AI Course

Capstone Project   
Final Report

For students (instructor review required)

ⓒ2023 SAMSUNG. All rights reserved.

Samsung Electronics Corporate Citizenship Office holds the copyright of this document.

This document is a literary property protected by copyright law so reprint and reproduction without permission are prohibited.

To use this document other than the curriculum of Samsung Innovation Campus, you must receive written consent from copyright holder.

|  |
| --- |
| Facial Age Estimation |

August 13th,2024

Insomnia team

Hà Tấn Hiệp

**Nguyễn Kiên An**

Nguyễn Vân An

Nguyễn Thái Bình

Trịnh Minh Hiếu

**Nguyễn Mạnh Khang**

Content

1. Introduction

1.1. Background Information

1.2. Motivation and Objective

1.3. Members and Role Assignments

1.4. Schedule and Milestones

2. Project Execution

2.1. Data Acquisition

2.2. Training Methodology

2.3. Workflow

2.4. System Diagram

3. Results

3.1. Data Preprocessing

3.2. Exploratory Data Analysis (EDA)

3.3. Modeling

3.4. User Interface

3.5. Testing and Improvements

4. Projected Impact

4.1. Accomplishments and Benefits

4.2. Future Improvements

5. Team Member Review and Comment

6. Instructor Review and Comment

1. Introduction

1.1. Background Information

Artificial Intelligence (AI) has been transforming the way we interact with and understand the world around us. One of the prominent and potentially widely applicable fields is facial recognition. Facial recognition is not merely about identifying a person’s identity; it also encompasses more complex capabilities such as age prediction, emotion recognition, and even psychological attribute analysis.

Age prediction based on facial features is a fascinating and challenging area. It requires meticulous analysis of facial characteristics such as wrinkles, skin texture, and overall shape. With advancements in deep learning technologies, AI models are becoming increasingly powerful and accurate in identifying and classifying these features. Age prediction not only enhances user experience in social media applications but can also support various fields like advertising, healthcare, and security.

The subsequent sections of this article will build on the foundation and significance introduced in this section, offering a comprehensive view of the potential and applications of AI in age prediction based on facial features.

1.2. Motivation and Objective

### In today’s moden society, appoaching Internet is more and more viable to everyone, especially children. However, they can easily access contents which are not appropriate for them from social media, games and many platforms. With out parents or adult, they can do something that no one expected. Many news write about kids who spent a lot of money into games without their parent’s permission. Lots of kids visit adult website because of curiousity. Ect.

In real life, it’s not rare to see childen buy beer, wine or cigarette from grocery or mall in Vietnam. Because sellers don’t check the ID whether they old enough to buy a product. Children can easily get in bar or night club and sometimes they can see adult content in advertisement

**That leads to a problem that we need something to percisely confirms person’s age with out checking ID.**

We want to give a solution for that problem by the power of deep learning. In this project, we heading to Facial Age Estimation (FAE) model using Pytorch framework and dataset from the Internet. FEA is a model which can estimate a person’s age using computer vision and convolution neural network (CNN).

We expect to have high accuracy and low loss model after training process complete. It can be use integrate into IoT devices like camera and sent warning to user. For example, if camera estimate customer is not old enough to buy cigarette, it will sent the warning in the monitor in front of seller.

We hope our final product can help prevent bad affect to children and raising people awareness of negative effect come from ignoring young age come from morden age and companies.

1.3. Members and Role Assignments

We have 6 people on team:

1. Trịnh Minh Hiếu: Team Leader
2. Nguyễn Mạnh Khang: Team Member
3. Hà Tấn Hiệp:Team Member
4. Nguyễn Thái Bình: Team Member
5. Nguyễn Vân Anh: Team Member
6. Nguyễn Kiên An: Team Member

Mission:

* Each of us has to research and design the model
* Trịnh Minh Hiếu: Training and Testing Model
* Hà Tấn Hiệp: Deploying model to website, collecting data to make dataset and writing report.
* Nguyễn Mạnh Khang: Collecting results and Tunning model
* Nguyễn Thái Bình: Collecting results (loss, accuracy) and comparing results of this model with other models to improve model.
* Nguyễn Vân Anh: Comparing results of this model with other models to improve model.
* Nguyễn Kiên An: Collecting and Preprocessing data

1.4. Schedule and Milestones

1. Data Acquisition: Start Date: 7/14/2024 – End Date: 7/23/2024 :
   1. Data Selection
   2. Data Visualization
   3. Data Preprocessing
2. Design Model: Start Date: 7/23/2024 – End Date: 7/30/2024
   1. Choose and Research the CNN architecture
   2. Implement in code
3. Training Model: Start Date: 7/23/2024 – End Date: 8/14/2024
   1. Implement in code
   2. Tuning Hyperparameter
   3. Training and Validation
4. Testing and improvement: Start Date: 8/1/2024 – End Date: 8/14/2024
   1. Adjusting the Hyperparameter
   2. Training again
   3. Comparing results
5. Creating User Interface: Start Date: 8/2/2024 – End Date: 8/14/2024
6. Writing Report: Start Date: 7/30/2024 – End Date: 8/14/2024

View details in excel file below:

[SIC\_AI\_Capstone Project\_Work Breakdown.xlsx](SIC_AI_Capstone%20Project_Work%20Breakdown.xlsx)

2. Project Execution

2.1. Data Acquisition

In this sub-section we see some important datasets in the field of Facial age estimation:

**MORPH**  
MORPH is a facial age estimation dataset, which contains 55,134 facial images of 13,617 subjects ranging from 16 to 77 years old.

**Adience**  
The Adience dataset, published in 2014, contains 26,580 photos across 2,284 subjects with a binary gender label and one label from eight different age groups, partitioned into five splits. The key principle of the data set is to capture the images as close to real world conditions as possible, including all variations in appearance, pose, lighting condition and image quality, to name a few.

**CACD (Cross-Age Celebrity Dataset)**  
The Cross-Age Celebrity Dataset (CACD) contains 163,446 images from 2,000 celebrities collected from the Internet. The images are collected from search engines using celebrity name and year (2004-2013) as keywords. Therefore, it is possible to estimate the ages of the celebrities on the images by simply subtract the birth year from the year of which the photo was taken

**FG-NET**  
FGNet is a dataset for age estimation and face recognition across ages. It is composed of a total of 1,002 images of 82 people with age range from 0 to 69 and an age gap up to 45 years.

**UTKFace**  
UTKFace dataset is a large-scale face dataset with long age span (range from 0 to 116 years old). The dataset consists of over 20,000 face images with annotations of age, gender, and ethnicity. The images cover large variation in pose, facial expression, illumination, occlusion, resolution, etc. This dataset could be used on a variety of tasks, e.g., face detection, age estimation, age progression/regression, landmark localization, etc. - consists of 20k+ face images in the wild (only single face in one image) - provides the correspondingly aligned and cropped faces - provides the corresponding landmarks (68 points) - images are labelled by age, gender, and ethnicity

For more information and download, please refer to the [UTKFace](https://susanqq.github.io/UTKFace/)  
**Note:** In this project, we used the cropped version of UTKFace from [Kaggle](https://www.kaggle.com/datasets/abhikjha/utk-face-cropped?select=utkcropped)

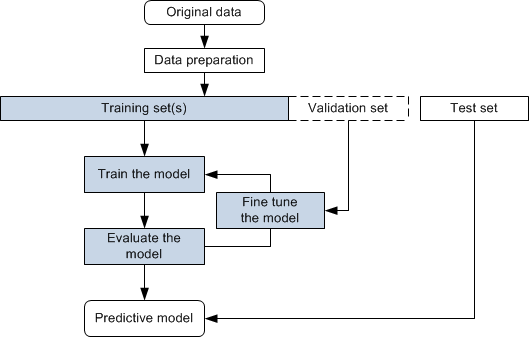
2.2. Training Methodology

The training process is the crucial part we studying. For model training, we applied transfer learning method by using pre-trained Resnet-50 weight and adjust the linear feature layer. We also added a dropout layer which is tunning random neuron weight equal to 0 so that the loss function can effeciently adjusts the weight and gives better performance while reducing overfitting.

During the training, we only save the best model which means the model with best metric will be saved every epoch. Later, this model can be loaded and use as a checkpoint and continue deeper learning.

2.3. Workflow

This is our basic workflow in this project. Although there are many minor taks we need doing like debugging, installing python’s libraried,ect. But we will break down into simplist and trying to explain as succinctly as possiable:



* After collected our data, we began to preprocessing the dataset. We firstly split training, validation and testing dataset in different ratio.
* Then we create data augmentation to increase dataset size and prevent overfitting.
* After handelling data, we design a CNN model inherited from ResNet-50 (2015).
* Next, we start training process with hyperparameters feed the training dataset and validation dataset as input into model.
* Finally, we use testing dataset to double-check the accuracy of output model and start over again at training process with different hyperparameters to find the best combination.

2.4. System Design

* Data preparation: split training, validation and testing dataset in different 0.68 : 0.12 : 0.2 ratio respectively using built-in function in **scikit-learn** library.
* Data preprocessing: Using torchvision- a library of Pytorch, many method are applied including adjust brightness, color filter; filpping and rotate image; resize image and normalizing for the normalization process.
* Model selection: for image classification, we chose CNN architecture, Resnet-50 specifically. The project will use python programming language and pytorch deep learning framework.
* Hyperparameter tunning:
  + Loss function: Mean Absolute Error (MAE), also known as L1 Loss, is a loss function used in regression tasks that calculates the average absolute differences between predicted values from a machine learning model and the actual target values. The mathematical equation for Mean Absolute Error (MAE) or L1 Loss is:  
    MAE =

Where:  
n is the number of samples in the dataset  
yᵢ is the predicted value for the i-th sample  
ȳ is the target value for the i-th sample

* + Metric: Mean absolute error (synchronize with loss function)
  + Optimizer: SGD function (Stochastic gradient descent function). It is an iterative method for optimizing an objective function with suitable smoothness properties (e.g. differentiable or subdifferentiable). It can be regarded as a stochastic approximation of gradient descent optimization, since it replaces the actual gradient (calculated from the entire data set) by an estimate thereof (calculated from a randomly selected subset of the data). Especially in high-dimensional optimization problems this reduces the very high computational burden, achieving faster iterations in exchange for a lower convergence rate.
  + Batch size: number of input feed into model every step.
  + Epochs: nummber of times, model learns from dataset.
* Testing: we use testing dataset created in data preparation step to test model and plot the diagram. We will base on that diagram to evaluate the trained model on the validation set to fine-tune model parameters and prevent overfitting.   
  Finally, tesing the real-world data for pratical estimation.

3. Results

3.1. Data Preprocessing

The labels of each face image is embedded in the file name which are made by UTKface, formated like:

* [age]\_[gender]\_[race]\_[date&time].jpg
* [age] is an integer from 0 to 116, indicating the age
* [gender] is either 0 (male) or 1 (female)
* [race] is an integer from 0 to 4, denoting White, Black, Asian, Indian, and Others (like Hispanic, Latino, Middle Eastern).
* [date&time] is in the format of yyyymmddHHMMSSFFF, showing the date and time an image was collected to UTKFace

After we have data match with label, data will be normalize. Technically, we normalize a tensor image with mean and standard deviation and then convert a PIL Image or numpy.ndarray to tensor. Later, that input images will be resize to the given size. Next is data augmentation to increase the size of dataset and generate new input to decrease overfitting chance in model. Some paramenter we use including:

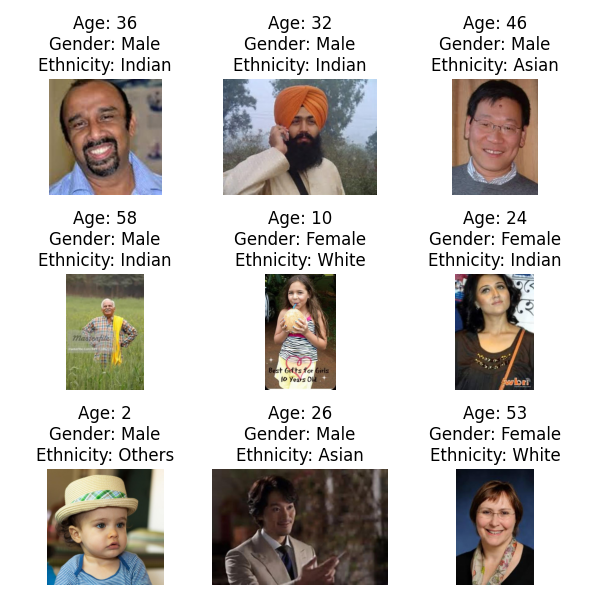
* Randomly change the brightness, contrast, saturation and hue of an image
* Rotate the image by angle
* Horizontally flip the given image randomly with a given probability.

3.2. Exploratory Data Analysis (EDA)

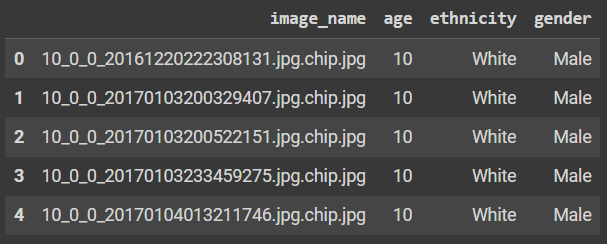
We used data on UTK dataset, which consists of images categorized by age, gender, and ethnicity.

A. Plot the Images in the UTK Dataset and create a csv file to save basic information

This step include loading and displaying sample images, obtaining image statistics to have a overview of dataset.

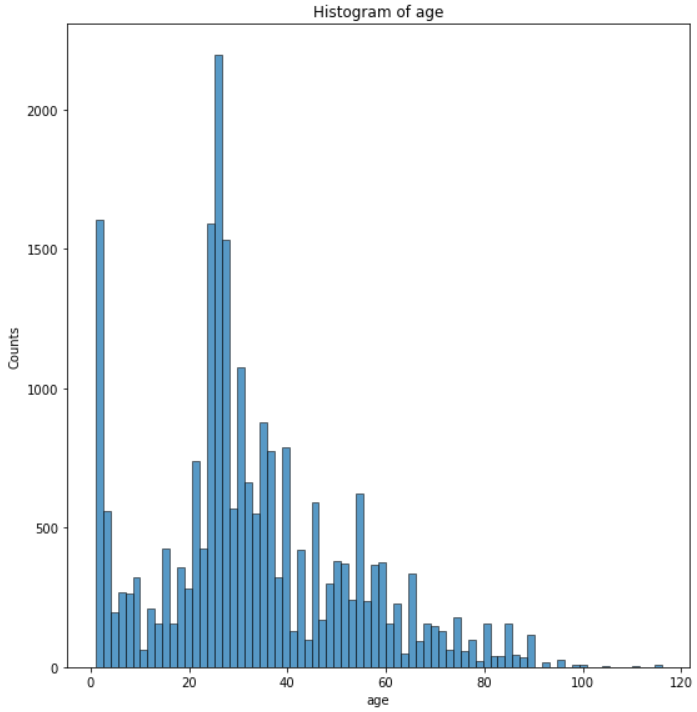
[](https://github.com/Ebimsv/Facial_Age_estimation_PyTorch/blob/main/pics/show_rand_samples.png)

The csv labels include overall information such as age, gender, and ethnicity for each image in the dataset.

­­[](https://github.com/Ebimsv/Facial_Age_estimation_PyTorch/blob/main/pics/csv_file.png)

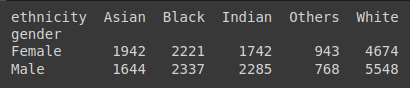
B. Univariate Analysis  
Univariate analysis is a type of exploratory data analysis (EDA) that focuses on examining one variable at a time. We used several histograms that can provide insights into the dataset's composition and help identify any imbalances or patterns. How ever, the Age histogram is the main plot of this process so we simply only plot it.

Histogram for Age:

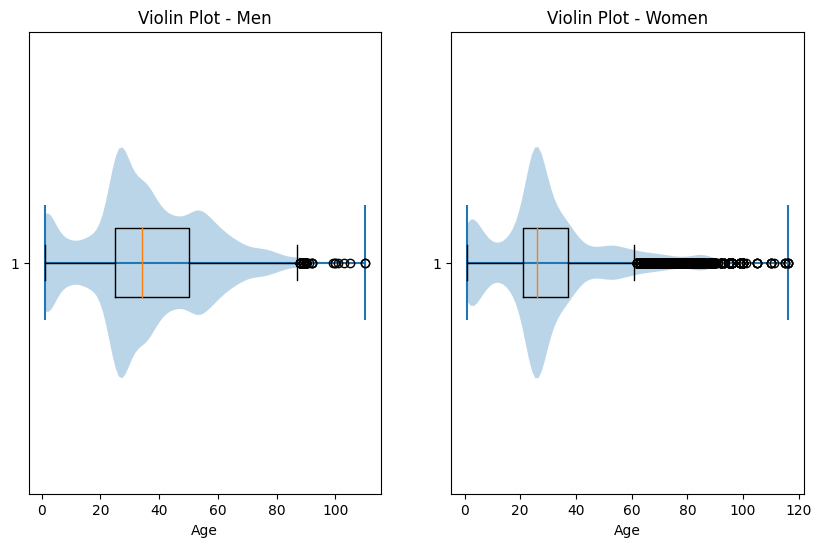
[](https://github.com/Ebimsv/Facial_Age_estimation_PyTorch/blob/main/pics/age_histogram.png)

C. Bivariate Analysis  
Bivariate analysis examines relationships between two variables

a. Cross-tabulation of gender and ethnicity  
Calculating the cross-tabulation of gender and ethnicity using the `pandas.crosstab()` function. This analysis can reveal the relationship between gender and ethnicity within the dataset and provide useful insights.

[](https://github.com/Ebimsv/Facial_Age_estimation_PyTorch/blob/main/pics/cross-tabulation.png)

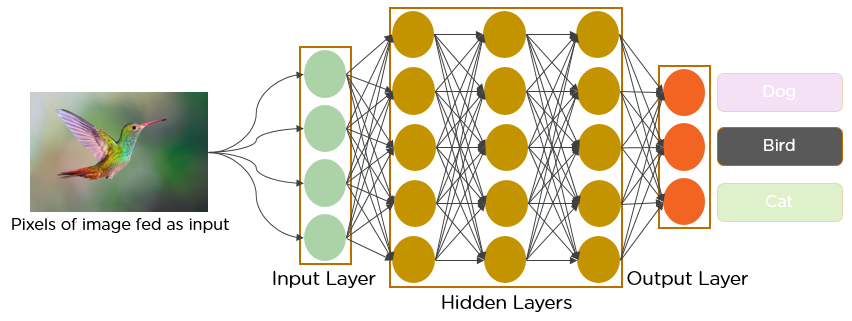
b. Violin plots and Box Plots for Age (Separated by Gender)  
These plots can help identify any differences or patterns in the age distribution between men and women in the UTK dataset.

[](https://github.com/Ebimsv/Facial_Age_estimation_PyTorch/blob/main/pics/violin_plot_age_men_women.png)

3.3. Modeling

In order to achive the best model we decided to use Resnet-50 which is a famous CNN architecture.

CNN model is a deep learning model using neural network and create fully connection network. The basis element of this architecture is the neuron node which consist of weight and bias. While weights determine the strength of connections between neurons, biases provide a critical additional layer of flexibility to neural networks. Biases are essentially constants associated with each neuron. Unlike weights, biases are not connected to specific inputs but are added to the neuron’s output. Entire network consist of input layer which is the input image, the output to give the result and the hidden layer where weight adjustment are happened behind. Using convolution layers, model can extract feature from image and learn pattern from them increasing the efficient of this architecture in computer vision.



A CNN model illustation

When deeper networks are able to start converging, a degradation problem has been exposed: with the network depth increasing, accuracy gets saturated (which might be unsurprising) and then degrades rapidly. Unexpectedly, such degradation is not caused by overfitting, and adding more layers to a suitably deep model leads to higher training error. Fig. 1 shows a typical example.

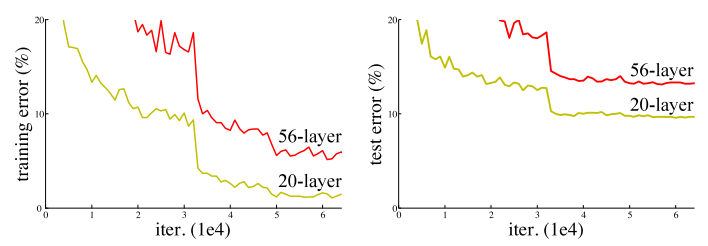


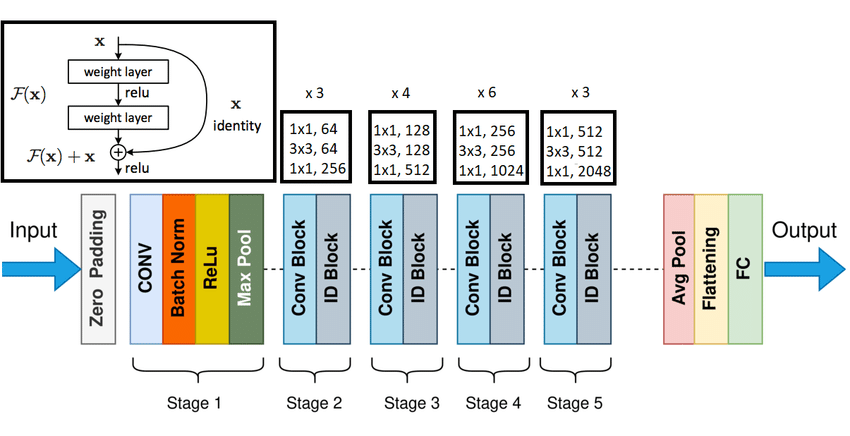
Fig. 1

In training process, backpropagation algorithm is commonly use. When the input feed into the network, this argorithm will calculate gradient of cost function for each parameter. This process will be repeated until all parameter of network is converging. In fact, gradients usually get smaller value when it descends deeper layers. This cause the update by gradient descent doesn’t change weights of the layer and make them can’t converge causing gradient vanishing.

Resnet architecture is excel in solving gradient vanishing problem. It stands for residual network, which refers to the residual blocks that make up the architecture of the network that allows for the training of very deep networks with hundreds of layers.

The ResNet architecture was developed in response to a surprising observation in deep learning research: adding more layers to a neural network was not always improving the results. To address this issue, the ResNet team, led by Kaiming He, developed a novel architecture that incorporated skip connections.

(Article: <https://arxiv.org/pdf/1512.03385> )



Each stage of the model consist of residual block which include a ‘linear’ path from input to output and another shortcut adding the input layer to right before the ReLU actiavtion function. The shorcut is call skipping connection, network should allow it to learn at least what the previous network learned, plus additional information. These connections allowed the preservation of information from earlier layers, which helped the network learn better representations of the input data. With the ResNet architecture, they were able to train networks with as many as 152 layers. However, we only use Resnet-50 architecture which has 50 layers for this project.

The tunning process involves several steps, including calculating the loss for an untrained model, overfitting the model on a small subset of the dataset, training the model for a limited number of epochs with various learning rates, creating a small grid using weight decay and the best learning rate, and finally training the model for longer epochs using the best model from the previous step.

**Step 1:** Calculate the loss for an untrained model using one batch  
This step helps us to understand that the forward pass of the model is working. The forward pass of a neural network model refers to the process of propagating input data through the model's layers to obtain predictions or output values.

**Step 2:** Train and overfit the model on a small subset of the dataset  
The goal of Step 2 is to train the model on a small subset of the dataset to assess its ability to learn and memorize the training data.

**Step 3:** Train the model for a limited number of epochs, experimenting with various learning rates  
This step helps us to identify the learning rate that leads to optimal training progress and convergence.

**Step 4:** Create a small grid using weight decay and the best learning rate and save it to a CSV file  
The goal of Step 4 is to create a small grid using weight decay and the best learning rate, and save it to a CSV file. This grid allows us to examine how weight decay regularization impacts the performance of the model.

**Step 5**: Train the model for longer epochs using the best model from step 4  
The goal of Step 5 is to train the model for longer epochs using the best model obtained from Step 4. This step aims to maximize the model's learning potential and achieve improved performance by allowing it to learn from the data for an extended period.

Please refer to train.py

**Step 6:** Save the best model from .pt to .jit  
The goal of this step is to convert the best model from .pt to .jit format. This conversion is primarily done to optimize and enhance the model's performance during deployment.

**Train and Evaluation Loop**

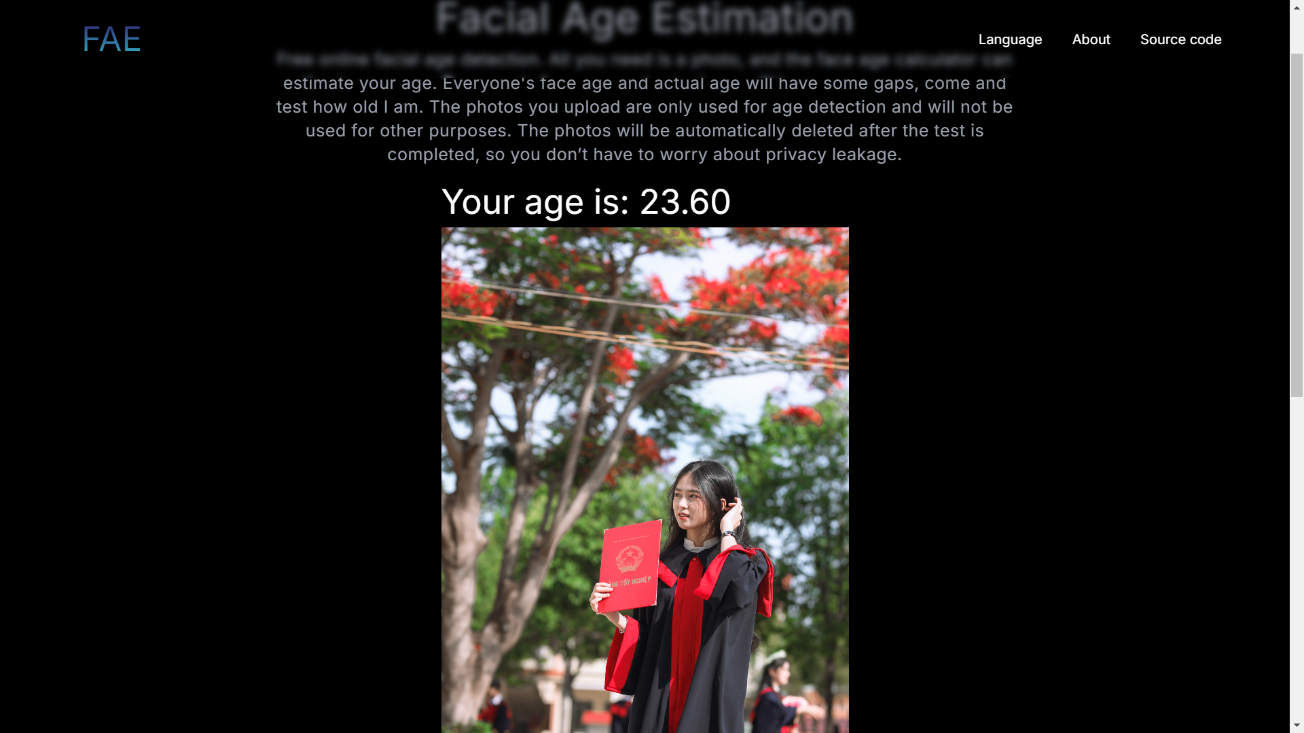
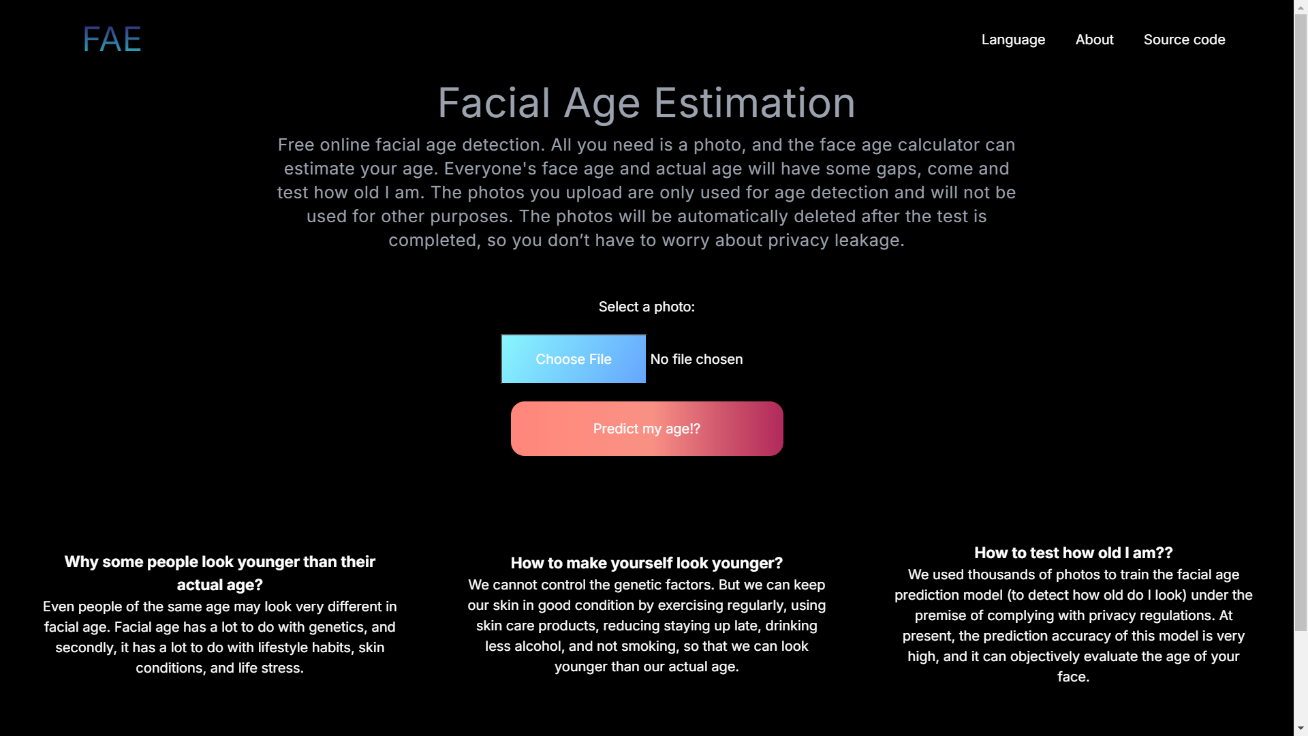
The train loop handles the training process, including forward and backward passes, updating model parameters, and monitoring training metrics. The evaluation loop performs model evaluation on a separate validation or test dataset and computes relevant evaluation metrics.

**Plotting Learning Curves with Matplotlib and TensorBoard**  
Learning curves visualize the model's training and validation performance over epochs, providing insights into the model's learning progress, convergence, and potential issues such as overfitting or underfitting.

TensorBoard is a tool for providing the measurements and visualizations needed during the machine learning workflow. It enables tracking experiment metrics like loss and accuracy, visualizing the model graph, projecting embeddings to a lower dimensional space, and much more.

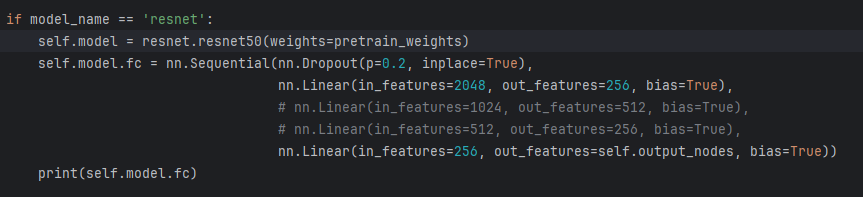
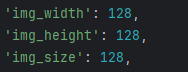
3.4. User Interface

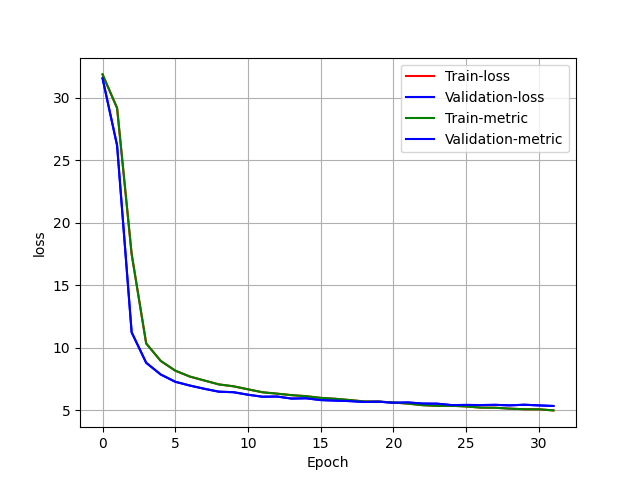
Model is deployed in a website with frontend using Nextjs-a React framework and Django-python framework for backend. The model will be load in the backend and await request from client. Server will handel the request and return a json as well as redirect to result page which will display image and result.



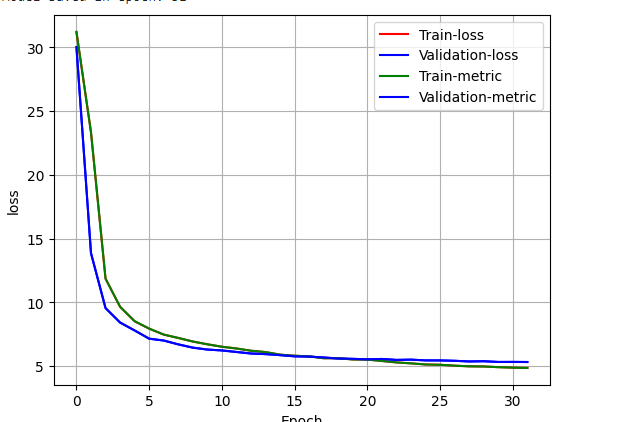
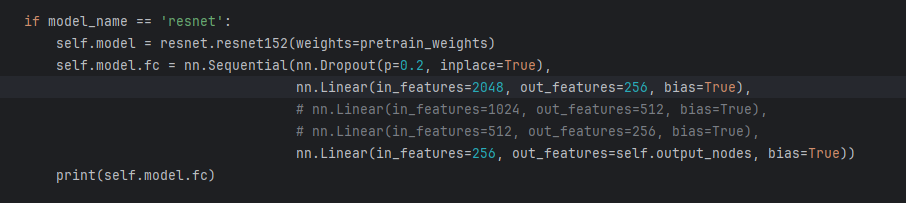
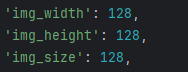
3.5. Testing and Improvements

We had changed some hyperparameter and test with Resnet50 and Resnet152-a more complex version of Resnet50. We have changed the size of image and layers of network (mostly in Resnet152) during the test tuning.

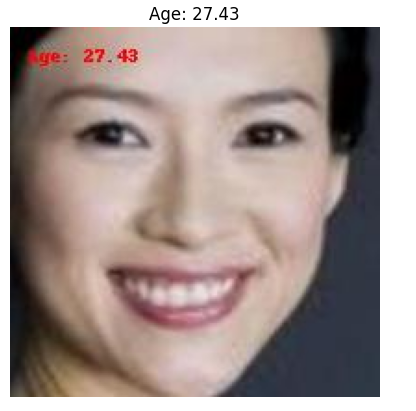
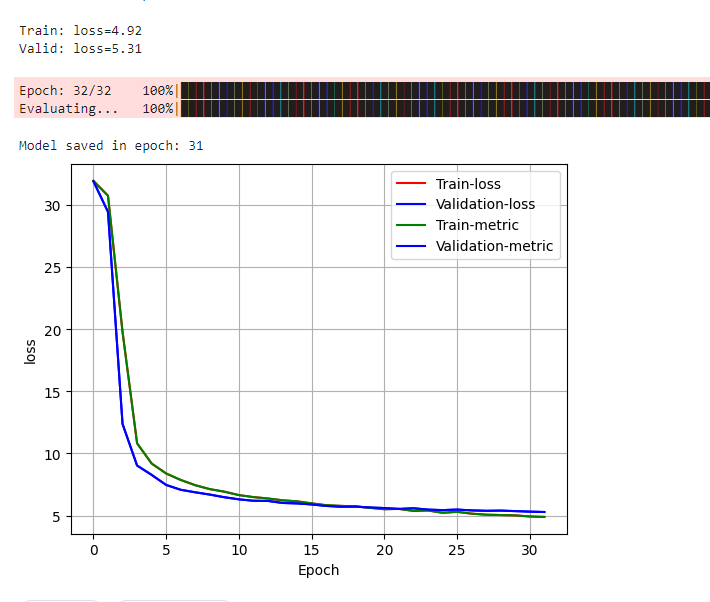
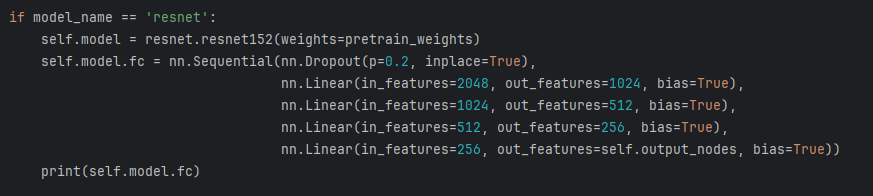
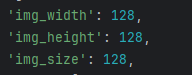




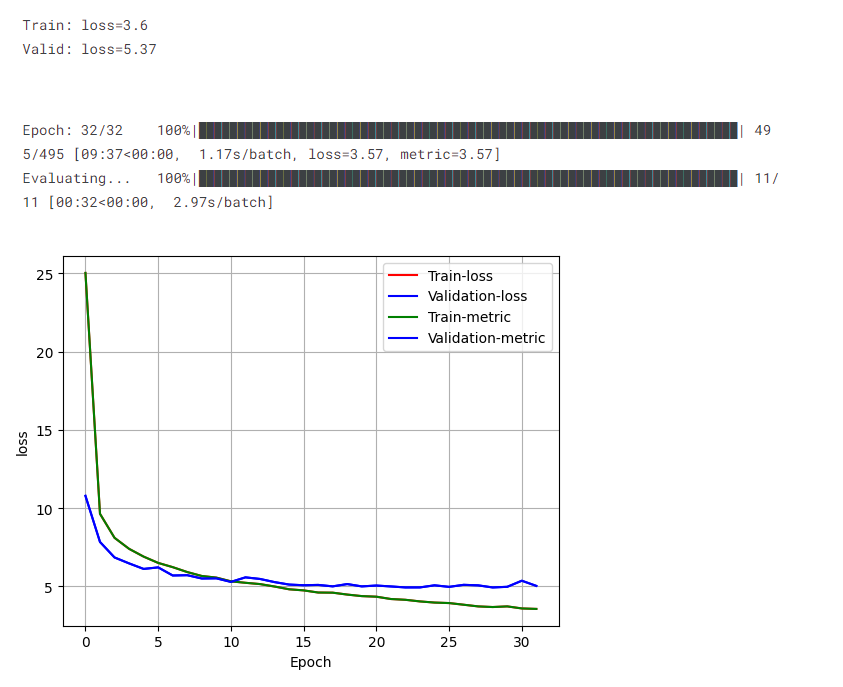
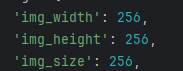
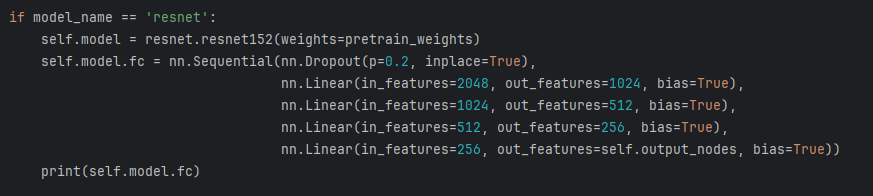
Resnet-50 with 2 linear layers, 128 x 128 image (primary model)



Resnet152 with 2 linear layers, 128 x 128 image



Resnet125 with 4 linear layers, 128 x 128 image



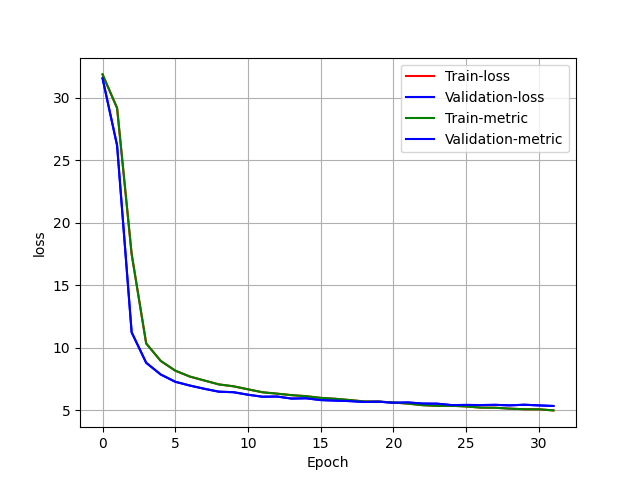
Resnet152 with 4 linear layer, 256 x 256 image

In all tests, Resnet50 outperform Resnet152 with better loss and accuracy. This is a example that not all deeper architecture network always work better than the shallower.

4. Projected Impact

4.1. Accomplishments and Benefits

Detail: <https://www.kaggle.com/code/himhtrinh/facial-age-estimation-ai-class-nhom3-studying>

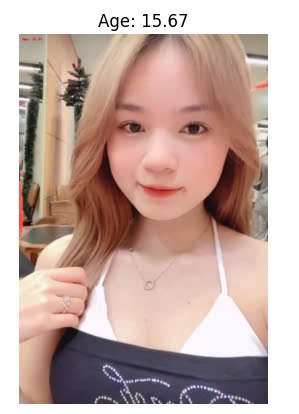


Final result is promissing with Resnet-50 gives a low loss (~5) at epoch 31. Model’s prediction is 27.76 meanwhile the real age of people in this image is 30 which has 2.24 loss. In the our point of view, the lady in the image is seem younger than her real age. We asked some people and they all answered that she is look younger than 30. The age they usually predicted was around 26-29. Which brings us into another challenge that in some cases, people can look younger or older than they are. The result is strongly depended on many factor like: people aging, skin’s health, light, brightness, emotion expression, ect.

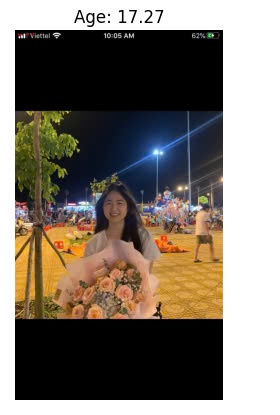
In order to giving the most objective view, we compared our model with a famous pre-trained model – Deepface and the result is suprisingly impressive. We had tested over 200 highschool student images whose was about 18-19 years old. During testing process, in some case, our model perform as well as Deepface and sometimes better.



Our model (same result) Deepface

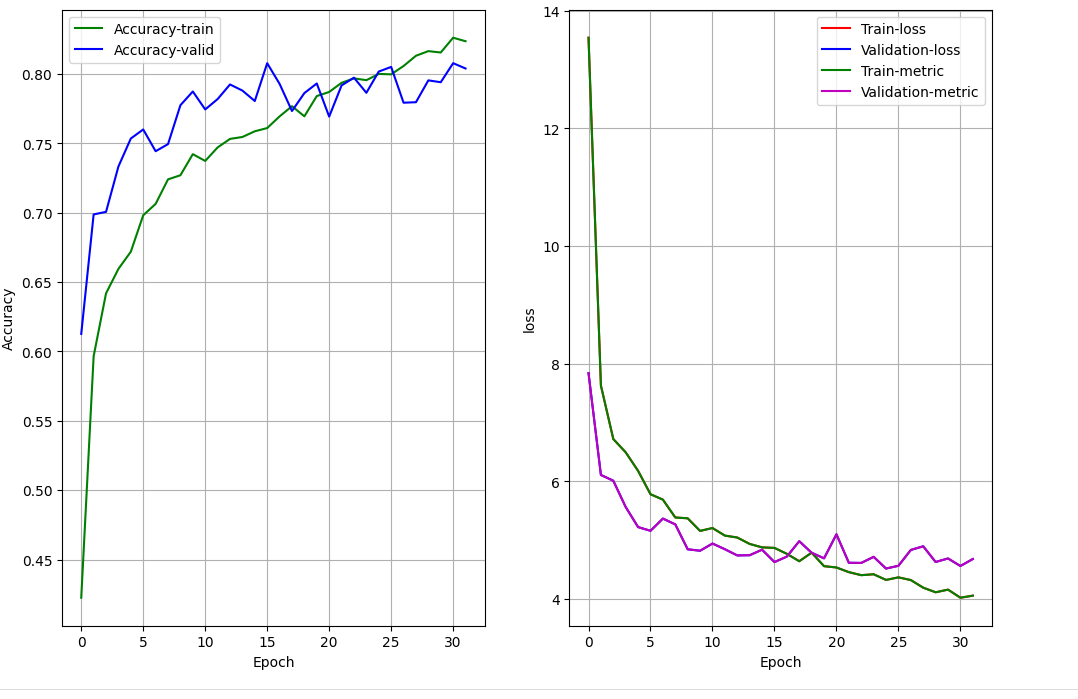


Our model Deepface

Our model (better result) Deepface

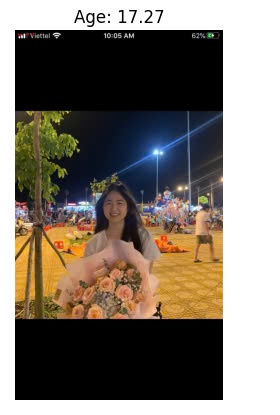
We also tested another morden architecture which is ViT (Vision Transformer). The training process was as same as we did with ResNet. The result after training was looking promising.



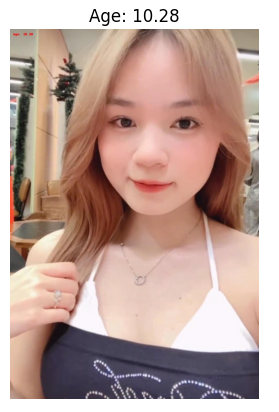
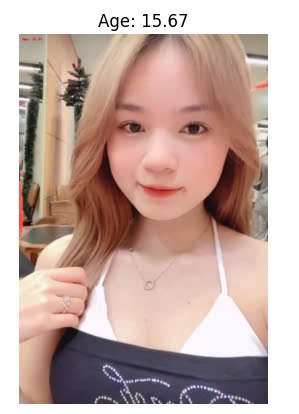
However, we want a real accuracy so we tested on real-life images. And once again, the result is suprisingly different from what we expected. ResNet-a older architecture actuallyl had the better performnce.



ResNet-50 (significantly better) ViT



ResNet-50 (better) ViT



ResNet-50 (significantly better) ViT

So far, the model performed relatively well in its designed task with decent prediction result compare to its scale. The loss is quite low and the outcome is not significantly different from the reality. This could be considered as a successful project.

Some benefits of facial age estimation in reality include:

* Protecting children from harmful or inappropriate content ensures their safety and mental well-being online.
* Streaming services like Netflix or Amazon Prime can use age estimation to recommend content suitable for the viewer’s age, ensuring that children do not accidentally access adult-rated shows or movies. Enhanced parental controls can use facial age estimation to verify the age of viewers and lock adult content for underage users.
* Websites with adult content can use facial age estimation to verify a user's age in real-time. If a user appears to be underage, the site can block access to adult content.
* Age estimation can be used to restrict in-game purchases or microtransactions that may not be appropriate for children.

4.2. Future Improvements

There are many aspects can be improved. We had stopped training at epoch 31 due to limitation of hardware. If we’d had better resources, the training process would have continue. The more epoch, the lower loss and higher accuracy. We can add a early stop function to stop the training process if the model’s metric doesn’t improve.

If the hardware problem was solved, we could using entire UTKFace dataset instead of using crop vesion. This will significantly increase the precisely of prediction in real-life situation.

Using other morden architecture is a interesting approach. We are thinking about using morden vision transformer (ViT) architecture after this project. Because of that, in our source code, you can find ViT and EfficientNet B0 in model.py which added into the code.

Adding another feature is also a viable option. Our model not only estimate the age but also ethic and gender. Later, this model can be used in ecconomy or marketing service to assist companies give the best service and product to appropriate custom.

5. Team Member Review and Comment

|  |
| --- |
|  |

|  |  |
| --- | --- |
| NAME | REVIEW and COMMENT |
|  | A two-man band in a nutshell |
|  |  |
|  |  |
|  |  |
|  |  |

6. Instructor Review and Comment

|  |  |  |
| --- | --- | --- |
| CATEGORY | SCORE | REVIEW and COMMENT |
| IDEA | \_\_/10 |  |
| APPLICATION | \_\_/30 |  |
| RESULT | \_\_/30 |  |
| PROJECT MANAGEMENT | \_\_/10 |  |
| PRESENTATION & REPORT | \_\_/20 |  |
| TOTAL | \_\_/100 |  |