Classifying Facial Attributes and Accessories with LeNet

Jirayu Petchhan*
Department of Electrical
Engineering
National Taiwan University of
Science and Technology
Taipei, Taiwan
d10907801@mail.ntust.edu.tw

***Abstract*— Classifying an image is a widely used implementation in the deep learning area. Many deep learning architectures are used in this implementation. In this paper, a classic model LeNet is used for classifying face attributes or accessories such as mustache, hat, bang, eyeglasses, hat, wavy hair, pointy nose, and oval face. For training the network, 100 images for each class is used. The network trained in single-label classification and multi-label-classification. The single-label classification network trained for 4, 5, 6 classes used as an output for classification. Experimental results show that the higher number of the classes achieves lower accuracy. And multi-label classification accuracy achieves a better result than single-label classification.**

***Keywords*— deep learning, LeNet, multi-label classification, face attributes**

I. Introduction

Classifying an image is a widely used implementation in the deep learning area. Many deep learning architectures are used in this implementation, such as LeNet, AlexNet [1], VGG [2], GoogleNet [3], and ResNet [4]. These deep learning architectures are trained by using one or more datasets. There are several datasets that available open-source on the internet, such as MNIST, COCO-stuff, ImageNet, etc.

One common example of this classification is by using a human face as an input image that can be retrieved from mentioned dataset above.

In this paper, LeNet deep learning architecture is used to classify human face attributes and accessories, such as mustache, hat, bang, eyeglasses, hat, wavy hair, pointy nose, and oval face.

The rest of this article is organized as follows. Section II gives a brief overview of face image recognition and classification and various methods used. Section III presents the approach for deep learning model description and training dataset. Section IV provides the experimentation methodology and result. Finally, in Section V, the conclusion of the article is summarized and concluded.

II. Related Works

Several studies have been conducted using human faces as input for deep learning models, such as face recognition [5] and facial expression recognition [6].

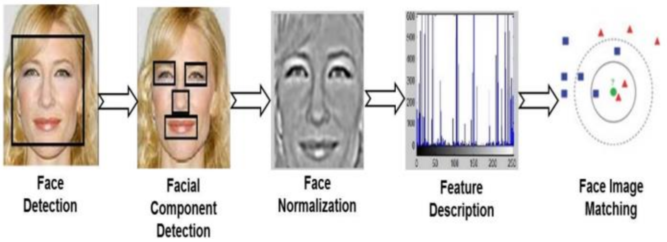

Fig. 1. Steps in face image retrieval system Training Graph for the third experiment

Fig. 1 shows the application of automated FIRSs [7] to find facial components such as nose, eyes, and mouth by first normalizing the appearance of the face. And then selecting an appropriate feature description, and developing matching methodologies. One of the matching approaches is by extracting the facial features of the given query image and comparing it with the facial features of face images stored in the face image database.

It is also a challenge to provide images as input to a deep learning model. This is because facial images can have certain characteristics such as face orientation, image quality, illumination, facial expressions. To solve this problem, a scalable face image retrieval with identity-based quantization and multi-reference re-ranking [8].

III. System procedure

After we surveyed and reviewed the contribution of several existing work. We can see that the dynamic LeNet model is being implemented which is a simple backbone network to adapt into any system that are diminutive and can be processed

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

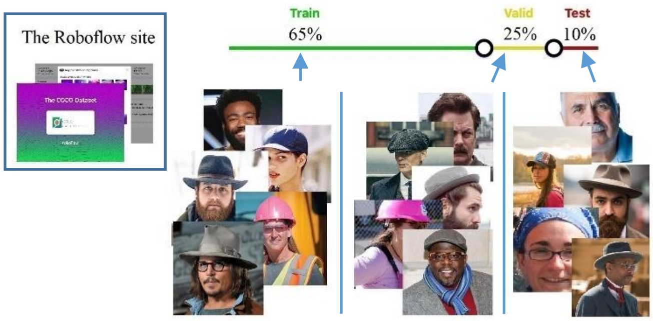

Fig. 2. The sample of web screen at the Roboflow site and percentage of datasets have been split into set of training, validating and testing.

expeditiously. Moreover, according to reviewing of LeNet network, it is able to provide high performance efficiency.

A. Dataset

Images that are deployed for the dataset become in many different sizes, but they need to be processed as pre-processing data. However, before we can proceed smoothly on the system net layer, we have designed our dataset for deploying as sample input into the network for teaching, practice and sharing for each downsampling layer. By our dataset contains 300 images that are allocated and segmented for employing in the learning process.

We grouped our sample into 7 classes. The first 3 classes were single label-class i.e. the 3 main classes which will have hats, eyeglasses and mustache. The remaining 4 classes were multi label-class and the rest of these 4 classes are a combination of 2 or more, by the last one class will be 3 classes combined. In creating a group of information that we will call this dataset as EHM set (i.e. Eyeglasses-Hat-Mustache), we provide a class of 30 to 60 images per class. We use a reference source for automatically creating the dataset through Roboflow [10].

This site can upload images any image you upload to Roboflow is considered a single source image and can be used in multiple datasets. Besides, The Roboflow is able to generate images that it can use Roboflow's preprocessing and augmentation tools to generate augmented versions of source images. The website of Roboflow creating dataset is available at https://roboflow.com/ . We have spilt our images in dataset into three parts: the training set = 65%, validating set = 25%, and testing set = 10% as displayed in Fig.2.

B. Spine Network

After we create our dataset as new inputs. In the next step, we have established the spine net for training our created dataset. We have chosen the LeNet [11] network simply for adaptation effectively. Our model has shown as below in the table 1.

In which to perform operations on the pre-processing layer, providing pre-processing data is essential, thus that the system network can learn fluently and efficiently. More than that, it also reduces learning with overfitting from the net. Convert images from any size which contain in any subfolder into the dataset to 50x50 pixel by pixel and then division operation to 255.0 for each channel (RGB) so that the data used as input is scaled in the range 0.0 to 1.0 to prevent the data from being too saturated in studying. Furthermore, we have switched the alignment of the color channels from BGR to RGB since the images provided may or may not be channeled as we want, which we want to be arranged in RGB only. The network we propose that uses approximately 3,630,000 parameters, which are all trainable learning parameters.

TABLE I. NUMBER OF DOWN SAMPLING STEP USAGE EACH LAYER (THE EXEMPLARY OF MULTI LABEL-CLASS)

Down sampling operations	**Image size**	**Channel size**	**Number of filtrations**
Source input	50x50	3 (RGB)	-
Layer1			
Convolution2D	50x50	20	5x5
Activation Fn. (ReLu)	50x50	20	-
MaxPooling2D	25x25	20	2x2
Layer2			
Convolution2D	25x25	50	5x5
Activation Fn. (ReLu)	25x25	50	-
MaxPooling2D	12x12	50	2x2
Layer3			
Fully connected	-	7200	-
Layer 4			
Dense (Hidden Layer)	-	500	-
Activation Fn. (ReLu)	-	500	-
Layer5			
Dense (Outputs)	-	7	-
Activation Fn. (Softmax)	-	7	-

During computation and self-learning in the spine network, we use Adaptive Moment estimation (Adam) [8] as a gradient-based optimization of stochastic performance. Simply, that's to proceed hyperparameter tuning into the learning model to reach parameters at the global minimal. Besides, we implement Categorical Cross-Entropy as the objective function to minimize the occurred error between the ground truth and the prediction since our samples, which contain EHS dataset, is a classified category into each class.

Finally, a little in post-processing that we have given an argmax function at outcoming result prediction to help make final predictions much easier and to increase confidence in the finale stage. By operating an interpreter and the programming, we took advantage of the Google Colaboratory [9] on-line interpreter, which makes computation more convenient because it has GPU support that accelerates computation when it comes to light computing.

IV. EXPERIMENTS & RESULT

In this section, we will explain how the experiment has been made and the result of our approach.

A. *Methodology*

In this project, we use Google Colab for training the model. We do several experiments. The first experiments classify the images into 4 classes. The second experiments classify the images into 5 classes. The third experiments classify the images into 6 classes. The fourth experiments multi classify the images into 3 classes. For all experiments, we use learning rate 0.0001 and decay 0.0001 divided by number of epochs. We also use Adam as the optimizer. When the model was trained, we set the steps per epochs to the number of training data divided by batch size. The batch size is 32. The total number of training set used is 851, 1065, 1272, and 614 (respectively for first, second, third, and fourth experiment). The number of validation set for each experiment is 0.25 from the training images. The number of testing set for each experiments is 61, 76, 91, 44 respectively. The training, validation, and testing has been augmented (rotation, flip, and resize) using roboflow [10].

B. *Result and Evaluation*

For all the experiments, we use the architecture model in the table 1. The first experiment is we try to classify image into 4 classes which are bang, eyeglasses, hat, test, and wavy. The first experiment uses the LeNet model architecture. We train the model with 25 epochs. Figure 3 show the training graph for the first experiments. The graph show the training result is not good enough because the validation loss is not stable.

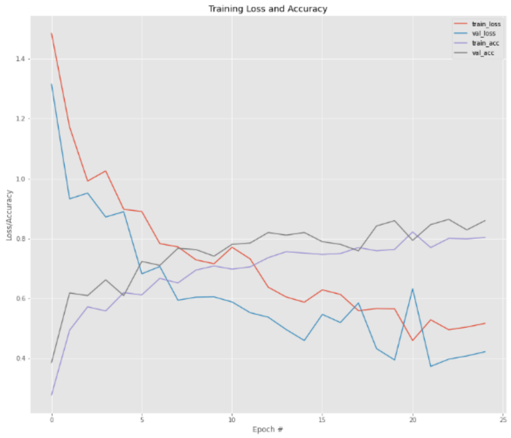

Fig. 3. Training Graph for the first experiment

The second experiment is we try to classify image into 5 classes which are bang, eyeglasses, hat, pointy nose, and wavy hair. LeNet model architecture is used in the second experiment. We train the model with 25 epochs. Figure 4 show the training graph for the second experiments. The training result is worse than the first experiments.

For the third experiment, we classify the image into 6 classes which are bang, eyeglasses, hat, oval face, pointy nose, and wavy hair. LeNet model architecture is also used in the third experiment. We train the model with 25 epochs. Figure 5 show the training graph for the third experiment. The training result is very bad because the validation loss is still high in the end of the training.

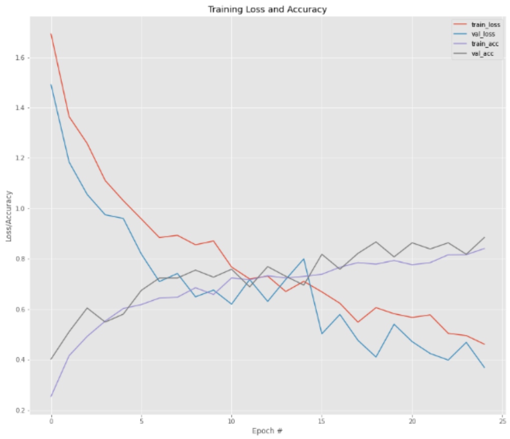

Fig. 4. Training Graph for the second experiment

The fourth experiment, last experiments, we make a multi classify the images into 3 classes which are hat, eyeglasses, and mustache. We use the LeNet model architecture for the fourth experiment. We train the model with 50 epochs. Figure 6 show the training graph for the fourth experiment. The training result is not good enough because there is many spike in validation loss which means sometimes the validation loss is high and sometimes low.

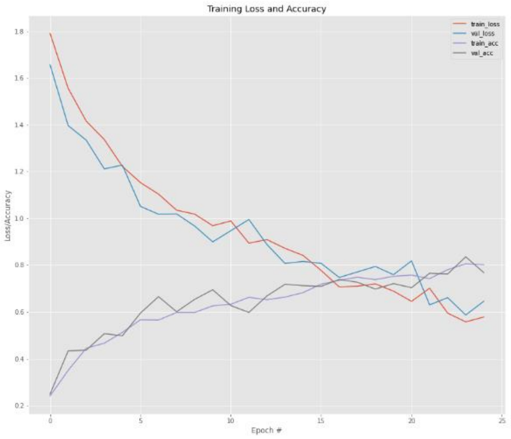

Fig. 5. Training Graph for the third experiment

We test each experiment using the prepared test image set. Tabel 2 show the test accuracy and the test error. As for calulation the test accuracy, we just calculate the correct predicted with ground truth divided by total test images.

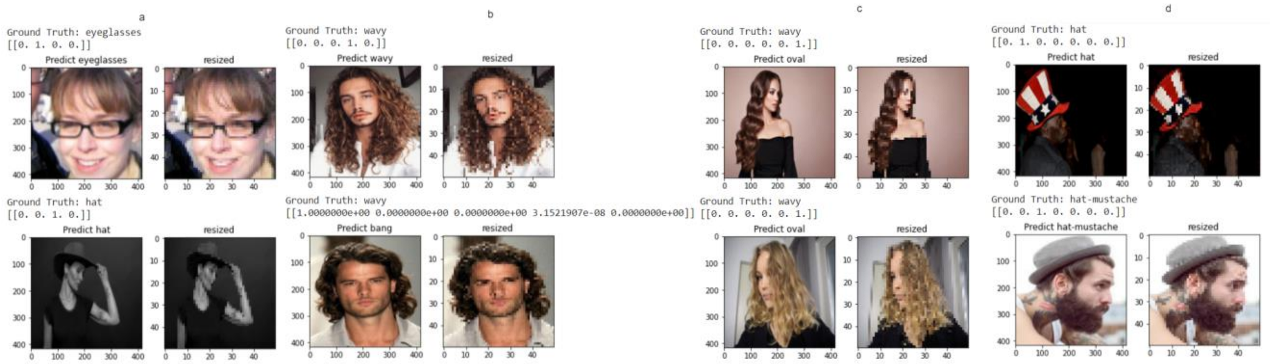

Fig. 7. Quality result from all experiment. (a) first experiment (b) second experiment (c) third experiment (d) fourth experiment

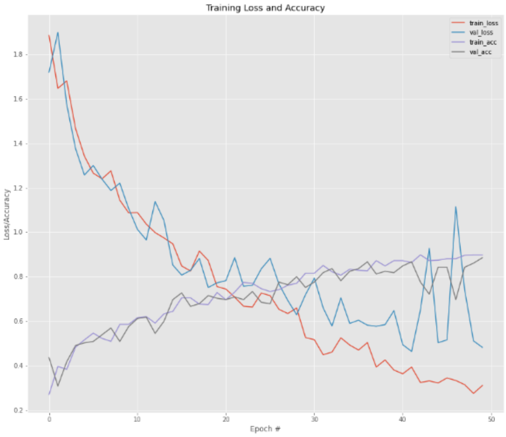

Fig. 6. Training Graph for the fourth experiment

TABLE II. TEST ACCURACY & TEST ERROR

Experiment	Classes	Test accuracy	Test error
1	4 classes	0.81	0.19
2	5 classes	0.75	0.25
3	6 classes	0.54	0.46
4	3 classes multi-clasifiication	0.81	0.19

V. Conclusion

We present an approach to classify facial attribute and accessories using LeNet model architecture. As for the single classification, the test accuracy gets lower when we add the number of class. As for the multi classification, we get the best result with test accuracy 0.81. Our approach is still far from good because there are still many possibilities to increase the accuracy such as adding the layer, trying different activation function, adding more dataset, etc.

VI. Contribution

Isak Martin Simbolon. Colected the dataset which are mustache and oval face (each 100 hundred images). Done the experiments.

Jirayu Petchhan. Colected the dataset which are hat and wave hair (each 100 hundred images). Done the experiments.

Richard Sugiarto. Colected the dataset which are bang and eyeglasses (each 100 hundred images). Done the experiments.

After collecting all the images, we collaborated to finish the multi-label classification together.

References

[1] Krizhevsky, A., Sutskever, I., & Hinton, G. (2012). ImageNet Classification with Deep Convolutional Neural Networks. In Advances in Neural Information Processing Systems (pp. 1097–1105). Curran Associates, Inc..

[2] Simonyan, K. & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, .

[3] C. Szegedy et al., "Going deeper with convolutions," 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, 2015, pp. 1-9, doi: 10.1109/CVPR.2015.7298594.

[4] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016, pp. 770-778, doi: 10.1109/CVPR.2016.90.

[5] G. Yue and L. Lu, "Face Recognition Based on Histogram Equalization and Convolution Neural Network," 2018 10th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), Hangzhou, 2018, pp. 336-339, doi: 10.1109/IHMSC.2018.00084.

[6] G. Wang and J. Gong, "Facial Expression Recognition Based on Improved LeNet-5 CNN," 2019 Chinese Control And Decision Conference (CCDC), Nanchang, China, 2019, pp. 5655-5660, doi: 10.1109/CCDC.2019.8832535.

[7] U. Park and A. K. Jain, "Face Matching and Retrieval Using Soft Biometrics," in IEEE Transactions on Information Forensics and Security, vol. 5, no. 3, pp. 406-415, Sept. 2010, doi: 10.1109/TIFS.2010.2049842.

[8] Z. Wu, Q. Ke, J. Sun and H. Shum, "Scalable face image retrieval with identity-based quantization and multi-reference re-ranking," 2010 IEEE

Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, 2010, pp. 3469-3476, doi: 10.1109/CVPR.2010.5539976.

[9] Google. *Google Colaboratory*. [Online]. Available from: https://colab.research.google.com/.

[10] Roboflow. *Roboflow: raw images to trained computer vision model*. [Online]. Available from: https://roboflow.ai/.

[11] Yann LeCun et al., "Gradient-Based Learning Applied to Document Recognition", in *PROC. of the IEEE*, November 1988. [Online]. Available: http://yann.lecun.com/exdb/publis/pdf/lecun-98.pdf.