Face Detection and Recognition using OpenCV

Ajay Kumar
Faculty of IT Department
Meerut Institute of Engineering and Technology, Meerut

Shivansh Chaudhary
IT Department,
Meerut Institute of Engineering and Technology, Meerut

Sonik Sangal
IT Department,
Meerut Institute of Engineering and Technology, Meerut

Raj Dhama
IT Department,
Meerut Institute of Engineering and Technology, Meerut

ABSTRACT

OpenCV from Intel is a free and open-source image and video processing library. It is related to computer vision in terms of feature and object recognition, as well as machine learning. This paper introduces the main OpenCV modules, features, and Python-based OpenCV. The paper also discusses common OpenCV applications and classifiers that are used in these applications, such as image processing, face detection, face recognition, and object detection. Finally, we discuss some literary reviews of OpenCV applications in computer vision fields such as face detection and recognition, recognition of facial expressions such as sadness, anger, and happiness, and recognition of a person's gender and age.

Keywords

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

1. INTRODUCTION

One of the most exciting and difficult issues in the field of artificial intelligence is computer vision. Computer Vision connects computer software to the pictures we perceive all around us. It allows software to understand and learn about the sights in its surroundings. As an example, the color, shape, and size of the fruit define its value. For the human brain, this task may appear straightforward, but in the Computer Vision pipeline, we take data, process it, and then train and educate the model to learn how to distinguish various fruits based on size, shape, and colour.

The major objective is to recognise and interpret the images, as well as to provide new images that are more beneficial to us in various aspects of life. The term "open source computer vision" is abbreviated as "OpenCV." The architecture is made up of pre-programmed software, databases, and plugins that facilitate the integration of computer vision applications [3]. With a big developer community, it is one of the most widely used toolkits. It is well-known for the scale at which it creates real-world industrial use cases. OpenCV is based on the C/C++, Python, and Java programming languages and may be used to create computer vision applications for Windows, Linux, macOS, Android, and iOS.OpenCV-4.5.2 and OpenCV-3.4.14 are the most current versions. It's free and open-source, and it's easy to use and set up. It's designed to boost numerical productivity, with a focus on real-time applications. The initial version was written in the C programming language; but, with the introduction of Version 2.0, which included a C++ implementation [2], it became more popular. To design new features, C++ is employed. OpenCV is available for free download from http://opencv.org. The most current distribution update (version: 4.5.2) as well as prior revisions are available on this platform. To be shown or stored using OpenCV, photos must be in BGR or Grayscale format. Otherwise, undesirable results may ensue.Face detection is a kind of computer vision that helps recognise and visualize face characteristics in still images and real-time recordings. In digital images and videos, this sort of object detection algorithm recognizes occurrences of semantic artifacts of a specified class (such as people, automobiles, and homes). Face recognition has grown more significant as technology has progressed, particularly in industries like photography, military, and marketing.Recognition is a new topic of study that has grabbed the interest of academics since it is easy to utilize thanks to OpenCV-based Python. Face recognition systems, such as Facebook's automated tag recommendation on photographs, have a number of uses in public safety, entertainment, man-machine communication, and social networking. It's also been observed in attendance control, financial offices, voter registration, and other sectors of educational and non-educational institutions. In this article, we discuss the importance of OpenCV in face detection and recognition, the algorithms that may be used in OpenCV for face detection and recognition, the OpenCV modules, and how to utilize OpenCV with Python, as well as the OpenCV applications. Finally, in order to enhance human life, we evaluated and analyzed current literature studies that utilize OpenCV to detect and recognise the human face in a number of domains.The following is how the rest of the paper is laid out: Face detection is discussed in section two. Face recognition is discussed in Section 3. The OpenCV library and algorithm are discussed in Section 4. OpenCV modules are discussed in section 5. OpenCV, which is written in Python, is discussed in Section 6. The evaluation of literature reviews as well as the comparison table are described in section 7. The paper comes to an end in Section 8.

2. FACE DETECTION

Because of its applicability in computer and human interaction, face detection has gotten a lot of press in recent years. Image processing includes face detection as a subset. Image processing is a method of compressing, enhancing, or extracting useful information from pictures. Face recognition software can recognise a single or many faces in a photograph, eliminating distracting background noise. A face recognition algorithm must divide images into two categories based on whether or not they include a face. The purpose of the face detection method is to completely study the image, identify the presence of faces in it, and eliminate the background. There are two sorts of face detection errors: false negative and false positive. When a face is recognised in a photograph that does not include any faces, this is known as a false positive. When the algorithm rejects the existence of anything in the image, it is called a false negative. The detection rate is the ratio of the number of faces correctly recognised by the system to the number of faces identified by humans. The face detection algorithm's detection rate should be as high as feasible

3. FACE RECOGNITION

Facial recognition is the most advanced and fast biometric technology in the world. It makes use of the most visible part of the human body, the face, in a non-intrusive way. Most people are unaware of the facial recognition process that is taking place on them, according to global data, making it one of the least invasive and time-consuming operations. In the input image, the facial recognition algorithm analyses the various aspects of a face. This biometric has been widely hailed as a wonderful way for detecting potential threats such as terrorists, scam artists, and other criminals, but it has yet to acquire universal acceptance in high-level application.In the not-too-distant future, biometric facial recognition technology is predicted to overtake fingerprint biometrics as the most frequent means of user identification and verification.

4. OpenCV LIBRARY

It's a huge open-source library for image processing, machine learning, and computer vision. Python, C++, and Java are just a few of the programming languages that OpenCV supports. It can distinguish objects, people, and even human handwriting in photos and videos. When used in conjunction with other libraries, such as Numpy, a high-performance library for turning machines, OpenCV achieves good results; that is, any services that Numpy can do may also be performed with OpenCV. It is developed in C++ and has a C++ interface as its main interface, but it also includes an older Language training that is less robust but nonetheless thorough. In the C++ GUI, you can see the most up-to-date technologies and techniques. Bindings for Python, Java, and MATLAB/OCTAVE are available.To foster greater acceptance, wrappers in a number of programming languages have been built. In version 3.4, JavaScript plugins for a subset of OpenCV functions are released as OpenCV.js, which may be utilized on web platforms. The OpenCV project was born out of Intel's research initiative to support CPU-intensive applications [11], which was first announced in 1999. Face detection and recognition methods are commonly implemented using OpenCV. The OpenCV algorithms listed below are some of the most often used.

4.1 Haar Cascade

The Haar Cascade method of object detection is quite successful. It's a machine-learning-based system that uses a huge number of positive and negative pictures to learn a series of behaviors. It grows accustomed to perceiving things through diverse lenses [12]. The Haar cascade classifier is shown in Figure 1. The Haar cascade flowchart is shown in Figure 2.

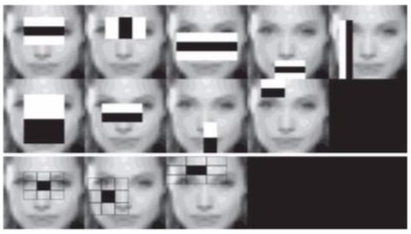

Fig. 1 - View of Haar cascade classifier

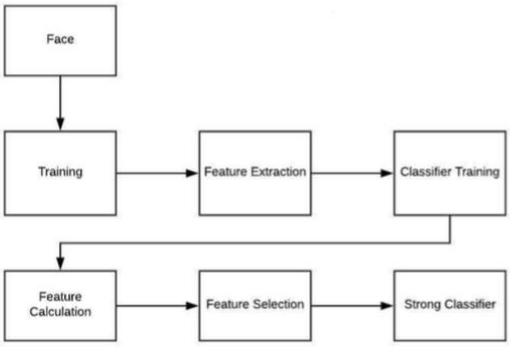

Fig. 2 - Haar Cascade flowchart [1]

4.2 LBP (Local Binary Pattern)

It's a basic but effective texture operator that identifies pixels in an image by thresholding their surroundings and interpreting the result as a binary integer. Because of its discriminative power and computational simplicity, the LBP texture operator has become a standard method in a range of applications. It may be seen as a unifying answer to the historically disparate statistical and structural models of texture analysis. In real-world applications, the LBP operator's resistance to monotonic grayscale changes generated, for example, by light fluctuations is likely its most important quality. Another distinguishing aspect is its computational simplicity, which enables it to interpret pictures in challenging real-time circumstances [13]. Figure 3 depicts the use of local binary patterns to describe face emotions.

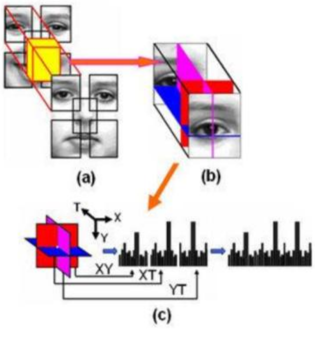

Fig. 3 - Description of facial expressions with local binary patterns [13]

4.3 Eigen Faces

It's a method for distributing face photos into existing facial spaces that uses PCA (Principal Component Analysis) to reduce dimensionality and select the strongest vectors. PCA's main purpose is to find the optimal vectors for explaining the distribution of facial pictures from picture space to face space. The primary component quantity is calculated using the m eigenvector, which is based on the eigenvalue distribution. The qualifying facial image's covariance matrix is used to determine the eigenvector and eigenvalue. The eigenvectors are sorted by eigenvalue (from highest to lowest), and M initial eigenvectors are chosen to create the primary variable

4.4 Fisher Faces

It is a face recognition system that has been shown by multiple researchers to reliably recognise faces. FisherFace is a calculation model that combines the PCA (Principal Component Analysis) and Fisher's Linear Discriminant Analysis calculation models (FLD). To simplify and accelerate the FLD procedure, PCA is utilized to decrease input data. FLD, on the other hand, is used to create a distribution matrix that may be utilized to help with classification and identification. The PCA and FLD computation models are used to create a sequence of FisherFaces. There are four essential phases in the face recognition process: Face recognition, PCA estimation, computation, and classification are all tasks that must be completed.

4.5 LBPH (Local Binary Pattern Histogram)

LBP is a very powerful texture operator. The threshold value of each nearby pixel is compared to the value of the center pixel. In the context of binary numbers, it considers the outcomes. Because of its discriminative strength and simplicity, LBP is a widely used approach in a range of applications. In 1994, LBP was discovered for the first time. It appears to have improved since then, becoming a more efficient texture categorization system. On the same dataset, it was eventually revealed that combining LBP with histograms of directed gradient descriptors improves its accuracy. LBP also includes features like monotonic gray-scale improvements and statistical simplicity, allowing it to analyze pictures in real-time applications.

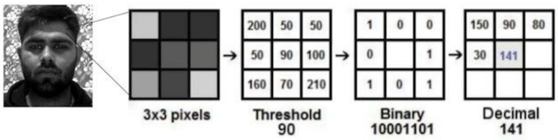

Fig. 4 - LBPH Algorithm for Face Recognition [1]

4.6 YOLO

It's a popular object identification architecture introduced by Ross Girshick in 2015, and it's one of the most well-known convolution neural network-based object recognition designs. The Region Proposal Network's implementation makes (Faster R-CNN) easier and faster (RPN). RPN is a fully convolutional network that has been trained side-by-side to predict object borders and ratings at each detection. Because RPN is so important to (Faster-R-CNN) and is still one of the most powerful entity detection frameworks available to academics, the majority of this article will focus on RPN architecture and concepts like anchor boxes and suppression non-maximum [18]. The Faster_ R-CNN step is depicted in Figure 6.

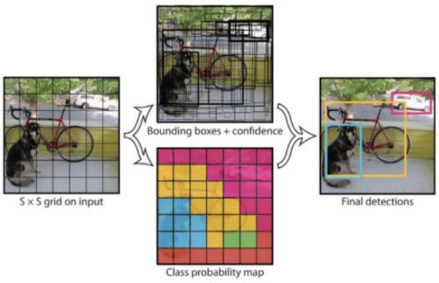

Fig. 5 - The Yolo process [18]

4.7 Faster R-CNN

It's a popular object identification architecture introduced by Ross Girshick in 2015, and it's one of the most well-known convolution neural network-based object recognition designs. The Region Proposal Network's implementation makes (Faster R-CNN) easier and faster (RPN). RPN is a fully convolutional network that has been trained side-by-side to predict object borders and ratings at each detection. Because RPN is so important to (Faster-R-CNN) and is still one of the most powerful entity detection frameworks available to academics, the majority of this article will focus on RPN architecture and concepts like anchor boxes and suppression non-maximum [18]. The Faster_ R-CNN step is depicted in Figure 6.

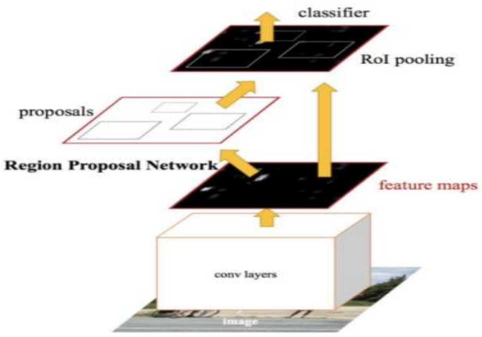

Fig. 6 - Faster_ R-CNN step [18]

4.8 Single Shot Detectors (SSDs)

The SSD technique focuses on a feed-forward convolutional network that creates a permanent border-box array and results in the presence of class-based entity instances in these boxes, as well as a non-maximum deletion step for final detection. The initial network layers are based on a widely used picture categorization scheme [16], [20]. The SSD is seen in Figure 7.

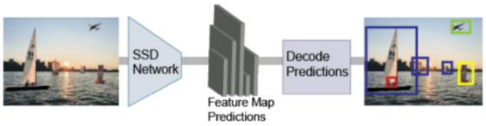

Fig. 7 - SSD [16]

5. MODULES OF OPENCV

5.1 Main Modules

The major modules are the most important parts of OpenCV, and they must be included in all package versions. They're referred to as core modules since they encompass functions like picture recognition, filtering, and transformation.

5.2 Extra Modules

By default, extra modules are not included in the OpenCV distribution. These modules are given additional machine vision capabilities, such as text detection [21].

The following are the primary modules:

1. Contains the majority of OpenCV's main functionality.
2. Imgproc: Tools for image processing are utilized, such as transformations, manipulations, and filters.
3. Imgcodecs: Features for reading and writing images are available.
4. Videoio has video reading/writing features.
5. Highgui: Creates a graphical user interface (GUI) for displaying output.
6. Motion detection and monitoring are demonstrated in this video.
7. calib3d: For determining translation between several images, this package offers calibration and 3D reconstruction methods.
8. features2d: This library includes item identification and classification methods that use keypoint detection and descriptor extraction algorithms.
9. Objdetect is a function that detects objects.
10. Dnn: Classify and detect items in the context of other objects.
11. Ml: Ml is a machine learning language that is used for regression and classification. It contains the overwhelming bulk of machine learning.
12. Flann: Supports efficient techniques for scanning big data sets for high-dimensional qualities utilizing closest neighbor search. FLANN (Quick Library for Estimated Nearest Neighbors)[22] is an abbreviation for Fast Library for Approximate Nearest Neighbors (FLANN). Photo: For photography-related computer vision, it removes noise and provides HD photos, among other things.
13. Stitching is a technique for putting together a picture.
14. Shape: Take care of difficulties like shape morphing, pairing, and distance.
15. Deal with improvement and resolution algorithms in Superres.
16. Videostab: Algorithms for video stabilization.

6. OPENCV BASED ON PYTHON

Guido van Rossum, who concentrated on producing simple, easy-to-understand code, invented Python. The programmer will be able to express the same thoughts in fewer characters if the code is shorter. Python is slower than C/C++ and other programming languages. Python can successfully expand C/C++, but another advantage of Python is that it is generalizable and adaptable. This function aids in the creation of C++ code that may then be transformed.For simplicity of use, we've turned them into Python modules[3]. These two factors combine to provide significant advantages: first, the code is as fast as C++ code (which it is), and second, Python is easy to write in. This is how the OpenCV Python bindings work. The Python implementation was developed on top of the C++ code that was originally written. Numpy's contribution is useful in this case. Numpy is a well-structured package that frequently includes statistical features at the Python level. It works in a similar way as MATLAB. All OpenCV structures are converted to Numpy arrays, and array processing is performed on them. This is beneficial since any Numpy routines may now be used with OpenCV, making you more efficient. This may be used with SciPy, which also supports NumPy.

7. ASSESSMENT OF LITERATURE REVIEWS

OpenCV is an image and video processing toolkit that may be used for a variety of tasks, including image and video analysis. Face Detection and Recognition Using OpenCV is the subject of a recent review.

7.1 Face Detection

Using the OpenCV library, Alcantara et al. [22] suggested a method for tracking and identifying the human head in real-time video. The suggested system would utilize a Haar-like classifier to identify the head, Haar Training to train the system, and the CMT object tracking algorithm to track the head, with CMT's tracking accuracy of 68 percent and detection accuracy of 83 percent, respectively. Gupta [7] suggests a method for identifying emotions in both real-time and static images. So, before trying emotion recognition, they must first use OpenCV's Haar classifier to recognise faces in static images or real-time movies. After the face has been recognised, it may be cropped and processed to find other facial landmarks. The datasets are then conditioned using face landmarks and classified using the SVM, a machine learning technique, according to the eight emotions. Using SVM, they were able to achieve an accuracy of 93.7 percent. To increase accuracy, these face landmarks can be adjusted.

Using the Visual Studio 2015 software platform and OpenCV technologies, Lee et al[4] .'s study intends to tackle the classic challenge of face detection in varied lighting circumstances and construct an intelligent and effective human face detection system. They proved that the picture processing technique used in their study accomplished facial recognition across a variety of lighting situations through experimentation, which is a significant development in face recognition technology.

Gupta et al. [16] suggested a way to enhance traditional university attendance systems and eliminate time spent calculating traditional attendance using image processing approaches. The primary function of the Student Attendance layout structure is to perform, incorporate, and manage attendance notes for a student, calculate an automatic estimate of the number of present and absentees based on the topic and affability of the class, and then generate an automated document or spreadsheet. They employed the OpenCV library, Haar-Cascade for face detection, and LBPH for face recognition; following that, individual student training took place, and the device finally produced a spreadsheet with the number of pupils present in the classroom together with an image or video shot in real time.

Das et al. [19] use various key Machine Learning tools such as Scikit, OpenCV, TensorFlow, and Keras to present a simplified solution to Face-Mask Detection. The recommended approach detects the face in the image and assesses whether or not it is wearing a mask. As a surveillance mission artist, it may monitor a face and a mask in motion. The system achieves precision of up to 95.77 percent and 94.58 percent on two different datasets. They look at the best parameter values for the Serial CNN model in order to identify masks correctly without causing over-fitting.

Hoque et al. [5] created software that can recognise people's faces in real-time video streams. ATMega328p At the center of the control method is an Arduino Uno-based Micro Controller with Pan-Tilt capability and OpenCV. Hair is used to identify human faces in a variety of ways, including Cascade, Camshift, Housedorff gap, AdaBoost, Viola Jones, and others. To categorize faces, they employed the Haar Classifier Cascade method.Mehariya et al. [20] devised a solution to address the issue of students failing to attend lectures in university classrooms or any other place where workers are required to be present. Manual absence management is a tiresome process and a waste of time, as non-attendance wastes infrastructure and the typical absence counting method may be tricked. They utilized OpenCV to figure out how many students were in the class and then created an algorithm that gave them the optimum occupancy ratio. The method proposed would be different from what has been done previously. The occupancy ratio is used to allocate a complicated classroom, and the detect Multi Scales process is used to count the number of pupils in a class.This saves space and helps us to plan our calendar more efficiently. Classroom size, projectors in classes, and dynamic allotment capabilities are all taken into consideration. College and university files are maintained and stored using Google's cloud database (Firebase).Personal data, rosters, timetables, and attendance sheets are all part of the package.

Sriratana et al. [11] combined the Viola and Jones algorithms with the OpenCV library and Python programmed on the Raspberry Pi controller board to produce a Personal Identifier scheme. All of this is successfully shown by the framework.

Cost-effective deployment, easy installation, and real-time detection are all anticipated goals. During evaluations of 150 samples, just 8-9 percent of mistakes were identified, proving the system's great accuracy and performance.

Patel et al. [23] proposed a technique for detecting if a driver is sleeping when driving a car or other large vehicle, and if the driver is suspected of being drowsy, the system will alert the driver to wake up and cease driving. The driver's tiredness while driving is one of the causes of public road collisions. It's critical to devise a method for predicting somnolence as soon as the driver feels tired. This might assist to cut down on the high number of injuries that occur.

This strategy would help to reduce the number of sleep-related injuries among drivers. He employed real-time image processing with a vision device and the technique of facial expression and eye blinking in the OpenCV context.

7.2 Face Recognition

Boyko et al. [21] examined the effectiveness of two major computer vision libraries (Dlib and OpenCV) and created two simple face recognition algorithms for each. He shows that the OpenCV library is more efficient and has better face detection and identification results than the Dlib library based on his findings. It also means that OpenCV is the ideal choice for creating IoT recognition software.

Sarkar and Sikka [9] study and analyze several face embedding classification classifiers. They also concentrate on a Python-based face recognition pipeline that may be used to create a face recognition framework on small, low-power hardware.

The indicated approach makes use of specific models and structures, resulting in cutting-edge efficiency without the use of powerful hardware. The recommended technique achieves a 99.4% accuracy on the LFW dataset.

Sharma [18] proposed a face recognition system for specific purposes such as access and security, purchases, and criminal identities. Facial recognition will be the emphasis of the identification approach, which will include face detection, feature extraction and categorization, and real-time face recognition. They employed Haar-like for face detection and LBPH for face recognition, all while working in the Python environment using OpenCV. Kivy is used to create user interfaces that allow the proposed system to run on several platforms.

A technique for real-time surveillance inside school buses was introduced by James and Nettikadan [12]. Using image processing, a camera can identify a student based on a picture. This gadget watches the videos on the inside of the bus and recognises the pupils and their movements. The suggested system maintains track of the pupils' numbers and recognises their faces. If absolutely necessary, the system might even sound like a warning to grab the public's attention by implementing OpenCV technologies in Python. The Haar- Cascades classifier was used to detect faces, while Eigenfaces and Lbph were used to recognise them. This eliminates the bulk of the problems of manual attendance systems, such as simple attendance record manipulation.

Balachandran et al. [13] used AI to create a useful application for facial recognition. The VGGFace framework was used to create the neural network. There are two parts to the application: training and recognition. The training step entails adding new faces to the scheme, and the identifying phase entails establishing the identity of a face. The software will run without a hitch on several cores.

Apoorva. et al. [15] offer a method for robust real-time facial recognition. One of the facial recognition techniques is Haar-cascade. They used Haar-like classifiers to track faces on the OpenCV website. Face recognition has a high level of accuracy. Because the calculation time is short, the suggested technique would correctly recognise many faces, which is useful for quickly searching for suspects.

By merging face recognition technology with the (OpenCV) method, Srivastava et al. [17] developed an Attendance System. This tool can simplify the attendance automation process and enable faculty access students' information by simply keeping track of clock-in and clock-out timings.

Soomro et al. [6] developed a self-contained authentication system.

through the use of a face recognition technique (NI VISION, LabVIEW, NI MyRIO, OpenCV). Internal system flaws, particularly those caused by a pause, should be identified and authenticated personnel should be able to remember them in real-time using the framework provided. Parallel processing techniques and modules such as the NI MyRIO FPGA are incorporated in the framework, which is made up of both hardware and software.Sharma et al. [24] offer a recognition system that might be useful to a blind person. Many tasks were carried out in this article using a hand gesture identification approach and a facial recognition system. Algorithms evaluate dynamic visuals derived from a dynamic video. Skin colour identification in YCbCr colors was employed in the Hand Motion scheme, and a convex hand-based defect character point was used to recognise various attributes such fingertips and angles between fingers. Various actions, such as rotating the fan or turning on the lights, can be performed in response to the identified gesture. To detect the mask and identify the face, OpenCV, Haar-Cascade, and LBPH are utilized, respectively. Salihbasic and Orehovacki [14] detail the entire process of developing an Android software that can distinguish a person's gender, age, and face. Face detection and recognition techniques, as well as development tools, are explained and explored in the context of developing an Android mobile application. The software solution demonstrates how to utilize the OpenCV library and shows photographs of the smart phone app's actual outcomes.

Present are Zhu and Cheng [25]. In the Intelligent door lock system, an efficient Attitude Tracking Algorithm (EATA) based on OpenCV is employed for facial recognition. To enable the procedure, a customized application has been developed.

of devices, as well as the location's display. For monitoring and security purposes, an automated door surveillance system utilizing Raspberry Pi Python, USB camera, and OpenCV is provided. At some point in the past, the system status database was built. The device is inexpensive, easy to set up, and use. The Attitude Tracking Algorithm is highly precise and reliable.

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

[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 Poul-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	Diver’s 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.

We examined the accuracy of OpenCV approaches and classifiers used in a number of studies on face detection and recognition in different fields of computer vision using Table 1. Based on our observations of the 20 evaluated works (publications) (2018–2020) in the field of computer vision that employed OpenCV to examine the techniques of face detection and identification. OpenCV has been shown to be useful in a variety of disciplines, including face detection, face recognition, and facial expression identification [26], [27]. It was also discovered that OpenCV may be used in the security area, such as criminal identification [28]. Others utilized OpenCV to automate the attendance of students or employees in workplaces.Other researchers employed OpenCV to reduce traffic accidents on the highways at the same time.

8. CONCLUSION

Computer Vision is an Artificial Intelligence subject in which computers are trained to interpret pictures and extract key elements from them. OpenCV is a python library created in C++ that provides a variety of features for computer vision applications. Object identification, facial recognition, medical diagnosis, and more applications of computer vision may be found. The importance of OpenCV in face detection and identification is highlighted in this research. We show how to utilize OpenCV to demonstrate popular face detection and recognition techniques. Then list the OpenCV modules, explain how to use OpenCV with Python, and list some OpenCV applications.

9. 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). IEE

[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. In2019 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