applied sciences

MDPI

Article

Unified Texture Descriptor in the Form of Color and Machine Learning, Applied to Face Identification

Jorge Aguilar-Santiago [1], Cuauhtémoc Acosta-Lúa [2], Ariadna-Berenice Flores-Jiménez [1], Carlos-Eduardo Padilla-Leyferman [1], Juan-Carlos Estrada-Gutiérrez [2], María-Eugenia Sánchez-Morales [2] and Maricela Jiménez-Rodríguez [1,*]

[1] Department of Basic Sciences, La Ciénega University Center, University of Guadalajara, Av. Universidad No. 1115, Lindavista, Ocotlán 47810, Jalisco, Mexico; jorge.asantiago@academicos.udg.mx (J.A.-S.); ariadna.flores@academicos.udg.mx (A.-B.F.-J.); carlos.pleyferman@cuci.udg.mx (C.-E.P.-L.)
[2] Department of Technological Sciences, La Ciénega University Center, University of Guadalajara, Av. Universidad No. 1115, Lindavista, Ocotlán 47810, Jalisco, Mexico; cuauhtemoc.acosta@academicos.udg.mx (C.A.-L.); jcarlos.estrada@academicos.udg.mx (J.-C.E.-G.); eugenia.sanchez@academicos.udg.mx (M.-E.S.-M.)
* Correspondence: maricela.jrodriguez@academicos.udg.mx

Abstract

Facial recognition is widely employed in identification systems where access restriction is a desirable quality, providing greater security and preventing the access of unauthorized parties. With this problem in mind, this paper proposes a new texture descriptor that works in the RGB, YCbCr and HSV color modes. It obtains perceptual descriptive data from the photographs; the descriptor considers similarities between neighboring pixels and diminishing noise interference, making it robust to minor disturbances and bringing greater stability between similar images. Furthermore, the descriptor achieves good classification performance when used with both a neural network (NN) and the statistical classifier Support Vector Machine (SVM) on color images. A performance comparison of the proposed descriptor against LBP and OC_LBP employing three different face databases is presented to demonstrate that the proposed descriptor can work with the RGB, YCbCr and HSV color modes, presenting the best results when employing a neural network as a classifier with color image databases in the three studied color modes.

Keywords: texture descriptors; SVM; neural networks; YCbCr; RGB; HSV; facial recognition

1. Introduction

Nowadays, facial recognition systems have become of great importance due to their application in person identification; additionally, they have become more employed in mobile devices, the authentication of students in virtual classes, employee identification, access authorization, etc. For facial recognition to be performed correctly, it is necessary to take the following into consideration: rotation and facial posture, facial expression, lighting, and contrast, etc. [1]. Image scale must also be taken into consideration; several papers propose novel techniques for facial recognition, where a genetic algorithm and SVM are deployed [2]. To perform facial recognition, it is necessary to have at least one descriptor to generate characteristic vectors, along with a classifier to identify or determine the class, face or category that the studied part of interest belongs to. Therefore, some papers propose descriptors where long videos are segmented into smaller ones, employing GLCM to extract the statistical data from the video, reducing its size. The recognition is performed through

Check for updates

Academic Editor: Andrea Prati

Received: 20 May 2026
Revised: 8 June 2026
Accepted: 11 June 2026
Published: 15 June 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

the Viola–Jones algorithm; face segments are extracted by implementing data mining [3]. In another study, the facial image is cut into different regions, left eye, right eye, nose and mouth; face regions are embedded in nodes. Considering the relation between two nodes, a Generative Adversarial Network (GAN) is implemented for facial identification; graphic convolutional neural networks are employed to detect the relation between facial regions [4]. Angular distance is calculated to find coincidences between two facial images. SVM and Similarity Metric Learning are deployed as classifiers to verify these coincidences, along with POEM, LBP, Monogenic Binary Coding, HOG, HASC, GOLD, RICLBP and CLBP as texture descriptors [5]. Characteristics are extracted from images stored in a database; the obtained texture characteristic vectors are placed in a repository that contains the color signature and data regarding the texture shape and measurements. The YCbCr color mode and GLCM texture descriptor are employed [6]. Two texture descriptors were proposed: Local Grouped Order Pattern (LGOP) and Non-Local Binary Pattern (NLBP). They are combined to build the histogram characteristics [7]. A technique for feature extraction in volumetric images was proposed to discriminate data. Vectors based on Local Binary Patterns and Gray Level Co-Occurrence Matrices are generated and fused; for classification, Support Vector Machine, k-Nearest Neighbors and Random Forest were employed [8]. A texture descriptor working under HSV color mode was designed; the bin number of the H channel is quantified. Votes are conducted with the saturation values and vice versa; furthermore, the relation of co-occurrence between the pixels is taken into consideration [9]. A texture descriptor was implemented, combining the representation of complex objects, employing a cellular automaton with local descriptors based on Local Binary Patterns [10]. Another related approach is RGB pixel N-grams where RGB textures are deployed and tested with three classifiers [11]. An algorithm was optimized to perform facial recognition, employing several hybrid-like texture descriptors and a convolutional neural network called AlexNet [12]. A texture descriptor was developed for facial recognition in images with controlled and uncontrolled environments, where lighting homogeneity was achieved employing histogram matching [13]. The combination of a 3D model, the LBP texture descriptor and a simple CNN was deployed to perform facial recognition [14]. Another paper implements the Local Binary Patterns (LBPs), Local Ternary Patterns (LTPs) and Complete Local Binary Patterns (CLBPs) texture descriptors in conjunction with a convolutional neural network to identify facial expressions in people [15]. Two novel texture descriptors were designed for a paper published in [16], ZigZag Binary Patterns (CZZBPs) and Color Median Binary Patterns (CMBZZBPs) where zigzag RGB data is taken to calculate medians; SVM and neural networks are deployed for classification purposes. A new algorithm was developed to optimize Local Binary Patterns (LBPs), where the data is segmented into three matrices, of which two are fixed to analyze and optimize through singular value decomposition [17]. A method employing 2D facial images was presented; Speeded-Up Robust Features (SURF) and Scaled-Invariant Feature Transform (SIFT) are deployed for feature extraction while Random Forest and Decision Tree are applied for classification [18]. To perform facial recognition efficiently, some challenges must be overcome beforehand, such as photographs taken in uncontrolled environments, poor lighting, and different poses; additionally, as a rule of thumb, texture descriptors have been designed to work with grayscale images or RGB color mode. Thus, others color modes such as HSV and YCbCr could be implemented. Skin tones were detected in RGB facial images of different ethnicities; OpenCV was employed to identify skin colors and recommend makeup bases; and HSV and YCbCr were used for color detection with delta-E [19]. In [20], an RGB image is taken as input and then converted into two color spaces to represent skin tones. The S channel from HSV is employed to detect eyes and nostrils, while lips are identified with YCbCr. In [21], a technique to detect attacks based on the HSV and YCbCr color spaces was

designed using a CNN that identifies the characteristics to perform classification. Several CNNs were tested (ResNet50, VGG16 and MobileNetV2) to detect if facial images are real or forged. In another study, image quality is taken into consideration. Low quality images are processed with a CNN in HSV mode; medium-quality images are processed through SIFT in RGB mode, and high-quality images with SURF in YCbCr mode. For classification, the Manhattan distance and the Nearest Neighbor classifier were employed for statistical analysis along with a CNN model [22]. An algorithm was proposed for facial detection considering the tones of human skin; RGB, HSV and YCbCr space colors are applied to generate a feature vector which is later employed for gender classification [23]. A facial recognition algorithm was designed where three new hybrid color spaces is constructed. These color spaces, namely YSCr, ZnSCr and BnSCr, were formed by employing the components Cb and Cr extracted from the YCbCr color space; the S component was extracted from HSV; and the Zn and Bn components were taken from the XYZ color space. These features are extracted and then fused together; the resulting vectors are classified with combinations of Machine Learning classifiers [24]. The YCbCr, HSV and Lab color modes are integrated for facial detection in difficult or obscured environments [25]. In another study, the HSV, YCbCr, RGB and L*a*b color modes were also applied for face detection and the YCbCr space achieved the highest degree of accuracy [26]. In [27], classifiers LDA, SVM and XGBoost were compared employing the color modes: RGB, HSV, YCbCr and CIE (Lab); based on this the optimal texture descriptor was selected and applied in face presentation attack detection. The HSV, RGB and YCbCr are widely deployed for facial detection considering skin tones or to mitigate phishing attacks; however, the papers limit exploiting the characteristics of a color space for texture description without proposing a technique that could perform description through multiple color spaces.

This paper proposes a simple and efficient texture descriptor that can be employed in different color modes with statistical classifiers and neural networks.

The highlights of this manuscript can be summed up as follows:

1. The descriptor can be deployed to identify any object of interest, such as faces, fingerprints, healthy or sick plants, and lacerations, etc.
2. The texture descriptor can operate in different color modes: RGB, YCbCr and HSV.
3. The texture descriptor is based in the identification of similarities between subpixels composing an observation window of 3×3 employing the logical operator AND.
4. The subpixel binary values are considered to find coincidences, making it robust to noise, representing the image structures with more detail and less invariance to lighting changes or small perturbances, extracting data of greater quality.
5. A comparison between the accuracy levels of the SVM classifier and the NN is presented.
6. The proposed descriptor was also compared against LBP and OC_LBP with three databases to determine the efficiency of the proposed descriptor with respect to others commonly used in the literature.
7. Performs image classification invariably in controlled and uncontrolled environments, with poses of varying degrees and different expressions.

This paper is structured in the following manner: Section 2 details the methodology employed in the correct operation of the proposed descriptor and how the classification techniques are carried over; Section 3 exhibits the obtained results from the different tests, an ablation study of the NN, confusion matrices and ROC curves; Section 4 details the conclusion reached in this paper; and last, but not least, the last Section lists the references considered to sustain the scientific evidence verifying the current state of the art.

2. Methodology

This paper proposes a novel texture descriptor for image classification in different color modes (RGB, YCbCr and HSV). Figure 1 illustrates the proposed process in this research: (1) an expert user organizes images or photographs of a subject or person according to their class before training; (2) every image of each class is processed; it is divided in the channels that conform it depending on the color mode of interest (RGB, YCbCr or HSV); (3) the proposed descriptor found in Section 2.3.3 is employed to generate (4) the feature vectors of the class and (5) the statistical classifier (SVM) and NN performs the classification; once training has been completed, a new image is classified, and, lastly, (6) it is decided to which class it belongs considering the requirements of the selected classifier.

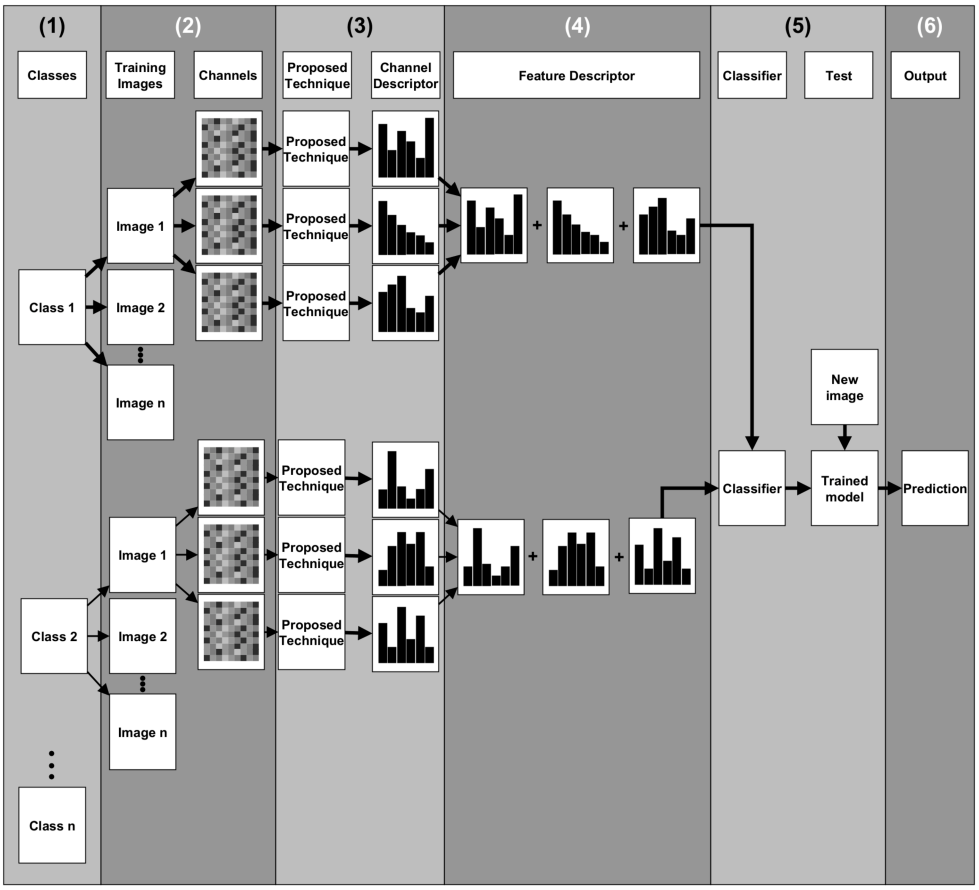


Figure 1. General operation for image identification.

2.1. Databases

2.1.1. The Georgia Tech (GT)

Database [28] was deployed, an open source dataset consisting of photographs with the following technical specifications: 15 color images in JPEG color with a 640 × 480 per subject of 50 different people, giving a total of 750 images; the photographs were taken with different lighting conditions and scale, against a cluttered background, in two or three different sessions; the images present frontal faces, sometimes tilted or angled and with different facial expressions. Figure 2 presents a digital collage with a sample of the photographs found in the chosen database.

One of the most common challenges found in facial recognition is having enough images for training; more images result in better accuracy [29]. With this in mind, this paper deploys the Data Augmentation technique to increase the quantity of images in the database; this tool applies a series of transformations to the images—rotation, mirroring,

and brightness adjustments, etc. Data Augmentation was deployed to perform more robust training to ensure the model operated correctly in uncontrolled environments.



Figure 2. Images from the Georgia Tech face database [28].

2.1.2. FEI

This is a database crafted by the Artificial Intelligence Laboratory of FEI in São Bernardo do Campo, São Paulo, Brazil. In total, 14 images were taken of 50 individuals, and then Data Augmentation was applied to obtain 192 images for each class, resulting in a total of 9600 images; the images were re-dimensioned to a 320 × 240 resolution [30].

2.1.3. YALE B

The extended Yale Face Database B contains images of 28 individuals in 9 different poses and 64 different lighting conditions [31]; these images were also re-dimensioned to 320 × 240.

2.2. Data Augmentation

The procedure to generate new images consists of taking an image of each class, which is employed as base; from it, another image is generated applying a mirroring technique; two additional images are obtained by rotating the original image and the mirror 180 degrees in opposite poles; afterwards, three more images are procured varying the brightness levels; to conclude this process, from the three images with varying brightness, the other three images are obtained, a mirrored one and two rotated 180 degrees in opposite directions. This process is showcased in Figure 3.

Once the Data Augmentation procedure has been completed with the Georgia Tech database, the samples increased from 750 to 12,000 images, from which 11,200 were employed for training and 800 for testing.

2.3. Texture Descriptors

2.3.1. LBPs (Local Binary Patterns)

Each element of the 3 × 3 observation window holds an integer value p, where $P = [p1, p2, p3, p4, p5, p6, p7, p8, pC]$, and pC denotes the central pixel; this texture descriptor compared every neighbor element in the window with the central pixel pC; if it is equal or greater, a 1 value is assigned to the position of the neighbor it is being compared with; otherwise, a 0 is assigned instead. Figure 4 illustrates the process performed by LBP,

generating an 8-bit binary. This binary is turned into its corresponding decimal, obtaining a value between 0 and 255 which is stored in the feature vector. For this paper, the descriptor was adapted to generate a 256-element vector for each color channel, i.e., 256 for channel R, 256 for channel G and 256 for channel B; therefore, the resulting characteristic vector bears 768 elements [32].

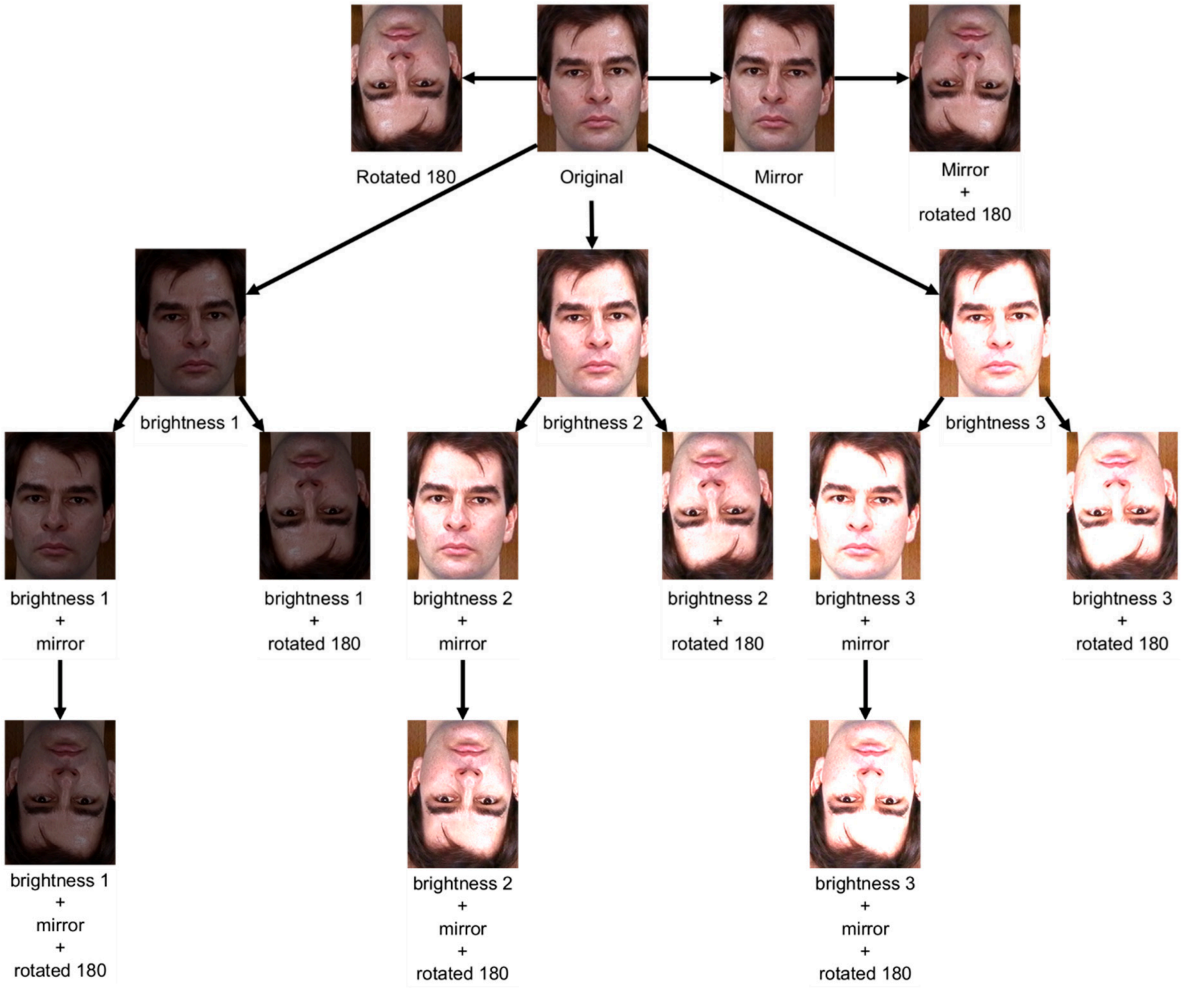


Figure 3. Images generated through the Data Augmentation database GT.

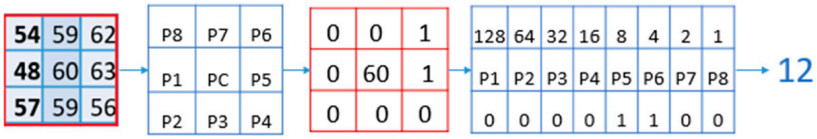


Figure 4. LBP descriptor operational scheme.

2.3.2. OC_LBP (Orthogonal Combination of Local Binary Patterns)

A 3 × 3 observation window is taken, composed of p = [*p*1, *p*2, *p*3, *p*4, *p*5, *p*6, *p*7, *p*8, *p*C], where pC represents the central pixel, which is compared with its *p*n neighbors. If the neighbor pn ≥ pC, a 1 is assigned; otherwise, it will return 0. The components are divided into 2: V1 = [*p*1, *p*3, *p*5, *p*7] and V2 = [*p*2, *p*4, *p*6, *p*8]. A dimensional space of R^{16} is obtained for each component, V1 and V2, which are fused, and henceforth, the resulting feature vector consists of R^{32}. This procedure is presented in Figure 5.

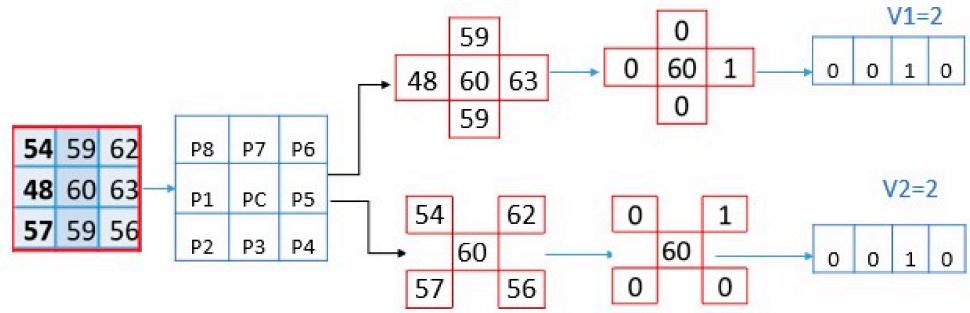


Figure 5. OC_LBP descriptor operational procedure.

2.3.3. Proposed Texture Descriptor

The process used to generate the vector that represents the textural characteristics of the image under study is explained below. Said vector has a texture space of 768 positions regardless of the chosen color space where the first 256 positions represent the intensity corresponding to channel Y (YCbCr), R (RGB) or H (HSV), depending on the color mode. The following 256 positions correspond to Cb, G or S and the last 256 positions conform to Cr, B or V. This is illustrated in Figure 6.

The operation of the proposed texture descriptor is explained below.

Phase 1. Data extraction.

1. Enter the input image, *Image*, and transform it to any of the YCbCr, RGB or HSV color modes to generate three color matrices or sub-images, one for each channel. The color space to be used is selected:

$$ChR,\ ChG,\ ChB\ =\ ImageRGB$$
$$or$$
$$ChY,\ ChCb,\ ChCr\ =\ ImageYCbCr$$
$$or$$
$$ChH,\ ChS,\ ChV\ =\ ImageHSV$$

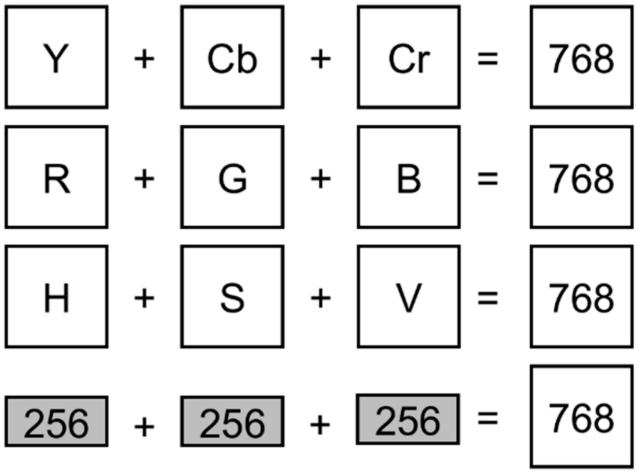


Figure 6. Feature vectors dimensions.

2. Employ a 3 × 3 observation window to traverse the subpixels of the sub-image or matrix of each channel; this process can be observed in Figure 7.

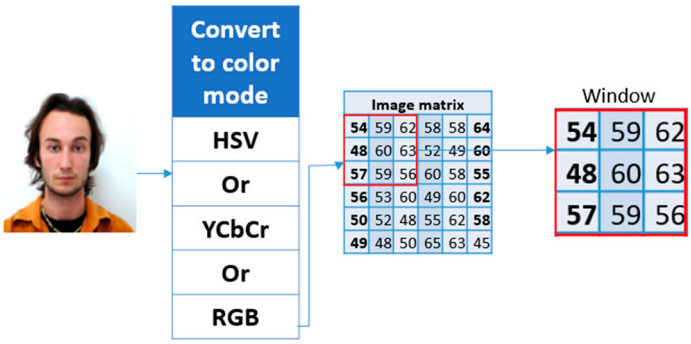


Figure 7. Sub-image pathing with a 3 × 3 observation window.

3. Two data groups of texture information are obtained from the observation window: the first group considers the data in a plus shape (+), while the second group considers the data in a cross shape (x), as illustrated in Figure 8.

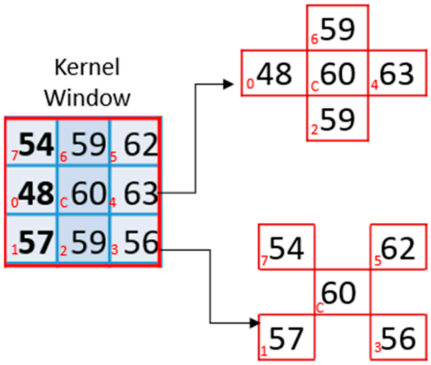


Figure 8. Observation window, data extraction in a cross or plus shape.

4. Obtain the subpixel values of the observation window in a plus shape, considering the subpixels *P*0, *P*2, *P*4 and *P*6 (see Figure 9).

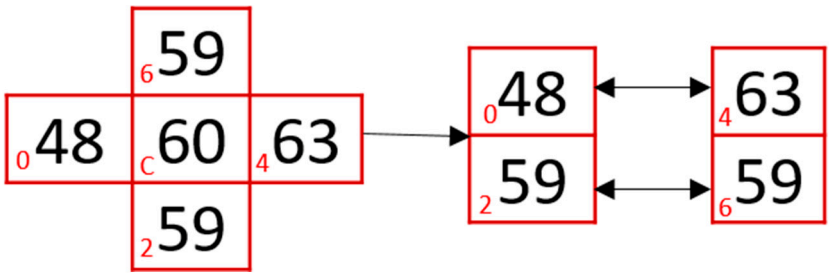


Figure 9. Plus-shape texture data.

5. Convert the pixel values *P*0 and *P*4 to their respective binary value, then apply an AND operation between them; the resulting value is stored in Plus0; see Figure 10.

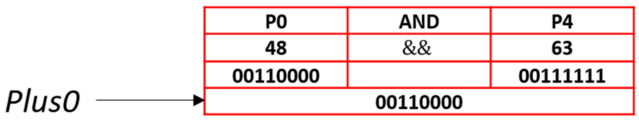


Figure 10. Obtaining Plus0.

6. Convert the pixel values *P*2 and *P*6 to their respective binary value, then apply an AND operation between them; the resulting value is stored in Plus1; see Figure 11.

P2	AND	P6
59	&&	59
00111011		00111011
00111011		

Plus1 →

Figure 11. Obtaining Plus1.

7. Determine the number of bits turned on in both *Plus*0 and *Plus*1; the results are stored in *BitsP*0 and *BitsP*1 respectively. See Figure 12.

Plus0	Bits ON
00110000	2

← *BitsP0*

Plus1	Bits ON
00111011	5

← *BitsP1*

Figure 12. Determining the number of bits in BitsP0 and BitsP1.

8. Convert *Plus*0 and *Plus*1 to a decimal to generate integer values between 0 and 255. See Figure 13.

Plus0	Plus1
00110000	00111011
48	59

Figure 13. Converting Plus0 and Plus1 into integers.

9. In case *BitsP*0 is equal to *BitsP*1, i.e., they have the same number of lit bits, the decimal values of *Plus*0 and *Plus*1 are compared and the greater decimal value is selected instead.

$$If\ BitsP0 == BitsP1$$

$$If\ Plus0 > Plus1$$

$$BestPlus = Plus0$$

$$If\ Plus1 > Plus0$$

$$BestPlus = Plus1$$

$$If\ Plus0 == Plus1$$

$$BestPlus = Plus0$$

10. In case *Plus*0 and *Plus*1 are different, *BestPlus* stores the greater value. See Figure 14.

Plus0=48	Plus1=59
highest value	59

→ *BestPlus*

Figure 14. Selecting *BestPlus*.

11. Collect the integer values of the *P*1, *P*3, *P*5 and *P*7 subpixels, considering the window values in a cross shape; see Figure 15.

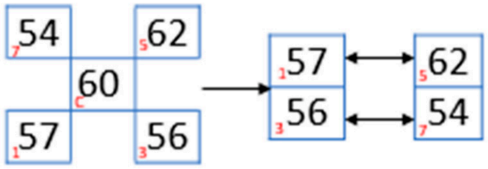


Figure 15. Extract texture data in a cross shape.

12. Convert the pixel values, *P*1 and *P*5, to their respective binary value, then apply an AND operation between them; the resulting value is stored in Cross = 0; see Figure 16.

P1	AND	P5
57	&&	62
00111001		00111110
Cross0 →	00111000	

Figure 16. Selecting the value of Cross0.

13. Convert the pixel values, *P*3 y *P*7, to their respective binary value, then apply an AND operation between them; the resulting value is stored in *Cross* = 1; see Figure 17.

P3	AND	P7
56	&&	54
00111000		00110110
Cross1 →	00110000	

Figure 17. Selecting the value of Cross1.

14. Determine the number of bits turned on in both, *Cross*0 and *Cross*1; the resulting values are stored in *Bits*C0 and *Bits*C1 respectively; see Figure 18.

Cross0	Bits ON
00111000	3

← *BitsC0*

Cross1	Bits ON
00110000	2

← *BitsC1*

Figure 18. Obtaining the total number of bits turned on.

15. Convert *Cross*0 and *Cross*1 from binary to decimals to generate integer values between 0 and 255; see Figure 19.

Cross0	Cross1
00111000	00110000
56	48

Figure 19. Converting *Cross*0 and *Cross*1 to integers.

16. In case *Bits*C0 is equal to *Bits*C1, i.e., they have the same number of lit bits, the decimal values of *Cross*0 and *Cross*1 are compared and the greater decimal value is selected instead.

$$If\ BitsC0 == BitsC1$$

$$If\ Cross0 > Cross1$$

$$BestCross = Cross0$$

$$If\ Cross1 > Cross0$$

$$BestCross = Cross1$$

$$If\ Cross0 == Cross1$$

$$BestCross = Cross0$$

17. If *Cross*0 and *Cross*1 are different, *BestCross* receives the greatest value. See Figure 20.

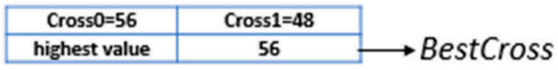


Figure 20. Greater value stored in Cross.

18. Determine the number of bits turned on in both *BestPlus* and *BestCross*; the results are stored in *bitsBestPlus* and *bitsBestCross*; see Figure 21.

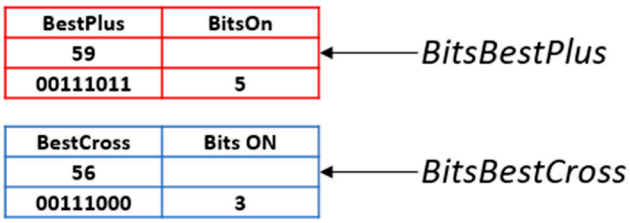


Figure 21. Obtaining the number of bits turned on for *BestPlus* and *BestCross*.

19. The number of bits turned on is compared to select the greater value between *bitsBestPlus* and *bitsBestCross*; if *bitsBestPlus* > *bitsBestCross*, then *Best* = *BestPlus*. Otherwise, if *bitsBestCross* > *bitsBestPlus*, then *Best* = *BestCross*; see Figure 22.
20. If *bitsBestPlus* is equal to *bitsBestCross*, i.e., they have the same number of lit bits, the decimal values of *BestPlus* and *BestCross* are compared and the greater decimal value is selected instead.

$$If\ bitsBestPlus\ == bitsBestCross$$

$$If\ BestPlus\ > BestCross$$

$$Best = BestPlus$$

$$If\ BestCross\ > BestPlus$$

$$Best = BestCross$$

$$If\ BestPlus\ == BestCross$$

$$Best = BestPlus$$

21. Increase the value found in position *Best* of the features vector.

$$Features_v[Best] = Features_v[Best] + 1$$

22. Repeat steps 2 to 21 until all values representing each observation window for the respective channels (Y, R or H; Cb, G or S; and Cr, B or V) are generated, according to the previously selected color mode.

A feature vector with 768 elements is generated for the chosen color mode (see Figure 23). When the H, Y or R channel is studied, the values from positions 0 to 255 are modified; when it is the S, Cb or G channel, the values from 256 to 511 are modified; and when it is the V, Cr or B channel the positions 512 to 767 are modified instead.

A pseudocode of Algorithm 1 is presented:

Algorithm 1: Pseudo-code Descriptor

```
Input: ImageRGB, ImageYCbCr, ImageHSV
Output: Features vector
Begin
Repeat
        Obtain P0, P1, P2, P3, P4, P5, P6, P7 from the image observation window.
        Convert to binary P0, P1, P2, P3, P4, P5, P6, P7
        Plus0 = P0 AND P4; Plus1 = P2 AND P6; Cross0 = P1 AND P5; Cross1 = P3 AND P7
        Set bits to BitsP0 = Plus0, BitsP1 = Plus1, BitsC0 = Cross0, BitsC1 = Cross1
        Convert to decimal Plus0, Plus1
        If BitsP0 > BitsP1 Then BestPlus = Plus0
                Else BestPlus = Plus1
        End If
        If BitsP0 == BitsP1 Then
                If Plus0 > Plus1 Then BestPlus = Plus0
                  Else BestPlus= Plus1
                End If
           If Plus0 == Plus1 Then BestPlus = Plus0
           End If
        End If
        Convert to decimal Cross0, Cross1
        If BitsC0 > BitC1 Then BestCross = Cross0
              Else BestCross = Cross1
        End If
        If BitsC0 == BitsC1 Then
                If Cross0 > Cross1 Then BestCross = Cross0
                 Else BestCross = Cross1
           End If
           If Cross0 == Cross1 Then BestCross = Cross0
                End If
        End If
        Convert to binary BestPlus, BestCross
        Set bits to BitsBestPlus = BestPlus, BitsBestCross = BestCross
        If BitsBestPlus > BitsBestCross Then Best = BestPlus
                Else Best = BestCross
        End If
        If BitsBestPlus == BitsBestCross Then
                If BestPlus > BestCross Then Best = BestPlus
                  Else Best = BestCross
           End If
           If BestPlus == BestCross Then Best = BestPlus
                End If
        End If
        Feature_c[Best] = feature_c[Best] + 1
Until Scan the entire image using the observation window
Return feature_c
End
```

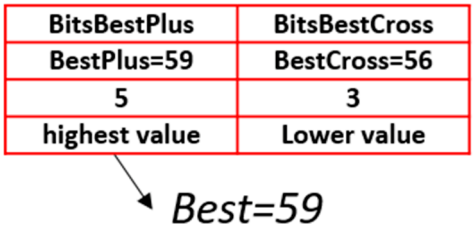


Figure 22. Selecting the greatest value to store in the feature vector.

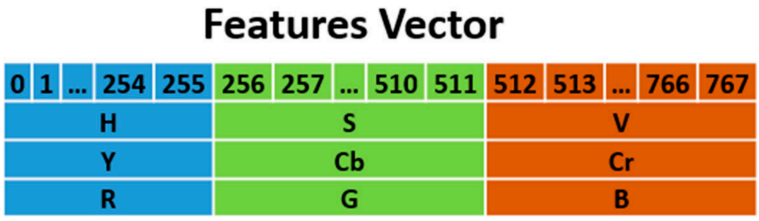


Figure 23. Features vector.

2.3.4. Neural Network Training Method

1. A human expert selects and groups images according to the subjects' faces, and then they are grouped by classes, i.e., for each subject a different *cl* class is defined.

$$cl_faces = \{cl1,\ cl2,\ cl3, \ldots,\ cln\}$$

2. Select the color mode (HSV, YCbCr or RGB) to employ.
3. Employ the texture descriptor in 2.3.3 to generate the vector with the features of a photograph corresponding to a particular class.

$$Feature_v = cln$$

4. Repeat step 3 until all *features_v* corresponding to the photographs of a particular class are generated.
5. Repeat steps 3 and 4 until all *Feature_v* vectors are generated for all *cl_faces* of the classes or subjects found in the database to train an NN.
6. Store the parameters of the complete neural network.

2.3.5. Neural Network Classification Method

1. Input a new facial image for authentication.
2. Select the color mode (HSV, YCbCr or RGB) to employ.
3. Apply the proposed feature descriptor found in Section 2.3.3 to generate its characteristic vector, $feature_{test}$.
4. Place the characteristic vector of the user to authenticate, $feature_{test}$, as an input for the trained NN.
5. The trained neural network has an output layer that identifies each of the users or classes that the neural network was trained with. Therefore, when entering the data of the vector $feature_{test}$ of a new person, the model returns the results of all output neurons, and the neuron with the highest value surpassing an established threshold must be the class or person corresponding to the data. This threshold is configured according to the obtained data; for example, if the data is under controlled conditions, or if it must be adjusted, each neuron has an output with a value of 0 to 1. A zero value indicates the input does not coincide with the class (not authenticated), and a one indicates a perfect match. The result is compared against the threshold; if one of

the neurons passes the threshold, it is taken as the authentication of the corresponding user of the eigenvector; if two or more neurons pass the threshold, the higher value must be chosen.

6. Finally, the class to which the image of the user to be authenticated belongs is indicated; otherwise, a "user not authenticated" screen is displayed.

2.3.6. Statistical Classification Method Through SVM

This type of classifier allows classification to be performed in a statistical manner considering probability to assign a class to a set of pattern features. The main advantages are simplicity and computer efficiency, although its performance depends on adjusting the classifier parameters correctly and the quality of the input data [33] [34]. SVM is a type of supervised statistical classifier, amply deployed due to its high accuracy to find the optimal hyperplane to separate classes; it employs just a subset of the training data, called support vectors [33].

SVM was implemented for image classification; each class cl consists of a set of images, i.e., for each image in a cl class, a characteristic vector is generated and represented by x, where $x_i \in \Re^n$; a label, $y_i \in \{-1, 1\}$, is assigned to it, where $i = 1, \ldots, l$; the hyperplane is defined by (1).

$$w * x + b = 0 \tag{1}$$

From each given pair (w, b), the image x_i can be abstracted considering function (2)

$$f(x_i) = sign(w * x + b) = \begin{Bmatrix} 1\ y_i = 1 \\ -1\ y_i = -1 \end{Bmatrix} \tag{2}$$

where y_i indicates the class the image belongs to [35].

3. Results

The texture descriptor proposed in Section 2.3.3 was implemented and coded in the programming language Python 3.12. Additionally, the neural network belonging to the Pytorch library was employed for classification and SVM as a statistical classifier with the Sklearn library. The tests were carried out in an ASUS ProArt PX13 computer with the following specs: AMD Ryzen™ AI 9 HX 370 processor, integrated NPU with up to 50 TOPS for artificial intelligence, integrated AMD Radeon™ 890M graphics with a NVIDIA® GeForce RTX 4060 dedicated graphics card, 32 GB of RAM LPDDR5X, 1TB SSD NVMe PCIe 4.0 and Windows 11 Home operative system.

3.1. Evaluation Metrics

This paper determines the performance of a proposed neural network considering the true positives (TPs), false positives (FPs), true negatives (TNs) and false negatives (FNs) [36,37], measuring the following metrics:

Accuracy calculates the proportion of total samples classified correctly:

$$Accuracy = \frac{TN + TP}{TP + TN + FP + FN} \tag{3}$$

Precision is employed to determine the true positive predictions:

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

Recall measures the number of real positive cases that were accurate [38]:

$$Recall = \frac{TP}{TP + FN} \quad (5)$$

F1-score considers precision and recall to determine the balance between the two metrics, generating a value between 0 and 1, where 1 indicates greater performance [39]:

$$F1 - score = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (6)$$

Table 1 exhibits a comparison of the proposed descriptor against other related works found in the literature. According to the results, the neural network presents a greater performance in the three color modes, except with the YALE B database due to the grayscale images.

Table 1. Comparison of evaluation metrics.

Dataset	Descriptor	Classifier	Accuracy	Precision	Recall	F1-Score	References
YALEB	Proposed YCBCR	NN	27.78	58.63	26.79	31.33	Proposed
YALEB	Proposed RGB	NN	29.46	64.69	29.46	34.75	Proposed
YALEB	Proposed HSV	NN	25.00	50.86	24.11	27.13	Proposed
FEI	Proposed YCBCR	NN	99.88	99.88	99.88	99.87	Proposed
FEI	Proposed RGB	NN	99.00	99.10	99.00	98.99	Proposed
FEI	Proposed HSV	NN	100	100	100	100	Proposed
GT	Proposed YCbCr	NN	97.50	97.90	97.50	97.47	Proposed
GT	Proposed RGB	NN	92.50	94.64	92.50	92.42	Proposed
GT	Proposed HSV	NN	97.50	98.13	97.50	97.41	Proposed
AT&T	LBPH		98 ± 1.24	0.97 ± 0.015	0.98 ± 0.012	0.97 ± 0.015	[40]
Casia	LBP	NLBP-Net	99.593	99.057	99.224	99.057	[41]
Jaffe	LTCP	Random Forest	96.13				[42]
CK+	LTCP	Random Forest	97.52				[42]
Pins Face Recognition	CNN	CNN	91.296	92.035	91.296	91.304	[43]
LFW		FaceNet	0.9407	0.9410	0.9403	0.9406	[44]
CALFW		ArcFace	0.8130	0.8129	0.8129	0.9129	[44]

Table 2 presents the performance of the neural network employed as a classifier with the Yale B database [31]; a comparative study of the proposed descriptor against LBP and OC_LBP was performed. The images found in the Yale B database have 64 different lighting conditions and are only available in grayscale, causing the accuracy degrees to be lower for all descriptors; in this case LBP grants the greater results in classification.

Table 3 shows the results obtained by the neural network with the three aforementioned descriptors, but employing the FEI image database instead [30]. It can be observed that the proposed descriptor delivers greater results for the three color modes compared to the other two descriptors.

Table 4 presents the results of a comparative study of the three descriptors with the neural network employing the Georgia Tech database [28]. It can be clearly appreciated that the proposed descriptor bears greater results when compared with LBP and OC_LBP.

Table 2. Comparison of NN with Yale B, employing proposed descriptor, LBP and OC_LBP.

Dataset	Descriptor	Accuracy	Precision	Recall	F1-Score
YALEB	PROPOSED YCBCR	27.78	58.63	26.79	31.33
	PROPOSED RGB	29.46	64.69	29.46	34.75
	PROPOSED HSV	25.00	50.86	24.11	27.13
	LBP YCBCR	45.60	55.65	43.93	43.58
	LBP RGB	45.14	55.29	43.53	42.12
	LBP HSV	43.98	55.42	42.41	42.10
	OC_LBP YCBCR	28.94	50.24	27.90	26.91
	OC_LBP RGB	27.31	41.30	26.34	23.38
	OC_LBP HSV	25.00	44.31	24.11	22.21

Table 3. Comparison of NN with FEI, employing the proposed descriptor, LBP and OC_LBP.

Dataset	Descriptor	Accuracy	Precision	Recall	F1-Score
FEI	PROPOSED YCBCR	99.88	99.88	99.88	99.87
	PROPOSED RGB	99.00	99.10	99.00	98.99
	PROPOSED HSV	100	100	100	100
	LBP YCBCR	94.38	95.44	94.38	94.07
	LBP RGB	92.62	94.40	92.62	92.53
	LBP HSV	95.50	96.26	95.50	95.41
	OC_LBP YCBCR	97.00	97.63	97.00	96.95
	OC_LBP RGB	94.38	95.81	94.38	94.20
	OC_LBP HSV	95.00	95.95	95.00	95.00

Table 4. Comparative of NN, Georgia Tech, employing the proposed descriptor, LBP and OC_LBP.

Dataset	Descriptor	Accuracy	Precision	Recall	F1-Score
GT	PROPOSED YCBCR	97.50	97.90	97.50	97.47
	PROPOSED RGB	92.50	94.64	92.50	92.42
	PROPOSED HSV	97.50	98.13	97.50	97.41
	LBP YCBCR	82.12	82.39	82.12	80.35
	LBP RGB	72.88	73.74	72.88	69.56
	LBP HSV	83.38	84.40	83.38	81.66
	OC_LBP YCBCR	89.50	88.24	89.50	87.51
	OC_LBP RGB	82.62	82.27	82.62	79.35
	OC_LBP HSV	91.12	91.66	91.12	89.55

Table 5 displays the classification results with SVM and the three descriptors, employing the Yale B Database [31]. Generally, all descriptors present a low performance, but the proposed descriptor, when employing RGB, yields the best results.

In Table 6, the classification results of SVM and the three descriptors employing the FEI database can be observed [30]. Parameters are low for all deployed descriptors; the proposed descriptor yields a greater degree of accuracy with YCbCr and HSV.

Table 7 presents the classification results obtained with SVM and the three descriptors, deploying the Georgia Tech database [28]. It can be observed that the proposed descriptor offers a greater degree of accuracy in the color modes under study.

Table 5. Comparative of SVM with Georgia Tech, employing the proposed descriptor, LBP and OC_LBP.

Dataset	Descriptor	Accuracy	Precision	Recall	F1-Score
YALEB	PROPOSED YCBCR	7.41	5.39	7.41	3.91
	PROPOSED RGB	21.43	24.78	21.43	19.43
	PROPOSED HSV	7.41	5.39	7.41	3.91
	LBP YCBCR	3.70	0.15	3.70	0.28
	LBP RGB	20.32	24.36	19.83	15.62
	LBP HSV	3.70	0.15	3.70	0.28
	OC_LBP YCBCR	3.70	0.28	3.70	0.52
	OC_LBP RGB	10.65	9.13	10.65	8.15
	OC_LBP HSV	3.70	0.28	3.70	0.52

Table 6. Comparative of SVM with FEI, employing the proposed descriptor, LBP and OC_LBP.

Dataset	Descriptor	Accuracy	Precision	Recall	F1-Score
FEI	PROPOSED YCBCR	68.12	74.78	68.12	67.59
	PROPOSED RGB	29.12	28.96	29.12	26.57
	PROPOSED HSV	69.88	75.67	69.88	69.63
	LBP YCBCR	7.00	7.19	7.00	4.87
	LBP RGB	5.97	9.97	5.79	4.69
	LBP HSV	5.81	5.95	5.69	4.13
	OC_LBP YCBCR	8.25	3.87	8.25	4.69
	OC_LBP RGB	6.90	13.24	6.80	5.25
	OC_LBP HSV	7.38	5.28	7.38	4.86

Table 7. Comparative of SVM with GT, employing the proposed descriptor, LBP and OC_LBP.

Dataset	Descriptor	Accuracy	Precision	Recall	F1-Score
GT	PROPOSED YCBCR	98.00	98.53	98.00	97.92
	PROPOSED RGB	88.00	89.35	88.00	86.10
	PROPOSED HSV	97.50	98.13	97.50	97.16
	LBP YCBCR	29.25	29.45	29.25	25.36
	LBP RGB	43.30	55.76	44.82	40.45
	LBP HSV	34.00	33.55	34.00	29.24
	OC_LBP YCBCR	35.12	35.23	35.12	32.47
	OC_LBP RGB	50.88	53.06	50.88	45.79
	OC_LBP HSV	42.25	39.04	42.25	37.35

3.2. Ablation Test

To evaluate the performance and efficiency of the neural network, an ablation study was conducted, in which the operation of the proposed neural network was analyzed under different characteristics. The ablation process is presented in Figure 24; a detailed explanation of how this procedure was performed can be found below:

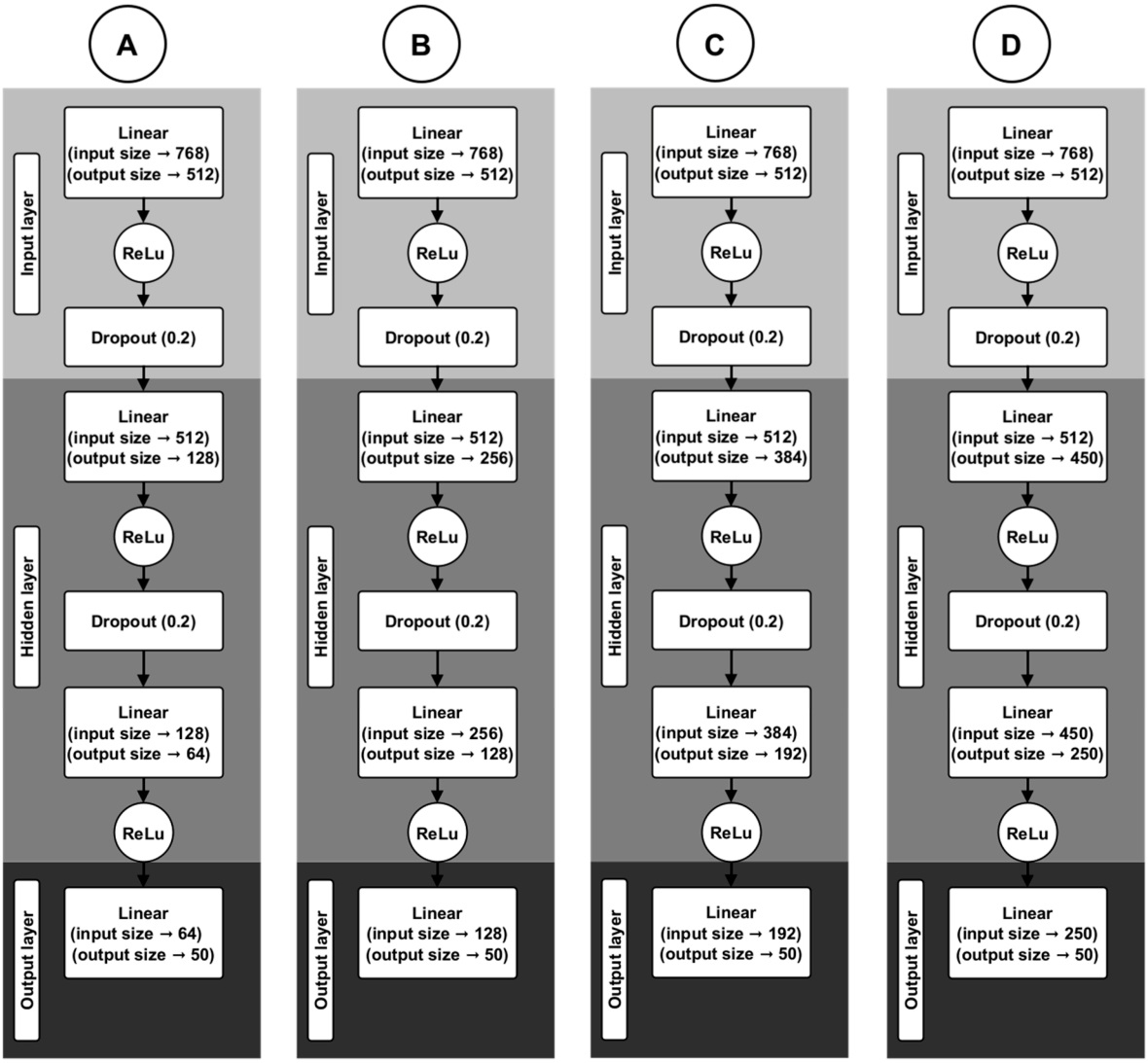


Figure 24. Ablation study by neuron.

Input layer

As an input layer for training, which was adapted to have 768 neurons corresponding to the dimensional space of the feature vector generated by the proposed descriptor (Section 2.3.3), the ReLU (Rectified Linear Unit) activation function was employed; afterwards, a Dropout layer of 0.2 was applied to prevent overfitting and improve the model. Finally, the output of the input layer delivers data from 512 neurons.

Hidden layers

The first lineal hidden layer receives 512 neurons, which will be reduced for its output considering four options: (A) 128 neurons, (B) 256 neurons, (C) 384 neurons, and (D) 450 neurons for last. The data is processed by other ReLU activation functions and Dropout layers.

The second lineal layer receives the number of layers according to the aforementioned options and reduces them further down—(A) 64 neurons, (B) 128 neurons, (C) 192 neurons and (D) 250 neurons—employing another ReLU activation function.

Output layers

Finally, the output layer receives the neurons filtered according to options (A), (B), (C) or (D) as input, processing them for an additional lineal layer; all options were configured with 50 output neurons, given there are 50 different classes.

This aforementioned process is illustrated in Figure 24 [45,46].

The results obtained with the ablation study employing the RGB color mode, in accordance with the different test options, are exhibited in Table 8. It can be observed the best option is B, obtaining an accuracy of 95%.

Table 8. Accuracy levels obtained from the ablation study.

Number of Hidden Neurons								
Test	(A)		(B)		(C)		(D)	
Neurons in each hidden layer	128	64	256	128	384	192	450	250
Accuracy	91.50%		95%		93%		91%	

Figure 25 presents the ablation study by incrementing the number of hidden layers to determine which configuration of the neural network showcases the best performance.

Table 9 presents the accuracy in accordance with the number of layers. As can be observed, the best performance was obtained with two hidden layers in the neural network. Based on these results, the neural network structure was implemented with one input layer, three hidden layers and one output layer.

Table 9. Ablation study according to the number of layers.

Number of Hidden Layers	Accuracy
2	93.5%
3	99%
4	93%
5	91.5%

According to the results of the ablation study, the neural network was deployed with the following architecture (see Figure 26): the input layer consists of 768 neurons, according to the size of the feature vector serving as input which is generated considering the data obtained by the proposed descriptor found in Section 2.3.3; afterwards, the data is processed by three hidden layers, the first with 512 neurons, the second with 256 neurons and the third with 128 neurons; for last, the output layer which size depends of the database employed for the study. In this case, the output layer consists of 50 neurons because the database of Georgia Tech, which has 50 classes, was applied.

The neural network was deployed with the Adam optimizer applying a learning rate of 0.001 and the ReLU activation function.

Table 10 presents a comparison of the state of the art vs. the tests performed with our proposed descriptor (Section 2.3.3) and the NN. The database FEI, the structure that obtained the best results (see Table 2), was applied to the RGB, YCbCr and HSV color modes, along with the results obtained with our descriptor (Section 2.3.3) and the SVM classifier. The training times for the NN with each color mode were YCbCr = 23.34 s,

RGB = 25.03 s and HSV = 27.97 s; for the SVM classifier the times were YCbCR = 6.6927 s, RGB = 10.02 s and HSV = 6.6649 s.

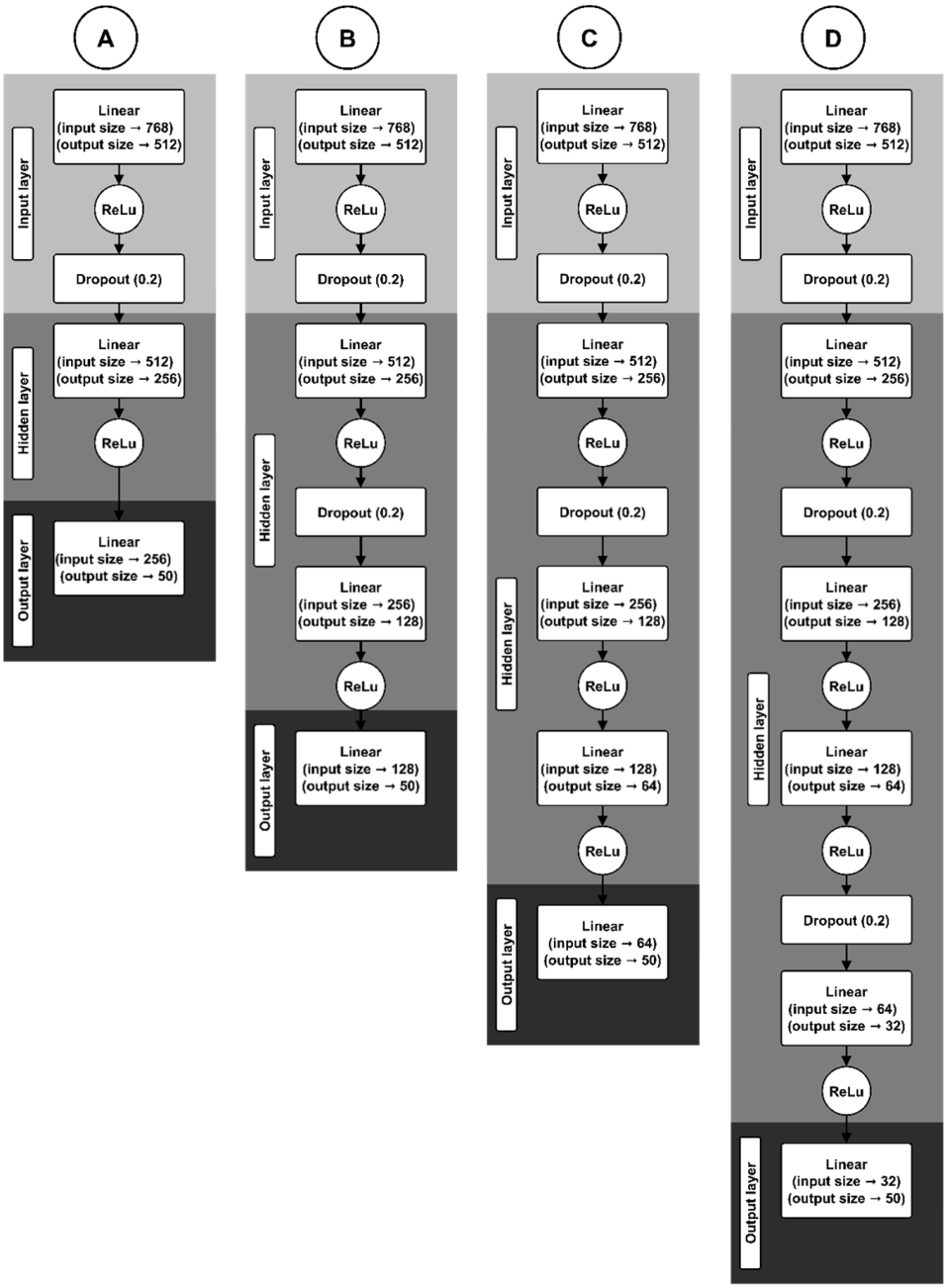


Figure 25. Ablation study by layers.

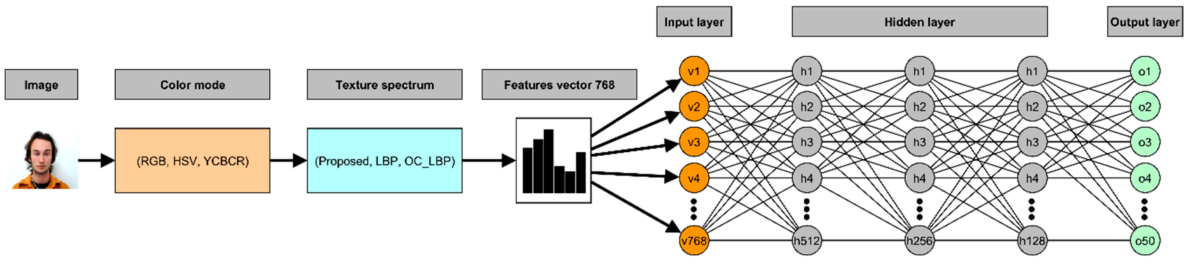


Figure 26. Structure of the neural network.

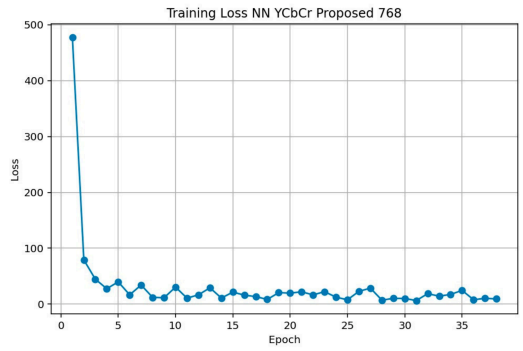
Table 10. Comparison between the proposed descriptor and the state of the art.

Reference	Classifier	Color Mode	Accuracy
Proposed	Proposed descriptor/Neural network	YCbCr	99.88%
Proposed	Proposed descriptor/Neural network	RGB	99%
Proposed	Proposed descriptor/Neural network	HSV	100%
Proposed	Proposed descriptor/SVM	YCbCr	68.12%
Proposed	Proposed descriptor/SVM	RGB	29.12%
Proposed	Proposed descriptor/SVM	HSV	68.88%
[47]	VGG-Face CNN/K- nearest neighbor	RGB	96.5%
[47]	VGG-Face CNN/SVM	RGB	94.3%
[48]	LBP/CNN	RGB	90%
[49]	Haar/LBPH (Local Binary Pattern histogram)	RGB	96%

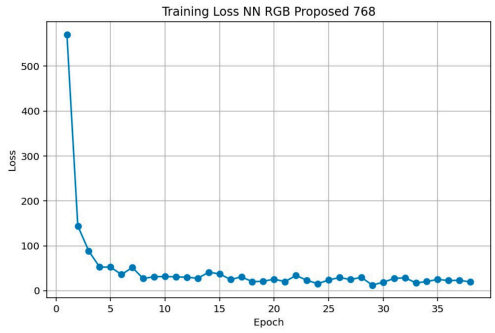
As can observed in Table 10, the proposed descriptor performs better with the three color modes when employing NN as classifiers.

3.3. Loss Function

To determine the proper operation of the neural network, implement the proposed model, and prevent overfitting, it is essential to study the loss process of the NN *vs* the number of epochs. Figure 27a–c exhibit the loss functions *vs* the epochs for the different color modes with the descriptor (Section 2.3.3) and the NN. Figure 27d–f display the function with LBP and the NN; Figure 27g–i display the loss functions of the NN with the OC_LBP descriptor. The horizontal axis presents the complete iterations over the training set, while the vertical axis represents the loss function, i.e., the model error is quantified according to the epochs. At the beginning, the loss function decreases rapidly; the lower the loss function value of the NN, the more accurate the predictions are, but if the value is high, the performance is deficient. For this research, 38 epochs were used because it was determined that the loss functions performed best with this parameter to guarantee a homogeneous configuration.

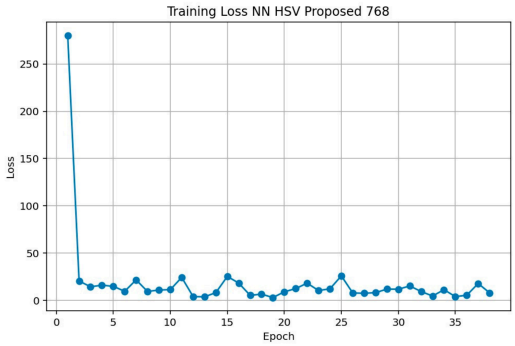


(**a**)

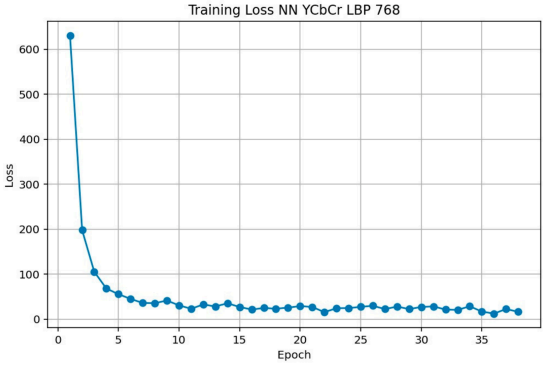


(**b**)

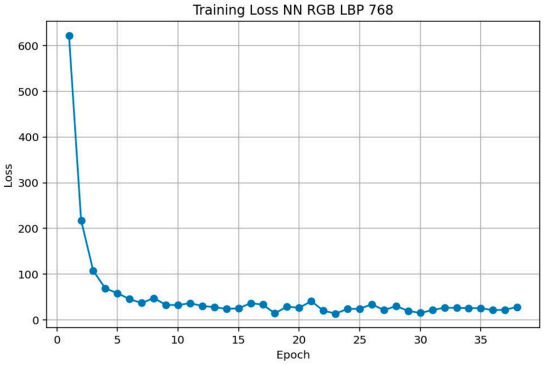
Figure 27. *Cont.*



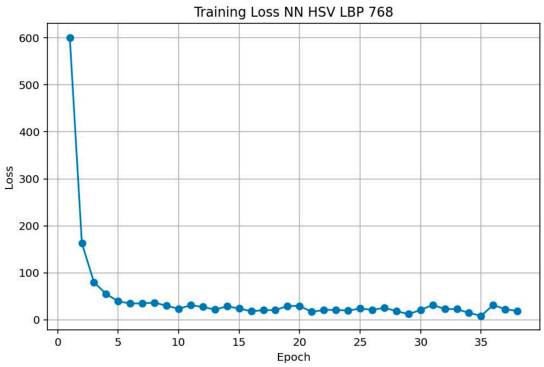
(**c**)



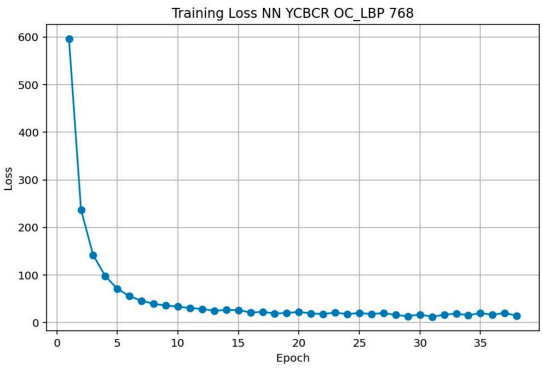
(**d**)



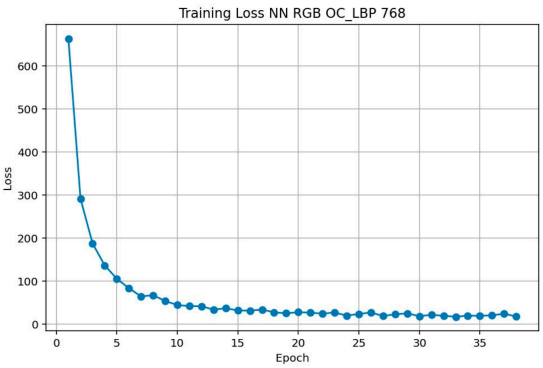
(**e**)



(**f**)

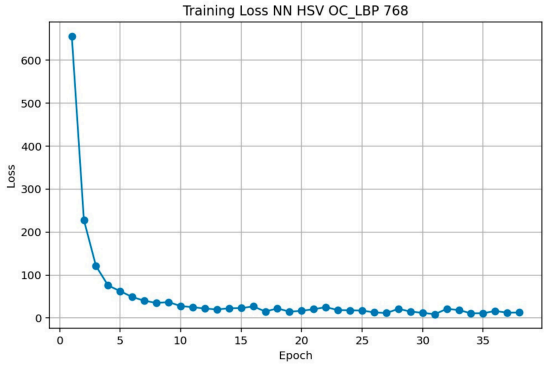


(**g**)



(**h**)

Figure 27. *Cont.*



(**i**)

Figure 27. Loss functions: (**a**) Proposed descriptor with YCbCr, (**b**) proposed descriptor with RGB, (**c**) proposed descriptor with HSV, (**d**) LBP descriptor with YCbCr, (**e**) LBP descriptor with RGB, (**f**) descriptor LBP with HSV, (**g**) OC_LBP descriptor with YCbCr, (**h**) OC_LBP descriptor with RGB, (**i**) OC_LBP descriptor with HSV.

3.4. Confusion Matrices

The correct classification can be observed through these matrices; the elements correctly identified can be found through the main diagonal while the elements outside of it were not identified in the correct class. Figure 28a–f showcase the results obtained from the classification tests employing the FEI database; it can be observed that the three color modes with a neural network classifier present an accuracy of 99% to 100% with the proposed description; on the other hand, the accuracy degree decreases considerably when SVM is deployed with this database, yielding better results with YCbCr and HSV.

3.5. ROC Curves (Receiver Operating Characteristic)

This test observes a comparison between the true positive rate and the false positive rate with a measurement unit known as the Area Under the ROC Curve (AUC). The AUC oscillates with a value between 0 and 1, where an AUC of 0.5 signals a performance akin to a random classifier, while an AUC greater than 0.5 indicates a superior performance than that of a random classifier. The best possible case is an AUC of 1, representing a perfect performance of the classification model, while an AUC of 0 indicates the worst possible performance. Figure 29a–c presents the ROC curves obtained with the proposed descriptor in the RGB, YCbCr and HSV color modes; all tests were carried out with an NN as a classifier deploying the FEI database.

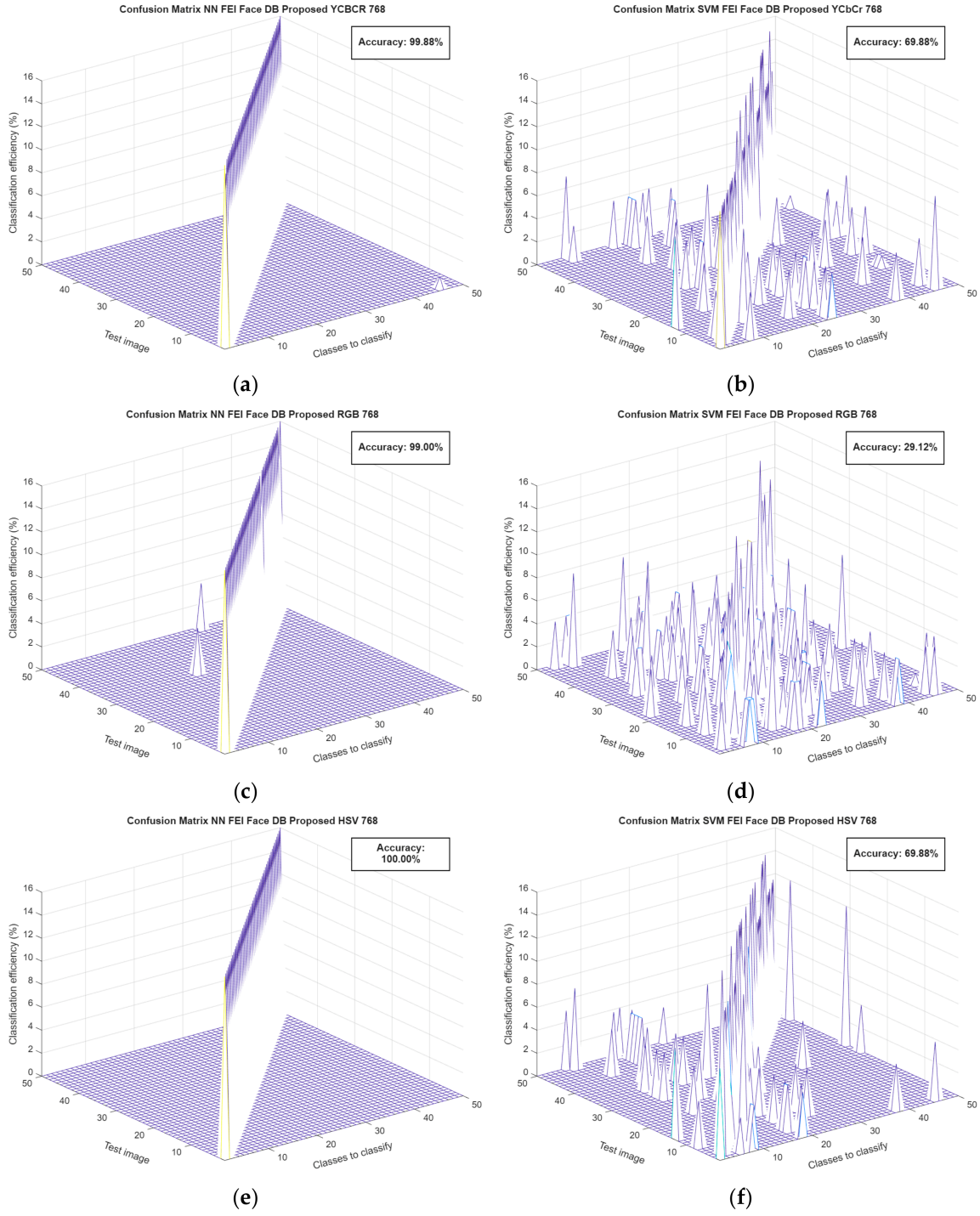


Figure 28. Confusion matrices using the FEI Database: (**a**) Proposed descriptor, YCbCr and NN; (**b**) proposed descriptor, YCbCr and SVM; (**c**) proposed descriptor, RGB and NN; (**d**) proposed descriptor, RGB and SVM; (**e**) proposed descriptor, HSV and NN; (**f**) proposed descriptor, HSV and SVM.

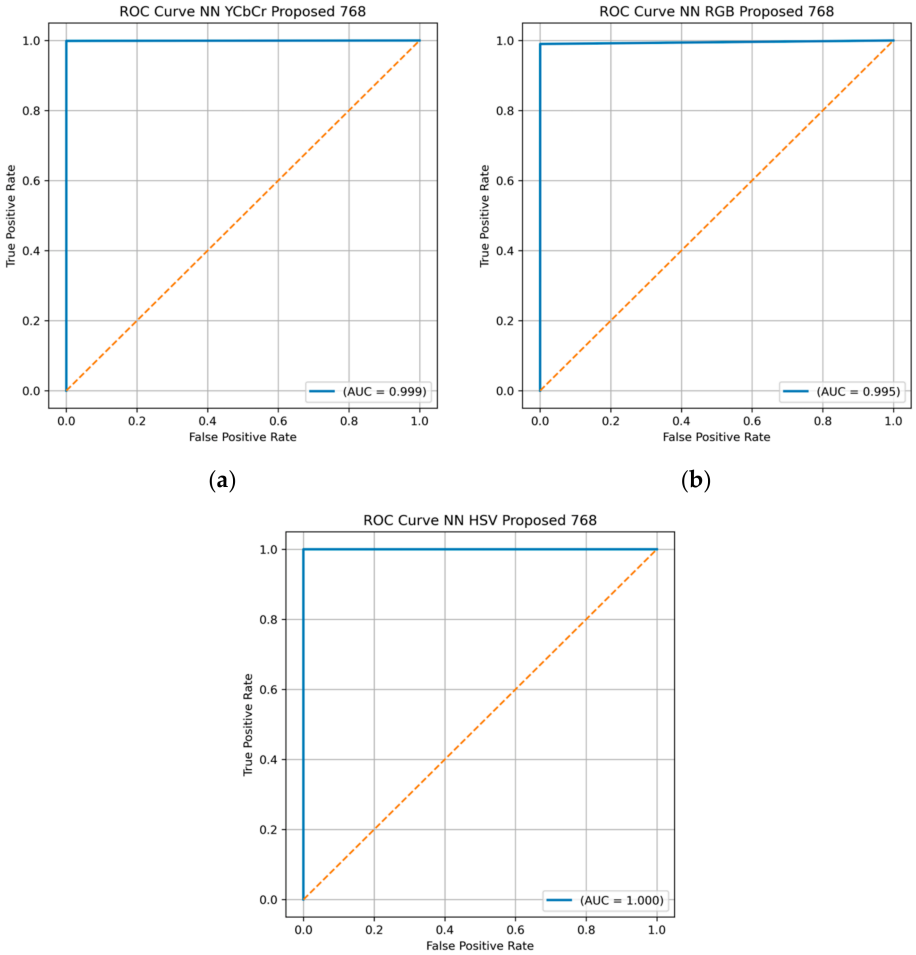


Figure 29. ROC Curve with FEI database: (**a**) Proposed descriptor with RGB; (**b**) proposed descriptor with YCbCr; (**c**) proposed descriptor with HSV.

4. Conclusions

The descriptor considers the subpixel values that bear the most resemblance, reduces noise generated by small variances between neighbor pixels, wields a greater focus in shapes, and, not in isolated pixels, discriminates irrelevant patterns more efficiently. The descriptor works in all three color modes, YCbCr, HSV and RGB. According to the results obtained of a comparative between our proposed descriptor and LBP and OC_LBP, employing NN and SVM as classifiers, it was determined that LBP when paired with NN works better with the Yale B database; our descriptor yielded better results with the Yale B database when matched with a SVM classifier; the results returned were generally low due to the grayscale composition of the images and the 64 lighting degrees. With the FEI database our descriptor performed better than LBP and OC_LBP with both an NN and SVM as classifiers, reaching an accuracy of 99% to 100% with the NN; although, with the SVM, the accuracy ranks were low for the three descriptors even though our proposed descriptor still returned the best results when employing the YCbCr color mode. In conclusion, the Georgia Tech database yielded the best results for our descriptor in the three color modes ranging from 92.5% to 97.5% for the NN and between 88% and 97.5% for the SVM. The proposed descriptor has a wide range of practical applications: biometric security, self-diagnosis, product quality control, crop disease detection, to mention some, and many other areas related to computer vision. For future work, it is proposed to evaluate the

descriptor with other databases of images with different conditions in both controlled and uncontrolled environments, employing medical and plant photographs.

Author Contributions: J.A.-S., M.J.-R., J.-C.E.-G., C.A.-L. and A.-B.F.-J. proposed the methodology and formal analysis; A.-B.F.-J., J.A.-S. and M.J.-R. developed the software; A.-B.F.-J., C.-E.P.-L. and M.J.-R. developed the numerical experiment, and M.J.-R., J.-C.E.-G. and M.-E.S.-M. performed the results analysis. All authors wrote the article. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in this study are included in the article.

Acknowledgments: The authors thank Mexico's Secretary of Science, Humanities, Technology and Innovation (SECIHTI) and the University of Guadalajara (UdeG) for the support granted.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript.

NN	Neural Network
SVM	Support Vector Machine
LBP	Local Binary Pattern
ROC	Receiver Operating Characteristic
AUC	Area Under the Curve
GT	Georgia Tech
OC_LBP	Orthogonal Combination of Local Binary Patterns

References

1. Li, L.; Mu, X.; Li, S.; Peng, H. A Review of Face Recognition Technology. *IEEE Access* **2020**, *8*, 139110–139120. [CrossRef]
2. Zhi, H.; Liu, S. Face recognition based on genetic algorithm. *J. Vis. Commun. Image Represent.* **2019**, *58*, 495–502. [CrossRef]
3. Khan, A.R.; Harouni, M.; Sharifi, S.; Bahaj, S.A.; Saba, T. Face Detection in Close-up Shot Video Events Using Video Mining. *J. Adv. Inf. Technol.* **2023**, *14*, 160–167. [CrossRef]
4. Luo, M.; Cao, J.; Ma, X.; Zhang, X.; He, R. FA-GAN: Face Augmentation GAN for Deformation-Invariant Face Recognition. *IEEE Trans. Inf. Forensics Secur.* **2021**, *16*, 2341–2355. [CrossRef]
5. Lumini, A.; Nanni, L.; Brahnam, S. Ensemble of texture descriptors and classifiers for face recognition. *Appl. Comput. Inform.* **2017**, *13*, 79–91. [CrossRef]
6. Alsmadi, M.K. Content-Based Image Retrieval Using Color, Shape and Texture Descriptors and Features. *Arab. J. Sci. Eng.* **2020**, *45*, 3317–3330. [CrossRef]
7. Song, T.; Feng, J.; Luo, L.; Gao, C.; Li, H. Robust Texture Description Using Local Grouped Order Pattern and Non-Local Binary Pattern. *IEEE Trans. Circuits Syst. Video Technol.* **2021**, *31*, 189–202. [CrossRef]
8. Barburiceanu, S.; Terebes, R.; Meza, S. 3D Texture Feature Extraction and Classification Using GLCM and LBP-Based Descriptors. *Appl. Sci.* **2021**, *11*, 2332. [CrossRef]
9. Bhunia, A.K.; Bhattacharyya, A.; Banerjee, P.; Roy, P.P.; Murala, S. A novel feature descriptor for image retrieval by combining modified color histogram and diagonally symmetric co-occurrence texture pattern. *Pattern Anal. Appl.* **2020**, *23*, 703–723. [CrossRef]
10. Florindo, J.B.; Metze, K. A cellular automata approach to local patterns for texture recognition. *Expert Syst. Appl.* **2021**, *179*, 115027. [CrossRef]
11. Pavón, F.B.P.; Gil, M.C.O.; Noguera, J.L.V.; Gómez-Adorno, H.; Calzada-Ledesma, V. RGB pixel n-grams: A texture descriptor. *Signal Process. Image Commun.* **2023**, *118*, 117028. [CrossRef]
12. Elaggoune, H.; Belahcene, M.; Bourennane, S. Hybrid descriptor and optimized CNN with transfer learning for face recognition. *Multimed. Tools Appl.* **2022**, *81*, 9403–9427. [CrossRef]
13. Leyferman, C.E.P.; Bonilla, J.T.G.; Gutiérrez, J.C.E.; Rodríguez, M.J. A novel technique for texture description and image classification based in RGB compositions. *IET Commun.* **2023**, *17*, 1162–1176. [CrossRef]

14. Bahroun, S.; Abed, R.; Zagrouba, E. Deep 3D-LBP: CNN-based fusion of shape modeling and texture descriptors for accurate face recognition. *Vis. Comput.* **2023**, *39*, 239–254. [CrossRef]
15. Mukhopadhyay, M.; Dey, A.; Kahali, S. A deep-learning-based facial expression recognition method using textural features. *Neural Comput. Appl.* **2023**, *35*, 6499–6514. [CrossRef]
16. Karanwal, S.; Diwakar, M. Two novel color local descriptors for face recognition. *Optik* **2021**, *226*, 166007. [CrossRef]
17. Sedaghatjoo, Z.; Hosseinzadeh, H.; Bigham, B.S. Bigham, Local Binary Pattern(LBP) Optimization for Feature Extraction. *arXiv* **2024**, arXiv:2407.18665. [CrossRef]
18. Gupta, S.; Thakur, K.; Kumar, M. 2D-human face recognition using SIFT and SURF descriptors of face's feature regions. *Vis. Comput.* **2021**, *37*, 447–456. [CrossRef]
19. Kamble, S.; Muntean, C.H.; Simiscuka, A.A. A Hybrid HSV and YCrCb OpenCV-based Skin Tone Recognition Mechanism for Makeup Recommender Systems. In Proceedings of the 2024 International Wireless Communications and Mobile Computing (IWCMC), Ayia Napa, Cyprus, 27–31 May 2024; pp. 1224–1229. [CrossRef]
20. Liang, T.-M.; Chiu, M.-S.; Wu, Y.-C.; Yeh, M.-T.; Hsu, C.-H.; Chung, Y.-N. Applying Image Processing Technology to Face Recognition. *J. Inf. Hiding Multimed. Signal Process.* **2022**, *13*, 107–112.
21. Perdana, R.A.; Hindratno, M.N.; Kamil, A.S.; Juliansyah, M.R.; Kusumajaya, R.; Hamdani, M. Face Presentation Attack Detection using Color Spaces Features and Convolutional Neural Network, en. In Proceedings of the 2022 6th International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE), Yogyakarta, Indonesia, 13 December 2022; pp. 115–120. [CrossRef]
22. Pazouki, M.M.; Toygar, Ö.; Hosseinzadeh, M. A Hybrid Approach for Color Face Recognition Based on Image Quality Using Multiple Color Spaces. *Sak. Univ. J. Comput. Inf. Sci.* **2024**, *7*, 361–377. [CrossRef]
23. Dogan, D.; Acarman, T. Skin Color-Based Frontal Face Detection and Gender Classification. In Proceedings of the 2024 10th International Conference on Control, Decision and Information Technologies (CoDIT), Split, Croatia, 1–4 July 2024; pp. 1305–1309. [CrossRef]
24. Uçar, A. Color Face Recognition Based on Steerable Pyramid Transform and Extreme Learning Machines. *Sci. World J.* **2014**, *2014*, 1–15. [CrossRef]
25. Yadav, A.; Sudhamsu, G.; Kaur, N.; Jagtap, J.; Chohan, M.K.; Ramudu, M.; Kapila, I.; Alkhayyat, A. Face Detection Technique for Challenging Conditions Using Multi-color Model Integration. *Natl. Acad. Sci. Lett.* **2025**, 1–6. [CrossRef]
26. Nugroho, H.A.; Goratama, R.D.; Frannita, E.L. Face recognition in four types of colour space: A performance analysis. *IOP Conf. Ser. Mater. Sci. Eng.* **2021**, *1088*, 012010. [CrossRef]
27. Du, Y.; Qiao, T.; Xu, M.; Zheng, N. Towards Face Presentation Attack Detection Based on Residual Color Texture Representation. *Secur. Commun. Netw.* **2021**, *2021*, 1–16. [CrossRef]
28. Ara Nefian Face Recognition Page. Available online: https://www.anefian.com/research/face_reco.htm (accessed on 12 January 2026).
29. Park, S.-H.; Ahn, S.-Y.; Lee, S.-W. Deep Feature Retention Module Network for Texture Classification. *Appl. Sci.* **2024**, *14*, 4011. [CrossRef]
30. FEI. [En línea]. Available online: https://fei.edu.br/ cet/facedatabase.html?utm_source=chatgpt.com (accessed on 5 June 2026).
31. Georghiades, A.S.; Belhumeur, P.N.; Kriegman, D.J. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Trans. Pattern Anal. Mach. Intell.* **2001**, *23*, 643–660. [CrossRef]
32. Aguilar-Santiago, J.; Guillen-Bonilla, J.T.; García-Ramírez, M.A.; Jiménez-Rodríguez, M. Identification of Lacerations Caused by Cervical Cancer through a Comparative Study among Texture-Extraction Techniques. *Appl. Sci.* **2023**, *13*, 8292. [CrossRef]
33. Khaparde, C. Sentiment Analysis Using SVM Classifier in Data Mining: A Machine Learning Approach. *Int. J. Comput. Sci. Inf. Technol.* **2025**, *24*, 23.
34. Alnuaimi, A.F.A.H.; Albaldawi, T.H.K. Concepts of statistical learning and classification in machine learning: An overview. *BIO Web Conf.* **2024**, *97*, 00129. [CrossRef]
35. An, Y.; Li, H.; Su, T.; Wang, Y. Determining Uncertainties in AI Applications in AEC Sector and their Corresponding Mitigation Strategies. *Autom. Constr.* **2021**, *131*, 103883. [CrossRef]
36. Al-Dabbas, H.M.; Azeez, R.A.; Ali, A.E. Two Proposed Models for Face Recognition: Achieving High Accuracy and Speed with Artificial Intelligence. *Eng. Technol. Appl. Sci. Res.* **2024**, *14*, 13706–13713. [CrossRef]
37. Arshed, M.A.; Mumtaz, S.; Ibrahim, M.; Dewi, C.; Tanveer, M.; Ahmed, S. Multiclass AI-Generated Deepfake Face Detection Using Patch-Wise Deep Learning Model. *Computers* **2024**, *13*, 31. [CrossRef]
38. Kanakala, S.; Ningappa, S. Detection and Classification of Diseases in Multi-Crop Leaves using LSTM and CNN Models. *J. Innov. Image Process.* **2025**, *7*, 161–181. [CrossRef]
39. Sujon, K.M.; Hassan, R.; Choi, K.; Samad, M.A. Accuracy, precision, recall, f1-score, or MCC? empirical evidence from advanced statistics, ML, and XAI for evaluating business predictive models. *J. Big Data* **2025**, *12*, 268. [CrossRef]

40. Ahsan, M.M.; Li, Y.; Zhang, J.; Ahad, M.T.; Gupta, K.D. Evaluating the Performance of Eigenface, Fisherface, and Local Binary Pattern Histogram-Based Facial Recognition Methods under Various Weather Conditions. *Technologies* **2021**, *9*, 31. [CrossRef]
41. Jaswanth, P.; Chowdary, P.Y.; Ramprasad, M.V.S. Deep learning based intelligent system for robust face spoofing detection using texture feature measurement. *Meas. Sens.* **2023**, *29*, 100868. [CrossRef]
42. Arya, R.; Vimina, E.R. Local Triangular Coded Pattern: A Texture Descriptor for Image Classification. *IETE J. Res.* **2023**, *69*, 3267–3278. [CrossRef]
43. Najeeb, M.A.; Alariyibi, A. Imbalanced Dataset Effect on CNN-Based Classifier Performance for Face Recognition. *Int. J. Artif. Intell. Appl.* **2024**, *15*, 25–41. [CrossRef]
44. Zhalgas, A.; Amirgaliyev, B.; Sovet, A. Robust Face Recognition Under Challenging Conditions: A Comprehensive Review of Deep Learning Methods and Challenges. *Appl. Sci.* **2025**, *15*, 9390. [CrossRef]
45. ReLU—PyTorch 2.9 Documentation. Available online: https://docs.pytorch.org/docs/stable/generated/torch.nn.ReLU.html (accessed on 25 December 2025).
46. Dropout—PyTorch 2.9 Documentation. Available online: https://docs.pytorch.org/docs/stable/generated/torch.nn.Dropout.html (accessed on 25 December 2025).
47. Moustafa, A.A.; Elnakib, A.; Areed, N.F.F. Age-invariant face recognition based on deep features analysis. *Signal Image Video Process.* **2020**, *14*, 1027–1034. [CrossRef]
48. Zhu, Y.; Jiang, Y. Optimization of face recognition algorithm based on deep learning multi feature fusion driven by big data. *Image Vis. Comput.* **2020**, *104*, 104023. [CrossRef]
49. Adetoyi, O.E.; Awe, B.P. Face Recognition Enabled Door Access Control System. *FUOYE J. Eng. Technol.* **2022**, *7*, 28–31. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.