IJE TRANSACTIONS A: Basics Vol. 37 No. 01, (January 2024) 25-36

International Journal of Engineering

Journal Homepage: www.ije.ir

# Smart City Surveillance: Edge Technology Face Recognition Robot Deep Learning Based

A. Medjdoubi*, M. Meddeber, K. Yahyaoui

*Faculty of Exact Science, Department of Computer Science, University of Mustapha Stambouli, Mascara, Algeria*

*PAPER INFO*

*Paper history:*
Received 20 July 2023
Received in revised form 10 September 2023
Accepted 16 September 2023

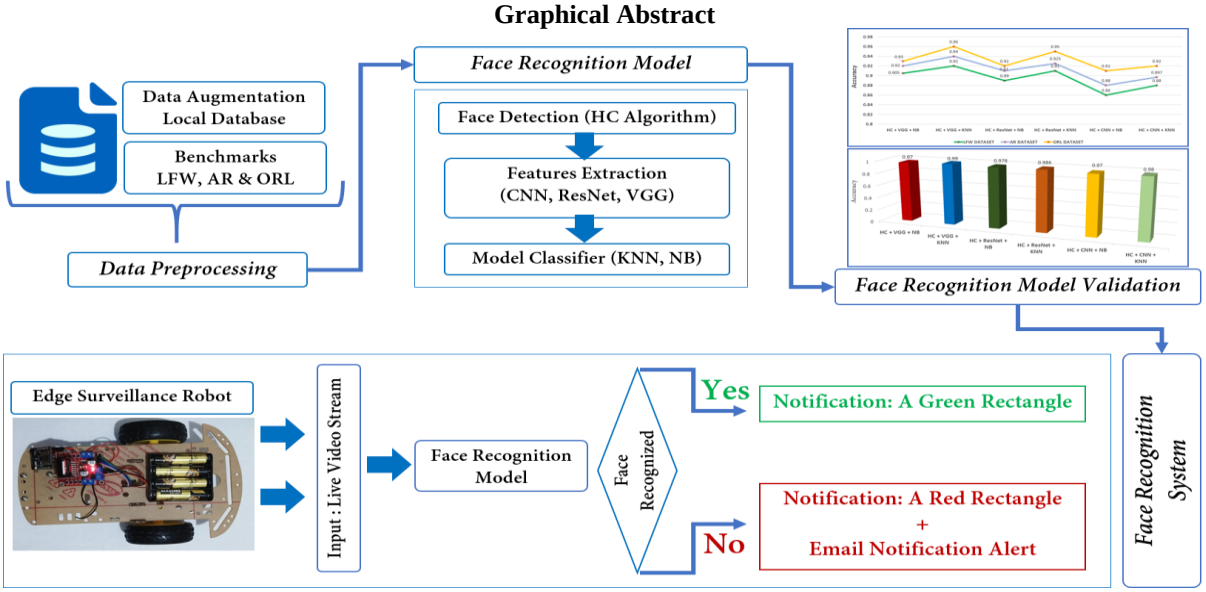
*Keywords*:
*Convolutional Neural Network*
*Deep Learning*
*Edge Technology*
*Face Recognition*
*Smart City*
*Security System*

*ABSTRACT*

In the contemporary context, the imperative to strengthen security and safety measures has become increasingly evident. Given the rapid pace of technological advancement, the development of intelligent and efficient surveillance solutions has garnered significant interest, particularly within the realm of smart city (SC). Surveillance systems have been transformed with the emergence of edge technology (ET), the Internet of Things (IoT), and deep learning (DL) to become key components of SC, notably the domain of face recognition (FR). This work introduces a smart surveillance car robot based on the ESP32-CAM micro-controller, coupled with a FR model that combines DL models and traditional algorithms. The Haar-Cascade (HC) algorithm is employed for face detection, while feature extraction relies on a proposed convolutional neural network (CNN) and predifined DL models, VGG and ResNet. While the classification is made by two distinct algorithms: Naive Bayes (NB) and K-nearest neighbors (KNN). Validation experiments demonstrate the superiority of a composite model comprising HC, VGG, and KNN, achieving accuracy rates of 92.00%, 94.00%, and 96.00% on the LFW, AR, and ORL databases, respectively. Additionally, the surveillance car robot exhibits real-time responsiveness, including email alert notifications, and boasts an exceptional recognition accuracy rate of 99.00% on a custom database. This ET surveillance solution offers advantages of energy efficiency, portability, remote accessibility, and economic affordability.

***doi***: *10.5829/ije.2024.37.01a.03*

**Graphical Abstract**




*Corresponding Author Email: abdelkader.medjdoubi@univ-mascara.dz (A. Medjdoubi)

Please cite this article as: Medjdoubi A, Meddeber M, Yahyaoui K. Smart City Surveillance: Edge Technology Face Recognition Robot Deep Learning Based. International Journal of Engineering, Transactions A: Basics, 2024;37(01): 25-36.

## 1. INTRODUCTION

Technology is undergoing rapid and continuous transformation, exerting a profound impact on our daily lives. Our society is increasingly shaped by innovation, scientific progress, and the pervasive deployment of artificial intelligence (AI) in practical applications, including smart automobiles, robotic systems, and wearable devices like smartwatches and smartphones. In this context, ensuring security and safeguarding individuals have assumed unparalleled significance, given the extensive exchange and generation of data by this technological wave (1). The imperative of ensuring the safety and well-being of individuals has become an essential facet of modern life, particularly in the context of SCs. The overarching objective of SCs is to enhance the overall quality of life for their inhabitants, with a particular emphasis on security. This goal is achievable through the integration of advanced technologies, such as intelligent surveillance systems that incorporate cutting-edge components like smart video cameras, sensors, and data analytics techniques (2). The demand for surveillance systems is expanding, given their integral role in safeguarding individuals, preserving public safety, and facilitating efficient urban governance. As SCs continue to expand, the adoption of edge technology (ET) presents new opportunities and advantages. ET enables data processing and analysis in proximity to the data source, as opposed to relying solely on centralized systems. It is the convergence of ET and IoT that has reshaped the lives of millions, simplifying and fortifying virtually every aspect of their daily routines (3). ET, also referred to as edge computing, represents a paradigm where certain services are delivered in close proximity to, or directly on, the devices that initiate the requests (4, 5). This paradigm has contributed significantly to the transformative impact of technology on contemporary society. It provides a variety of advantages; we listed below some main points among them :

- Real-Time Response and Decision Making: ET enables immediate data analysis at the data source, enhancing public safety through rapid processing and response (6, 7).
- Scalability and Flexibility: Distributed architecture accommodates data growth, expands surveillance coverage, and supports real-time video analytics (8).
- Resilience and Reliability: ET's decentralization ensures uninterrupted surveillance, even during network disruptions or emergencies (6, 8).
- Cost Efficiency: ET reduces reliance on expensive cloud resources, optimizing resource use and lowering operational expenses (8).

Many and various industrial security solutions are based on these IoT, ET, and DL advanced technologies. Taking the example from literature (9), a DL-based blockchain-driven scheme for SC security. I this work, the blockchain aspect is employed in a distributed manner at the fog layer to secure the integrity, decentralization, and security of manufacturing data. DL is applied at the cloud layer to boost productivity, automate data processing, and increase the communication bandwidth of smart factory and smart manufacturing applications. They give a case study of vehicle production with the newest service scenarios for the proposed scheme and a comparison to previous research studies utilizing critical characteristics such as security and privacy tools. The solution comprises five layers: a device layer for data collection using physical IoT devices; an edge layer that contains industrial gateways for data communications; a cyber-layer blockchain-based to verify and validate the data; a data analytics layer for data processing and analysis using DL to extract knowledge; and an application layer that employs the knowledge and the results to realize self-management, self-automation, scalable production, and rapid development in smart manufacturing.

Providing solutions that incorporate IoT and ET into the surveillance aspect is the need of the hour and what the researchers and scientists are striving to do, since it’s characterized by various advantages but mainly high efficiency and a reasonable cost (2). For that, video surveillance is one of the most crucial and fundamental parts of ensuring security for SC. Providing new tools and methods that enable people to monitor, control, and increase the stability of the city with fewer human errors (2). This paradigm shift centers around real-time video streaming and data generation through intelligent ET and IoT devices, increasing image processing, collecting more vision data to analyze every day, and information extraction. Consequently, the visual component emerges as an essential element for smart biometric security applications (10). In particular, the FR systems, that play a significant role in enabling the identification of individuals based on their physiological, physical, and behavioral attributes (11). The human face, in this context, stands as a prominent element in biometric identification and verification. This method boasts several advantages, including user friendliness, contactless operation, remote applicability, and real-time decision-making (11, 12). The evolution of FR systems has witnessed substantial progress, transitioning from controlled environments to the processing of faces captured in uncontrolled settings and extending from traditional feature extraction techniques to the adoption of DL methods (8).

Various works and studies have been put out by researchers as for FR IoT and DL-based solutions. DL-based intelligent FR in an IoT-cloud environment is proposed by Masud et al. (12). This research suggests a tree-based DL for automatic FR in a cloud setting. As mentioned in the paper, the proposed deep model is

computationally less expensive without affecting accuracy. In the model, an input volume is split into numerous volumes, and a tree is produced for each volume. A tree is described by its branching factor and height. Each branch is represented by a residual function, which is made of a convolutional layer, a batch normalization, and a non-linear function. The proposed model is examined in standard databases. A comparison of performance is also done with state-of-the-art deep models for FR. The results of the experiments reveal accuracies of 98.65%, 99.19%, and 95.84% on the FEI, ORL, and LFW databases, respectively. Kumar et al. (13) suggested a framework using visual recognition for IoT-based SC surveillance. In this manuscript, a quick subspace decomposition over Chi Square transformation is proposed. This technique extracts the characteristics of visual data using a local binary pattern histogram (LBPH). The redundant features are removed by using rapid subspace decomposition over the Gaussian-distributed Local Binary Pattern (LBP) features. As mentioned in the paper, the redundancy removal is a major contribution to memory and time consumption for battery-powered surveillance systems, which makes the methodology suitable for all image recognition applications and deployment into IoT-based surveillance devices due to improved dimension reduction. The technique was implemented on the Raspberry Pi and validated over well-known databases. The least error rate is attained by the suggested technique with maximal feature reduction in minimum time, with 0.28%, 1.30%, 9.00%, and 2.5% rates over AR, O2FN, LFW, and Dyn Tex++ databases, respectively. Other FR-DL-based solution is presented by Charoqdouz and Hassanpour (14). The goal of this study is to estimate the feature vector of a full-face image when there are numerous angular facial images of the same individual. This method extracts the essential elements of a facial image using the non-negative matrix factorization (NMF) method. Then, the feature vectors are merged using a generative adversarial network (GAN) to estimate the feature vector associated with the frontal image. The experiments were made on the FERET dataset, which contains angular pose images with an angle of up to 40 degrees. As concluded by the research, the proposed strategy can greatly increase the accuracy of FR technology. Shahbakhsh and Hassanpour (15) provided a solution to increase the image resolution at the feature level and improve the accuracy of FR methods. The methodology is based on extracting the edges of the face image using LBP and unsharp masking techniques and then adding them to the input image as a preprocessing that can help the used GAN model extract the high-frequency information of the low-resolution data images. The GAN examines picture edges and reconstructs high-frequency information to preserve the facial structure. The experiments were made on the FERET database along with images from other datasets such as MUCT, FEI, and Face94. The results demonstrated a considerable influence on the accuracy of FR in low-resolution photos compared to other state-of-the-art FR technologies.

DL, as a transformative facet of AI, has demonstrated remarkable performance not only in FR systems but also in various signal processing applications such as speech recognition, image recognition, and video recognition (16). Leveraging the inherent advantages of DL, IoT, and ET to construct intelligent FR security applications is not merely a contemporary pursuit but a pressing necessity. The fusion of these technologies offers the promise of delivering high-performance efficiency, aligning with the evolving needs of modern urban security (16).

The major contributions in the paper are: first, implementing a complex DL-FR system capable of making real-time decisions and predictions close to the IoT origin data source. Secondly, presenting a smart surveillance solution with a real-world ET robot application. Finally, we propose a SC surveillance solution that offers several advantages: low economic cost, less power consumption, portability, and remote utility.

The paper is divided into four sections. The first section described above highlights and discusses the research problem and presents the literature survey. The second section explains what the FR system is, what its essential phases are, and how it is functioning. The third section illustrates the suggested methodology based on approaches, techniques, and the materials used. The fourth section of the paper presents the experimental setup, results, and discussion. The last section concludes the paper.

## 2. FACE RECOGNITION SYSTEM

Over the course of the previous three decades, FR has garnered significant attention as an efficient image analysis and pattern recognition tool (17, 18). This heightened interest is largely attributable to the advancements in relevant technologies, which have facilitated a diverse range of commercial applications across various domains. These domains encompass transportation, where FR is employed for person identification and passport verification; law enforcement, where it aids in tracking suspicious individuals and identifying criminals; smart homes, which utilize FR for intelligent access control; healthcare, with the implementation of patient tracking systems; and marketing, where FR facilitates smart payment and personalized advertising (14, 18, 19).

The functionality of the FR system entails the creation of additional subtasks in a pipeline that are essential for its proper execution; an illustration of these subtasks is presented in Figure 1.

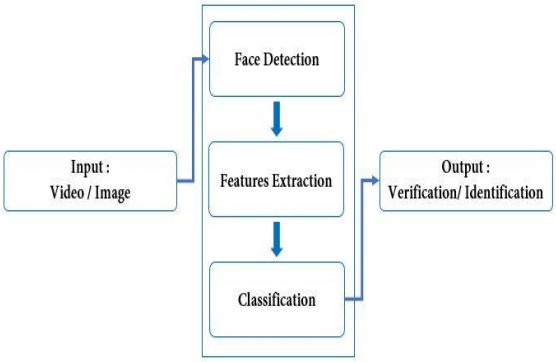



**Figure 1.** Face recognition system

- Face detection: It's the initial phase in the system; it shows if the image contains a face(s) or not by determining human facial location and dimensions. A lot of methods and algorithms are proposed for this task; some of them, among others, are the HC method by Viola and Jones (20) to identify facial characteristics with Haar-like features. Dlib HOG (21), which employs support vector machines (SVM) in conjunction with histograms of oriented gradients. Dlib CNN (21) combines a maxi-mum-margin object detector with a CNN to extract facial features. Another method is FaceNet (22), which is also based on CNN.
- Features Extraction: A key step that follows the detection phase consists of extracting the features, also called the signatures, which must be sufficient to represent a human face in the form of a vector, as we can call it, encoding the face. It is necessary to examine the uniqueness and individuality of the human profile (23). It should be mentioned that this process can be completed in the face detection step (17). Many different techniques can do this operation, such as deep neural nets like ResNet (23), FaceNet, and OpenFace (24). A novel technique was introduced by Firouzi et al. (25). An optimized hexagonal canny edge detection method. Images were first transformed to a hexagonal grid, followed by hexagonal image filtering using a Gaussian filter. Then, the magnitude and direction of gradients are estimated on the three axes x, y, and z. In the next stage, non-maximum suppression is applied to the amplitude and direction of gradients. Finally, threshold selection, double thresholding, untrue edge tracking, and edge removal are done successively to reach edges in the hexagonal domain.
- Classification: The encoders or the face vectors from the previous phase are classified into one of the classes offered during the training to make the general decision of either a known or unknown person. So, at this final stage, we have the process of finding an image label with a level of confidence. It is based on using supervised learning algorithms, which are frequently employed to do this type of operation. At this point in the system outcome, we will be doing either identification by comparison to other faces in order to determine the person's identity or verification of the person, which entails matching one face to another for accomplishing tasks like guaranteeing access, for example.

## 3. METHODOLOGY

The suggested ET surveillance robot works via a web-based application applying a FR model based on DL. This section will describe all the aspects, including the main steps, methods, and tools used to build our system.

**3. 1. Face Recognition Study Proposal** The proposed FR models in this work were developed using multiple processes, as depicted in Figure 2. Each stage will be illustrated and discussed. Starting with data preparation and preprocessing, moving on to face detection, feature extraction, and classification. Each FR model will be a combination of several algorithms and DL models.

**3. 1. 1. Data Preprocessing** As we can see in Figure 2, the data section is divided into two parts: one for benchmarks and the other for a local dataset. Benchmarks are a common database used for FR problems; in this work, we chose three of them: the LFW, the ORL, and the AR database. Such databases are for validating our FR-proposed models. The first one is the LFW[1] database, which has 13233 images of 5749 people. The second is the AR[2] database, which contains over 4,000 images corresponding to 126 people's faces. The third is the ORL[3] database, which comprises 400 images, represented by 40 subjects that have 10 photos per subject.

Moving to the second part of data preparation, in order to acquire a successful, correct outcome, a substantial amount of training data is required, but on the other hand, it is difficult and challenging work to do in a real-time context. Therefore, we create our own database using the data augmentation approach. It is utilized to generate more and new samples by modifying the existing data images. This database is used for testing our proposed FR models to be deployed with surveillance car robot. Therefore, we combine three different human face pictures together; two of them were chosen randomly from the AR database, and the third was from the author

[1] https://www.ece.ohio-state.edu/~aleix/ARdatabase.html
[2] https://cam-orl.co.uk/facedatabase.html
[3] http://vis-www.cs.umass.edu/lfw/

itself. Each image will be labeled with a class name, "Class_0", Class_1," and Class_2," as shown in Figure 3.

The augmentation process is made by applying different operations as described and listed in Table 1. Using this strategy, we built a local database containing 2430 face images with 810 data images per class.

Reshaping all the images to 64 by 64 grayscale photos is another preprocessing step to guarantee that all the data have the same shape. Next, a normalization strategy will be employed to verify that all data are in the same range.

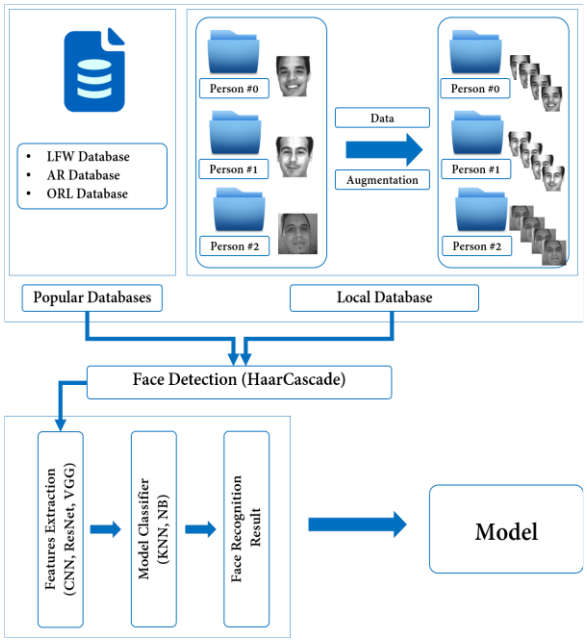



**Figure 2.** The proposed face recognition models

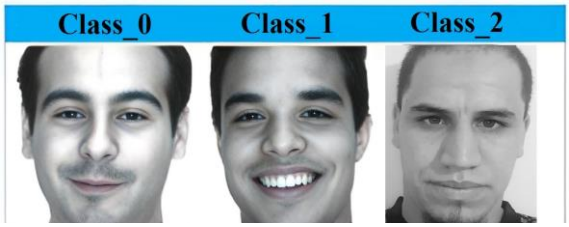



**Figure 3.** Local database classes labelling

**TABLE 1.** Data augmentation operations

| Parameter | Interval or Value |
|---|---|
| Zoom | [-5, 5] |
| Shear | 0.5 |
| Height shift | 0.3 |
| Width shift | 0.3 |
| Rotation | [0, 15] |

[1] https://github.com/fchollet/keras

**3. 1. 2. Face Detection** After preparing four distinct databases, a second step of face detection took place using HC. This algorithm uses a collection of predetermined features based on the foundation of Haar-like features, including edges, lines, and corners, as illustrated in Figure 4, to identify facial features like the eyes, nose, and mouth. It takes the frontal face as an input that will be exposed to a cascade application of the classifier. The image is divided into portions before the classifier is applied to each area. If a region is recognized as a face, the cascade continues to the next phase; otherwise, it is discarded, and the next region is processed. Figure 5 represents an example of this process.

**3. 1. 3. Features Extraction** CNNs have changed the domain of computer vision (CV) by imitating the human visual system's hierarchical processing. CNNs learn and extract internal representations or features straight from raw picture data, making them highly effective for tasks like FR, image categorization, object detection, and more. In this part, we will discuss the training process of our DL models. Three different DL models will be applied; two of them are predefined: the VGG (26) and the ResNet model. The third one is a proposed CNN built using the TensorFlow (27) and Keras[1] frameworks. Our suggested CNN is composed of 13 layers and has an architecture as explained in Table 2.

Following normalization, feature extraction and redundancy reduction take place in a pair of

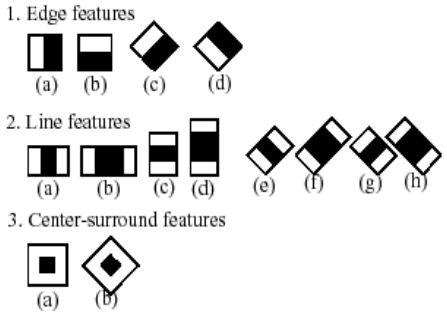



**Figure 4.** The Haar features used in the HC algorithm

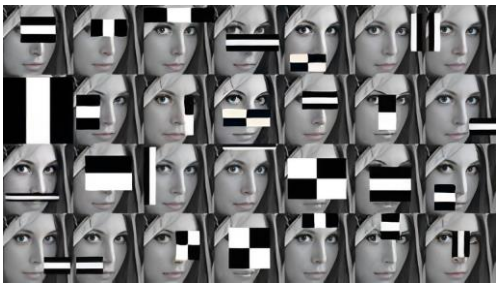


**Figure 5.** Haar features application over an image

**TABLE 2.** Proposed CNN model architecture

| Layer Type | Number of Layer |
|---|---|
| Two-Dimensional Convolutional Layer (Conv2D) | 04 layers |
| Batch Normalization Layer | 02 layers |
| Max Pooling Layer | 02 layers |
| Dense Layer | 03 layers |

convolutional layers coupled with pooling layers. Simple features come together efficiently over time. The features are then partially blended, with each resulting feature accounting for a portion of the configuration of the designated class. Finally, the fully connected layer receives these top attributes and delivers an estimation of the classification. To understand in more detail the process of feature extraction by the CNN, we have to illustrate each layer functionally. Figure 6 (28) shows a graphical representation of a CNN model where each layer is responsible for a specific operation.

Convolutional layers play the role of feature extractors. To generate feature maps, inputs are convolved with learned weights, and the results are then sent through a nonlinear activation function. Equation 1 specifies the output of the $kth$ feature map, denoted by $Y_{\mathrm{k}}$.

$$Y_{\mathrm{k}} = W_{\mathrm{k}} \copyright M \quad (1)$$

where:

- $W_k$ is the convolution filter.
- The sign "©" refer to the 2D convolution operation.
- $M$ refer to the input data image.

The batch normalization layer has the role of normalizing the input and tackling the vanishing gradient problems (16). It uses a normalization approach that takes place between the layers of the network instead of in the input data, in the form of mini-batches. By including additional layers, the learning process can be faster and more reliable. When a layer's input comes from a prior layer, the new layer standardizes and normalizes it.

For the pooling layer, as the name suggests, the operation of pooling or named subsampling also works as a bridge between several convolutional layers. The most popular pooling approaches in CNNs are max-pooling and average-pooling. It minimizes the spatial dimensions of the feature map, thereby lowering the computing process and aiding in translational invariance. In our study, we utilized the max pooling operation. It involves dividing the input feature map into regions called pooling windows and selecting the maximum value from each region. Equation 2 shows the process of this operation, where the output of this layer is denoted by $Y_{\mathrm{kij}}$ representing the $kth$ maximum element of a region from an image.

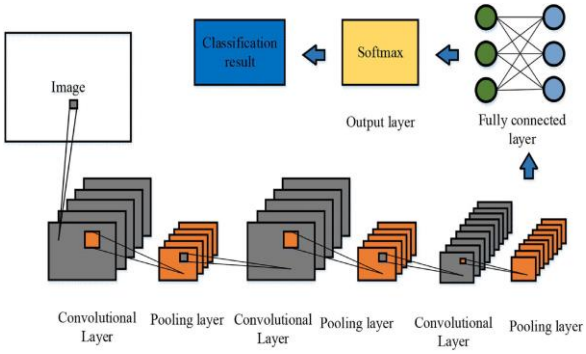



**Figure 6.** CNN model graphical representation

$$Y_{\mathrm{kij}} = maximum_{(\mathrm{p},\ \mathrm{q}) \in R_{\mathrm{ij}}} (M_{\mathrm{kpq}}) \quad (2)$$

where:

- $M_{\mathrm{kpq}}$ denotes $kth$ element at location $(p, q)$ from input data image $M$ contained by the $maximum$ of region $R_{\mathrm{ij}}$.

Figure 7 demonstrates even more this operation. A convolution layer's inputs are converted to the pooling layer's inputs. The greatest value in each 2x2 sub-area is mapped using a 4x4 mask.

After the feature extraction, the classification phase takes over, utilizing dense layers, which are completely connected layers employed to map the learned features to specific classes. These layers are critical for accurate classification and comprehending complicated relationships between features.

The training process for our DL models involved employing several parameters discussed as follows:

- Learning Rate: The learning rate controls how much the model's weights are changed during training. In our study, we fixed it to 0.001.
- Batch Size: Batch size defines the number of training samples utilized in each iteration of gradient descent, which in our case was 32.
- Number of Epochs: The number of epochs denotes the number of times the full training dataset is transmitted forward and backward through the network. We adjust it to 150.
- Optimizer: The optimizer governs the weight adjustments during training. Our DL models were trained using the RMSprop optimizer (29).

During the neural network training, a common challenge frequently addressed is the phenomenon known as overfitting. It occurs when the model learns to perform very well on the training data but fails to generalize to new, unknown input. As a solution to our work, three strategies are being applied, as given below:

- Regularization: One of the main methods employed is the integration of regularization techniques, such as L1 and L2 regularization. These methods contain penalty terms in the loss function, deterring the

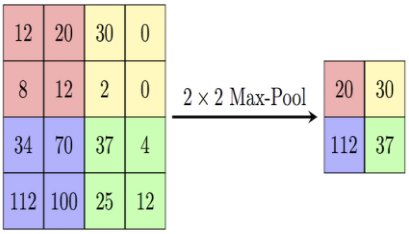



**Figure 7.** Illustrative example of max pooling operation

model from allocating excessive value to certain features or parameters. This, in turn, promotes a more balanced model. In our case, we use the L2 method.

- Dropout: We add some dropout regularization that entails the random deactivation of a fraction of neurons during training, effectively introducing an element of unpredictability and preventing co-adaptation among neurons. This enhances the network's ability to generalize by reducing reliance on specific neurons.
- Early Stopping: Monitor the validation accuracy during training and stop when it starts to drop. In order to apply it to our models, we use call-backs, an object that can perform actions at various stages of training. We fixed the monitoring operation for 30 consecutive epochs.

**3. 1. 4. Model Classifier** In this stage, we will use two supervised algorithms, the NB and the KNN, each at a time, to finish each FR model combination. The method is to classify the features encoded previously. The classification will be into one of the classes labeled with an amount of confidence, like a supervised classification strategy. It is about identifying and recognizing the face, and the result will be either a recognized person with a known label or an unknown person. After each training and classification phase, a model combination will be saved and exported for deployment purposes and to test our surveillance car robot. The best model of the six-model combination will be picked based on the accuracy results in the created database.

**3. 2. Edge Surveillance System Proposal** The suggested system in this study is predicated on the three parts detailed in the subsections below. The first one is the edge system, the second is the recognition system, and the third is the monitoring system, as indicated in Figure 8, where the functionality is as follows:

- Step 1: The edge surveillance robot captures the area with the camera module.
- Step 02: The recognition system captures the video transmitted by the robot, processes it, and extracts the images in real time.
- Step 03: If a face is detected, the FR model will take place and apply all the necessary steps to recognize it.
- Step 04: Presenting the outcomes on the live streaming web application, and also a notification email alert will be provided in case of unrecognized faces.

**3. 2. 1. Edge Surveillance System** Our proposed edge system is a surveillance car robot developed using several components, including:

- ESP32-CAM module.
- L298N Motor Driver
- Car robot that comes with 1 car chassis, 2 DC gear motors, 2 car tires, 1 universal wheel, 1 battery holder, and jumper wires.

Starting with the main module, the ESP32-CAM, which is a compact microcontroller and a low-cost ESP32-based development board featuring an inbuilt camera that works independently, powered by its own Wi-Fi and Bluetooth connectivity, among other specifications, as follows:

- Built in 520 KB of SRAM plus 4 MB of PSRAM.
- An SD card slot is supported.
- Built-in flash LED.
- The camera supports, which in our case, an OV5640 camera with a 5-megapixel image sensor.

The final result of the assembled car robot is represented in Figure 9.

After putting all the parts together and in order to make the system functioning, both for live streaming via the ESP32-CAM and for car robot control via the L298N motor drive, we have to program the ESP32-CAM board. Using the Arduino integrated development environment, we upload the main code written in C++ to synchronize the tasks between the components. The key functions are presented in Figure 10. Managing and using the robot will be done via a web application using a web browser.

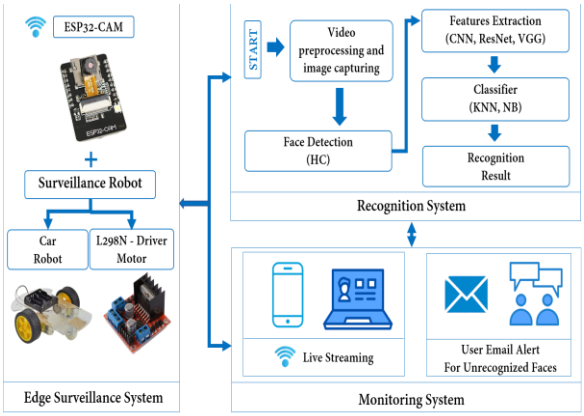



**Figure 8.** The proposed face recognition surveillance system

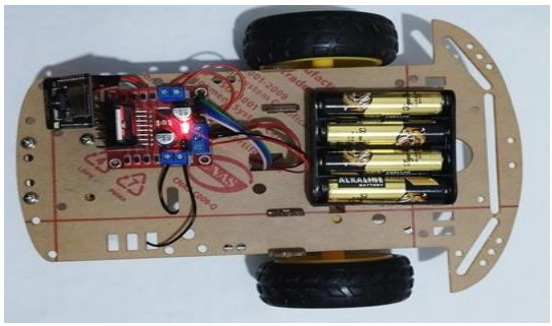


**Figure 9.** Edge surveillance car robot

**3. 2. 2. Recognition System** We applied a preprocessing step to the live streaming video provided through the ESP32-CAM module to extract the facial photos. Each video frame is turned into an image in order to identify the face inside it, after which features are extracted using DL models in order to classify them using the classification methods based on the confidence threshold.

**3. 2. 3. Monitoring System** The ability to drive the car and view the live streaming video using the same web application has been included as a final element in the proposed edge surveillance system, as shown in Figure 11. The buttons allow us to steer it in whatever direction we like while also managing the speed and lights. Our system has an email notification feature to alert users to unfamiliar faces as a security response.

```
void Car_Movement_control(int inputValue)
{
  Serial.printf("Got value as %d\n", inputValue);
  switch(inputValue)
  {

    case UP:
      rotateMotor(RIGHT_MOTOR, FORWARD);
      rotateMotor(LEFT_MOTOR, FORWARD);
      break;

    case DOWN:
      rotateMotor(RIGHT_MOTOR, BACKWARD);
      rotateMotor(LEFT_MOTOR, BACKWARD);
      break;

    case LEFT:
      rotateMotor(RIGHT_MOTOR, FORWARD);
      rotateMotor(LEFT_MOTOR, BACKWARD);
      break;

    case RIGHT:
      rotateMotor(RIGHT_MOTOR, BACKWARD);
      rotateMotor(LEFT_MOTOR, FORWARD);
      break;

    case STOP:
      rotateMotor(RIGHT_MOTOR, STOP);
      rotateMotor(LEFT_MOTOR, STOP);
      break;

    default:
      rotateMotor(RIGHT_MOTOR, STOP);
      rotateMotor(LEFT_MOTOR, STOP);
      break;
  }
}
```

```
void setup(void)
{
  setUpPinModes();
  Serial.begin(115200);

  WiFi.softAP(ssid, password);
  IPAddress IP = WiFi.softAPIP();
  Serial.print("AP IP address: ");
  Serial.println(IP);

  server.on("/", HTTP_GET, handleRoot);
  server.onNotFound(handleNotFound);

  wsCamera.onEvent(onCameraWebSocketEvent);
  server.addHandler(&wsCamera);

  wsCarInput.onEvent(onCarInputWebSocketEvent);
  server.addHandler(&wsCarInput);

  server.begin();
  Serial.println("HTTP server started");

  setupCamera();
}
```

```
void rotateMotor(int motorNumber, int motorDirection)
{
  if (motorDirection == FORWARD)
  {
    digitalWrite(motorPins[motorNumber].pinIN1, HIGH);
    digitalWrite(motorPins[motorNumber].pinIN2, LOW);
  }
  else if (motorDirection == BACKWARD)
  {
    digitalWrite(motorPins[motorNumber].pinIN1, LOW);
    digitalWrite(motorPins[motorNumber].pinIN2, HIGH);
  }
  else
  {
    digitalWrite(motorPins[motorNumber].pinIN1, LOW);
    digitalWrite(motorPins[motorNumber].pinIN2, LOW);
  }
}
```

**Figure 10.** Edge surveillance car robot main functions


**Figure 11.** Surveillance system web application

## 4. RESULTS AND DISCUSSION

The first part of our analysis will focus on the results of model validation toward benchmarks, while the second part will deal with the local data set that was developed to represent the results of the deployed model and the performance of the edge surveillance robot. The experiments made in this work, from training, testing, and validating the models, were using Google Colab [33], the free tier of the platform that delivers the needs with a system configuration as shown in Table 3.

**TABLE 3.** Google Colab platform free tier configuration

| Hardwar | Specifications |
|---|---|
| GPU | Nvidia Tesla K80 |
| CPU | 2x Intel Xeon @ 2.20GHz |
| RAM | 13 GB |
| HARD DISK | 78 GB |

The model's performance evaluation was based on the accuracy metric as described in Equation 3.

$$Accuracy = \frac{TN + TP}{TN + TP + FN + FP} \quad (3)$$

where: TN (true negative) are the examples correctly classified as negative class; TP (true positive) are the examples correctly classified as positive class; FP (false positive) are the examples incorrectly classified as positive class; and FN (false negative) are the examples incorrectly classified as negative class.

**4. 1. Face Recognition Model Validation Results** As demonstrated in Figure 12, evaluating the models over the LFW, AR, and ORL databases provides different outcomes in terms of accuracy.

**4. 1. 1. LFW Database** The training set was 90% from this database, while the rest was for the testing set with 1330 samples using the train-test cross-validation. Each one of the model's combinations has its own accuracy. We can observe from Figure 12 that the best model with the highest accuracy over this database is HC + VGG + KNN, with a value of 92.00%.

**4. 1. 2. AR Database** As the first database, we apply cross-validation to get 10% of the data for testing with 400 samples. Analyzing the results of each combination represented in Figure 12 it is noticed that the HC + VGG + KNN model combination fit the best, with a 94.00% accuracy rate.

**4. 1. 3. ORL Database** The results demonstrate the accuracy of each combination in this database with a testing set of 40 samples with the same percentage of 10% from the whole database. Also, we notice that the HC + VGG + KNN model combination is the best, with 96.00% accuracy.

**4. 1. 4. Validation Results Summary** As a sum up of the model's validation process, we conclude that the HC + VGG + KNN combination provides the highest accuracy, as presented in Table 4. Observing that our models are affected directly by the number of classes, we notice that when the number of classes decreases, the accuracy of the models will increase.

**4. 2. Surveillance System Results** In order to detail the suggested surveillance system results, we first represent the best model that was chosen for deployment into the system, where the combination was extracted after analyzing the model's accuracy results over the created database. Secondly, we represent the results of testing our surveillance car robot.

**4. 2. 1. Face Recognition Model Testing Results** The testing set had 10% of the database and 243 data

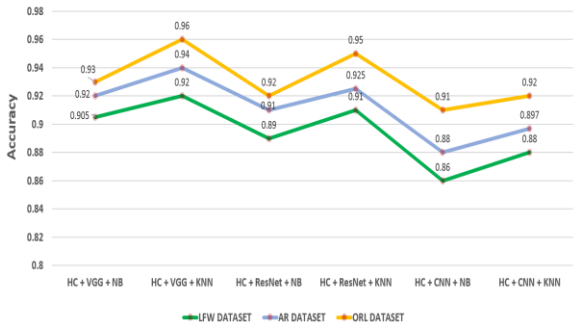



**Figure 12.** Model's performance over the benchmarks Model's performance over the benchmarks

**TABLE 4.** Comparison of the model's validation accuracy

| Models \ Databases | LFW with 5749 classes | AR with 126 classes | ORL with 40 classes |
|---|---|---|---|
| HC + VGG + NB | 90.50% | 92.00% | 93.00% |
| **HC + VGG + KNN** | **92.00%** | **94.00%** | **96.00%** |
| HC + ResNet + NB | 89.00% | 91.00% | 92.00% |
| HC + ResNet + KNN | 91.00% | 92.50% | 95.00% |
| HC + CNN + NB | 86.00% | 88.00% | 91.00% |
| HC + CNN + KNN | 88.00% | 89.70% | 92.00% |

samples. By observing the model performance, we found the best model, which consists of HC + VGG + KNN, has the highest accuracy rate of 99.00%. Starting with the confusion matrix represented in Table 5, the samples labeled "Class_0" and "Class_1", were all correctly classified, while the class labeled "Class_2" had two miss-classified examples.

We can observe in Figure 13 the performance of the other models in terms of accuracy results. Noting that the HC + ResNet + KNN model also gives a high accuracy of 98.76% with a small deference to the chosen model.

**4. 2. 2. Face Recognition Deep Learning Model Testing Results** To do comparison analysis, we apply another training phase with the same parameters as mentioned earlier on the same local database using the CNN, VGG, and ResNet models, without the NB and

**TABLE 5.** Confusion matrix of the HC+VGG+ KNN model on local database

| Actual Class \ Predicted Class | Class_0 | Class_1 | Class_2 |
|---|---|---|---|
| **Class_0** | 78 | 0 | 0 |
| **Class_1** | 0 | 83 | 0 |
| **Class_2** | 0 | 2 | 80 |

KNN classifiers. Aiming to observe the performance of these DL models since they can accomplish the classification task on their own. Figure 14 displays the test performance in terms of accuracy results. It indicates that this strategy of using the DL model for classification is superior to the approach of adding a traditional classification algorithm to the FR model. In terms of accuracy, the VGG outperforms the other models with a 99.15% rate, where we note 99.04% and 98.70% rates for the ResNet and the CNN, respectively.

To ascertain the robustness and effectiveness of each network architecture in terms of their performance, we added even more experiments based on accuracy and standard deviation for a series of experiments that were conducted repeatedly five times. The mean and standard deviation of the diagnostic accuracy are shown in Table 6. It can be concluded that the FR model based on HC and VGG outperforms all the other methods.

### 4. 2. 3. Surveillance Robot System Testing Results

The surveillance car robot created throughout the research has been examined in a real-world setting to test its efficiency. The system's reaction was in real time, with an average of 0.30 seconds. The live video streaming provided by the robot will be analyzed and processed by the HC + VGG + KNN model, where it has successfully predictions over the captured face images. Figure 15 shows a system response inside a house environment to a known case where the face was successfully recognized.

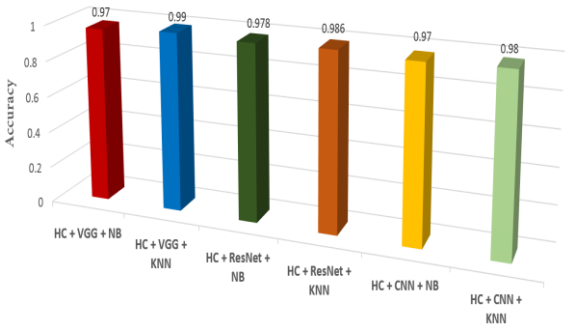



**Figure 13.** Model's performance over local database

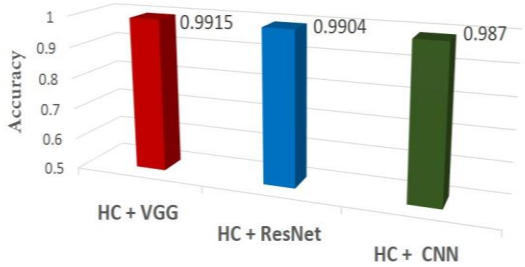



**Figure 14.** Deep learning model's performance over local database

**TABLE 6.** The mean and Standard deviation of accuracy results

| Metrics / FR Models | Accuracy | Standard deviation |
|---|---|---|
| HC + VGG | 99.22% | 0.10 |
| HC + ResNet | 99.004% | 0.12 |
| HC + CNN | 98.71% | 0.17 |

To do more evaluations for the surveillance robot, we made some changes to the face positioning. The model was unable to recognize the features since it was not in frontal positioning, as represented in Figure 16a. We put our edge robot in another situation by changing the environment outside the house in cloudy weather, where it works as it should in terms of movements, video streaming, and image capturing, but the illumination affected the processing operations and the model was unable to recognize the face. The case scenario is illustrated in Figure 16b.

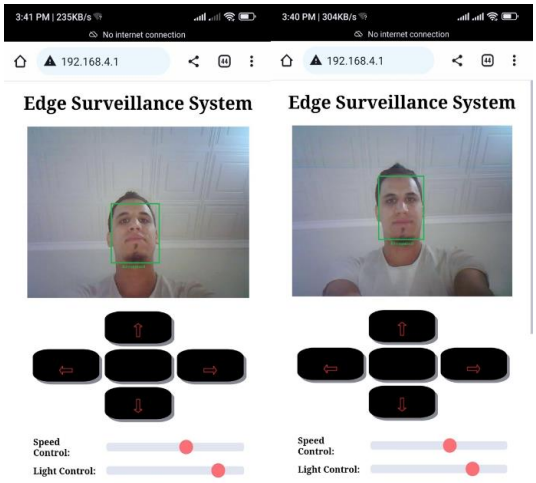



**Figure 15.** Edge system response for normal case scenario

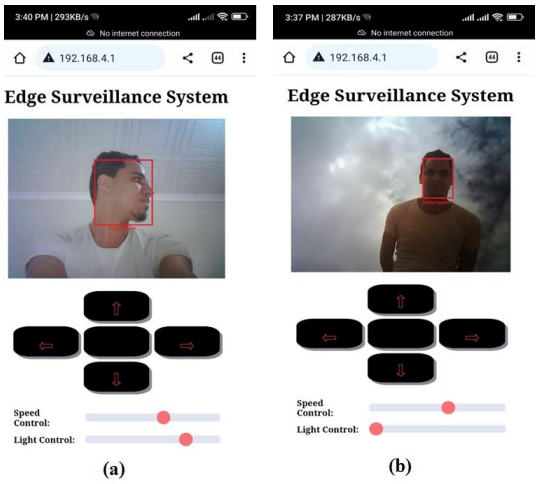



**Figure 16.** (a) Edge robot response for non-frontal face scenario. (b) Edge robot response with bad lighting scenario

## 5. CONCLUSION

In this study, we developed a FR system based on DL and ET. We have successfully built an edge car robot that works with an implemented system that consists of various algorithms and a DL approach for SC surveillance. A notification feature is included in the system to alert the user about intruders or unidentified people nearby via email address. Our suggested solution has several advantages, including being a low-cost, affordable solution, less power-consuming, lightweight, supporting remote usage, and accomplishing the task of recognition in real time with high efficiency. This paper also demonstrates how edge devices with specific computational capabilities may now run complicated AI models while maintaining effectiveness and efficiency in terms of results. Further work and more aspects need to be covered, such as lightning changes since they affect the prediction results, face positioning, and face expressions, to give the model the best performance possible.

## 6. REFERENCES


1. Bellavista P, Chatzimisios P, Foschini L, Paradisioti M, Scotece D, editors. A support infrastructure for machine learning at the edge in smart city surveillance. 2019 IEEE Symposium on Computers and Communications (ISCC); 2019: IEEE. 10.1109/iscc47284.2019.8969779
2. Ezzat MA, Abd El Ghany MA, Almotairi S, Salem MA-M. Horizontal review on video surveillance for smart cities: Edge devices, applications, datasets, and future trends. Sensors. 2021;21(9):3222. 10.3390/s21093222
3. Nauman A, Qadri YA, Amjad M, Zikria YB, Afzal MK, Kim SW. Multimedia Internet of Things: A comprehensive survey. Ieee Access. 2020;8:8202-50. 10.1109/access.2020.2964280
4. Jiménez-Bravo DM, Lozano Murciego Á, Sales A, Augusto Silva L, De La Iglesia DH. Edge Face Recognition System Based on One-Shot Augmented Learning. 2022. 10.9781/ijimai.2022.09.001
5. Shi W, Pallis G, Xu Z. Edge computing [scanning the issue]. Proceedings of the IEEE. 2019;107(8):1474-81.
6. Silva BN, Khan M, Han K. Towards sustainable smart cities: A review of trends, architectures, components, and open challenges in smart cities. Sustainable cities and society. 2018;38:697-713. 10.1016/j.scs.2018.01.053
7. Lai CS, Jia Y, Dong Z, Wang D, Tao Y, Lai QH, et al. A review of technical standards for smart cities. Clean Technologies. 2020;2(3):290-310. 10.3390/cleantechnol2030019
8. Al-Turjman F, Zahmatkesh H, Shahroze R. An overview of security and privacy in smart cities' IoT communications. Transactions on Emerging Telecommunications Technologies. 2022;33(3):e3677. 10.1002/ett.3677
9. Singh SK, Azzaoui A, Kim TW, Pan Y, Park JH. DeepBlockScheme: A deep learning-based blockchain driven scheme for secure smart city. Human-centric Computing and Information Sciences. 2021;11(12):1-13. 10.22967/HCIS.2021.11.012
10. Khan PW, Byun Y-C, Park N. A data verification system for CCTV surveillance cameras using blockchain technology in smart cities. Electronics. 2020;9(3):484. 10.3390/electronics9030484
11. Kortli Y, Jridi M, Al Falou A, Atri M. Face recognition systems: A survey. Sensors. 2020;20(2):342. 10.3390/s20020342
12. Masud M, Muhammad G, Alhumyani H, Alshamrani SS, Cheikhrouhou O, Ibrahim S, et al. Deep learning-based intelligent face recognition in IoT-cloud environment. Computer Communications. 2020;152:215-22. 10.1016/j.comcom.2020.01.050
13. Kumar M, Raju KS, Kumar D, Goyal N, Verma S, Singh A. An efficient framework using visual recognition for IoT based smart city surveillance. Multimedia Tools and Applications. 2021:1-19. 10.1007/s11042-020-10471-x
14. Charoqdouz E, Hassanpour H. Feature extraction from several angular faces using a deep learning based fusion technique for face recognition. International Journal of Engineering, Transactions B: Applications. 2023;36(8):1548-55. 10.5829/ije.2023.36.08b.14
15. Shahbakhsh MB, Hassanpour H. Empowering face recognition methods using a gan-based single image super-resolution network. International Journal of Engineering, Transactions A: Basics. 2022;35(10):1858-66. 10.5829/ije.2022.35.10a.05
16. Faisal F, Hossain SA, editors. Smart security system using face recognition on raspberry Pi. 2019 13th International Conference on Software, Knowledge, Information Management and Applications (SKIMA); 2019: IEEE. 10.1109/skima47702.2019.8982466
17. Adjabi I, Ouahabi A, Benzaoui A, Taleb-Ahmed A. Past, present, and future of face recognition: A review. Electronics. 2020;9(8):1188. 10.20944/preprints202007. 0479.v1
18. Hassanpour H, Ghasemi M. A three-stage filtering approach for face recognition. International Journal of Engineering. 2021;34(8):1856-64. 10.5829/ije.2021.34.08b.061
19. Kakarla S, Gangula P, Rahul MS, Singh CSC, Sarma TH, editors. Smart attendance management system based on face recognition using CNN. 2020 IEEE-HYDCON; 2020: IEEE. 10.1109/hydcon48903.2020.9242847
20. Viola P, Jones MJ. Robust real-time face detection. International journal of computer vision. 2004;57:137-54. 10.1023/B:VISI.0000013087.49260.fb
21. King DE. Dlib-ml: A machine learning toolkit. The Journal of Machine Learning Research. 2009;10:1755-8. 10.1145/1577069.1755843
22. Schroff F, Kalenichenko D, Philbin J, editors. Facenet: A unified embedding for face recognition and clustering. Proceedings of the IEEE conference on computer vision and pattern recognition; 2015. 10.48550/ARXIV.1503.03832
23. Kaur P, Krishan K, Sharma SK, Kanchan T. Facial-recognition algorithms: A literature review. Medicine, Science and the Law. 2020;60(2):131-9. 10.1177/0025802419893168
24. Baltrusaitis T, Zadeh A, Lim YC, Morency L-P, editors. Openface 2.0: Facial behavior analysis toolkit. 2018 13th IEEE international conference on automatic face & gesture recognition (FG 2018); 2018: IEEE. 10.1109/fg.2018.00019
25. Firouzi M, Fadaei S, Rashno A. A new framework for canny edge detector in hexagonal lattice. International Journal of Engineering. 2022;35(8):1588-98. 10.5829/ije.2022.35.08b.15
26. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:14091556. 2014. 10.48550/ARXIV.1409.1556
27. Developers T. TensorFlow (Zenodo, 2021). DOI.
28. Liu J. Convolutional neural network-based human movement recognition algorithm in sports analysis. Frontiers in psychology. 2021;12:663359. 10.3389/fpsyg.2021.663359
29. Tieleman T, Hinton G. Divide the gradient by a running average of its recent magnitude. coursera: Neural networks for machine learning. Technical report. 2017. https://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf

Persian Abstract

**چکیده**

در زمینه معاصر ، ضرورت تقویت اقدامات امنیتی و ایمنی به طور فزاینده ای آشکار شده است. با توجه به سرعت سریع پیشرفت تکنولوژیکی ، توسعه راه حل های نظارت هوشمند و کارآمد علاقه قابل توجهی را به خود جلب کرده است ، به ویژه در قلمرو شهر هوشمند (SC). سیستم های نظارتی با ظهور فناوری edge (ET) ، اینترنت اشیاء (IoT) و یادگیری عمیق (DL) به اجزای کلیدی SC تبدیل شده اند ، به ویژه حوزه تشخیص چهره (FR). این کار یک ربات ماشین نظارت هوشمند را بر اساس میکروکنترلر ESP32-CAM معرفی می کند ، همراه با یک مدل FR که مدل های DL و الگوریتم های سنتی را ترکیب می کند. الگوریتم Haar-Cascade (HC) برای تشخیص چهره استفاده می شود ، در حالی که استخراج ویژگی به یک شبکه عصبی کانولوشن پیشنهادی (CNN) و مدل های dl پیش تعریف شده ، VGG و ResNet متکی است. در حالی که طبقه بندی توسط دو الگوریتم متمایز ساخته شده است: ساده لوح Bayes (NB) و k-نزدیکترین همسایگان (KNN). آزمایشات اعتبارسنجی برتری یک مدل ترکیبی شامل HC، VGG و KNN را نشان می دهد و به ترتیب میزان دقت ۹۲.۰۰ ٪ ، ۹٤.۰۰ ٪ و ۹٦.۰۰ ٪ را در پایگاه داده های LFW، ar و ORL به دست می آورد. علاوه بر این ، ربات ماشین نظارتی پاسخگویی در زمان واقعی را نشان می دهد ، از جمله اطلاعیه های هشدار ایمیل ، و دارای نرخ دقت تشخیص استثنایی ۹۹.۰۰ ٪ در یک پایگاه داده سفارشی است. این راه حل نظارت ET مزایای بهره وری انرژی ، قابلیت حمل ، دسترسی از راه دور و مقرون به صرفه بودن اقتصادی را ارائه می دهد.