved in brain cancer classification problem include:

* Prepare data
* Data preprocessing: Divide the data into training and test sets, and normalize the sizes of images to include in the training model.
* Training models: use the training set to train 2 models: Vision Transformer and MobileNet V2.
* Evaluation: Use the test set to evaluate the accuracy of 2 models
* Compare the performance of the models: Compare the performance of Vision Transformer and MobileNet V2 models to determine which model has the highest accuracy for the brain cancer classification problem.

The classification of brain cancer can help to better diagnose and treat the disease, helping medical professionals make the right diagnosis and quickly. In addition, the use of models such as Vision Transformer and MobileNet V2 for this problem also contributes to the development of the field of machine learning in the medical field, making medical image processing and analysis more efficient, saving time and cost for disease diagnosis and treatment.

# CHAPTER 2: THEORETICAL BASIS

## Introduction

### Introduction of AI

AI (Artificial Intelligence) is the field of research and development of methods and technologies to create intelligent systems and automate tasks that were previously only able to be performed by humans.

Applications of Artificial Intelligence have appeared in many different fields, from applications in daily life such as virtual assistants, self-driving cars, voice recognition to applications in industries such as medical diagnosis, automated manufacturing, and financial risk management.

Diagram

Description automatically generated

#### Fig 2: AI applications

AI is one of the most important technology trends of today and in the future. The proper development and use of artificial intelligence will bring a lot of benefits to society and people.

### Classification problem

The classification problem is one of the important problems in the field of machine learning, often used to determine the class or label of data objects. In this problem, the goal is to build a model to classify objects into one of the known classes.

Examples of classification problems:

* Classify emails as spam or not spam
* Classify an image containing a dog, cat, or bird subjects
* Categorize articles as sports, entertainment, business or politics

Diagram

Description automatically generated

#### Fig 3: Examples of classification problems

The classification problem plays an important role and is becoming more and more popular in many different fields today. Applications of classification problems play an important role in solving many practical problems, from data analysis, forecasting, to recommender, control, monitoring and management systems.

To evaluate the performance of a classification problem, we have many ways such as accuracy, precision, recall, f1-score,... In which, accuracy is used most commonly. Accuracy: used to evaluate the accuracy of the predicted model compared to the actual data. This scale calculates the ratio between the amount of correctly predicted data and the total amount of data.

## Convolutional Neural Network (CNN)

A Convolutional Neural Network (CNN) is a type of deep learning algorithm that is particularly well-suited for image recognition and processing tasks. It is made up of multiple layers, including convolutional layers, pooling layers, and fully connected layers.

The convolutional layers are the key component of a CNN, where filters are applied to the input image to extract features such as edges, textures, and shapes. The output of the convolutional layers is then passed through pooling layers, which are used to down-sample the feature maps, reducing the spatial dimensions while retaining the most important information. The output of the pooling layers is then passed through one or more fully connected layers, which are used to make a prediction or classify the image.

### Background

**Convolution operation**: the process of moving the sliding window of the same size as the kernel from top to bottom, from left to right for the input matrix, and at the same time performing the summation of the multiplications of the corresponding elements. (dot product) and then enter the value in the resulting matrix. The dot product is mathematically represented as follows:

A picture containing company name

Description automatically generated

Diagram

Description automatically generated

#### Fig 4: Convolution operation

In the above figure, the matrix I on the left is the representation of the input image, each value of the matrix corresponds to a pixel with 0 corresponding to black, 1 corresponding to white. The matrix K in the middle is called the sliding window (we can call K the kernel, filter or feature detector). The matrix K has size k ∗ k where k are odd numbers, usually 3 ∗ 3 or 5 ∗ 5. Usually the size of the sliding window is chosen based on the size of the input matrix and the position of the Convolution layer. Finally is the resulting matrix I ∗ K, the size of which will decrease depending on the size of the sliding window and the Padding technique will be shown below.

**Stride:** the distance between 2 sliding windows when scanning. If stride is 1, then the sliding window will scan 2 cells right next to each other, with stride is 2, the sliding window will scan the first and third cells, skipping the second cell. However, it should be noted that choosing stride is similar to choosing a sliding window size because the larger these values are, the smaller the feature map will be. And also from there we define the concept of padding.

**Padding**: the technique used to keep the original size of the feature map. It does this by "wrapping" the input matrix by cells with the value 0. When the padding is 1, we add a cell with the value 0 around the cell at the outer edge of the input matrix.

Diagram

Description automatically generated

#### Fig 5: Convolution operation with stride = 1 and padding =1

In the above figure, we see that the size of the output matrix is equal to the size of the input matrix. The reason for this is because we have "wrapped" the input matrix by 0 values with a thickness of 1 (or padding = 1).

### Convolutional Neural Network

There are three types of layers that make up the CNN which are the convolutional layers, pooling layers, and fully-connected (FC) layers. When these layers are stacked, a CNN architecture will be formed.

Diagram

Description automatically generated

#### Fig 6: CNN recognize the handwriting of MNIST dataset

In the figure above, the CNN architecture consists of 5 layers: input layer, convolution layer, pooling layer and 2 fully connected layers.

**Convolution Layer:** This is the most important layer and also the difference of CNN from other classical models. Accordingly, instead of connecting all the elements of the input matrix, we use a sliding window to slide through the input matrix by convolution. The goal of the convolution layer is to extract local features from the input image. A CNN network is a collection of many convolutional layers stacked on top of each other with the aim of extracting features from visual to abstract. Normally, in the first convolutional layer, the convolutional layer will find features such as edges, corners, colors, etc. of the image. There are two types of convolutional layer results, the first is that the output matrix size is reduced compared to the input matrix (this type is called Valid Padding). And the second type is the matrix size equal or increased compared to the input matrix (this type is called Same Padding).

A picture containing chart

Description automatically generated

#### Fig 7: Convolution Layer

**Pooling Layer:** In most cases, a Convolutional Layer is followed by a Pooling Layer. The primary aim of this layer is to decrease the size of the convolved feature map to reduce the computational costs. This is performed by decreasing the connections between layers and independently operates on each feature map. Depending upon the method used, there are several types of Pooling operations. It basically summarizes the features generated by a convolution layer. In Max Pooling, the largest element is taken from the feature map. Average Pooling calculates the average of the elements in a predefined size Image section. The total sum of the elements in the predefined section is computed in Sum Pooling. The Pooling Layer usually serves as a bridge between the Convolutional Layer and the FC Layer.

A screenshot of a computer

Description automatically generated with low confidence

#### Fig 8: Max Pooling Layer

Formula to calculate the size of the output matrix after the pooling layer:

Text

Description automatically generated

**Fully Connected Layer**: is one of the layer types commonly used in deep neural networks. This layer is used to connect inputs to outputs in a deep neural network.

Chart, diagram

Description automatically generated

#### Fig 9: Fully Connected Layer

The FC layer performs matrix multiplication between the input and the weight matrix of the class, then adds the bias vector and applies an activation function to compute the output. For example, if the FC layer has n inputs and m outputs, then the weight matrix will have size n x m, and the bias vector will have size m. The activation function commonly used in the FC layer is ReLU or Sigmoid.

The FC layer is often used in neural networks to learn features of data and perform classification, prediction, or prediction. For example, in a CNN neural network, the FC layer is used to connect the convolutional layers and the output layer to perform image classification. In an RNN neural network, the FC layer is used to perform character or time series prediction.

The main limitation of the fully connected layer is that it includes a lot of parameters that need to be calculated, which will complicate the training process. So we always try to eliminate the number of nodes in the class as well as the connections of that node. Dropped nodes and connections can be satisfied by using dropout technique.

**Dropout**: Usually, when all the features are connected to the FC layer, it can cause overfitting in the training dataset. Overfitting occurs when a particular model works so well on the training data causing a negative impact in the model’s performance when used on new data.

To overcome this problem, a dropout layer is utilized wherein a few neurons are dropped from the neural network during the training process resulting in reduced size of the model. On passing a dropout of 0.3, 30% of the nodes are dropped out randomly from the neural network.

Dropout results in improving the performance of a machine learning model as it prevents overfitting by making the network simpler. It drops neurons from the neural networks during training.

From these layers, we can build a complete CNN network:

Diagram

Description automatically generated

#### Fig 10: CNN architecture

## Mobilenet V2

### Introduction of MobileNet

MobileNet is a neural network (CNN) architecture designed to work efficiently on mobile devices with limited computing resources. MobileNet uses the main building blocks of Depthwise Separable Convolution to reduce computation costs.

MobileNet was developed by Google and was first introduced in 2017. It is widely used in image recognition, object classification, face recognition and speech signal classification on mobile devices. mobile device. By using MobileNet, mobile applications can process image and audio-related tasks faster and more efficiently, which saves computing resources and saves battery.

Depthwise Separable Convolution block is a building block used in the MobileNet neural network architecture to reduce the computational cost and number of parameters of the network. This block consists of two phases:

Diagram

Description automatically generated

#### Fig 11: Depthwise Separable Convolution block

* **Depthwise Convolution**: Apply individual filters to each input channel. These filters are smaller in size than those in conventional Convolution, because they only operate on a single input channel rather than on the entire input. This reduces the number of parameters required for the network.
* **Pointwise Convolution**: Use a regular Convolution layer to combine the input channels. The process closely resembles using a Fully Connected (FC) layer in traditional neural networks. However, instead of connecting all input units to all output units, we only connect the input channels to the output channels. This reduces the number of parameters required for the network.

So the Depthwise Separable Convolution block is a combination of Depthwise Convolution and Pointwise Convolution. These blocks are used as an alternative to conventional Convolution layers in MobileNet to reduce the computational cost and number of network parameters.

MobileNet comes in different versions, numbered according to the depth of the network. For example, MobileNet V1 has 28 million parameters and MobileNet V2 has 3.4 million parameters.

**Advantages:**

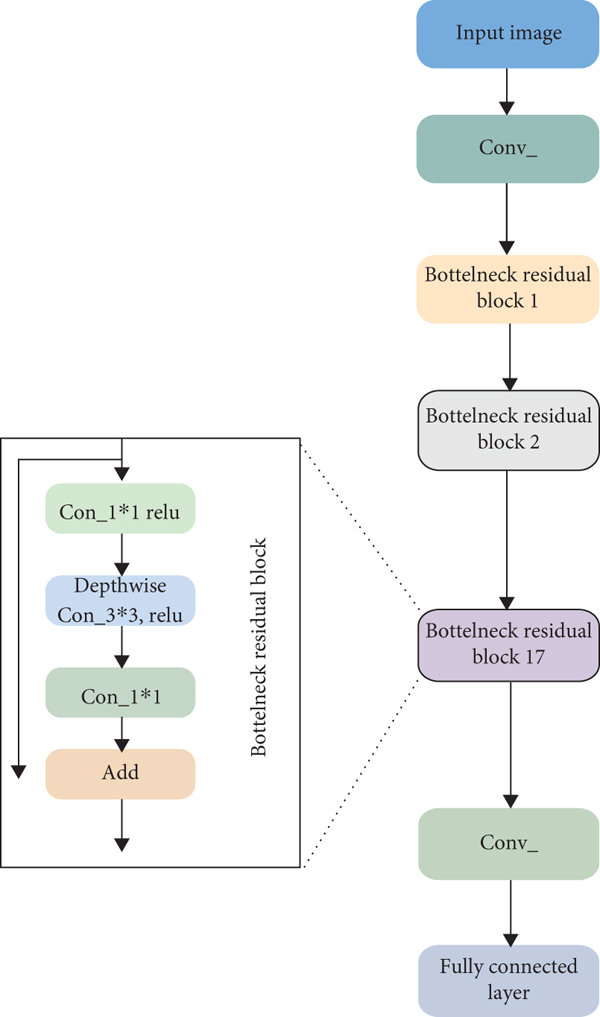
* Compute Efficiency: MobileNet drastically reduces compute costs, making the network efficient on mobile devices with limited compute resources.
* Low parameter count: MobileNet has a low parameter count, making the network easy to deploy on mobile devices.
* High accuracy: MobileNet achieves high accuracy in image recognition, object classification and face recognition tasks.

**Disadvantage:**

* The complexity of the network is lower than that of other deep networks, but it is not capable of learning complex features like deeper networks.
* Since MobileNet uses Depthwise Separable Convolution building blocks, it tends to reduce the complexity of the network by reducing the number of parameters. However, this can lead to the loss of important information during training.
* MobileNet still needs to be updated and improved to improve accuracy and processing speed in image recognition, object classification and face recognition tasks.
* If high resolution images are to be processed, MobileNet may not be suitable because the Depthwise Separable Convolution block cannot perform high resolution convolutions.

### Mobilenet V2

MobileNet V2 is an improved version of the original MobileNet architecture, introduced by author Mark Sandler and colleagues from Google Research in 2018. This release focuses on improving network performance, including increasing computation speed and accuracy.



#### Fig 12: MobileNet V2 architecture

The MobileNet V2 architecture can be divided into three major blocks: input blocks, BottleNeck blocks, and connection blocks.

**Input blocks:**

* The input block of MobileNet V2 is a Convolution layer with 32 filters of size 3x3, using stride=2 to reduce the size of the input.
* Then there is a BatchNormalization layer and an Activation layer .

**BottleNeck blocks:**

The BottleNeck block is a major building block in the MobileNet v2 network architecture, used to increase the depth and reduce the number of parameters of the network. The BottleNeck block consists of three layers: Depthwise Convolution, Pointwise Convolution, and Linear Projection.

* Depthwise Convolution: This layer uses Depthwise Separable Convolution to compute features. It reduces the computational cost and the number of parameters by applying an individual filter to each input channel.
* Pointwise Convolution: This layer uses a regular Convolution layer to combine the features computed by the Depthwise Convolution layer. This helps the network learn some complex relationships between features.
* Linear Projection: This layer is used to reduce the depth of the network. It uses a 1x1 Convolution layer with as many filters as the desired number of output channels. This layer is often used when the number of input and output channels are different.

BottleNeck helps to reduce the number of parameters of the network by reducing the depth of the network and using conventional Convolution layers to combine features. It also helps to increase the computational efficiency of the network by using Depthwise Separable Convolution to compute the features.

**Connection blocks:**

* The connector block includes a Convolution layer to generate the output.
* Before being fed to the Convolution layer, the input will be passed through a GlobalAveragePooling layer to calculate the average of the feature values in each channel.
* Then, a Fully Connected layer with the number of layers corresponding to the number of output layers of the problem to be solved.
* Finally, is an Activation layer to give the final output of the network.

Overall, the MobileNet V2 architecture is designed to optimize the accuracy and computational speed of the network, while reducing the number of parameters to avoid overfitting.

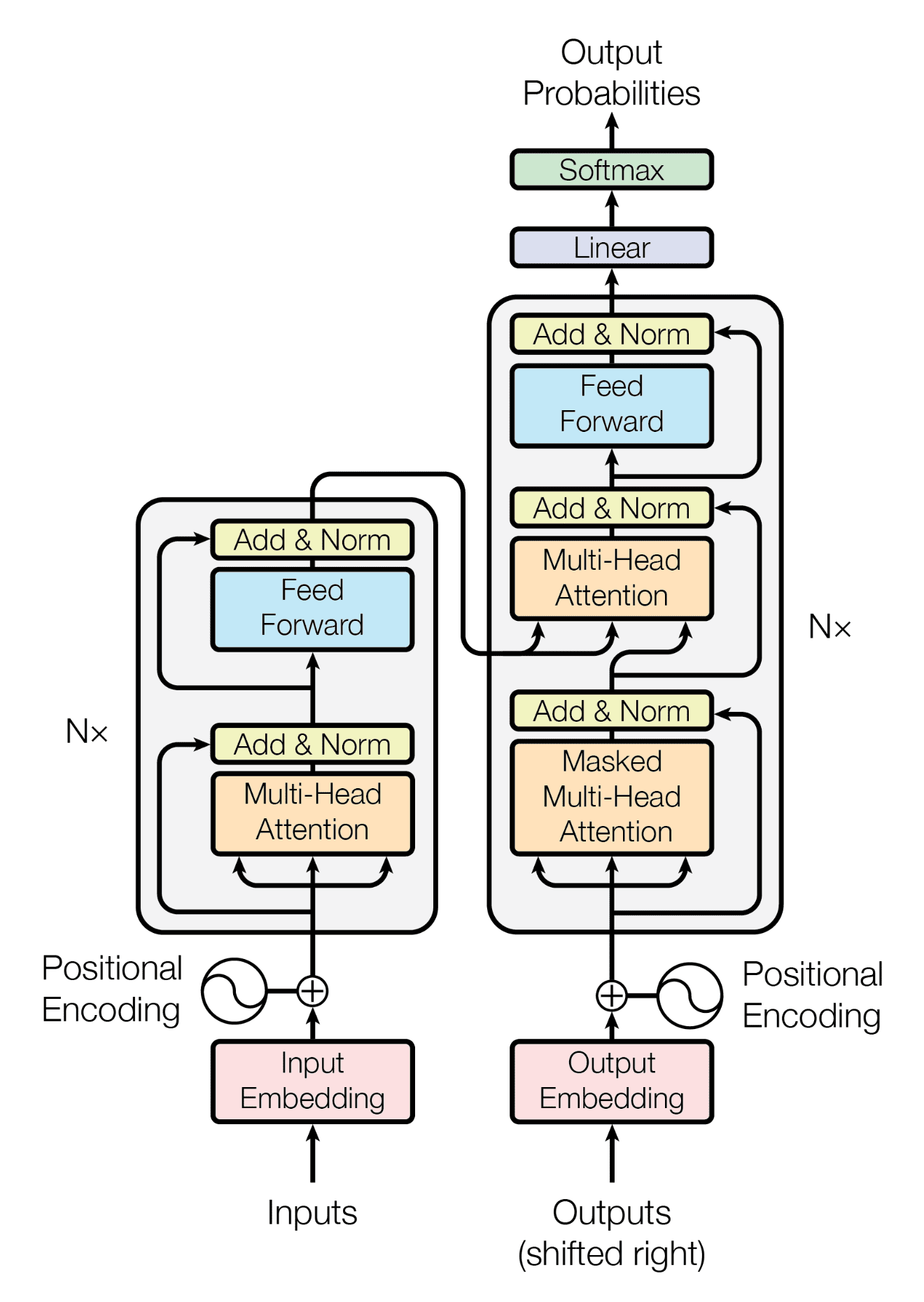
**Ưu điểm của MobileNet V2 so với MobileNet V1:**

* Higher performance: MobileNet V2 has better performance than MobileNet V1, especially on ImageNet dataset. With a 1% increase in average accuracy compared to MobileNet V1.
* Lower parameter count: MobileNet V2 uses BottleNeck blocks to reduce the number of parameters. While MobileNet V1 uses Depthwise Convolution layers directly to reduce the number of parameters. Therefore, MobileNet V2 has a lower number of parameters compared to MobileNet V1.
* Faster computation speed: MobileNet V2 has better performance and less number of parameters than MobileNet V1, so it has faster computation speed and less power consumption during training and forecast.
* Better generalization: MobileNet V2 has better generalizability than MobileNet V1, i.e. it has good predictive ability on new data sets for which the model has not been trained before.
* High flexibility: MobileNet V2 is highly customizable, allowing users to fine-tune their network architecture to fit the needs of specific applications.

## Vision Transformer (ViT)

The Vision Transformer (ViT) model architecture was introduced in a research paper published as a conference paper at ICLR 2021 titled “An Image is Worth 16\*16 Words: Transformers for Image Recognition at Scale”. It was developed and published by Neil Houlsby, Alexey Dosovitskiy, and 10 more authors of the Google Research Brain Team.

Vision Transformer (ViT) architecture is a neural network architecture used for image processing. What's special about this architecture is that it uses the Transformer architecture, which has succeeded in natural language processing, to process the features of the image.



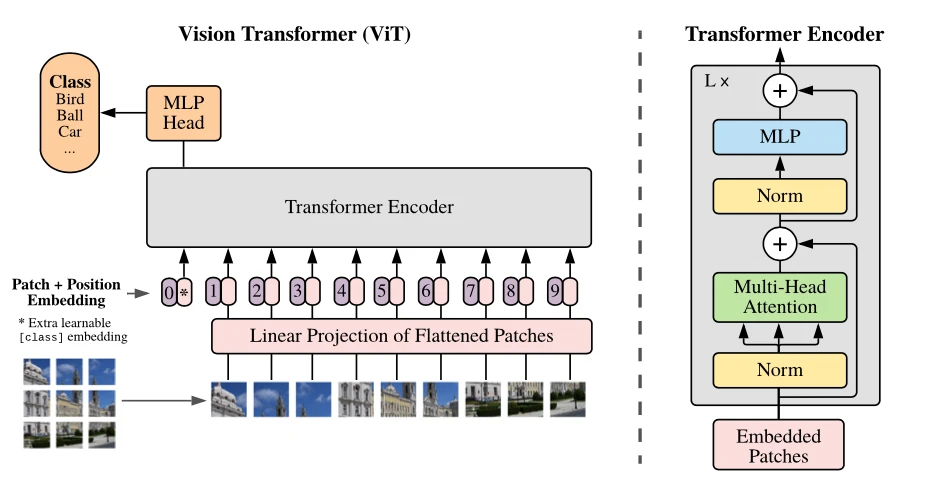
#### Fig 13: Transformer architecture

The Transformer architecture uses Attention mechanism, allowing the model to focus on important parts of the input data, which improves the model's performance. The difference of Transformer architecture compared to traditional neural network architectures is to use Self-Attention layers to replace traditional neural network layers such as Convolutional or Recurrent. Transformer architecture consists of two main parts, Encoder and Decoder. During training, the Encoder takes text sentences or images as input and extracts their features. Then, the Decoder will base on these features to generate the corresponding outputs.

Traditional architectures in image processing, such as Convolutional Neural Networks (CNNs), use convolutional layers to extract image features and then use fully connected layers. enough to classify images. However, the ViT architecture is completely different, it uses a large number of self-attention layers in the Transformer architecture to extract the features of the image. The ViT architecture breaks down images into blocks, called patches, and then passes them through a linear mapping layer to generate vectors representing each patch. These representative vectors will be used as input to the Transformer architecture to perform the image classification.

The overall structure of the vision transformer architecture consists of the following steps:

* Split an image into patches (fixed sizes)
* Flatten the image patches
* Create lower-dimensional linear embeddings from these flattened image patches
* Include positional embeddings
* Feed the sequence as an input to a state-of-the-art transformer encoder
* Pre-train the ViT model with image labels, which is then fully supervised on a big dataset
* Fine-tune the downstream dataset for image classification



#### Fig 14: Vision Transformer architecture

ViT divides the image into square regions of fixed size, called "patches". Patches are split and passed through a convolution first layer to create a sequence of vectors representing these patches. These vectors are then passed into a number of Transformer classes to learn the relationships between them and determine the classification label for the input image.

Vision Transformers (ViT) is an architecture that uses self-attention mechanisms to process images. The Vision Transformer Architecture consists of a series of transformer blocks. Each transformer block consists of two sub-layers: a multi-head self-attention layer and a feed-forward layer.

The self-attention layer calculates attention weights for each pixel in the image based on its relationship with all other pixels, while the feed-forward layer applies a non-linear transformation to the output of the self-attention layer. The multi-head attention extends this mechanism by allowing the model to attend to different parts of the input sequence simultaneously.

ViT also includes an additional patch embedding layer, which divides the image into fixed-size patches and maps each patch to a high-dimensional vector representation. These patch embeddings are then fed into the transformer blocks for further processing.

The final output of the ViT architecture is a class prediction, obtained by passing the output of the last transformer block through a classification head, which typically consists of a single fully connected layer.

ViT has shown its effectiveness in many image classification problems on large data sets, even achieving the same accuracy as traditional CNN models with more complex architecture.

**Advantages**:

* Ability to learn and synthesize whole image information: ViT uses the Transformer architecture to learn and synthesize information about the entire image, making predictions based on all this information. This greatly improves the accuracy of the model.
* Reusability for a variety of tasks: ViT's architecture is very flexible and can be used for a variety of tasks such as image classification, object detection, keyword guessing, and more. It can be trained on many data sets and used for many different applications.
* Computational efficiency: ViT allows training on large data sets without using GPU with large capacity. This reduces costs and increases computational efficiency.
* Easily resize the input image: With ViT, the input image size can be changed without affecting the accuracy of the model, since it uses scale shift solving during training.
* High performance: ViT has achieved good results in many image classification tasks on popular datasets such as ImageNet and COCO, surpassing many traditional models such as ResNet, Inception.

**Disadvantages**:

* Requires large amounts of data: ViT requires a large amount of data to train well. With small data sets, ViT may not achieve the best performance.
* High computational resource requirements: ViT has many parameters and requires large computational resources for training and prediction.
* Inefficient for large images: ViT is not effective for large images due to the decomposition of the image into small patches. Larger images will require more blocks, resulting in an increase in the size of the input matrix and a larger computational resource requirement.

# CHAPTER 3: EXPERIMENTS

## Dataset

The dataset used in the article is the Brain Tumor Detection 2020 set collected from the Kaggle platform. Indeed, in the description the author stated: “Brain tumor is considered one of the most invasive diseases, common in children and adults. Brain tumors account for 85 to 90% of all Central Nervous System (CNS) tumors. Each year, about 11,700 people are diagnosed with brain tumors. The 5-year survival rate for people with malignant brain tumors or CNS tumors is about 34% for men and 36% for women. Brain tumors are classified as: Benign Tumor, Toxic Tumor, Pituitary Tumor, etc. Correct treatment, planning and correct diagnosis should be taken to improve patient life expectancy.” From there, this dataset was collected. This dataset is imaged using MRI techniques to detect brain tumors. These images will then be grouped by the radiographer.

Graphical user interface, application

Description automatically generated

#### Fig 15: Dataset on Kaggle

This data set includes 3000 images with 1500 images of normal people and 1500 images of people with brain tumors. These images have different sizes and resolutions, so they are very close to reality.

## Experiments, results and comparisons

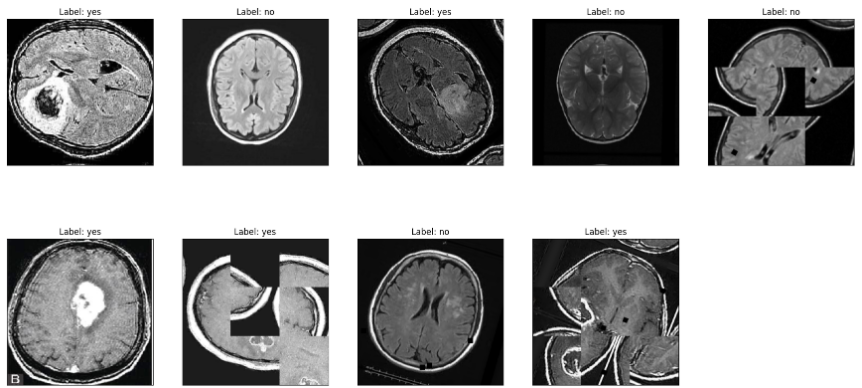
We conducted dataset experiment with two models, Vision Transformer and MobileNetV2. The implementation process includes:

* Prepare dataset: We represent the data to the Dataframe as shown, then read the image dataset

Text

Description automatically generated with medium confidence

The result obtained after reading the image and visualizing the data is that when considering an image, we can also determine the label of that image.



* Then we proceed to configure some parameters to build the model

Text

Description automatically generated

* Finally, we proceed to train the model, the training process is shown in the figure below:

Chart, line chart

Description automatically generated Chart, line chart

Description automatically generated

#### Fig 16: training process of ViT (left) and MobileNetv2 (right) models

Summarizing the model training, we obtain the accuracy of the two models as:

| **Model** | **Accuracy** |
| --- | --- |
| Vision Transformer | 90 % |
| MobileNet v2 | 98 % |

Thus, we see that the MobileNetv2 model has significantly better accuracy than the ViT model. In addition, the training time of the ViT model is nearly 2 times higher than that of the MobileNet v2 model.

The reason for this is due to 2 factors:

* Firstly, the dataset we have is not too large, while the Transformer model requires a large amount of data.
* Second, the input image size is also relatively large (224 \* 224 pixels).

# REFERENCES

1. [Vision Transformers (ViT) in Image Recognition: Full Guide - viso.ai](https://viso.ai/deep-learning/vision-transformer-vit/)
2. [Br35H :: Brain Tumor Detection 2020 | Kaggle](https://www.kaggle.com/datasets/ahmedhamada0/brain-tumor-detection)
3. [The Transformer Attention Mechanism - MachineLearningMastery.com](https://machinelearningmastery.com/the-transformer-attention-mechanism/)
4. [MobileNet V2 Classification Classification Model (roboflow.com)](https://roboflow.com/model/mobilenet-v2-classification)
5. [Vision Transformer (ViT) (huggingface.co)](https://huggingface.co/docs/transformers/model_doc/vit)

1. https://www.kaggle.com/datasets/ahmedhamada0/brain-tumor-detection [↑](#footnote-ref-0)
2. MRI stands for Magnetic Resonance Imaging, also known as magnetic resonance imaging (HACT) in Vietnamese. This is a medical imaging method used to create detailed images of organs and tissues in the human body. To create an MRI image, an MRI machine uses a large magnet to create a strong magnetic field, which combines with radio waves and a computer to create 3D images of organs and tissues in the body. . MRI images provide doctors with detailed information about the structure and function of organs and tissues, especially organs and tissues in the brain. MRI is commonly used in the diagnosis and monitoring of brain diseases, including brain cancer, stroke, and Alzheimer's disease. [↑](#footnote-ref-1)