Electrical Engineering Project

Department of Electrical Engineering

Faculty of Engineering

Kasetsart University

Age and Gender Prediction

By

Jakkaphob Kongthanarith 6210554725

Academic Year (A.D.) 2022

Age estimation and gender identification

By

Jakkaphob Kongthanarith 6210554725

Academic Year (A.D.) 2022

Electrical Engineering Project

Department of Electrical Engineering

Faculty of Engineering

Kasetsart University

Submitted in partial fulfillment of the requirements for the degree of Bachelor degree of Engineering

Electrical Engineering

Approved by SeniorProjectAdvisor .... Date: ....

(Assoc. Prof. Dr. Ekachai Phaisangittisagul)

ProjectCommittee .... Date: ....

(Prof. Dr. Wiroonsak Santipach)

Jakkaphob Kongthanarith Academic Year 2022
Age and Gender Prediction
Bachelor degree of Engineering, Department of Electrical Engineering
Faculty of Engineering, Kasetsart University

Abstract

From the past to present, security has been improved time by time to prevent the crime from happening and also bring the criminal to justice. There are several methods for detecting and identifying the identity of people, one of them is detection by gender and age.

This project goal is to create and compare which deep learning model are best suited for predicting age and gender using face image of the target. In this way, we build a deep learning model which can be used for further identity recognition.

Keywords: Deep learning, Age estimation, Gender identification

Department Reference No. ....ECP-02..................

Acknowledgement

First, I would like to express my sincere gratitude for the advice, resource, and comment of my advisor, Assoc. Prof. Dr. Ekachai Phaisangittisagul. I would not have come this far without him.

Second is my project committee, Prof. Dr. Wiroonsak Santipach for his kindness toward my project. And also the rest of people who have supporting me from time to time.

Jakkaphob Kongthanarith

Table of content

List of Figures

List of Table

Symbols and abbreviations

TP stands for True positive

TN stands for True negative

FP stands for False positive

FN stands for False negative

n = Total number of files used for calculating MAE value

i = Initial value. In this case, it starts from the first file in the range.

X_i = The predicted result of model for an individual image.

X = The actual result for that individual image.

1 Introduction

With the criminal risen though the country, catching them is the main purpose of security protocol. The detection for recognize their identity is expected to be accurate. It's a fact that both age and gender of human are quite hard to detect even using a machine, the reason is human face are unique and contain many feature that are varied by their own. Human faces cannot be completely indicated by their birth and moreover, the structure and features of the face can be changed not by both inside factors such as genes or growth but also by outside factors such as surgery or cosmetics. These factors might cause a fault in the model and make it more difficult to point out the exact age or gender. But as time passed, technology also developed with a rapid speed. Models nowadays can find the pattern that has been hidden from our human eyes and improvise themselves to fit the given task. I want to create a model hoping for them to be able to predict the human age and gender with the performance as high as the situation allows.

1.1. Objectives

To Find the best performance model that have a highest accuracy when make a prediction of age and gender out of human face.

1.2. Scope

Build a deep learning model which give the accuracy result more than 75% and MAE less than 1

2 Related Theories

2.1. Deep Learning

A technique which is a part of machine learning method that uses a neural network along with feature learning, which is a technique for the system to discover a pattern of each image that detect and classify pattern from the data. Neural network is a core of deep learning that artificially simulates the activity of the human brain for the computer to detect, classify and recognize the object with the use of input data, formula of different technique and bias.

Neural network in Deep learning consists of 3 layers part and each part will have one or more layers according to a preset or model used. The first part is called input layer, it brings the data into the system for further processing by the next layer. Second part is hidden layer. It is usually integrated into 3 further stages.

1) Convolutional stages – the stage that uses the kernel filter to pull out the main feature of the picture that is being attached in that moment. The filter will start scanning from the top left or the beginning point and end when it reaches all the area of the input, using the stride to determine how many areas the filter will move at a time.

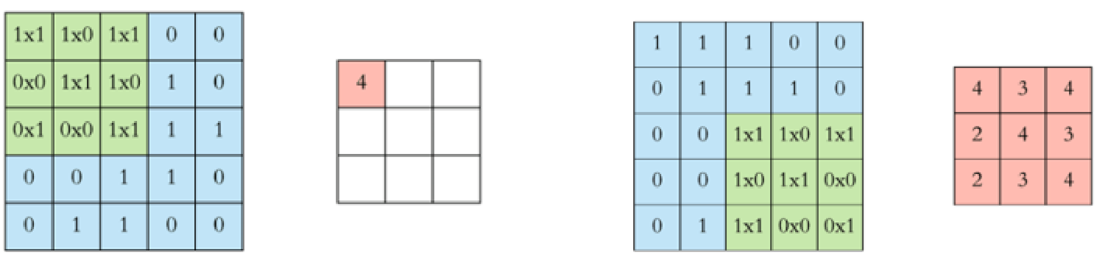


Figure 1Convolutional stage Start and Stop

Reference: https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2

2) Pooling stage - It received the output from the convolutional stages and extracted the main features from it. There are 2 ways of pooling which is max pooling and average pooling. Max pooling brings out the most value that appeared when the filter has scan, while the average pooling finds the average value that was found by dividing the sum of all value by exact amount of number.

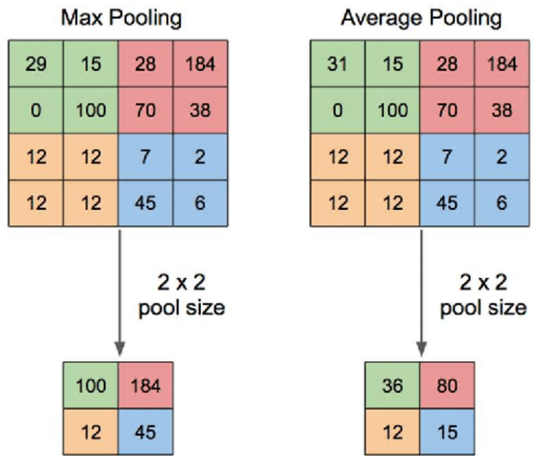


Figure 2 Max pooling and average pooling

Reference: https://www.researchgate.net/figure/Illustration-of-Max-Pooling-and-Average-Pooling-Figure-2-above-shows-an-example-of-max_fig2_333593451

3) Fully connected stages - Decide if the neuron will be activated or stay put using the fullu connected layer along with activation function. The function will calculate the weighted sum and add the bias for the calculation. This is used to convert an output into non-linearity form.

2.1.1. Sigmoid Function

One of the effective functions is that deep down the input into 2 function between 1(yes) or 0(no). and easy to find the derivative.

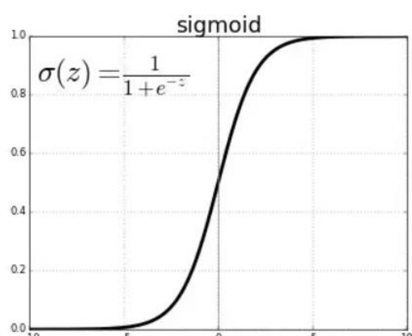


Figure 3 Sigmoid Activation function

Reference: https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6

$$\text{Sigmoid function: } S(x)=1/(1+e^{-x}) = e^{x}/(e^{x}+1) \qquad (1)$$

$$\text{Derivative of Sigmoid function } S'(x)=S(x)*(1-S(x)) \qquad (2)$$

2.1.2. ReLU Function

A popular choice for activation function since it can erase the Vanishing-gradient problem, also easy to find the derivative since the output is only 0 or 1. ReLU may have a disadvantage such as return 0 if the input is negative or the output have no limit(0 to infinity) It may have a difficult time to solve this problem, but since the main problem like vanishing gradient that can't be solved once happen won't occur if using ReLU It considered to be a beneficial trade-off.

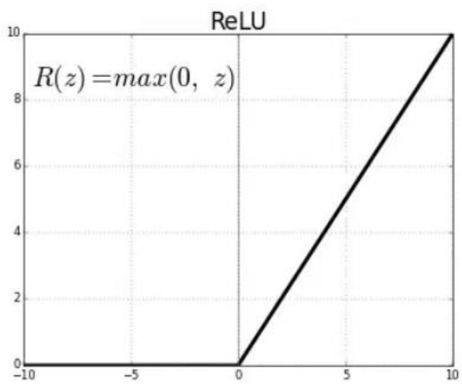


Figure 4 ReLU Activation Function

Reference: https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6

$$\text{ReLU Function: } f(x)=\max(0, x) = \begin{cases} 0 & \text{for } x \le 0 \\ x & \text{for } x > 0 \end{cases} \qquad (3)$$

$$\text{Derivative of ReLU Function: } f'(x)= \begin{cases} 0 & \text{for } x \le 0 \\ 1 & \text{for } x > 0 \end{cases} \qquad (4)$$

After going through the process of hidden layer, the last layer is the output layer, which will give out the output based on input data and what we want to find.

2.2. Process of building a model of deep learning

2.2.1. Gather the data

Choose the data based on the expected result. In my project, the data use will be a human face dataset divided into male and female and have their age range classified. I have a thought of focusing on Asian ethnicity but after a second thought, it might create a bias when make a prediction for different ethnicity beside Asian, thus, I decide not to divide the ethnicity.

We can create more images in a different aspect such as tiles, flip, or cropped some part of the image. The process is called Data Augmentation which is used to increase the amount of dataset and make the model able to detect the image in various position. It is also used to prevent bias that might occur because of the insufficient amount of image files.

2.2.2. Preparing the data

To make the training result fast and accurate, editing the data is one of the methods that will affect the result. The data set may not be in a form to be properly used for training. So, we have to convert them into the state that will be efficient to the machine, both in its speed and performance.

2.2.3. Convert the image into gray scale

Deep learning only takes the structure in the image into account, in other words, it does not let the factor of color affect it prediction. With this fact, we can reduce the color of image to grayscale to increase the efficiency of machine performance for faster train and test speed. The value of an image after gray scale conversion will be the range of 0(White) to 1(Black) and a several shades of gray depend on what the original image are. The output value of pixels in grayscale images can be created by using the formula below.

$$\text{Average Method Grayscale} = R/3 + G/3 + B/3 \tag{5}$$

$$\text{Weighted Method Grayscale} = 0.299R + 0.587G + 0.114B \tag{6}$$

R, G, B is the red, green, blue color pixels of the image, respectively

2.2.4. Image Normalization

A process that changes the range of pixel's intensity into different values but contains the same ratio for other pixels in the images. I divide the values of image pixels by 255 to rescale the pixel values of an image to a fixed range of 0 to 1. Doing this will result in better contrast detection of a model, lower over fitting problem, and compatibility of the model that require a fixed size of image.

2.2.5. Rescaling the Image

Some Images have a larger or smaller size than the others, resulting in error or non-realistic prediction. We solve the situation by first rescale the image into an appropriate size.

2.2.6. Choosing a model

1) VGG 16 - I chose this model because it includes deep architecture with many layers of convolutional and pooling features fit for age and gender prediction. It also trained with a large image dataset, improving its performance and might reduce the time used for training for my model. VGG-16 is a famous and classic model that has been used as a based model for various projects, thus the performance is trustworthy and can be further adjusted with various methods. The last reason is I want to perform a transfer learning, and VGG-16 is a decent choice for using as a based model. By freezing the top layer and perform a fine-tuning method, A promising result can be expected.

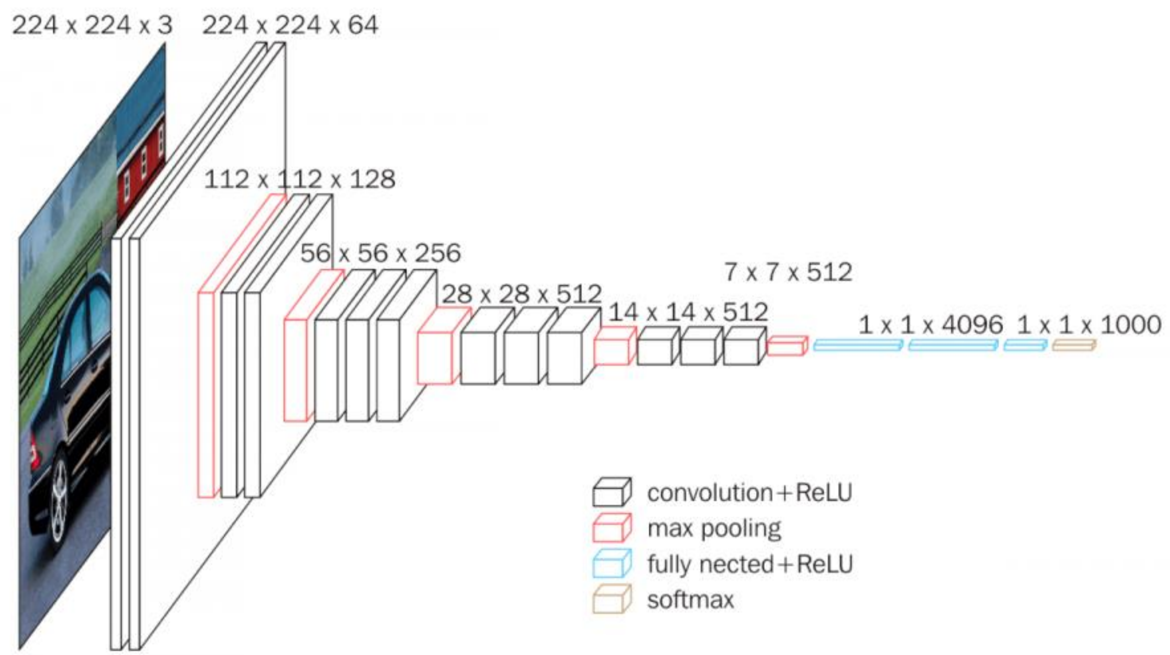


Figure 5 VGG-16 Architecture

Reference: https://neurohive.io/en/popular-networks/vgg16/

2) Neural Network: The project focused on the composure of human faces such as their facial, face structure, hair and stain. Neural networks are suited to these tasks due to their ability to learn these features from the dataset and optimize the parameter for better prediction results.

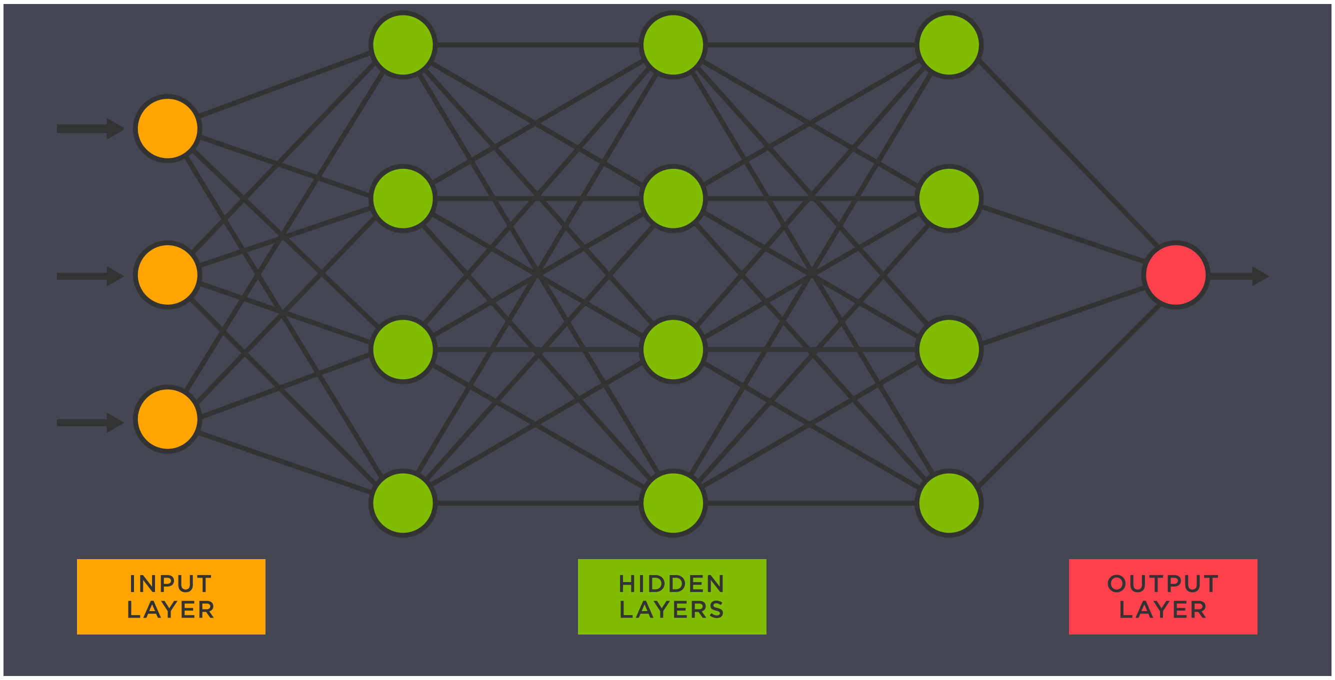


Figure 6 Neural Network Structure

Reference: https://www.tibco.com/reference-center/what-is-a-neural-network

3) Convolution Neural Network - As known as CNN. It is a type of neural network that has been developed with Neural Network as a based model. The model was designed to be more focused on a divided part of the images and take their features into calculation. I chose this model to compare it with the rest with the expectation of its performance to be at least better than based neural network.

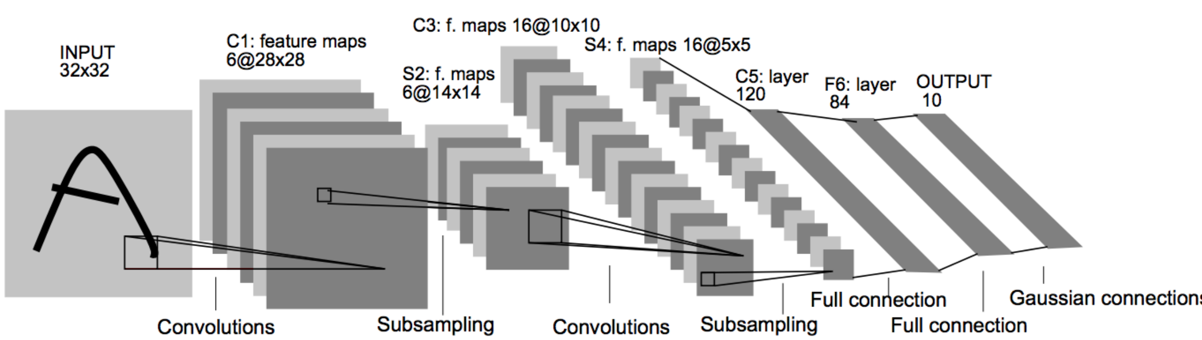


Figure 7 CNN Architecture

Reference: https://www.jeremyjordan.me/convnet-architectures/

2.2.7. Hyper parameter tuning

Adjust the parameter of a model to be fitted in the observation. There are several methods for Hyper parameter tuning as follows.

2.2.8. Neural network layer

The parameters such as neural network layer, Activation layer, learning rate, loss function need to be adjusted to make it compatible to each of the models. This method will determine the value of each parameter that we want to test. We then run the model for every possible parameter. The training round or 'Epoch' also needs to be set for the most fit and precise outcome.

2.2.9. Learning Rate

It will affect how many steps the model takes for it to learn the dataset. Setting it too low or high will cause a model to adjust the weight larger or smaller than it should be, resulting in a high number of losses of the model. The learning rate should be adjusted little by little or automatically adjusted during model training.

2.2.10. Initialization Method

Initialization is the process of setting the initial values of the parameters such as bias and weight in a neural network. The method's purpose is to determine the initial values of these parameters. There are various Initialization functions that have their forte and their choice of use depends on the problem we focus on.

1) Zero initialization - This method initializes all weights and biases to zero. While this may seem like a simple and intuitive initialization method, it can lead to problems such as all neurons give output as zero. Each neuron in the same layer will compute the same linear combination of the input, as a result, the neurons won't be able to differentiate between different features. This method is not recommended in machine learning, especially the model that requires learning.

2) Glorot normal initialization (Xavier normal initialization) - Initialization method which initializes the weights using a normal distribution with zero mean and variance calculated based on the number of input and output units in the layer. The variance of the output is equal to the variance of the inputs. It is default model for convolutional layers in Keras if no initialization method is specified.

3) Glorot uniform initialization (Xavier uniform initialization) - Initialization method which initializes the weights using a uniform distribution within a specific range that is calculated based on the number of input and output units in the layer. This method is designed to ensure that the variance of the outputs is roughly equal to the variance of the inputs. It is default model for most layers in Keras if no initialization method is specified. This initialization method is the best fit for my model due to the performance of model output.

4) He normal initialization (Kaiming Initialization) - Initialization method which initializes the weights using a normal distribution with zero mean and variance that based on the number of input units in the layer. This method is usually used with ReLU activation functions for further avoiding exploding gradient problem, which is the case that the gradients become very large during backpropagation, making the model train unstable.

5) He uniforms initialization - This method initializes the weights using a uniform distribution within the range that is calculated based on the number of input units in the layer. This method is also effective to be used with ReLU activation functions.

2.3. Model architecture modification

Design the model structure to be compatible with the Data set and also the method of training that will be used.

2.3.1. Dense and Convolutional layer

Both are used for feature extraction a learning the dataset pattern. They contain a trainable parameter that can automatically optimized during training the model. An Activation function and Initialization Method can be adjusted for better performance of the model.

2.3.2. Activation Function

We use the activation function depending on the expected output to be in the difference form depend on what we focused such as 'Sigmoid' activation function will return the output in the range of 0 to 1 or 'Linear' activation function will return output depend on the information's slope and bias. Using activation function also caused the model to learn more complex and detailed pattern of the input. Making the learning progress of the model to be in a form of nonlinearity.

2.3.3. Flatten Layer

Flatten layers are used to convert the multi-dimensional input into 1 dimensional array. Resulting in conversion of fully connected layer to output of convolutional or pooling layer. With this, we can get the single output we expected.

2.3.4. Output layer

Create an output of model that can be either labels or values depending on the task. In my project, I use a regression method for my prediction result, thus the output will be in a form of single value.

2.4. Training the model

This step will make the model learn about set of weights and biases. Making them able to predict the expected outcome with precise accuracy.
The purpose of model training is to find a set of model parameters that can predict the output variable given the input features. There are a several steps in model training.

2.4.1. Data preparation

Split the prepared Dataset into training and testing Dataset for the model to be trained with. Also setting the shape of the model to match our dataset shape is a must.

2.4.2. Model selection

Select the model that will be used to train our dataset, the model should be able to categorize the pattern of the model's feature. The more effective the model is, the better the predicted result will be.

2.4.3. Parameter initialization

Initialize the model parameter (or weights) to the values. The parameter can be set randomly, or it can be set to recommended value by the pretrained model we used.

2.4.4. Model compile

Choose a loss function, optimizer, and metrics that will be used to evaluate the model's performance during training. They are responsible for specifying the model performance during each epoch in training progress. These mentioned techniques or methods are used for pointing the model performance in every epoch that are progressed.

1) Loss Function - A function that calculates the difference between the predicted output of the model and the actual output. A good model should have loss function as low as possible.

2) Optimizer - Algorithm that adjusts the weights and biases of the model during training to decrease the loss function. There are different types of optimizers which their usage is dependent on the circumstance of which kind of problem that the model is currently dealt with.

3) Metrics - Technique for specifying the model training performance, the type of metrics will be varied by a type of model output, whether it's a classification type model or regression type model. Common metrics for regression problems include mean squared error (MSE) and mean absolute error (MAE).

2.5. Model Evaluation

A procedure to observe whether the model prediction performance reaches the expected result or not. The evaluation method is varied by the goal of the project, which can be 'Right or Wrong (Accuracy)', 'Range between Actual and Predicted value (MAE)' and etc.

1) Confusion Matrix - A table to evaluate the performance of the model, The size of the table will be compared with the predicted result and actual result. By using Confusion matrix. We can evaluate the result of the model by judging from the right and wrong answer. The higher amount of 'True Positives' and 'True Negatives', the higher the performance of the model is.

	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	True Positives (TPs)	False Positives (FPs)
Predicted Negative (0)	False Negatives (FNs)	True Negatives (TNs)

Figure 8 Confusion Matrix

Source: https://medium.com/@pagongatchalee/confusion-matrix-
%E0%B9%80%E0%B8%84%E0%B8%A3%E0%B8%B7%E0%B9%88%E0%B8%AD%E0%B8%87%E0%B8%
A1%E0%B8%B7%E0%B8%AD%E0%B8%AA%E0%B8%B3%E0%B8%84%E0%B8%B1%E0%B8%8D%E0%B
9%83%E0%B8%99%E0%B8%81%E0%B8%B2%E0%B8%A3%E0%B8%9B%E0%B8%A3

1.1) True Positive (TP) - Both prediction and actual result is match with the first condition

1.2) True Negative (TN) = Both prediction and actual result is match with the second prediction

1.3) False Positive (FP) = The prediction doesn't match with the actual result

1.4) False Negative (FN) = The actual result doesn't match with the prediction

The TP,TN,FP,FN can be use to calculate the Accuracy, Precision, Sensitivity, Specificity, F1-score with the formula mentioned below

Accuracy = (TPs + TNs) / (TPs+TNs+FPs + FNs) (8)

Precision = TP/(TP+FP) (9)

Sensitivity = (TP)/(TP+FN (10)

Specificity = (TN)/(TN+FP) (11)

*F1-score = 2 * [(Precision * Sensitivity)/ (Precision + Sensitivity)* (12)

2) Mean absolute error - In short for MAE. It is a method used to evaluate the average difference between predicted values and the actual values. I used it to evaluate my age model performance because to the ages ranged is very difficult to pin-point the exact values dues to various factor such as Similar face structure of babies, those who have their faces not developed along with their ages, achene, or stain on their faces, etc. Thus, I will use the MAE to indicate overall difference of the prediction result of my model and the actual ages of the Dataset.

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |Xi - X| \quad (13)$$

X_i and X can be swapped since after differentiation of each, the value will be convert into positive value before being kept in the formula.

3) Percentage difference - A method to calculate the difference between predicted output and actual output in terms of percentage. It is useful to indicate the model output that can't be evaluated by ordinary right or wrong correction method. The output percentage difference inversely varied to the model performance.

$$Percentage\ difference = \frac{(|Value1 - Value2|)*100}{(\frac{(Value1+Value2)}{2})} \quad (13)$$

Value 1 and Value 2 are the predicted output and actual output, regardless of the order.

Due to the problem of a gap between predicted age and actual age can rise up to 50% even though the value of predicted age and actual age are just a single digit, the formula for Percentage difference is being adjusted to

$$Percentage\ difference = \frac{(|Value1 - Value2|)*100}{(Max\_age)} \quad (14)$$

Max_age refers to the maximum number in the age range, the formula computes the percentage difference between the predicted ages by scaling the absolute difference by the maximum age value.

3 Material

3.1. Hardware

3.1.1. Notebook

Type and adjust the code for Create a deep learning model.

Figure 9 My Notebook

Processor: 12th Gen Intel(R) Core (TM) i7-12700H 2.30 GHz

Ram: 16.0 GB

3.2. Software and Website

3.2.1. Google Colab

Colaboratory, or ‘Colab’ for short, is a product from Google Research. The website has a clear user interface and also decent to execute the code. With it use the Python language for executing the code, Colab is an effective choice for anyone who wants to start creating their deep learning project or to run the code for learning and practicing. Colab is a hosted Jupyter notebook service that requires no setup to use, while providing access free of charge to computing reReferences including GPUs

Figure 10 Google Colab logo

Reference: https://commons.wikimedia.org/wiki/File:Google_Colaboratory_SVG_Logo.svg

3.2.2. Python Language

Python is a high-level, general-purpose programming language created by Guido van Rossum. It's an easy-to-understand language with a lot of flexible, fast, have a lot of supportive community and have a massive size of standard library.

Figure 11 Python Logo

Reference: https://www.python.org/community/logos/

3.2.3. Kaggle

Kaggle is a subsidiary of Google LLC, an online community of data scientists and machine learning practitioners. Kaggle allows users to find and publish data sets, explore and build models in a web-based data-science environment. I use this website to study code structure and how each process of code combines and affects each other.

Figure 12 Kaggle Logo

Reference: https://en.wikipedia.org/wiki/Kaggle

4 Methods

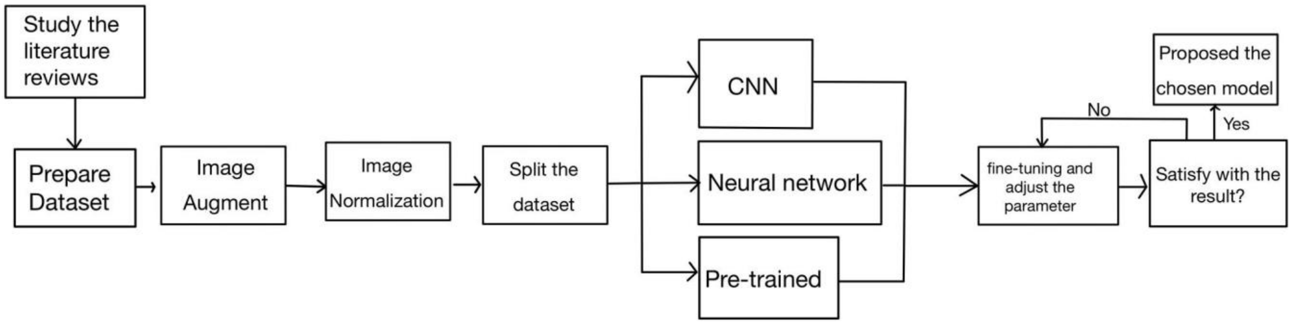


Figure 13 Proposed methodology chart

4.1. Study on Literature review and Research

Study about model training and the method fit to use with age and gender prediction. The literature reviews I got from browsing the internet are as follows.

1) Age and Gender Prediction from Face Images Using Convolutional Neural Network

2) Age Estimation Based on Convolutional Neural Network

3) A cascaded convolutional neural network for age estimation of unconstrained faces

It teach me how model training works, which model should be effective and adaptable for my project and a rough idea of how I should create my model.

4.2. Prepare the dataset

After browsing through Kaggle to find a dataset and testing with via Google Colab. I conclude that UTK face from the Kaggle website contains an ideal images file from the dataset. It contained 23708 faces of Male and Female Human. Because All of them are straight faces which are fit to train with my age and gender prediction model. The files have been arranged by their ages, gender, ethnicity, and file names respectively. Since my project only focused on age and gender, I can easily collect the former two into a variable and ignore the rest. The shape of both train and test Data set is (None,224,224,1) and (None,224,224,3) depending on the model requirement.

The amount of images file I intended to use to train my model is 25 for both male and female with the age range of 1 to 90, reaching the total of 4500 image files that will be used to train age and gender separate model. For combine model, I reach a conclusion that even though I perform an image augmentation, the amount of original file in the UTK Face are too few that it might cause the model to learn the same pattern of image and also it might create a unnecessary burden to my Colab's GPU resource since the face of old age human are likely to be have similar, thus, I change the max age range from 90 to 79 for combine model, receiving a total of 3750 image files.

Table 1 Amount of Image Files for each model type

Type of Model	Age Range	Files per Male and Female	Total files
Separate Age and Gender	1-90	25	4500
Combine Age and Gender	1-79	25	3750

4.3. Image Augmentation

Although Most of the Dataset are in a decent quality. But there were not enough files for some ages to be used properly. Due to the majority of images of UTK face are in approximate range 0-50. I have to perform an image augmentation on the ages that was lacking image files. The operation I used for augmentation is image enhance for brightness and contrast since crop, rotate or flip the image files might cause a confusion for the machine.

4.4. Image Normalization

I decided to normalize the image by dividing their pixel values by 255 to prevent overfitting and not taking too much resource for my Google Collab. Normalization with this method can help for faster processing time and make the training proceed with a stable rate.

```
[ 20.,  22.,  52.]      [0.07843138, 0.08627451, 0.20392157]
[ 12.,  14.,  44.]  --> [0.04705882, 0.05490196, 0.17254902]
[  8.,  10.,  40.]      [0.03137255, 0.03921569, 0.15686275]
```

Figure 14 A part of pixels value before and after normalization

4.5. Split the Dataset

Splitting my Dataset into Train and Test. Is set the ratio for training dataset and testing dataset to 8:2. This ratio has been used for all 3 of my models. The Total number of training dataset and testing dataset for separate model and combine model are shown as the following table.

Table 2 Number of image files for unproposed age and gender separate model

Model	Training Dataset	Testing Dataset
Neural Network	3160	790
Convolutional Neural Network	3160	790
VGG-16 Pretrained+Fine-tuning	3160	790

Table 3 Number of images for proposed age and gender separate model

Model	Training Dataset	Testing Dataset
Neural Network	3160	790
Convolutional Neural Network	3160	790
VGG-16 Pretrained+Fine-tuning	3160	790

4.6. Age and Gender model

This step will make the model learn about a set of weights and biases. Making them able to predict the expected outcome with precise accuracy. The input shape will be 224,224 on both width and height while the color channel depended on the model requirement. The Output for gender is within the range of 0 to 1 before putting in a function that will display the gender. The result of age on the other hand will be directly shown. The epoch and batch sizes of the model will be adjusted according to the results and GPU resource limit. All of the models are using regression method.

I design a Neural Network and CNN model on my own. After that I import a VGG-16 model from Keras then make an adjustment. I tried to add pooling layer and convolutional layer for VGG-16 pretrained model, but the result is lower than the one I already got, so it left me no choice but to discard them. I tried to change the initialization method to He and Glorot. After running the code and checking for the result, I found out that not specifying the initialization method and leave them to its default method give the best result for models.

The procedure of my 3 models is using the same method except the structure of each different model. Their procedure works as the flow graph shows.

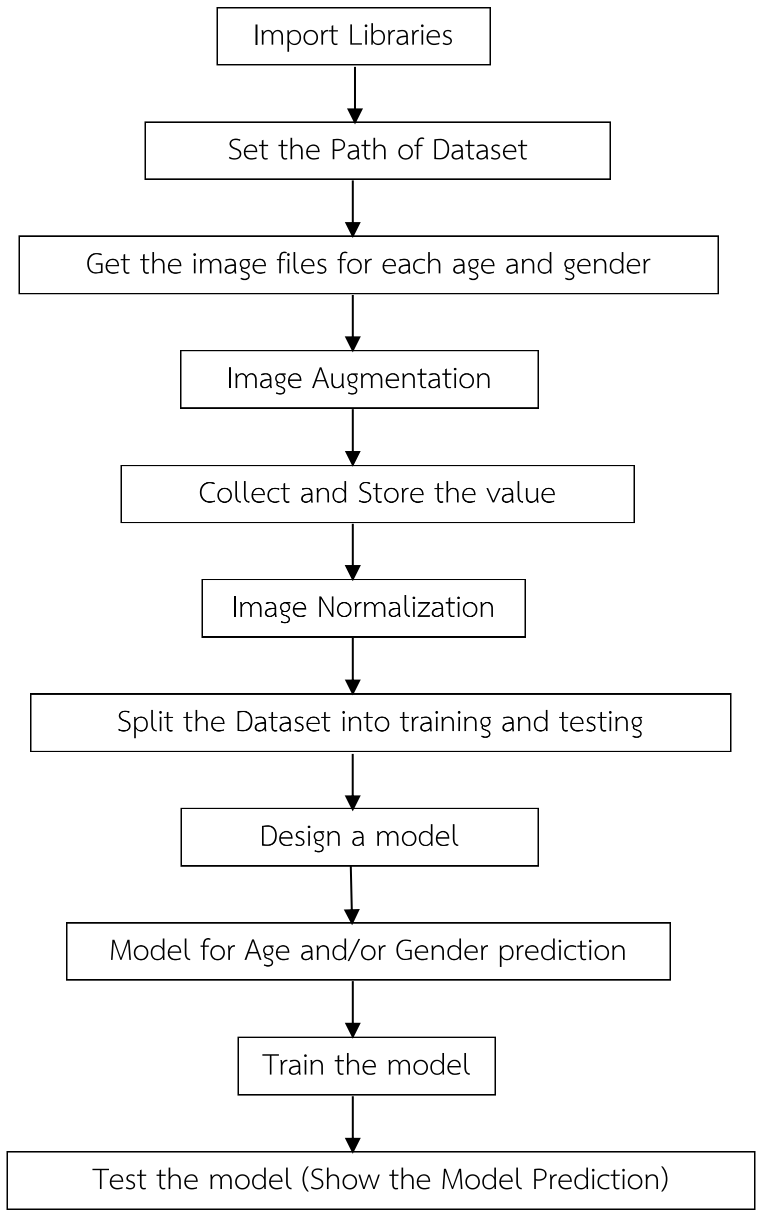


Figure 15 Process pf algorithm in Age and/or gender separate model

4.6.1. Unproposed Age and Gender Separate model

To test whether the model really works as it should be, I choose to create the age and gender model separately from the other to reduce both complexity of the task and the resource used for running the model. The order of the flow chart is Age model for Neural Network, Gender Model for Neural Network, Age model for CNN Gender model for CNN, Age model for VGG-16 fine-tuning and Gender model for VGG-16 Fine-tuning consecutively.

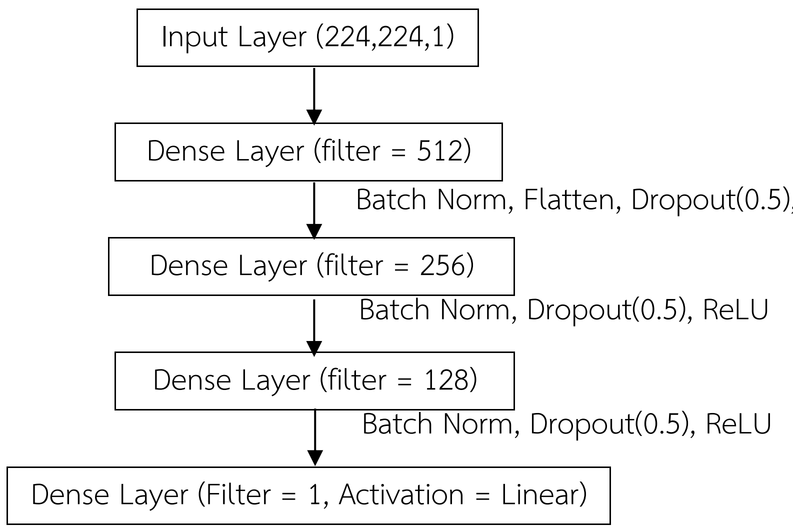


Figure 16 Architecture of unproposed Neural Network age model

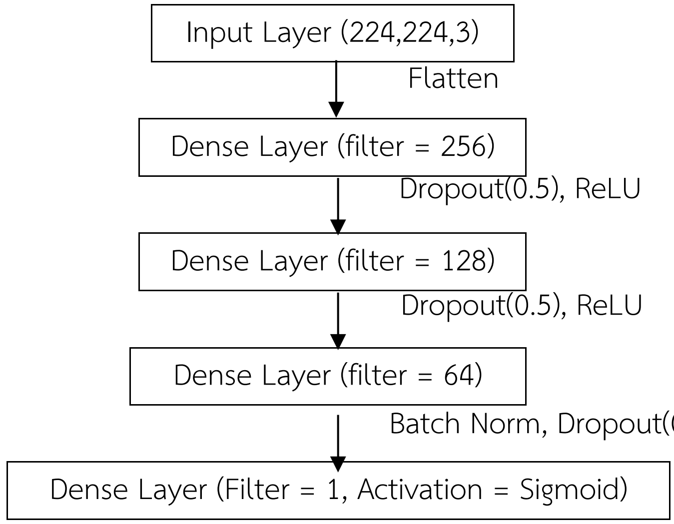


Figure 17 Achitecture of unproposed neural network gender model

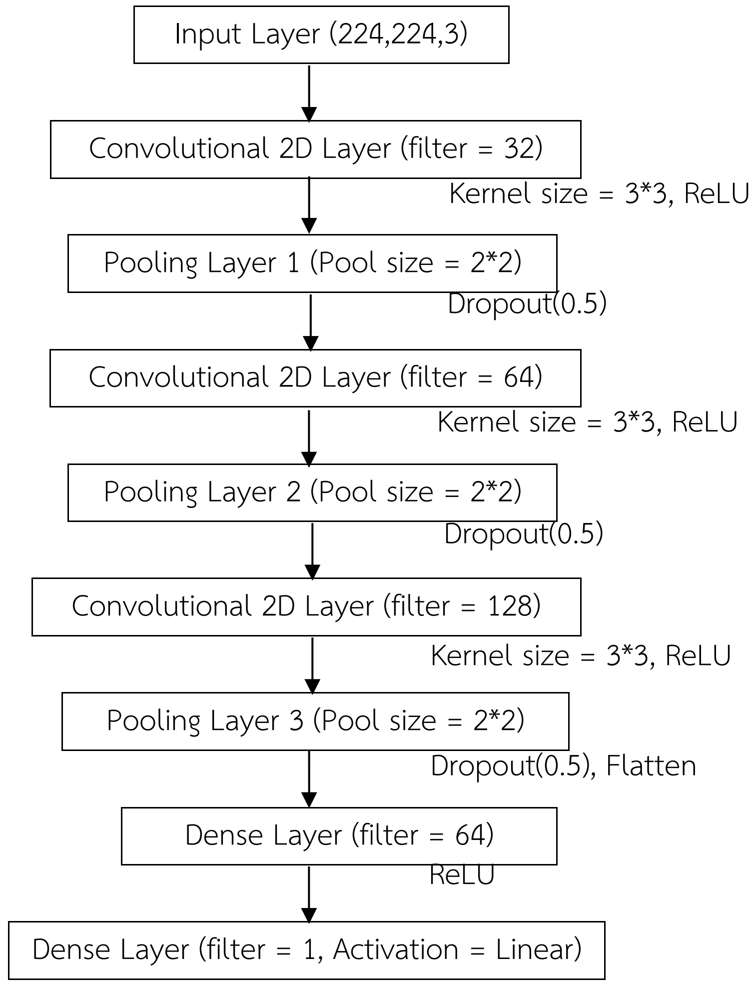


Figure 18 Architecture of unproposed CNN ange model

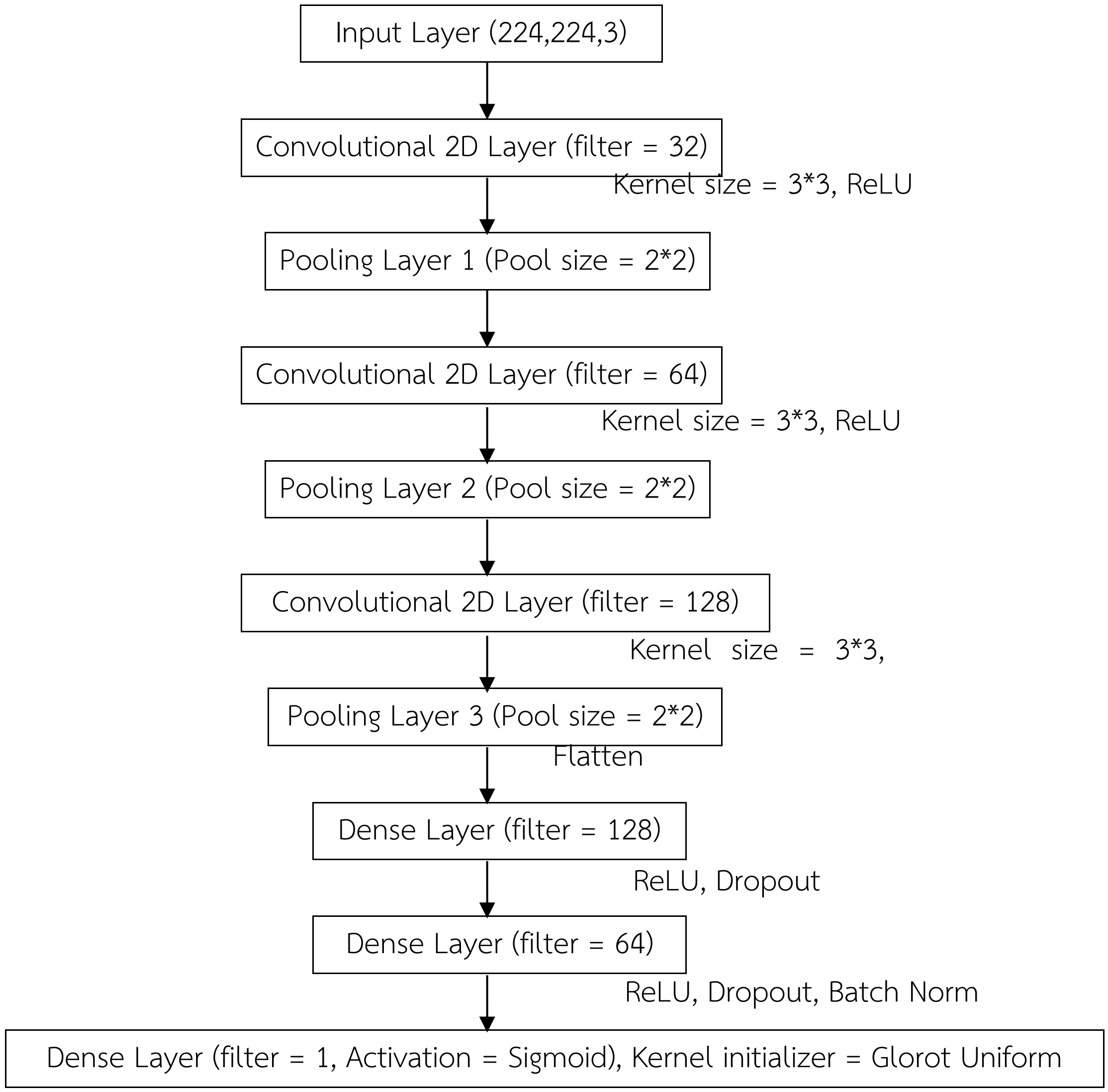


Figure 19 Architecture of unproposed CNN gender model

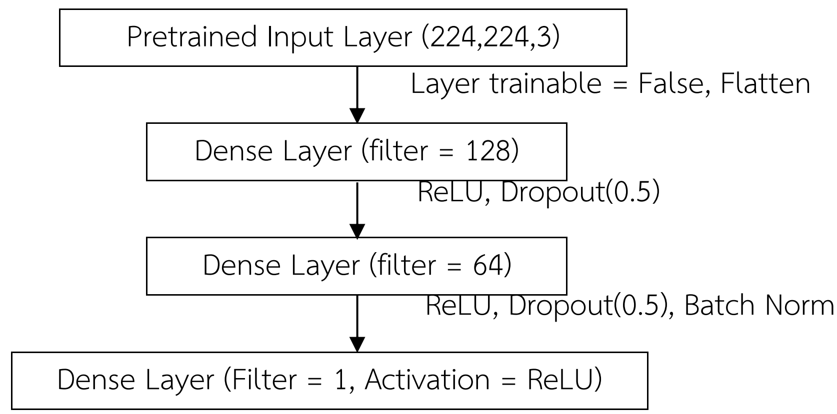


Figure 20 Architecture of unproposed model

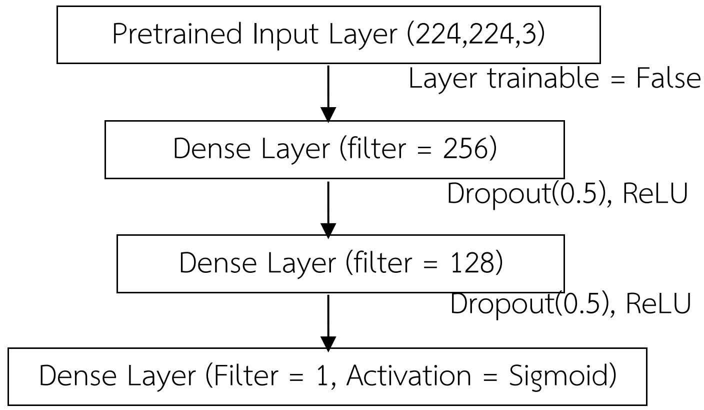


Figure 21 Architecture of unproposed pretrained gender model

4.6.2. Proposed Age and Gender Combine model.

After I completed all the separate model, I move on to the next phase that is to combine both 'Age' part and 'Gender' part together. There was an unexpected error such as lack of model performance and resource usage rate went over limit. A fine-tuning is performed to solve the problem.

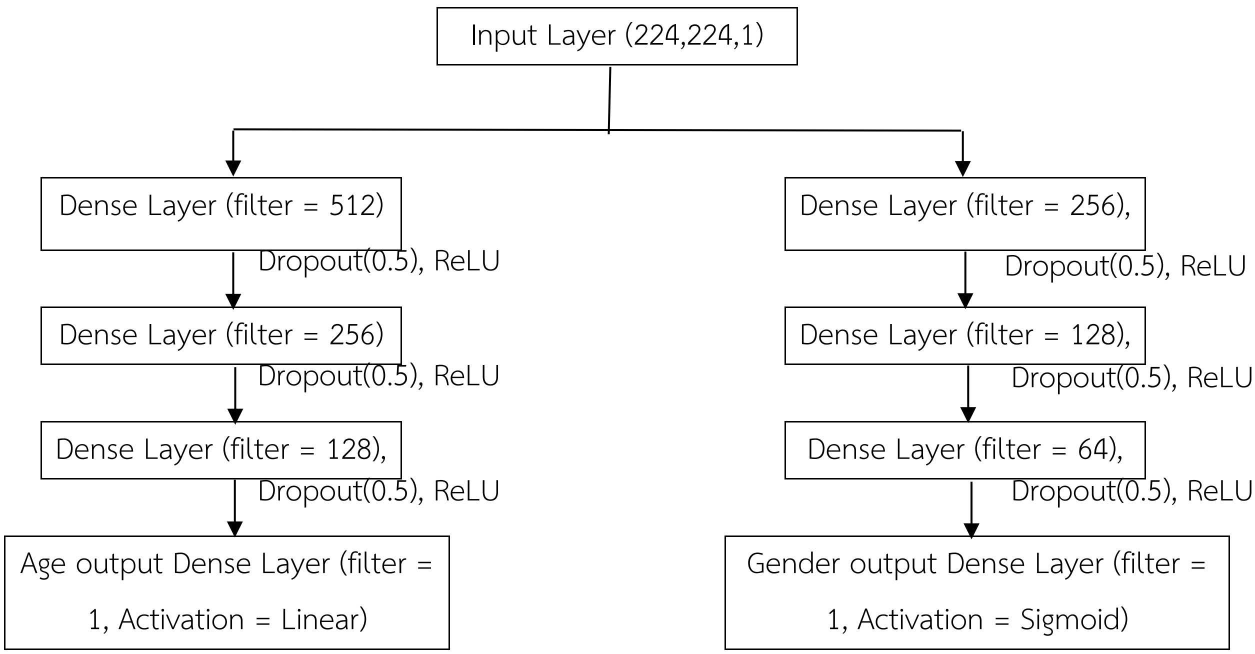


Figure 22 Architecture of Neural Network Combine model

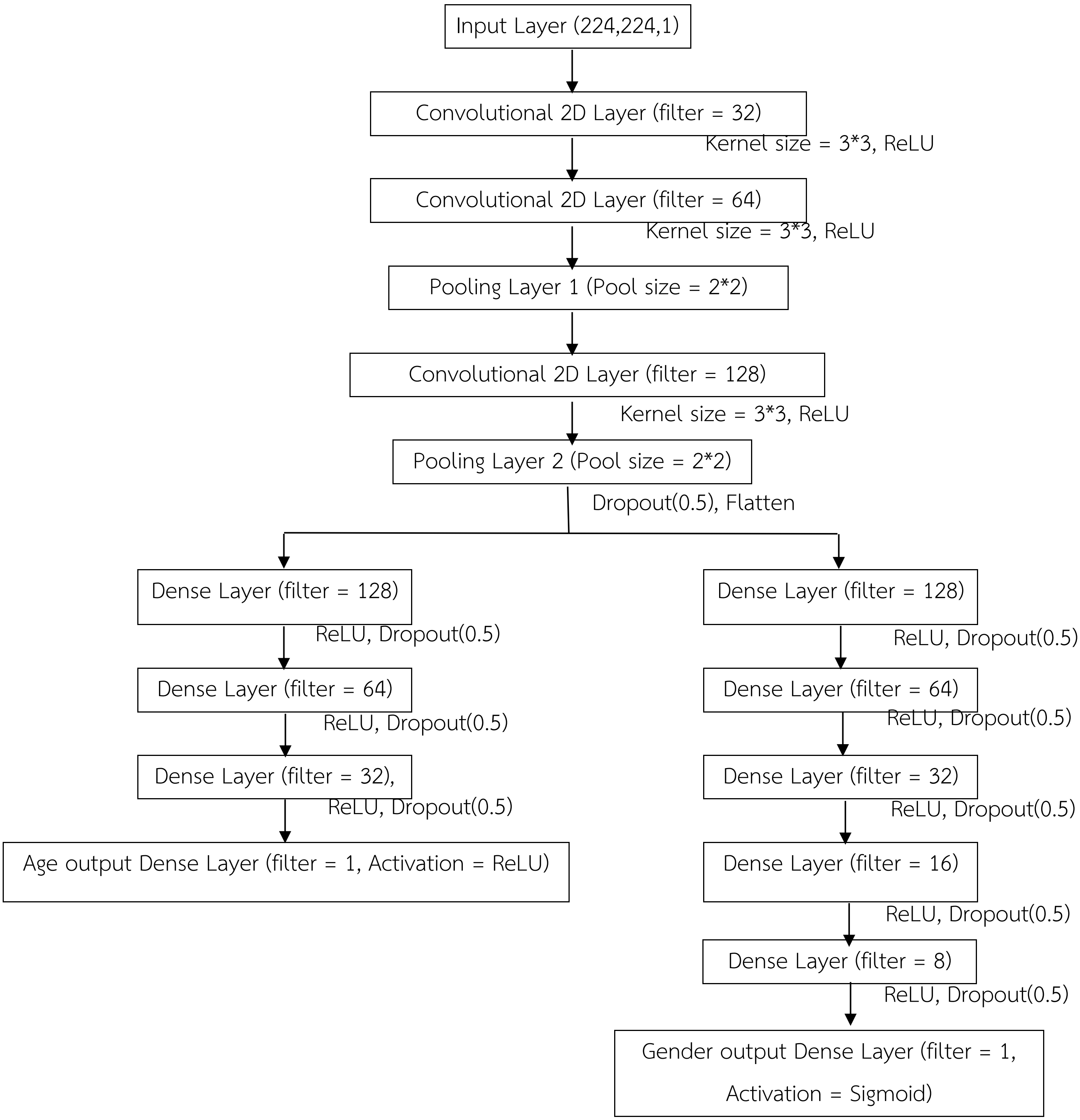


Figure 23 Architecture of CNN combine model

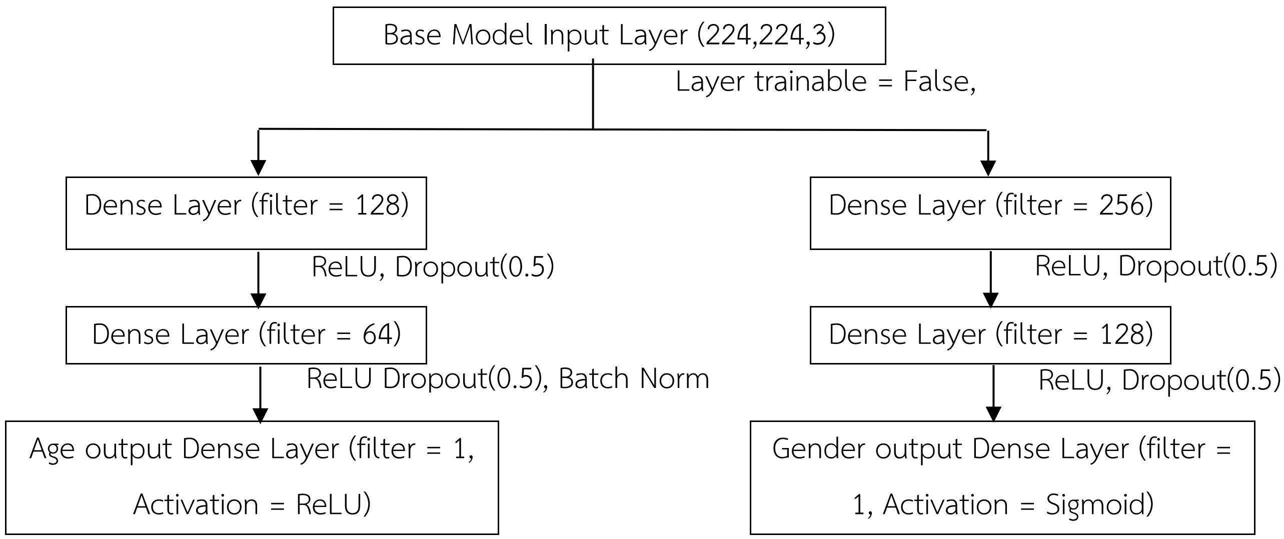


Figure 24 Architecture of Pretrained combine model

4.7. Fine-tuning and Adjust the Model

4.7.1. Epoch

Too few epochs will make a model overfit or underfit, also too many epochs will cause a burden for my Colab GPU resource. since I'm training both age and gender model at the same time, resulting in crashing of the Colab website and reset the training progress and I have to re-run the code all the way from the start.

4.7.2. Batch size

Batch size affects the loss and performance of the model, lowering batch size might made the model train faster but the less complicated it will learn. This variable is the one I manually adjust again and again, judging from the outcome of my model result.

After Trial and errors, I set the Epoch and Batch size of both separate and Combine model as shown in the table below.

Table 4 Epoch and Batch size of unproposed age and gender separate model

Model	Epoch (Age)	Batch size (Age)	Epoch (Gender)	Batch size (Gender)
Neural Network	25	64	45	64
CNN	45	64	45	64
VGG-16 Pretrained+Fine-tuning	45	64	45	64

Table 5 Epoch and Batch size of Proposed Combined Age and Gender model

Model	Epoch (Age)	Batch size (Age)
Neural Network	60	32
CNN	100	80
VGG-16 Pretrained+Fine-tuning	100	128

4.7.3. Learning rate

Adjusting Learning rate is a complicated problem. After searching for a solution, I set the initial Learning rate of separate model to $1*10^{-6}$ and let it automatically decreased with the 'ReduceLROnPlateau'. It will reduce the Learning rate by 0.1 when the validation loss is the same for 3 epochs. After I move to the combine phase of Age and Gender model, I found out that the function 'ReduceLROnPlateau' reduce my model performance for CNN and Pretrained, For Neural Network, it increases the consumption of my resource, so I have to delete the function and let the model run with their own pace.

5 Result and Discussion

5.1. Unproposed model

5.1.1. Unproposed Separate Neural Network model result

Table 6 MAE and Loss of unproposed Neural Network age model

Model	Train MAE	Test MAE	Train Loss	Test Loss	RMSE
Neural Network	11.2588	11.7882	200.6904	236.1201	15.36

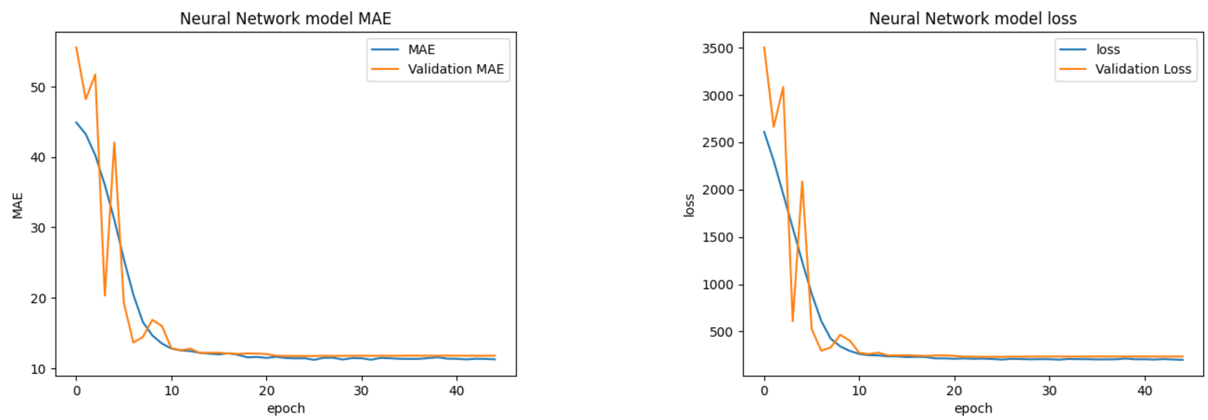


Figure 25 Validation MAE and Loss of unproposed neural network age model

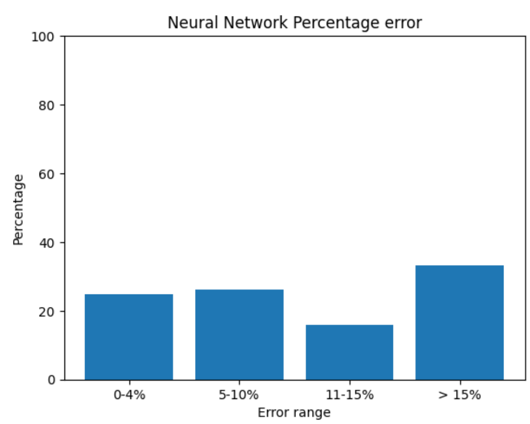


Figure 26 Percentage Error of unproposed Neural Network age model

Table 7 Unproposed Neural Network age percentage error

Range of error	Number of files	Percentage
0-4%	224	24.88%
5-10%	236	26.22%
11-15%	142	15.77%
More than 15%	298	33.11%

5.1.2. Unproposed Neural Network gender model result

Table 8 Accuracy of unproposed Neural Network Gender Model

Model	Train Accuracy	Test Accuracy	Train Loss	Test Loss
Neural Network	0.7875	0.7833	0.4566	0.4356

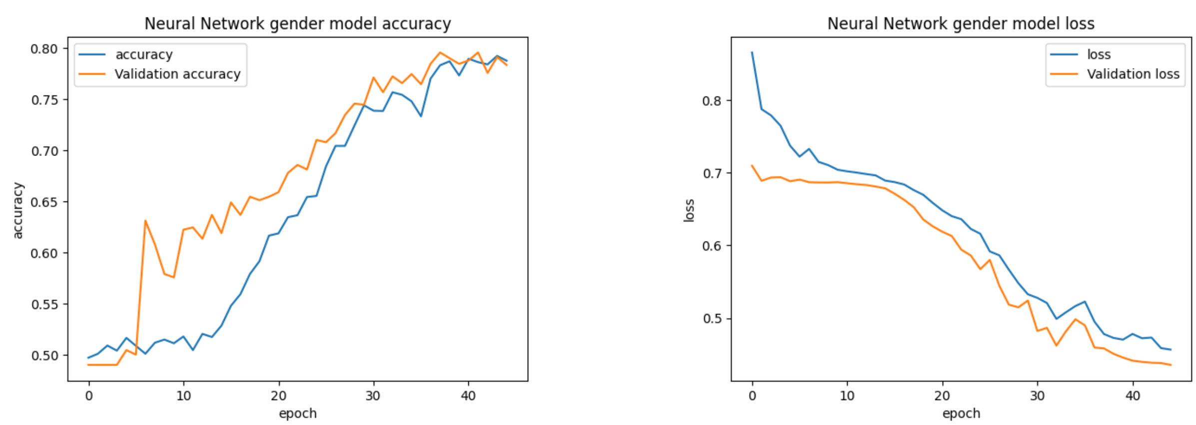


Figure 27 Accuracy of loss graph of the unproposed Neural Network gender model

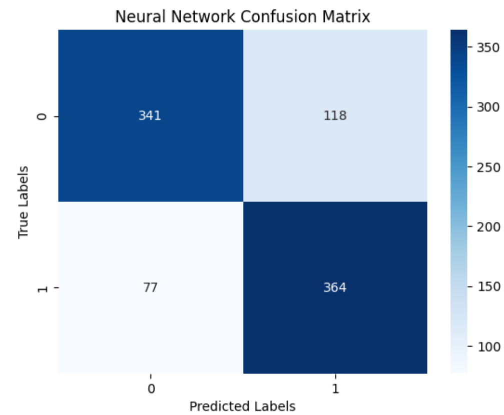


Figure 28 Unproposed Neural Network Confusion matrix

5.1.3. Unproposed CNN age model result

Table 9 Mae Loss and RMSE of Unproposed CNN age Model

Model	Train MAE	Test MAE	Train Loss	Test Loss	RMSE
CNN	13.9538	15.8131	324.2462	367.9556	15.36

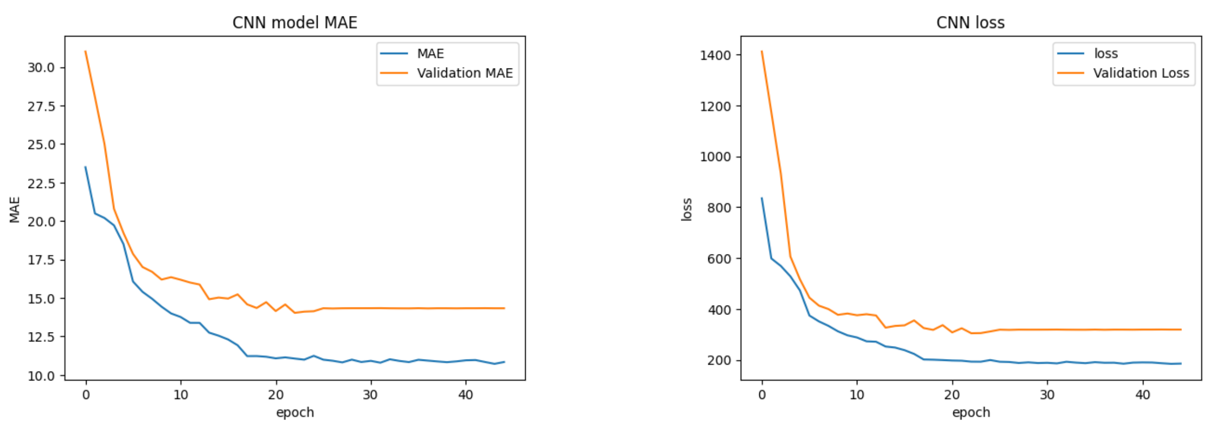


Figure 29 MAE and Loss graph of the unproposed CNN age model

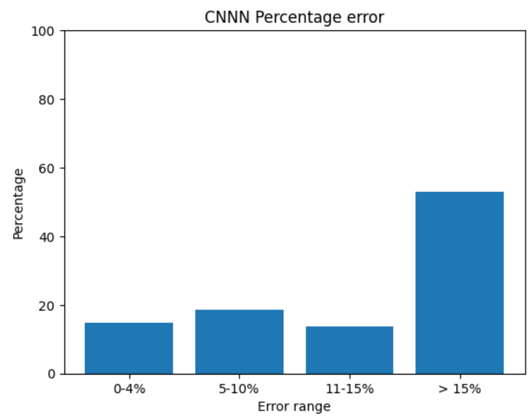


Figure 30 Percentages Error of unproposed CNN age model

Table 10 Unproposed CNN age percentage error

Range of error	Number of files	Percentage
0-4%	117	14.81%
5-10%	147	18.61%
11-15%	108	13.67%
More than 15%	418	52.91%

5.1.4. Unproposed CNN gender model result

Table 11 Accuracy of unproposed CNN Gender Model

Model	Train Accuracy	Test Accuracy	Train Loss	Test Loss
Neural Network	0.9475	0.8506	0.1464	0.3325

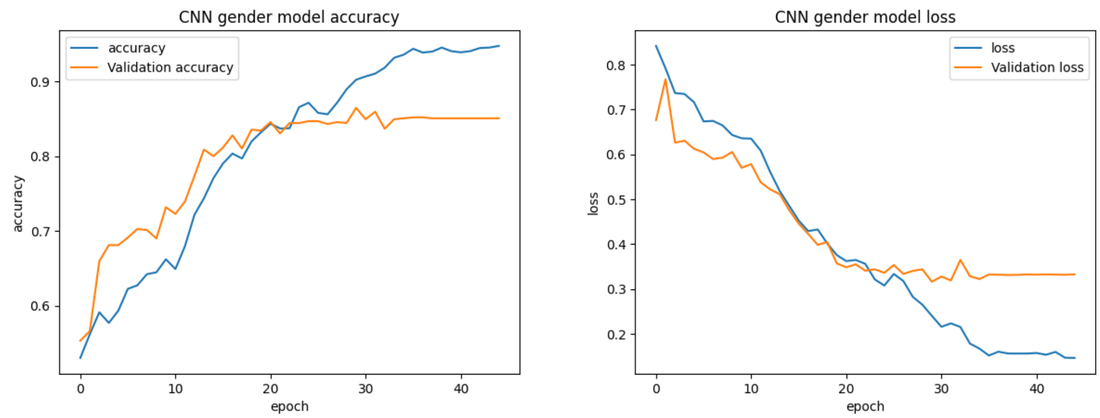


Figure 31 Accuracy and loss graph of the unproposed CNN gender model

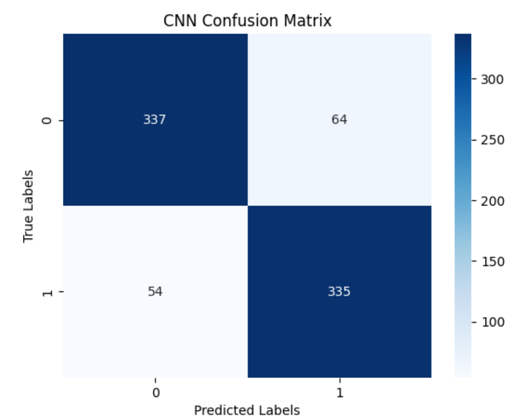


Figure 32 Unproposed CNN Confusion Matrix

5.1.5. Unproposed Pretrained age model result

Table 12 MAE Loss of unproposed Pretrained age model

Model	Train Accuracy	Test Accuracy	Train Loss	Test Loss
CNN	14.1870	12.603	305.3044	231.500

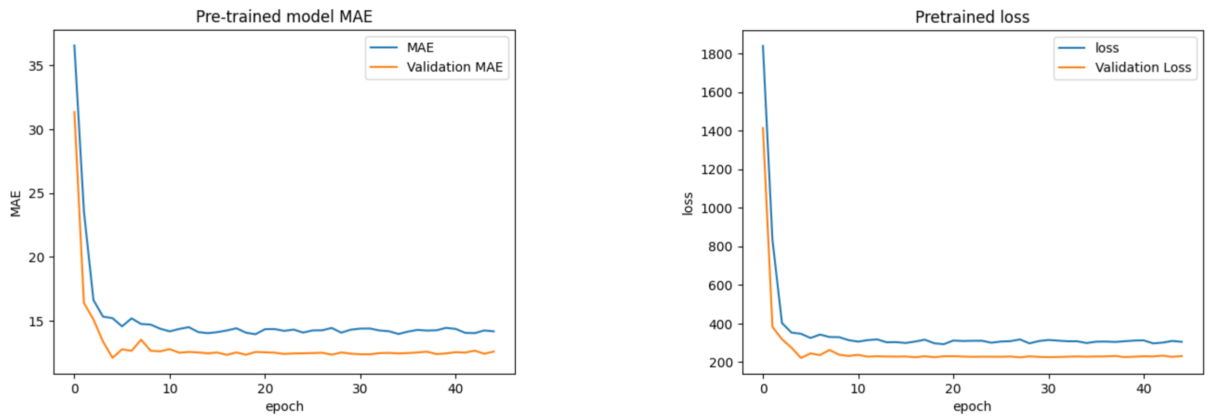


Figure 33 MAE and Loss graph of unproposed Pretrained age model

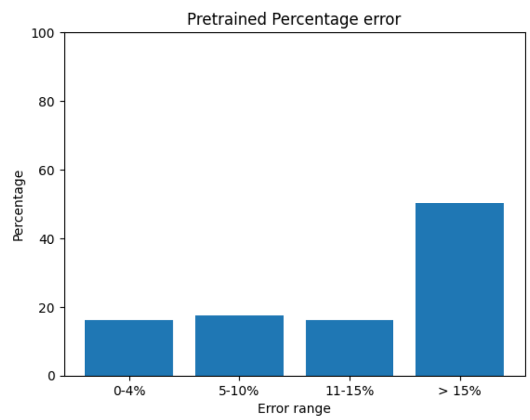


Figure 34 Percentages Error of Unproposed age model

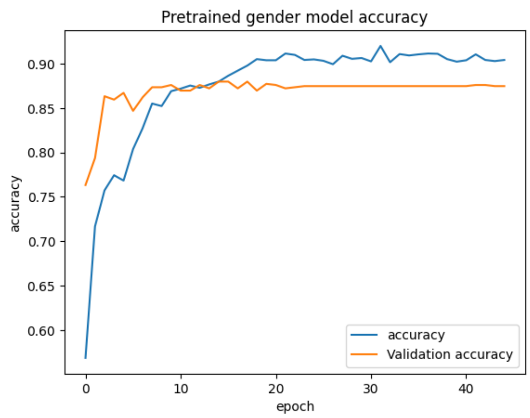
Table 13 Percentage error of unproposed pretrained+fine-tuning age model

Range of error	Number of files	Percentage
0-4%	127	16.08%
5-10%	138	17.47%
11-15%	128	16.20%
More than 15%	397	50.25%

5.1.6. Unproposed Pretrained gender model result

Table 14 Accuracy and loss of unproposed pretrained gender model

Model	Train Accuracy	Test Accuracy	Train Loss	Test Loss
Neural Network	0.9041	0.8747	0.2157	0.2927



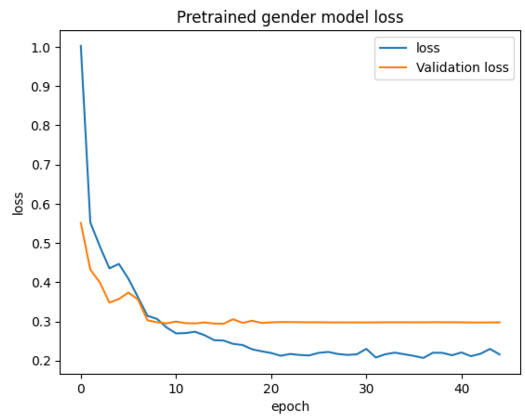


Figure 35 Accuracy and loss graph of the unproposed Pretrained gender model

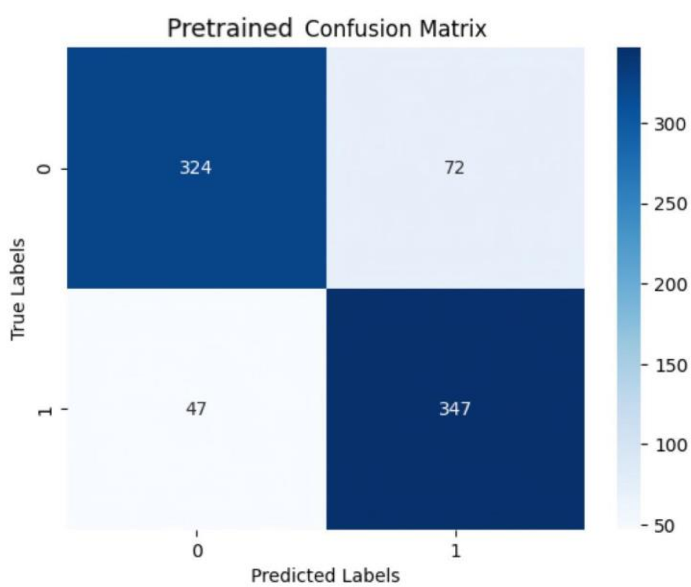


Figure 36 Unproposed Pretrained Confusion Matrix

5.2. Proposed Age and Gender combined model

5.2.1. Neural Network combine model result

Table 15 Mae Loss and RMSE of proposed Neural Network age model

Model	Train MAE	Test MAE	Train Loss	Test Loss
Age	18.4853	17.1355	484.3796	406.8231

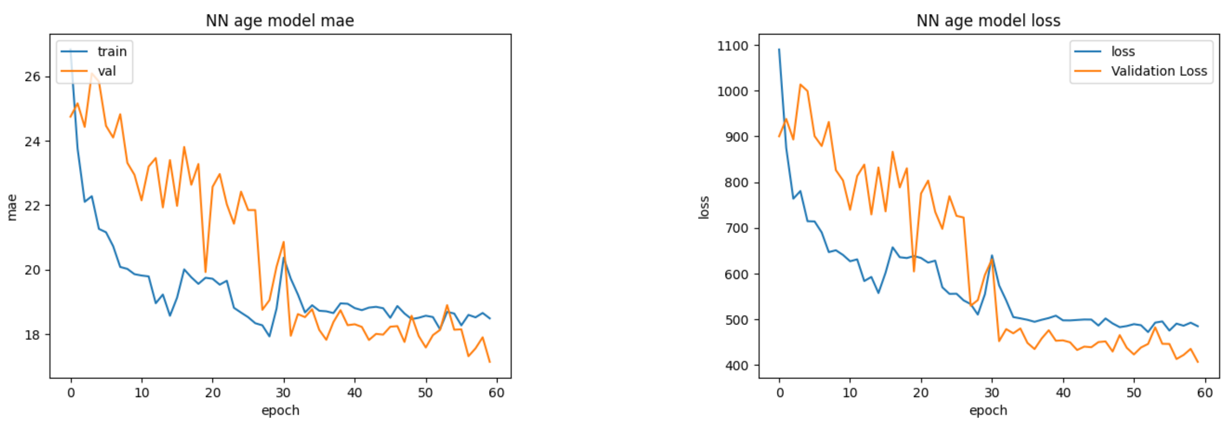


Figure 37 MAE and Loss Graph of proposed Neural Network age model

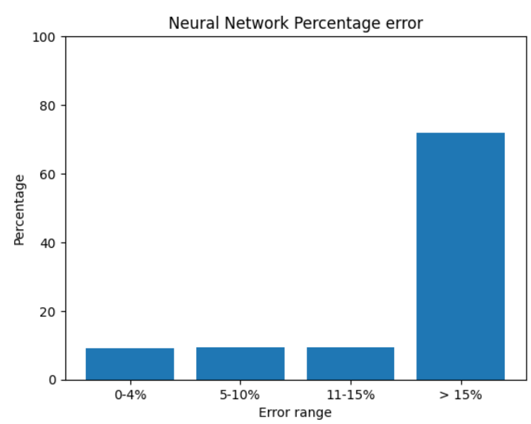


Figure 38 Percentages Error of proposed neural network age model

Table 16 Percentage error of proposed of proposed Neural Network age model

Range of error	Number of files	Percentage
0-4%	117	14.81%
5-10%	138	17.47
11-15%	99	12.53
More than 15%	436	55.19

Table 17 Accuracy and Loss of proposed Neural Network Gender Model

Model Type	Train Accuracy	Test Accuracy	Train Loss	Test Loss
Gender	0.8256	0.7709	0.3805	0.4442

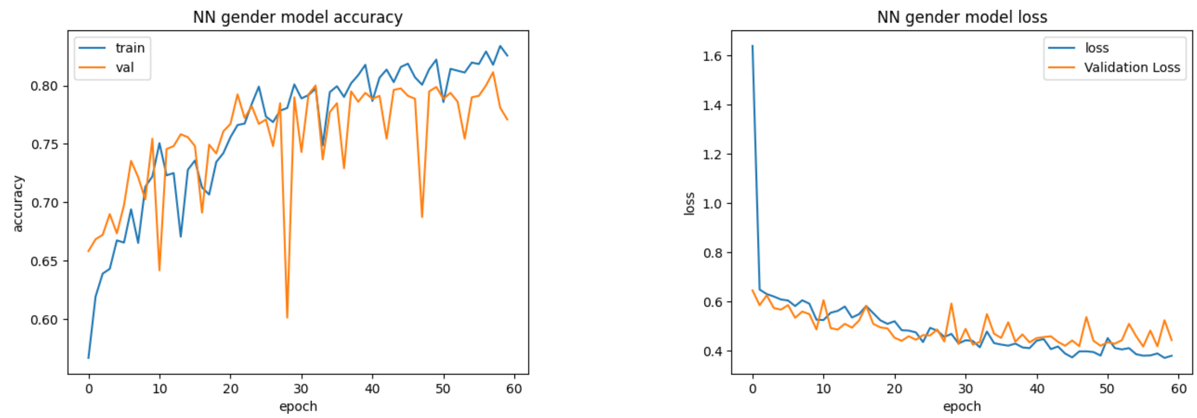


Figure 39 Accuracy and loss graph of the proposed Neural Network gender model

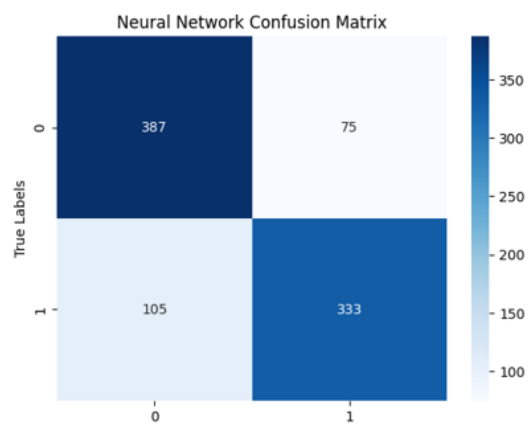


Figure 40 proposed Neural Network Confusion matrix

5.2.2. Proposed CNN model result

Table 18 MAE Loss and RMSE of proposed neural network age model

Model Type	Train MAE	Test MAE	Train Loss	Test Loss
Age	10.5386	15.1492	197.7404	357.4378

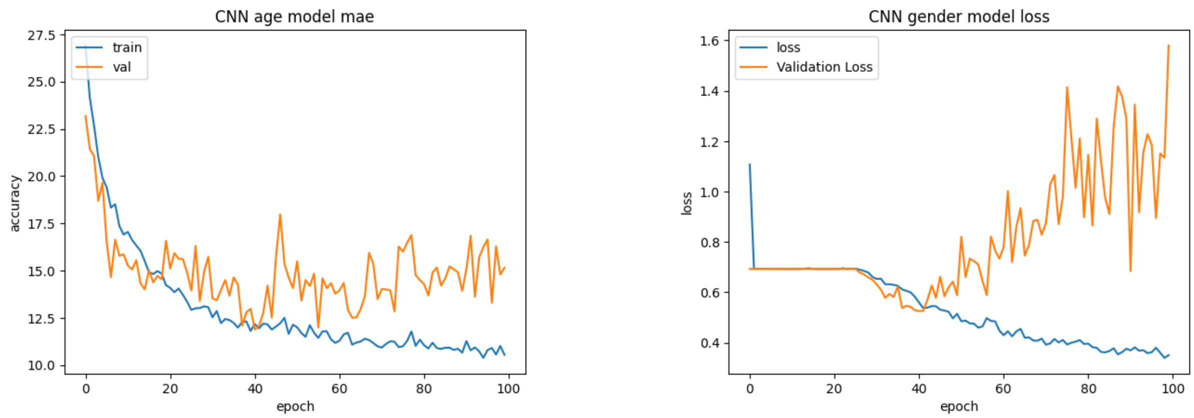


Figure 41 MAE and Loss graph of proposed CNN age model

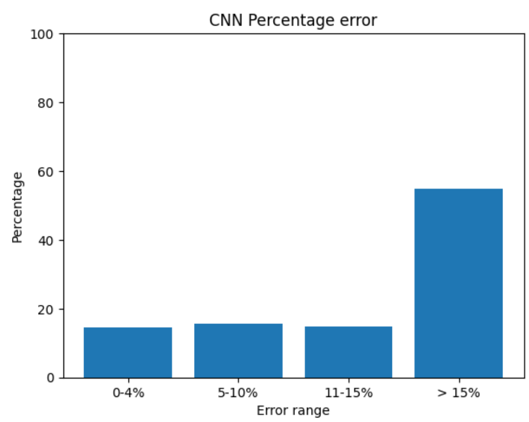


Figure 42 Percentages Error of proposed Neural network age model

Table 19 Percentage error of proposed Neural Network age model

Range of error	Number of files	Percentage
0-4%	115	14.56%
5-10%	124	15.70%
11-15%	118	14.94%
More than 15%	433	54.81%

Table 20 Mae loss and Rmse of proposed CNN Gender Model

Model Type	Train Accuracy	Test Accuracy	Train Loss	Test Loss
Gender	0.8513	0.8468	0.3507	1.5800

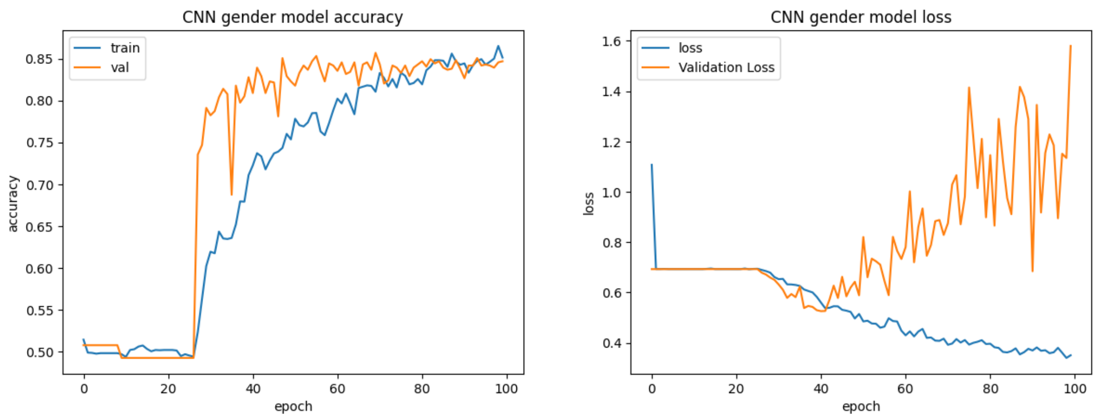


Figure 43 Accuracy and loss graph of the proposed CNN gender model

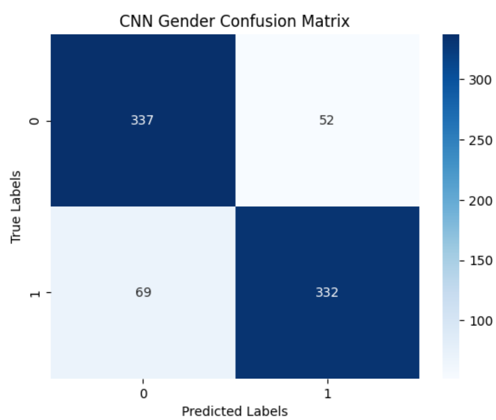


Figure 44 Proposed CNN Confusion Matrix

5.2.3. Proposed Pretrained model result

Table 21 Mae and loss of proposed Pretrained+Fine-tuning age model

Model Type	Train MAE	Test MAE	Train Loss	Test Loss
Age	7.4472	9.9541	95.2591	164.0147

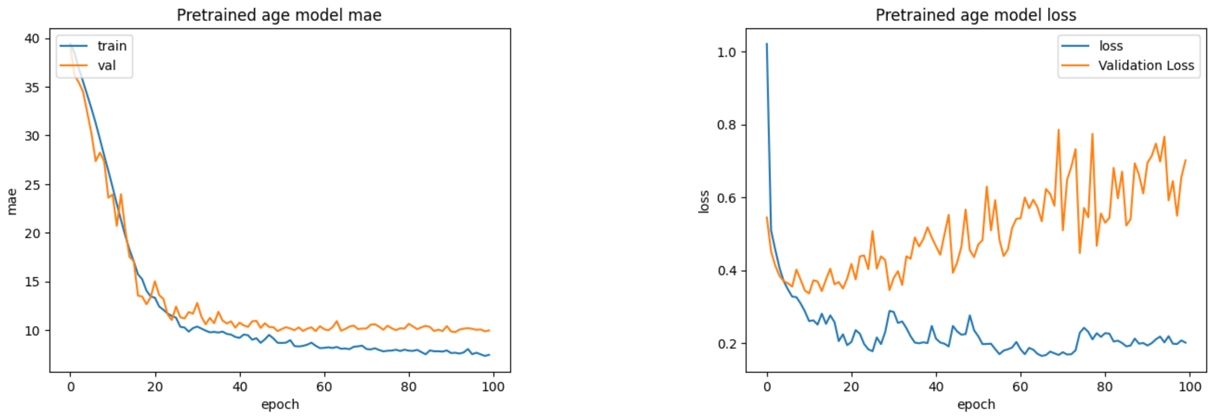


Figure 45 MAE and Loss graph of proposed CNN age model

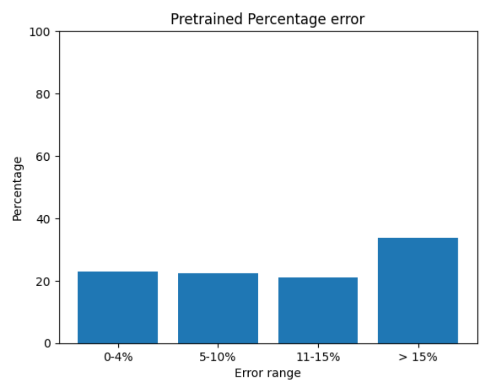


Figure 46 Percentages Error of proposed Pretrained+Fine-tuning age Model

Table 22 Percentage error of proposed Pretrained+Fine-tuning age model

Range of error	Number of files	Percentage
0-4%	181	22.91%
5-10%	177	22.41%
11-15%	166	21.01%
More than 15%	266	33.67%

Table 23 Accuracy and Loss of of proposed pretrained +fine-tuning gender model.

Model Type	Train Accuracy	Test Accuracy	Train Loss	Test Loss
Gender	0.8936	0.8433	0.2258	0.3708

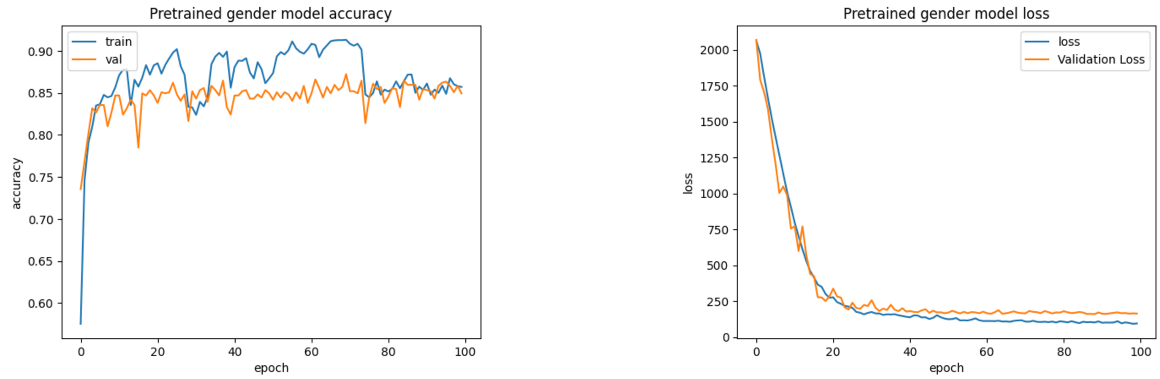


Figure 47 Accuracy and loss graph of the proposed CNN gender model

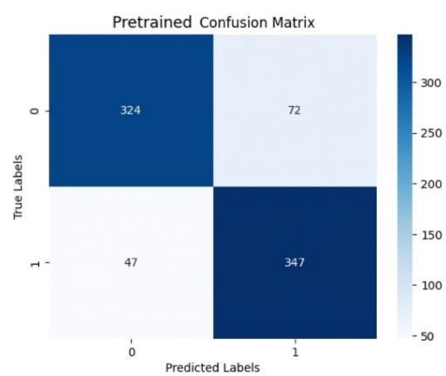


Figure 48 Proposed Pretrained Confusion matrix

5.3. Comparison of Proposed model MAE and Accuracy

Table 24 Comparison of proposed model

Model	Train MAE	Test MAE	Train Accuracy	Test Accuracy
Neural network	18.4853	17.1355	0.8256	0.7709
CNN	10.5386	15.1492	0.8513	0.8468
Pretrained	7.4472	9.9541	0.8570	0.8494

Judging from the result of the proposed model, the Model that uses VGG-16 pretrained model as a base model then performs a fine-tuning provide the best result which is to be expected. VGG-16 performance is already decent since it was trained by an enormous amount of dataset, which brings its efficiency to the highest out of 3 models. The second is CNN which has its accuracy slightly lower than pretrained model by a bit while the MAE is clearly higher. Even though CNN has been introduced since 1980 but its still widely used in various tasks because of its ability to extract the features, finding pattern and can learn the features of the data by itself. Neural networks which have its performance lowest of the three are not mean the model is bad, but due to limited GPU resources in Google Colab, the number of batch size and epoch are being limited to the point that the model can’t even be trained properly when its age and gender part is being combined and give a poor result.

6 Conclusion and suggestion

The prediction of the model is based on learning algorithm of the model to the target features, which in this case is the face of human male and female. Each model has different advantages, disadvantages, and usage requirements which the user must inspect and fine-tune by themselves to make the model fit with the task at hand. The result of Neural network model is quite a shock while the rest can work when being combined.

For the main task of this project is to predict age and gender via human faces which is composed of different features. Those features will be stored and adjusted to the shape that the model is being able to learn. All the model must be fine-tuned to increase their performance which is a challenging task because we got only a glimpse of how it works, and the rest is up to trial and error method. Adjusting a parameter or function are also not guaranteed for the output to be better, also the resource is another factor that affect the training, the more detailed the model train, the more chance it will cause an over fitting problem or even the software that being use d to train reach their limit. We must carefully observe the change of the result and comeback for more fine-tuning until the outcome is great enough. The result shows that between 3 combined models, the performance of CNN are not better than Pretrained model since VGG-16 has been trained with an enormous amount of image net. Both Neural Networks and CNN performance are also decrease when both Age and Gender part are combined together, the exception belong to pretrained model that its result stay at its standard which is reasonable since the pretrained model should have its stability for various type of training more than the model that was design from scratch.

From the result, CNN and VGG-16 fine-tuned have a decent result since the former is already a popular and well-known method while the former has the pre-trained based model for helping performing prediction. Neural networks can be more suitable model if we can manage the resource usage or use either age or gender part separately. Nonetheless, the human face is a variable that is not constant and variant by many situations, so it’s impossible even for the machine to perfectly predict the correct input. The proposed can be used for predicting the identity of the target but in the case which we need a pin-point accuracy, we have to improve the model further.

7 References

[1] Raphael Angulu , Jules R. Tapamo and Aderemi O. Adewumi “Age estimation via face images: a survey” Internet: https://www.researchgate.net/publication/325613148_Age_estimation_via_face_images_a_survey, June 2018

[2] S.A. Khan, Munir, Ahmad, Muhammad Nazir, N.Riaz “A comparative analysis of gender classification techniques” Internet: https://www.researchgate.net/publication/286707847_A_comparative_analysis_of_gender_classification_techniques, January 2014

[3] Kamekshi Gaur, Dr. Pawan Kumar “Literature Review on Gender Prediction Model using CNN Algorithm” Internet: https://www.irjet.net/archives/V9/i3/IRJET-V9I356.pdf , Mar 2022

[4] Syed Taskeen Rahman, Asiful Arefeen, Shashoto Sharif Mridul, Asir Intisar Khan, Samia Subrina “Human Age and Gender Estimation using Facial Image Processing” Internet: https://ieeexplore.ieee.org/document/9230933, January 2020

[5] Koichi Ito, Hiroya Kawai, Takehisa Okano, Takafumi Aoki “Age and Gender Prediction from Face Images Using Convolutional Neural Network” Internet: https://ieeexplore.ieee.org/abstract/document/8659655, 07 March 2019, Publicher: IEEE, DOI: 10.23919/APSIPA.2018.8659655

[6] Chenjing Yan, Congyan Lang, Tao Wang, Xuetao Du & Chen Zhang " Age Estimation Based on Convolutional Neural Network” Internet: https://link.springer.com/chapter/10.1007/978-3-319-13168-9_22, PCM 2014, pp 211–220

[7] Jun-Cheng Chen, Amit Kumar, Rajeev Ranjan, Vishal M. Patel “A cascaded convolutional neural network for age estimation of unconstrained faces” Internet: https://ieeexplore.ieee.org/abstract/document/7791154, December 2016, DOI: 10.1109/BTAS.2016.7791154, Publisher: IEEE

Student information

1. Name – Surname Jakkaphob Kongthanarith ID 6210554725

Department of Electrical engineering Faculty of Engineering, Kasetsart University

Home Address 129/55 Perfectplace Traima, Bangrak noi, Meung Nonthaburi, Nonthaburi, 10600

Home Phone - Mobile Phone 0836265399

E-mail Jakkaphob.k@ku.th

Educational Background:

Educational Level	School/Institute	Graduation year
High School	Assumption College	2018
Junior School	Assumption College	2012