computation

Article

Face Detection & Recognition from Images & Videos Based on CNN & Raspberry Pi

Muhammad Zamir [1], Nouman Ali [1,*], Amad Naseem [1], Areeb Ahmed Frasteen [1], Bushra Zafar [2], Muhammad Assam [3], Mahmoud Othman [4] and El-Awady Attia [5,6]

[1] Department of Software Engineering, Mirpur University of Science & Technology, Mirpur 10250, Pakistan
[2] Department of Computer Science, Government College University, Faisalabad 38000, Pakistan
[3] Department of Software Engineering, University of Science and Technology, Bannu 28100, Pakistan
[4] Computer Science Department, Faculty of Computers and Information Technology, Future University in Egypt, New Cairo 11835, Egypt
[5] Department of Industrial Engineering, College of Engineering, Prince Sattam Bin Abdulaziz University, Al Kharj 16273, Saudi Arabia
[6] Mechanical Engineering Department, Faculty of Engineering (Shoubra), Benha University, Cairo 13518, Egypt
* Correspondence: nouman.ali@live.com

Abstract: The amount of multimedia content is growing exponentially and a major portion of multimedia content uses images and video. Researchers in the computer vision community are exploring the possible directions to enhance the system accuracy and reliability, and these are the main requirements for robot vision-based systems. Due to the change of facial expressions and the wearing of masks or sunglasses, many face recognition systems fail or the accuracy in recognizing the face decreases in these scenarios. In this work, we contribute a real time surveillance framework using Raspberry Pi and CNN (Convolutional Neural Network) for facial recognition. We have provided a labeled dataset to the system. First, the system is trained upon the labeled dataset to extract different features of the face and landmark face detection and then it compares the query image with the dataset on the basis of features and landmark face detection. Finally, it compares faces and votes between them and gives a result that is based on voting. The classification accuracy of the system based on the CNN model is compared with a mid-level feature extractor that is Histogram of Oriented Gradient (HOG) and the state-of-the-art face detection and recognition methods. Moreover, the accuracy in recognizing the faces in the cases of wearing a mask or sunglasses or in live videos is also evaluated. The highest accuracy achieved for the VMU, face recognition, and 14 celebrity datasets is 98%, 98.24%, 89.39%, and 95.71%, respectively. Experimental results on standard image benchmarks demonstrate the effectiveness of the proposed research in accurate face recognition compared to the state-of-the-art face detection and recognition methods.

Keywords: internet of things; surveillance; convolutional neural networks; face detection

check for updates

Citation: Zamir, M.; Ali, N.; Naseem, A.; Ahmed Frasteen, A.; Zafar, B.; Assam, M.; Othman, M.; Attia, E.-A. Detection and Recognition from Images and Videos Based On CNN and Raspberry Pi. *Computation* **2022**, *10*, 148. https://doi.org/10.3390/computation10090148

Academic Editor: Yudong Zhang

Received: 26 June 2022
Accepted: 16 August 2022
Published: 30 August 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 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 (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Image and video classification is an open research domain for the computer vision research community [1–3]. There are various application domains of image and video classification, such as industrial automation, face recognition, medical image analysis, security surveillance, content-based multimedia analysis, and remote sensing [4–6]. The recent focus of research for image and video analysis is the use of deep learning models and high resolution images to design effective decision support systems that can be used with IoT (Internet of Things) [7–9]. Due to the internet, there is a comfort in life and research is focused on the design of smart gadgets that can control devices remotely [10,11]. The smart devices are accessed through the internet and they can do surveillance through sensors and cameras for smart homes and cities and for different decision support systems [12–14].

Face recognition plays a vital role in the design of security systems and there are various applications of such systems while building an IoT block [15].

IoT is known as the network of connected devices, collecting and sharing information so that those devices can be more complex by connecting to software, a network connection, and sensors and electronic devices [16,17]. Every year billions of devices are connected to the internet to share data with each other. In 2015, almost 15 billion devices were connected and in just five years, devices doubled. In the next five years there were 75 billion devices. There are many applications of IoT in our daily life and security surveillance is one of the building blocks for any IoT-based system [18,19]. Security surveillance systems are based on cameras and motion sensors that can capture images and send decisions to users through any communication medium [18,19].

Recent trends show that 15% of businesses deployed IoT devices for their operation. Due to recent trends and the adoption of technologies' government standard requiring business to be on top of cybersecurity. Recent IoT trends are a cybersecurity concern and the rapid growth of 5G accelerating the world of IoT because IoT and 5G go hand in hand. Wearable Technology Fitness and lifestyle base wearable smart devices, such as smart watches, are beneficial trends. There is also bundled IoT for enterprises because enterprises are driving much of the investment in IoT. Artificial Intelligence and IoT devices were leveraged to analyze patient genetic information and blood samples and to diagnose disease and patient needs.

The most popular IoT based device is Raspberry Pi [20]. Raspberry Pi is used for projects such as home surveillance and it can be easily connected to the Internet using an Ethernet port and USB Wi-Fi. We selected Raspberry Pi for our project due to its efficiency. According to Majumder et al. [21], the sensor of Raspberry Pi can detect motion up to 7 m and the sensor detects the infrared light. If it detects the motion, the output is high, and the time delay can be set accordingly from 0.3 to 300 s. When it does not detect the motion, it automatically changes the output from high to low and vice versa.

Our aim is to effectively detect and recognize the face from image and video based on CNN and Raspberry Pi. According to Chao [22], the main aim of any face detection system is to detect the presence of a face in an image. If the face is present, then locate each face perfectly and the face detection algorithm has to generate boundaries around all the faces. The complexity to recognize faces in images varies due to changes in image background, color of background, poses, expression, position and inclinations, skin color, presence of glasses, facial hair, lightening condition, and image resolution. Figure 1 represents the generic framework for face recognition.

Public security is a priority in the current era; there is a growing need for autonomous systems capable of monitoring hotspots to ensure public safety. In this research article, we have explored a face recognition model that uses video and images as training data and can be used in a decision support system for classification of images and video content. We aim to present a low cost and highly reliable real-time face detection and recognition system that can be used in any IoT-based application. We have explored a deep learning technique that can be used to perform facial recognition using easily-attainable components and libraries, such as Raspberry PI and Dlib, Face Recognition Library, and Open Source Computer Vision Library (OpenCV). It also covers various face recognition machine learning algorithms. The results show that in real-time applications, the system provides better performance despite the limitations of Raspberry PI, such as low CPU and GPU processing power. The entire system is developed on the Raspberry PI board because of its efficiency with powerful architecture and portability.

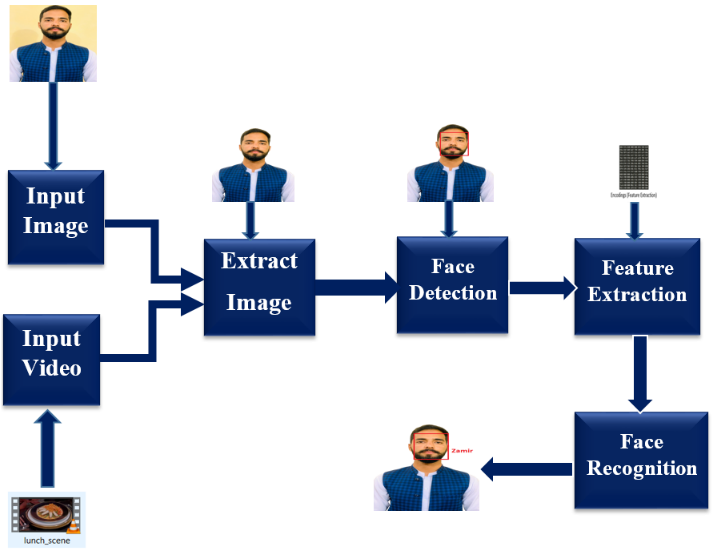


Figure 1. Generic framework for face recognition.

Different criteria are used to assess the effectiveness of the proposed research i.e., using video files, using real time live video, performance comparison using HDD and SSD, standard image benchmarks, and images with face masks. The proposed research is evaluated on three standard image benchmarks i.e., Virtual Makeup (VMU) [23], Face Recognition [24], and 14 Celebrity Face Dataset [25], and a self-created image dataset. In the VMU image benchmark, there are images of women with and without makeup and in the self created dataset, we have used the images of the authors of this research article with and without a face mask. We have used a deep-learning model for training and testing and to train the system, we have used a CNN feature extractor. For testing, we used images and real-time videos to make decisions. The final decision taken by this smart system is sent to the user through email. The rest of this research article is organized as follows: Section 2 is the literature review, Section 3 is the proposed method of research, Section 4 presents the results and discussion, and the conclusion is presented in Section 5.

2. Literature Review

The facial recognition system is a computer vision model that can match a human face with a digital image or with a video frame [26,27]. Facial detection or face verification is one of the challenging research areas in computer vision and there are many real-time applications of face detection [28,29]. The most early research articles relevant to face recognition can be referred to in these published articles [30,31]. Later on, after the 1990s, the focus of the research was to develop a research model that can automatically detect human faces [28,29]. Turk and Pentland [32] proposed a research model that can detect and recognize human faces. In this research, the authors provided an approach for the detection and identification of human faces and described an effective, semi-real-time face recognition system that followed the person's head and then identified the person by comparing facial features. They develop an algorithm for face recognition called "eigen-face". This was the first ever algorithm that presented good and effective results [32].

Another of the most popular algorithms used in face recognition is the Fisherface algorithm. Fisherface recognizes the face based on the reduction in face space dimension using the Principle Component Analysis (PCA) and Linear Discriminant Analysis (LDA) methods to obtain the characteristics of an image. Fisherface is also immune to the noise induced image and blurring effect on the image [33]. In the last two decades, many research models have been proposed that are based on low-level feature extraction and mid-level feature representation, whereas recent research is more focused on the use of deep learning models [28,29]. In the last 10–15 years, many novel algorithms have been developed that can detect human faces and these algorithms are used in applications such as Facebook, WhatsApp, biometric verification, and auto driven cars [28,29].

Liu [34] explored the limitations and drawbacks of face recognition models and discussed that the design of a robust system with a large-scale recognition ability could be a possible future research direction. Li et al. [35] used a machine learning model for face detection, feature extraction, and feature selection. The recent trends in this field are shifted to the use of the complex Convolutional Neural Network (CNN) and these techniques are widely used for security surveillance [36–38]. According to Ding et al. [38], due to the complex structure of CNN, there are some limitations of these embedded devices as their memory is limited. Because of this, the authors proposed a FPGA-based accelerator for face feature extraction, which supports the acceleration of the entire CNN. Neural networks have diverse applications in different domains, such as healthcare, aerial image classification, face recognition, etc. [39–41].

The recent trends for security surveillance are now shifted to the use of IOT-based devices and face recognition is a building block for such smart devices [42,43]. These smart devices are connected through the Internet and can provide many features that are not available in traditional face recognition systems; they can also provide a smart home network [42]. These smart IoT-based devices can be controlled through the Internet while using a web interface or a mobile app. According to [21], Raspberry Pi is a type of smart embedded device and a camera can be used to record video and capture images. Furthermore, a motion detector sensor can assist in detecting the motion. The detected details are sent to the admin by using the Wi-Fi module of this smart embedded device.

3. Proposed Methodology

This section is about the proposed methodology; the details of open CV, the image processing module, the used hardware, and face recognition based on CNN are mentioned in the following subsections.

3.1. Open CV

OpenCV stands for an open source computer vision library [44] and it is designed to perform commonly used functions and it is an application that can be built while using the computer vision library. It includes 2500 optimized algorithms and comprehensive machine learning algorithms. It supports multiple operating systems such as Windows, Linux, Android, and MAC. It can be interfaced with different languages, such as Java, C++, Python, and MATLAB. We have used OpenCV for this research because it runs on multiple operating systems and multiple languages and can easily be interfaced with Raspberry Pi. The main features of Open CV are reading and writing images, capturing and saving video, the recognition of faces from real-time videos, and the detection of the features.

3.2. Image Processing Module

The image processing module can perform different operations of image processing, such as image filtering, enhancement, color space conversion, and histogram computations, among other operations. In Java, OpenCV org.opencv.imgproc is a package that is used for image processing. The step-wise details about the image processing module used in our research model are mentioned below:

- First read the image.
- Load known encoding.
- Convert image into RGB color.
- Detect the coordinate of the bounding box of face from the image to extract the face from the whole image.
- Then, it computes the facial embedding or features of each image that we detect on training time.
- Extract the face from the image.
- Try to match the input face to the training dataset.

- Then, it takes the decision that either the face is known or not.
- If the face is known, it marks the label with the image folder name, or else it is marked as unknown.
- The flow chart summarizes the steps of how our code recognizes a face from an image.

3.3. Hardware

We have implemented the face recognition code on a hardware device. Hardware details are: Raspberry Pi, Pi-Camera, memory card, display screen, and speaker. Memory card is used as a hard disk of the Raspberry Pi. The operation system and all the libraries are saved on the memory card and the card is inserted in the card port of the Raspberry Pi. The screen is attached to the HDMI port of Raspberry Pi and the power of 5A is provided through the power adopter. The Operating System (OS) is Raspbian that will automatically open the GUI interface and the user interacts with the Raspberry Pi while using a mouse and keyboard. The Pi-camera is connected with the Raspberry Pi through the camera module. This Pi- camera is used to obtain the image or real-time video of the user which is then used for further processing. As this research is about the face recognition, for real time simulation of the project, Raspberry Pi is used. The camera is used to gather the real-time video and from this video we have captured the frames. Each of the captured frames are then processed as a single image. We first detect the face in the frames using landmark and after the detection of faces, the features are extracted from the faces.

We have divided the dataset into training and test sets and trained our system while using the training set, which is done offline. After the successful training of the dataset, the features of each person are saved as encodings in a file. Then, this file is copied in the Raspberry-Pi and for testing, we have given the path of the encoding file in the code. The most matched encodings are then gathered, and the name of the person is then labeled on the face. Figure 2 shows the image of the Pi-camera and associated hardware while the step-wise block diagram of the proposed methodology is shown in Figure 3.

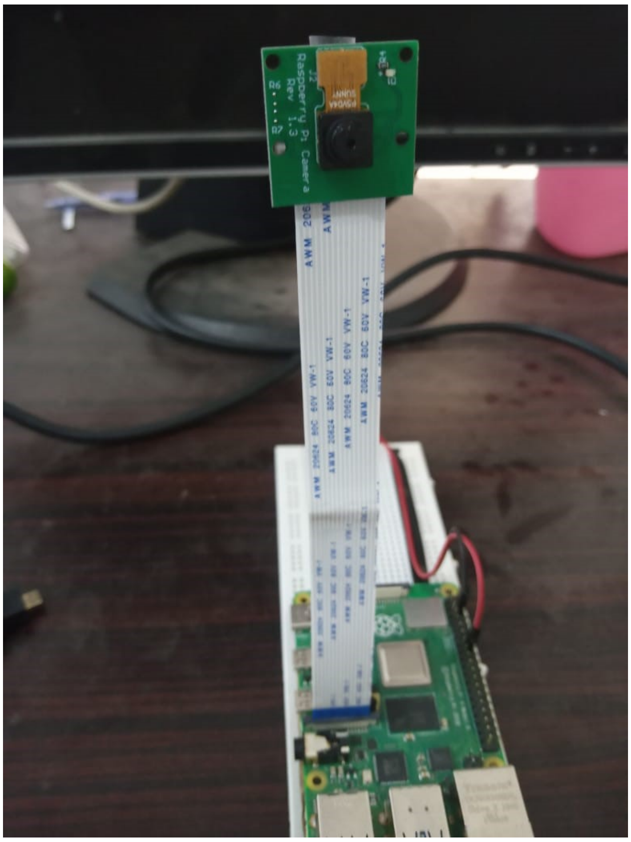


Figure 2. Pi-camera and associated hardware.

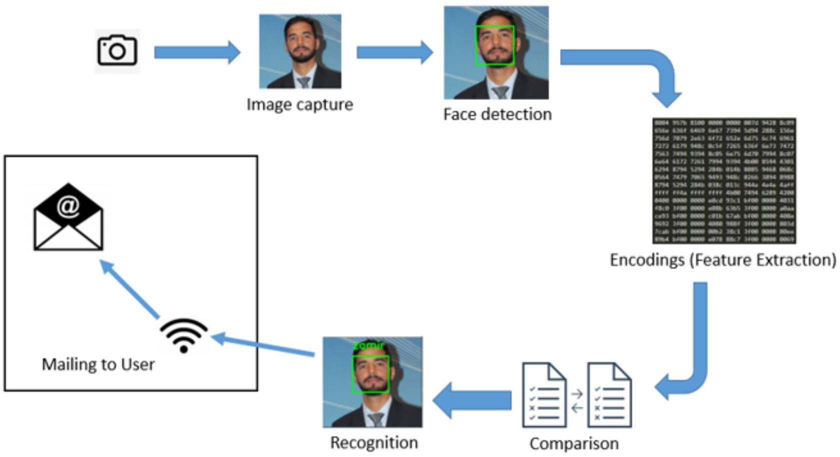


Figure 3. Proposed methodology.

3.4. Face Recognition Based On CNN

The success of CNN is due to its network architecture, enormous training data, and loss function that enhances the discriminative power of CNN [45,46]. The convolutional layer is the first layer of the CNN model. In this layer, first we apply two filters of 5 × 5 on 25 × 25 input image. The same filters are applied for the three channels, respectively. Then, we have applied the activation function on it. Suppose we have an image "X" and apply filter "f" on it; its equation will look like this [47].

$$Z = X \times f \tag{1}$$

The following parameters of the convolutional layer are used to measure computation speed. C_i indicates number of input channels; C_o indicates number of output channels; K_w indicates width of kernel and Kh indicates height of output channel

$$[(C_i \times K_w \times K_h) + (C_i \times K_w \times K_h - 1) + 1] \times CO \times W \times H \tag{2}$$

The Rectified Linear Unit (ReLU) allows the most effective and fast training by setting the native value to zero and keeping the appropriate value. This is sometimes referred to as the performance because sometimes only the element referred to the next filter is used. In the pooling layer, we applied one filter of size 4 × 4 and max function to each of the previous three outputs of size 25 × 25. Here, we have reduced the size to obtain three tensor outputs of size 6 × 6. It simplifies line-by-line sampling by reducing the number of polling parameters. For a feature map dimension nh × nw × nc, the dimension of output is obtained after a pooling layer is [48]

$$P = (nh - f + 1)/s \times (nw - f + 1)/s \times nc \tag{3}$$

where:

- n_h height of feature map;
- n_w width of feature map;
- n_c number of channels in the feature map;
- f size of filter;
- s stride length.

On the previous three 6 × 6 size outputs, here we obtain another output that combines all the pixel value of the previous output [47]:

$$Z = W^T \cdot X + b \tag{4}$$

"Z" is the name of the variable where we store the value; it is the function that we apply in the occurrence of an element. "W" is the weight that we assign to each element; "X" is the image and "b" is the borrow.

The parameter of the fully connected layer is used to calculate the amount of computation form given in the below equations used.

$$a_1 = W_1 1 \times x_1 + W_1 2 \times x_2 + W_1 3 \times x_3 + b_1 \tag{5}$$

$$a_2 = W_2 1 \times x_1 + W_2 2 \times x_2 + W_2 3 \times x_3 + b_2 \tag{6}$$

$$a_3 = W_3 1 \times x_1 + W_3 2 \times x_2 + W_3 3 \times x_3 + b_3 \tag{7}$$

$X1, X2, X3$ Input for fully connection layer a_1, a_2, a_3 for export.

$$params = (I + 1) \times O = I \times O + O \tag{8}$$

where I = input neuron, O = output neuron. Each output neuron connects with input neurons [49].

$$FLOPs = [I + (I - 1) + 1] \times O = (2 \times I) \times O \tag{9}$$

I = input neurons, O = output neurons. The value in brackets indicates the amount of computation required to compute a neuron. I-1 indicates the amount of addition, +1 Express bias, × O for calculation number of output neurons. Softmax can return the max value from the list that works like a probability function. For small input, it turns into small or negative and for large input it turn into large but it works between 0 and 1.

ResNet is used to train the CNN model and it helps to improve the performance and stability. The residual scaling factor in ResNet is used to set the value manually. We believe that by changing the training parameters and by using a small value, one can improve the stability of the model. We further adopted a small trick to altering the activation function value in the RELU layer of the CNN model. The proposed method slightly increases the training parameter but improves its performance and training stability. In Figure 4, we notice there is a direct connection that is called skip connection and it is the heart of the residual block [50].

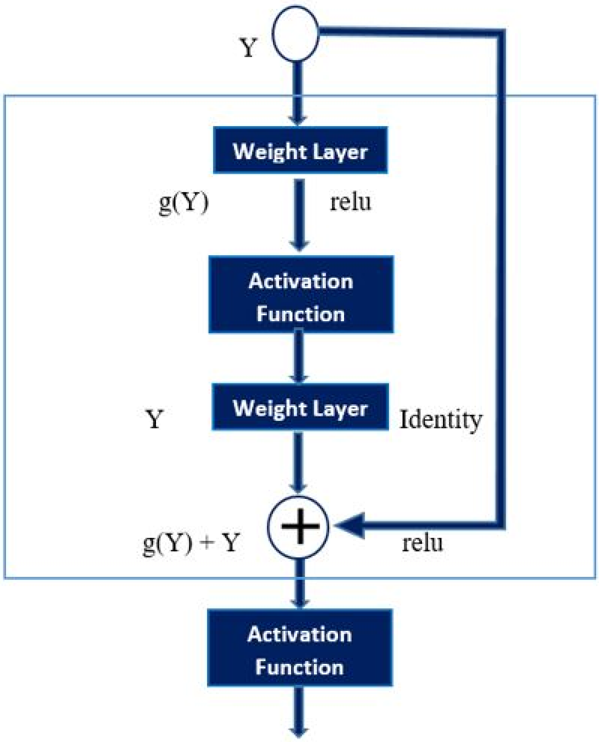


Figure 4. Illustration of residual block.

The CNN algorithm extracts the features and saves them in an encoding file in the project. These encodings are saved as digital numbers for a full-face image and these numbers for each image are 128. In the testing of images, these 128 images are again extracted and then compared with the saved encodings. The highly matched features of images are compared and then the face is recognized.

Our faces have different features that can be identified, such as our eyes, mouth, nose, etc. We used the Dlib algorithm to detect the face and we obtained a map of the point that surrounds each feature. For the Dlib facial recognition, the output is computed by 128-d and the training of the network is completed by using the triplet dataset. Figure 5 represents the overall flowchart of the proposed research.

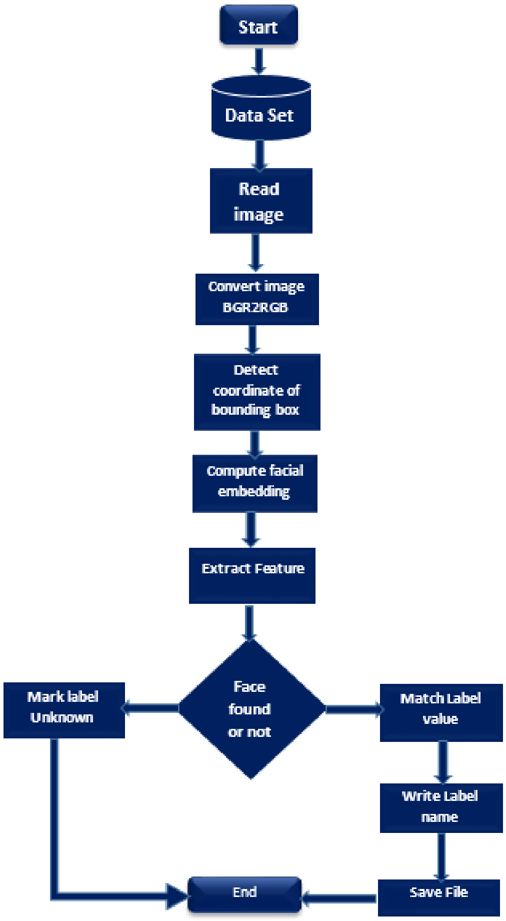


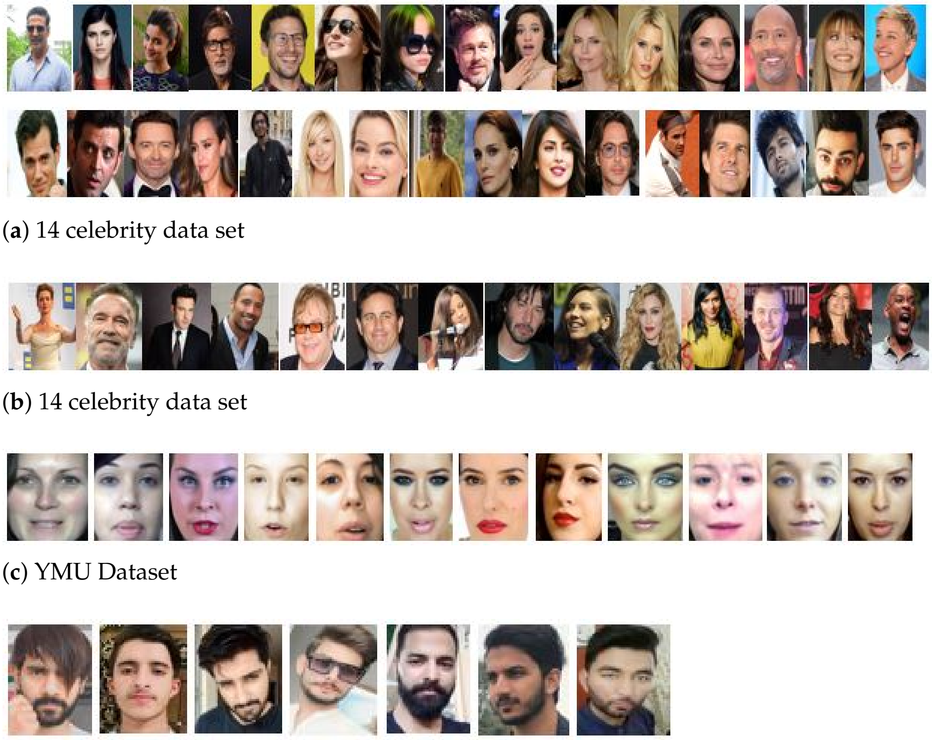
Figure 5. Flow chart of proposed research.

4. Datasets and Results

The proposed research methodology is evaluated on four different datasets, namely: three open benchmark datasets named Face Recognition dataset, 14 celebrity dataset, Virtual Makeup (VMU) dataset [23], and our own created dataset. We have created our own dataset consisting of 700 images of 7 people with 100 images of each person. We used the 80/20 and 70/30 rule for dataset training and testing. Table 1 shows the dataset used for the evaluation of the proposed research methodology. The sample images from each image benchmark are shown in Figure 6, respectively.

Table 1. Datasets used for evaluation of the proposed research.

Name of Dataset	Total Images	No. of Classes	Ratio	No. of Training Images	No. of Test Images
VMU Dataset [23]	156	12	70:30 80:20	110 125	46 31
Face Recognition Dataset [24]	2562	31	70:30 80:20	1794 2049	768 512
14 Celebrity Face Dataset [25]	220	14	70:30 80:20	154 176	66 44
Own created dataset	700	7	70:30 80:20	490 560	210 140



(**a**) 14 celebrity data set

(**b**) 14 celebrity data set

(**c**) YMU Dataset

(**d**) Own created dataset

Figure 6. The photo gallery based on a random selection of images taken from each dataset.

4.1. Evaluation Metrics

The metrics used for the evaluation of the proposed research are discussed below.

[I] Confusion Matrix: Both Precision and recall can be intercepted from the confusion matrix [51]. The confusion matrix is used to represent how well a model made its predictions. In Table 2, T_P is the true positive mean if we give the known image to test a model, it leaves the correct result mark label unknown. T_N is true negative, which means that if we give an unknown image to test a model, it gives the correct result mark label unknown. F_P is false positive, which means that if we give an unknown image (negative) to test a model, it gives the wrong correct result mark label known (positive). F_N is false negative, it means that if we give the known image (positive) to test a model, it gives the correct result mark label unknown (negative).

Table 2. Confusion Matrix.

		Predicted Case	
		Positive	**Negative**
Actual Case	Positive	T_P	F_N
	Negative	F_P	T_N

[II] Accuracy (ACC): Accuracy is the measurement of how accurately the model recognizes a face. Accuracy is the ratio of sum of true positive and true negative over the total number of images.

[III] Recall: In recall, instead of looking for false positives, it looks at the number of false negatives. Recall is used whenever a false negative is predicted.

[IV] Precision: Precision is the ratio of true positive to the total of the true positive and false positives. Precision measures how much positive junk got thrown in the matrix. The smaller the number of false positives, the greater the model precision and vice versa.

[V] F-measure/F1-Score: F1-score is one of the important evaluation criteria in deep learning. It is also known as the harmonic mean of precision and recall. It combines precision and recall into a single number.

4.2. Training of Proposed Research Model

We need to train a strong model for training because of the growing scale of face recognition. For this, we contribute a cleaned and noise-controlled subset of our own created dataset by removing the extra background and decreased image size. With a cleaned and noise-controlled dataset, we achieve higher results because the system is a trained and efficiently noise-controlled dataset [52]. The self-created dataset consists of 700 images and we created this dataset by capturing 100 images of each person. All the images are different in terms of view-point, orientation, etc. For recognition purposes, we trained the system on the images placed in the training dataset. We have used the mid-level feature extractor Histogram of Oriented Gradient (HoG) [53] and a CNN model [54]. In order to sort the optimal performance and to perform a comparison, we have compared the proposed research based on CNN with HoG [53], which is a mid-level feature extractor. We have tested algorithms and results are presented in the form of tables and graphs; more detail is mentioned below.

4.3. Testing of the Proposed Research Model

After the successful training of the system, we saved the labels and features in the folder name encoding. This folder is used to compare the input image and the image that is to be tested. The algorithm compares the features of both images and gives the output that is the label for that person (we used the name of the person as the label). To recognize the face from an image, we have to place the image in an input folder. Then, we extract the features of the input image and compare them with the encoding folder. The computed results are mentioned in Table 3.

Secondly, we train the system with the HOG algorithm and a very small dataset of about five images of the same person and test the input image with the HOG algorithm. We have tested a single image of a person on both algorithms and trained the system with the HOG algorithm using a large dataset of about 30 images per person. We have tested a single image of a person on both algorithms and trained the system with the HOG algorithm using a small dataset of about five images per person. The results presented in Table 3 show that the larger dataset gives more accurate results than the small dataset. However, with the large dataset, system performance is slow.

Table 3. Performance of algorithms under different training conditions.

	Train System	**Input Image**	**Accuracy**
Small Dataset	HOG	HOG	80%
	HOG	CNN	90%
	CNN	HOG	94%
	CNN	CNN	98%
Large Dataset	HOG	HOG	90%
	HOG	CNN	93%
	CNN	HOG	95%
	CNN	CNN	98.3%

4.3.1. Testing of the Proposed Research Model Using Video Files

Face recognition from a video is a challenging process as compared to when we are using a single image. In the case of videos, frames are extracted from videos and these frames are sent one by one as an the input image to the system to identify and recognize the face in an image. We have tested a small video on both algorithms and trained the system with HOG and then CNN algorithm using a large dataset of about 30 images per person. The results are presented in Figure 7.

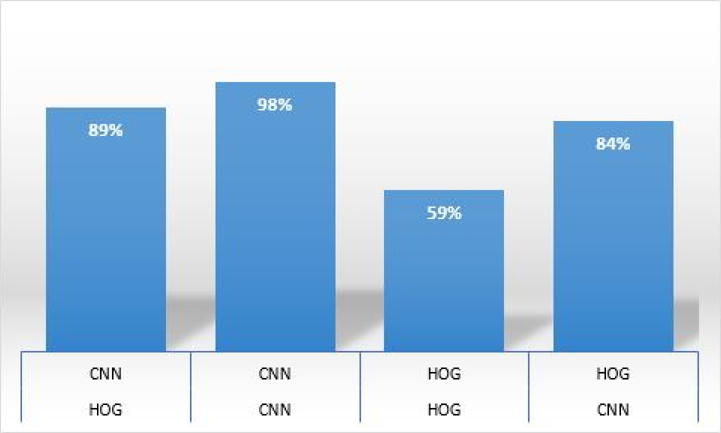


Figure 7. Comparison of HoG and CNN while Using Video to Detect Faces.

In Figure 7, the first row of the label indicates the model we used for the training of our dataset and the label in the second row represents the model we used for testing to extract the feature from the input video and match it with the trained data. The first bar in Figure 7 shows that the accuracy of the CNN model is 89%, when we trained dataset using CNN and tested the system by the HOG feature descriptor. Similarly, the second bar depicts that the accuracy of CNN is 98%, when we used CNN for both the training and testing of the system. The third bar shows that accuracy to recognize face from video is 59% when we used the HOG feature descriptor for both the training and testing of the system. Bar 4 shows that the accuracy to recognize the face from the video is 84% when we trained the model on the HOG feature descriptor model, and CNN is used to extract features of a face from the input video and match them with the trained dataset.

We have tested a small video while using both algorithms and trained the system with HOG and then the CNN algorithm using a small dataset of about five images per person. The results are shown in Figure 8. The first row of the label in Figure 8 indicates the model we used for the training of our dataset and the label in the second row represents the model that we used to extract features from the input video during testing. The two graphs show that the highest value of accuracy in face recognition is achieved by CNN.

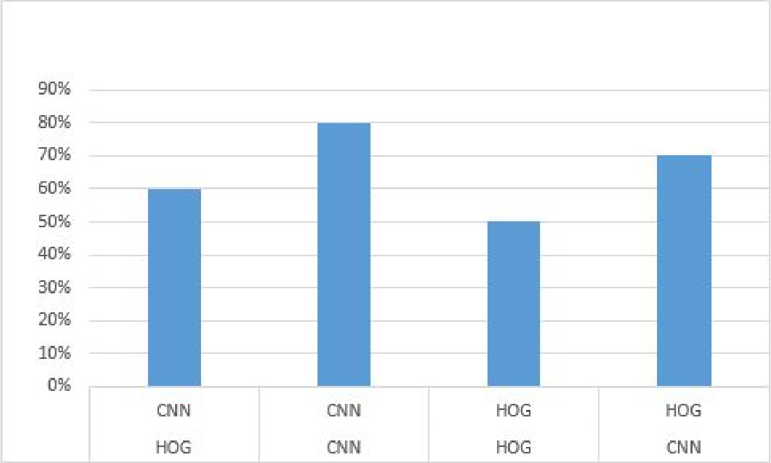


Figure 8. Results When Using a Small Duration Video as an Input to Detect Faces.

4.3.2. Testing of the Proposed Research Model Using Live Videos

To detect the face from live videos, we used an external component, which is a camera, to take the live videos. We have carried out some tests while using CNN and this process is computationally expensive. After testing, the recognized face must be sent to the admin. It notifies the admin by sending a name/label for that person as an email if the face in an image is matched with the encoding. If the face does not match with the images in the dataset, it will send the "unknown" word to the admin in the email.

4.4. Comparison of Proposed Research with HoG While Using Hard-Disk-Drive (HDD) and Sold-State-Drive (SSD)

Table 4 presents a comparison of the proposed research with HoG while using the operating system installed on SSD and with HoG while using the operating system installed on HDD. This shows that in the case of HDD, the computer takes almost double the time to train the system. The CNN algorithm takes more time to train the system because of its complexity and the feature extraction in HOG is simpler than CNN. Table 4 presents a comparison of the proposed research with HoG in terms of memory consumption while using SSD and HDD. It shows that with HDD CNN, it consumes more memory by about 30% more than the HOG, and in the case of SSD, the values are low.

Table 4. Performance comparison of HOG and CNN using HDD and SSD.

	Algorithm	**Time Taken on 1 Image**	**Time Taken on 50 Images**	**Memory Consumption**
Using HDD	HOG	0.25 min	12.5 min	20% during training
	CNN	1.5 min	75 min	40% during training
Using SSD	HOG	0.06 s	3 min	15% during training
	CNN	0.5 min	30 min	25% during training
Without SSD	HOG	0.5 min	30 min	25% during training
	CNN	2 min	120 min	30% during training

4.5. Experimental Results on Standard Image Benchmarks

This section provides a discussion on results obtained on standard image benchmarks and presents a comparison with the state-of-the-art research.

4.5.1. Results for VMU Image Dataset

The proposed research is evaluated while using the Face Recognition Dataset, 14 celebrity dataset bench mark, Virtual Makeup (VMU) image benchmark [23], and the self-created dataset which consists of images of seven different people, 100 each. For each dataset, we use the 80:20 and 70:30 rule for their training and testing. For the VMU image benchmark, 75% images are used for training and 25% for testing (the same is represented in Figure 9).

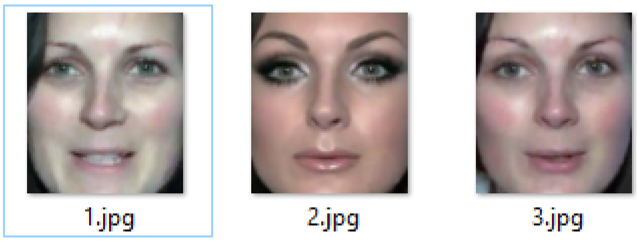


Figure 9. Training images for Person-1 for VMU Image Benchmark.

As CNN extracts 128 different features, the histogram for these images is shown in Figure 10. The result for testing image histogram is represented in Figure 11. The confusion matrix for the VMU image benchmark is shown in Figure 12. The accuracy achieved for VMU data using the 70:30 ratio is 89.5% and 80:20 is 98%, respectively.

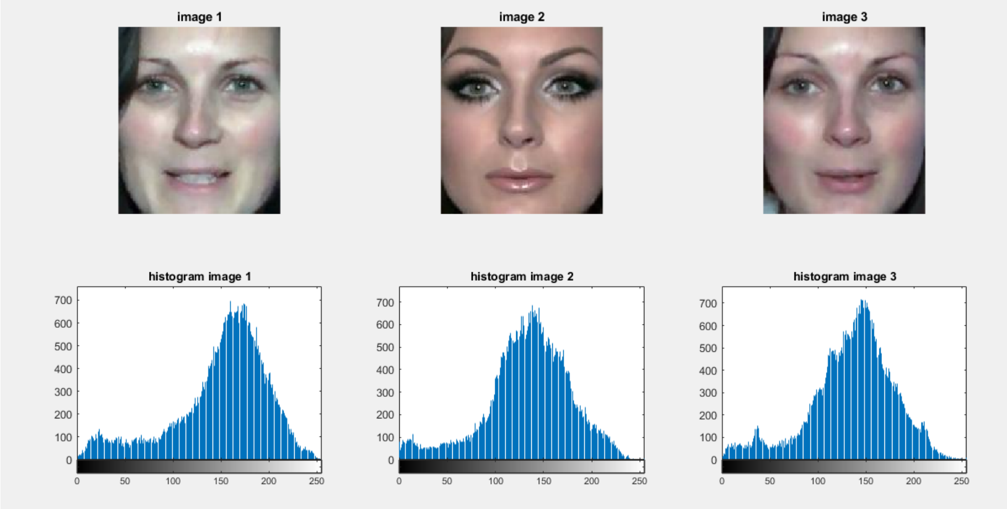


Figure 10. Histogram-based representation of person 1.

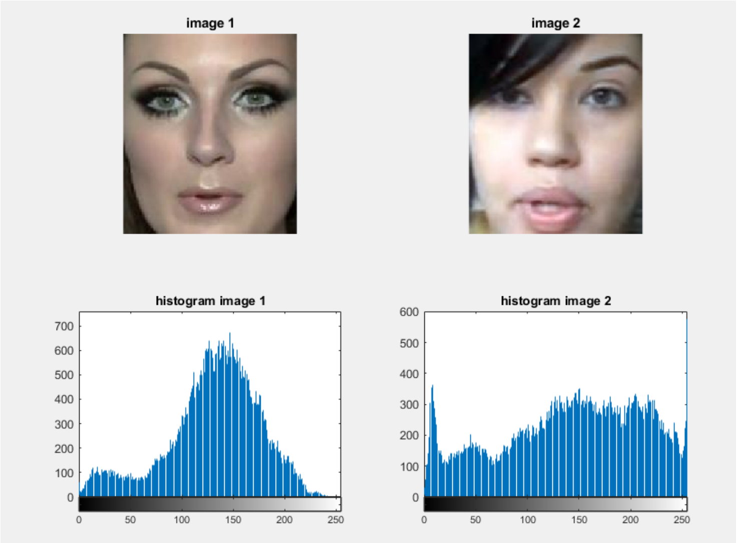
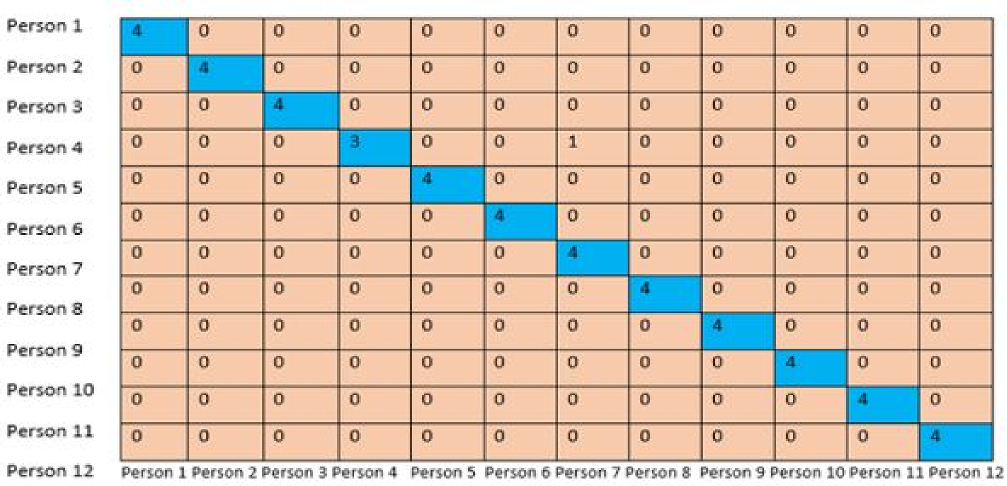


Figure 11. Histogram of Testing Image.



	Person 1	Person 2	Person 3	Person 4	Person 5	Person 6	Person 7	Person 8	Person 9	Person 10	Person 11	Person 12
Person 1	4	0	0	0	0	0	0	0	0	0	0	0
Person 2	0	4	0	0	0	0	0	0	0	0	0	0
Person 3	0	0	4	0	0	0	0	0	0	0	0	0
Person 4	0	0	0	3	0	0	1	0	0	0	0	0
Person 5	0	0	0	0	4	0	0	0	0	0	0	0
Person 6	0	0	0	0	0	4	0	0	0	0	0	0
Person 7	0	0	0	0	0	0	4	0	0	0	0	0
Person 8	0	0	0	0	0	0	0	4	0	0	0	0
Person 9	0	0	0	0	0	0	0	0	4	0	0	0
Person 10	0	0	0	0	0	0	0	0	0	4	0	0
Person 11	0	0	0	0	0	0	0	0	0	0	4	0
Person 12	0	0	0	0	0	0	0	0	0	0	0	4

Figure 12. Confusion matrix for VMU image benchmark.

Figures 13 and 14 represent testing images of 12 people and the output of the CNN model, respectively.

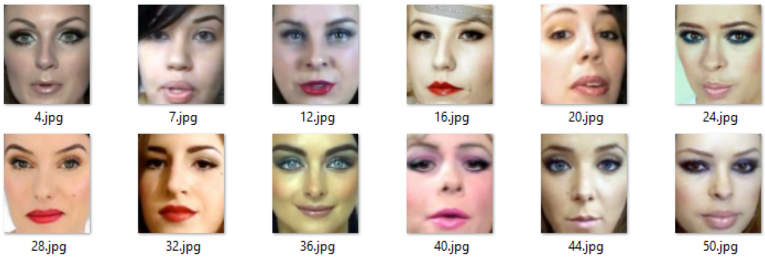


Figure 13. Testing images of 12 people.

Table 5 shows a comparison of the classification accuracy of the VMU dataset with the state-of-the-art research. It can be evidently seen that the proposed research achieves better performance as compared to the related research.

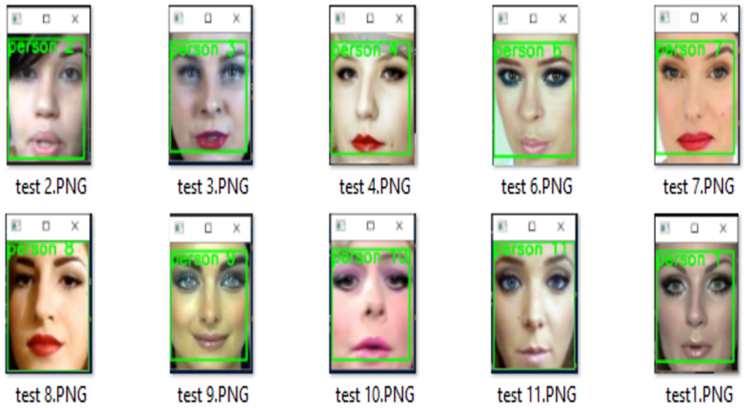


Figure 14. Output of CNN model.

Table 5. Comparison of VMU with the state-of-the-art research.

Algorithm	Accuracy of Face Recognition
Guo et al. [55]	82.2%
Tripathi et al. [56]	87.35%
Sajid et al. [23]	88.48%
Sajid et al. [23]	92.99%
Wang and Fu et al. [57]	93.75%
Proposed Research (CNN)	98%

4.5.2. Results for Face Recognition Dataset

Table 6 shows the quantity analysis comparison of the dataset that we used in our research. In this table, we discuss the classification accuracy and performance of the proposed research method on three different datasets; two are open bench mark datasets and one is own created dataset. The result accuracy varies when we change the ratio of training and testing. When its ratio is 70:30, its accuracy is 97.39%. When we change the ratio from 70:30 to 80:20, its accuracy is increased up to 98.24%. Similarly, Table 6 demonstrates Precision, Recall, and F1-score results (98.61%, 98%, and 98.45%) when the ratio is 70:30 and 99.10%, 98.88%, and 98.98% when its training and testing ratio changed from 70:30 to 80:20. Its accuracy is higher than another dataset that we used in our research to train and test the modal because its size is also larger than the others; it contains 2562 images. The graphical comparison of the proposed methods on standard Image Benchmarks is shown in Figure 15.

Table 6. Experimental Results of the proposed methods on Standard Image Benchmarks.

Dataset	Training to Test Ratio	Accuracy	Precision	Recall	F-Score
Face Recognition Dataset	70:30	97.39%	98.61%	98%	98.45%
Face Recognition Dataset	80:20	98.24%	99.10%	98.88%	98.98%
14 Celebrity Dataset	70:30	89.39%	91.00%	93.02%	92.00%
14 Celebrity Dataset	80:20	88.63%	93.54%	90.62%	92.05%
Own Created Dataset	70:30	95.23%	96.51%	97.64%	97.07%
Own Created Dataset	80:20	95.71%	98.09%	96.26%	97.16%

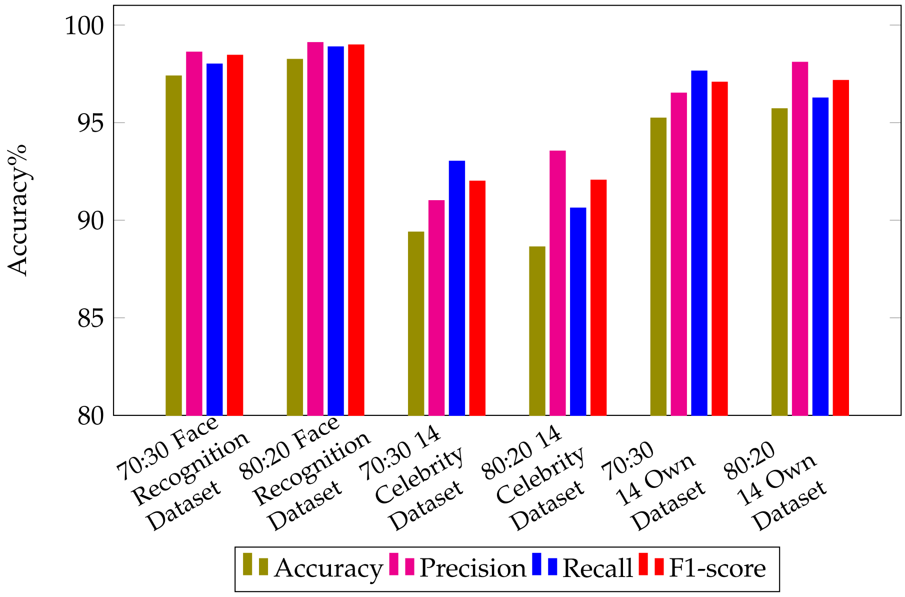


Figure 15. Performance comparison with standard image benchmarks.

4.5.3. Results for 14 Celebrity Dataset

The 14 Celebrity dataset contains a total of 220 images of 14 different celebrities. Model performance and results on this dataset are explained in Table 6. Similarly, like the Face Recognition dataset, we train and test this dataset on the same ratio, 70:30 and 80:20. The 14 Celebrity dataset accuracy is 98.39% when we trained our modal on the 70% image of the dataset and tested on the remaining 30% of images in the dataset. Its precision, recall, and F1-score are 91%, 93.02%, and 92%. When we change the ratio to 80:20, its accuracy increased up to 98.63% and precision, recall, and F1-score results are 93.54%, 90.62%, and 92.05%, as can be seen in Table 6.

4.5.4. Results for Own Created Dataset

Our own created dataset contains a total of 700 images of seven different people, 100 of each. Its accuracy is higher than the 14 celebrity dataset because the dataset size is also larger than the 14 Celebrity dataset. Model accuracy, precision, recall, and F1-score is 95.23%, 96.51%, 97.64%, and 97.07% when we train and test the model on 70:30 ratio; while the ratio is changed from 70:30 to 80:20, its accuracy, precision, recall, and F1-score are 97.71%, 98.09%, 96.26%, and 97.16%, respectively, as can be seen in Table 6.

Now we will discuss the results associated with the self created dataset. The accuracy of HoG and CNN while using our own created dataset is shown in Table 7. The output images with face masks and detected labels are shown in Figure 16. During COVID-19, almost every person was wearing a face mask. It is difficult for the already existing system to recognize a face in a mask and if the system does recognize it, its accuracy is low because the system is trained on faces without masks. So, we trained an efficient system that recognizes faces in masks [58].

Table 7. Accuracy of HoG and CNN while using images with face mask.

Algorithm	**Accuracy of Face Recognition**
HOG	90%
CNN	98.3%

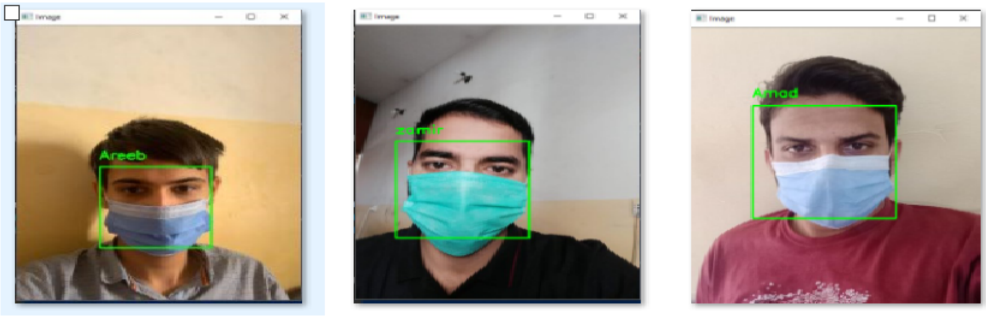


Figure 16. Output images with face mask.

The above-mentioned results represent the high performance of the proposed face detection algorithm, which is based on CNN. The computation time for CNN is higher as it consists of a complex layered structure.

5. Conclusions

Image and video classification is an open research domain for the computer vision research community. There are various application domains of image and video classification, such as industrial automation, face recognition, medical image analysis, security surveillance, content-based multimedia analysis, and remote sensing. The recent focus of research for image and video analysis is the use of deep learning models and high resolution images to design effective decision support systems that can be used with IoTs (Internet of Things). In this research article, we have presented a deep learning method based on CNN (Convolutional Neural Network) to recognize faces while using Raspberry-Pi. We have provided three standard benchmark labeled datasets (VMU,14 celebrity dataset, Face recognition dataset) and one of our own created datasets to the system. First, the system is trained upon the labeled dataset to extract different features of face and landmark face detection and then it compares the query image with the dataset on the basis of features and landmark face detection. Finally, it compares faces and votes between them and gives a result that is based on voting. We have compared the classification accuracy of CNN with a mid-level feature extractor, i.e., Histogram of Oriented Gradient. We have compared the classification accuracy of our model with the state-of-the-art previous research and have achieved higher accuracy results. Experimental results demonstrate the effectiveness of the proposed research in accurate face detection compared to the state-of-the-art face detection and recognition methods. In the future, we aim to extend this work to solve the real-life engineering design problems by applying metaheuristic techniques. The high-dimensional hardware configuration design could be another possible future contribution.

Author Contributions: Conceptualization, M.Z., N.A. and A.N.; Data curation, M.Z., N.A., B.Z., A.A.F. and M.A.; Formal analysis, M.Z., N.A. and A.N.; Investigation, M.Z.; Methodology, M.Z., N.A., B.Z., A.A.F. and A.N.; Project administration, N.A., B.Z., A.A.F., M.O. and E.-A.A.; Resources, M.A., M.O. and E.-A.A.; Software, M.A.; Supervision, N.A., M.O. and E.-A.A.; Validation, A.N. and E.-A.A.; Writing—original draft, M.Z., N.A., A.N., A.A.F., M.A. and E.-A.A.; Writing—review & editing, N.A., B.Z. and M.O. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

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

References

1. Wang, P.; Fan, E.; Wang, P. Comparative analysis of image classification algorithms based on traditional machine learning and deep learning. *Pattern Recognit. Lett.* **2021**, *141*, 61–67. [CrossRef]
2. Latif, A.; Rasheed, A.; Sajid, U.; Ahmed, J.; Ali, N.; Ratyal, N.I.; Zafar, B.; Dar, S.H.; Sajid, M.; Khalil, T. Content-based image retrieval and feature extraction: A comprehensive review. *Math. Probl. Eng.* **2019**, 9658350. [CrossRef]
3. Saqlain, M.; Rubab, S.; Khan, M.M.; Ali, N.; Ali, S. Hybrid Approach for Shelf Monitoring and Planogram Compliance (Hyb-SMPC) in Retails Using Deep Learning and Computer Vision. *Math. Probl. Eng.* **2022**, 4916818. [CrossRef]
4. Shabbir, A.; Rasheed, A.; Shehraz, H.; Saleem, A.; Zafar, B.; Sajid, M.; Ali, N.; Dar, S.H.; Shehryar, T. Detection of glaucoma using retinal fundus images: A comprehensive review. *Math. Biosci. Eng.* **2021**, *18*, 2033–2076. [CrossRef]
5. Sajid, M.; Ali, N.; Ratyal, N.I.; Dar, S.H.; Zafar, B. Facial asymmetry-based Feature extraction for different applications: A review complemented by new advances. *Artif. Intell. Rev.* **2021**, *54*, 4379–4419. [CrossRef]
6. Rasheed, A.; Zafar, B.; Rasheed, A.; Ali, N.; Sajid, M.; Dar, S.H.; Habib, U.; Shehryar, T.; Mahmood, M.T. Fabric defect detection using computer vision techniques: A comprehensive review. *Math. Probl. Eng.* **2020**, 8189403. [CrossRef]
7. Zhang, J.; Ye, G.; Tu, Z.; Qin, Y.; Qin, Q.; Zhang, J.; Liu, J. A spatial attentive and temporal dilated (SATD) GCN for skeleton-based action recognition. *CAAI Trans. Intell. Technol.* **2022**, *7*, 46–55. [CrossRef]
8. Zou, Q.; Xiong, K.; Fang, Q.; Jiang, B. Deep imitation reinforcement learning for self-driving by vision. *CAAI Trans. Intell. Technol.* **2021**, *6*, 493–503. [CrossRef]
9. Ali, N.; Bajwa, K.B.; Sablatnig, R.; Chatzichristofis, S.A.; Iqbal, Z.; Rashid, M.; Habib, H.A. A novel image retrieval based on visual words integration of SIFT and SURF. *PLoS ONE* **2016**, *11*, e0157428. [CrossRef]
10. Bellini, P.; Nesi, P.; Pantaleo, G. IoT-Enabled Smart Cities: A Review of Concepts, Frameworks and Key Technologies. *Appl. Sci.* **2022**, *12*, 1607. [CrossRef]
11. Qazi, S.; Khawaja, B.A.; Farooq, Q.U. IoT-Equipped and AI-Enabled Next Generation Smart Agriculture: A Critical Review, Current Challenges and Future Trends. *IEEE Access* **2022**, *10*, 21219–21235. [CrossRef]
12. Afzal, K.; Tariq, R.; Aadil, F.; Iqbal, Z.; Ali, N.; Sajid, M. An optimized and efficient routing protocol application for IoV. *Math. Probl. Eng.* **2021**, 9977252. [CrossRef]
13. Malik, U.M.; Javed, M.A.; Zeadally, S.; ul Islam, S. Energy efficient fog computing for 6G enabled massive IoT: Recent trends and future opportunities. *IEEE Internet Things J.* **2021**, *9*, 14572–14594. [CrossRef]
14. Fatima, S.; Aslam, N.A.; Tariq, I.; Ali, N. Home security and automation based on internet of things: A comprehensive review. In *Proceedings of the IOP Conference Series: Materials Science and Engineering*; IOP Publishing: Bristol, UK, 2020; Volume 899, p. 012011.
15. Saponara, S.; Giordano, S.; Mariani, R. *Recent Trends on IoT Systems for Traffic Monitoring and for Autonomous and Connected Vehicles*; MDPI: Basel, Switzerland, 2021; Volume 21, p. 1648.
16. Zobaed, S.; Hassan, M.; Islam, M.U.; Haque, M.E. Deep learning in iot-based healthcare applications. In *Deep Learning for Internet of Things Infrastructure*; CRC Press: Boca Raton, FL, USA, 2021; pp. 183–200.
17. Kumar, A.; Salau, A.O.; Gupta, S.; Paliwal, K. Recent trends in IoT and its requisition with IoT built engineering: A review. *Adv. Signal Process. Commun.* **2019**, 15–25. [CrossRef]
18. Ahmad, R.; Alsmadi, I. Machine learning approaches to IoT security: A systematic literature review. *Internet Things* **2021**, *14*, 100365. [CrossRef]
19. Harbi, Y.; Aliouat, Z.; Refoufi, A.; Harous, S. Recent Security Trends in Internet of Things: A Comprehensive Survey. *IEEE Access* **2021**, *9*, 113292–113314. [CrossRef]
20. Jabbar, W.A.; Wei, C.W.; Azmi, N.A.A.M.; Haironnazli, N.A. An IoT Raspberry Pi-based parking management system for smart campus. *Internet Things* **2021**, *14*, 100387. [CrossRef]
21. Majumder, A.J.; Izaguirre, J.A. A smart IoT security system for smart-home using motion detection and facial recognition. In Proceedings of the 2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC), Madrid, Spain, 13–17 July 2020; pp. 1065–1071.
22. Chao, W.L. Face Recognition. Available online: http://disp.ee.ntu.edu.tw/~pujols/Face%20Recognition-survey.pdf (accessed on 5 April 2022).
23. Sajid, M.; Ali, N.; Dar, S.H.; Iqbal Ratyal, N.; Butt, A.R.; Zafar, B.; Shafique, T.; Baig, M.J.A.; Riaz, I.; Baig, S. Data augmentation-assisted makeup-invariant face recognition. *Math. Probl. Eng.* **2018**, 2850632. [CrossRef]
24. Patel, V. Face Recognition Dataset. Available online: https://www.kaggle.com/datasets/vasukipatel/face-recognition-dataset (accessed on 5 March 2022).
25. Danup, N. 14 Celebrity Faces Dataset. Available online: https://www.kaggle.com/datasets/danupnelson/14-celebrity-faces-dataset (accessed on 7 March 2022).
26. Sajid, M.; Ali, N.; Dar, S.H.; Zafar, B.; Iqbal, M.K. Short search space and synthesized-reference re-ranking for face image retrieval. *Appl. Soft Comput.* **2021**, *99*, 106871. [CrossRef]
27. Ratyal, N.; Taj, I.A.; Sajid, M.; Mahmood, A.; Razzaq, S.; Dar, S.H.; Ali, N.; Usman, M.; Baig, M.J.A.; Mussadiq, U. Deeply learned pose invariant image analysis with applications in 3D face recognition. *Math. Probl. Eng.* **2019**, 3547416. [CrossRef]
28. Wang, H.; Guo, L. Research on face recognition based on deep learning. In Proceedings of the 2021 3rd International Conference on Artificial Intelligence and Advanced Manufacture (AIAM), Manchester, UK, 23–25 October 2021; pp. 540–546.

29. Ge, H.; Zhu, Z.; Dai, Y.; Wang, B.; Wu, X. Facial expression recognition based on deep learning. *Comput. Methods Programs Biomed.* **2022**, *215*, 106621. [CrossRef] [PubMed]
30. Kaya, Y.; Kobayashi, K. A basic study on human face recognition. In *Frontiers of Pattern Recognition*; Elsevier: Amsterdam, The Netherlands, 1972; pp. 265–289.
31. Kanade, T. *Computer Recognition of Human Faces*; Birkhäuser: Basel, Germany, 1977; Volume 47.
32. Turk, M.A.; Pentland, A.P. Face recognition using eigenfaces. In Proceedings of the 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Maui, HI, USA, 3–6 June 1991; pp. 586–587.
33. Anggo, M.; Arapu, L. Face recognition using fisherface method. *J. Physics Conf. Ser.* **2018**, *1028*, 012119. [CrossRef]
34. Liu, C. The development trend of evaluating face-recognition technology. In Proceedings of the 2014 International Conference on Mechatronics and Control (ICMC), Jinzhou, China, 3–5 July 2014; pp. 1540–1544.
35. Li-Hong, Z.; Fei, L.; Yong-Jun, W. Face recognition based on LBP and genetic algorithm. In Proceedings of the 2016 Chinese Control and Decision Conference (CCDC), Yinchuan, China, 28–30 May 2016; pp. 1582–1587.
36. Jiang, H.; Learned-Miller, E. Face detection with the faster R-CNN. In Proceedings of the 2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017), Washington, DC, USA, 30 May–3 June 2017; pp. 650–657.
37. Shamrat, F.J.M.; Al Jubair, M.; Billah, M.M.; Chakraborty, S.; Alauddin, M.; Ranjan, R. A Deep Learning Approach for Face Detection using Max Pooling. In Proceedings of the 2021 5th International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, 3–5 June 2021; pp. 760–764.
38. Ding, R.; Su, G.; Bai, G.; Xu, W.; Su, N.; Wu, X. A FPGA-based accelerator of convolutional neural network for face feature extraction. In Proceedings of the 2019 IEEE International Conference on Electron Devices and Solid-State Circuits (EDSSC), Xi'an, China, 12–14 June 2019; pp. 1–3.
39. Tufail, A.B.; Ma, Y.K.; Kaabar, M.K.; Martínez, F.; Junejo, A.; Ullah, I.; Khan, R. Deep learning in cancer diagnosis and prognosis prediction: A minireview on challenges, recent trends, and future directions. *Comput. Math. Methods Med.* **2021**, 9025470. [CrossRef] [PubMed]
40. Mehmood, M.; Shahzad, A.; Zafar, B.; Shabbir, A.; Ali, N. Remote Sensing Image Classification: A Comprehensive Review and Applications. *Math. Probl. Eng.* **2022**, 5880959. [CrossRef]
41. Tufail, A.B.; Ullah, I.; Khan, W.U.; Asif, M.; Ahmad, I.; Ma, Y.K.; Khan, R.; Kalimullah; Ali, S. Diagnosis of diabetic retinopathy through retinal fundus images and 3D convolutional neural networks with limited number of samples. *Wirel. Commun. Mob. Comput.* **2021**, 6013448. [CrossRef]
42. Ray, A.K.; Bagwari, A. IoT based Smart home: Security Aspects and security architecture. In Proceedings of the 2020 IEEE 9th International Conference on Communication Systems and Network Technologies (CSNT), Gwalior, India, 10–12 April 2020; pp. 218–222.
43. Khan, I.; Wu, Q.; Ullah, I.; Rahman, S.U.; Ullah, H.; Zhang, K. Designed circularly polarized two-port microstrip MIMO antenna for WLAN applications. *Appl. Sci.* **2022**, *12*, 1068. [CrossRef]
44. Bradski, G.; Kaehler, A. *Learning OpenCV: Computer Vision with the OpenCV Library*; O'Reilly Media, Inc.: Sebastopol, CA, USA, 2008.
45. Huang, Y.; Wang, Y.; Tai, Y.; Liu, X.; Shen, P.; Li, S.; Li, J.; Huang, F. Curricularface: Adaptive curriculum learning loss for deep face recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 5901–5910.
46. Yousafzai, B.K.; Khan, S.A.; Rahman, T.; Khan, I.; Ullah, I.; Ur Rehman, A.; Baz, M.; Hamam, H.; Cheikhrouhou, O. Student-performulator: Student academic performance using hybrid deep neural network. *Sustainability* **2021**, *13*, 9775. [CrossRef]
47. Wang, J.; Li, Z. Research on face recognition based on CNN. In Proceedings of the IOP Conference Series: Earth and Environmental Science, Banda Aceh, Indonesia, 26–27 September 2018; Volume 170, p. 032110.
48. Aydin, I.; Othman, N.A. A new IoT combined face detection of people by using computer vision for security application. In Proceedings of the 2017 International Artificial Intelligence and Data Processing Symposium (IDAP), Malatya, Turkey, 16–17 September 2017; pp. 1–6.
49. yzimm. Parameters and Flops in Convolutional Neural Network CNN. Available online: https://chowdera.com/2021/04/20210420120752555v.html (accessed on 7 June 2022).
50. Sajid, M.; Ali, N.; Ratyal, N.I.; Usman, M.; Butt, F.M.; Riaz, I.; Musaddiq, U.; Aziz Baig, M.J.; Baig, S.; Ahmad Salaria, U. Deep learning in age-invariant face recognition: A comparative study. *Comput. J.* **2022**, *65*, 940–972. [CrossRef]
51. Shabbir, A.; Ali, N.; Ahmed, J.; Zafar, B.; Rasheed, A.; Sajid, M.; Ahmed, A.; Dar, S.H. Satellite and scene image classification based on transfer learning and fine tuning of ResNet50. *Math. Probl. Eng.* **2021**, *2021*. [CrossRef]
52. Wang, F.; Chen, L.; Li, C.; Huang, S.; Chen, Y.; Qian, C.; Loy, C.C. The devil of face recognition is in the noise. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 765–780.
53. Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 20–26 June 2005; Volume 1, pp. 886–893.
54. Minhas, R.A.; Javed, A.; Irtaza, A.; Mahmood, M.T.; Joo, Y.B. Shot classification of field sports videos using AlexNet Convolutional Neural Network. *Appl. Sci.* **2019**, *9*, 483. [CrossRef]
55. Guo, G.; Wen, L.; Yan, S. Face authentication with makeup changes. *IEEE Trans. Circuits Syst. Video Technol.* **2013**, *24*, 814–825.

56. Tripathi, R.K.; Jalal, A.S. Make-Up Invariant Face Recognition under Uncontrolled Environment. In Proceedings of the 2021 3rd International Conference on Signal Processing and Communication (ICPSC), Coimbatore, India, 13–14 May 2021; pp. 459–463.
57. Wang, S.; Fu, Y. Face behind makeup. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016.
58. Deng, J.; Guo, J.; An, X.; Zhu, Z.; Zafeiriou, S. Masked face recognition challenge: The insightface track report. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 1437–1444.