Masked Face Recognition Based on Transfer learning

Ming Liu

A project report submitted to the Auckland University of Technology
in partial fulfillment of the requirements for the degree of
Master of Computer and Information Sciences (MCIS)

2022

School of Engineering, Computer & Mathematical Sciences

Abstract

Masked face recognition has made great progress in the field of computer vision since the popularity of Covid-19 epidemic in 2020. In countries with severe outbreaks, people are required to wear masks in public. The current face recognition technology, which takes use of the whole face as input data is quite well established. However, when people are use of face masks, which will reduce the accuracy of face recognition. Therefore, we propose a mask wearing recognition method based on MobileNetV2 and solve the problem that many of models cannot be applied to portable devices or mobile terminals. The results indicate that this method has 98.30% accuracy in identifying the masked face. Simultaneously, a higher accuracy is obtained compared to VGG16. This approach has been proven to work well for the practical needs.

Keywords: Computer vision, deep learning, MobileNetV2, masked face recognition, transfer learning

Table of Contents

List of Figures

List of Tables

Attestation of Authorship

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person (except where explicitly defined in the acknowledgments), nor material which to a substantial extent has been submitted for the award of any other degree or diploma of a university or other institution of higher learning.

Signature: Date: 26 May 2022

Acknowledgment

Fisrt of all, I would like to deeply thank my parents for providing me with the financial support to complete this project and to finish my Master's study at Auckland University of Technology (AUT), New Zealand.

I also wish to convey my deep appreciation for my primary supervisor Wei Qi Yan. He has been instrumental in the completion of my project. He provided me with excellent learning resources and helped me with academic problems. I don't think I would have been able to complete the project and achieve my Master degree easily without his patient guidance.

Ming Liu

Auckland, New Zealand

May 2022

Chapter 1
Introduction

This chapter is composed of five parts: The first part introduces the background and motivations, the second part includes the research question, followed by the contributions, objectives, and structure of this report.

1.1 Background and Motivation

Since the outbreak of the COVID-19 epidemic, its rapid spread has posed a serious threat to people's ordinary lives. To prevent cross-infection and the expansion of the epidemic, the World Health Organization (WHO) has proposed that wearing masks properly in public places and maintaining a safe distance is an effective way to prevent its spread (Asadi & Cappa, 2019). However, due to lack of awareness, uncomfortable to wear masks and other reasons, it is difficult to reach the standard of people wearing masks consciously (Wang, 2018). A survey conducted in Japanese society shows that no more than 23.1% of the population wears masks properly (Masaki, 2020). Therefore, it is very important to detect whether masks are worn in public places and whether they are worn in a standard way.

In 2006, the theory of deep learning was proposed, the field of imaging represented by computer vision is developed rapidly. Due to the increased amount of data and computing power today, it makes deep learning work well to its advantage, especially in target detection and image feature extraction (Zhang, 2015). Face recognition algorithms are already popularly applied to our real life. However, the mainstream masked face detection algorithms before the COVID-19 epidemic required tagged samples, the network models require a high computer hardware configuration, with the problem that could not be applied to portable devices or mobile (Jamadar, 2020).

The most general approach to masked face recognition is to consider face recognition as a classification task. The classification network is trained based on the large dataset. The fully connected classification layer in the last layer of the network is removed and the remaining network layer is employed as the face feature extraction network. The output of the last layer of this network is feature data, but now the sizes of deep nets are enormous (Adjabi & Benzaoui, 2020). The existing experiments have shown that the accuracy of mask recognition using deep learning method Retina Face and VGGFace2 is only 94.5% (Aswal, 2020). This accuracy is not optimal for practical applications. Image processing using convolutions is one of the frequently used methods

in the field of computer vision. By tackling less data where the face is marked, transfer learning in deep learning can handle it well which means that these models are already trained by using other data (Gultom, 2018). MobileNetV2 from Keras is one of the representatives of transfer learning techniques (Lee & Lum, 2020). The MobileNet model was firstly proposed in 2017 that is a lightweight convolutional neural network designed for mobile devices. After gradual development, optimization, and update iterations, MobileNetV2 was presented in 2019. This lightweight network transfers the mask wearing recognition problem to a classification problem by using a target detection network. It effectively reduces the number of network model parameters and greatly diminishes the computing time (Andrew, 2017). The parameters of the model size are around 5M (Elmahmudi & Ugail, 2019).

The trend of image recognition is to use deeper layers in the training model, which results in increased computing resources and reduces the efficiency of the artificial neural network (ANN). The Vision Transformer (ViT) has proven to be a faster model for big data processing, and the operation principle of ViT we locate the image to the attention mechanism according to the attention mechanism set forth in the paper Attention is all you need (Ashish, 2017) and classify the category through the multi-layer perceptron.

Transfer learning is harnessed by us to simplify ANN training. Transfer learning uses weights that have been previously trained in a large dataset, only the untouched data set is finely adjusted, so that the new model created by the specific gravity has higher accuracy (Barman, 2019). The weight of the transfer learning training is generally based on a large dataset like ImageNet, consisting of more than ten thousand of categories which help to learn with higher accuracy in training and improve the generalization of the model.

For visual object recognition of whether a mask is worn or not, we train the model using two separate datasets from Kaggle having the same faces with and without face masks. We take use of OpenCV framework in addition to image processing methods. We obtained a real-time detection accuracy 98.3% in our results. This accuracy is 2% higher than the same test conducted in 2020 (Hu & Ge, 2020). We used the Masked Face-Net

dataset (Adnane, 2021) to train on the correct wearing of masks, which contains 137,016 mask images and consists of four categories: mask, mask chin, mask mouth chin and mask mouth nose. This dataset collected by the Flickr-Faces-HQ dataset has more age and ethnic distinctions than other datasets (Karras & Laine, 2018).

1.2 Research Questions

As we mentioned, the purpose of this study was to investigate the use of MobileNetV2 and transformer on mask wear and whether it is worn properly. We also consideration how to improve its accuracy. Therefore, the questions studied in this report are,

(1) *What are the advantages of ANN for classifying images and extracting features?*

(2) *Why the attention mechanism used in ViT has better performance than other model architectures?*

The core purpose of this study is to use deep learning to identify mask wear. Therefore, we need to choose the best model to obtain a higher recognition accuracy. In the study of use mask properly, we need to evaluate several models to get the best results.

1.3 Contributions

The core of this project report is to classify and capture image features through deep learning and transformer technology to obtain high-precision real-time recognition. Compared to traditional models it has better generalization. By the end of this report, we are able to: (1) Find deep learning models with optimal performance; (2) ANN is used for classification recognition of mask images; (3) improve accuracy using ANN weights that have been trained with large datasets; (4) analysis of the accuracy of different ViT models in various scenarios. In addition, we compare the main models in this experiment with ResNet 50 and discuss their advantages and shortcomings in training, validation, and testing.

1.4 Objectives of This Report

Firstly, we introduce the use of MobileNetV2 for real-time applications and compare it with traditional deep learning methods. Secondly, the latest transformer model is proposed according to the Paper Attention is all you need. We applied a Gaussian error Linear Units (GeLU) with 20 training epochs of hyperparameters and a stochastic gradient descent (SGD) optimizer, to training on three ViT model architectures which is ViT Base/16, ViT large/14 and ViT Huge/14. We found the best classification was using ViT Huge/14.

1.5 Structure of This Report

In Chapter 2, we will cover the literature review discussing the current research results on CNN mask detection with transfer learning and examine the development of MobileNetV2. In addition, we will discuss the different accuracy results obtained by training with various datasets.

In Chapter 3, we will introduce the research methods. Focus on the MobileNetV2 and transformer structure. Information of dataset we use, and data augmentation also present in this section.

In Chapter 4, we will analyze our results of each ViT model. Also discuss the training validation and testing accuracy under various improvements. Our experiment data, some details that we still need to be improved are all mentioned in this section.

In Chapter 5, we will summarize and analyze the experimental results.

In Chapter 6, we will draw our conclusion and future work.

Chapter 2
Literature Review

In this section, we highlight the literature on convolutional neural network (CNN) research and discuss past research work in the field of human face recognition.

2.1 Introduction

With the increasing of global security problems, intelligent surveillance is gaining its attention from the public. The demands of surveillance for public security are soaring, such as security in banks, shopping malls, airports and markets, etc. Meanwhile, a growing number of residents are paying their close attention to privacy protection of their homes.

2.2 Face recognition Detection Model

As an important task in the field of computer vision, masked face recognition has been modeled for a long time. The studies on mask testing have been increasing since 2016, especially after COVID-19 epidemic. Since in the beginning, we cannot collect all face image data for face recognition algorithms, the face recognition task is practiced on the open set (Peng, 2021). This leads to the algorithmic models that can distinguish unknown features only on a limited dataset.

With the development of masked face recognition algorithms, a series of loss algorithms for faces have emerged. From the initial Softmax-Loss (Agrawal, 2017), Triplet-Loss, Center-Loss to A-Softmax (Liu, 2017), L-Softmax, Arcface-Loss (Deng, 2017), and to the AdaCos proposed in CVPR 2019. AdaCos does not require hyperparameters compared to the loss function and takes use of adaptive scaling parameters to automatically enhance the nets during the training process, showing the advantages of improving the speed of network convergence and making the network more stable, which can significantly improve the accuracy of face recognition.

Video-based masked face recognition has been accomplished and applied to railroad transportation systems. Krishan et al. proposed a face mask detection and normative wear recognition method based on YOLOv3 and YCrCb. YOLOv3 was applied to detect whether the mask is worn or not, the elliptical skin color model of YCrCb is applied to detect skin color in the masked region, and then to determine whether the mask is worn.

The mAP for masked face detection is 89.07% in the experiment, the recognition rate of mask regulation wearing reached 82.48% (Ahuja, 2021). This value is proved to be unsatisfactory in our experiments.

After YOLOv4 was released in 2020. Sharma et al. proposed a mask wearing detection method based on fusion of high and low frequency components of images with YOLOv4 net (Akhil, 2021). The experiments were conducted by web crawlers to build the dataset and manual data annotation, trained by Darknet framework to conduct object detection, the model detection accuracy reached 98.5% after the training, with an average detection speed of 35.2 ms. Compared to YOLOv3, YOLOv4 offers a significant improvement in accuracy. Since YOLOv4 is use of a mish activation function that is smoother than the Leaky ReLU activation function in YOLOv3, the gradient descent is much effective (Jia & Yang, 2018).

Since face recognition models have been easily exposed to sunlight in open-air environments, changes in sunlight and facial expressions can have an impact on algorithm performance. Lahasan et al. conducted a work to address these challenges (Lahasan, 2019). The evaluation is classified into three parts: Occlusion feature extraction, occlusion recognition, and occlusion recovery. The mask is employed as an example of the object of facial expression recognition, grouped it into holistic and part-based approaches. Experimental results show that the local matching method has better performance compared to the reconstruction method in partial-based mechanisms. The combination of local matching methods and optimization based on metaheuristic techniques can increase the stability of marked face recognition. However, it requires a large enough number of facial images for training purpose.

2.3 Convolutional Neural Network

There has been a great deal of research on mask classification during the COVID-19 epidemic, and optimization of mask identification has been a hot topic. An article published by Albert on mask detection using a CNN model with transfer learning (Albert

2021), uses crowd data sets with 13 categories. One of the manual tagging results consists of 3200 images from 500 users. The authors improve the performance of this small dataset by using data augmentation and transfer learning methods, while testing different deep learning architectures such as MobileNet and VGG16. VGG16 has more training data than the MobileNet model, which contains at least 3.5 million parameters, and VGG16, which has 134.4 million parameters. VGG16 uses transfer learning and data augmentation to improve accuracy and obtain the value of 0.834 (Kalenichenko, 2017)

The research work on face mask detection using deep transfer learning with machine learning methods (Mohamed, 2021) uses the Real-World Masked Face Dataset (RMFD), which contains over 80,000 unmasked faces and 5,000 masked faces, all extracted from real-world faces (Karras, 2021). Labeled Faces in the Wild (LFW) composed of 13,000 simulated marked face is applied to the experiment (Kawulok & Celebi, 2008). The hybrid deep transfer learning model used in the experiments using ResNet50 as a feature extractor. The experiments used both decision trees and support vector machines to obtain the excellent performance. 99.46% and 100% accuracy were obtained on the RMFD dataset and LFW dataset, respectively. This research inspired our experiment.

Image super-resolution and classification network architectures are adopted in the experiments to enable transfer learning to achieve 98.7% accuracy (Li, 2020). This is a great inspiration for our next research on transfer learning. Another study annotated the Medical Masks Dataset (MMD), which was divided into three categories: masked, unmasked and not properly masked. RestNet50 and YOLOv2 were taken to extract features from the annotated and non-annotated images, respectively, and it was found that using the annotated images as the dataset for model training had better accuracy than the non-annotated images (Wang, 2012). However, it is difficult to find many annotated images to apply for validation in deep learning studies. This inspired us to use a more diverse dataset which is Masked Face-Net in our experiments.

Chapter 3
Methodology

The main content of this section is to introduce research methods, which we use in our experiment. The chapter mainly covers the MobileNetV2 architecture, transfer learning in deep learning applied to recognition of mask wear. Two methods for enhancing data training are also presented.

3.1 MobileNetV2

MobileNetV2 is a lightweight convolutional neural network designed for embedded or mobile devices. The structure of the network has two types of stride blocks, which have three layers in both blocks as shown in Figure 3.1,

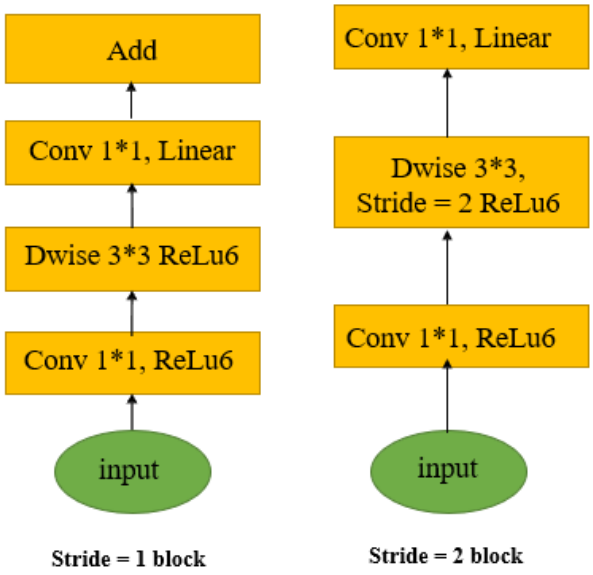


Figure 3.1 MobileNetV2 block

The network mainly takes use of deeply separable convolution. In the first layer, there are several channels expanded by 1×1 convolution with ReLU. This allows for feature extraction in higher dimensions. The 3×3 depth-wise convolutional contains in second layer. In the third layer, the feature dimensionality is reduced by 1×1 convolutional. The activation function is not applied to the final dimensionality reduction layer because using the activation function for low-dimensional features would lose some of the extracted image space features. Deeply separable convolution can greatly reduce the number of model parameters and the amount of computation, which is able to improve the computational speed of the network, the training process can make full use of device resources, the model can also be built in embedded devices and mobiles to achieve the result of real-time recognition. The first layer is stride 1 block, the second layer is depth-separable convolution with residual module as shown in Table 3.1,

Table 3.1 Deeply separable convolution with residual structure

input	operator	output
h*w*c	1*1 conv2d, ReLU	h*w*c
h*w*tc	3*3 dw, ReLU	(h/s)*(w/s)*tc
(h/s)*(w/s)*tc	1*1 conv2d, ReLU	(h/s)*(w/s)*c

From Table 3.1, where h, w, c represents the height of the image, width of the image and the number of channels, respectively.

Since the MobileNetV2 network has stride 2 layers, through using convolutional filtering with a step size of 2, it will cause a large loss of information. In this paper, we consider using the attention optimization module according to Squeeze-and-Excitation Networks as shown in Figure 3.2.

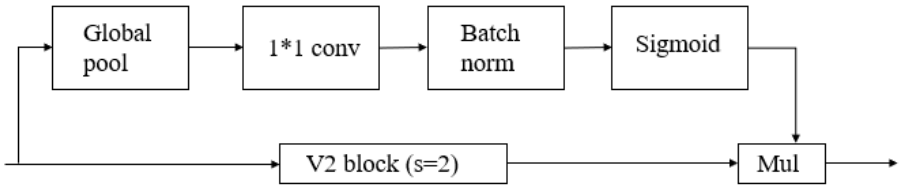


Figure 3.2 Attention optimization module

Pertaining to global average pooling in MobileNetV2 net, the use of an average pooling layer degrades network performance. In a 7×7 feature map, the perceptual domain of the center point and the edge points are the same, the center point includes the complete image while the edge points have only part of the image, so each point has a different weight. However, the average pooling layer represents all pixels with the same weight, it leads to a decrease in performance (Wei, 2019). In this paper, we take advantage of 7×7 size convolution kernels for grouped convolution instead of global average pooling in MobileNetV2 network, which allows the network to learn the weights by itself instead of treating the weights of each point as the same, it makes the network has more generalization ability.

The input image size of the MobileNetV2 lightweight model for face recognition is 224×224, the model consists of four parts. The first part outputs thirty-two 112×112 features maps through 3×3 ordinary convolutions with a step size of 2, padding of 1 and takes a grouped convolution. The second part is composed of six different mobile modules and finally outputs 160 7×7 feature maps. In the third part, the feature dimension is expanded by 1×1 ordinary convolution, the final face feature map is obtained by 1280 1×1 convolutions. In the last part, we implement the classification layer through full connectivity. The network structure as shown in Table 3.2,

Table 3.2 MobileNetV2 architecture

Input	Operator	t	c	n	s
$224^2 * 3$	conv2d	-	32	1	2
$112^2 * 32$	bottleneck	1	16	1	1
$112^2 * 16$	bottleneck	6	24	2	2
$56^2 * 24$	bottleneck	6	32	3	2
$28^2 * 32$	bottleneck	6	64	4	2
$14^2 * 64$	bottleneck	6	96	3	1
$14^2 * 96$	bottleneck	6	160	3	2
$7^2 * 160$	bottleneck	6	320	1	1
$7^2 * 320$	conv2 1*1	-	1280	1	1
$7^2 * 1280$	avgpool 7*7	-	-	1	-
1 *1 * 1280	conv2 1*1	-	k	-	

In Table 3.2, *c* represents the number of channels, *n* indicates number of repetitions of the residual structure, *s* stands for the step size of the first layer of the inverted residual architecture for n repetitions.

Regarding the model to perform well on the test set, we aim to achieve generalization. In this experiment, we take use of stochastic gradient descent (SGD) algorithm for model training. To speed up the convergence and reduce the oscillation in the process of model convergence, the momentum factor is added to the experimental training process in this experiment, the model weight update strategy is shown as eq. (3.1).

$$w = \frac{1}{m}\sum_{j=1}^{m}\frac{\partial L(y^j, f(x^j;w))}{\partial w} \tag{3.1}$$

The parameter is updated as

$$\nu = \beta\nu - \alpha w. \tag{3.2}$$

where β represents the momentum factor which was set to 0.9 in the experiment, α is learning rate and the initial value is set to 0. 01. The learning rate is set to 0. 001, 0. 000 1, and 0. 000 01 for epochs of 40, 50, and 60, respectively.

3.2 Transformer

In recent years, there has been an increasing interest in the study of transformer. It was used not only in natural language processing (NLP), but also in the field of computer vision. The application of transformer is the current trend in different areas of computer vision including audio-visual processing, image classification and face recognition (Deng & Zhong, 2021). The transformer structure proposed by Ashish (2017) is published to overcome the sequence-to-sequence problem and using full attention structure instead of Long Short-Term Memory (LSTM) models (Jurgen & Sepp, 1997). This architecture takes advantage of attention, abandoning the traditional encoder-decoder model that had to be combined with convolutional neural networks (CNN) or recurrent neural networks (RNN) (Katharina, 2017). The main purpose of this approach is to reduce computation and improve parallel efficiency without compromising the final experimental results, and two new Attention mechanisms are proposed, namely scaled dot-product attention and multi-head attention. However, in the field of natural language processing (NLP), considering that the computation of RNN or LSTM is restricted to sequential computation, the relevant algorithms can only compute sequentially from left to right or from right to left, so the problems of gradient loss and long training time will occur. Since the computation of time slice *T* depends on the computation results at moment *T*-1, this limits the parallelism capability of the model. In Figure 3.3, the experiment of the transformer is based on machine translation, which is essentially an encoder-decoder structure.

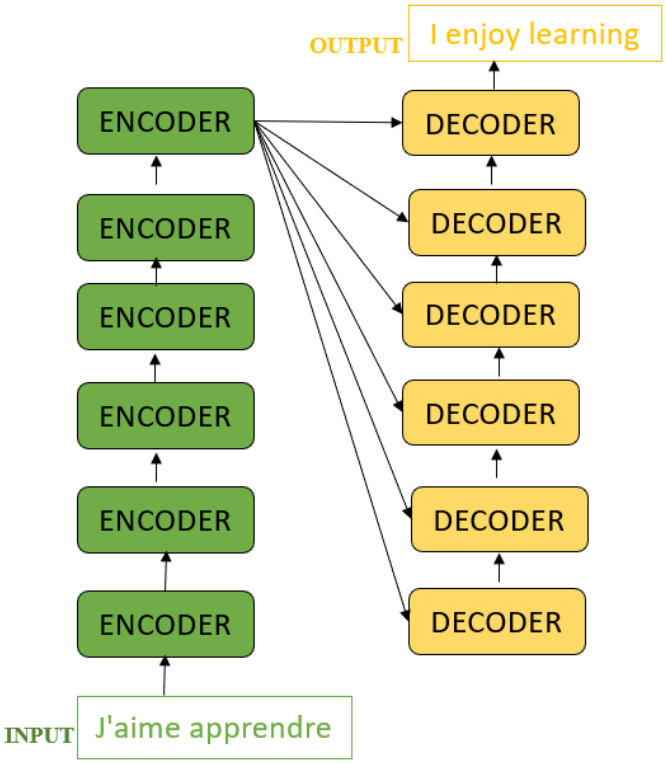


Figure 3.3 Transformer model

The encoder consists of six blocks, each block is composed of a Self-Attention and Feedforward Neural Network (FFNN). The decoder similarly has six decoder blocks. However, each block has an extra Encoder-Decoder Attention as shown in Figure 3.4. As with all generative model, the output of the encoder is used as the input to the decoder.

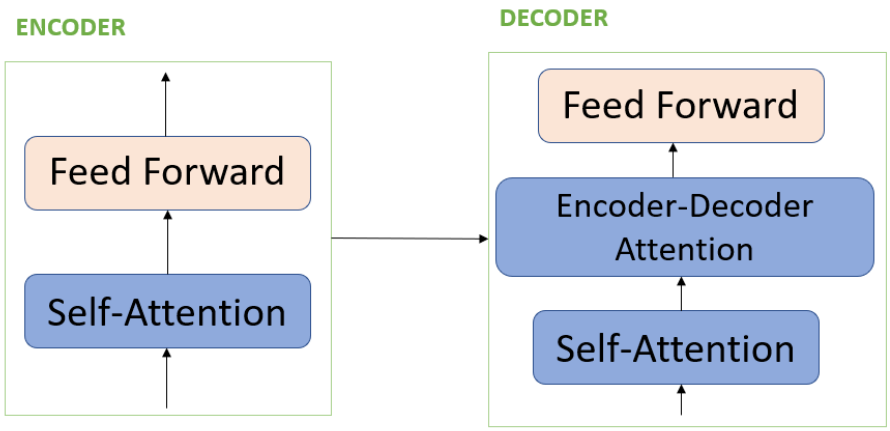


Figure 3.4 Block structure

In Figure 3.5, the encoder consists of *Nx* identical layers. Each layer consists of two sub-layers, a multi-head self-attention mechanism and a fully connected feed-forward network, where each sub-layer adds residual connection and normalization, the output of the sub-layer can be represented:

$$So = N\ (x + (S(x))) \tag{3.3}$$

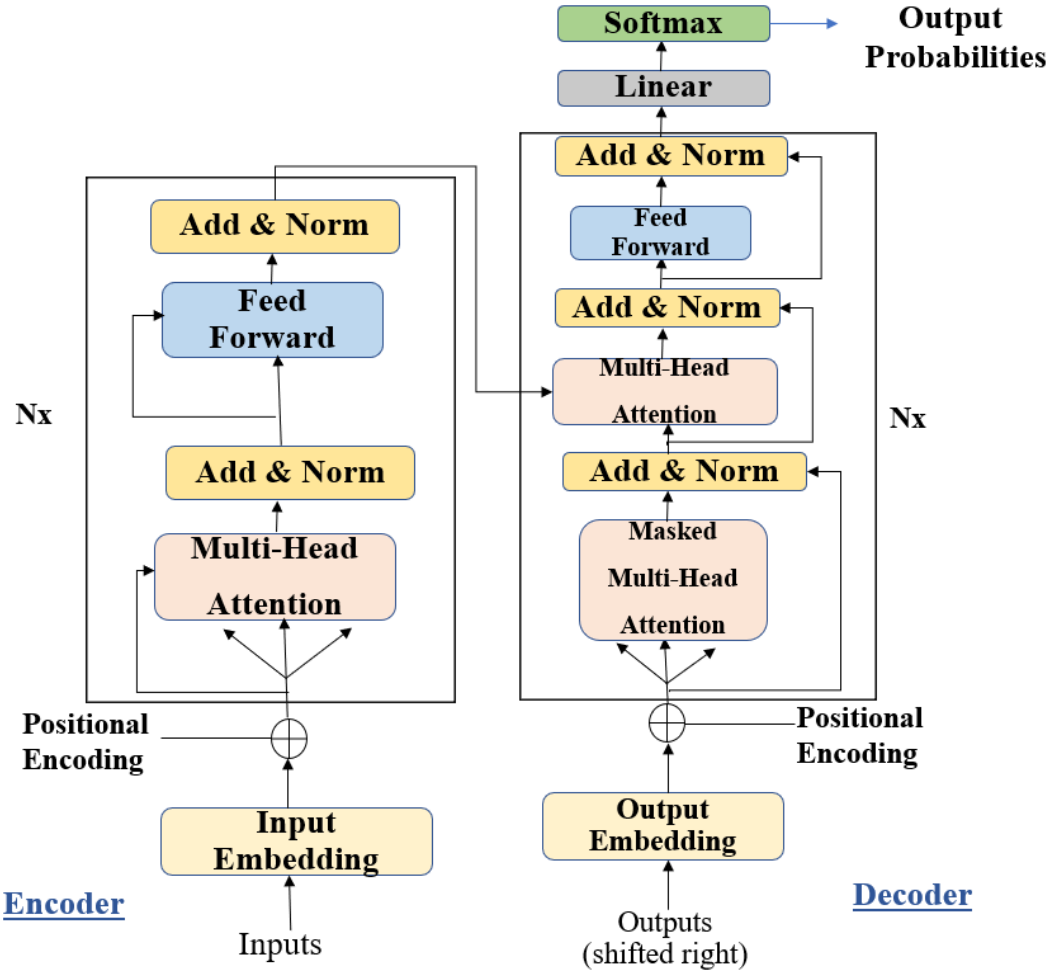


Figure 3.5 Transformer specific model structure

3.3 Multi-Head Self-attention

The core idea of the attention mechanism of the transformer algorithm is to calculate the mutual relation of each word with all the words in the sentence, and then consider the interrelationships reflect to some extent the relevance and importance of the different words in the sentence. Therefore, the importance (weight) of each word can be adjusted to obtain the new expression of each word, which contains not only the word, but also the relationship between other words and the word. As a result, the word vector can be expressed more comprehensively.

Using an attention mechanism that reduces the distance between any two positions in the sequence to a constant, the attention layer can capture a global contact in a single step, since it directly compares the sequence, but the cost is that the computational effort is $N * 2$. However, since it is a pure matrix operation, this computational amount is not serious. In contrast, RNNs require a step-of-step recursion to capture, whereas a CNN needs to expand the perception area by cascading, which is an obvious advantage of the attention layer.

The dividing the model into multiple heads to form multiple subspaces allows the model to focus on different facets of information. Transformer or Bert specific layers have unique functions (Nguyen, 2021), with the bottom layer being more syntactically focused and the top layer being more semantically focused. Most heads in the same layer have the same pattern of attention. In some cases, Multi-head is not necessary, removing some heads will still work well, because in the case of enough heads, these heads already can focus on location information, grammatical information and rare words, some more heads will appear noise. In addition, the difference between the heads decreases as the number of levels increases. The effect of initialization on the variance of the transformer layers and pointed out that the large variance of the bottom header is due to the gradient vanishing problem of the transformer (Ivan, 2019). Therefore, a reasonable initialization can reduce the variance of the underlying headers and improve the effect.

3.4 Vision Transformer

In the previous deep learning experiment, image classification techniques have been using CNN as the main architecture. However, Transformer can be used for image classification research and reduces the calculation time by five times more than the current convolution architecture and obtains higher accuracy (Alexey, 2010).

Vision Transformer (ViT) is different from the usual image classification structure. Figure 3.6 shows the ViT model structure.

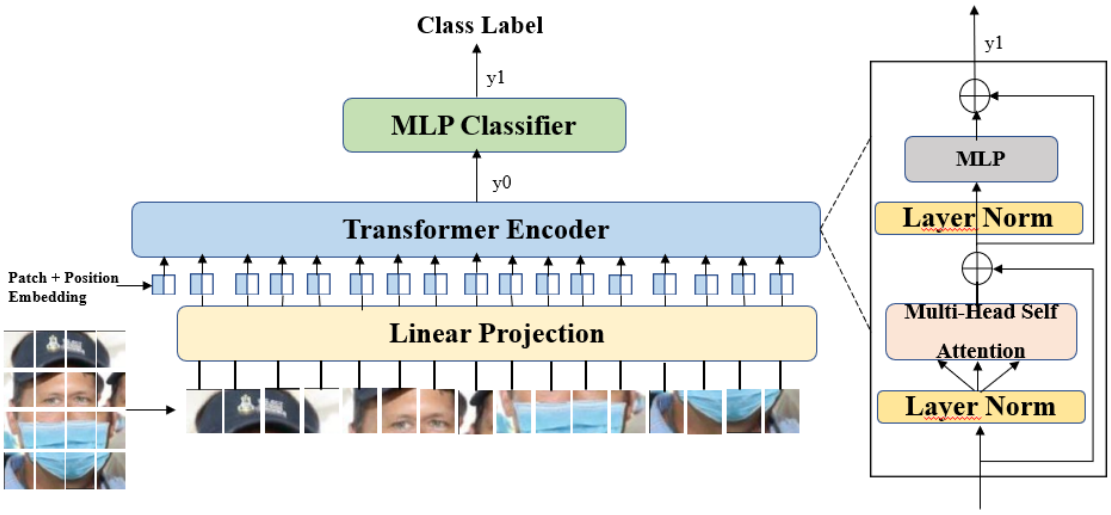


Figure 3.6 Vision Transformer model structure

ViT directly tiled the image into patches, which leads to ignoring the information between each patch. Important facial features under this method are segmented into different tokens, our model modifies the ViT marking method to make the image patches overlap and the information between the patches can be displayed more clearly. The image is divided into several parts according to the number of patches. This approach not only improves the performance of the original ViT, but also adds additional computational cost. This thought was extracted from a face recognition application by Zhong (2021). It is necessary to turns the two-bit digital image into a one-dimensional vector. Eq. (3.4) shows this process,

$$x \in R^{HWC} \rightarrow x \in R^{N\ (P^2C)} \tag{3.4}$$

where H, W represents the resolution of the image, C represents the number of channels. It is converted into $R^{N\ (p2C)}$, where p represents the number of patches, $N = HW/p^2$ and the embedding results will be encoded by the transformer. Such transformers for natural language processing tasks require an embedded input. Jimmy (2016) in his article Layer Normalization distributes the sum of the inputs of a neuron in a mini batch of training by layer normalization process. Compared to batch normalization, normalized input sums by mean and variance have a greater time advantage. Eq. (3.5) shows the key components of capturing an image using Multi-Headed Attention,

$$AH(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{3.5}$$

where AH represents the Attention Head, Q represents the pure input value from the embedding, K is the input substitution, V is the dot product of the ratio between Q and K and has SoftMax activation. The word vector multiplication Q,K,V parameter matrix plays an important role in self-attention. A dot product is performed between each word in the sequence to remove the computational similarity and includes the word itself. The value of the dot product of q_i and k_i will be the largest for the same magnitude. In the weighted average of SoftMax, the proportion of the word itself is also the largest, which results in little specific gravity of the other words and cannot effectively utilize the context information to enhance the semantic representation of the current word. In contrast, multiplying the Q,K,V parameter matrix can make each word different, and the above influence can be reduced to a great extent. Figure 3.7 shows the diagram of this Self-Attention calculation,

In order to make h different projections of queries, keys, and values, mapping the dimensionality of d^k and d^v (Ashish, 2017), the results are stitched together by Scaled Dot-Product Attention and finally output by a linear mapping, which enables the model to obtain location information under different subspaces by Multi-Headed Attention. The model is shown in Figure 3.8.

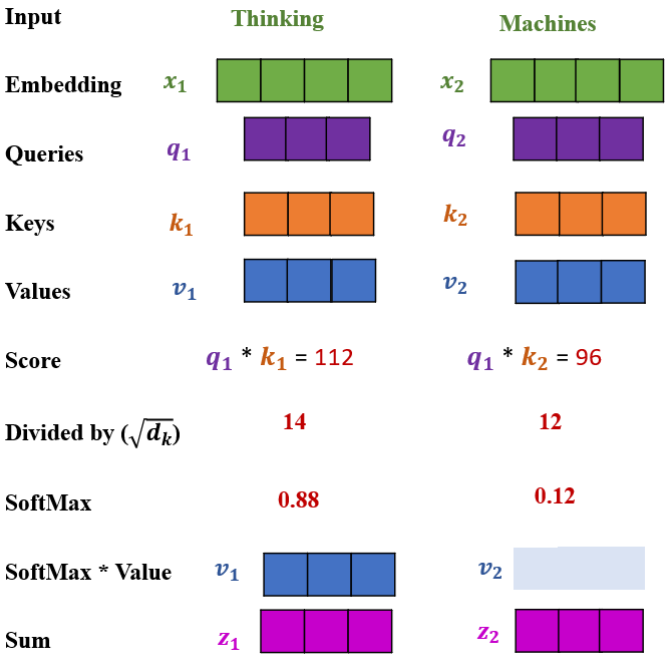


Figure 3.7 The diagram of self-attention calculation

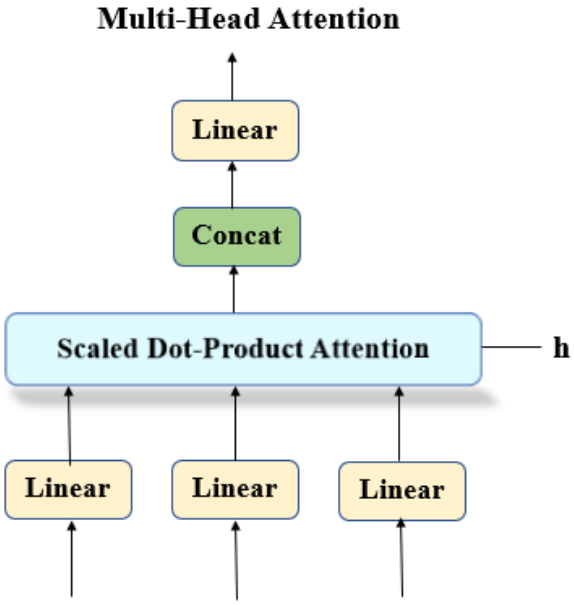


Figure 3.8 Multi-head linear attention

Multi-head linear attention projects *Q, K, V* by *h* different linear variations and stitches the different attention results together. transformer usually uses Multi-Head Attention in three places. Firstly, encoder-decoder Attention. The input is the output of

Encoder and the Self-Attention output of the decoder, where the self-attention of the Encoder is used as key and value. The self-attention of the decoder is used as query. Second, encoder self-attention which input Q, K, V are the input embedding and position embedding of the Encoder. The last is decoder self-attention. In the self-attention layer of the decoder, it can access the front of the current position, and the input Q, K, and V are the input embedding and positional embedding of the decoder. Specifically, as seen in Eq. (3.6) and Eq. (3.7),

$$MH(Q,K,V) = Concat(AH_1, \dots, AH_n)W^o \tag{3.6}$$

$$h_i = A(QW_i^Q,\ KW_i^K, VW_I^V) \tag{3.7}$$

where *MH* represents the Multi Head, the number of heads is multiplied by W^o value. This enables the best feature extraction of the transformer encoder to handle the important parts. This processing model differs from the past, and most of the features currently extract the critical portion of the image using the residual neural network (ResNet) feature extractor (Kaiming, 2015). However, this self-attention on the transformer encoder does not require other features to be extracted, it can produce better results. The last layer of the transformer encoder is a multi-layer perceptron and using GeLU activation (Kevin, 2016). Each of its outputs is based on our categories in the dataset, which are divided into four categories in our case.

Another computational method of similar complexity but with additive attention, which proves that when d_k is small the computational results are similar to dot-product and when $d_k$is large, it performs better compared to dot-product without scaling. The attention in the transformer still needs to be scaled. A large dot product of vectors will push the SoftMax function to a region with a small gradient, which will be mitigated by scaled. Analogy Sigmoid, a relatively large input would cause a gradient of SoftMax to become very small. As an improvement, a dropout can be added after the SoftMax, which we will demonstrate in future experiments.

3.5 Information of Dataset

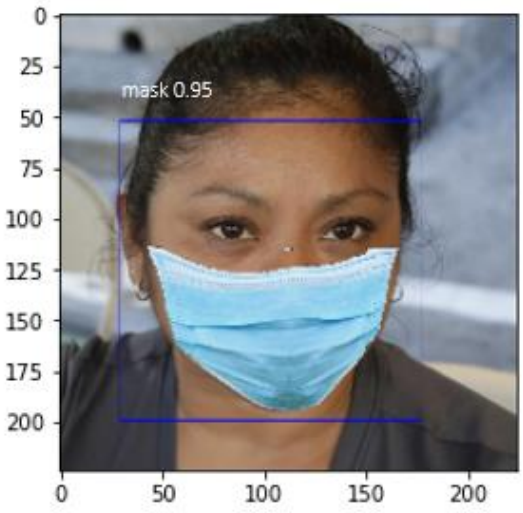
Artificial neural networks (ANN) are cores of deep learning algorithms, which names and structures are inspired by the human brain, imitating the way biological neuron signals are transferred to each other, enabling the machine to operate like our human brain. ANN can iteratively process the image through image recognition to learn extraction features of various images and classify the use of the mask. ANN provide the optimal performance based on the size and uniqueness of the data (Greg & Paul, 2010).

Through our research work, we see that the masked face-net dataset (Adnane, 2021) meets to our research needs. The dataset contains sixty-nine files, a total of 137,016 digital images of the same people with and without the mask. These images have been categorized into four tags which are full masked, mask covering chin, mask covering mouth and chin and mask covering nose and mouth. Compared to the real-world masked face recognition dataset (RMFRD), which contains only 95,000 images (Dalkiran, 2020). The data in the masked face-net network collected by the Flickr-Faces-HQ (FFHQ) dataset contains a much wider range of skin color, age, and race differences (Karras, 2018). We select one of these files separately for training and testing. The class imbalance is avoided because proportion of photos with and without masks is the same. We set the size of all images to 224×224. The experimental environment is MS Windows 10 operating system. We take use of an NVIDIA GeForce RTX 3080 to train the model on Anaconda using Jupyter for simulation.

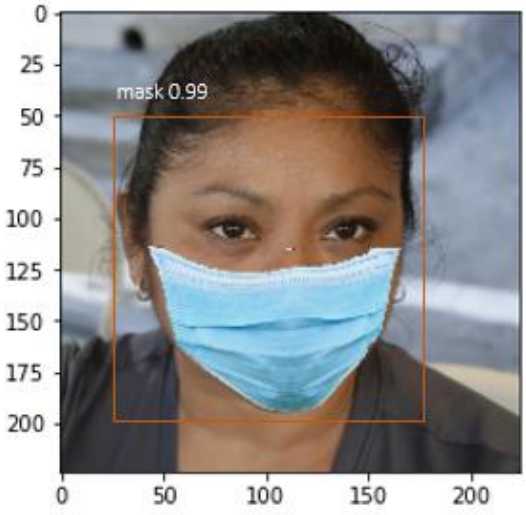
3.6 Data Augmentation

If we are use of such a huge dataset, ANN has a much better classification structure due to the learning of each pixel in the digital image. At the same time, the data augmentation function enables data enrichment. The process of data augmentation uses digital image processing to change images, such as simply by flipping, rotating, shifting, and other minor changes to convert them into new form of digital images (Yang, 2016). Novanto

(2021) etc., proposed that the data augmentation has a significant impact on the training results of the ANN, with higher accuracy and lower loss values and helps ANN to recognize different patterns compared to no data augmentation. We also demonstrated in our experiments that the confidence level of the individual masked image using the standard MFD test is 0.95, while using data augmentation based MFD is 0.99 as shown in Figure 3.9,



(a)



(b)

Figure 3.9: Individual masked image (a) Standard MFD (b) Data augmentation based MFD

The most popular technology is an auto augmentation (Cubuk, 2018), which takes use of a search algorithm controller RNN to perform random selection enhancement on a batch of images, sampling the data, and applying the probability of the optimal search algorithm. However, Auto Augmentation has the shortcoming that a large amount of time needs to be spent when processing such large datasets. In 2019, Cubuk etc., proposed another improvement technique for Rand Augmentation, which reduces the computational process by eliminating the search for the best increment at some phase and using a random distribution on the dataset to eliminate the search space for the best classification result, reducing the search space from 10^{34} to 10^{2}. Although rand augmentation spent less time than auto augmentation, the accuracy of 1% is only improved compared to the latest algorithm, and the search space cannot be increased linearly with the size of the data set.

3.7 Transfer Learning

The original intention of transfer learning is to save time in manually labeling samples, using pre-trained weights of neurons to train architectures that allow models to migrate through existing source domain data to target domain data, thereby training a model suitable for the target domain. If the data has similarities in the transfer learning process, the type and resolution of the digital image can be well applied. The VIT architecture produces a better result if multiple datasets are used for training and before other data being fine-tuned (Alexey, 2020). However, the type and diversity of data has a significant impact on the use of transfer learning (Karl, 2016). When performing transfer learning, we default different tasks to having relevance, but how to mathematically describe the intensity of correlation between tasks is a subjective decision to the human. Therefore, we selected ImageNet as a pre-trained model of fine-tuning in our study (Richard, 2009), since it contains over 14million annotated image datasets according to the WordNet hierarchy (Jonathan, 20115), ensuring a high generalization of the learned network.

3.8 Gradient-weighted Class Activation Mapping (Grad-CAM)

The layers of convolutional units in CNN as target detectors in the model network, although no supervision of the target position is provided. While it owns the ability to locate objects in the convolutional layer, this function is lost when fully connected layer is employed for classification, and the concept of Grad-CAM is proposed on this basis (Ramprasaath, 2016). The application of ANN over many years is realized by human expertise, but it is important to visually interpret it, this enables the user to understand the operation of the system. Each training process in ANN requires a gradient to calculate and update the weights. The GRAD-CAM uses this gradient to obtain a coarse positioning map, and global average pooling is used to reflect which pixel points the model is using to classify the image in the form of the heat map as shown in Figure 3.10,

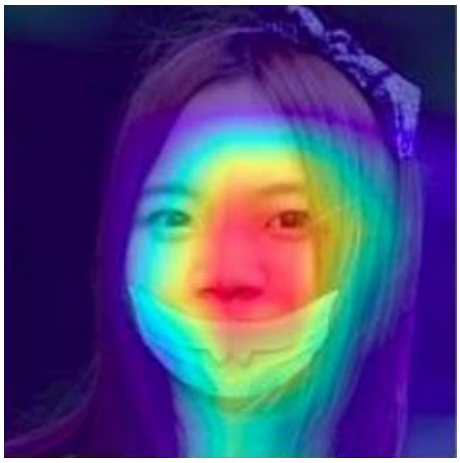


Figure 3.10: Classified the face that don’t wear mask correctly

As shown in Figure 3.10, the wearer exposes the nose portion resulting in an incorrect mask wearing. In the report, we compare the ViT model trained using the transfer learning and the ResNet model based on the residual network, respectively. The key to the ResNet network is the residual cell in its structure, which contains cross-layer links as shown in Figure 3.11,

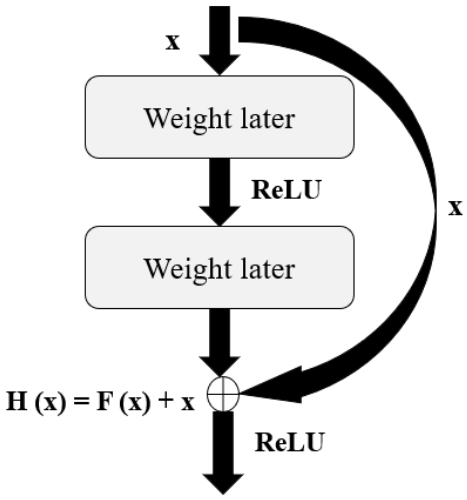


Figure 3.11: Deep residual network model

The curve in the Figure 3.11 can transfer the input directly across the layer, perform the same mapping, and then add to the result of the convolution operation. The input image is X, the output is H, the output after convolution is a nonlinear function of $F(X)$, the final output is $H(X) = F(X) + X$, and such output can still perform non-linear transformation. The ResNet50 network model we used in our experiments contains 49 convolutional layers and one fully connected layer as shown in Figure 3.12,

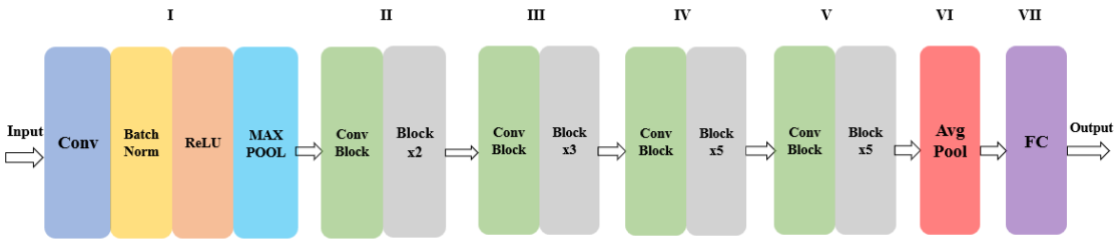


Figure 3.12: ResNet50 architecture

The ResNet50 network model we used in our experiments contains 49 convolutional layers and one fully connected layer. We split it into seven parts, the first part mainly convolves the input, regularization, activation function, and maximum pooling calculation, and the second, third, fourth, and five-part structure contains residual blocks, where the green titles are used to change the dimensions of the residual blocks. The input

of the network is 224 x 224 x 3, the output is 7 x 7 x 2048 through convolution calculation of the fifth part, the pooling layer converts it into a feature vector, and finally, the classifier calculates this vector and outputs the category probability.

The operation of ViT is different from ResNet50, which does not use convolution blocks, but rather the architecture will process the last layer of token-independent attention blocks. Eq. (3.8) and Eq. (3.9) shows the formulae for calculating A^k-weights and Grad-CAM, respectively.

$$a_k^c = \frac{1}{H*W} \sum_{I=1}^{H} \sum_{J=1}^{W} \frac{\partial Y^{(c)}}{\partial A_{ij}^k} \tag{3.8}$$

$$L_{Grand-CAM}^{(C)} = \text{RELU} \left(\sum_K a_k^{(c)} A^k \right) \tag{3.9}$$

where *H*, W represent the height and width of the image, a_k^c represents the weighting of A^k, K represents the *Kth* channel of *A* in the feature layer. Category *C* represents the $L_{Grand-CAM}^{(C)} \in R^{W*V}$. First, the gradient of this class *C* is calculated, and the activation value of the feature map A^k is defined as *a* before the $y^{(c)}$ is activated by SoftMax activation. Through Eq. (3.6), it can be known that a_k^c is to perform backpropagation of the prediction score $y^{(c)}$ through the prediction category *C*, and then calculate the weight of each channel *K* in the feature layer *A* by using the gradient information of the reverse transmission to the feature layer. The data of each channel of the feature layer is weighted and summed by *a*, and the Grad-CAM is obtained by the ReLU activation function (Agarap 2018). The purpose of using the ReLU function is to filter out negative pixels and focus on significant portions of the gradient mapping on the image.

Chapter 4
Results

The main content of this section is to collect training data and analysis accuracy of each Vision Transformer model results, we also discuss the limitations of the project in the end of this section.

4.1 Evaluation Indicators

In this report, we introduce accuracy, recall, and precision to evaluate the performance of image classification task models. The accuracy rate is the proportion of correct predictions over all samples using the formula as shown in Eq. (4.1). TP in the formula denotes the sample predicted to be positive among all positive samples; TN shows the sample predicted to be negative among all negative samples; FP indicates the sample predicted to be positive among all negative samples; FN reflects all positive samples predicted to be negative samples. Although accuracy can be employed to determine overall correctness, it is not a good indicator of results in the case of an unbalanced sample. The high accuracy obtained can be validated by sample balance.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \tag{4.1}$$

Recall is the proportion of the number of correctly predicted positive samples to the actual number of positive samples, its equation is shown in Eq. (4.2),

$$\text{Recall} = \frac{TP}{TP+FN} \tag{4.2}$$

The precision is the probability of predicting the actual positive sample in a positive sample, its equation is shown in Eq. (4.3), where a higher recall means a higher probability that an actual useless user will be predicted

$$\text{Precision} = \frac{TP}{TP+FP} \tag{4.3}$$

4.2 Training Result Analysis

Combined with the evaluation metrics, the accuracy, recall and average accuracy of each classification of the two datasets were calculated, the evaluation results are shown in Table 4.1.

From the evaluation results, MobileNetV2 neural network has good effect on the recognition whether the mask is worn or not, the average accuracy, recall and precision rates of each class are above 97%. To further verify the effectiveness of the algorithm, we compare the algorithm in this paper with other deep learning algorithm VGG16 using the

same dataset. The result is shown in Table 4. In our experiments, we made use of batch size variations of 2, 4, 8, 16 and 32. A batch size of 8 means that the data set is divided into eight batches for neural network training.

Table 4.1 Experiment results in the training data

	Precision	Recall	F1-score
With_mask	0.9641	0.9812	0.9719
Without_mask	0.9882	0.9764	0.9746
accuracy			0.97
macro avg	0.97	0.98	0.97
weighted avg	0.97	0.98	0.97

Table 4.2 Different model using batch size variations

	2 batch		4 batch		8 batch		16 batch		32 batch	
	Train Acc	Val Acc	Train Acc	Val Acc	Train Acc	Val Acc	Train Acc	Val Acc	Train Acc	Val Acc
VGG 16	0.9341	0.9551	0.9512	0.9722	0.9811	0.9826	0.9710	0.9829	0.9870	0.9832
MobileNetV2	0.9507	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000

The result is shown in Table 4.2, the more batches get the better training and validation accuracy. Even on the batch four, MobileNetV2 already got 100% accuracy of training and validation. Compared to VGG16, the maximum accuracy rate of 98.70% was only obtained in 32 batches, we chose to use 32 batches as in the number of epoch experiment as shown in Table 4.3.

The neural network is trained based on the dataset until it is reset to the beginning of the round. In Table 5, MobileNetV2 achieved 100% training and validation accuracy on epoch 20. Meanwhile, VGG16 only gets 55.01% and 47.82% training and validation accuracy, respectively. VGG16 still does not reach 100.00% accuracy on epoch 50. As a

result, the pre-trained MobileNetV2 model has better accuracy than VGG16 which can obtain the best model.

Table 4.3 Different model using epochs variations

	10 Epoch		20 Epoch		30 Epoch		40 Epoch		50 Epoch	
	Train Acc	Val Acc	Train Acc	Val Acc	Train Acc	Val Acc	Train Acc	Val Acc	Train Acc	Val Acc
VGG 16	0.1596	0.2217	0.5501	0.4782	0.8757	0.9103	0.9629	0.9225	0.9857	0.9847
MobileNetV2	0.9687	0.9624	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000

4.3 Results of ViT Comparison

In the previous section, we pointed out using ViT for mask detection. According to Alexey's article on transformer research, a variant of ViT as shown in Table 4.4.

Table 4.4: Details of Vision Transformer model variants

Model	Layers	Hidden Size D	MLP Size	Heads	Params
ViT Base/16	12	768	3072	12	86M
ViT Large/16	24	1024	4096	16	307M
ViT Huge/14	32	1280	5120	16	632M

As we can see from the table above, the ViT Base/16 indicates that it contains 16 x 16 input patch size, 12 layers of encoder, 768 hidden sizes, 3072 of multilayer perceptron (MLP) in encoder, 12 heads and 86 million overlay parameters, where the sequence length of the transformer is inversely proportional to square of the patch size. ViT Large/16 indicates that it contains 16 x 16 input patch size, 24 layers of encoder, 1024 hidden sizes, 4096 of MLP in encoder, 16 heads and 632 million overlay parameters. ViT Huge/14

indicates that it contains 14 x 14 input patch size, 32 layers of encoder, 1280 hidden sizes, 5120 of MLP in encoder, 16 heads and 632 million overlay parameters. We have investigated what impacts on the accuracy of training, validation, and test data for architecture of different models in experiments as shown in Table 4.5,

Table 4.5 Details of accuracy of Vision Transformer model

Model	Train	Validation	Test
ViT Base/16	0.76144	0.76536	0.83542
ViT Large/16	0.72248	0.77217	0.78749
ViT Huge/14	0.81259	0.81782	0.92573

Table 4.5 shows the training results of the three ViT models after 20 epochs and selected the highest precision values. As can be seen from the experiment results, Vit Huge/14 has the highest accuracy for train, validation, and test portions, with the accuracy of each data set exceeding 0.8, especially the accuracy of the test set reaches nearly 0.93. Meanwhile, though ViT Large/16 has nearly three times the size parameters of ViT Base/16, only the performance of the validation portion exhibits slightly outstanding performance, and the rest of the performance is the worst. Training and evaluation results for three ViT architectures as shown in Figure 4.1 and Figure 4.2, respectively.

The training accuracy of the ViT architecture is indicated by each epoch from figure4.1. The ViT Huge/14 architecture obtained the highest accuracy on the 20th epoch, which is 0.81259. It reached a precision of nearly 0.8 at the 7th epoch but fell back to its initial precision of 0.52 at the 8th epoch, and then improved at the 13th epoch. ViT Base/16 obtained the highest precision of 0.76144 at the 19th epoch. In comparison, the accuracy of the ViT Large/16 is maintained below 0.6 before the 15th epoch, and the highest accuracy of 0.72248 is generated under the 18th epoch.

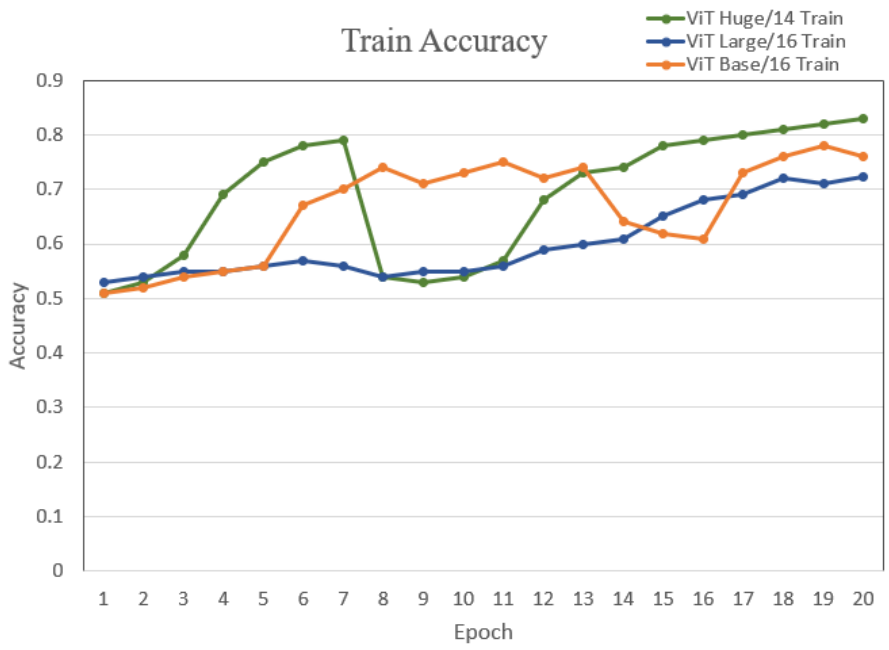


Figure 4.1 Training accuracy of three ViT in 20 epochs

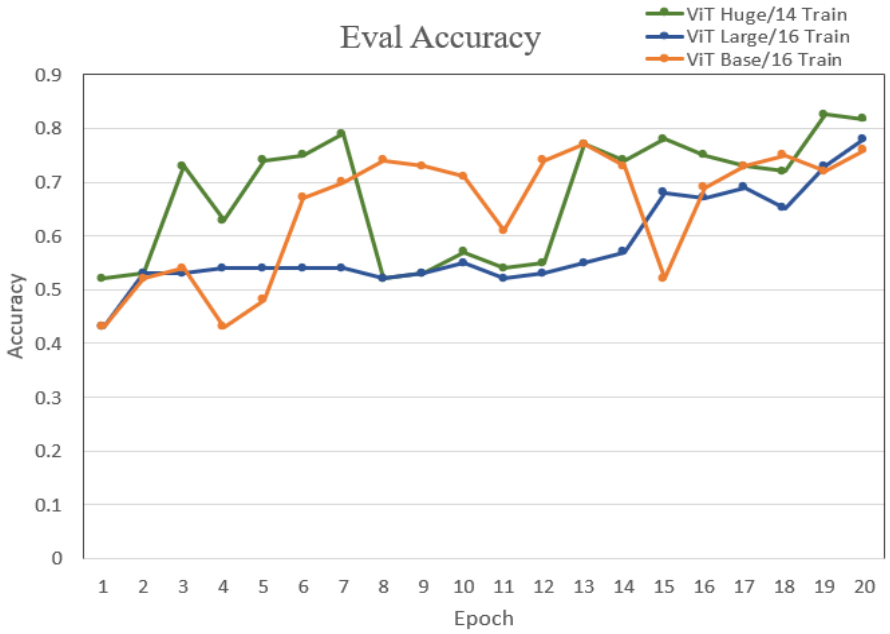


Figure 4.2 Evaluation accuracy of three ViT in 20 epochs

From Figure 4.2, we see that the evaluation result and the training result of each ViT model architecture are not too large. The evaluation accuracy of 0.81782 obtained by ViT Huge/14 at the 19th epoch is higher than the 0.76536 obtained by ViT Base/16 at the 13th

epoch and the 0.77217 obtained by ViT Large/16 at the 20th epoch. As can be seen from the test results, the overall comparison of the accuracy of the ViT Huge/14 structure is generally higher than the other two ViT architectures. We use the ViT Huge/14 architecture to perform the next study on the accuracy of the amplification on the training and validation set.

4.4 Results of Augmentation on Accuracy of Training and Validation sets

Based on the above experiments, we chose ViT Huge/14 with higher accuracy and avoiding overfitting to improve the accuracy using a random augmentation method on the training and validation set. We compared the impact on accuracy with and without augmentation in our experiment. Table 4.6 shows the results.

Table 4.6 Details of accuracy using and without augmentation on ViT Huge/14

Model	Train	Validation	Test
Using Augmentation	0.82547	0.83121	0.92573
Without Augmentation	0.81259	0.81782	0.92573

Table 4.6 shows the accuracy comparison between the ViT Huge/14 training and validation sets using and without data augmentation respectively. As we can see from the table, the data set using augmentation gives a higher degree of accuracy. In particular, the validation set has improved significantly, with the validation value improving from 0.81782 to 0.83121. There was also a slight improvement in the training set, with the precision of the training data improving from 0.81259 to 0.82547. This indicates that the raw data processed by data augmentation has achieved a higher degree of performance and accuracy, especially in the validation set. Figure 4.3 and Figure 4.4 shows training accuracy and validation accuracy using and without augmentation on ViT Huge/14 in 20

epochs.

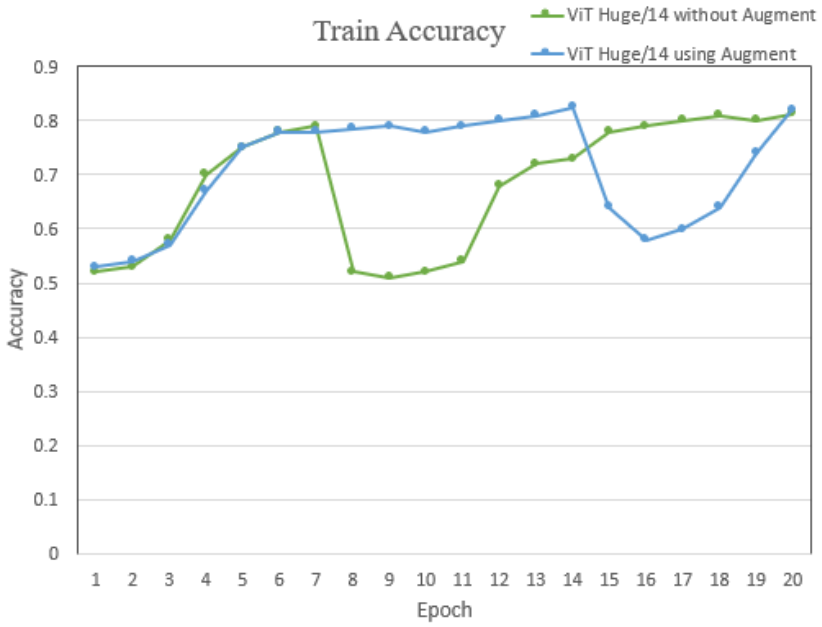


Figure 4.3 Training accuracy of ViT Huge/14 using and without augmentation in 20 epochs

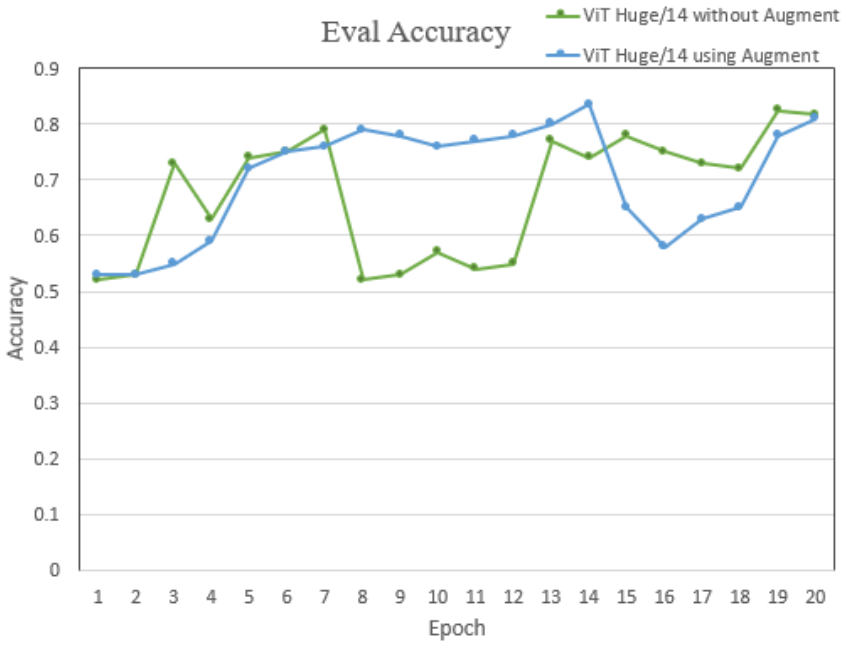


Figure 4.4 Validation accuracy of ViT Huge/14 using and without augmentation in 20 epochs

From Figure 4.3, we see, in the training of the ViT model, the highest accuracy is achieved by comparing the training set without data augmentation in the 20th epoch, and the ViT Huge/14 with the data augmentation obtains the highest accuracy on the 14th epoch to reach 0.82547. On the evaluation set, the ViT Huge/14 using data augmentation also obtains higher accuracy, reaches 0.83121 in the 14th epoch, and the lifting amplitude is more obvious. As a result, data augmentation performed on a data set can generate higher accuracy and faster convergence.

4.5 Results of Transfer Learning on Accuracy of Training and Validation Sets

We demonstrate that the use of data augmentation makes a significant improvement in the accuracy of the validation and training sets. To test the impact of pre-training weights, we compared the ViT Huge/14 model on the ImageNet-21K after data augmentation with and without pre-training weights. The results we derived are shown in Table 4.7.

Table 4.7 Details of accuracy using and without transformer learning on ViT Huge/14

Model	Train	Validation	Test
With Transfer Learning	0.98144	0.95009	0.95874
Without Transfer Learning	0.82547	0.83121	0.92573

Table 4.7 shows the accuracy using and without transformer learning on ViT Huge/14. It can be seen from the table that the pre-training weights are used to obtain higher accuracy than the without pre-training weights, and the lifting amplitude is obvious.

On the training set, an accuracy of almost 0.98144 was obtained using the pre-trained weights, which is significant compared to the 0.82547 obtained when not used. It is also possible to obtain a higher accuracy on the validation set by using a pre-trained weight model, which achieves 0.95009 and without using a pre-training weight only 0.83121. This method also has the same effect on the test set, its accuracy is improved to 0.95874. Figure 4.5 and Figure 4.6 shows training accuracy and validation accuracy using and without transformer learning on ViT Huge/14 in 20 epochs,

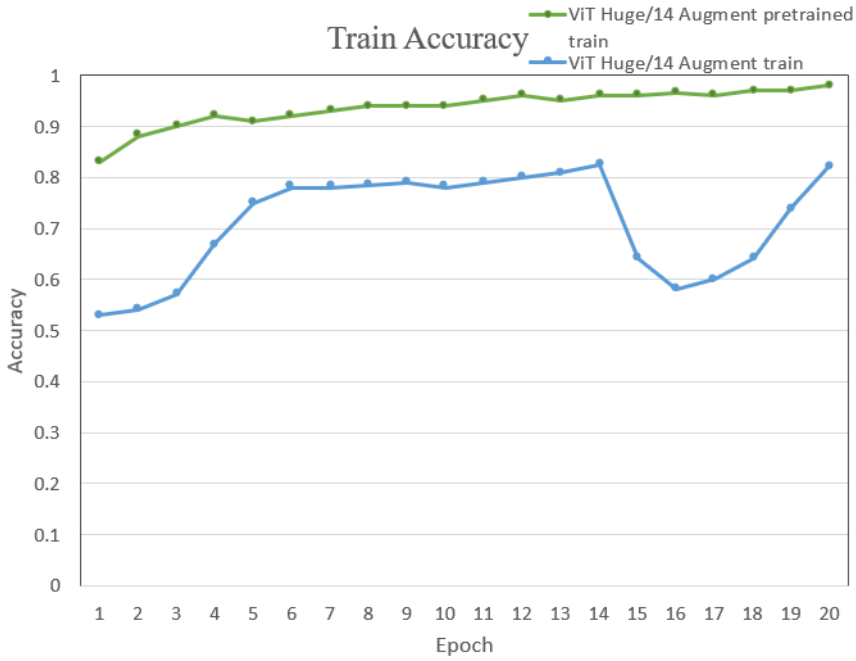


Figure 4.5 Training accuracy of ViT Huge/14 using and without Transformer learning in 20 epochs

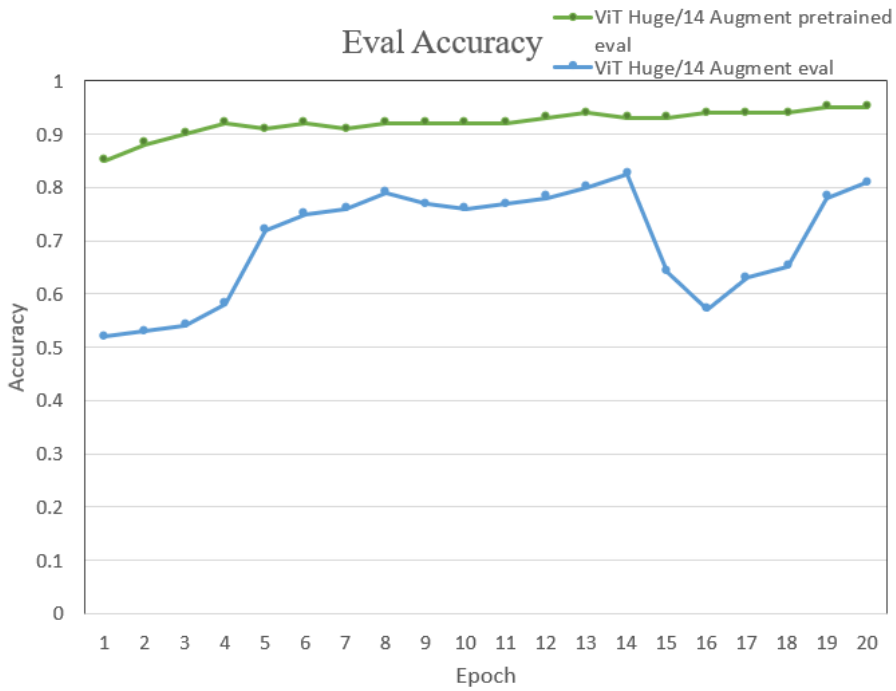


Figure 4.6 Validation accuracy of ViT Huge/14 using and without Transformer learning in 20 epochs

In Figure 4.5, we see the training of the ViT model, the highest accuracy is achieved by comparing the training set without transformer learning in the 20 epochs, and the ViT Huge/14 with transformer learning obtains the highest accuracy on the 18th epoch to reach 0.98144. On the evaluation set, the ViT Huge/14 using data augmentation also obtains higher accuracy, reaches 0.95009 in the 20th epoch. The minimum precision on both datasets exceeds 0.8, which is better than the highest precision without using pre-training, and the two models do not have a substantial decrease in accuracy due to gradient loss when without using pre-training set in 14th epoch. The lifting amplitude is more obvious. As a result, pre-training performed on a data set can generate higher accuracy and faster convergence.

4.6 Comparison of ViT and ResNet50 Accuracy Results

Based on the two sets of experiments, using both data augmentation and pre-training weights can significantly improve the accuracy and reduce losses in the training,

validation, and test sets. Therefore, in the following study we will compare the accuracy improvements on the training, validation and test sets using these two approaches for the ViT model architecture and the RestNet 50 architecture mentioned in Figure 3.23, respectively. The test results for each architecture are visible on Table 4.8.

Table 4.8 Details of accuracy using augmentation and transformer learning on each model

Model	Train	Validation	Test
ViT Huge/14	0.98144	0.95009	0.95874
ViT Base/16	0.96412	0.93875	0.83542
ViT Large/16	0.98867	0.97210	0.91533
ResNet 50	0.90155	0.91128	0.87428

In Table 4.8, we see that ViT Large16 produced a maximum precision of 0.98867 and 0.97210 in the training and validation sets, respectively. This value is higher than the 0.98144 for the training set and 0.95009 for the validation set obtained in the previous studies using ViTHuge/14. However, the highest accuracy on the test set is still 0.95874 obtained by ViTHuge/14, higher than the 0.91533 of ViTLarge/16, which reflects the problem of over-fitting of the ViTLarge/16 architecture. Compared to the ViT architecture, the ResNet50 architecture is not as accurate in all aspects, with 0.90155, 0.91128 and 0.87428 respectively. In Figure 4.7 and 4.8 we show the training and validation set results for all model architectures on 20 epochs, respectively.

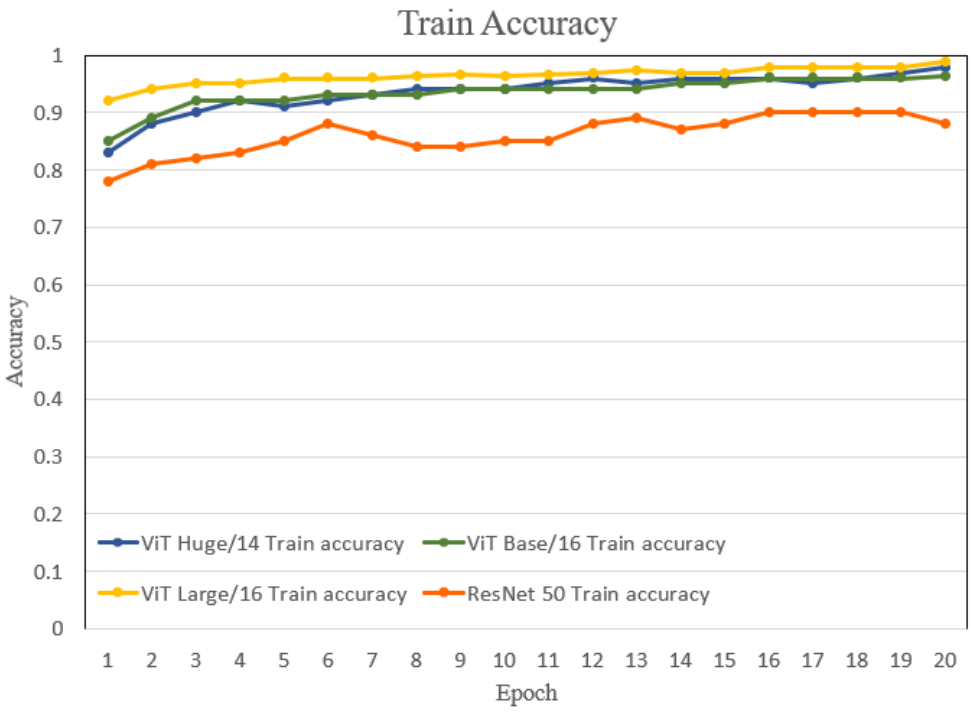


Figure 4.7 Training accuracy of ViT and ResNet50 using augment and pretrained in 20 epochs

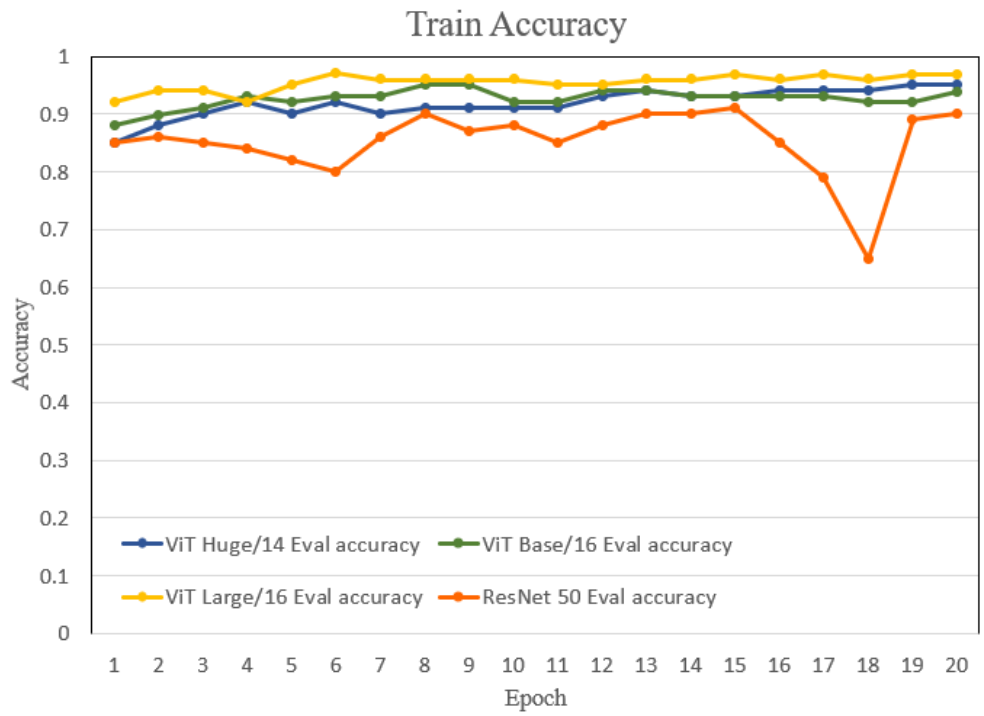


Figure 4.8 Validation accuracy of ViT and ResNet50 using augment and pretrained in 20 epochs

The performance of ViTLarge/16 performs significantly better on the training and validation set. The highest precision was achieved at the 6th epoch in the validation set. The values of ViTHuge/14 and ViTBase/16 are more average, but ResNet 50 performs very poorly in both training and validation. In Figure 4.8, ResNet 50 shows a significant drop in accuracy from 15th epoch to 18th epoch on the validation set, which is due to the loss of gradient.

4.7 Analysis of Confusion Matrix

We are use of the confusion matrix as a visualization tool to compare the classification results with the actual test values. According to the above experiment, the best test performance was obtained by ViT Huge/14 using pre-training weights and dataset augmentation. Table 4.9 shows the values for each type of prediction accuracy. The confusion matrix for test data as shown in Figure 4.9,

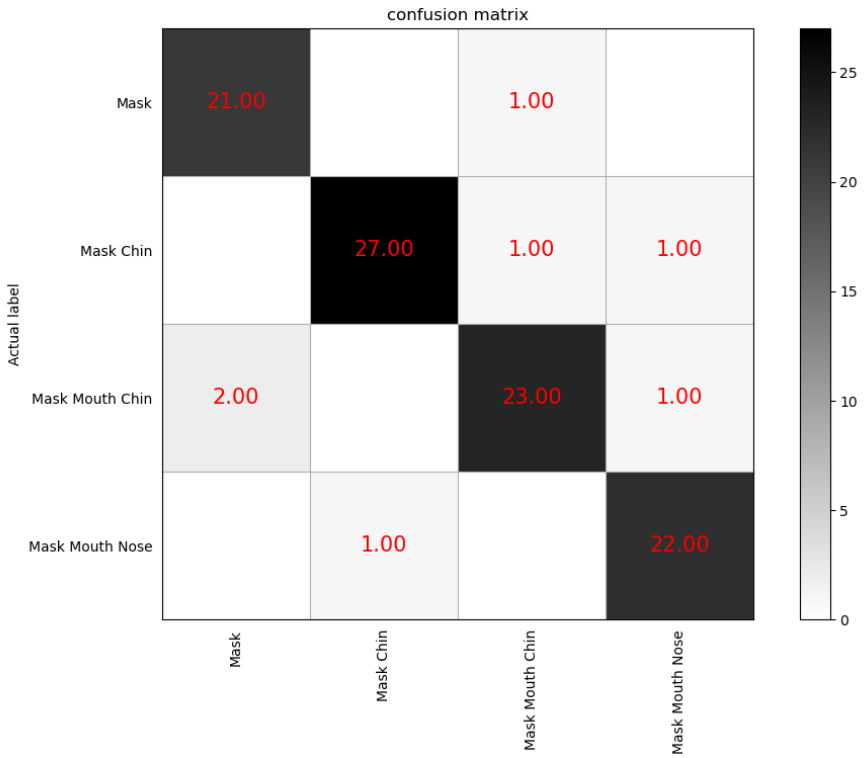


Figure 4.9 Confusion matrix of mask recognition

Table 4.9 Confusion matrix on each class

Actual label	Mask	Mask Chin	Mask Mouth Chin	Mask Nose Mouth
Mask	0.9545	0.0000	0.0455	0.0000
Mask Chin	0.0000	0.93103	0.0345	0.0345
Mask Mouth Chin	0.0769	0.0000	0.88462	0.0385
Mask Nose Mouth	0.0000	0.0435	0.0000	0.95652

In Table 4.9, the prediction accuracies for the mask class and the masked chin class were 0.9545 and 0.93103, respectively. The predicted value for the masked mouth and chin class is slightly lower than the other categories which is 0.88462 and the category of masked nose and mouth obtain 0.95652 accuracy. The probability of error presentation in the experiment is very low, the highest is 0.0769 tested in the Mask Mouth Chin classification, and the overall experimental result is quite excellent.

4.8 Limitations of the Research

(1) The use of sharpness-aware minimization (SAM) training ViT in the experiments generates a new round of forward and backward propagation, resulting in a double increase in computational cost per update.

(2) With the increase of our data set, the effect of SAM is also weakened, we need to develop a learning method capable of improving large-scale data sets.

Chapter 5
Discussions

In this chapter, we analysis and compare the experimental results.

5.1 Discussions

In this paper, we proposed the MobileNetV2 lightweight with the marked face recognition algorithm which is offered for public face detection in the current epidemic environment. In our experiments, we propose to use transfer learning to extract facial features from the data and perform classification. According to the experiments, we find that MobileNetV2 model has better accuracy by comparing with VGG16. The deep learning method in this paper generates higher efficiency.

Chapter 6
Conclusion and Future Work

In this section, we summaries the results of our experiments and propose directions for future research.

6.1 Conclusion

By performing data enhancement on different ViT architectures to classify various mask wearing modes, we conclude that the training of the existing data set is greatly improved by using a random enhancement method. We process the digital image with a predetermined number of patches and use linear projection to transform the processed 3D digital image into 2D. Before using the encoder, we use the attention mechanism in migration learning to focus on the features of the image. The resulting feature map is trained in a multilayer perceptron to classify classes based on those already in the library.

Through experimental results, we see that the prediction accuracies for the mask class and the masked chin class were 0.9545 and 0.93103, respectively. The classs of masked nose & mouth obtains 0.95652 accuracy. The experimental results obtained from our first attempt at using the transfer learning method show a significant performance improvement compared to existing convolutional baseline methods.

6.2 Future Work

With the increase of our data set, the effect of SAM is also weakened, we need to develop a learning method capable of improving large-scale data sets.

References

Abien, F. (2018). Deep Learning using rectified linear units (ReLU). *arXiv Neural and Evolutionary Computing (cs.NE). arXiv: 1803.08375v2.*

Adjabi, I., Benzaoui, A., Ouahabi, A. (2020). Past, present and future of face recognition: A review. Electronics. In *Mathematics&Computer Science Artificial Intelligence* (Vol.9, no.8, pp. 1-53). *arXiv preprint arXiv:10.3390*.

Adnane, C., Karim, H., Halim, B., & Mahmoud, M. (2021). MaskedFace-Net – A dataset of correctly/incorrectly masked face images in the context of COVID-19. Smart Health 19, pp.177-191. Springer.

Agrawal, A., Choudhary, A. Gopalakrishnan, K., Khaitan, K. (2017). Deep convolutional neural networks with transfer learning for computer vision-based data-driven pavement distress detection. In *Constr, Buid. Matter*, Vol. 157, pp. 322-330. Springer.

Ahuja, U., Kumar, K., Kumar, M., Sachdeva, M., Singh, S. (2021). Face mask detection using YOLOv3 and Faster R-CNN models: COVID-19 environment, Multimedia Tools Appl, pp. 19753-19768, Springer.

Akhil, K., Kalia, A., Kaushal, M., Sharma, A. (2021). A hybrid tiny YOLO v4-SPP module based improved face mask detection vision system. *Journal of Ambient Intelligence Humanized Computing*, Vol. 14752, no.5, pp 142-145. IEEE.

Albanie, S., Hu, J., Shen, L., Sun, G., Wu, E. (2019). Squeeze-and-excitation networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 42, no. 8, pp. 2011-2023. IEEE.

Albert, R., Jesus, T., Jaime, L., & Sandra, V. (2021). Incorrect facemask-wearing detection using convolutional neural networks with transfer learning. *Human Health and Healthcare,* pp. 524-536. MDPI.

Alexey, D., Lucas, B., Dirk, W., Thomas, U., Mostafa, D., & Georg, H. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv Computer Vision and Pattern Recognition(cs.CV). arXiv: 2010.11929.*

Analia R., Karlina I., Susanto S. (2020). The face mask detection for preventing the spread of COVID-19 at Politeknik. In *International Conference on Applied Engineering (ICAE)*, pp. 115—124. IEEE.

Andrew, G., Marco, A., Weijun, W., Weyand, T. (2017). MobileNets: Efficient convolutional neural networks for mobile vision applications, Vol.12, pp 233-240.

Arymurthy, A., Gultom, Y., Masikome, J. Batik. (2018). Classification using deep convolutional network transfer. *Journal Ilmu Komputer Dan Informasi,* Vol. 11, no. 2, pp. 59.

Asadi S., Cappa C., Wexler. A. (2019). Aerosol emission and superemission during human speech increase with voice loudness. *Scientific Reports*, pp. 1-10.

Ashish, V., Noam, S., Niki, P., Jakob, U., Llion, J., Lukasz, K. (2017). Attention is all you need. *arXiv Computation and Language. arXiv:1706.03762*

Aswal, V., Charniya, N., Shaikh, S., Tupe, O. (2020). Single camera masked face. *International Seminar on Research of Information Technology and Intelligent Systems*, pp. 57—60. IEEE.

Barret, Z., Dandelion M., Ekin, D., & Vijav, V. (2018). Auto augment: Learning augmentation policies from data. arXiv *Machine Learning (stat.ML)*. arXiv: *1805.09501*

Barret, Z., Ekin, D., Jonathon, S., & Quocv, L. (2019). RandAugment: Practical automated data augmentation with a reduced search space. *IEEE Computer Vision and Pattern Recognition.*

Bosheng Q., & Dongxiao L. (2020). Identifying facemask-wearing condition using Image super-resolution with classification network to prevent COVID-19. *Sensing and Imaging,* pp. 5236-5241. MDPI.

Cabani, A., Benhabiles,H., Hammoudi, K., Melkemi, M. (2020). Masked face-net-A dataset of correctly/incorrectly masked face images in the context of COVID-19. *Smart Health*, vol. 19, pp 31-40.

Cai, Q., Peng, C., Shi, X. (2021). Lightweight face recognition algorithm based on MobileNetV2. *International Journal of Intelligence Science*, Vol.11, pp. 230-239. IEEE.

Chen, L., Howard, A., Sandler, M., Zhmoginov, A., Zhu, M. (2020). MobileNetv2: Inverted residuals and linear bottlenecks. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Cui, W., Yan, W. (2016) A scheme for face recognition in complex environments. International Journal of Digital Crime and Forensics (IJDCF) 8 (1), 26-36.

Cui, W. (2015) A Scheme of Human Face Recognition in Complex Environments. Master's Thesis, Auckland University of Technology.

Dan, H., & Kevin, G. (2016). Gaussian error linear units (GELUs). arXiv *Machine Learning (stat.ML)*. arXiv: *1606.08415.*

Deng, J., Guo, J., Xue, N., Yang, J. (2016). ArcFace additive angular margin loss for deep face recognition. In *International Conference on Computer Vision, Visual Communications, and Image Processing*.

Elmahmudi, A., & Ugail, H. (2019). Deep face recognition using imperfect facial data. *Futur. Gener. Comput. Syst.*, Vol. 99, pp. 213-225, Springer.

Gao, X., Nguyen, M., Yan, W. (2021). Face image inpainting based on generative adversarial network. In *International Conference on Image and Vision Computing New Zealand*.

Gu, Q., Yang, J., Kong, L., Yan, W., Klette, R. (2017). Embedded and real-time vehicle detection system for challenging on-road scenes. *Optical Engineering* 56 (6), 063102, pp 14-25.

Geoffrey, E., Jamie, R., & Jimmy, L. (2016). Layer normalization. *arXiv Machine Learning (stat.ML). arXiv: 1607.06450.*

Goh, Y.H.., Lee, Y.B., Lum, K.Y. (2020). American sign language recognition based on MobileNetV2. *Adv. Sci. Technol*, vol. 5, no. 6, pp. 481-488. ASTES

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation,* pp. 1735-1780.

Hu, L., & Ge, Q. (2020). Automatic facial expression recognition based on MobileNetV2 in real-time. *Journal of Physics,* Vol. 15449, no. 2, pp. 112-116. IOP science

Ivan, T., Rico, S., & Zhang, B. (2019). Improving deep Transformer with depth-scaled initialization and merged attention. *arXiv Computation and Language (cs.CL). arXiv: 1908.11365v1.*

Jamadar, S., Joshi, P., Surve, M., Vharkate, M. ZIB. (2015). Automatic attendance system using face recognition technique. *Recent Technol. Eng,* Vol. 9, no. 1, pp. 2134—2138.

Jiao, Y., Weir, J., Yan, W. (2011) Flame detection in surveillance. Journal of Multimedia 6 (1).

Jingxiang, Y., Yongqiang, Zh., & Chen, Y. (2016). Hyperspectral image classification using two-channel deep convolutional neural network. In *International Geoscience and Remote Sensing Symposium (IGARSS)*, pp. 10—15.

Jonathan, K., Sanjeev, S., Sean, M., Zhiheng, H., & Michael, B. (2015). ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*. (pp. 211-252). Springer.

Kaiming, H., Xiangyu, Z., Shaoqing, R., & Jia, S. (2015). Deep residual learning for

image recognition. *arXiv: 1512.03385.*

Katharina, K., Wenpeng, Y., & Mo, Y. (2017). Comparative study of CNN and RNN for natural language processing. *arXiv:1702.01923*

Lahasan, B., Lutfi, S.L., Segundo, R. (2019). A survey on techniques to handle face recognition challenges: Occlusion, single sample per subject and expression. *Artificial Intelligence Review*, pp. 949-979, Springer.

Li, M., Liu, W., Wen, Y., Yu, Z. (2017). Sphere face: Deep hypersphere embedding for face recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6738-6746.

Liu, M., Yan, W. (2022) Masked face recognition in real-time using MobileNetV2. *ACM ICCCV.*

Nguyen, Q., Thai, Q., & Yu., Y. (2021). A transformer architecture based on BERT and 2D convolutional neural network. *Bioinformatics,* Vol. 22, pp. 178-195.

Ramprasaath, R., Selvaraju, M., Ramakrishna, V., & Devi, P. (2016). Grad-CAM: Visual explanations from deep networks via gradient-based localization. *arXiv Computer Vision and Pattern Recognition (cs.CV). arXiv: 1610.02391v4.*

Samuli, L., Tero, K., & Timo, A. (2018). A style-based generator architecture for generative adversarial networks. *arXiv: 1812.04948.*

Shen, D., Xin, C., Nguyen, M., Yan, W. (2018) Flame detection using deep learning. International Conference on Control, Automation and Robotics.

Wang, H. (2018). Real-time Face Detection and Recognition Based on Deep Learning (Master's Thesis), Auckland University of Technology.

Wang, H., Yan, W. (2022) Face detection and recognition from distance based on deep learning. Aiding Forensic Investigation Through Deep Learning and Machine Learning Framework. IGI Global.

Weihong, D., & Yaoyao, Z. (2021). Face Transformer for recognition. *arXiv Computer*

Vision and Pattern Recognition. arXiv:2103.14803.

Xiao, L., Peng, L., & Zhong, H. (2012). Simultaneous image classification and annotation based on probabilistic model. *Journal of China Universities of Posts and Telecommunications*, Vol. 19, pp 107-115.

Xin, C. (2020) Detection and Recognition for Multiple Flames Using Deep Learning. Master's Auckland University of Technology, New Zealand.

Xin, C., Nguyen, M., Yan, W. (2020) Multiple flames recognition using deep learning. Handbook of Research on Multimedia Cyber Security, 296-307.

Yan, W., Kankanhalli, M. (2002) Erasing video logos based on image inpainting. IEEE International Conference on Multimedia and Expo, 521-524.

Yan, W., Kankanhalli, M., Wang, J., Reinders, M. (2003) Experiential sampling for monitoring. ACM SIGMM Workshop on Experiential Telepresence, 70-72.

Yan, W., Kankanhalli, M. (2015) Face search in encrypted domain. Pacific-Rim Symposium on Image and Video Technology, 775-790.

Yan, W. (2021). Computational Methods for Deep Learning Theoretic, Practice and Applications, Springer.

Yan, W. (2019). Introduction to Intelligent Surveillance: Surveillance Data Capture, Transmission, and Analytics, Springer.

Zhang, T. (2011). Adaptive forward-backward greedy algorithm for learning sparse representations. *IEEE Transactions on Information Theory*, pp. 4689-4708.

Zhu, W., Yan, W. (2022). Traffic sign recognition based on deep learning. Multimedia Tools and Applications, (pp.17779-17791). Springer.