Classifying Between Masked Faces and Normal Faces with CNN and SSH

Thang Pham[1], Bao Tran[2], Duy Pham[3], and Long Nguyen[4]

[1]University of Waterloo
[2]University of Rochester
[3, 4]University of Massachusetts Amherst
[1, 2, 3, 4]VinAI Research

Abstract

Environmental pollution is one the biggest problems we may face nowadays. Among different types of environmental pollution, air pollution is the most dangerous. Air pollution may cause diseases, allergies, and even death to humans as well as other living organisms such as animals and crops. Vietnam is consistently ranked as one of the most polluted countries in the world, and because of the low air quality, many people usually wear masks when they travel on the road. As a result, we may want to know the percentage of people wearing masks during a period of time as well as which group of people are most likely to wear masks when they go outside. To help answer those questions, this paper introduces a model that can classify between masked faces and normal faces in addition to the Mask Classifier dataset consisting of about 11000 face images of Vietnamese people. Experimental results with the Mask Classifier model show that it can achieve about **96.5%** accuracy during testing time. The code is available at https://github.com/aome510/Mask-Classifier.

1 Acknowledgement

This project was supported by VinAI research. The authors would like to thank Prof Minh Hoai Nguyen for mentoring our project, Dr Hung Bui for supporting our internship at VinAI, and Prof Phuong Dac Ho for connecting us to VinAI Research.

2 Introduction

In this paper, we will describe two phases: data preparing phase and training-testing model phase. For face detection purpose, we mostly use SSH (Single Stage Headless Face Detector) [1].

3 Data Preparing

In this section, we will describe four different face datasets including CelebA dataset [2], WiderFace dataset [3], MAFA dataset [4] and our dataset (Mask Classifier dataset). After preparing all listed datasets, we use **scripts\gen_data.sh** to combine them together into a single dataset for training and testing our model. In the combined dataset, we label 0 as masked face and 1 as normal face.

3.1 CelebA dataset

CelebA is a large-scale face attributes dataset with more than 200000 celebrity images. We use about 8% of CelebA dataset for our combined dataset. CelebA dataset can be downloaded at here. After that, we use **gen_data_celebA** function from **gen_data.py** to generate cropped celebA dataset (in order to do this, we must set up SSH beforehand).

3.2 WiderFace dataset

WiderFace is a face detection benchmark dataset consisting of about 32,203 images and 393,703 faces with a wide variety of scales, poses, and occlusions. We only use about 12000 normal, clear, and medium-occluded faces from this dataset. We have uploaded the modified version of WiderFace at here. After downloading WiderFace's modified version, we use **gen_data_widerface** function from **gen_data.py** to use a part of this dataset (we use about $9000 - 10000$ images for our training and testing purpose).

3.3 MAFA dataset

MAFA is a face dataset consisiting of about 30000 images in total and each image contains at least one masked face. MAFA train-test dataset and annotation can be downloaded here. There are six annotation attributes, but we only care about Location of faces, Occlusion degree, and Mask type attributes. We label a face with $Occlusion\ degree >= 2$ and $Mask\ Type \neq 3$ masked face, otherwise it is labeled normal face. Please read **gen_data_mask_train** and **gen_data_mask_test** functions from **gen_data.py** for further details about how to generate masked faces dataset (MAFA) for our combined dataset.

3.4 Mask Classifier dataset

Mask Classifier is a dataset consisiting of about 11000 face images of people on the road in Vietnam. Mask Classifer dataset and annotation can be downloaded here. Images from this dataset are generated from 9 videos with a total average length of about 3 mins. To generate these images, we consider an interval of $10 - 20$ frames from those videos and use SSH to crop faces from a video frame. After that, we put those cropped images to different folder for different videos.

We label an image 0 for masked face, 1 for normal face, and 2 for difficult-to-detect face or non-face. We use only faces with label 0 and 1 for training and testing our model. Please read **data\mask\classifier\README.txt** and **gen_data_mask_classifier** function from **gen_data.py** for further details.

4 Training and Testing Mask Classifier Model

4.1 Training

To train Mask Classifer model, we deploy Keras framework. To prepare the dataset for training and testing on Keras, we use **gen_data** function from **gen_data.py**. After that, we train our model with **train.py**. We deploy two different CNN architectures (resnet50 [5] and inception_resnet_v2 [6]). The default network architecture is resnet50. To try training on different networks, please refer to `https://keras.io/applications/` for further details. To train the network, first we initialize the network with imagenet weights. After that, we replace the last sofmax activation layer to sigmoid activation layer and change the loss function to Binary Crossentropy loss. We train the whole network without freezing any layers. During the training phase, we use Stochastic Gradient Descent with base learning rate of 0.0001, weight decay and momentum. We also apply different types of data augmentation. The network begins to stabilize and achieve around $98\% - 99\%$ on training set and $95\% - 96\%$ on validation set after about $1x$ *epoches*. Pretrained Mask Classifer model can be downloaded here.

4.2 Testing

We use **demo.py** to demo our model on videos - webcam and to test the model on different datasets. During the experiment, we observe that our model perform quite well on medium to large size face images. The model can achieve about **96.5%** accuracy when predicting. However, in some cases, because of the wide variety of textures on masks, as well as the large range of poses and occlusions, the model still predicts incorrectly. To demo our model on videos or large images, first we use SSH to detect faces from an image or a video frame (we only use the SSH pyramid option when demoing on images to imporve performance). After that, we deliver previous detected faces to our trained network to classify between masked faces and normal faces. We draw bounding boxes around detected faces with color yellow or green based on the classify results. We also display the probability of a face to be normal face and the percentage of masked faces on a picture. Overall, we achieve an average performance of about **10-15 FPS** on a Nvidia's GeForce GTX Titan X machine.

References

[1] M. Najibi, P. Samangouei, R. Chellappa, and L. Davis, "SSH: Single stage headless face detector," in *The IEEE International Conference on Computer Vision (ICCV)*, 2017.

[2] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[3] S. Yang, P. Luo, C. C. Loy, and X. Tang, "Wider face: A face detection benchmark," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[4] S. Ge, J. Li, Q. Ye, and Z. Luo, "Detecting masked faces in the wild with lle-cnns," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2682–2690, July 2017.

[5] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv 1409.1556*, 09 2014.

[6] C. Szegedy, S. Ioffe, and V. Vanhoucke, "Inception-v4, inception-resnet and the impact of residual connections on learning," *AAAI Conference on Artificial Intelligence*, 02 2016.