e-ISSN: 3023-4409

INTERNATIONAL JOURNAL OF APPLIED METHODS IN ELECTRONICS AND COMPUTERS

www.ijamec.org

International Open Access

Volume 11
Issue 04

December, 2023

Research Article

Detection of Face-Mask in Real Time: A Cascaded Bi-Level Feature Extraction Technique Approach

Solomon Adelowo Adepoju [a] **,** ***Enobong Thomas Adahada*** [b,] ***,Opeyemi Aderiike Abisoye*** [c] ***Abdumalik Danlami Mohammed*** [d]

[a,b,c,d] *Department of Computer Science, Federal University of Technology Minna, Nigeria*

ARTICLE INFO

Article history:
Received 18 September 2023
Accepted 21 November 2023

ABSTRACT

Due to COVID-19's rapid spread, millions of people around the world have been affected and there has been extensive destruction. One of the most effective ways of preventing its spread and transmission during the pandemic period wearing of a mask and was required then in most public areas. As a result, this necessitate the use of programmed real-time mask detection devices in place of manual reminders. Face mask detection requires real-time processing of a huge amount of data with constrained processing resources, hence local descriptors that are quick to calculate, quick to match, and cheap to store are highly sought after. To achieve improved matching and reduction in memory use and computational complexity, the study offers a combination of Features from Accelerated Segment Test (FAST) corner detector with Histogram of Oriented Gradient (HOG) feature descriptor to allow faster matching and minimize memory usage and computation cost. The features obtained were then classified into face mask present and face mask absent using SVM, NB and CNN. Results obtained gives an accuracy of 99.41% which was higher than that previous results of 99.27% and 95% accuracy. Furthermore, it took the system only 48secs to extract the features obtained from face for training and testing. This outcome confirmed the suitability of the suggested method for real-time face mask detection.

This is an open access article under the CC BY-SA 4.0 license.
(https://creativecommons.org/licenses/by-sa/4.0/)

1. Introduction

COVID-19 (Coronavirus Disease 2019) surfaced unexpectedly in 2019 and has had a global impact. Whenever an infected person coughs or sneezes, COVID-19 is spread via saliva or nasal secretion droplets. It is very contagious and can spread quickly especially in crowded and congested areas. Coronavirus spread has been demonstrated to be reduced when face masks are worn [1], making it one of the most effective prophylactic methods known [2]. The correct technique to wear a face mask, according to the World Health Organization (WHO), is to adjust it to cover the mouth, nose, and chin [3]. The protection provided by masks is greatly decreased when they are not worn correctly. Security personnel are currently located in some public spaces, advising individuals to wear masks. Unfortunately, because of its inefficiency, this technique exposes the security officers to virus-infected air and generates congestion at the entrances. As a result, prompt action is necessary [4].

Computer vision is a cross-disciplinary field of computing that deals with how computers learn from digital images or films [5] Picture processing, image categorization, object identification, and image recognition are examples of traditional computer vision tasks. Object identification can discover examples of a given class of visual elements in photographs, which is a suitable solution to the issue described above [2], [6]. As a result, mask detection has become an important computer vision problem for supporting global civilization.

Face detection and recognition is a branch of computer vision that can be used to detect face masks.

* Corresponding author. E-mail address: *sa.adepoju@gmail.com*

DOI: 10.58190/ijamec.2023.66

Face detection’s major goal is to identify the part of a photo or video stream that represents a face and recognize the location[7]. Face detection is utilized in this study to determine which areas of an image or video stream the face mask detection algorithm should focus on to determine the presence of a mask.

Computer vision technology such as feature point descriptors are essential. Due to the vast volume of data that must be analyzed in real time and the restricted processing resources available, local descriptors that are quick to calculate, quick to match, and cheap to store are highly sought after. Working with short descriptors is a good approach to hasten up matching and save storage. Short descriptors can be achieved by employing dimensionality reduction like Principal component analysis, to an original descriptor such as Scale Invariant Feature Transform (SIFT) [8], [9] or Speed Up Robust Feature (SURF) [10], [11]. Though powerful, these strategies of dimensionality reduction necessitate computing the entire descriptor before proceeding with additional processing, which is time-consuming and computationally expensive.

Deep neural networks also suffer from computational complexity and require a large amount of data to perform better than other techniques [12]. In order to speed up matching, reduce storage requirements, and simplify computing, this study suggests a cascade of the FAST corner detector and the HOG feature descriptor. HOG is resistant to geometric and photometric alterations since it works on local cells. Combining FAST with HOG will further improve computational speed as HOG will only have to describe the points detected by FAST and not the whole points in the image.

This research work is birthed from the fact that Face masks are required for public use, especially in public areas and large gatherings, in order to contain the spread of the fatal pandemic. The setback is, few individuals are willing to wear the face mask, and those who do are likely not to do so in the recommended way, making efforts to stop the fatal pandemic from spreading nearly impossible. Computer vision technology, such as feature point descriptors, is critical. Since face mask detection requires a considerable quantity of data to be processed in real-time or on devices with limited processing resources, there is an increasing demand for local descriptors that are fast to compute, fast to match, and memory efficient.

2. Related works

Since COVID-19 outbreak, wearing of face masks have become a vital requirement in people's daily lives, and identifying masks on people has become a significant direction of study.

By mixing the anchor point distribution technique and data enlargement, Wang et al. [13] introduced a novel anchor-level attention approach for obstructed face identification that could increase the characteristics of facial areas and enhance the precision without sacrificing speed. The study of [13], on the other hand, did not consider of mask detection.

Cabani et al. [14] introduced masked facial pictures based on facial feature locations and created a huge dataset of 137,016 masked face shots, which provided more training data. Concurrently, a smartphone app was created that taught people how to use masks properly by determining if they covered both the nose and the mouth. The models' detection speed, on the other hand, was not addressed.

Incorrect Face Mask Detection (FMD) was proposed by Tomás et al [15]. The suggested model employs Convolutional Neural Networks (CNN) and transfer learning to determine whether a mask is being utilized, as well as other faults that are often overlooked yet can contribute to viral propagation. This study gathered data by asking participants to snap various pictures with the mask in various orientations using an app. The suggested model has the drawback of being expensive to train because it uses a lot of computation power such as memory space and time.

Fan and Jiang [3] developed a Feature Pyramid Network (FPN) blended with a content attention method to deal with the problem of discriminating between right and erroneous mask-wearing states. ResNet was selected as the backbone network because it can run on both high- and low-cost hardware.This research was successful, with a 95% accuracy rate. The detection speed, on the other hand, was not mentioned.

Ejaz and Islam [7] proposed a model for face mask recognition. The suggested technique initiates by identifying the face regions. The Google FaceNet anchoring model is then used to retrieve facial traits. Finally, a Support Vector Machine (SVM) was used to execute the categorization task. Additional performance indicators like as time, precision, and f-score were not taken into account while evaluating the effectiveness of the algorithm.

For improved feature extraction and categorization Loey et al. [4] utilized a blended transfer learning framework and machine learning tactics. The developed framework consists of two parts. The first module is made to extract features using Resnet50. The subsequent stage is intended for use in the categorization of face masks utilizing decision trees, SVM, and the ensemble technique. Three face-masked databases were utilized in the study. The Real-World Masked Face Dataset (RMFD) [16], the Simulated Masked Face Dataset (SMFD) [16], and the Labeled Faces in the Wild (LFW) [17] are the 3 datasets used. The final accuracy reached using the ensemble classifier was 99.64% on the RMFD, 99.49% on the SMFD, and 100% on the Labeled Faces in the Wild LFW dataset. Nonetheless, the research focused on mask classification accuracies, and the speed of feature identification and extraction was not adequately addressed.

3. Methodology

This section contains a description of the procedures utilized to conduct the study. Data collection, image preprocessing, extraction of features and classification are some of the approaches used. Figure 1 depicts the suggested system.

3.1. Data Collection

The FMD task must determine whether or not an individual is wearing a mask. Masked face image samples are required for the FMD task. The proposed technique was trained and tested using the Real-world Masked Face Recognition Dataset (RMFD).

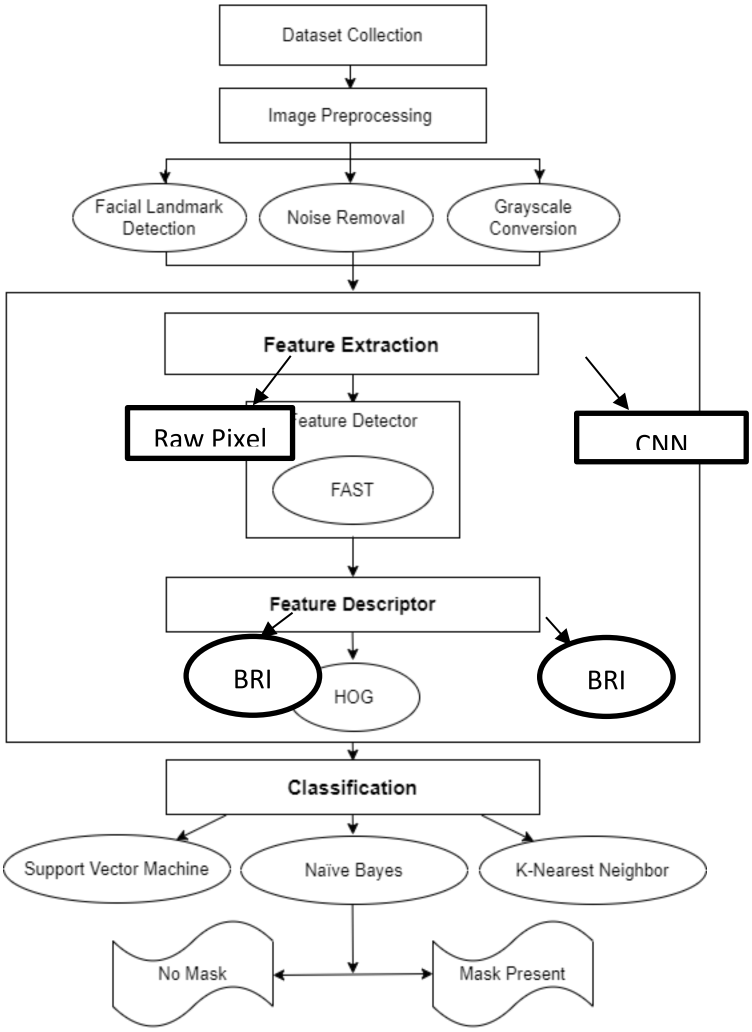

Figure 1 The Proposed Method

3.2. Real-world Masked Face Dataset (RMFD)

The RMFD dataset was developed by [16]. The front-facial photos of prominent people and their matching masked facial images were crawled from huge Internet resources using a python crawler program. The unnatural facial images that resulted from the incorrect correlation were then manually deleted. Finally, semi-automatic annotating programs like LabelImg and LabelMe were used to crop the accurate facial areas. There are 90,000 face photographs without masks in the dataset, 5000 facial images with masks in the dataset, and 525 subjects. A set of face photos are shown in Figures 1 and 2. Figure 1 shows a face without a mask, whereas Figure 2 shows a face with a mask.

Figure 2 Face image without mask

Figure 3 Face image with mask

In this paper, given that there are 90,000 images without a mask and only 5000 images with masks, to avoid data imbalance, 6000 face images without masks and all the 5000 faces with masks were used for the model training and testing. This makes the used dataset a total of 11 000 images.

In the training phase, 80% of the total dataset was utilized for training while 20% were used for testing the trained model.

3.3. Image pre-processing

Three procedures were performed during the face preparation step. Facial landmark detection, grayscale dialogue, and noise removal are among these procedures. The subsections following go into each of these procedures in depth.

3.4. Facial landmark detection

Face characteristics such as the nose, brows, mouth, jaw and eyes are used to limit and demarcate central emphasis areas [18], [19]. In this research, the Viola-Jones technique was employed to distinguish facial features. Haar-basis filtering, a scalar entity in the center of the photo, and several Haar-like structures are used in the Viola-Jones algorithm [19]. The four parts of this method for face identification are Haar feature selection, integral photo screening, Adaboost training, and a cascade classifier [20]. The Viola-Jones technique converts the input image into an integral image first. The integral photo is an operational approach for obtaining the pixel summation intensity in a square of an image.

3.5. Grey-Scale Conversion

In image processing, a grey-scale photo is one in which every pixel has only intensity distribution and is represented by a single data point indicating only a quantity of light. Grey-scale photographs, which are a type of grey monochromatic, are entirely composed of grey areas. The contrast changes from black to white at the lowest intensity and vice versa at the highest [21]. To prepare the photos for feature retrieval, the trimmed coloured face pictures were transformed to greyscale images.

3.6. Median Filter

For image processing, image de-noising is a vital step in picture processing. The median filter is utilized to de-noise images. The median filter is a non-linear filter that reacts to pixel value rankings within the filter area [22]. The median filter is commonly used to attenuate noise of various forms. The pixel's centre value is substituted by the median of the pixel values under the filter area. The median filter is successful for salt and pepper noise. Image smoothers and signal processing frequently use these filters. The median filter has a substantial advantage over linear filters in that it can remove the effect of extremely large input noise values [23].

4. Feature extraction

The information like points and lines which are included in image features is critical for image analysis. Many methods are used to obtain features from images. The image corner detection will be done using the FAST method, and feature description will be done using the HOG approach.

4.1. Feature from Accelerated Segment Test (FAST)

FAST is a well-known technique for detecting objects of interest in images that was first proposed by Rosten and Drummond [24]. FAST only has one parameter: the intensity threshold amid the centre pixel and the pixels in a round ring surrounding it [25]. FAST is quick in matching. The precision is also rather superb. Because FAST is not a scale-space detector, identifying the boundaries at a specific scale can provide much more than a scale-space approach such as SIFT [8]-[9].

A 16-pixel circle is used by the FAST corner detection to determine whether a potential point p is indeed a corner. Every pixel in the circle is labeled from 1 to 16 in a clockwise direction. When a group of N consecutive pixels in a circle are all brighter than the luminosity of prospective pixel p (signified by I_P) plus a predefined threshold t, or all darker than the brightness of candidate pixel p minus threshold value t, p is designated as a corner [26]. The conditions can be written as:

- **Condition A:** A set of N neighboring pixels S, $\forall x \in S$, the intensity of $x > I_P$ + threshold, or $I_x > I_P + t$
- **Condition B:** A set of N contiguous pixels S, $\forall x \in S, I_x > I_P - t$

Candidate p can be categorized as a corner if one of the 2 conditions is satisfied.

4.2. Histogram of Oriented Gradient (HOG)

HOG is a gradient approximation-based feature extraction method for object recognition in image analysis. The HOG extraction approach calculate the number of times a gradient orientation happens in a certain part of an image exploration window [27]. This research employed HOG extracted features since it is insensitive to geometric and photometric variations [28]. The steps to generate HOG features are as follows: After preprocessing and scaling the image, the magnitude and orientation of each pixel in the photo are computed, and the magnitude and orientation are estimated using the methods in equations 1 and 2 [29].

$$\text{Total Gradient Magnitude} = \sqrt{(G_x)^2 + (G_y)^2} \quad (1)$$

Where G_y is the gradient in the y-axis, and G_x is the gradient in the x axis.

$$\text{Orientation} = tan\ (\theta) = G_y/G_x \quad (2)$$

The value of the angle (θ) is presented in equation 3

$$\theta = atan\ (G_y/G_x) \quad (3)$$

The fundamental benefit of utilizing HOG in this work is that it encodes edge and luminance structure, which is a key element of local form, in a regional representation that is highly resistant to local photometric and geometric changes [30]: If translations are substantially thinner than the local spatial width or oriented bin dimensions, it makes no impact [27].

4.3. Convolutional Neural Network (CNN)

The CNN has proven to be a highly effective approach of extracting features and recognition in the field of computer vision and analysis [31]. The most typical deep learning model is CNN [32]. CNN is a multi-layer neural network with multiple 2D surfaces in each layer and several autonomous neurons in each plane. CNNs contain a huge number of links, and their design is built up of numerous layers that produce some type of regularization, such as pooling, convolution, and fully linked layers [33]. The pooling layer, input layer, convolutional layers and non-linear layer make up the conventional CNN structure. The fully connected layers execute the categorization of the features derived by the convolutional and pooling layers, whereas the convolutional and pooling layers are accountable for feature extraction [31].

4.4. Raw Pixel-based features

Images are characterized by pixels, which indicates that the easiest way to generate image characteristics is to use these raw pixel values as distinct features. The number of retrieved features will be equal to the image's pixel count.

4.5. Binary robust independent elementary features (BRIEF)

BRIEF is a binary descriptor built on pair-wise intensity comparisons proposed by [34]. A specified number of pairs (128, 256, or 512) are picked at random in a patch surrounding a keypoint. The intensity difference test is used to calculate the BRIEF descriptor for these pairs. If the intensity at one site is greater than at elsewhere, the test returns 1. Else, it returns a value of 0. Because it is based on binary contrasts, this descriptor is operationally faster than SIFT or SURF. BRIEF is unaffected by changes in illumination, but not by scale or rotation. A BRIEF descriptor is 32-dimensional in its usual configuration [35].

4.6. Binary Robust Invariant Scalable Keypoints (BRISK)

The BRISK method is a scale and rotation invariant feature point identification and characterization algorithm. It finds the binary feature descriptor by creating the characteristics descriptor of the localized image using the grayscale connection of random point pairs in the localized image's neighborhood [36].

5. Classification

Machine learning capability to generalize is defined by its ability to correctly classify unknown data based on models created using the training dataset [37]. Face photos were classified into two classes: no mask (0) and mask present (1). using SVM classification models.

5.1. Support Vector Machine (SVM)

The SVM algorithm is a supervised learning model [38]. The structural risk reduction approach is used in this technique, which enables it to compact an array of raw data into a support vector set and learns how to accomplish a categorization decision function [39]. By collecting data points, the SVM model iterates over a set of labelled training instances to create a hyper-plane that provides an optimal path cap. Support vectors help to increase class distinction [40]. Equation 4 expresses the decision rule of a SVM in the input space

$$\gamma = h(x) = sign\left(\sum_{j=1}^{n} u_j y_j\, K(x, x_j) + \right) \tag{4}$$

where x is the feature vector to be classified, j is the training instance index, n is the quantity of training examples, and y_j is the training example label (1 or –1). u_j and v are fitted to the data to optimize the boundary, and j, $K(x,x_j)$ is the kernel function. Support vectors are training variables for which $u_j \neq 0$ [41]

5.2. K-Nearest Neighbour (KNN)

In KNN, classification of items is based on its “distance” from its neighbours. Then it is allocated to the most common class of its k closest neighbours [42]. If k = 1, the algorithm becomes the nearest neighbour algorithm, and the object is allocated to the nearest neighbour’s class. This number K indicates how many neighbours an object has [43] .

The Euclidean distance depicts a linear distance between two points in Euclidean space [43]–[45]. Given two vectors y_i and yj , where y_i =(y_{i1}, y_{i2}, y_{i3}, …, y_{in}) and given y_j =(y_{j1}, y_{j2}, y_{j3}, …, y_{jn}), The equation for the Euclidean distance between yi and yj is by:

$$D(y_i, y_j = \sqrt{\sum_{k=1}^{n} \left(y_{ik} - y_{jk}\right)^2} \tag{5}$$

The following is a description of the K-NN algorithm:

Step 1: Assigns a positive integer k to each new sample.

Step 2: In the database, select k entries that are closest to the new case.

Step 3: The most common category is found for such entries.

Step 4: We assign a category to the new sample.

5.3. Naïve Bayes

This a supervised learning method and a statistical classification scheme are both demonstrated in the NB Model. It is based on an intrinsic probabilistic model and aids in measuring the results’ probabilities to obtain principled uncertainty about the model [46]. The NB classifier is a machine learning algorithm that uses probability which is based on the Bayes theorem and the assumption of great feature independence. Learning involves numerous linear parameters in the number of problem functions, and NB classifiers are very scalable

[47], [48]. The Bayes theorem provides a way to compute the posterior probability P(x|y)fromP(x), P(y) andP(y|x) in NB. Equation (6) and (7) presented the equation for posterior probability P(x|y).

$$P(x|y) = \frac{P(y|x) \times P(x)}{P(y)}$$

$$P(x|y) = \frac{P(y_1|x) \times P(y_2|x) \times \ldots \times P(y_n|x) \times P(x)}{P(y_1, \ldots, y_n)}$$

6. Performance Metrics

The suggested system's performance is compared using Precision, Execution Time, Recall, F-Score, and Accuracy performance evaluation metrics.

6.1. Precision

This metric measures how many correct positive forecasts have been made. The proportion of accurately forecasted positive instances divided by the number of predicted positive instances is used to compute it. The formula in equation 5 can be used to calculate the precision.

Precision=TP / (TP+ FP) (5)

Where TP and FP represent True Positive and False Positive

6.2. Recall

The recall is an indicator that shows how many correct positive predictions were produced out of all possible positive predictions. The recall is calculated using the formula in equation 6.

Recall=TP/ (TP+ FN) (6)

FN is False Negative

F-score

The harmonic average of recall and precision is known as the F-measure. Equation 7 represents this definition numerically.

$$\mathrm{F-measure} = 2 * \frac{\mathrm{precision} * \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}} \quad (7)$$

6.3. Accuracy

The rate of correctly classified instances can be used to define ac

curacy. Equation 8 represents this Accuracy definition numerically:

Accuracy=(TP + TN)/(TP + TN +FP + FN) (8)

6.4. Execution Time

Execution time measures the time taken for the model to perform feature extraction.

7. Results and Discussion

With regard to face mask identification and classification, experiments were conducted on five distinct feature descriptor techniques: Raw pixel features, BRIEF, BRISK, HOG, and CNN descriptors. The SVM and KNN classifiers were used to classify these retrieved characteristics. Table 1 shows the outcomes of the various feature descriptor strategies using SVM.

Table 1. Face Mask Classification Result using SVM classifier

Feature Extractors	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Raw Pixel-based	95	95	95	95
FAST + BRIEF	95.54	96.57	96.57	96.57
FAST + BRISK	91.40	99.32	84.88	91.54
FAST + HOG	99.46	99.41	98.83	99.12
CNN	99.12	100	98.32	99.15

From Table 1, it can be deduced good accuracy value are obtained from the extracted features . However, the FAST+HOG with a value of 99.46% gave the highest categorization accuracy as compared to Raw Pixel-based, FAST+BRIEF, FAST+BRISK and CNN with a classification accuracy of 95.00%, 95.54%, 91.40% and 99.12%, respectively. Based on the precision metric, CNN produced a higher precision value of 100% than Raw Pixel-based, FAST+BRIEF, FAST+BRISK and FAST+HOG with 95%, 96.57%, 99.32% and 99.41 %, respectively.

Looking at the obtained recall values, the FAST+HOG has a high recall value of 98.83%, this demonstrates that the number of correct positive assumptions made from all positive predictions is higher than the number of correct positive predictions made by Raw Pixel-based, FAST+BRIEF, FAST+BRISK and CNN with a recall of 95%, 96.57%, 84.488% and 98.32% respectively. From the recall value, it can also be seen that FAST+BRISK had the lowest recall value.

Observing from the F1-score, CNN produces the highest F1-Score of 99.15%, followed by FAST+HOG with an F1-Score of 99.12%. Comparing the various feature descriptors based on the execution, the FAST+BRIEF descriptor had the least execution time of 46.04 seconds, followed by FAST+HOG with 48.00 seconds. The

FAST+BRISK descriptor took longer than the other descriptors when describing and extracting the image features. From the execution time, Accuracy, F1 score, Precision and Recall, it is observed that the FAST+HOG descriptor is more suitable for a reliable Face mask identification than the other four descriptors. Figure 4 shows a visual representation of SVM classification performance on all the feature descriptors.

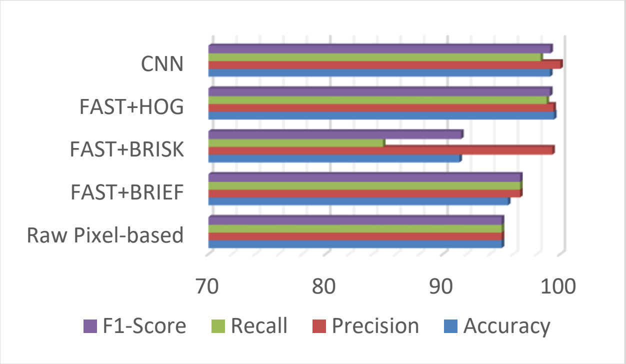

Figure 4 Performance Comparison of CNN, HOG, BRISK, BRIEF and Raw Pixel for SVM classification

Figure 4 gives a clear visualization of the accuracy, recall, precision, and f-score for CNN, FAST + HOG, FAST + BRISK, FAST + BRIEF and Raw Pixel-based. The accuracy, precision, recall, and f1-score values represented in the chart are displayed in Table 1.

Table 2: Face Mask Classification Result using K-Nearest Neighour (KNN)

Feature Extractors	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Raw Pixel-based	92.99	97.60	90.06	93.68
FAST + BRIEF	93.94	94.80	95.04	94.92
FAST + BRISK	87.89	86.49	92.49	89.39
FAST + HOG	98.10	99.43	97.22	98.31
CNN	97.46	99.89	95.65	97.77

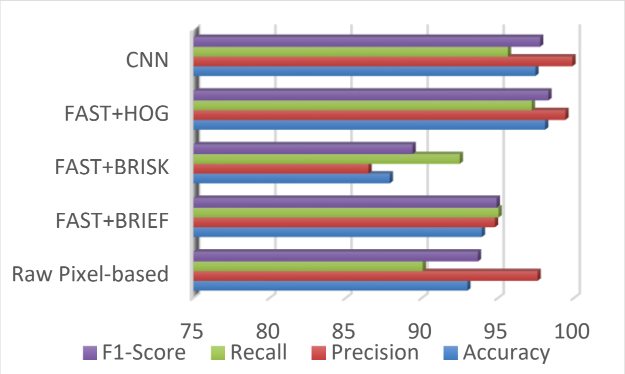

Figure 5 Comparison of CNN, HOG, BRISK, BRIEF and Raw Pixel for KNN classification

Table 2 displays the outcomes of the five various feature descriptor methods using the KNN classifier. The accuracy values that each of the retrieved picture characteristics produced were satisfactory. In contrast to raw pixel-based, FAST+BRIEF, FAST+BRISK, and CNN, which had classification accuracy values of 92.99%, 93.94%, 87.89%, and 97.46%, respectively, FAST+HOG had the best accuracy of 98.10%. In comparison to CNN, the precision values of raw pixel, FAST+BRIEF, FAST+BRISK, and FAST+HOG were all lower (99.89% vs. 97.60%, 94.80%, 86.49%, and 99.43%, correspondingly). When recall values are compared, the FAST+HOG has a high recall value of 97.22%, indicating that the ratio of correct positive assumptions made from all the true positives is higher than that of the raw pixel-based, FAST+BRIEF, FAST+BRISK, and CNN, with recalls of 90.06%, 95.04%, 92.49%, and 95.65%, respectively. The accuracy, precision and recall values also make it evident that FAST+BRISK had the lowest performance. FAST+HOG is ranked first with a f1-score of 98.31%, followed by CNN with an F1-score of 97.77% in terms of evaluation. The execution time of the various feature descriptors remains the same. Figure 5 offers a graphic representation of KNN's performance on each of the feature descriptions.

Table 3 Face Mask Classification Result using Naive Bayes (NB)

Feature Extractors	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Raw Pixel-based	91.40	92.22	92.74	92.74
FAST + BRIEF	92.35	94.54	92.51	93.51
FAST + BRISK	82.80	87.13	82.32	84.66
FAST + HOG	96.83	97.16	97.16	97.16
CNN	93.97	92.61	96.45	94.49

The results of the five different feature descriptor techniques using NB classifier are shown in Table 3. It is clear from Table 3 that each of the retrieved image features generated a satisfactory accuracy value. Nevertheless, FAST+HOG provided the highest classification accuracy with a value of 96.83% as opposed to raw pixel-based, FAST+BRIEF, FAST+BRISK, and CNN with a classification accuracy of 91.40%, 92.35%, 82.80%, and 93.97%, respectively. Raw pixel, FAST+BRIEF, FAST+BRISK, and CNN all had lower precision values than FAST+HOG (97.16% vs. 92.22%, 94.54%, 87.13%, and 92.61%, respectively). When comparing the obtained recall values, the FAST+HOG has a high recall value of 97.16%, demonstrating that the proportion of correct positive predictions made from all the positive predictions is higher than that of the raw pixel-based, FAST+BRIEF, FAST+BRISK, and CNN, with recalls of 92.74%, 92.51%, 82.32%, and 96.45%, respectively. It is also clear from the recall value that FAST+BRISK had the least recall value. In terms of evaluation using the F1-score, CNN comes in second with an F1-score of 94.99%, followed by FAST+HOG with a score of 97.16%. A visual representation of NB's performance on each of the feature descriptors is provided in Figure 6.

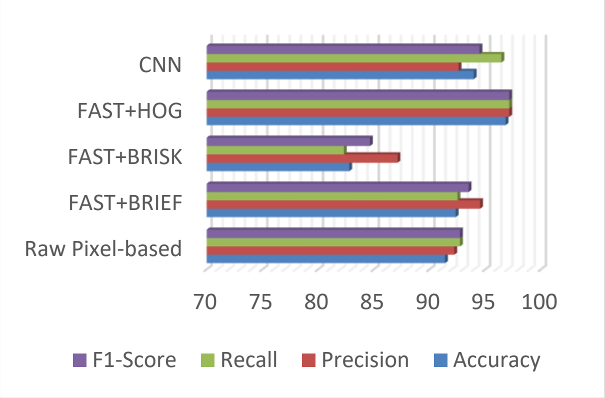

Figure 6 Comparison of CNN, HOG, BRISK, BRIEF and Raw Pixel for NB classification

The accuracy, recall, precision, and f-score for CNN, FAST + HOG, FAST + BRISK, FAST + BRIEF, and raw pixel-based using NB classifier are clearly displayed in Figure 6. Table 3 results show the values for the chart's accuracy, precision, recall, and f1-score.

Table 4 Comparison of feature descriptors based on Execution Time

Feature Extractors	Execution Time (Seconds)
Raw Pixel-based	52.36
FAST + BRIEF	46.04
FAST + BRISK	313.10
FAST + HOG	48.00
CNN	162.92

During the feature extraction the time taken for each of the five descriptors to extract the image features are displayed in Table 4. Visual representation of the execution time for each of the description is shown in figure 5

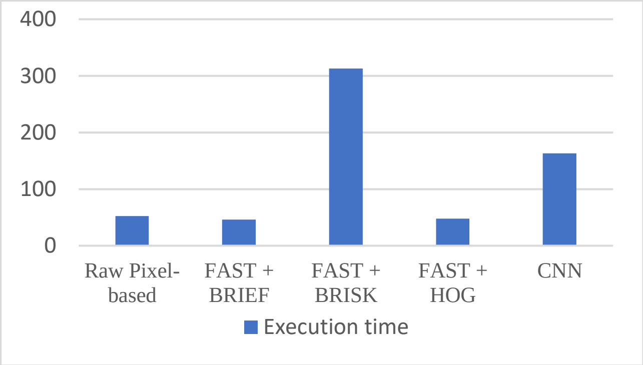

Figure 5 Comparison of CNN, HOG, BRISK, BRIEF and Raw Pixel for KNN classification

Comparing the various feature descriptors based on the execution, the FAST+BRIEF descriptor had the least execution time of 46.04 seconds, followed by FAST+HOG with 48.00 seconds. The FAST+BRISK descriptor took longer than the other descriptors when describing and extracting the image features. From the

Accuracy, Recall, precision, F1-Score, and execution time obtained, it can be concluded that the FAST+HOG descriptor is more appropriate for a reliable Face mask identification than the other four descriptors.

Table 5 Comparison with Previous Works

Methods	**Acc (%)**	**Prec (%)**	**Rec(%)**	**F1-Score (%)**	**Exec Time (Secs)**
Loey et al. [4]	99.27	99.27	99.27	99.27	-
Wang [16]	95				-
Proposed method	99.46	99.41	98.83	99.12	48.00

According to the accuracy values are Shown in table 5, the proposed FAST+HOG approach has the best performance, with 99.46%, whereas the methods proposed by Loey et al. [4] and Wang [16] have 99.27% and 95% accuracy, correspondingly.

Conclusion and Future Work

Face mask detection was performed more reliably in this study than in previous research on the subject. This is owing to the system's capability to detect face masks in photos with great accuracy, precision and speed. From this study, it can be concluded that applying the FAST algorithm for distinctive keypoints detection improves the speed of the various feature descriptors as these descriptors focus on the description of the identified keypoint features and ignore totally the other sets of features not identified by FAST. The FAST+HOG descriptor had the best performance with an accuracy of 99.46%, precision of 99.41%, recall of 98.83%, f1-score of 99.12% and execution time of 48 seconds using the SVM classification model. On the other hand, FAST+BRISK had the least performance with an accuracy of 91.40%, precision of 99.32%, recall of 84.88%, f1-score of 91.54% and execution time of 313.10 seconds using the same classification model. The result is not much different from that of using KNN classification model. When recall values are compared, the FAST+HOG has a high recall value of 97.22%, indicating that the ratio of correct positive assumptions made from all the true positives is higher than that of the raw pixel-based, FAST+BRIEF, FAST+BRISK, and CNN, with recalls of 90.06%, 95.04%, 92.49%, and 95.65%, respectively. The accuracy, precision and recall values also make it evident that FAST+BRISK had the lowest performance. FAST+HOG is ranked first with a f1-score of 98.31%, followed by CNN with an F1-score of 97.77% in terms of evaluation. The proposed metyhod also outperformed other moethods using the NB classification model in that the FAST+HOG has a high recall value of 97.16%, demonstrating that the proportion of correct positive predictions made from all the positive predictions is higher than that of the raw pixel-based, FAST+BRIEF, FAST+BRISK, and CNN, with recalls of 92.74%, 92.51%, 82.32%, and 96.45%, respectively. It is also clear from the recall value that FAST+BRISK had the least recall value. In terms of evaluation using the F1-score, CNN comes in second with an F1-score of 94.99%, followed by FAST+HOG with a score of 97.16%. In conclusion, a system was developed to perform face mask detection using cascaded bi-level feature extraction techniques for access restriction in public buildings.

Only the SVM, NB and KNN classification models were considered in this study. These are classic classification models that do not require large dataset to train. SVM handles outliers better, these classifiers are also easy to use in training and testing of dataset, hence their choice. To improve the system's robustness, further classification models, such as Decision tree, and discriminate analysis, can be employed or integrated in future study. In this work, only the Real-world Masked Face Dataset was considered for analysis. This dataset suffers from the problem of imbalanced data. For future work, the number of datasets used can be increased, and more face mask images can be added to the Real-world Masked Face Dataset to reduce the imbalance nature of the dataset.

References

[1] F. Shahid, A. Zameer, and M. Muneeb, “Predictions for COVID-19 with deep learning models of LSTM, GRU and Bi-LSTM,” Chaos Solitons Fractals, vol. 140, p. 110212, 2020, doi: 10.1016/j.chaos.2020.110212.

[2] X. Jiang, T. Gao, Z. Zhu, and Y. Zhao, “Real-time face mask detection method based on yolov3,” Electronics (Switzerland), vol. 10, no. 7, Apr. 2021, doi: 10.3390/electronics10070837.

[3] X. Fan and M. Jiang, “RetinaFaceMask: A Single Stage Face Mask Detector for Assisting Control of the COVID-19 Pandemic,” May 2020, [Online]. Available: http://arxiv.org/abs/2005.03950

[4] M. Loey, G. Manogaran, M. H. N. Taha, and N. E. M. Khalifa, “A hybrid deep transfer learning model with machine learning methods for face mask detection in the era of the COVID-19 pandemic,” Measurement (Lond), vol. 167, Jan. 2021, doi: 10.1016/j.measurement.2020.108288.

[5] M. K. Tripathi and D. D. Maktedar, “A role of computer vision in fruits and vegetables among various horticulture products of agriculture fields : A survey,” Information Processing in Agriculture, no. xxxx, 2019, doi: 10.1016/j.inpa.2019.07.003.

[6] M. M. Boulos, “Facial Recognition and Face Mask Detection Using Machine Learning Techniques,” 2021. [Online]. Available: https://digitalcommons.montclair.edu/etdhttps://digitalcommons.montclair.edu/etd/728

[7] Ejaz Sabbir and Islam Rabiul, “Masked Face Recognition Using Convolutional Neural Network,” in International Conference on Sustainable Technologies for Industry 4.0 (STI), 2019.

[8] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, “ORB: An efficient alternative to SIFT or SURF,” Proceedings of

the IEEE International Conference on Computer Vision, no. May, pp. 2564–2571, 2011, doi: 10.1109/ICCV.2011.6126544.

[9] E. Karami, S. Prasad, and M. Shehata, "Image Matching Using SIFT, SURF, BRIEF and ORB: Performance Comparison for Distorted Images," 2017, [Online]. Available: http://arxiv.org/abs/1710.02726

[10] E. Oyallon and J. Rabin, "An Analysis of the SURF Method," Image Processing On Line, vol. 5, no. 2004, pp. 176–218, 2015, doi: 10.5201/ipol.2015.69.

[11] B. B. Swapnali and K. S. Vijay, "Feature Extraction Using Surf Algorithm for Object Recognition," International Journal of Technical Research and Applications, vol. 2, no. 4, pp. 197–199, 2014, [Online]. Available: www.ijtra.com

[12] B. Zohuri, "Deep Learning Limitations and Flaws," Modern Approaches on Material Science, vol. 2, no. 3, Jan. 2020, doi: 10.32474/mams.2020.02.000138.

[13] J. Wang, Y. Yuan, and G. Yu, "Face Attention Network: An Effective Face Detector for the Occluded Faces," Computer Vision and Pattern Recognition , Nov. 2017, [Online]. Available: http://arxiv.org/abs/1711.07246

[14] A. Cabani, K. Hammoudi, H. Benhabiles, and M. Melkemi, "MaskedFace-Net – A dataset of correctly/incorrectly masked face images in the context of COVID-19," Smart Health, vol. 19, Mar. 2021, doi: 10.1016/j.smhl.2020.100144.

[15] Tomas Jesus, Rego Albert, Viciano-Tudela Sandra, and Lioret Jaime, "Incorrect Facemask-Wearing Detection Using Convolutional Neural Networks with Transfer Learning," Healthcare, vol. 9, no. 1050, 2021, doi: 10.3390/healthcare9081050.

[16] Z. Wang et al., "Masked Face Recognition Dataset and Application," Mar. 2020, [Online]. Available: http://arxiv.org/abs/2003.09093

[17] A. Alzu'bi, F. Albalas, T. Al-Hadhrami, L. B. Younis, and A. Bashayreh, "Masked face recognition using deep learning: A review," Electronics (Switzerland), vol. 10, no. 21. MDPI, Nov. 01, 2021. doi: 10.3390/electronics10212666.

[18] U. Scherhag, C. Rathgeb, J. Merkle, R. Breithaupt, and C. Busch, "Face Recognition Systems Under Morphing Attacks : A Survey," vol. 7, 2019.

[19] Y.-Q. Wang, "An Analysis of the Viola-Jones Face Detection Algorithm," Image Processing On Line, vol. 4, pp. 128–148, 2014, doi: 10.5201/ipol.2014.104.

[20] P. ; Viola and M. Jones, "Rapid Object Detection Using a Boosted Cascade of Simple Features," 2004. [Online]. Available: http://www.merl.com

[21] C. Saravanan, "Color image to grayscale image conversion," 2010 2nd International Conference on Computer Engineering and Applications, ICCEA 2010, vol. 2, no. April 2010, pp. 196–199, 2010, doi: 10.1109/ICCEA.2010.192.

[22] R. Verma, M. Rohit Verma, and J. Ali, "A comparative study of various types of image noise and efficient noise removal techniques," 2013. [Online]. Available: www.ijarcsse.com

[23] F. Iftikhar and J. Mohammed, "Algorithm for Image Processing Using Improved Eliminat ion of Gaussian Noise from FPGA Based Co-Processors IJRES Journal An Improved Median Filt er Based on Efficient Noise Det ect ion for High Qualit y Image Rest orat ion," 2011.

[24] H. Zhang, J. Wohlfeil, and D. Grießbach, "EXTENSION and EVALUATION of the AGAST FEATURE DETECTOR," ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. 3, pp. 133–137, 2016, doi: 10.5194/isprs-annals-III-4-133-2016.

[25] A. v Kulkarni, J. S. Jagtap, and V. K. Harpale, "Object recognition with ORB and its Implementation on FPGA," International Journal of Advanced Computer Research, no. 3, 2013.

[26] M. Ghahremani, Y. Liu, and B. Tiddeman, "FFD: Fast Feature Detector," IEEE Transactions on Image Processing, vol. 10, no. 10, Dec. 2020, doi: 10.1109/TIP.2020.3042057.

[27] T. Surasak, I. Takahiro, C. H. Cheng, C. E. Wang, and P. Y. Sheng, "Histogram of oriented gradients for human detection in video," Proceedings of 2018 5th International Conference on Business and Industrial Research: Smart Technology for Next Generation of Information, Engineering, Business and Social Science, ICBIR 2018, pp. 172–176, 2018, doi: 10.1109/ICBIR.2018.8391187.

[28] J. J. Priyankha and K. Suresh, "Crop Disease Identification Using a Feature Extraction HOG Algorithm," Asian Journal of Applied Science and Technology (AJAST), vol. 1, no. 3, pp. 35–39, 2017.

[29] C. Shu, X. Ding, and C. Fang, "Histogram of the oriented gradient for face recognition," Tsinghua Sci Technol, vol. 16, no. 2, pp. 216–224, 2011, doi: 10.1016/S1007-0214(11)70032-3.

[30] K. J. Sreelekshmi and T. Y. Mahesh, "Human Identification Based on the Histogram of Oriented Gradients," International Journal of Engineering Research & Technology (IJERT), vol. 3, no. 7, pp. 1611–1614, 2014.

[31] M. K. Benkaddour and A. Bounoua, "Feature extraction and classification using deep convolutional neural networks, PCA and SVC for face recognition," Traitement du Signal, vol. 34, no. 1–2, pp. 77–91, 2017, doi: 10.3166/TS.34.77-91.

[32] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, no. 7553, pp. 436–444, 2015, doi: 10.1038/nature14539.

[33] A. Ferreira and G. Giraldi, "Convolutional Neural Network approaches to granite tiles classification," Expert Syst Appl, vol. 84, pp. 1–11, 2017, doi: 10.1016/j.eswa.2017.04.053.

[34] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "BRIEF: Binary Robust Independent Elementary Features," 2010.

[35] M. Kashif, T. M. Deserno, D. Haak, and S. Jonas, "Feature description with SIFT, SURF, BRIEF, BRISK, or FREAK? A general question answered for bone age assessment," Comput Biol Med, vol. 68, pp. 67–75, Jan. 2016, doi: 10.1016/j.compbiomed.2015.11.006.

[36] Y. Liu, H. Zhang, H. Guo, and N. N. Xiong, "A FAST-BRISK Feature Detector with Depth Information," 2018, doi: 10.3390/s18113908.

[37] R. Englund, "Machine Learning for Technical Information Quality Assessment," no. March, 2016.

[38] P. Walsh, "Support Vector Machine Learning for ECG Classification," Smart Healthcare and Safety Systems, vol. 10, pp. 195–204, 2019.

[39] J. Cao, M. Wang, Y. Li, and Q. Zhang, "Improved support vector machine classification algorithm based on adaptive feature weight updating in the Hadoop cluster environment," pp. 1–12, 2020.

[40] S. Ghosh, "A Study on Support Vector Machine based Linear and Non-Linear Pattern Classification," no. Iciss, pp. 24–28, 2019.

[41] Y. Tang, "Deep Learning using Linear Support Vector Machines," 2013.

[42] V. Krishnaiah, G. Narsimha, and S. N. Chandra, "Heart Disease Prediction System Using Data Mining Technique by Fuzzy K-NN Approach," in Advances in Intelligent Systems and Computing, 2015, pp. 371–384. doi: 10.1007/978-3-319-13728-5.

[43] A. Kataria and M. D. Singh, "A Review of Data Classification Using K-Nearest Neighbour Algorithm," International Journal of Emerging Technology and Advanced Engineering, vol. 3, no. 6, pp. 354–360, 2013.

[44] H. Parvin, H. Alizadeh, and B. Minati, "A Modification on K-Nearest Neighbor Classifier," Global Journal of Computer Science and Technology, vol. 10, no. 14, pp. 37–41, 2010.

[45] J. Đ. Novakovic, A. Veljovic, and S. S. Ilic, "Experimental Study of using the K-Nearest Neighbour Classifier with Filter Methods," Computer Science and Technologies, no. 451, pp. 90–99, 2016.

[46] A. Sopharak et al., "Machine learning approach to automatic exudate detection in retinal images from diabetic patients," J Mod Opt, vol. 57, no. 2, pp. 124–135, 2010, doi: 10.1080/09500340903118517.

[47] D. Berrar, "Bayes ' Theorem and Naive Bayes Classifier Bayes ' Theorem and Naive Bayes Classifier," Encyclopedia of Bioinfor- matics and Computational Biology, no. January 2018, pp. 0–18, 2019, doi: 10.1016/B978-0-12-809633-8.20473-1.

[48] B. Harangi, B. Antal, and A. Hajdu, "Automatic exudate detection with improved naïve-Bayes classifier," Proc IEEE Symp Comput Based Med Syst, 2012, doi: 10.1109/CBMS.2012.6266341.