Optimization of Eigenface Algorithm for Facial Recognition Using AVX2

A Project Proposal Presented to
the Faculty of the College of Computer Studies
De La Salle University

In Partial Fulfillment
of the Requirements for the Degree of
Bachelor of Science in Computer Science

by
Ibaoc, Christian Gabriel P.
Sia, Chantal Hyacynth S.
Valero, Nigel Kristoffer C.
Veloso, Antonio Luiz G.

Submitted to:
Professor Roger Luis Uy

June 27, 2024

ABSTRACT

Eigenfaces Algorithm is a popular and relatively old algorithm invented in the 1980s. It consists of a series of conversions and calculations that aim to isolate the specific features of certain training images, and when given an input image, it can detect the closest training image to the input. This means that the algorithm itself needs image inputs, both for training and for application purposes. The group's proposed project is to optimize the existing Eigenfaces algorithm, used predominantly for facial recognition, using SIMD AVX2 capabilities. The system has already been implemented by the group in MATLAB in a previous project, however the detection was limited to having a significant delay as there are multiple heavy calculations. The goal is to implement a Facial Recognition system in a C program that detects faces in real time using the optimizations stated.

I. Description of the Project

The Eigenface algorithm is an old but still popular algorithm for special feature detection. The concept is simple enough, provide the algorithm with training images, and when provided with another image as an input, it should be able to detect if that particular image is identifiable from the training images. The actual framework is much more complicated, as seen in Figure 1.

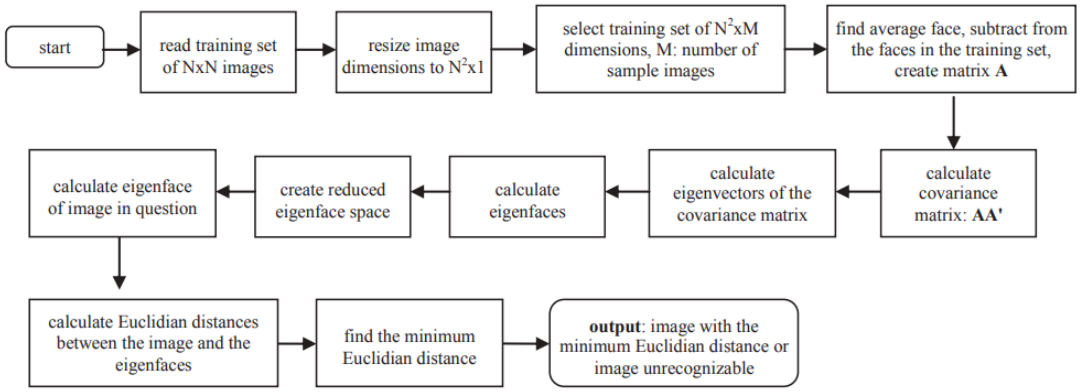

Figure 1: The flowchart of Eigenfaces [4]

The inputs of this system can be classified depending on the stage the system is currently in. At the start, the algorithm will be given a set of inputs to initialize its starting eigenvector and covariance matrix. After this stage, the inputs can now be considered samples for the algorithm to detect. In this stage, the group's goal is to optimize the code well enough to allow for real time face detection.

The algorithm itself performs multiple vector calculations, all of which are proposed areas of optimization. The most definite areas are the calculations in the training stage, which include the normalization of each training image to get Ψ, the adjustment of each vector to the mean, Φ_i, the calculation of the covariance matrix C, and the calculation of eigenvalues Λ and eigenvectors V.

$$\Psi = \left(\frac{1}{M}\right)\sum_{i=1}^{M}\Gamma_i$$

$$\Phi_i = \Gamma_i - \Psi \quad \text{for i} = 1, 2, \ldots, \text{M}$$

$$C = \frac{1}{M} \sum_{i=1}^{M} \Phi_i \Phi_i^T$$

$$C\mathcal{V} = \Lambda\mathcal{V}$$

Figure 2: The formulas associated with areas of optimization [2]

To date, an existing implementation of the Eigenface algorithm with parallelism is implemented using CUDA functionality [1]. While the proposed concept is similar, the difference lies in the method of parallelism, as well as the areas of optimization. In their paper, there were no optimizations made in the recognition stage, and instead their focus was on the training stage. In this study, the group will also be implementing AVX2 optimizations in the recognition phase. The sample image to be recognized will also undergo the formulas in Figure 2, with an added calculation of euclidean distance to each adjusted eigenvector.

As all the input images of size $N \times N$ will be converted to vectors of size $N^2 \times 1$, passing these values into the SIMD registers will not be too difficult to manage. From there, the calculations shown in Figure 2 are all vector-oriented, and thus with the proper implementation of AVX2 instructions, the group will acquire the proper results. The group will likely be using YMM registers for higher degree of parallelism.

There are other areas in the algorithm that the group will likely not include unless otherwise stated. These are the matrix transformations calculations, which includes the conversion of the $N \times N$ matrix to a $N^2 \times 1$ vector. In Singh's paper, their algorithms also include the process of sorting the vectors, of which will not be included in the proposed optimization areas of this study.

The expected output of this study is a functional facial recognition system complete with a GUI that allows the user to detect faces, or no face otherwise, in real time, with the limitation that the algorithm is given the proper training set to

begin with. The group will expect a decrease in execution time after implementing AVX2 optimizations to multiple areas of interest in the algorithm.

II. Target Implementation

The algorithm will be implemented in C, as a standalone application if possible, capable of storing and clearing training images, and switching to recognition mode. In C, external assembly functions are possible, and so using AVX2 SIMD will be effective. Imported GUI libraries will be used, as well as a library for image acquisition, OpenCV[3].

III. References

[1] M. R. Kawale, Y. Bhadke and V. Inamdar, *"Parallel implementation of eigenface on CUDA,"* 2014 International Conference on Advances in Engineering & Technology Research (ICAETR - 2014).

[2] S. Singh, A. Tripathi, A. Mahajan, and S. Prabhakaran, *"Analysis of Facial Recognition in MATLAB"*. International Journal of Scientific & Engineering Research, February 2012.

[3] *"What is OpenCV Library?"*, April 2024, GeeksForGeeks. Retrieved from: https://www.geeksforgeeks.org/opencv-overview/

[4] M.üge Çarıkçı, Figen Özen, *"A Face Recognition System Based on Eigenfaces Method"*, Procedia Technology, Volume 1, 2012.