© Universiti Tun Hussein Onn Malaysia Publisher's Office

JSCDM

Journal homepage: http://penerbit.uthm.edu.my/ojs/index.php/jscdm
e-ISSN : 2716-621X

Journal of Soft Computing and Data Mining

Face Detection and Recognition Using OpenCV

Ramadan TH. Hasan[1*], Amira Bibo Sallow[2]

[1]IT Department, Technical College of Informatics Akre,
Duhok Polytechnic University, Duhok, Kurdistan Region, IRAQ

[2]College of Engineering,
Nawroz University, Duhok, Kurdistan Region, IRAQ

DOI: https://doi.org/10.30880/jscdm.2021.02.02.008
Received 21 May 2021; Accepted 05 August 2021; Available online 15 October 2021

Abstract: Intel's OpenCV is a free and open-access image and video processing library. It is linked to computer vision, like feature and object recognition and machine learning. This paper presents the main OpenCV modules, features, and OpenCV based on Python. The paper also presents common OpenCV applications and classifiers used in these applications like image processing, face detection, face recognition, and object detection. Finally, we discuss some literary reviews of OpenCV applications in the fields of computer vision such as face detection and recognition, or recognition of facial expressions such as sadness, anger, happiness, or recognition of the gender and age of a person.

Keywords: OpenCV, Face Detection, Object Detection, Eigenfaces, Faster R-CNN, Fisherfaces

1. Introduction

Computer Vision is one of the most fascinating and challenging tasks in the field of Artificial Intelligence. Computer Vision serves as a link between computer software and the visuals we see around us. It enables computer software to comprehend and learn about the visuals in its environment. As an example: The fruit is determined by its color, shape, and size. This job may seem simple for the human brain, but in the Computer Vision pipeline, we first collect data, then conduct data processing operations, and then train and educate the model to learn how to differentiate between fruits based on size, shape, and color. The main goal is to identify and comprehend the images and offer new images that are more useful for us in different life fields [1], [2].

The term "OpenCV" is an abbreviation for "open source computer vision." The architecture is made up of software, databases, and plugins that are pre-programmed with support for integrating computer vision applications [3]. It is one of the most used toolkits with a large developer group. It is well-known for the size at which it builds real-world usage cases for industrial use. OpenCV follows C/C++, Python, Java programming languages and can be used to build computer vision software for desktop and smartphone platforms such as Windows, Linux, macOS, Android, and iOS. The most recent releases are OpenCV-4.5.2 and OpenCV-3.4.14. It is free and open-source, as well as simple to use and install. It is intended for numerical productivity with a heavy emphasis on real-time applications. The first version was in the C programming language; however, its success increased with the release of Version 2.0, which had a C++ implementation [2]. C++ is used to create new features. OpenCV can be downloaded for free from http://opencv.org. This platform includes the most recent distribution update (version: 4.5.2) as well as older iterations. Photos must be in BGR or Grayscale format in order to be displayed or saved via OpenCV. Otherwise, unfavorable outcomes could occur [1].

Face detection is a form of computer vision that aids in detecting and visualizing facial features in captured pictures or real-time videos. This type of object detection technique detects instances of semantic artifacts of a given class (such as people, cars, and houses) in digital pictures and videos. Face recognition has become increasingly important as technology has advanced, especially in fields such as photography, defense, and marketing [4], [5].

*Corresponding author: Ramadan.honor@gmail.com

Recognition is a modern field of study that has piqued the interest of researchers since it becomes simple to use by using OpenCV-based Python. Face recognition technologies have a variety of applications in public protection, entertainment, man-machine contact, and social networking, such as Facebook's automated tag recommendation on images. It's also been seen in educational and non-educational institutes' attendance control, financial offices, voter registration, and other areas [6], [7]

In this paper, we emphasize the important role of OpenCV in face detection and face recognition, what algorithm can be used in OpenCV for face detection and face recognition, then state the OpenCV modules and explain OpenCV based on Python and mention the applications for OpenCV are. Finally, we assessment and compared recent literature reviews that use OpenCV to detect and recognize the human face in a variety of fields in order to improve human life.

The rest of this paper is organized as follows: in section 2, face detection is described. In section 3, face recognition is explained. In section 4, OpenCV library and OpenCV algorithm are explained. In section 5, Modules of OpenCV are explained. In section 6, OpenCV based on Python is described. In section 7, the assessment of literature reviews and comparison table are discussed. Section 8 concludes this paper.

2. Face Detection

Face detection has received much attention in recent years because of its applications in computer and human interaction. Face detection is a subset of image processing. Image processing is primarily a technique for compressing, improving, or extracting valuable information from images. Facial recognition technology can identify single or many faces in a picture, removing unwanted background noise. A face identification algorithm must basically categorize pictures into two groups based on whether or not they include a face. The face detection algorithm's goal is to thoroughly examine the picture, identify the existence of faces in the image, and remove the background from the image. Face detection mistakes are classified into two types: false negative and false positive. A false positive occurs when a face is identified in a picture that does not include any faces. A false negative occurs when the algorithm rejects the existence of anything in the picture. The detection rate is the ratio of the number of faces identified by humans to the number of faces detected properly by the system. The detection rate of the face detection algorithm should be as high as possible[1], [8].

3. Face Recognition

Facial recognition is the world's most advanced and quick biometric technology. It takes advantage of the most visible human body component, the face, in a non-intrusive way. According to worldwide data, most individuals are unaware of the face recognition process that is taking place on them, making it one of the least invasive procedures with the least amount of delay. The facial recognition algorithm examines the many features of a face in the input picture. This biometric has been extensively, and maybe exaggeratedly, lauded as a great method for identifying possible dangers such as terrorists, scam artists, and so on, but it has yet to gain widespread acceptance in high-level use. Biometric face recognition technology is expected to surpass fingerprint biometrics as the most common method of user identification and authentication in the near future [8], [9], [10].

4. OpenCV Library

It is a massive open-source image processing, machine learning, and computer vision library. OpenCV is compatible with a broad range of programming languages, including Python, C++, and Java. It will analyze photographs and videos to recognize artifacts, faces, and even human handwriting. When paired with many other libraries, like Numpy, a high-performance library for turning machines, achieve a good performance; that is, all services that can be performed in Numpy can also be integrated with OpenCV.It is written based on C++ and has a C++ interface as its main interface, but it also has a less robust but still detailed older Language training. Both the latest technologies and algorithms are visible in the C++ GUI. Python, Java, and MATLAB/OCTAVE bindings are available [3]. Wrappers in a variety of programming languages have been created to promote broader acceptance. JavaScript plugins for a variant of OpenCV functions are published as OpenCV.js in version 3.4, which can be used on web platforms. The OpenCV project, which was officially unveiled in 1999, was originally Intel's research program to support CPU-intensive applications [11].OpenCV is a popular platform for implementing face detection and recognition algorithms. The following are some often used OpenCV algorithms.

4.1 Haar Cascade

Haar Cascade is an effective method for detecting objects. It's a machine-learning-based method in which a cascade of actions is learned from a large number of positive and negative images. It becomes used to seeing things in different frames [12]. Fig. 1 shows the view of the Haar cascade classifier. Fig. 2 shows the Haar cascade flowchart.

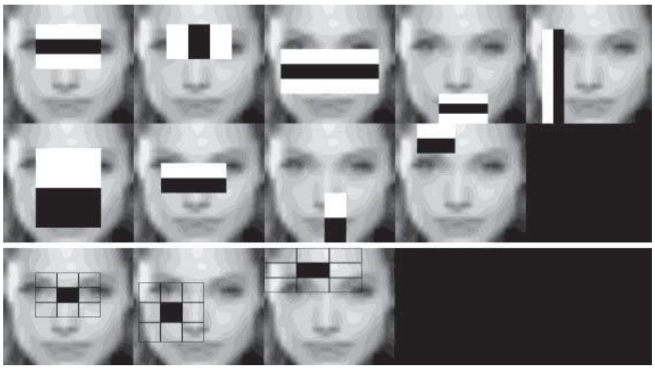


Fig. 1 - View of Haar cascade classifier

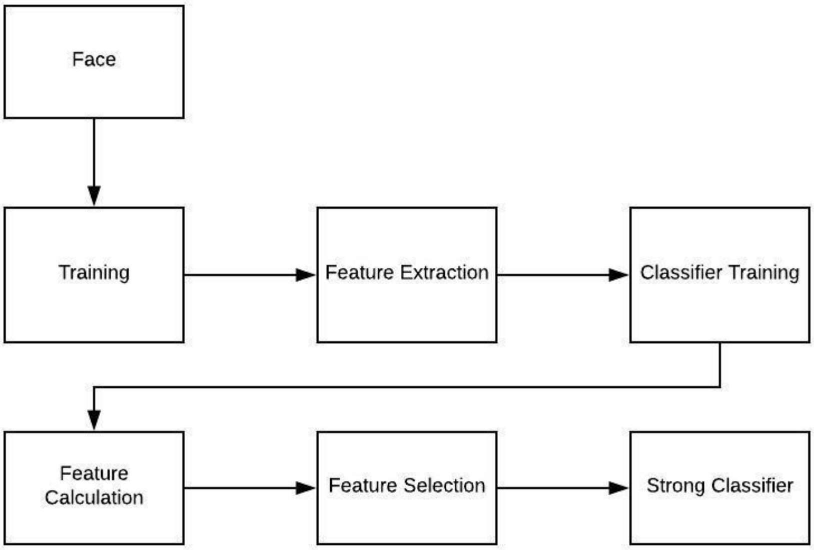


Fig. 2 - Haar Cascade flowchart [1]

4.2 LBP (Local Binary Pattern)

It is a simple but effective texture operator that labels pixels in an image by thresholding the pixels' neighborhood and treating the result as a binary number. The LBP texture operator has become a common approach in a variety of applications due to its discriminative power and computational simplicity. It can be viewed as a unifying solution to texture analysis's historically divergent statistical and structural models. The LBP operator's robustness to monotonic grayscale changes induced, for example, by illumination variations is perhaps its most significant property in real-world applications. Another key feature is its computational simplicity, which allows it to analyze images in difficult real-time scenarios [13]. Fig. 3 shows the description of facial expressions with local binary patterns.

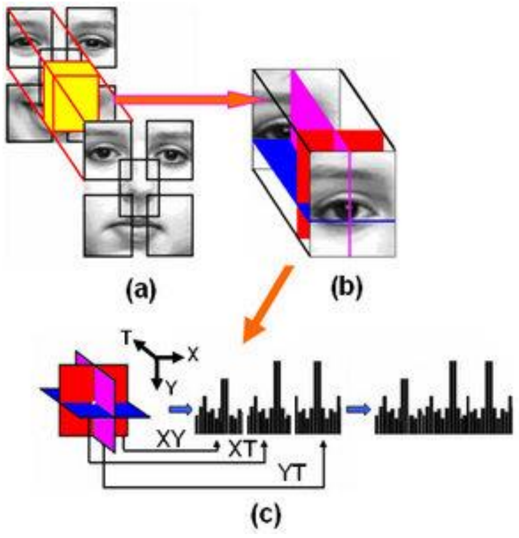


Fig. 3 - Description of facial expressions with local binary patterns [13]

4.3 EigenFaces

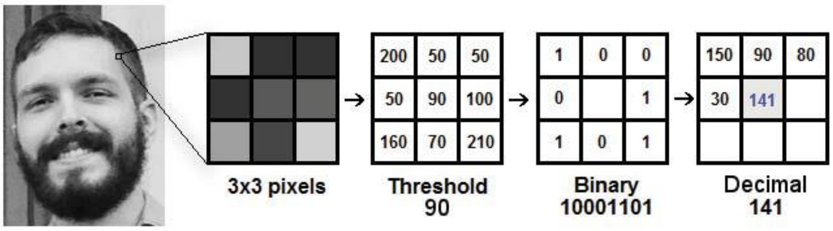
It is a method that employs PCA (Principal Component Analysis) to minimize dimensionality and find the strongest vectors for distributing facial images onto existing facial spaces. The primary goal of PCA is to identify the best vectors to explain the distribution of facial images in picture space into face space. According to the eigenvalue distribution, m eigenvector is used to construct the principal component amount. Eigenvector and eigenvalue are calculated from the qualified facial image's covariance matrix. The eigenvector is sorted by eigenvalue (high to low) and M first eigenvectors are chosen to form the principal variable [1], [14].

4.4 FisherFaces

It is a face recognition system that several researchers have demonstrated to identify faces accurately. FisherFace is a calculation model that combines the PCA (Principal Component Analysis) calculation model with Fisher's Linear Discriminant (FLD). PCA is used to minimize input data in order to simplify and accelerate the FLD operation. On the other hand, FLD is used to generate a distribution matrix to aid in classification and identification. A series of FisherFaces is generated using the PCA and FLD calculation models. This facial recognition process consists of four major steps: face identification, PCA estimation, calculation, and classification [15].

4.5 LBPH (Local Binary Pattern Histogram)

LBP is a highly effective texture operator. That compares each neighboring pixel's threshold value to the value of the center pixel. It takes into account outcomes in the context of binary numbers. LBP is a common technique in a variety of applications due to its discriminative strength and simplicity. LBP was identified for the first time in 1994. Since then, it seems to have evolved into a more efficient texture classification algorithm. It was later discovered that combining LBP with histograms of directed gradient descriptors increases its accuracy on the same dataset. LBP has additional capabilities such as monotonic grey-scale improvements and statistical simplicity, allowing it to interpret images in real-time applications [1]. Fig. 4 shows the LBPH algorithm for face recognition.



200	50	50
50	90	100
160	70	210

Threshold
90

1	0	0
0		1
1	0	1

Binary
10001101

150	90	80
30	141	

Decimal
141

Fig. 4 - LBPH Algorithm for Face Recognition [1]

4.6 YOLO

Yolo is an abbreviation for (you only look once). It is the most recent real-time object detection system that uses a single neural network to process the entire image [16]. This network splits the picture into sections and estimates the bounding boxes and probabilities for each [17]. The estimated probabilities are used to weigh these bounding boxes. The testing phase examines the whole picture, so the image's global meaning guides its predictions. Can detect objects form videos or image [18] [19]. Fig. 5 shows the Yolo process.

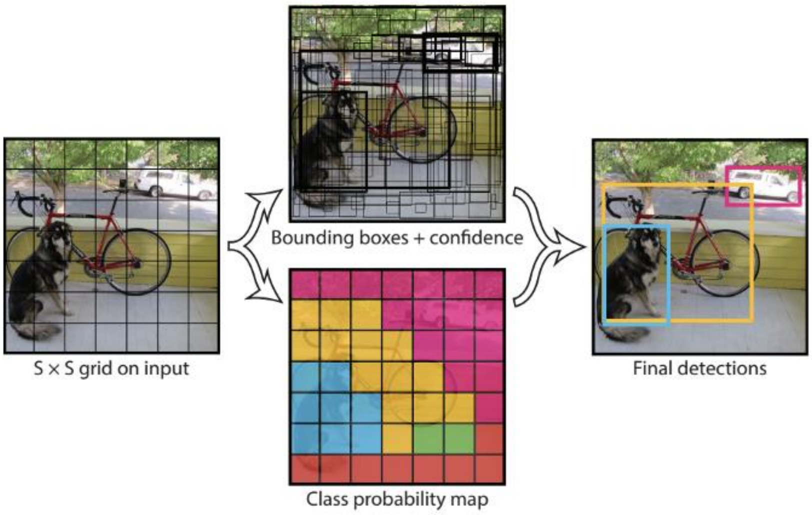


Fig. 5 - The Yolo process [18]

4.7 Faster R-CNN

It is a popular object detection architecture proposed in 2015 by Ross_Girshick and is one of the most well-known object detection architectures that employ convolution neural networks. The implementation of the Region Proposal Network enables (Faster R-CNN) easier and quicker (RPN). RPN is a completely convolutional network that has been trained side-to-side, and forecasts object boundaries and object ratings at each detection. Since RPN is so critical to (Faster-R-CNN), and remains one of the strongest entity detection frameworks open to researchers, most of this piece would concentrate on RPN architecture and the notions of anchor boxes and suppression non-maximum [18]. Fig. 6 shows the Faster_ R-CNN step.

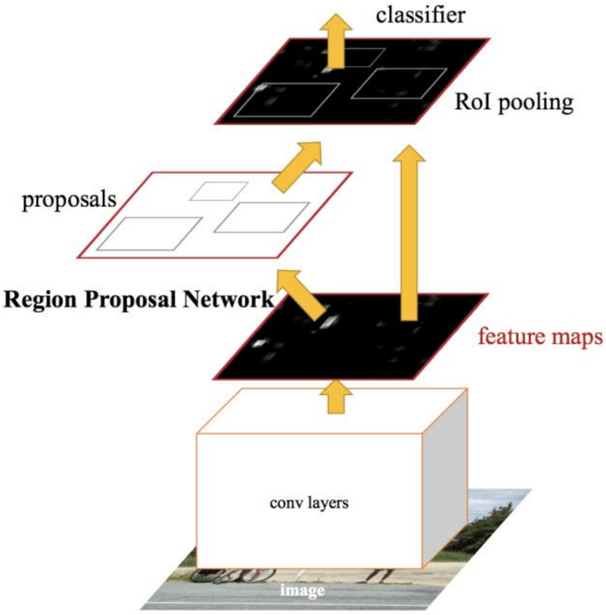


Fig. 6 - Faster_ R-CNN step [18]

4.8 Single Shot Detectors (SSDs)

The SSD method is focused on the feed-forward convolutional network that generates a permanent border-box array and results in the existence of class-based entity instances in these boxes and a non-maximum deletion stage to generate final detection. The early network layers are built on a common image classification design [16], [20]. Fig. 7 shows the SSD.

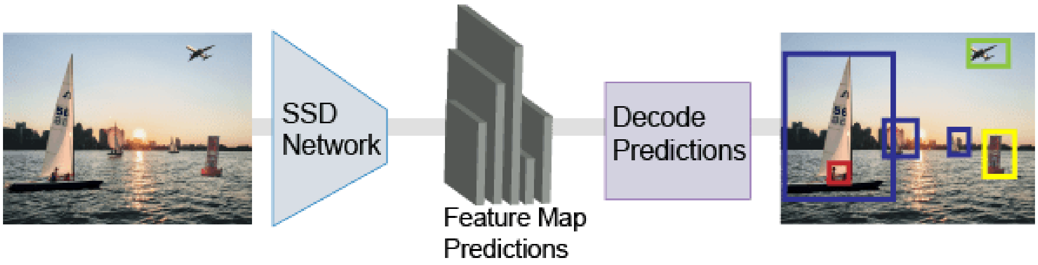


Fig. 7 - SSD [16]

5. Modules of OpenCV

5.1 Main Modules

Main modules are the main components of OpenCV and are included in the package versions by necessity. They are called core modules since they include essential functions, including image recognition, filtering, and transformation [8], [21].

5.2 Extra Modules

Extra modules are not included in the OpenCV release by default. Additional machine vision functionalities, such as text detection, are applied to these modules [21].

The main modules are described below:

- Core: Have most of OpenCV's core features.
- Imgproc: Picture processing tools such as transforms, manipulations, and filtering are used.
- Imgcodecs: Image reading/writing features are included.
- Videoio: Video reading/writing features are included.
- Highgui: Used for creating GUI to display output.
- Video: Motion detection and monitoring.
- calib3d: This package includes calibration and 3D reconstruction functions for estimating translation between multiple pictures.
- features2d: Item identification and categorization algorithms utilizing keypoint detection and descriptor extraction algorithms, which are included in this library.
- Objdetect: used for detecting object.
- Dnn: Classify and detecting objects, between other objects.
- Ml: Used for regression and classification, and it encompasses the vast majority of machine learning.
- Flann: Supports optimized algorithms for searching for high-dimensional attributes in massive data sets using nearest neighbor search. Fast Library for Approximate Nearest Neighbors (FLANN) is an acronym for Quick Library for Estimated Nearest Neighbors (FLANN)[22]. Photo: Removes noise and creates HD images, among other roles for photography-related computer vision.
- Stitching: Used for stitching image.
- Shape: Deal with issues such as shape transformation, pairing, and distance.
- Superres: Deal with enhancement and resolution algorithms.
- Videostab: Have algorithms for stabilizing video.
- Viz: create 3D display window for widgets.

6. OpenCV based on Python

Python was created by Guido van Rossum, who focused on writing clear, easy to understand code. With shorter code, the programmer will articulate the same ideas in fewer characters. Python is slower than some programming languages like C/C++. C/C++ can be expanded with Python effectively, but one other valuable aspect of Python is that it is generalizable and flexible. This function assists in the implementation of C++ codes that we can then transform into Python modules for ease of usage[3]. These two things offer great benefits: first, the code is as quick as C++ code (as it is), and second, Python is simple to program in. The Python bindings for OpenCV function like this. The Python implementation was built on top of the initial C++ code. Numpy's contribution helps here. Numpy is a highly structured library that often offers Python-level statistical capabilities. It does things in a MATLAB-like fashion. Conversion to Numpy arrays is done, and array processing is done on all the OpenCV structures. This is very good because whatever functions it can do in Numpy, it can now combine with OpenCV, making you more efficient. SciPy, which also supports NumPy, may be used along with this. So Python with OpenCV offers a good way to build out computer vision projects [11] quickly.

7. Assessment of Literature Reviews

OpenCV is a library for image and video processing that is used for image and video analysis, among other things. We review some recent literature concerning Face Detection and Recognition Using OpenCV.

7.1 Face Detection

Alcantara et al. [22] proposed a system for tracking and detecting the human head in real-time video using the OpenCV library. The proposed system will employ a Haar-like classifier to detect the head, Haar Training to train the system, and use CMT object tracking algorithm to track the head, where the result show 68% accuracy of CMT to tracking the person's head and 83% accuracy in detecting the person's head.

Gupta [7] proposes a method for detecting emotions in real-time and fixed pictures. So, before attempting emotion detection, they must first identify faces in static photographs or real-time videos using the Haar classifier of OpenCV. If the face has been identified, it may be cropped and analyzed to detect additional facial landmarks. The datasets are then conditioned using facial landmarks and categorized according to the eight emotions using the SVM, a machine learning algorithm. They obtained an accuracy of about 93.7 percent using SVM. These facial landmarks may be tweaked to improve precision.

Lee et al.'s [4] research aims to solve the classic issue of face detection in various lighting conditions and create an intelligent and effective human face detection system using Visual Studio 2015 software framework and OpenCV technologies. They demonstrated by experimentation that the photo processing approach applied in their paper completed facial recognition under various lighting conditions, which really is a significant advancement in face recognition technology.

Gupta et al. [16] proposed a system to improve universities' traditional attendance systems and avoid time wasted in counting traditional attendance based on image processing methods. The Student Attendance layout structure's primary function is to perform, incorporate, and manage attendance notes for a pupil, perform an automatic estimate on the amount of presented and absentees depending on the topic and affability of the class, and then produce an automated document or spreadsheet. They used OpenCV library, Haar-Cascade for face detection, and LBPH for face recognition; after that, individual student training occurred, and eventually, the device produced a spreadsheet that provided the number of students present in the classroom with a picture or video captured live.

Das et al. [19] offer a simpler approach to Face-Mask Detection by using several fundamental Machine Learning packages such as Scikit, OpenCV, TensorFlow, and Keras. The suggested procedure accurately senses the face in the picture and then determines whether or not it has a mask on it. It may track a face and a mask in motion as a surveillance mission artist. On two separate datasets, the system achieves a precision of up to 95.77 percent and 94.58 percent, respectively. They investigate optimized parameter values for the Serial CNN model to properly detect the existence of masks without triggering over-fitting.

Hoque et al. [5] designed software that can identify people's faces or faces from live video streaming. ATMega328p Arduino Uno-based Micro Controller with Pan-Tilt function and OpenCV are at the heart of the control scheme. The hair–Cascade, Camshift, Housedorff gap, AdaBoost, Viola Jones, etc. are all used to identify human faces. They used the Cascade algorithm Haar Classifier to classify faces.

Mehariya et al. [20] designed a system to overcome the problem of students not attending lectures in classrooms in universities or any place requiring workers' attendance. As non-attendance is a waste of the infrastructure, manual absence management is a tedious task and a waste of time, as the traditional absence counting system may be cheated. They used OpenCV to find the number of students in the class and built an algorithm giving the best occupancy ratio. The method suggested would be different from the conventional approach. The detect Multiscales procedure is used to count the number of students in a class, and the occupancy ratio is used to assign a complex classroom. The first way to classify the individual in a picture is the Multiscale detect method. It does not take items into consideration. This count is saved in the forecasting firebase, and the occupancy ratio is used for complex classroom distribution. This reduces the waste of room and allows us to prepare the schedule effectively. Factors such as classroom scale, projectors in classrooms, and dynamic allotment capability are taken into account. Google's cloud database (Firebase) is used to maintain and store college/university files. The personal records, rosters, schedules, and attendance sheets are included.

Sriratana et al. [11] created a Personal Identifier scheme by combining the Viola and Jones algorithms with the OpenCV library, and Python coded on the Raspberry Pi controller board. The framework successfully demonstrates all planned goal points, including cost-effective deployment, simple installation, and real-time detection. Just 8-9 percent of errors were discovered during tests of 150 samples, demonstrating the system's high precision and performance.

Patel et al. [23] suggested a method that senses whether or not the driver is asleep when operating a car or other big vehicle, and if the driver is suspected sleepy, the system will alert the driver to wake up and stop driving. One of the reasons for public road collisions is the driver's drowsiness when driving. It is necessary to develop an efficient approach to predict somnolence as soon as the driver becomes drowsy. This may help to reduce the large number of injuries that occur. This method would aid in the reduction of drivers' sleeping injuries. In the OpenCV setting, he used real-time image processing with a vision device and the technique of facial expression and eye blinking.

7.2 Face Recognition

Boyko et al. [21] compared the efficiency of two main libraries in computer vision (Dlib and OpenCV) and designed two basic face recognition systems for the two libraries. Based on his results, he demonstrates that the OpenCV library is more efficient and has improved face detection and recognition results than the Dlib library. It also implies that OpenCV is best suited for developing recognition software for the IoT framework.

Sarkar and Sikka [9] investigate and evaluates various classifiers used in facial embedding classification. They also focus on a Python-based face recognition pipeline that can be used to build a face recognition framework on compact low-power hardware devices. The technique mentioned employs pertained models and structures, resulting in cutting-edge efficiency without the need for strong hardware. On the LFW dataset, the suggested approach reaches an accuracy value of 99.4.

Sharma [18] suggested a facial recognition system for specialized applications, including entry and protection, purchases, and criminal identity. The identification method will be focused on facial recognition, which will be face detection, feature extraction and classification, and real-time face recognition. They used Haar-like for face detection and LBPH for face recognition, all this was done by using OpenCV in the Python environment. Kivy is used to build user interfaces to render the proposed system executable through several platforms.

James and Nettikadan [12] introduced a system for monitoring in real-time inside school buses. By utilizing image processing, using an image to identify a student by a camera. This machine tracks the videos inside the bus and recognizes the students and their gestures. The proposed system recognizes the students' faces and keeps a count of their numbers. If it is absolutely necessary, the system may even sound like a warning to draw the public's attention by using technologies from OpenCV and implementing them in Python. Face detection was performed using the Haar-Cascades classifier, and face recognition was performed using Eigenfaces and Lbph. This removes the majority of the disadvantages of manual attendance systems, such as the simple manipulation of attendance records.

Balachandran et al. [13] designed an efficient application for facial recognition using AI. For the neural network, the VGGFace framework was used. The application is divided into two phases: training and recognition. The training phase involves adding new faces to the scheme, while the identifying phase involves determining a face's identification. The program will operate on several cores with no problems.

Apoorva. et al. [15] proposes a technique for robust face recognition in a real-time setting. Haar-cascade is one of the facial recognition algorithms. On the OpenCV website, they tracked faces using Haar-like classifiers. Face detection has a good level of precision. Since the computation period is concise, the proposed method will successfully identify more than one face that is helpful for rapidly looking for suspects.

Srivastava et al. [17] created an Attendance System by combining facial recognition technologies with (OpenCV) algorithm. By simply maintaining a record of clock-in and clock-out times, this program can simplify the attendance automation process and help faculties to access students' details.

Soomro et al. [6] built a standalone authentication program using a facial recognition method with (NI VISION, LabVIEW, NI MyRIO, OpenCV). The developed framework should identify and enable authenticated individuals to bear in mind internal system weaknesses in a real-time context, especially those induced by a pause. The entire framework is made up of hardware and software, with parallel processing methods and modules like the NI MyRIO FPGA being included.

Sharma et al. [24] propose a system of recognition that a blind individual may benefit from. This paper applied a hand gesture identification method and facial recognition system to conduct many tasks. Dynamic images are from a dynamic video and are interpreted by algorithms. In the Hand Motion scheme, skin color recognition was carried out in YCbCr colors and a convex hand-based defect character point was used to identify various characteristics, such as fingertips and angles between fingers. Different activities such as spinning the fan or lights may be carried out according to the recognized gesture. OpenCV, Haar-Cascade, and LBPH are respectively used to detect the mask and to identify the face.

Salihbasic and Orehovacki [14] outline and describe the whole method of creating an Android app to recognize a person's gender, age, and face in depth. Face detection and recognition techniques and development software used in the development of an Android mobile application is defined and discussed. The software solution explains how to use the OpenCV library and displays the actual results of the smartphone app using photos.

Zhu and Cheng [25] present. An efficient Attitude Tracking Algorithm (EATA) based on OpenCV is used for facial recognition in the Intelligent door lock system. A specialized application has been created to allow the operation of devices as well as the display of the location. An automated door surveillance system using raspberry pi python, USB camera, and OpenCV is given for monitoring and security reasons. The system status database was created at a certain point in time. The gadget is very cheap, simple to set up, and simple to use. The Attitude Tracking Algorithm is extremely accurate and performs well.

Table 1 - Comparison table

Ref.	Aim	classifier	accuracy	Result/purpose
[21] 2018	SVM	OpenCV more accurate than dlib	83% head detect	The OpenCV library is more productive, has improved facial recognition and detection accuracy
[22] 2018	Head Detection and Tracking	Haar-like CMT Cascade	68% tracking	The proposed system successfully detected the head of a human using OpenCV libraries, specifically using Haar-like attribute detection.
[9] 2018	comparative of classifying the face using different classifiers	MLP Extra Tree (Random Forest) KNN RadialSVM GaussianNB LinearSVM (Logistic Regression)	99.1% 86.4% 95.9% 99.3% 98.4% 98.8% 99.2% 99.4%	The results show that Logistic Regression outperforms the other algorithms for face classification in terms of speed and accuracy.
[7] 2018	real-time recognition Facial emotion	Haar SVM	93.7%	The findings suggest that with today's computing power, user-independent, completely automated real-time coding of facial expressions in a continuous video stream is a goal that can be achieved.
[4] 2018	Face Detection under Different Lighting	Haar	80%	The experiment demonstrated that the picture processing system has facial recognition in various lighting conditions.
[18] 2019	Designing of Face Recognition System	Haar-like LBPH	80%	The system is tested by more than 150 people and has a reliability of approximately 80%. It is measured with multiple cameras in various settings, and lighting conditions, and the findings are about the same. A Logitech C90 USB webcam is used here.
[12] 2019	Student Monitoring System for School Bus	Haar-Cascades LBPH Eigenfaces	85%	The system watches the bus and detects the students and their movements, acknowledges the faces of the students, and their count is also tracked and alerts the audience if necessary.
[13] 2019	Face Recognition in Parallel Computer	Keras VGG-Face	module performance depends on the number of processors	The results collected were rather similar to what was predicted. The efficiency gained by simply running the program on a machine with more computing capacity and cores than a simple laptop was important.
[14] 2019	Recognition of the gender, age, and face of the person	cascade LBP LBPH	successfully work but affected by mobile type, face coverage, expressions of face,	Gender, face, and age recognition was achieved

			person pose	
[15] 2019	criminal identification by face recognition automated	Haar-like	80%	Since the computation period is very short, the proposed method will successfully identify more than one face, which is helpful for rapidly looking for suspects.
[16] 2020	Automated Attendance System	KNN HaarCascade LBPH	97%	Prevent students from marking fake attendance for other students. It would also save faculty resources in universities by eliminating the need for them to take attendance of students who are present in class, and it will be able to send monthly attendance reports to students' parents by email.
[17] 2020	Attendance System in real-time	HaarCascade Paul – Viola.	95%	The results explicitly demonstrate that as the facial angle increases, face identification and recognition rate decreases.
[6] 2020	real-time system of electronic voting authorized by face recognition	Haar-cascade camshift	90%	The primary goal of this project is to completely concentrate on the electronic voting mechanism and the protection of every organization.
[19] 2020	Face Mask Detection to protect from Covid-19	Cascade CNN	between 95.77%, 94.58%	Since wearing a mask which becomes mandatory before the Covid-19 crisis is resolved, the implemented model can make a significant contribution to the public health care system.
[5] 2020	Autonomous Face Detection System from Real-time Video Streaming	Haar Cascade	83%	Recognize human faces with some kind of camera and issue an alarm with a buzzer and an automatic-on light bulb that makes it noticeable from a long distance.
[20] 2020	Counting students in classroom allocation	Cascade	90 to 100%.	By installing cameras in the hallway, the model may be used to detect students who are skipping classes.
[24] 2019	A Face Recognition and Static Hand Gesture System for the Blind	Haar Cascade LBPH	hand gesture recognized is 95.2% Facial recognition is 92%.	The developed system will function as a virtual assistant for a blind individual using hand gestures and face recognition.
[11] 2018	Personal Identifier application	Cascade Haar	90% Just 8-9 percent of errors were discovered after analyzing 150 samples.	The proposed application helps companies calculate work attendance and detect cases of fraud in work attendance compared to old work attendance monitoring methods.
[25] 2020	door lock intelligence based on face recognition	Attitude Tracking Algorithm (ATA)	95%	According to the testing findings, the suggested system is more efficient, uses less power, and is more cost-effective.
[23] 2018	Drivers' Somnolence Detection method in Real Time	EEG ANN	90%	This method would aid in the reduction of drivers' sleeping injuries. In the OpenCV setting, he used real-time image processing with a vision device and the technique of facial expression and eye blinking.

Based on Table 1, we compared the accuracy of techniques and classifiers of OpenCV used in a number of the studies reviewed of face detection and face recognition in different areas of computer vision. From our observations of the 20 reviewed studies (publications) (2018–2020) discussing the used methods of face detection and recognition in the field of computer vision using OpenCV. It was found that OpenCV can be used in different fields like face detection, recognition the faces, and Recognize facial expressions [26], [27]. Also, it was found that we can use OpenCV in the security field like criminal identification [28]. Additionally, others used OpenCV to automate the attendance of students or workers in institutions. At the same time, other researchers used OpenCV to reducing traffic accidents on the roads.

8. Conclusion

Computer Vision is the subfield of Artificial Intelligence, where computers are trained to process the image and extract the important features from the images or videos. Open Computer Vision (OpenCV), a python library written in C++, provides various functionalities for computer vision applications. Applications of computer vision are object detection, face recognition, medical diagnosis, etc. In this paper, we emphasize the important role of OpenCV in face detection and face recognition. We illustrate the popular algorithms in OpenCV that are used for face detection and face recognition. Then state the OpenCV modules and explain OpenCV based on Python and mention the applications for OpenCV. Finally, we assessment and compared recent literature reviews that use OpenCV to detect and recognize the human face in a variety of fields in order to improve human life.

Acknowledgement

The authors would like to thank the IT Department, Technical College of Informatics Akre, Duhok Polytechnic University, Duhok, Kurdistan Region, Iraq and College of Engineering, Nawroz University, Duhok, Kurdistan Region, Iraq.

References

[1] Jagtap, A. M., Kangale, V., Unune, K., & Gosavi, P. (2019, February). A Study of LBPH, Eigenface, Fisherface and Haar-like features for Face recognition using OpenCV. In *2019 International Conference on Intelligent Sustainable Systems (ICISS)* (pp. 219-224). IEEE

[2] Sigut, J., Castro, M., Arnay, R., & Sigut, M. (2020). OpenCV basics: a mobile application to support the teaching of computer vision concepts. *IEEE Transactions on Education*, *63*(4), 328-335

[3] Adusumalli, H., Kalyani, D., Sri, R. K., Pratapteja, M., & Rao, P. P. (2021, February). Face Mask Detection Using OpenCV. In *2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV)* (pp. 1304-1309). IEEE

[4] Mostafa, S. A., Mustapha, A., Gunasekaran, S. S., Ahmad, M. S., Mohammed, M. A., Parwekar, P., & Kadry, S. (2021). An agent architecture for autonomous UAV flight control in object classification and recognition missions. *Soft Computing*, 1-14

[5] Hoque, M. A., Islam, T., Ahmed, T., & Amin, A. (2020, March). Autonomous face detection system from real-time video streaming for ensuring the intelligence security system. In *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)* (pp. 261-265). IEEE

[6] Soomro, Z. A., Memon, T. D., Naz, F., & Ali, A. (2020, January). FPGA Based Real-Time Face Authorization System for Electronic Voting System. In *2020 3rd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)* (pp. 1-6). IEEE

[7] Gupta, S. (2018, January). Facial emotion recognition in real-time and static images. In *2018 2nd international conference on inventive systems and control (ICISC)* (pp. 553-560). IEEE

[8] Dino, H., Abdulrazzaq, M. B., Zeebaree, S. R., Sallow, A. B., Zebari, R. R., Shukur, H. M., & Haji, L. M. (2020). Facial expression recognition based on hybrid feature extraction techniques with different classifiers. *TEST Engineering & Management*, *83*, 22319-22329

[9] Kashinath, S. A., Mostafa, S. A., Mustapha, A., Mahdin, H., Lim, D., Mahmoud, M. A., ... & Yang, T. J. (2021). Review of Data Fusion Methods for Real-Time and Multi-Sensor Traffic Flow Analysis. *IEEE Access*

[10] Abdulrazaq, M. B., Mahmood, M. R., Zeebaree, S. R., Abdulwahab, M. H., Zebari, R. R., & Sallow, A. B. (2021, February). An analytical appraisal for supervised classifiers' performance on facial expression recognition based on relief-F feature selection. In *Journal of Physics: Conference Series* (Vol. 1804, No. 1, p. 012055). IOP Publishing

[11] Sriratana, W., Mukma, S., Tammarugwattana, N., & Sirisantisamrid, K. (2018, July). Application of the OpenCV-Python for Personal Identifier Statement. In *2018 International Conference on Engineering, Applied Sciences, and Technology (ICEAST)* (pp. 1-4). IEEE

[12] James, C., & Nettikadan, D. (2019, April). Student monitoring system for school bus using facial recognition. In *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)* (pp. 659-663). IEEE

[13] Balachandran, B., Saad, K. F., Patel, K., & Mekhiel, N. (2019, December). Parallel Computer for Face Recognition Using Artificial Intelligence. In *2019 14th International Conference on Computer Engineering and Systems (ICCES)* (pp. 158-162). IEEE

[14] Salihbašić, A., & Orehovački, T. (2019, May). Development of android application for gender, age and face recognition using opencv. In *2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)* (pp. 1635-1640). IEEE

[15] Apoorva, P., Impana, H. C., Siri, S. L., Varshitha, M. R., & Ramesh, B. (2019, March). Automated criminal identification by face recognition using open computer vision classifiers. In *2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)* (pp. 775-778). IEEE

[16] Gupta, N., Sharma, P., Deep, V., & Shukla, V. K. (2020, June). Automated attendance system using OpenCV. In *2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO)* (pp. 1226-1230). IEEE

[17] Srivastava, M., Kumar, A., Dixit, A., & Kumar, A. (2020, February). Real time attendance system using face recognition technique. In *2020 International Conference on Power Electronics & IoT Applications in Renewable Energy and its Control (PARC)* (pp. 370-373). IEEE

[18] Sharma, V. K. (2019, May). Designing of Face Recognition System. In *2019 International Conference on Intelligent Computing and Control Systems (ICCS)* (pp. 459-461). IEEE

[19] Das, A., Ansari, M. W., & Basak, R. (2020, December). Covid-19 Face Mask Detection Using TensorFlow, Keras and OpenCV. In *2020 IEEE 17th India Council International Conference (INDICON)* (pp. 1-5). IEEE

[20] Mehariya, J., Gupta, C., Pai, N., Koul, S., & Gadakh, P. (2020, July). Counting Students using OpenCV and Integration with Firebase for Classroom Allocation. In *2020 International Conference on Electronics and Sustainable Communication Systems (ICESC)* (pp. 624-629). IEEE

[21] Boyko, N., Basystiuk, O., & Shakhovska, N. (2018, August). Performance evaluation and comparison of software for face recognition, based on dlib and opencv library. In *2018 IEEE Second International Conference on Data Stream Mining & Processing (DSMP)* (pp. 478-482). IEEE

[22] Alcantara, G. K. L., Evangelista, I. D. J., Malinao, J. V. B., Ong, O. B., Rivera, R. S. D., & Ambata, E. L. U. (2018). Head detection and tracking using OpenCV. In *2018 IEEE 10th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM)* (pp. 1-5). IEEE

[23] Patel, R., Patel, M., & Patel, J. (2018, April). Real Time Somnolence Detection System In OpenCV Environment for Drivers. In *2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)* (pp. 407-410). IEEE

[24] Sharma, S., & Jain, S. (2019, March). A static hand gesture and face recognition system for blind people. In *2019 6th International Conference on Signal Processing and Integrated Networks (SPIN)* (pp. 534-539). IEEE.

[25] Zhu, Z., & Cheng, Y. (2020). Application of attitude tracking algorithm for face recognition based on OpenCV in the intelligent door lock. *Computer Communications*, *154*, 390-397

[26] Chandan, G., Jain, A., & Jain, H. (2018, July). Real time object detection and tracking using Deep Learning and OpenCV. In 2018 international conference on inventive research in computing applications (ICIRCA) (pp. 1305-1308). IEEE

[27] Khan, M., Chakraborty, S., Astya, R., & Khepra, S. (2019, October). Face Detection and Recognition Using OpenCV. In 2019 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS) (pp. 116-119). IEEE

[28] Palekar, R. R., Parab, S. U., Parikh, D. P., & Kamble, V. N. (2017, April). Real time license plate detection using openCV and tesseract. In 2017 international conference on communication and signal processing (ICCSP) (pp. 2111-2115). IEEE