FPT Education
FPT UNIVERSITY

AIP391 Project

Determine Gender Based On Photos

Team 1: Hoang Quoc Trung (leader), Nguyen Hung Thinh

Advisor: Nguyen Quoc Trung

Abstract: The analysis and extraction of possible information from human face images has been studied by scientific researchers since the early 90s of the last century. This is because there is a lot of information useful information can be extracted from face images, such as identity, gender, emotions, interaction only, ethnicity, health status, etc. Previous studies on gender detection have been completed based on various static body features, such as face, eyebrows, hand shape, body shape, fingernails, etc. In this study, we will use data available on Kaggle with more than 200k images to training model Gender Classification (male/female). To achieve this goal, we will use and test CNN models, specifically Transfer Learning models (VGG-16, Resnet-50, Efficientnet-B0, Inception-V3, Mobilenet, Xception) and choose VGG-16 model best applied to the project and achieved 98.11% accuracy on the test data set of nearly 17 thousand male and female images.

Keywords: Gender classification, VGG-16, CelebA dataset, gender detection

1. Introduction

In the modern era, Artificial Intelligence is part of our daily lives. Especially image processing and machine learning are playing a very important role in artificial intelligence. There is a lot of hard work done using image processing and machine learning techniques like gender classification, face recognition Etc. Where gender is an important attribute as it has quite a few real-world applications, such as in interactions with robots, in targeted advertising, in censuses. From there, we'll introduce simple AI technology using computer vision, machine learning, and deep learning tools to help determine a person's gender through their photos. This model can recognize their faces and predict whether their gender is male or female.

2. Related Works

In this report, we tested 6 CNN models and chose VGG-16 as the main model for the project. Following are the architectural ideas of 5 CNN models.

2.1 Inception-V3

With conventional CNNs, we don't know how much kernel_size is pre-determined. The objects, the information in the images are very different, sometimes large, sometimes small, so choosing the kernel size for Conv Layer is quite difficult to choose which parameter is appropriate, kernel_size = 1x1 or 3x3 or 5x5 will be better, too big will only capture general information, but too small will only get specific items. Therefore, CNN networks have been mainly solved by going deeper, with many different kernel layers to capture information, that leads to both high computational cost and easy to create Overfit for the network. Inception network was born to solve the problem of using kernel_size with what size is reasonable, they have the idea of going for width instead of depth. The most important element in the Inception network is the Inception module, a complete Inception network consisting of many small Inception modules put together. The idea of the Inception module is very simple, instead of using 1 Conv layer with a fixed kernel_size parameter, we can completely use multiple Conv layers with different kernel_size parameters (1x1, 3x3, 5x5, 7x7) at the same time. , etc) and then concatenate the outputs together, meaning that concurrently combining these kernal filters into the same block can result in what is called an Inception block or mixed block in keras. The Inception block will consist of 4 parallel branches. This method has the effect of helping to extract a variety of features on cognitive regions of different sizes, extracting more information on an image [1].

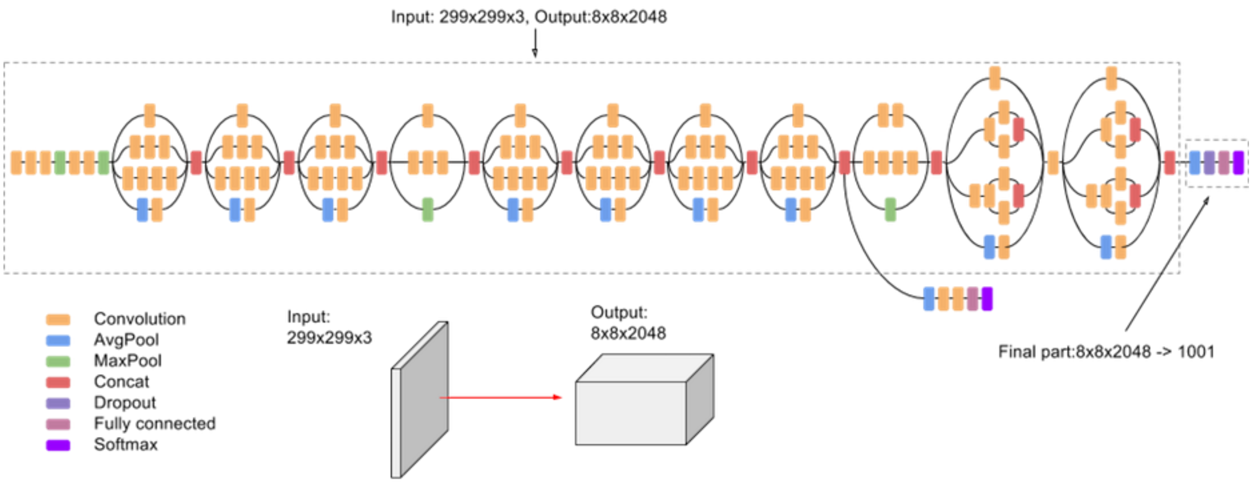
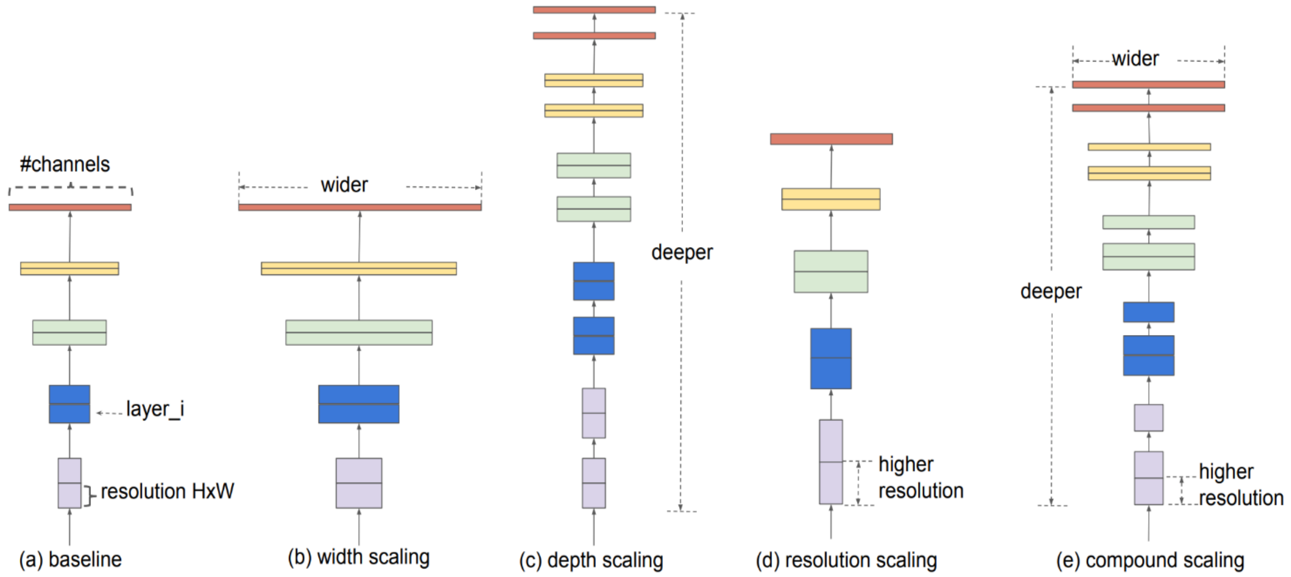


Figure 1: Inception-V3 Architecture [2]

2.2 EfficientNet-B0

This model offers a new approach to scale ConvNet model to get better results with less number of params, by Depth Scaling, Width Scaling, Resolution Scaling.



(a) baseline (b) width scaling (c) depth scaling (d) resolution scaling (e) compound scaling

Figure 3: A clear idea of what model zooming means across different sizes [3]

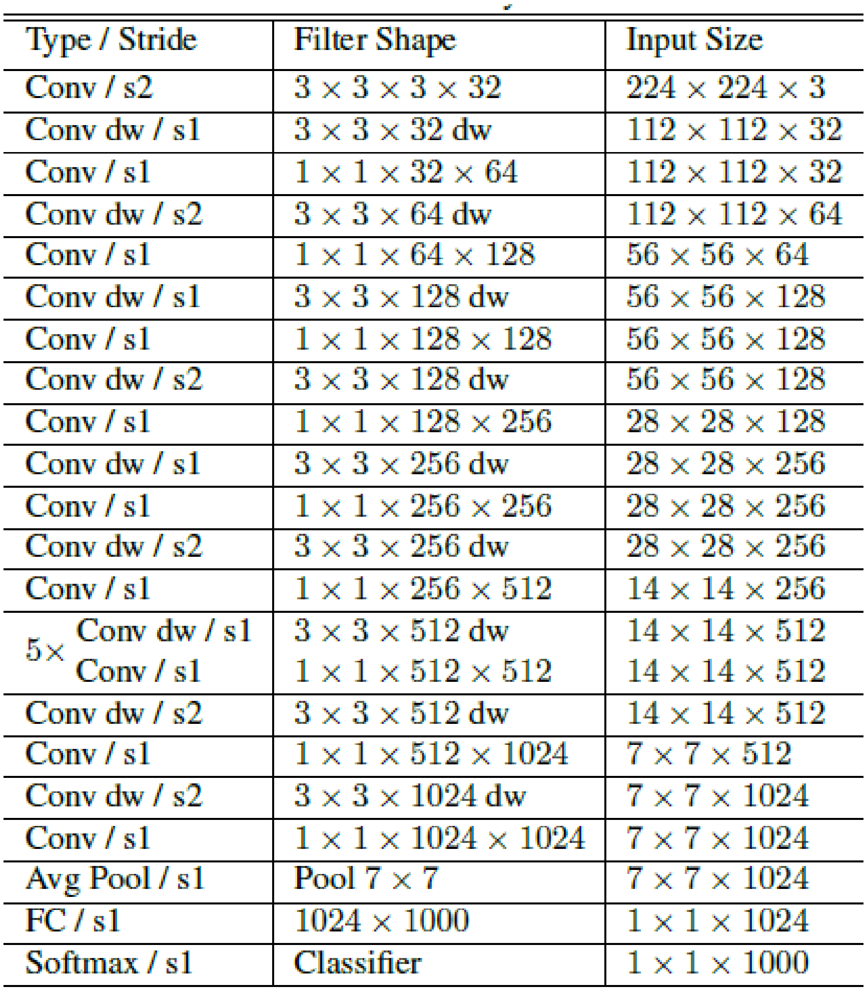
Formulate the problem: In order to narrow the search space, the authors have limited that all layers must be zoomed uniformly with a constant ratio. Their goal is to maximize model accuracy for any given amount of resources, which can be viewed as an optimization problem. Where w, d, rw,d,r are the coefficients to scale the width, depth and resolution of the network; $\widehat{H}_i, \widehat{W}_i, \widehat{C}_i$ are predefined parameters in the base network [3].

$$\max_{d,w,r} Accuracy(N(d, w, r))$$

$$N(d, w, r) = \bigodot_{j=1...s} \widehat{F}_i^{d.\widehat{L}_i} X_{\langle r.\widehat{H}_i, r.\widehat{W}_i, r.\widehat{C}_i \rangle}$$

2.3 MobileNet

This architecture can reduce several million parameters but still maintain good accuracy, thanks to the use of a mechanism called Depthwise Separable Convolutions [4]. Depthwise Separable Convolutions divides the basic CNN into two parts: Deepwise Convolution and Pointwise Convolution. From Figure 4 we see that the model has 30 layers with the following characteristics: Layer 1 is Convolution layer with stride equal to 2, layer 2 is Depthwise layer, layer 3 is Pointwise layer, layer 4 is Depthwise layer with stride equal to 2 (different from step 2, dw layer 2 has stride size of 1), layer 5 is Pointwise layer, and layer 30 is Softmax, used for classification.



Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
5× Conv dw / s1 Conv / s1	$3 \times 3 \times 512$ dw $1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$ $14 \times 14 \times 512$
Conv dw / s2	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024$ dw	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
FC / s1	1024×1000	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times 1000$

Figure 4: MobileNet network architecture [4]

2.4 Xception

Architecture Xception [5], which stands for “Extreme Inception”. The Xception architecture has 36 convolutional layers forming the feature extraction base of the network. The data first goes through the entry flow, then through the middle flow which is repeated eight times, and finally through the exit flow. Note that all Convolution and Separable Convolution layers are followed by batch normalization (not included in the diagram). All Separable Convolution layers use a depth multiplier of 1 (no depth expansion).

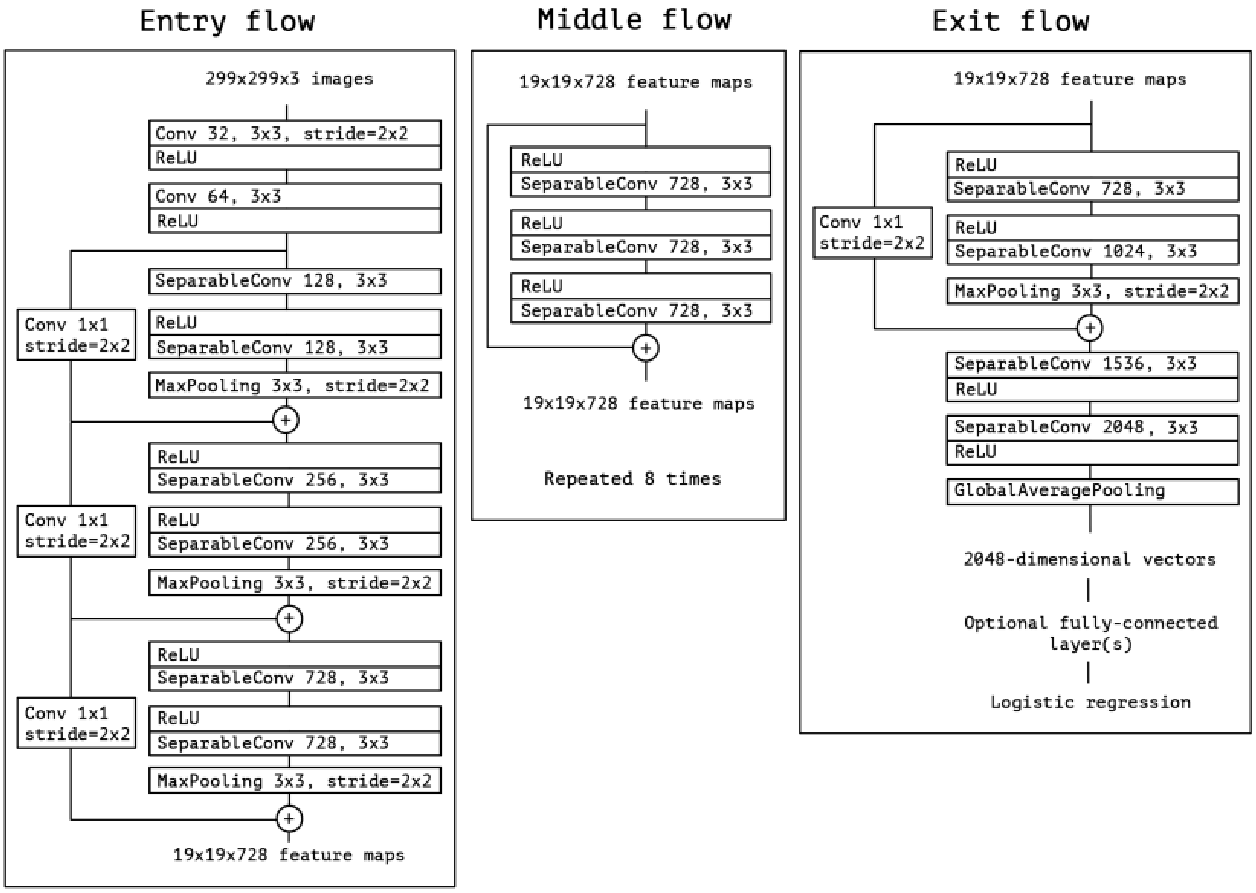


Figure 5: Xception architecture [5]

2.5 Resnet-50

ResNet-50 is a convolutional neural network that is 50 layers deep. ResNet, short for Residual Networks is a classic neural network used as a backbone for many computer vision tasks. The fundamental breakthrough with ResNet was it allowed us to train extremely deep neural networks with 150+ layers [6]. Convolutional Neural Networks have a major disadvantage 'Vanishing Gradient Problem'. During backpropagation, the value of gradient decreases significantly, thus hardly any change comes to weights. To overcome this, ResNet is used. It make use of "Skip Connection" in Figure 6.

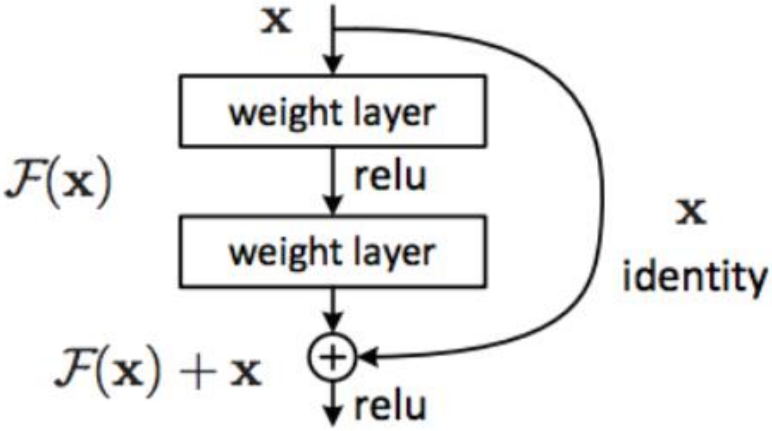

x
weight layer
$\mathcal{F}(\mathbf{x})$
relu
weight layer
$\mathbf{x}$
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

Figure 6: Skip Connection [6]

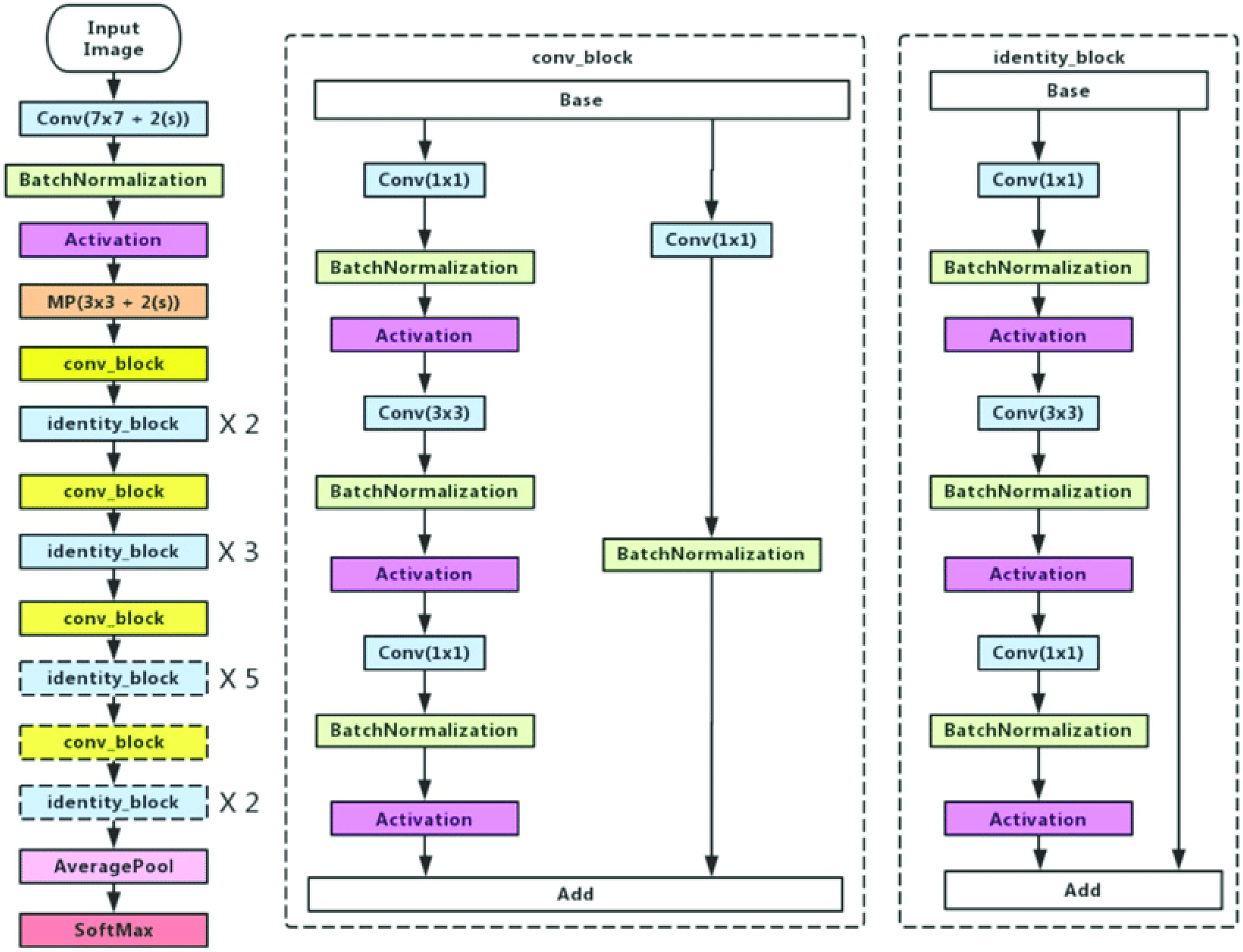


Figure 7: ResNet-50 architecture [7]

3. Data Preparation

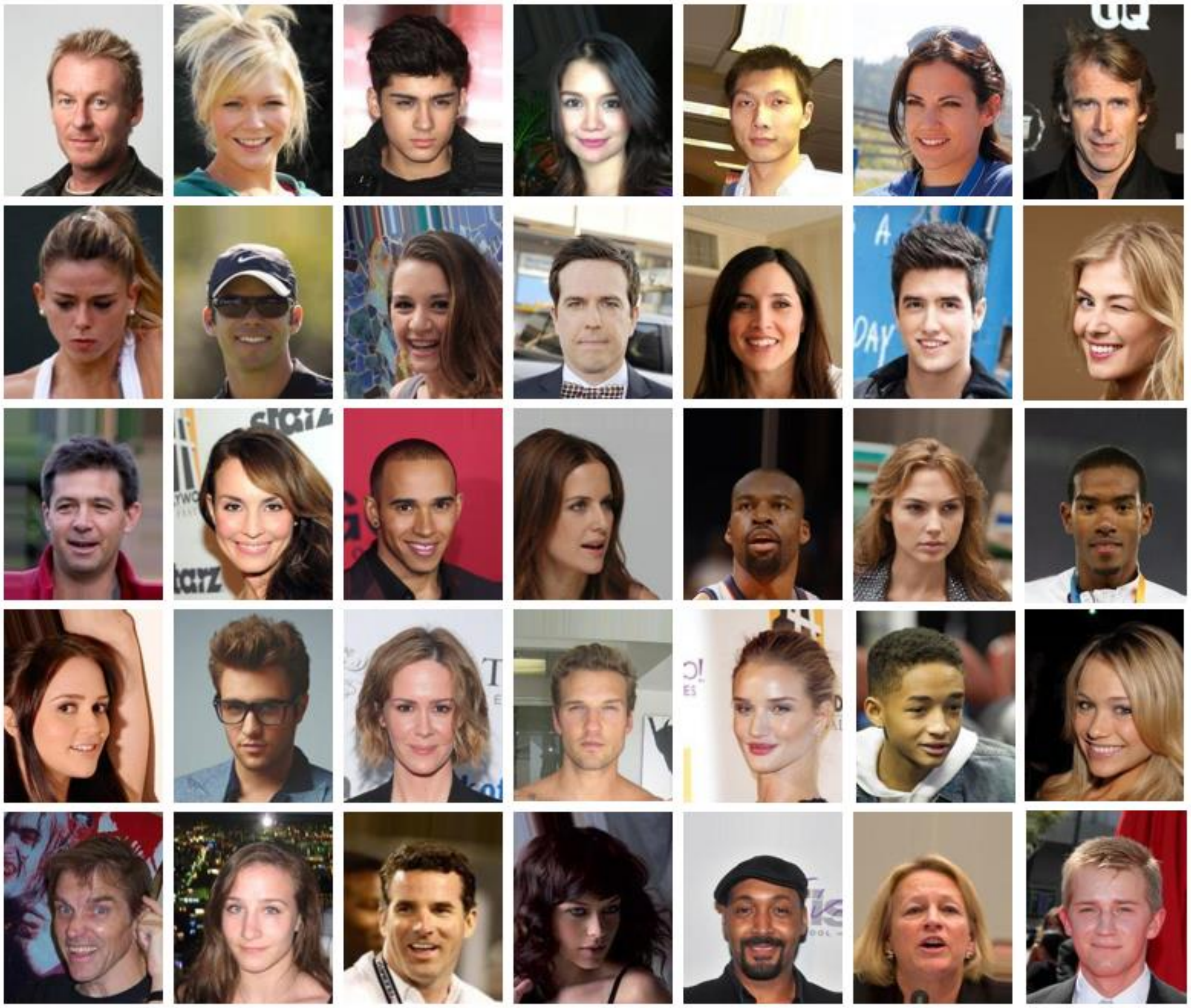
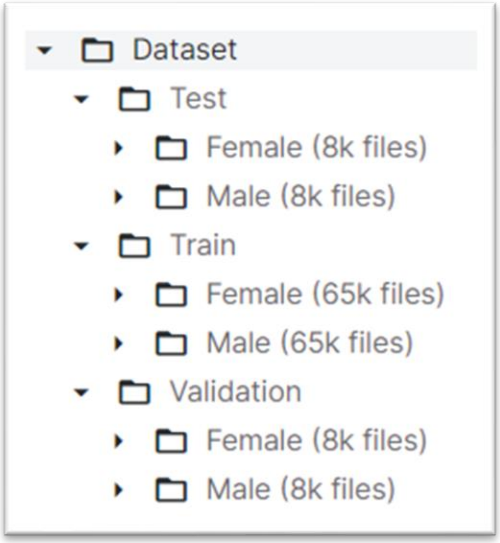


Figure 8: Some example images of the dataset

In our project, gender is predicted through photo or video so we will use image to train model. For data, we searched and got it from the source on Kaggle [8] . The dataset consisting of nearly 200K images is approximately 1.3 GB in size with attached labels. In it, all images have a common size of 218x178, illustrated in Figure 8. In this data set, there are 3 data sets Train/Test/Validation, in each of which includes sets containing face images of men and women, mainly the faces in this dataset are European. After having the data, we start to process the data balance to make the number of female and male images equal to avoid in the case of data imbalance. The overall data is illustrated in Figure 9.



- Dataset
  - Test
    - Female (8k files)
    - Male (8k files)
  - Train
    - Female (65k files)
    - Male (65k files)
  - Validation
    - Female (8k files)
    - Male (8k files)

Figure 9: Overview data

4. Methods

4.1 Convolutional Neural Network (CNN):

A neural network is a network of connected synthetic "neurons" that communicate with one another. An effectively trained network will function accurately when given an image or example to detect since the associations have numeric levels that are set during the preparation process. Complex structures are checked for organization issues by building different and unique layers in a CNN. Convolution layers, pooling/subsampling layers, non-straight layers, and totally related layers are the four types of layers that are the most frequent. Although neural systems and other case Identification techniques have been around for up to five years, convolutional neural systems have made important advancement. The features of using CNN for image assertion are discussed in this section. Tenacity in the form of changes and mutilation in the image, lower system needs, best and easier management. A CNN employs a system similar to a multilayer perceptron that has been optimized for low processing requirements

[9-11]. From the characteristics of CNN, we use CNN networks to train a model to predict gender through images [12].

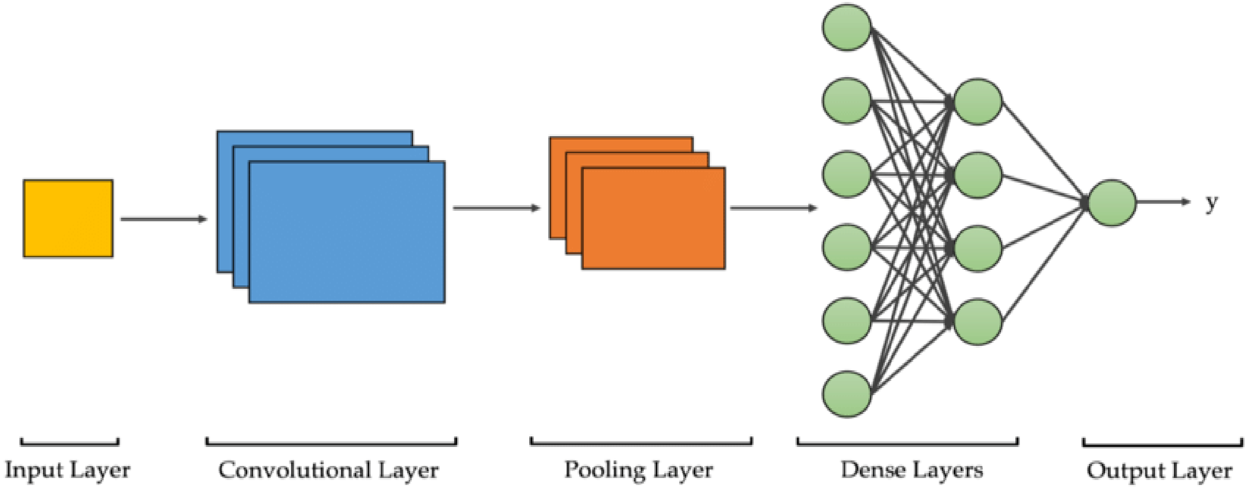
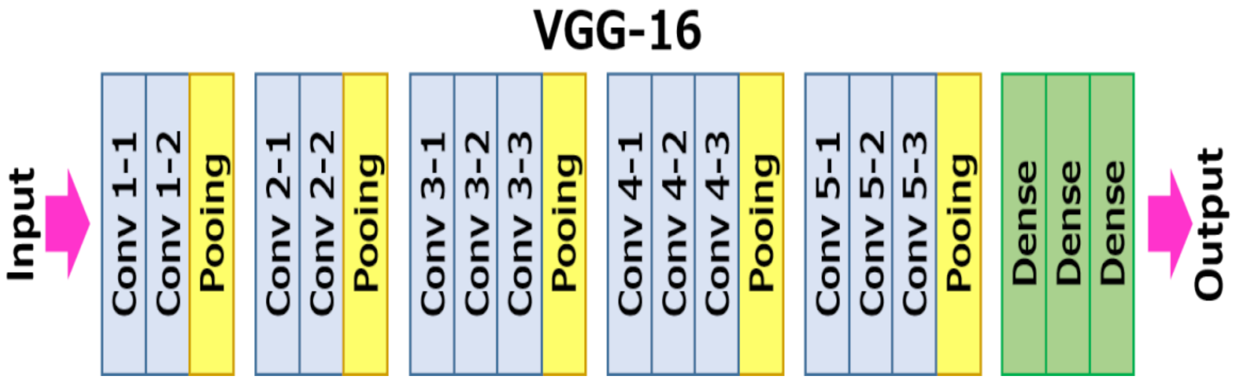


Figure 10: A basic convolutional neural network structure for image classification [13]

4.2 VGG–16

In all the methods as we mentioned above, in this project we use the model VGG-16 as the main core. In VGG-16 has an architecture consisting of 13 2-dimensional convolutional layers and 3 fully connected layers, in which all convolutional layers use kernal size of 3x3, VGG-16 also inherits ReLU activation function in AlexNet.



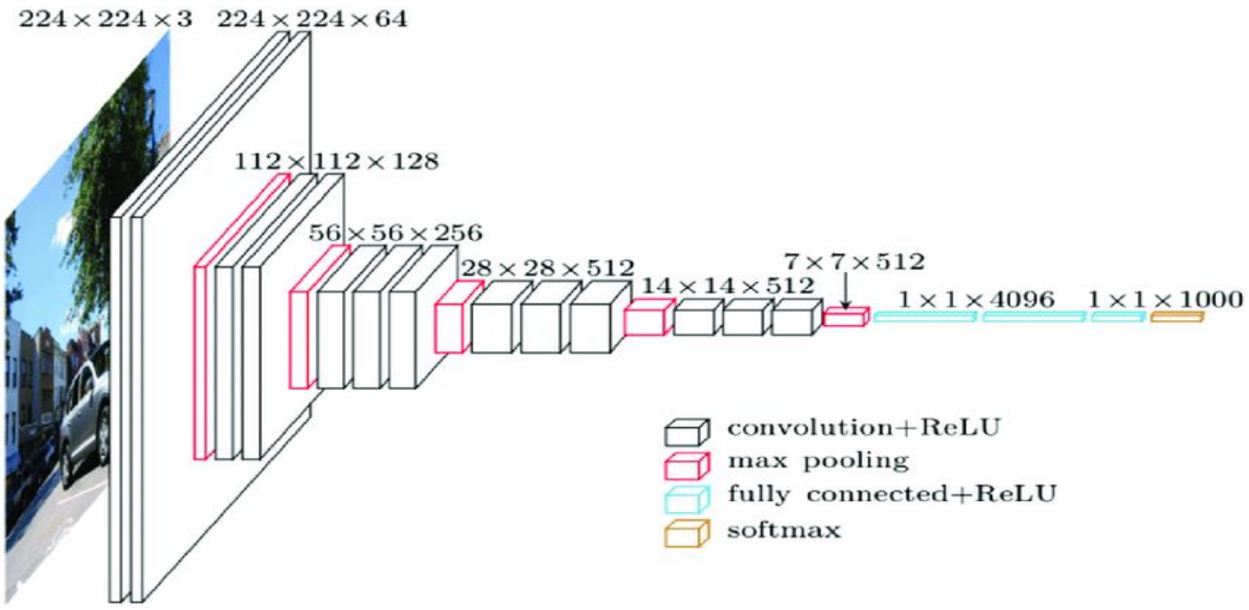
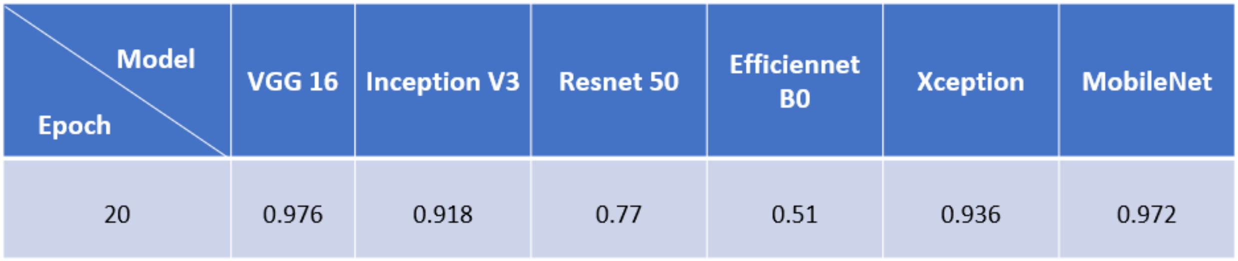


Figure 11: An overview of the VGG-16 model architecture [14]

Based on the pre-built VGG-16 architecture [15], we reuse the convolution layer architectures to extract image features and modify the fully connected layers to fit the data. Source code on my profile Kaggle [16].

5. Results and Discussion

After testing through 6 CNNs model on a test data set of about 16779 images in 20 epochs, the results are shown in Figure 12. From that result, that partly helps us use VGG as the main model.



Model Epoch	VGG 16	Inception V3	Resnet 50	Efficiennet B0	Xception	MobileNet
20	0.976	0.918	0.77	0.51	0.936	0.972

Figure 12: Accuracy on test set of deep learning models

Source code: Inception-V3 [17], Resnet-50 [18], Efficienet-B0 [19], Xception [20], MobileNet [21].

After using the VGG - 16 model for training, the following results obtained in 60 epochs are shown in Figure 13, It can be seen that the accuracy of the training set and the validation set is quite high and not too far apart, so the results are quite good, while the loss of train and validation is quite small and does not differ much from each other.

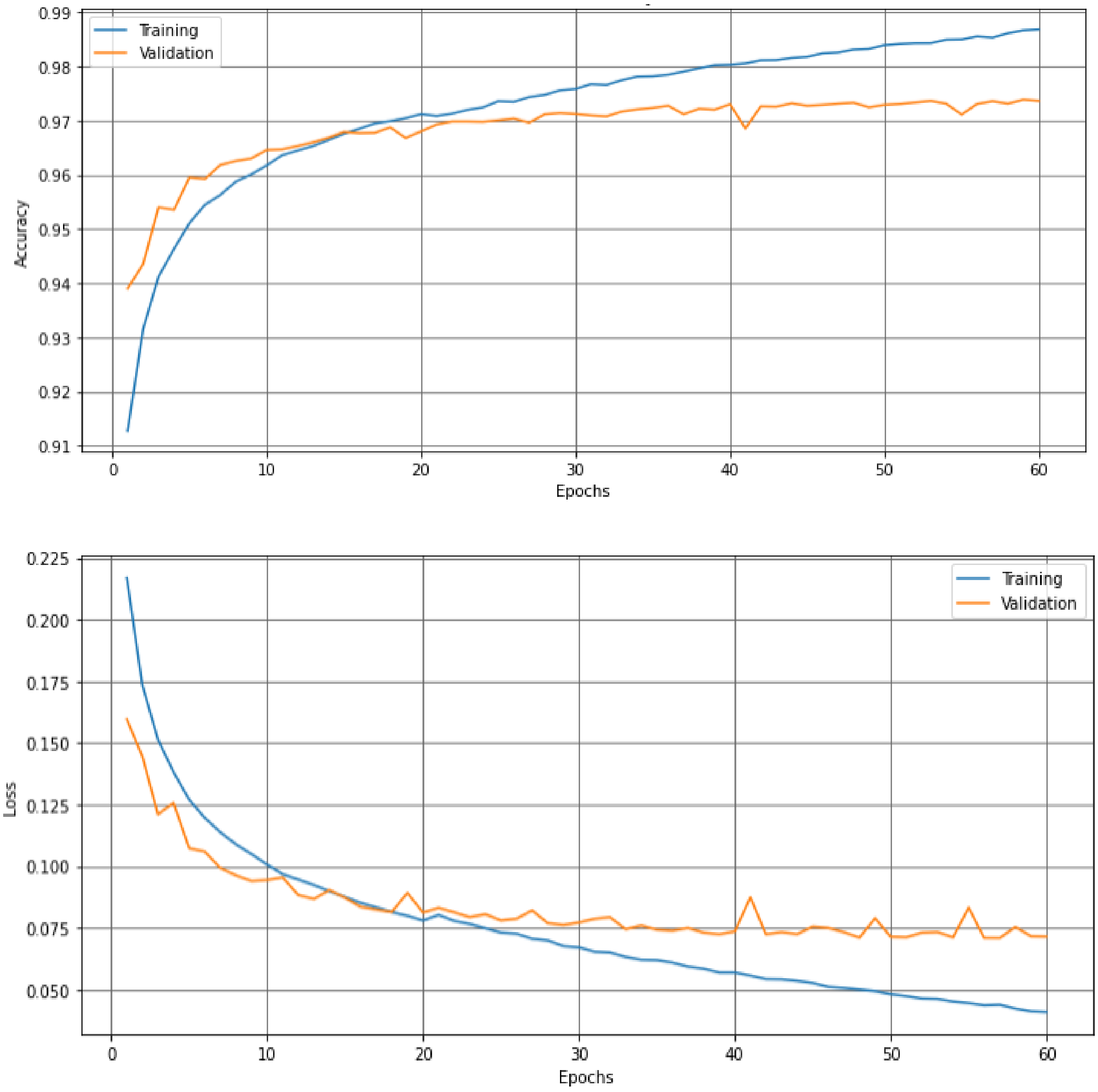


Figure 13: Accuracy and Loss of data train set and data validation set

Evaluation on the test set of about 16779 images belonging to 2 classes, the accuracy is 0.981167%. In which total wrong images 316 out of 16779 images are shown in Figure 14.

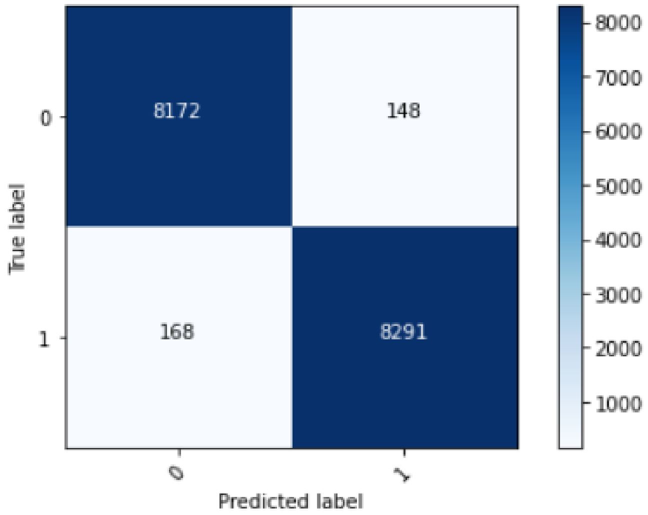


Figure 14: Confusion Matrix

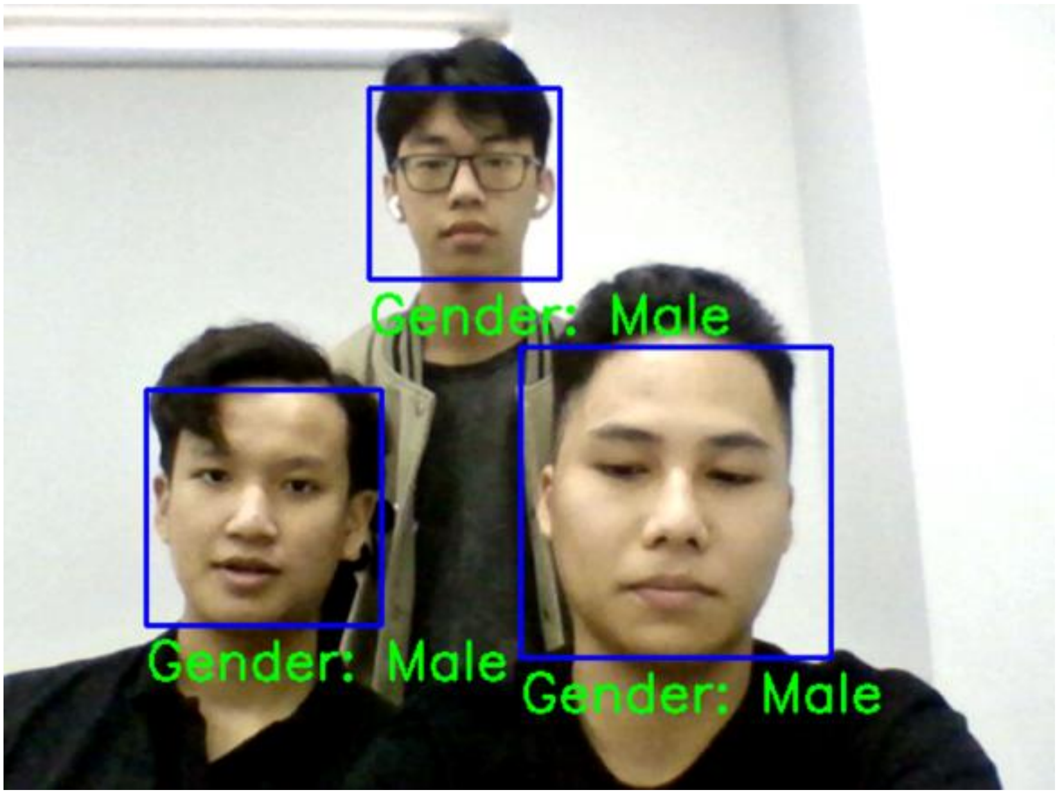


Figure 15: Demo on opencv camera, source code [22]

6. Conclusion

The model recognizes well when the face image is looking straight ahead, still not good for other face directions. Because the face data for the train is mostly straight-forward. Accuracy for European faces is better than for Asian faces.

The Next Development Directions: We will improve the face detection method with other models like (Yolo, MTCNN, etc) instead of using the built-in model supported by Open CV. Collect more Asian face data for even greater accuracy. Add some attributes to the model like age, facial expressions instead of gender recognition. Further improved data to accurately predict faces at different angles.

References

[1] Zbigniew, W., n.d. (2016). “Rethinking the Inception Architecture for Computer Vision”.

[2] PaperWithCode

https://production-media.paperswithcode.com/methods/inceptionv3onc--oview_vjAbOfw.png

[3] Quoc V., L., n.d. (2019). “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks”.

[4] Hartwig, A., n.d. (2017). “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications”.

[5] François, C., n.d. (2017). “ Xception: Deep Learning with Depthwise Separable Convolutions”.

[6] Jian, S., n.d. (2015). “Deep Residual Learning for Image Recognition”.

[7] ResearchGate

https://www.researchgate.net/figure/Left-ResNet50-architecture-Blocks-with-dotted-line-represents-modules-that-might-be_fig3_331364877

[8] Gender Classification 200K Images Data | CelebA.

https://www.kaggle.com/datasets/ashishjangra27/gender-recognition-200k-mages-celeba

[9] El-Habil, B. et al. (2022). "Cantaloupe Classification Using Deep Learning.” International Journal of Academic Engineering Research (IJAER).

[10] El-Khatib, M.J., et al. (2019). "Glass Classification UsingArtificialNeural Network." InternationalJournal of Academic Pedagogical Research.

[11] El-Mahelawi, J. K., et al. (2020). "Tumor Classification UsingArtificialNeuralNetworks." InternationalJournal of Academic Engineering Research.

[12] S. . Tilki, H. B. . Dogru, and A. A. . Hameed (2021). “Gender Classification using Deep Learning Techniques”.

[13] ResearchGate

https://www.researchgate.net/figure/A-basic-convolutional-neural-network-structure-for-image-classification-Convolutional_fig1_343462095

[14] Neurohive

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

[15] Andrew, Z., n.d. (2015). “Very Deep Convolutional Networks for Large-Scale Image Recognition”.

[16] Trung Hoang, Thinh Hung (2022). “Gender Classification: VGG16 and Fine Turning”

https://www.kaggle.com/code/hongtrung/gender-classification-vgg16-and-fine-turning

[17] Trung Hoang, Thinh Hung (2022). “Gender Classification: Inception V3”

https://www.kaggle.com/code/hongtrung/gender-classification-inception-v3

[18] Trung Hoang, Thinh Hung (2022). “Gender Classification: ResNet-50”

https://www.kaggle.com/code/hongtrung/gender-classification-resnet-50

[19] Trung Hoang, Thinh Hung (2022). “Gender Classification: EfficientNet B0”

https://www.kaggle.com/hongtrung/gender-classification-efficientnet-b0

[20] Trung Hoang, Thinh Hung (2022). “Gender Classification: Xception”

https://www.kaggle.com/code/hongtrung/gender-classification-xception

[21] Trung Hoang, Thinh Hung (2022). “Gender Classification: MobileNet”

https://www.kaggle.com/code/hongtrung/gender-classification-mobilenet

[22] Trung Hoang, Thinh Hung (2022). “Demo Gender Recognition Through Face On Camera”

https://www.kaggle.com/code/hongtrung/demo-gender-recognition-through-face-on-camera