Emmanuel OWONO (ISYMED) Wenqi LUO(ISE-VA) Xingjian HUANG(ISE-VA) Valex TEKOUO(ISE-VA)

ARTICLE SUMMARY

The purpose of this article is to describe the process on how to make real-time face detection and face recognition based on Convolutional Neural Network (CNN) using a camera on Raspberry Pi module B.

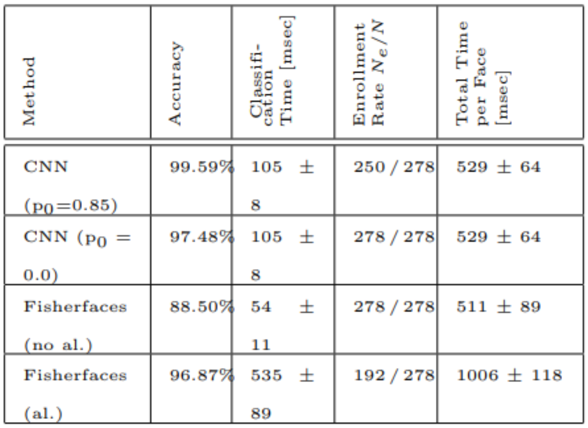
Tools: Raspberry Pi, camera module Pi NolR ,NVIDIA GeForce GTX 780 GPU

Libraries: NumPy, OpenCV, Theano

Therefore, they use 40 images per person (6 people) and took them in different environment (indoor and outdoor).

Once the images they took were not on the same size they need to reshape them and use LBP to reduce dependence on lighting. They reduced the dimension of the data by pooling to avoid overfitting. To manipulate all images, they used the size of 46*46 pixels by making CNN on making classification and regression to detect and identify the person. To train the model they used the method of gradient descent to reduce the scale of the loss function so our algorithm could have good performance. They used outdoor images for test and had the performance which varies from 88% to 99%. The model was then transferred to a Raspberry Pi.

They have this method using different method.



Method	Accuracy	Classifi-cation Time [msec]	Enrollment Rate N_e/N	Total Time per Face [msec]
CNN (p_0=0.85)	99.59%	105 ± 8	250 / 278	529 ± 64
CNN (p_0 = 0.0)	97.48%	105 ± 8	278 / 278	529 ± 64
Fisherfaces (no al.)	88.50%	54 ± 11	278 / 278	511 ± 89
Fisherfaces (al.)	96.87%	535 ± 89	192 / 278	1006 ± 118

Table 1: Accuracy and performance for various approaches. The time for classification also includes the time for preprocessing.

Finally, these authors got conclusions as follows:

1. As a benchmark they chose various OpenCV classifiers, and the highest performance was achieved using the Fisherface classifier.

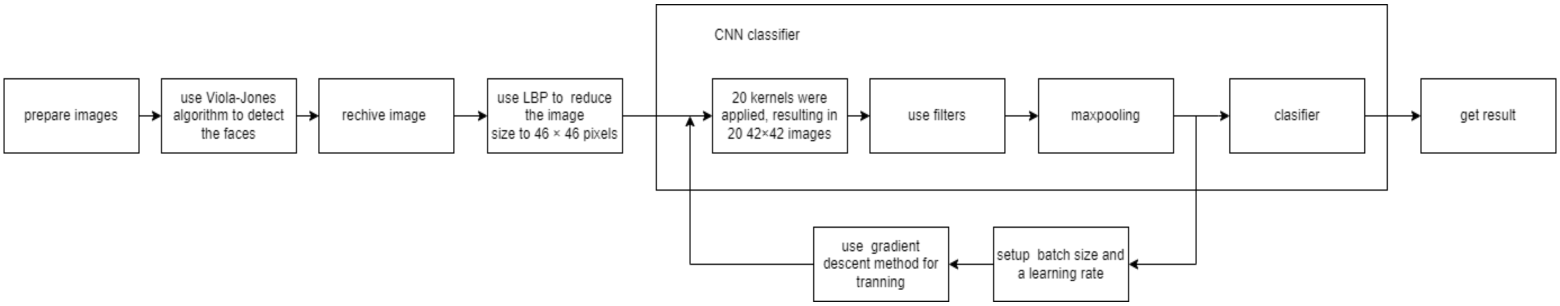
2. Without any pre-processing the CNN featured an initial recognition rate of 24%. Applying the LBP operator reduced the dependence on illumination, boosting the recognition rate to 82%.

3. This accuracy could be increased even further by rejecting those images for which the best estimate, returned by the regression.

4. A timing analysis on the Raspberry Pi revealed that the Viola-Jones algorithm required 423±64ms to detect a face in a given image. The CNN required another 105±8ms for face recognition whereas the Fisherface approach needed 535 ± 89ms for the same task.

And in conclusion, this approach provides significant speed and accuracy improvements compared to OpenCV, and is suitable for resource-constrained hardware

Schema bloc


prepare images
use Viola-Jones algorithm to detect the faces
rechive image
use LBP to reduce the image size to 46 × 46 pixels
CNN classifier
20 kernels were applied, resulting in 20 42×42 images
use filters
maxpooling
clasifier
get result
setup batch size and a learning rate
use gradient descent method for tranning