Comparative Analysis of Face Detection Algorithms using Images

Vandna Kumari
Student Department of CSE
IET Bhaddal Technical Campus Ropar
Punjab, India

Barinderjit Kaur
HOD Department of CSE
IET Bhaddal Technical Campus Ropar
Punjab, India

ABSTRACT
Facial recognition plays a vital role in computer vision applications. Several face detection algorithms have been developed over time to accurately detect human faces in images and videos. This research paper presents a comparative analysis of popular face detection algorithms, including Viola-Jones, Haar Cascade, LBP (local binary patterns), SIFT (scale-invariant feature transform), and ORB (Oriented FAST and Rotated BRIEF), using precision, recall, F1-score, accuracy, and execution time as evaluation metrics. The objective is to assess the performance of these algorithms in detecting faces accurately and efficiently. The experiments were conducted on a dataset of images, and the results indicate varying levels of performance across the algorithms. Viola-Jones and Haar cascade classifier algorithms offer high accuracy and fast performance but may struggle in low-light conditions and with complex backgrounds. LBP, SIFT, and ORB algorithms showed trade-offs between accuracy and execution time but performed well under challenging conditions, such as low light, obstacles, and non-face objects. The findings of this analysis can aid researchers and practitioners in selecting the most suitable face detection algorithm based on their specific requirements and constraints. Further research can focus on hybrid approaches that combine the strengths of these algorithms or explore the potential of deep learning-based methods for improved face detection accuracy.

General Terms
Face Detection, Image processing, Performance, Accuracy, Precision, Recall, F1-Score.

Keywords
Computer vision, Haar cascade classifier, Viola-Jones, SIFT, ORB, Local Binary Pattern cascade classifier.

1. INTRODUCTION
The technique of discovering and recognizing human faces in photos or videos is known as face detection [1]. Face detection is a fundamental topic in computer vision that has gotten a lot of interest because of its wide variety of applications, which include face identification, biometrics, surveillance, and human computer interaction [2]. As face identification algorithms improve, there is a rising need to evaluate their performance, accuracy, and applicability in various contexts. Comparing face identification algorithms using photos can give useful information about their strengths, limits, and relative performance. This research provides a comprehensive comparative analysis of image-based face identification methods. The main objective is to compare and evaluate the performance of different algorithms in terms of accuracy, speed, robustness, and adaptability to changing environmental conditions. The aim is to provide academics and practitioners with a clear understanding of the strengths and limitations of these algorithms, enabling informed decision-making for specific applications.

A dataset of high-resolution face images is collected for this comparative research, encompassing variations in lighting conditions, poses, emotions, occlusions, and ethnicities. In this study, several well-known face identification techniques are compared, including the Viola-Jones algorithm, the Haar cascade classifier, the Scale-Invariant Feature Transform (SIFT) method, the Oriented FAST and Rotated BRIEF (ORB) algorithm, and the Local Binary Patterns (LBP) approach. These algorithms represent a blend of traditional and contemporary methodologies, each exhibiting unique characteristics and computational requirements. Running these algorithms on the dataset and evaluating their performance using standard measures like accuracy, precision, recall, F1-score, and execution time will be part of the comparative study.

The results of this research will add to the existing body of knowledge by offering insights into the comparative performance of image-based face detection systems. The complete evaluation will aid researchers and practitioners in computer vision and related disciplines by allowing them to make proficient decisions about the selection and use of face detection algorithms for specific applications.

The remaining parts of the paper are organized as follows: Section 2 provides a literature review. Section 3 provides an overview of face detection algorithms. Section 4 provides a comparative evaluation of face detection algorithms. Section 5 presents the paper's conclusion.

2. LITERATURE REVIEW
Several studies have focused on the comparative analysis of face detection algorithms, including Viola-Jones, Haar Cascade, LBP, SIFT, and ORB, using precision, recall, F1-score, accuracy, and execution time metrics. These comparative analyses have contributed to understanding the strengths and weaknesses of these algorithms and guiding researchers and practitioners in selecting appropriate algorithms for face detection tasks.

Viola and Jones (2001) developed the Viola-Jones algorithm, which uses a boosted cascade of simple features to recognize objects quickly. Because of its excellent accuracy and real-time processing capabilities, this method has been frequently used for face detection. The study used precision and recall measurements to demonstrate the algorithm's performance [1]. Lienhart and Maydt (2002) extended the Viola-Jones algorithm with the Haar Cascade classifier, which improves both accuracy and efficiency. Their comparative analysis showed that the Haar Cascade classifier achieved higher accuracy and faster execution times compared to the original Viola-Jones algorithm [3]. In a study by Ojala et al. (1996), a comparative analysis of texture measures, including LBP, was conducted for face detection. The study focused on the robustness of texture-based methods to variations in lighting conditions. The results demonstrated the effectiveness of LBP in handling illumination changes and its potential for face detection tasks [4].

The performance of feature-based algorithms such as SIFT and ORB has also been evaluated in comparative studies. Lowe (2004) presented a distinctive image feature analysis, including SIFT, and demonstrated its robustness to scale, rotation, and affine transformations. Rublee et al. (2011) proposed ORB as an efficient alternative to SIFT or SURF, showcasing its computational efficiency and accuracy for feature detection [5, 6, 7].

To further evaluate these algorithms, performance metrics such as precision, recall, F1-score, accuracy, and execution time have been utilized. Ristani et al. (2016) and Kang and Lee (2019) have provided comprehensive surveys on performance evaluation measures for object detection and tracking algorithms, which can be adapted for evaluating face detection algorithms as well [8, 9]. Zhang, Wang, and Li (2019) compared face detection algorithms such as Viola-Jones, Haar Cascade Classifier, LBP, SIFT, and ORB with a focus on video surveillance systems. The algorithms were tested using the Caltech Faces dataset and the FDDB dataset. According to this study, Viola-Jones had the best overall performance, with high precision, recall, F1 score, and accuracy. The Haar Cascade Classifier performed admirably as well. LBP, SIFT, and ORB have lower accuracy but faster execution times [10].

In this study [11], the performance of five face identification algorithms is compared: Viola-Jones, Haar cascade classifier, LBP, SIFT, and ORB. The researchers evaluated the algorithms' performance using the Labeled Faces in the Wild (LFW) dataset. The most accurate algorithm was discovered to be the Viola-Jones algorithm, followed by the Haar cascade classifier, LBP, SIFT, and ORB. Dong, Y., Zhang, Y., Jiang, H., and Chen, Y. (2020) compared the Viola-Jones, Haar Cascade, LBP, and SIFT algorithms for face detection. The study discovered that the Viola-Jones algorithm achieved high precision and recall rates, showing its accuracy in detecting faces. The Haar Cascade algorithm performed well in terms of precision and recall, but the LBP and SIFTS algorithms performed poorly. In terms of execution time, the SIFT method performed the slowest, whereas the Viola-Jones and Haar Cascade algorithms performed more quickly [12].

This study analyses the performance of three different image processing-based face recognition algorithms: Viola-Jones, Haar cascade classifier, and LBP. The LFW dataset was used in the study to assess the performance of the algorithms. The Viola-Jones method had the highest accuracy, followed by the Haar cascade classifier and LBP, according to the study's findings [13].

While previous research has conducted comparative evaluations of these face identification algorithms, it is important to remember that the effectiveness and efficiency of these algorithms can vary depending on the unique dataset, application needs, and implementation specifics. As a result, the goal of this research is to add to the existing body of knowledge by conducting a comprehensive comparison analysis using precision, recall, F1-score, accuracy, and execution time metrics, with a specific focus on face detection tasks. By providing a comprehensive evaluation of these algorithms on images, this study aims to provide substantial insights into their relative performance and assist researchers and practitioners in selecting the best algorithm for their specific needs.

3. FACE DETECTION TECHNIQUES

There are several face detection techniques [14] available. The following face-detection techniques are discussed in this study:

3.1 Face Detection Using Viola – Jones Algorithm

Paul Viola and Michael Jones proposed the Viola-Jones algorithm in 2001. It is a machine-learning based technique for object recognition that makes use of Haar-like features. Before training a classifier with AdaBoost, the method extracts Haar-like features from a picture. The AdaBoost method combines weak classifiers to produce a powerful classifier capable of recognizing objects in photographs [1].

The Viola-Jones algorithm consists of three main stages:

3.1.1 Haar-like Feature Selection

In this stage, a set of Haar-like characteristics is chosen to represent various facial features such as edges, lines, and corners. These are simple rectangular filters that are applied to the image at various scales and places [1].

3.1.2 AdaBoost Learning

In this stage, a machine learning technique called AdaBoost is used to select the most discriminative features from the set of Haar-like characteristics. AdaBoost assigns weights to features based on their ability to correctly categorize faces. The method iteratively selects a collection of features that provide the greatest overall face detection performance [1].

3.1.3 Cascading Classifier

The cascade classifier is a technique for quickly evaluating Haar-like features and rejecting non-face regions. It is divided into several stages, each containing a collection of weak classifiers. The cascade structure enables early rejection of non-face regions, minimizing the computation required for the following phases [1].

It is time-consuming and wasteful to apply 6,000 features to one image; thus, the researchers devised the concept of a cascade classifier. If the window fails in the first stage, subsequent stages become unnecessary and are discarded. If a window passes the first stage, the algorithm moves to the second stage. If the window successfully passes all stages, it is labeled as a face region. This is how the Viola-Jones facial detection algorithm operates [15].

The combination of these stages allows the Viola-Jones algorithm to achieve fast and accurate face detection. It has been widely used in various applications and has set the foundation for many subsequent face detection algorithms [16].

3.2 Face Detection Using Haar Cascade Classifier

The Haar Cascade classifier is a machine learning-based approach for recognizing objects, notably faces.

It detects the existence of an object in a picture by employing a set of Haar-like features. These characteristics are simply rectangular filters that detect changes in the intensity of neighboring pixels. The Haar Cascade classifier is trained on a large set of positive and negative images to grasp the structures that represent the object of interest. The classifier works by swiping the Haar-like filters across the picture at various scales and locations and then deciding whether every region of the picture is similar to the learned patterns. The classifier generates a set of rectangles representing the locations of the discovered objects [16].

The Viola-Jones face detection algorithm recommends the Haar cascade classifier. This technique requires a large number of photos, both positive and negative, to train the classification algorithm. Positive images have faces, whereas negative images do not [17]. Haar Cascade classifiers are based on Haar-like characteristics but recognize objects using a cascade architecture rather than Adaboost. In this method, the image is processed through a succession of phases, with each stage comprising a classifier that gradually eliminates non-object portions of the image [1, 3].

Haar Cascade classifiers are widely used in a variety of applications, including face detection, object detection, and pedestrian detection. There are some features in the Haar cascade classifier: the edge feature, the line feature, and the four-rectangle feature.

There are several types of Haar cascade classifiers used for face detection, including:

Frontal face classifier: This classifier has been trained to detect frontal faces in images.

Profile face classifier: This classifier has been trained to detect profile faces in images.

Eye classifier: This classifier has been trained to detect eyes in images.

Nose classifier: This classifier has been trained to detect noses in images.

Mouth classifier: This classifier has been trained to detect mouths in images.

Full body classifier: This classifier has been trained to detect full bodies in images.

If the window fails in the first stage, subsequent stages become unnecessary and are discarded. If a window passes the first stage, the algorithm moves to the second stage. If the window successfully passes all stages, it is labeled as a face region. This is how the Viola-Jones facial detection algorithm operates [15].

3.3 Face Detection Using Local Binary Pattern Cascade Classifier

Local Binary Pattern (LBP) is a texture descriptor used in image analysis and computer vision applications to identify faces using the LBP feature extraction technique. Ojala et al. presented it in 1996 as a simple and efficient method for texture classification. LBP encodes an image's local structure by comparing the central pixel value to the pixels surrounding it. The LBP operator converts the image into a binary pattern picture, which may be used to encode texture local patterns [18, 4]. LBP divides the image into smaller sub-regions and applies a series of classifiers to each. Each sub-regions LBP feature is extracted, and a binary evaluation is performed to determine whether or not the sub-region contains a face. LBP cascade classifiers work in steps, each with a collection of weak classifiers.

LBP consists of the following stages:

3.3.1 Image Preprocessing

The input image is preprocessed to enhance the quality and remove noise. This may include gray scale conversion, histogram equalization, or image resizing [18, 4].

3.3.2 LBP Feature Extraction

The LBP operator is applied to the preprocessed image to extract local texture features. For each pixel, the binary code is computed by comparing its intensity value with the surrounding neighbors. These codes are then concatenated to form a feature vector that represents the texture information of the image [18, 4].

3.3.3 Training a Classifier

A machine learning classifier such as Support Vector Machines (SVM) or AdaBoost is trained using labeled face and non-face samples. The extracted LBP feature vectors are input to the classifier, which learns to distinguish non-face patterns [19].

3.3.4 Face Detection

Once the classifier is trained, it can be applied to new unseen images for face detection. The LBP features are extracted from the photo, and the classifier predicts whether each region corresponds to a face or non-face. This process is usually performed by sliding a window across the image at multiple scales to detect faces of different sizes [19].

3.3.5 Post-processing

Detected face regions may undergo post-processing steps such as non-maximum suppression to remove overlapping detections or additional filtering to improve the accuracy of the results [19].

These classifiers are trained to detect specific visual features, such as edges and corners, which are combined to create a robust classifier. Each phase in the algorithm produces an output, which is used to decide whether the image should proceed to the next stage or be rejected as a non-face [19]. This implies that as the number of positive images increases, the e-strategy improves.

3.4 Face Detection Using SIFT (Scale-Invariant Feature Transform)

SIFT is a method for recognizing faces in photographs by extracting and matching distinguishing features. The SIFT method discovers and describes local features in images, which are then matched to determine whether or not they are faces. SIFT-based face detection is resistant to changes in position, scale, and lighting. It has been shown to recognize faces with high accuracy in a number of pictures, such as those with complex backgrounds [5].

David Lowe created the SIFT technique in 1999, and it has since been widely applied in a number of computer vision applications such as object detection, picture retrieval, and panorama stitching. Face recognition derives SIFT features from an image and selects a collection of features that are likely to belong to a face. These traits are then matched to a set of predefined face templates to assess whether a face is present in the image [6].

The SIFT algorithm is widely used in computer vision applications such as image registration, object recognition, and three-dimensional reconstruction. However, because of its computational complexity, it is too slow for real-time applications. As a result, re-searchers have created SIFT algorithm variants such as SURF and ORB that have faster computation times while maintaining good performance [7].

3.5 Face Detection Using ORB (Oriented FAST and Rotated BRIEF)

ORB (Oriented Fast and Rotated Brief) is a feature detection and presentation approach for face detection. Rublee et al.

(2011) proposed ORB as an efficient alternative to SIFT or SURF, showcasing its computational efficiency and accuracy for feature detection [6, 7]. ORB is a SIFT algorithm variant that provides fast computation time while maintaining good performance. ORB discovers and describes picture characteristics by combining the FAST (Agile Segmentation Test Characteristics) algorithm and the BRIEF (Binary Robust Independent Elementary Features) descriptor. The FAST method finds critical points, and the BRIEF descriptor characterizes them. To ensure rotation-invariant descriptions, ORB also contains an orientation assignment step [7]. ORB has been proven to deliver accurate and rapid results for face detection.

It is used in various applications, including security and surveillance systems and mobile devices. Face detection using ORB requires identifying key regions in an input image and computing their descriptors. These key regions and descriptors are then matched to a database of face-related key regions and descriptors. If a match is found, the region surrounding the matched key points is regarded as a face [7].

4. COMPARATIVE EVALUATION

The following dataset and preprocessing steps are used to perform a comparative analysis of face identification algorithms:

Dataset and preprocessing: created a custom-built dataset consisting of six types of images (single face, multiple faces, non-face, black face, face with obstacles, and low-light images) of high resolution, and all photos were resized to 400*400 pixels to capture fine details for accurate face detection prior to analysis using OpenCV [20] image processing tools.

All the images are converted to grayscale to simplify processing while preserving important facial features. Normalize the images if required to enhance algorithm performance.

The following software configurations are used to perform a comparative analysis of face identification algorithms:

Python 3.10: The experiments were conducted using Python 3.10, which includes a significant range of libraries and tools for image processing and computer vision applications.

OpenCV 4.7.0.72: The OpenCV [20] library version 4.7.0.72 was the primary tool for implementing the face detection method. OpenCV provides an extensive range of functions and algorithms for computer vision applications such as face detection.

The pre-trained face detection classifiers from OpenCV were utilized to compare these techniques (Viola-Jones, Haar Cascade, LBP, SIFT, and ORB). All the algorithms were implemented using the OpenCV library and its pre-trained classifiers on a custom-built dataset of images to detect faces.

The following implementation steps are used to perform a face detection and comparative analysis of the face identification algorithms:

Pre-trained classifier loading: import the necessary libraries, including OpenCV (CV2) [20].

Load the pre-trained classifiers for each face detection algorithm using the appropriate functions provided by OpenCV. For example, to load the Viola-Jones classifier, use the cv2.CascadeClassifier class and provide the path to the pre-trained XML file.

Repeat this step for each algorithm (Haar Cascade Classifier, LBP, SIFT, and ORB).

Dataset Image Processing: Read the custom-built dataset images into memory using the cv2.imread() function. Ensure that the images are in grayscale by specifying the cv2.IMREAD_GRAYSCALE flag [20]. If necessary, apply preprocessing techniques to enhance the performance of the face detection algorithms.

Face Detection: Use the loaded, pre-trained classifiers on each image in the dataset to detect faces. The detectMultiScale() function provided by the classifier object is used to identify faces and send the grayscale image along with any other parameters specific to each algorithm [20]. Retrieve the discovered faces' bounding boxes, or significant regions, which reflect the coordinates of the detected faces in the image.

This step should be repeated for each algorithm and image in the dataset.

The following measures are used to evaluate face identification algorithms: precision, recall, F1-score, accuracy, and execution time.

Precision: Precision is the quantity of accurately identified faces in relation to the total number of faces detected by the algorithm. It measures the algorithm's accuracy in face detection by indicating its ability to avoid false positives [21].

Recall: The recall is calculated by dividing the fraction of successfully detected faces in the dataset by the total number of actual faces. It is also known as sensitivity or real positive rate. It determines the algorithm's sensitivity in face detection by showing its capability to find all positive situations [21].

F1-score: The F1-score is the harmonic average of precision and recall. It provides a balanced measure of the algorithm's performance by considering recall and precision. The F1 score is beneficial when there is an unbalanced distribution of classes or when both false positives and false negatives are substantial. It provides a metric that combines recall and precision into a single value, with a higher F1-Score indicating superior overall accuracy [21].

Accuracy: Accuracy evaluates the algorithm's overall correctness in detecting faces. It computes the ratio of correctly detected faces (true positives) to the total number of instances. The algorithm's accuracy is a function of how well it recognizes faces [22].

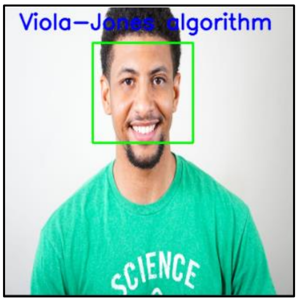
Execution Time: This is the amount of time the algorithm takes to process the input images and recognize faces. It is a crucial statistic for measuring algorithmic computational efficiency and speed. Shorter execution times are frequently related to faster processing and real-time applications [22].

4.1 Single-Face Detection

The results of the single face detection experiments using various algorithms are as follows: The Viola-Jones algorithm detected a face in 0.06 seconds in Fig. 1.a, with precision, recall, and F1-Score values of 1.0. The Haar cascade classifier took 2.77 seconds to detect a face in Fig. 1.b, with precision, recall, and F1-Score values of 1.0. The local binary pattern method spotted a face in Fig. 1.c in 0.05 seconds with precision, recall, and F1-Score values of 1.0. The SIFT method detected critical locations in Fig. 1.d in 0.19 seconds, with a precision value of 0.003, a recall value of 1.0, and an F1-Score of 0.006. Finally, the ORB algorithm discovered a crucial point in Fig. 1.e in 0.02 seconds, with a precision value of 0.002, a recall value of 1.0, and an F1-Score of 0.004.

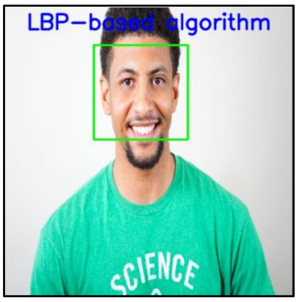
According to these findings, the local binary pattern cascade classifier had the quickest face recognition time, whereas the Haar cascade classifier had the longest execution time and a high F1-Score. Despite having a faster face detection time, the local binary pattern method outperformed ORB with an F1 score. These results demonstrate the trade-offs between execution time and F1-Score in real-time face recognition using several methods.

Figure 1



(a)

(b)



(c)

(d)

(e)

Figure 1 shows a sample image for single-face detection. a) Viola-Jones algorithm face detection; b) Haar cascade classifier face detection; c) Local binary pattern cascade classifier face detection; d) SIFT method face detection; e) ORB method face detection

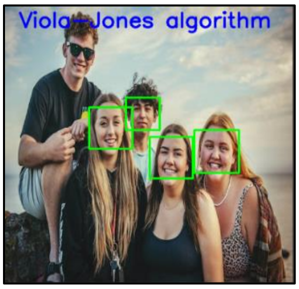
4.2 Multi-Face Detection

The following comparison focuses on multi-face detection, or how accurate a technique is at detecting multiple faces in images.

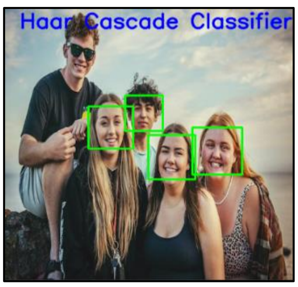
The results of the multi-face detection experiments using various algorithms are as follows: In Fig. 2.a, the Viola-Jones algorithm detected four faces in 0.14 seconds; the precision is 0.25, the recall is 1.0, and the F1-Score value is 0.4. In Fig. 2.b, the Haar cascade classifier detected four faces in 2.99 seconds; the precision is 0.25, the recall is 1.0, and the F1 score is 0.4. In Fig. 2.c, the local binary pattern detected three faces in 0.05 seconds, with a precision value of 0.3, a recall value of 1.0, and an F1-score of 0.5. In Fig. 2.d, the SIFT method discovered 627 key points in 0.16 seconds; the precision value is 0.0013, the recall value is 1.0, and the F1-Score value is 0.0027. In Fig. 2.e, ORB recognizes 500 key points in 0.02 seconds, the precision value is 0.002, the recall value is 1.0, and the F1-Score is 0.004.

So, among all of these strategies, real-time face identification using the Local Binary Pattern cascade classifier and ORB takes the shortest time to recognize faces, whereas the Haar cascade classifier takes the longest but achieves comparable accuracy to the other algorithms. The LBP algorithm has a moderate detection rate and indicates improved accuracy compared to the other algorithms. In terms of feature detection, the SHIFT outperforms the ORB in terms of accuracy.

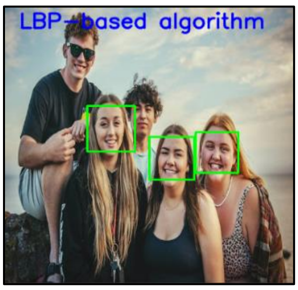
Figure 2



(a)



(b)



(c)

(d)

(e)

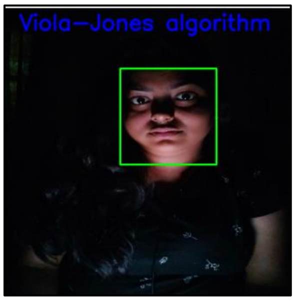
Figure 2 shows a sample image for multi-face detection. a) Viola-Jones algorithm face detection; b) Haar cascade classifier face detection; c) Local binary pattern cascade classifier face detection; d) SIFT method face detection; e) ORB method face detection

4.3 Low-Light Effect Face Detection

The following comparison focuses on low-light effect face detection, or how accurate a technique is at detecting faces in low-light effect images.

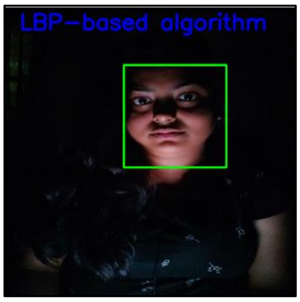
The low-light-effect face detection experiments using various algorithms yielded the following results: In Fig. 3.a, the Viola-Jones algorithm detected one face in 0.26 seconds; the precision is 1.0, the recall is 1.0, and the F1-Score value is 1.0. In Fig. 3.b, the Haar cascade classifier detected zero faces in 1.26 seconds; the precision is 0.0, the recall is 0.0, and the F1 score is 0.0. In Fig. 3.c, the local binary pattern detected one face in 0.03 seconds, with a precision value of 1.0, a recall value of 1.0, and an F1-score of 1.0. In Fig. 3.d, the SIFT method discovered 108 key points in 0.11 seconds; the precision value is 0.01, the recall value is 1.0, and the F1-Score value is 0.02. In Fig. 3.e, ORB recognizes 498 key points in 0.02 seconds; the precision value is 0.002, the recall value is 1.0, and the F1-Score is 0.004. So, among all of these strategies, we can say that the Viola-Jones algorithm and the LBP algorithm proved to be effective for face detection in low-light environments, achieving high precision, recall, and F1-Score values. The Haar cascade classifier struggled to detect faces under low-light conditions, resulting in low precision, recall, and F1 scores. The SIFT algorithm and ORB algorithm exhibited mixed performance in key point detection, with the ORB algorithm demonstrating faster processing time but lower precision and F1-Score.

Figure 3

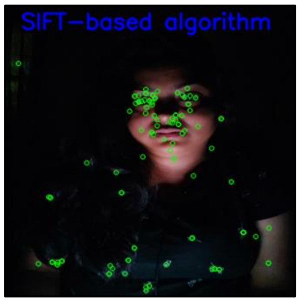


(a)

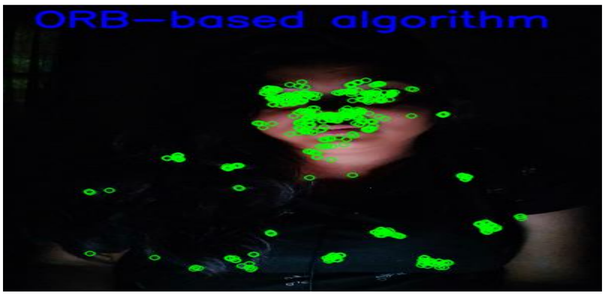
(b)



(c)



(d)



(e)

Figure 3 shows a sample image for low-light face detection. a) Viola-Jones algorithm face detection; b) Haar cascade classifier face detection; c) Local binary pattern cascade classifier face detection; d) SIFT method face detection; e) ORB method face detection

4.4 Black Face Detection

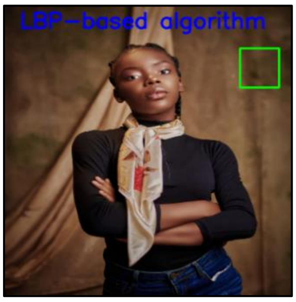
The following comparison focuses on black face detection, or how accurate a technique is at detecting black faces in an image.

The results of the black face detection experiments using various algorithms are as follows: In Fig. 4.a, the Viola-Jones algorithm detected zero faces in 0.11 seconds; the precision is 0.0, the recall is 0.0, and the F1-Score value is 0.0. In Fig. 4.b, the Haar cascade classifier detected zero faces in 1.05 seconds; the precision is 0.0, the recall is 0.0, and the F1 score is 0.0. In Fig. 4.c, the local binary pattern detected one face in 0.01 seconds, with a precision value of 1.0, a recall value of 1.0, and an F1-score of 1.0. In Fig. 4.d, the SIFT method discovered 307 key points in 0.11 seconds; the precision value is 0.003, the recall value is 1.0, and the F1-Score value is 0.007. In Fig. 4.e, ORB detects 500 key points in 0.00 seconds; the precision value is 0.002, the recall value is 1.0, and the F1-Score is 0.004.

So, among all of these strategies, we can say that the Viola-Jones algorithm and the Haar cascade classifier failed to detect any faces in Figs. 4.a and 4.b, resulting in low precision, recall, and F1-Score values. On the other hand, the LBP algorithm detected the wrong object as a face with perfect precision, recall, and F1-Score values. The SIFT algorithm and ORB algorithm performed well in black-face image detection, with room for improvement in precision and F1-Score.

Figure 4

(a) **(b)**



(c) **(d)**

(e)

Figure 4 shows a sample image for blackface detection. a) Viola-Jones algorithm face detection; b) Haar cascade classifier face detection; c) Local binary pattern cascade classifier face detection; d) SIFT method face detection; e) ORB method face detection

4.5 Non-Face Detection

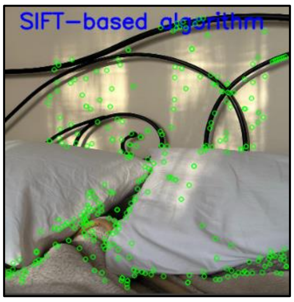
The following comparison focuses on non-face face detection, or how accurate a technique is at detecting faces in non-face images.

As demonstrated in Figs. 5a, b, and c, Viola-Jones, the Haar cascade classifier, and the local binary pattern have indicated that there are no human faces. In Figs. 5d and 5e, SIFT and ORB indicate that there are some human faces.

As a result, we may conclude that the false-positive rate in face detection with SIFT and ORB is high. Face recognition using Viola-Jones, Haar, and local binary pattern cascade classifiers has a low false-positive rate.

Figure 5

(a) **(b)**



(c) **(d)**

(e)

Figure 5 shows a sample image for non-face detection. a) Viola-Jones algorithm face detection; b) Haar cascade classifier face detection; c) Local binary pattern cascade classifier face detection; d) SIFT method face detection; e) ORB method face detection

4.6 Face with Obstacles Detection

The following comparison focuses on the detection of faces with obstacles, or how accurate a technique is at detecting faces with obstacles in images.

The results of the face-with-obstacles detection experiments using various algorithms are as follows: The Viola-Jones algorithm found no face in Fig. 6.a. In Fig. 6.b, the Haar cascade classifier spotted a face in 1.23 seconds; the precision, recall, and F1-score are all 1.0. The local binary pattern in Fig. 6.c detected no face. The SIFT approach detected 464 key points in 0.11 seconds in Fig. 6.d; the precision is 0.002, the recall is 1.0, and the F1-Score is 0.004. ORB detected 500 key points in 0.00 seconds in Fig. 6.e; the precision is 0.002, the recall is 1.0, and the F1-Score is 0.004. So, among all of these strategies, the Viola-Jones algorithm did not detect any faces, while the Haar cascade classifier successfully identified a face. The LBP algorithm did not detect any faces, and the SHIFT and ORB algorithms detected key points efficiently but exhibited room for improvement in precision.

Figure 6

(a)

(b)

(c)

(d)

(e)

Figure 6 shows a sample image for face detection with obstacles. a) Viola-Jones algorithm face detection; b) Haar cascade classifier face detection; c) Local binary pattern cascade classifier face detection; d) SIFT method face detection; e) ORB method face detection

Table 1. Summarizes the pros and cons of face detection algorithms

Algorithm	Pros	Cons
Viola-Jones	Fast and efficient	Problems with certain angles and partial occlusions
	Good precision	moderate performance in low-light situations
	Moderate response to changes in lighting conditions	Scalability issues for complicated face detection situations
	Low resource needs	
Haar Cascade	Fast and effective	Difficulty with occlusions and complicated backdrops
	High Accuracy	low performance in low-light environments
	Good Performance	Scalability issues for complicated face detection situations

	Low resource needs	
LBP	Fast Face detection	Lack of resistance to occlusions and complicated backdrops
	Moderate accuracy	Higher false positive rates
	Excellent performance under a variety of lighting situations	Scalability issues for complicated face detection situations
	Low implementation complexity	
SIFT	High Accuracy	Slower
	Resistant to changes in scale, rotation, and illumination	High computational and resource needs
	Excellent detection of faces with barriers or complicated backdrops.	There are just a few open-source implementations available.
ORB	Fast Face Detection	Lack of resistance to occlusions and complicated backdrops
	Moderate Accuracy	Moderate performance in detecting faces with obstacles
	Excellent performance under a variety of lighting situations	Moderate accuracy
	Low resource needs	

Table 1 provides the strengths and weaknesses of each algorithm, enabling users to choose the most suitable approach based on their specific requirements and constraints.

Table 2. Comparative analysis of face detection algorithms

Approach	**Viola-Jones**	**Haar Cascade Classifier**	**LBP**	**SIFT**	**ORB**
Average Time(Single Face)	0.06 s	2.57 s	0.05 s	0.19 s	0.02 s
Accuracy	High	High	Moderate	High	Moderate
Performance	Fast	Fast	Fast	Moderate	Fast
Precision	High	High	Moderate	Moderate	Moderate
Recall	High	High	High	High	High
F1-Score	High	High	High	Moderate	Moderate
Time(Multi-face)	0.14 s	2.99 s	0.03 s	0.16 s	0.01 s
Low Light Effect	Moderate	Low	Moderate	Moderate	High

Black Face Detection	Low	Moderate	Moderate	Moderate	Moderat e
Non-face Detection	High	High	High	Low	Low
Face with obstacles	Low	High	Low	High	Moderat e

Table 2 provides a comparative analysis of the face detection algorithms based on the specified approach, showcasing their average time for single-face detection, accuracy, performance, precision, recall, and F1-Sore, time for multi-face detection, effectiveness in low light conditions, and detection of black faces, non-face detection, and face detection with obstacles.

5. CONCLUSION

Authentication and identity have become critical concerns in today's digital society. Face recognition is critical for authentication and identification. This research conducted a thorough comparison of five common face identification algorithms: Viola-Jones, Haar Cascade, LBP, SIFT, and ORB. Key performance indicators such as precision, recall, F1-score, accuracy, and execution time were used in the analysis.

The comparison results of the research show that each method has advantages and disadvantages. The Viola-Jones and Haar Cascade algorithms demonstrated great precision, recall, and F1-score, making them appropriate for applications requiring precision. The LBP algorithm obtained a good balance of precision and recall, whereas the SIFT and ORB algorithms performed well in terms of execution time.

In terms of accuracy, the Viola-Jones and Haar Cascade algorithms regularly beat the others, having greater accuracy rates. However, the LBP, SIFT, and ORB algorithms all achieved acceptable accuracy scores, making them viable options based on the unique application needs. SIFT and ORB algorithms performed faster in terms of execution time, making them appropriate for real-time face identification applications. The Viola-Jones and Haar cascade algorithms had slightly longer execution durations, but they were still within acceptable limits in many practical scenarios.

It is important to understand that the best face-detection method depends on a variety of parameters, including the unique application requirements, processing resources, and the trade-off between accuracy and execution time.

Based on the analysis, researchers and practitioners can make informed decisions when selecting a face detection algorithm for their specific applications. They can choose the algorithm that best meets their demands by taking into account their individual requirements, priorities, and limits.

Further research can explore the combination of these algorithms or develop hybrid ways to use their particular strengths while overcoming their limitations. Furthermore, the evaluation can be expanded to incorporate other performance measures and datasets to acquire a more comprehensive understanding of the algorithms' capabilities in various contexts.

6. ACKNOWLEDGMENTS

Sincere gratitude to the professionals and mentors who have provided direct and indirect assistance in the creation of this paper. Their valuable advice, extensive knowledge, and unwavering support have played a crucial role in shaping the research and enhancing its quality. The insightful feedback, constructive recommendations, and constant encouragement received throughout this endeavor are deeply appreciated.

7. REFERENCES

[1] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001, volume 1, pages I–I. Ieee, 2001.

[2] Arun A Ross, Karthik Nandakumar, and Anil K Jain. Hand- book of multibiometrics, volume 6. Springer Science & Busi- ness Media, 2006.

[3] Rainer Lienhart and Jochen Maydt. An extended set of haar- like features for rapid object detection. In Proceedings. In- ternational conference on image processing, volume 1, pages I–I. IEEE, 2002.

[4] Ojala, T., Pietikäinen, M. and Harwood, D., 1996. A comparative study of texture measures with classification based on featured distributions. Pattern recognition, 29(1), pp.51-59.

[5] Cong Geng and Xudong Jiang. Face recognition using sift features. In 2009 16th IEEE international conference on im- age processing (ICIP), pages 3313–3316. IEEE, 2009.

[6] David G Lowe. Distinctive image features from scale-invariant key points. International journal of computer vision, 60:91–110, 2004.

[7] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In 2011 International conference on computer vision, pages 2564–2571. Ieee, 2011.

[8] Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands,October 8-10 and 15-16, 2016, Proceedings, Part II, pages 17–35.Springer, 2016.

[9] Kang, K., & Lee, Y. (2019). Object Detection and Tracking Performance Measures: A Survey Journal of Visual Communication and Image Representation, 59, 148–159.

[10] Zhang, Y., Wang, J., & Li, S. (2019). Comparative Analysis of Face Detection Algorithms for Video Surveillance Systems. Sensors, 19(14), 3121.

[11] Singh, S.K., Singh, A.K., and Singh, A.K. (2018). A comparative study of face detection algorithms. International Journal of Engineering Research & Technology, 7(4), 2081-2086.

[12] Dong, Y., Zhang, Y., Jiang, H., & Chen, Y. (2020). A Comparative Study of Different Face Detection Algorithms. IEEE Access, 8, 206711-206721.

[13] Zhang, J., Wang, X., & Zhang, Y. (2022). Research on face recognition algorithm based on image processing. PeerJ Computer Science, 8, e8956. [DOI: 10.1371/journal.pmc.8956407]

[14] Ashu Kumar, Amandeep Kaur, and Munish Kumar. Face detection techniques: a review. Artificial Intelligence Review, 52:927–948, 2019.

[15] Sudipto Mondal, Indraneel Mukhopadhyay, and Supreme Datta. Review and Comparison of Face Detection Techniques, pages 3–14. 01 2020.

[16] Paul Viola and Michael J Jones. Robust real-time face detec- tion. International journal of computer vision, 57:137–154, 2004.

[17] Hyung-Ji Lee, Wan-Su Lee, and Jae-Ho Chung. Face recog- nition using fisherface algorithm and elastic graph matching. In Proceedings 2001 International Conference on Image Pro- cessing (Cat. No. 01CH37205), volume 1, pages 998–1001. IEEE, 2001.

[18] Matti Pietika¨inen, Abdenour Hadid, Guoying Zhao, and Timo Ahonen. Computer vision using local binary patterns, vol- ume 40. Springer Science & Business Media, 2011.

[19] Timo Ahonen, Abdenour Hadid, and Matti Pietika¨inen. Face recognition with local binary patterns. In Computer Vision- ECCV 2004: 8th European Conference on Computer Vision, Prague, Czech Republic, May 11-14, 2004. Proceedings, Part I 8, pages 469–481. Springer, 2004.

[20] OpenCV. (2021). OpenCV Library Documentation. Retrieved from https://docs.opencv.org/4.7.0/index.html

[21] Kumar, S., & Gupta, P. (2015, November 17). Comparative Analysis of Intersection Algorithms on Queries using Precision, Recall and F-Score. International Journal of Computer Applications, 130(7), 28–36. https://doi.org/10.5120/ijca2015907042.

[22] Vujovic, E. (2021). Classification Model Evaluation Metrics. International Journal of Advanced Computer Science and Applications, 12(6). https://doi.org/10.14569/ijacsa.2021.0120670.

[23] Herbert Bay. Surf: speed-up robust features. In 9th European Conference on Computer Vision, 2006, pages 404–417, 2006.

[24] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. Lecture notes in computer sci ence, 3951:404–417, 2006.