Report to head pose estimation using CNN

Songlin Piao[1]

***Abstract*— This is a technical report about head pose estimation. The baseline paper is from N. Ruiz et al. [1]. Authors use ResNet50 [2] to build a CNN network for estimating head pose (roll, pitch and yaw) from 2D images. This report tried to describe all the possible solutions to improve this baseline method.**

I. Topic1

Authors build a so-called hopenet using ResNet50 as a backbone. They conducted the tests on the AFLW2000 dataset. The results are shown in the Table 1 in the paper [1]. Through looking at the source code written by authors, it is found that authors used 66 bins in the range between -99° and 99°. Following solutions could improve the accuracy, but they are not yet proved due to the limited time and training resources.

- change the 66 bins to 99 bins and 132 bins
- test with ResNet 101 and 152, but this may slow down the speed.

II. Topic2

In order to make the algorithm satisfy the real time requirement, MobileNet is recommended. I have conducted a test with MobileNet V2. Please have a look at the file train_mobilenet2.py. The new network is trained with exactly same hyper parameters mentioned in the paper [1]. It is trained for 25 epochs using Adam optimization with a learning rate of 0.00001. The training time takes approximately 10.5 hours with GTX 1050Ti Laptop GPU.

TABLE I
Error Comparison between ResNet50 and MobileNet V2 on the AFLW2000 dataset.

	Yaw	Pitch	Roll	MAE
Multi-Loss ResNet50 ($\alpha = 1$)	6.920	6.637	5.674	6.410
Multi-Loss MobileNet-V2 ($\alpha = 1$)	10.9175	7.1786	6.6814	8.259

Table I shows error comparison of ResNet50 and MobileNet V2 on the AFLW2000 dataset. There is slight performance decrease in the case of MobileNet V2. As an opposite, MobileNet V2 has a faster inference speed and very small model size compared to ResNet50 as shown in Table II.

Fig. 1 shows the visualization of the head pose estimation from MobileNet V2 and Hopenet. Although MobileNet V2 shows higher error in the Table 1, but it seems in the first two images, the results from MobileNet V2 make more sense than the results from Hopenet. This needs to be further investigated.

[1]Songlin Piao - piaosonglin1985@hotmail.com

TABLE II
Speed and Model Size Comparison between ResNet50 and MobileNet V2.

	inference time	model size
Multi-Loss ResNet50 ($\alpha = 1$)	5.5845ms	95.9Mb
Multi-Loss MobileNet-V2 ($\alpha = 1$)	4.6118ms	10.1Mb

III. Further Improvements

So far, the input size 224x224 is used for training and testing. It is necessary to test with smaller input sizes such as 168x168 or 112x112. If the test results are not so bad, this would increase inference speed tremendously.

IV. Newly Added Source Codes

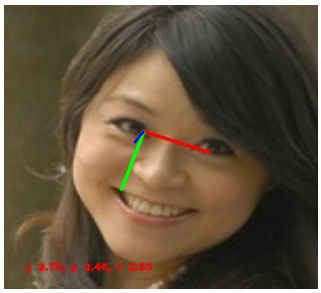
- extract_valid_files.py is used for extracting valid samples so that all roll, pitch and yaw are between -99° and 99°
- mobilenetv2.py is the implementation of MobileNet V2.
- train_mobilenet2.py is the training script using MobileNet V2.
- test_mobilenet2.py is the evaluation script using MobileNet V2.
- demo_mobilenet2_webcam.py is the demo program to estimate head pose from Webcam using MobileNet V2.
- the implementation of MobileNet V2 based model is added inside the file hopenet.py.
- MobileNet V1 is also implemented but not used yet.

In order to initialize the weights of MobileNet V2, the model mobilenetv2_1.0-0c6065bc.pth from `https://github.com/d-li14/mobilenetv2.pytorch` is used.

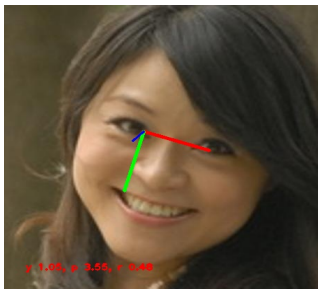
References

[1] N. Ruiz, E. Chong, and J. M. Rehg, "Fine-Grained Head Pose Estimation Without Keypoints," oct 2017. [Online]. Available: http://arxiv.org/abs/1710.00925

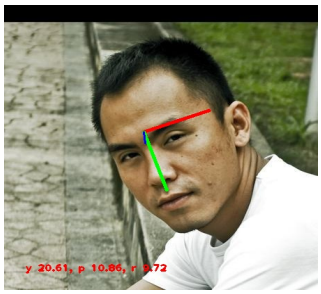
[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," dec 2015. [Online]. Available: http://arxiv.org/abs/1512.03385



(a)



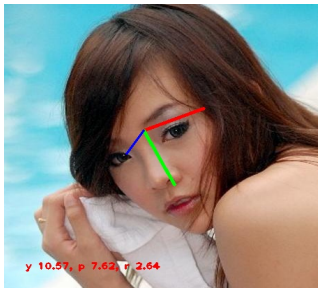
(b)



(c)

(d)

(e)



(f)

Fig. 1. The left column are the results from MobileNet V2 and the right column are the results from Hopenet.