Scripta

Fall 2023, Vol. 2, No. 3, 101-105
ISSN: 2769-9242

Comparative Performance of Facial Recognition Algorithms

*Khalil Lindo**, *Seonghyeon Hong*†, *Yoonwoo Ku*†, *Cyrus Khabbaz*‡, *Eshaan Dixit*‡

ABSTRACT

Artificial Intelligence and Machine Learning algorithms are growing in popularity and their influence is rising into our everyday lives. Facial detection technologies and recognition technologies have become crucial to maintaining security and privacy in the modern world. Nevertheless, the potent impacts of algorithmic bias and errors persist. Currently, many generic facial recognition softwares are drawing improper conclusions based on skin tone and similarities. These software applications use algorithms that depend on supervised learning datasets. Without representative data from minority populations, the software struggles to distinguish individuals within the same race. Dubbed the 'coded gaze' by MIT scholar Joy Buolamwini, its implementation into police identification software could lead to inaccuracies in suspect identification, significantly impacting marginalized communities. Our study determined that flaws originate most notably with changes in illumination and facial expressions. Testing and implementing algorithms that rely on unsupervised learning or make predictions through adaptable supervised learning would dramatically reduce the inaccuracies in modern facial recognition software. To determine the quickest and most effective facial recognition algorithm in scenarios with varying illumination and facial expressions, we conducted research and implemented five popularized algorithms. The study involved seven different students.

Introduction

With the popularization of Artificial Intelligence and Machine Learning algorithms, there has been a rise in their implementation into our everyday lives. Facial detection and recognition have become crucial to maintaining security and privacy in the modern world. However, the impacts of algorithmic bias and mistakes are potent. Currently, many generic facial recognition softwares are drawing improper conclusions based on skin tone and similarities; some can't detect darker skin tones at all while others can't distinguish between people of the same race. Coined by MIT scholar Joy Buolamwini as the "coded gaze," if implemented into police identifica-

*Senior Student, Fulton Science Academy.
Contact: alindo@fultonscienceacademy.org
†Sophomore Students, Gyeonggi Science High School, Suwon, South Korea.
‡Junior Students, Fulton Science Academy.

tion software it could lead to inaccuracies in suspect identification heavily aimed at different races of people [1]. The flaws are most notable when there are changes in illumination and facial expressions. Our team decided to research and implement five popularized facial recognition algorithms in order to determine the quickest and most effective one based on their F1-Score, accuracy, precision, and recall when tested under different illumination and facial expression. We used the Eigenfaces, Fisherfaces, SURF, CNN, and LBPH algorithms.

Theoretical Background

Eigenfaces Algorithm

Eigenfaces are the name given to a set of eigenvectors that are used in the computer vision problem of facial recognition [2]. A set of eigenfaces is generated by performing a mathematical process called Principal Component Analysis (PCA) [3]. Eigenfaces that are created will appear as a black and white image with the light areas and dark areas arranged in specific patterns. These patterns show how different features of a face are singled out to be evaluated and scored. Some examples of these features include symmetry, the style of facial hair, the size of the nose or mouth, the location of the hairline, etc. Some eigenfaces have patterns that are simpler and the image of the eigenface may not look like a face. When used in facial recognition, multiple images are saved as a collection of weights that describe the contribution of each eigenface to a specific image. Then, methods such as the nearest-neighbor method are used to find the Euclidean distance between two vectors, where the minimum can be classified as the closest subject.

Fisherfaces Algorithm

Fisherfaces algorithm is an algorithm that is used after eigenfaces to classify the images better [4]. Before using the Fisherfaces algorithm, PCA is used to extract features from the image and reduce the dimension of the images. Then, during the training process, Fisherfaces algorithm uses Linear Discriminant Analysis (LDA) to PCA-transformed data to maximize the distances between the means of the different classes and minimize the distances between each image from the same class [5]. LDA finds a subspace that maps the images of the same person in a single spot and images of the different people apart from each other. The basis vectors of such subspaces are called Fisherfaces. Then during the testing process, new images get projected to an eigenspace. The new image is then compared to the closest person.

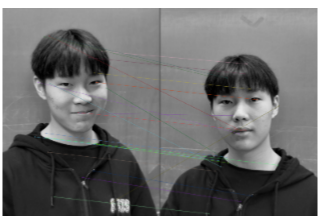
SURF Algorithm

The Speeded-Up Robust Features (SURF) Algorithm, extracts facial features from digital images and establishes local correspondence between a pair of images: the reference image and the image being compared [6]. SURF is faster and more efficient than its predecessor, the Scale-Invariant Feature Transform (SIFT) algorithm [7]. It achieves this by utilizing interest points and performing local analyses on these points to extract and store facial features from digital images.

During local analyses, SURF employs Random Sample Consensus (RANSAC) to efficiently eliminate matches that do not adhere to the homography restraint. It highlights the confirmed matches between interest points in the two images.

While SIFT descriptors are occasionally more accurate, SURF detectors are invariant to rotation, scale, and brightness. They utilize the Hessian Matrix and second-order Gaussian derivatives to enhance real-time image analysis [8].

SURF Algorithm Results



Same Person

Different Person

Table 1: Surf Algorithm Results: Images

- The F1-score, accuracy, precision, and recall of the SURF algorithm couldn't be measured in this experiment due to the way the algorithm analyzes an image. Nevertheless, it can return a new image for comparison and has detector lines to display similarities.

- In the left image, numerous detector lines between the two imported images indicate that they represent the same person. Conversely, the right image shows only one detector line, which is between the school logo, indicating that it's the only similarity between the two images.

CNN Algorithm

Utilizing the Keras, Tensorflow, and Numpy modules, Convolutional Neural Network (CNN) employs neural networks to scan and train images for identification [9–11]. The algorithm scans the entire image, creates sub-images, and then saves each value in a Numpy array. By using Keras and Tensorflow, the algorithm saves each sub-image as a vector value and employs each sub-image as inputs for the neural network. The neural network is constructed through the logistic regression model represented by the provided equation. Each input is stored within a series and subsequently summed to generate an output.

The entire process can be divided into four steps. The first two steps, convolution and subsampling, run concurrently. During convolution, a specific portion of an image is scanned, while subsampling analyzes each portion and saves the pixels in a vector. Every processed image is referred to as an epoch, which is then stored as a branch for a neural network. After the completion of Convolution and Subsampling, the process proceeds to the Full Connection Step, which reconstructs the image from the sub-image scan [12]. Subsequently, the Gaussian Step compares the tested image with the trained recreated image and generates a linear graph depicting its accuracy level.

LBPH Algorithm

LBPH stands for Local Binary Patterns Histograms [13]. LBP (Local Binary Pattern) is a texture operator that outputs binary values based on thresholding the neighbors of each pixel. In simple terms, higher values of the neighbor are assigned a higher binary value, whereas lower values receive lower binary values. After creating the binary matrix, the binary digits are converted into decimals and then set to the original center value. Subsequently, the image is recreated, resulting in our LBPH outcome: a new image that emphasizes the distinct characteristics of the subject.

By employing the grid X and grid Y values, we divide the image into smaller grids to create histograms representing occurrences of each pixel intensity. The histograms from each region are then combined into one concatenated histogram. The values in this histogram are compared to those of another image to evaluate the accuracy of the two images [13].

Experimental Process

Downloaded and fine-tuned, the testing of each algorithm involved initially comparing the same person under various lighting conditions, ranging in brightness, and facial expressions, including smiling, pouting, frowning, and grinning. Each comparison required the use of a common dataset to determine whether one person was different from the other.

Results

$$\text{Accuracy} = \frac{t_p + t_n}{t_p + t_n + f_p + f_n}$$

$$\text{Precision} = \frac{t_p}{t_p + f_p}$$

$$\text{Recall} = \frac{t_p}{t_n + f_p}$$

$$\text{F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Figure 1: Various Measurements

The equations above illustrate the F1-Score, accuracy, precision, and recall. According to the values in the graph shown below, Fisherfaces emerged as the most effective algorithm, while CNN demonstrated the potential to be the most adaptable. CNN's image processing can be subdivided into multiple epoch levels, and each can include its own Gaussian Step, thereby increasing its adaptability. Meanwhile, Fisherfaces relies on multiple tests with the same person and, at most, can discern the differences between two people but not a multitude of people. Therefore, CNN would be the best algorithm to implement into facial recognition software.

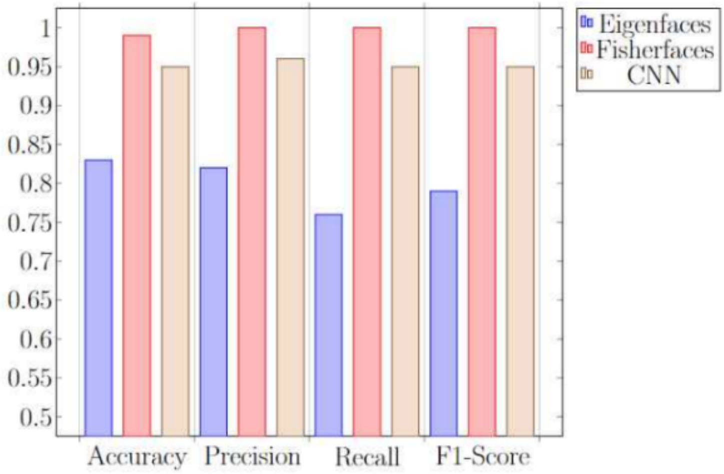


Figure 2: Measurement Values for Various Algorithms

Conclusion

Based on the results the fisherfaces algorithm would be the best to implement into facial recognition software that strictly relies on comparison between two people. However, the CNN seems to be the most adaptable for different conditions and is the more reliable algorithm to implement into the real world. In the future, we plan to test each algorithm with larger sets of data to detail and analyze each algorithm's strengths and weaknesses. We plan to develop a machine learning program that chooses which algorithm best suits a certain data set after evaluating the strength and weakness of each algorithm. Even though CNN is the best to use in most cases, there are still weaknesses associated with it that can be covered with another algorithm. Creating a program that can evaluate a situation and choose which algorithm to use based on that situation will practically eliminate further bias or inaccuracies.

References

1. Buolamwini, J. *Fighting the 'Coded Gaze'* 2022. `https://www.fordfoundation.org/news-and-stories/stories/fighting-the-coded-gaze/`.

2. Jalled, F. Face recognition machine vision system using Eigenfaces. *arXiv preprint arXiv:1705.02782* (2017).
3. Turk, M. A. & Pentland, A. P. *Face recognition using eigenfaces* in *Proceedings. 1991 IEEE computer society conference on computer vision and pattern recognition* (1991), 586–587.
4. Anggo, M. & Arapu, L. *Face recognition using fisherface method* in *Journal of Physics: Conference Series* **1028** (2018), 012119.
5. Adouani, A., Henia, W. M. B. & Lachiri, Z. *Comparative study of face detection methods in spontaneous videos* in *2019 International Conference on Control, Automation and Diagnosis (ICCAD)* (2019), 1–6.
6. Bay, H., Tuytelaars, T. & Van Gool, L. *Surf: Speeded up robust features* in *Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part I 9* (2006), 404–417.
7. Lowe, D. G. *Object recognition from local scale-invariant features* in *Proceedings of the seventh IEEE international conference on computer vision* **2** (1999), 1150–1157.
8. Oyallon, E. & Rabin, J. *An analysis of the SURF method* 2015. `https://www.ipol.im/pub/art/2015/69/?utm_source=doi.`.
9. Chollet, F. *et al. Keras* `https://github.com/fchollet/keras`.
10. Martín Abadi *et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems* Software available from tensorflow.org. 2015. `https://www.tensorflow.org/`.
11. Harris, C. R. *et al.* Array programming with NumPy. *Nature* **585,** 357–362. `https://doi.org/10.1038/s41586-020-2649-2` (Sept. 2020).
12. Martinez, J. *Introduction to convolutional neural networks vnns* 2020. `https://aigents.co/data-science-blog/publication/introduction-to-convolutional-neural-networks-cnns`.
13. Patil, S. & Kavitha, H. *Implementation of LBPH Algorithm for Covid Patient Isolation Violation Detection* in *2022 IEEE 7th International Conference on Recent Advances and Innovations in Engineering (ICRAIE)* **7** (2022), 20–24.