

Are face recognition algorithms parallelizable?

Óscar A. Alvarado-Morán

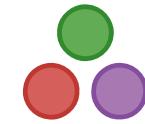
Oscar A. Esquivel-Flores

National Autonomous University of Mexico
Research Institute in Applied Mathematics and Systems

Abstract

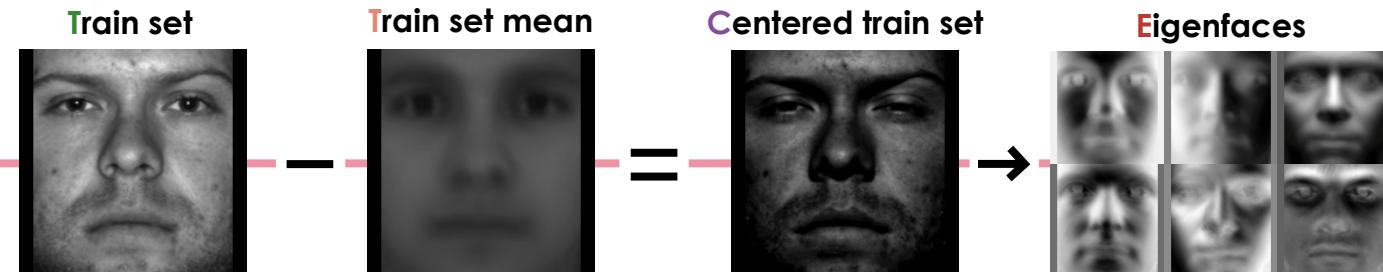
We implemented three face recognition algorithms using the well know Yale Face Database B: Eigenfaces, Histogram of Oriented Gradients (HOG) and Convolutional Neural Networks (CNN) running sequentially, then we analyze each of them to see which parts were parallelizable. Finally we just trained the CNN using the CUDAnative.jl implementation used in the Flux library.



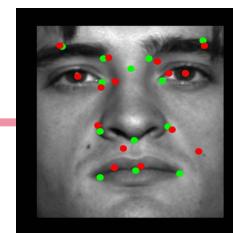


Time traveling with face recognition algorithms.

— 1991 - Eigenfaces —

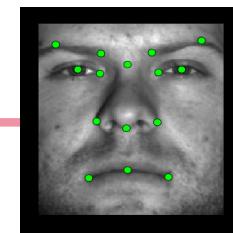


Predicted and real landmarks

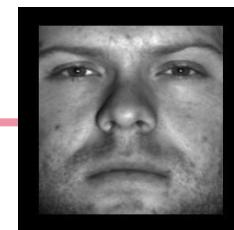


HOG Features

Landmarks in the train set by hand

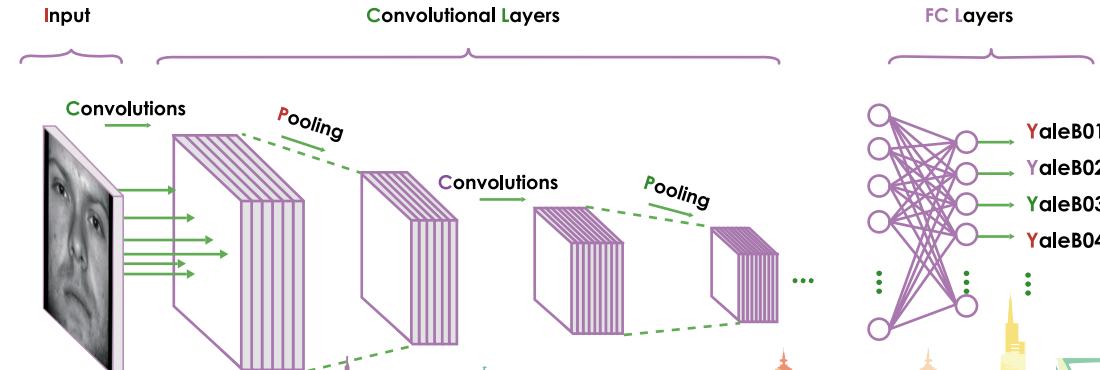


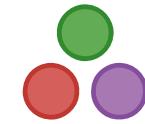
Train set



— HOGs - 1995 —

CNN - 1997





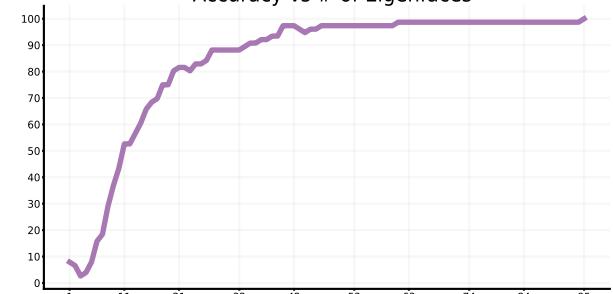
Sequential Parallel Accuracy

Eigenfaces
SVD:

47.807 s

-

Accuracy vs # of Eigenfaces



HOG
Features

67.003 ms

-

88.8 %

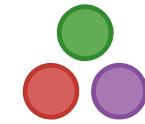
CNN
(Training)

10.18 s
each epoch,
7 epochs

0.79 s
each epoch,
7 epochs

96.05%





Conclusions

As we can see, Eigenfaces algorithm can reach 100% in accuracy but is the one which spend more time at training (SVD decomposition), so we suggest to implement it in parallel for future work. HOG algorithm is the fastest one but this extra time was spent doing the landmarks by hand, we don't suggest this algorithm to be used in real projects. We think that CNN is the winner in this comparison but is the one which has more abstraction at coding, this includes that Flux has some problems working with GPUs. To keep learning about parallelism in this subject, for future work we suggest face detection algorithms and face tracking.

References & More

