Received: 12 March 2023 | Revised: 26 April 2023 | Accepted: 9 May 2023
DOI: 10.1002/dac.5545

RESEARCH ARTICLE

WILEY

Convolutional neural network and unmanned aerial vehicle-based public safety framework for human life protection

Nihar Patel[1] | Nakul Vasani[1] | Rajesh Gupta[1] | Nilesh Kumar Jadav[1] | Sudeep Tanwar[1] | Fayez Alqahtani[2] | Amr Tolba[3] | Maria Simona Raboaca[4,5]

[1]Department of Computer Engineering, Institute of Technology, Nirma University, Ahmedabad, Gujarat, India

[2]Software Engineering Department, College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia

[3]Computer Science Department, Community College, King Saud University, Riyadh, Saudi Arabia

[4]Doctoral School, University Politehnica of Bucharest, Bucharest, Romania

[5]The National Research and Development Institute for Cryogenic and Isotopic Technologies—ICSI Rm., Râmnicu Vâlcea, Romania

Correspondence
Sudeep Tanwar and Rajesh Gupta, Department of Computer Engineering, Institute of Technology, Nirma University, Ahmedabad, Gujarat 382481, India.
Email: sudeep.tanwar@nirmauni.ac.in and rajesh.gupta@nirmauni.ac.in

Funding information
This work was funded by the Researchers Supporting Project Number (RSP2023R509), King Saud University, Riyadh, Saudi Arabia; Ministry of Research, Innovation, Digitization from Romania by the National Plan of R&D, Project PN 23 15 01 01, Subprogram 1.1. Institutional performance-Projects to finance excellence in RDI, Contract No. 19PFE/30.12.2021; and a grant of the National Center for Hydrogen and Fuel Cells (CNHPC)—Installations and Special Objectives of National Interest (IOSIN).

Summary

In this paper, we developed an object detection and identification framework to bolster public safety. Before developing the proposed framework, several existing frameworks were analyzed to bolster public safety. The other models were carefully observed for their strengths and weaknesses based on the machine learning and deep learning algorithms they operate on. All these were kept in mind during the development of the proposed model. The proposed framework consists of an unmanned aerial vehicle (UAV) utilized for data collection that constantly monitors and captures the images of the designated areas. A convolutional neural network (CNN) model is developed to recognize a threat and identifies various handheld objects, such as guns and knives, which facilitate criminals to commit crimes. The proposed CNN model comprises 16 layers with input, convolutional, dense, max-pool, and flattened layers of different dimensions. For that, a benchmarked dataset, that is, small objects handled similarly to a weapon (SOHAs), a weapon detection dataset is used. It comprises six classes of 8945 images, with 5947 used for training, 1699 used for testing, and 849 used for validation. Once the CNN model accomplishes the object identification and classification, that is, the person is criminal or non-criminal, the criminal is forwarded to various law enforcement agencies and non-criminal data are again forwarded to the CNN model for improvising its accuracy rate. As a result, the proposed CNN model outperforms several pre-trained models with an accuracy of 0.8352 and a validation accuracy of 0.7758. In addition, the proposed model gives a minimal loss of 0.83 with a validation loss of 0.97. The proposed framework decreases the burden on crime-fighting agencies and increases the accuracy of crime detection. Additionally, it ensures fairness and operates at a meager computational cost compared to similar pre-trained models.

KEYWORDS
convolutional neural network, deep neural network, human life protection, public safety, unmanned aerial vehicle

1 | INTRODUCTION

Public safety is a pivotal part of society, ensuring citizens' safety, and is the top priority of every government or regime. Each government has established various means and measures to safeguard its citizens, that is, article amendments in the law. Public safety contributes to a large part of the life index of any nation and is essential for its citizens' holistic and economic development. Various systems like deploying unmanned aerial vehicles (UAVs), surveillance cameras, and police patrolling are arranged by the governments to protect citizens from crimes. Nations also established various public safety organizations/agencies to protect citizens from crimes. But, for many nations (having large populations), ensuring public safety is quite challenging for many reasons. The major reason for this is the overburdened public safety-enhancing organizations. There isn't enough manpower per unit population for these organizations to work efficiently. So, for overly populated countries, fighting crime is considered too tedious for crime-fighting agencies. Sometimes, these agencies can also need more required infrastructure and funds. To add to these issues, the relationship between the common public and the officers of the aforementioned agencies could be more optimal too.

In the aforementioned view, several researchers presented various solutions that ensure public safety.[1–4] These solutions include using existing technologies like artificial intelligence (AI), deep learning (DL), blockchain technology, and cloud computing for crime identification and public safety. Among the aforementioned technologies, AI and DL are prominent in providing safety solutions by predicting or analyzing crime scenes.[5] Cloud computing makes it relatively simpler for crime-fighting agencies to work in synchronization with federal policies and helps avert security breaches and leakage of sensitive information. At the same time, there are many unsolved problems that cloud computing faces while dealing with public safety, like, compromised privacy and security, compliance with all rules and regulations, recovery of lost data in contingency, and high establishment costs. Neto et al[6] and Sultana and Wahid[7] proposed a fog-based video surveillance for security management and an Internet of Things (IoT)-based smart transportation system. The fog-based framework has been used to interact with IoT devices and the edge network with the remote cloud. It provides agile and accurate weapon detection, but the systems suffer from issues that would make them unwise to use practically. The drawbacks are high networking traffic, massive storage, and the required network resources. The systems are also solely dependent on the ability to stay connected to the cloud, which, if compromised, would make the whole system redundant. On the other hand, blockchain technology has a fair share of advantages and disadvantages when trying to improve public safety. Li et al[8] proposed a blockchain-based lawful evidence management scheme for digital forensics that support transparency, immutability, and auditability when managing such lawful evidence. Tsai[9] highlighted the application of blockchain technology for the criminal investigation process. Researchers utilize blockchain technology to aid digital forensic tasks concerning preliminary investigation, case management, and court phases. Blockchain provides improved safety, secure storage of data, and integrity while investigating, leading to enhanced public trust. But, simultaneously, it has problems like huge energy and cost requirements. Moreover, the blockchain has the benefit of immutability, which ensures the data, once recorded, cannot be deleted by anyone. UAVs can be pivotal to public safety, continuously sending live images of crime scenes or incident locations.[10] UAVs reduce human intervention and make the investigation process seamless and reliable. It can monitor streets from the air, making patrolling more efficient than humans.[11] Gur et al[12] and Karim et al[13] presented image processing-based approaches using UAVs to detect crimes on streets. Several things the authors have suggested, like using the latest graphic cards, using smaller drones for compact and congested streets, and integrating GPS systems to boost performance. They ensure the practicality of UAVs in detecting and preventing prevalent street crimes.[14]

Various aforementioned solutions related to public safety can be enhanced by integrating AI algorithms. It helps ease the citizens' safety and makes communities safe. It can support criminal investigations, predict organized crime by analyzing patterns, and help combat severe threats. He and Zheng[15] presented research that predicts crime rates in urban neighborhoods using the DL-based generative adversarial network (GAN) algorithm. The authors generate crime heat maps for a city and quickly examine the areas where the crime rate is too high. This helps law enforcement agencies to act. Their proposed model needs to pinpoint the location of a crime, which would leave officers short-handed. Further, the focus is on crime rate prediction, not protecting against crimes. So there is still scope for AI to analyze enormous amounts of data and reliably recognize patterns and connections between recorded crimes. Further, as per the literature, no existing approaches identify crimes based on the images captured from different sources and their locations. With these understandings, a gap between crimes and convictions can improve understanding of behavior patterns and sequences of events that contribute to crime.[16] This gives law enforcement agencies a head start in developing strategies to obstruct certain pathways related to public safety.

Motivated by the aforementioned gaps, this paper proposes a customized convolutional neural network (CNN)-based framework to assist law enforcement agencies in protecting crimes and ensuring public safety. The proposed framework detect street crimes by analyzing the captured images from the deployed UAVs and identifying patterns and sequence of crime events. UAVs constantly monitor citizens and the objects they carry. The captured images are fed as input to the proposed model to classify them into criminal or non-criminal categories by identifying the objects involved in the image. If the framework finds any weapons in the picture, the crime-fighting agencies are imminently informed to act and defend the public. The proposed model substantially decreases the burden on crime-fighting agencies while increasing the accuracy of crime detection and ensuring fairness as the process becomes automated. Additionally, it operates at a significantly lower computational cost than other pre-trained models performing a similar function, which makes it extremely cost-efficient.

1.1 | Research contributions

The following are the contributions of this paper.

- We proposed a CNN-based learning framework that assists law enforcement agencies in detecting crime. It decreases the burden such organizations face by reducing human intervention and potentially improves the accuracy of crime detection.
- To increase the efficiency and accuracy of the proposed framework, an optimized CNN is developed to avoid bias and ensure fairness in detecting crime patterns from the dataset. It updates the image database with every prediction by adding new instances to the dataset; this improves the overall detection rate of the CNN model.
- The computation cost and memory utilized by the proposed framework are significantly less than other pre-trained models like VGG19, Inception V3, and ResNet101, which makes it highly efficient and cost-effective to use. The proposed model's file size is also minuscule compared to other pre-trained models.
- Further, several evaluation metrics, such as accuracy, loss, and optimizer's performance, are used to evaluate and test the CNN model. The model is developed by running many image combinations, outperforming other trained models, such as VGG19, Inception V3, and ResNet101.

1.2 | Organization

The organization of this paper is as follows. Section 2 describes the related works in the field of public safety. Section 3 presents the system model problem formulation. Section 4 presents the proposed CNN-based framework for crime identification and detection. Section 5 emphasizes the result and discussion of the proposed framework. Finally, Section 6 concludes the paper.

2 | STATE OF THE ART

This section discusses the related work carried out by researchers in public safety and crime identification. The crime rate and detection have increased significantly with ever-changing societies and technologies. Traditional approaches for public safety record crime and classify the rate, which is duly based and dependent on the text and record-based systems. With technological advancements, crime detection and classification can also be performed on photos, proof of physically performed, and recorded crimes. Researchers have proposed many ML- and DL-based techniques to classify and predict crimes. For example, Tasnim et al[17] show the implementation of time-series analysis of the data using the long short-term memory (LSTM) model to explore and predict the data outcome. They have used natural-level processing to combine the output of different models into one for the prediction of the classes. However, the model has limited classes and cannot extend its capabilities if another class is added.

Then, Baek et al[18] proposed a novel approach for classification using support vector machine (SVM) and deep neural network (DNN) algorithms. Among these models, DNN gave a better error rate than SVM with better training accuracy. However, the model could not perform well in validating the trained model, thus overfitting to an extent. Safat et al[4] proposed a nascent approach that combined the features and capabilities of different models to predict the output considering many features of the dataset used with a good diagrammatic representation of insights. However, their approach is an exploratory dataset analysis without making a classification prediction of the required data. Kshatri et al[19] proposed an empirical analysis of the dataset using ML techniques, such as stacking, boosting, and ensemble classifiers for predicting and classifying the target attribute with the consideration of numerous predefined attributes. Their model reflected a better overall accuracy for the classification problem.

Han et al[20] presented an approach to forecasting cities with greater crime rates and predicting the outcome concerning the record of the past times. They have classified cities in the United States with selected attributes accurately. However, the training of the model is limited to certain variables only. Wang et al[21] proposed a model for the ACP approach with a considered construction of an artificial crime scene and gave an empirical analysis for the parallel crime scene. However, the model is too ideal and doesn't implicate the practical implementation. Zhang et al[22] presented an LSTM-based model that uses geospatial data for identifying adaptive attributes, which are further used for multi-class classification of cities with the greater crime rate and further prediction using several AI algorithms. However, appropriate accuracy is not achieved with their proposed model.

Mohammadpour et al[23] stress the significance of utilizing DL techniques in network intrusion detection systems because of the sophistication and increased complexity of contemporary cyberattacks. They examine a number of studies that used CNNs for NIDS, including those that used more modern architectures like residual networks and attention-based models, as well as models built using conventional convolutional layers. The authors also review many datasets utilized in CNN-based NIDS research, including the NSL-KDD and KDD Cup 1999 datasets. They discuss the problem of unbalanced datasets in NIDS and how some research has dealt with it using methods like data augmentation and oversampling.

Anand et al[24] examine several research that have applied CNNs and other DL techniques to network traffic and Android applications, among other contexts, to detect malware. They also talk about the drawbacks of these methods, such as the necessity for a lot of labeled data, the possibility of adversarial assaults, and the difficulty of understanding the judgments made by DL models. The authors also discuss recent advancements in DL-based malware detection, including ensemble approaches and transfer learning. They also emphasize how crucial it is to consider the resource limitations of healthcare apps, such as their constrained memory and processing capabilities, when constructing DL models for malware detection. Table 1 describes the relative comparison of state of the art approaches with the proposed one.

Through the comprehensive understanding and realization of all the recent approaches, the appropriate and optimized use of CNN was not implemented in a dynamic and real-time environment. The approaches were static to the available data without continuous updates for data generated over time. Thus, the proposed framework is a customized CNN classification and prediction model, which combines the working of DNN and CNN models in making appropriate predictions on the image datasets with multiple classes. The proposed framework identifies features of different images from the respective classes and extracts unique attributes related to the particular image. The parameters are updated and controlled with appropriate values to achieve the best result, such that possibilities of vanquishing and exploding gradients are reduced to a minimum. The framework is also open for fine-tuning over the layers for further adaptability with the dynamics of the evolving environment. Thus, combining CNN and DNN with tuned hyperparameters improves the training and validation accuracies with a justified loss convergence. The model is adaptive overall if additional classes are to be added later for training and classification purposes.

3 | SYSTEM MODEL AND PROBLEM FORMULATION

This section presents the system model and problem formulation of the proposed DNN- and CNN-based framework. Despite recent improvements in the public safety domain, several crimes are still committed and unable to predict and classified priorly. Although the crime rates are decreasing, the grassroots reality is gloomy. The most common crimes that happen every day include larceny or theft, burglary, aggravated assault, and robbery. The proposed system model deals with this problem and aims to reduce crime rates and increase public safety. In the proposed system, the civilians set is represented as $\{U_1, \ldots, U_i, \ldots, U_j, \ldots, U_m\} \in U$, where a civilian can be a criminal or a non-criminal based on whether they are carrying weapons (probable criminal) or not, whereas the set $\{K_1, \ldots, K_i, \ldots, K_j, \ldots, K_m\} \in K$ represents the weapons set. This criminal activity is captured by UAVs, $\{D_1, \ldots, D_i, \ldots, D_j, \ldots, D_m\} \in D$, which are deployed by agencies responsible for ensuring public safety.

TABLE 1 Comparative analysis of the proposed framework with the existing state-of-the-art schemes for public safety.

Authors	Year	Objective	Algorithms used	Pros	Cons
Tasnim et al.[17]	2022	Presented a transfer learning for predicting crime rates of two different cities in the United States (US) by performing exploratory data analysis on the data, returning better results than the current state of the arts	LSTM and NLP	Good precision and recall values	Limited to a small number of predefined classes
Baek et al.[18]	2021	Presented a combinational technique for making real-time predictions by employing CNN and DNN in the model architecture, overtaking the conventional approach of using SVM and Naive Bayes for ML predictions	CNN and SVM	Good training accuracy	Overfitting for large datasets
Safat et al.[4]	2021	A comparative method for forecasting the crime rates of two major cities of the US by performing time-series analysis on the dense data to show the versatility of DL and ML algorithms to predict promising results	LSTM and ARIMA	Fusion model for classification and time-series analysis	Performance limited to EDA with limited scope
Kshatri et al.[19]	2021	Among many established ML algorithms available in the domain, an efficient architecture performing well in both metrics of measure for time complexity and better accuracy than others	Ensemble and boost classifiers	Good overall accuracy	Limited to a small number of predefined classes
Han et al.[20]	2020	Identify the recurrent patterns of crimes occurring in geospatial regions using LSTM network and classify different regions with their respective crime rates over time	LSTM and STGCN	Optimized classification using spatiotemporal data	No accuracy measure and limited to a small number of predefined classes
Wang et al.[21]	2018	Incorporating human behavior characteristics and another stochastic factor for comparative predictions of the crime scene using artificial societies, computational experiments, and parallel execution (ACP) approach	ACP approach	Good software-defined analysis	Excessively theoretical for practical usage
Zhang et al.[22]	2020	Incorporate various ML algorithms for predicting the major crime hot spots and depicting the visual analysis of the predicted results	LSTM, KNN, random forest, naive Bayes, and CNN	Geospatial data used for classification; comparing models	High computational resources used and low accuracy
The proposed work	2023	Optimized CNN-based approach to enhance public safety	CNN and DNN model	Fusion of CNN and DNN; good training and validation accuracies	-

Equations (1) and (2) represent that a civilian can possess one or more weapons.

$$\exists U_i \in U \overset{has}{\rightarrow} K_i \in K \Leftrightarrow \{U_i, K_i\}, \tag{1}$$

$$\exists U_i \overset{has}{\rightarrow} \{K_2, K_3, ..., K_j\} \in K. \tag{2}$$

In Equations (3) and (4), if a civilian possesses a weapon, he is categorized as a criminal; otherwise, non-criminal.

$$U_i \cap K_i \neq \emptyset \rightarrow Criminal, \tag{3}$$

$$U_i \cap K_i = \emptyset \rightarrow Non\text{-}Criminal. \tag{4}$$

UAVs capture criminal acts and send these images to public safety agencies for further action, represented using Equation (5).

$$\{U_i, K_i\} \neq \emptyset \rightarrow D_i. \tag{5}$$

As mentioned in Section 2, many machine learning algorithms are already applied to ensure public safety and predict the actions required to reduce crime rates, although many of the prescribed algorithms are inefficient to work in a real-time environment. The need for a CNN algorithm is highlighted to be integrated into the UAVs to identify and detect real-time on-sight criminal activities. As its dynamics of detecting and classifying multiple frames of images give greater efficiency and flexibility for predictions in the real-time environment.

The CNN architecture variably changes its computation weights to ensure optimum accuracy. The CNN dynamics are based on the concept of back-propagation through the layered computation and reconfiguring the weights of the kernels, which are applied to input images decreasing the error over the span of time. Thus, in real time, the approach of using CNN does not take a long time to predict and classify the activities captured by the UAV.

The objective of the proposed system model is to identify civilians U_i who possess weapons K_i and report them to law enforcement agencies. Existing crime detection schemes work and catch criminals, but not at an efficient rate, as it involves human discretion instead of technology. There are latency issues in the current systems and no proper weapons detection. The proposed framework offers accurate weapon detection and crime prediction. Another advantage this provides is that it can also define the intensity of the crime, which would make it easy for law enforcement authorities to prioritize in case of multiple crimes being committed.

In this viewpoint, the proposed framework aims to maximize the weapon detection rate from the captured images at the crime scene. Equation (8) shows the objective function of the proposed framework, where $\mathcal{F}$ is the objective function, whereas Y_{Z_i} portrays different instances of data requests and Ψ represents the CNN model.

$$X_1 = \max \sum_{i=1}^{m} \text{Detection}_\text{R}\text{ate}(\Psi), \tag{6}$$

$$X_2 = \max \text{Secure}(Y_{Z_i}), \tag{7}$$

$$\mathcal{F} = X_1 + X_2. \tag{8}$$

$$\begin{aligned} & s.t. \\ & C1 \quad : Y_{Z_i} \geq 0, \forall i \in \{1, 2, ..., m\}, \\ & C2 \quad : X_1 \geq 0, \\ & C2 \quad : Y_{Z_i} \in D_i. \end{aligned}$$

4 | THE PROPOSED FRAMEWORK

This section describes the working of the proposed CNN model Ψ for public safety and crime identification (shown in Figure 1). The framework is divided into three distinct layers: the data, AI, and application. The functionality of each layer is described as follows.

4.1 | Data layer

Numerous UAVs $\{D_1,...,D_i,...,D_j,...,D_m\} \in D$ are involved in the data layer, which can be used for patrolling over the city and capturing suspicious scenes. UAVs transmit real-time images and forward them to the proposed classification framework (CNN model). From the images, it is to determine whether any criminal act performed by the users $\{U_1,...,U_i,...,U_j,...,U_m\} \in U$ and classify them as criminal or non-criminal. This is determined by whether or not the user possesses any weapon $\{K_1,...,K_i,...,K_j...,K_m\} \in K$ in the image. The data are captured in images and is further forwarded to the AI layer, which makes predictions.

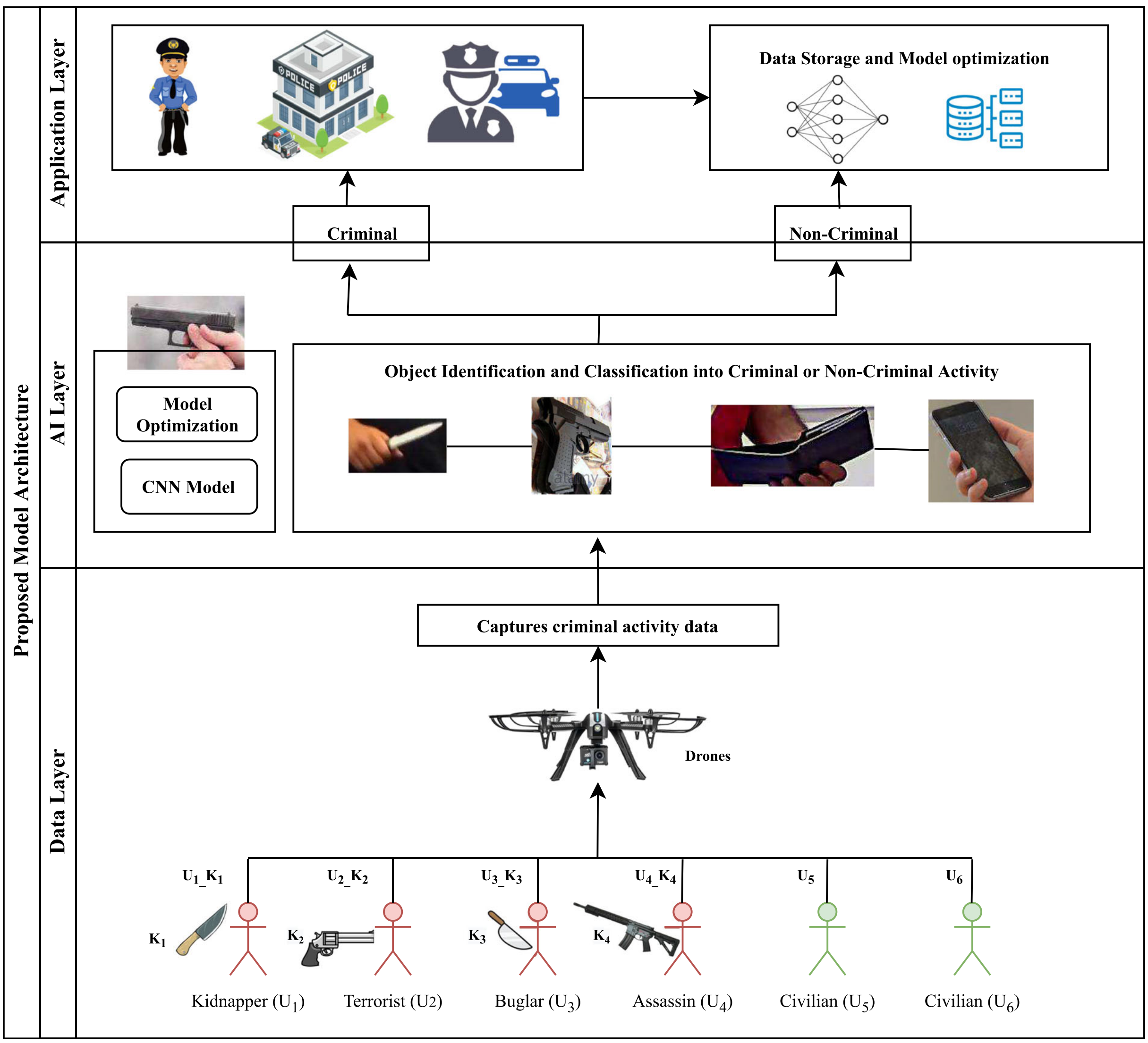

FIGURE 1 Proposed system model.

4.2 | AI layer

This layer's task is to make predictions and classify images based on the data received from the data layer. The AI layer comprises an object detection CNN model, prepared after data preparation and prepossessing, and can make predictions.[25] Figure 2 shows the sequential flow of the proposed framework for UAV-based public safety for human life protection.

4.2.1 | Dataset description

The dataset used to train the proposed weapon identification and classification framework is the small objects handled similarly (SOHAs) weapon detection dataset. It comprises images of weapons and other small objects handled in the same way as weapons. This dataset consists of six object classes: pistol, knife, bill, purse, smartphone, and card. The dataset consists of 8945 images, and the number of images in each class is mentioned in Table 2. In the proposed system, there are numerous sets of images, $\{I_1,...,I_i,...,I_j,...,I_m\} \in I$, which are present in different directories; correspondingly, they are assigned their respective labels according to the directory they are present in such as $\{L_1,...,L_i,...,L_j,...,L_m\} \in L$. The dataset δ with the use of scaled iterator is divided into batches of size 32 such that the respective images and corresponding labels are allotted within a particular batch which can be further used for training the neural network represented as follows.

$$\{I_1 L_1\} \rightarrow B_1, \tag{9}$$

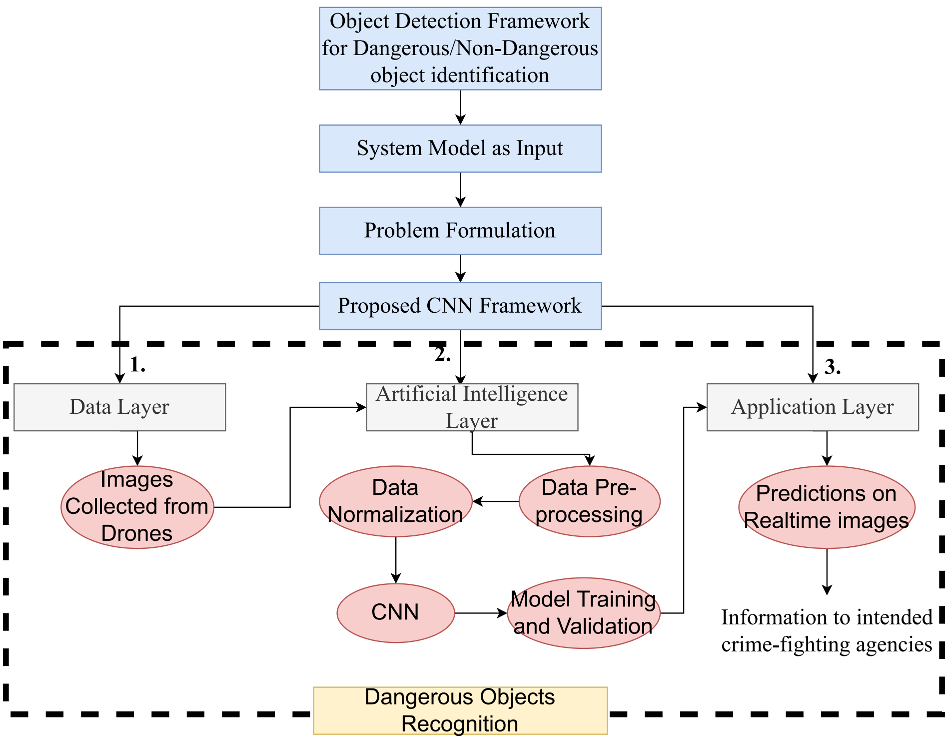

FIGURE 2 Sequential flow of the proposed framework.

TABLE 2 Number of images in each class.

Class Name	Pistol	Knife	Bill	Purse	Phone	Card
Number of images	3710	2170	688	784	1184	409

$$\{I_2 L_2\} \rightarrow B_2, \tag{10}$$

$$\{I_m L_m\} \rightarrow B_m. \tag{11}$$

4.2.2 | Data preprocessing

Firstly, we keep the data δ on the cloud storage and load it further for preprocessing for improved training and prediction values. Initially, the data were scaled to a standard value with all the pixels of an image between 0 and 1 for handling outliers of certain exceptions. The mathematical representation of scaling is described as follows.

$$\delta \rightarrow (x,y) \rightarrow \left(\frac{x}{255}, y\right). \tag{12}$$

Equation (12) represents the mapping and scaling of data, where x is the variable used to denote the tensors comprising values of pixels of images and y is the variable used to denote the label of class an image.

Following tensors, we performed data augmentation using various configurations of image properties like saturation, alignment, brightness, and adjusting contrast. We also elaborated on a particular section of the image that should be focused on for detection. After data augmentation, the images are divided into batches. Each batch comprises 32 images. The *scaled_iterator* is used over the randomized set of images while dividing images into batches to ensure that no image is repeated in consequent batches. A total of 280 batches are formed. These batches are divided into three categories: training, testing, and validation.

$$T_t = \delta \times 0.6, \tag{13}$$

$$T_s = \delta \times 0.1, \tag{14}$$

$$T_v = \delta \times 0.3 + 2. \tag{15}$$

Equations (13)–(15) represent the division of batches, where T_t represents the number of batches for training, T_s represents the number of batches for testing, and T_v represents the number of batches for validation.

4.2.3 | Model motivation

Based on our analysis in Section 2, most pre-trained CNN-based architectures like VGGNet19, ResNet101, and InceptionV3 are used for the proposed model's application. These pre-trained models are trained on the *ImageNet* dataset. We found several drawbacks during the analysis of these models. Figures 3 and 4 show the relative comparison of the proposed framework with the other pre-trained models which are discussed. These models have many layers, as shown in Figure 3, which increases the number of neurons involved in the network. Hence, increasing the trainable parameters, as cited in Figure 4. Due to this, the time and cost of training the model are exponentially higher than that of the proposed model. Considering the financial constraints law enforcement organizations face in developing nations using these pre-trained models is practically unfeasible in the real-world scenario. The pre-trained models also tend to overfit while training. This inspired us to develop the proposed framework, which outperforms the aforementioned models considerably while having only 14 layers and a considerably less number of trainable parameters, hence using way fewer computational resources.

The proposed CNN model is developed after multiple implementations and an extensive tuning of hyperparameters. Given the defined constraints, it gives us more practical solutions for public safety as it gives an enormous boost in performance. Table 3 shows the number of layers the pre-trained models use for prediction and classification.

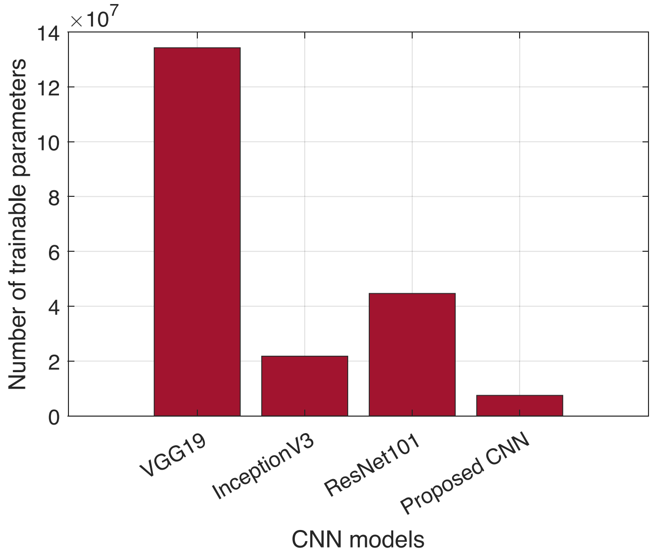

FIGURE 3 Number of trainable parameters for different models.

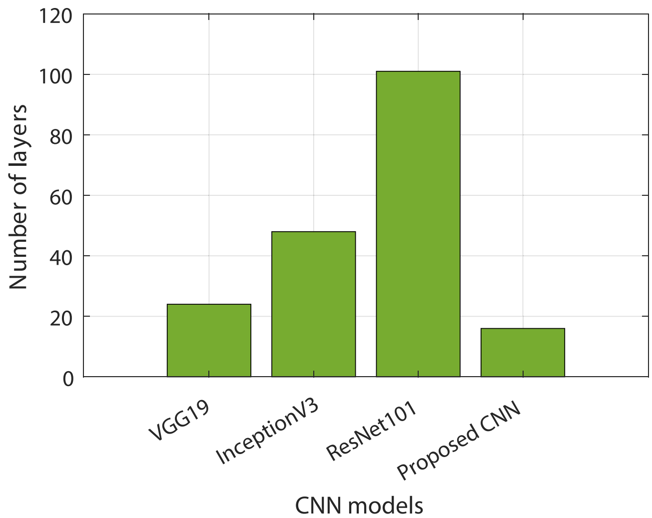

FIGURE 4 Number of layers for different models.

TABLE 3 Comparison of other pre-trained models with our model based on computational space required.

Model name	VGG19	InceptionV3	ResNet101	Proposed model
No. of trained parameters	134,268,738	21,776,548	44,654,504	7,483,910
No. of layers	24	48	101	16

4.2.4 | Model development

The proposed CNN model Ψ is made upon 16-layer architecture and combines various permutations and combinations of different activation functions, initializers, and regularizers with an additional max-pool layer. Figure 5 shows the architecture and layer-wise dissection of the entire proposed framework. The model comprises four convolutional layers, four max-pooling layers, three dense layers, three dropout layers, and an output layer, as shown in the figure. The proposed architecture is the sequence of convolution layers followed by dense neural layers, the input tensor of size

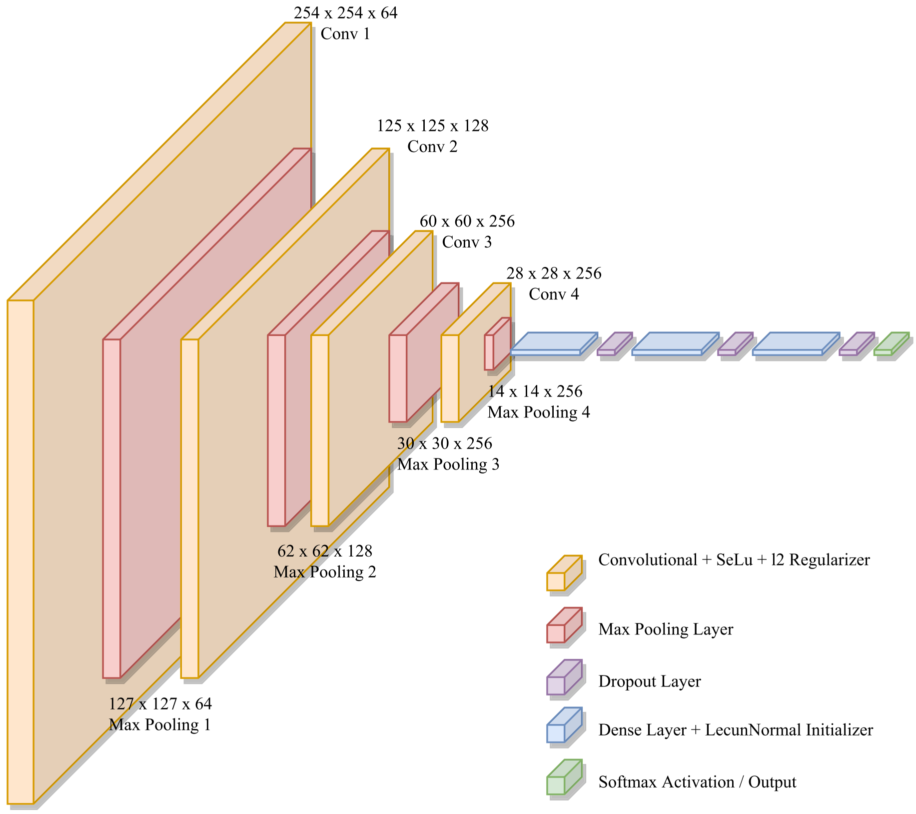

FIGURE 5 Proposed CNN model architecture.

255×255, first transferred through the convolutional block with 32 filters to extract features from the raw input. The convolution layer is paralleled with $L2$ regularizers to reduce the predictive model's variance, thus decreasing the influence of minor and unwanted variables. The subsequent *Max-Pooling* layer of size 2×2 reduces the complexity of extracted features by selecting the maximum value of the identified features and passing the tensor further of output shape $127 \times 127 \times 32$ through the architecture further with the convolutional layers of filter size 128 and 256, respectively. All convolution layers are comprised of *SeLU* activation function, which enables the model to actively consider negative values of the tensors and compute the feature values using the condition of *scaledexponential* that responds to both negative and positive values in an unbiased way. The performed convolution over the images works as follows.

$$\mu = \frac{N_x + 2P - N_h}{d} + 1. \tag{16}$$

Equation (16) represents the generation of output for the next layer from one layer to another. Here, μ represents the size of the output, N_x represents the size of the input, P represents the padding, N_h depicts the size of the filter, and d represents the stride. The regularization loss is performed over the convolution as follows:

$$\Gamma(W) = \frac{\alpha}{2}||W||_2^2 + \Gamma(W) \tag{17}$$

$$= \frac{\alpha}{2}\sum\sum \underset{i_j}{\overset{2}{w}} + \Gamma(W). \tag{18}$$

Equation (16) shows the output after applying the convolution kernel filter on the input image. This reduces the size of the input image and extracts the essential features for the convolutional model to train upon. This equation is essential for reducing the image's size and toning down the image into the most significant features.

This procedure is repeated for subsequent layers. After the series of convolution blocks, the flattened layer is introduced, which converts 3D tensors to 1D tensors with a size of 255. This can be fed further to the three dense layers of the neural network with a dimension of (128,256), (256,256), and (256,6), respectively, with *SeLU* and *Softmax* activation functions. These dense layers help select suitable features for enhancing the model's classification. After successfully passing through the convolution layers, the extracted features are allowed to go through the neural network for distributing the weights of the linear model, which is used for making predictions. Algorithm 1 describes the procedure of updating weights in the convolutional layer of the proposed CNN model.

Algorithm 1 Algorithm for updating weights in the convolutional layer.

```
procedure UPDATING WEIGHTS IN CNN(W, b)
    for i ← 1 to m do
        temp ← 0
        for j ← 1 to n do
            change ← 0
            change ← W[i][j] · X[j] + B[j]
            temp ← temp ± change
        end for
        Y[i] ← temp
    end for
end procedure
```

4.2.5 | CNN predictions

After model development, the model is ready to be deployed for use. The model predicts and categorizes the person in the image as criminal or not criminal. The CNN model classifies the image into one of the 6 classes based on the objects the image contains. This is done using the last dense layer where *Softmaxactivationfunction*, $\sigma(z)$ is used in multinomial logistic regression. It transforms a vector of K real numbers into a probability distribution with K alternative outcomes. As for our case, a list containing six elements would be the output, and the index number of the element with the highest value represents the object prediction for that image. The below equation represents the softmax activation function used in the proposed CNN model.

$$\sigma(z)_i = \frac{e^{z_i}}{\Sigma_{j=1}^{J} e^{z_j}}, \tag{19}$$

where each z_i value is one of the input vector's components and can take any real value. The normalization term at the bottom of the calculation guarantees that all of the function's output values will total 1, creating a proper probability distribution.

4.3 | Application layer

Once UAVs send the captured images back to the model and predictions are made, the application layer is pressed into service. The main functionality of the application layer is to alert the nearest public safety aiding agencies or departments of the criminal activities captured by drones. This enables agencies to act immediately and defend civilians against crimes, protecting public safety.

Another functionality of the application layer is that, regardless of the predictions made, the new images on which predictions are made are added to the data. This provides continuous data updating, enabling high model optimization, as the data would continue to increase, and the model would continuously be optimized with more images. This creates an improved model (the proposed model) and serves better with time and increased data.

5 | RESULTS AND DISCUSSIONS

5.1 | Experimental setup and simulation parameters

The work on the proposed framework is carried out on a cloud-based IDE, Google Collaboratory, on the scripting language Python. Several functionalities and APIs from various libraries are used to develop the framework. Two important functionalities of the TensorFlow library in use are $tf.Keras.regularizers.L2$ and $tf.Keras.initializers.LecunNormal$. $tf.Keras.regularizers.l2$ is used to tone down the overfitting which initially occurred during model building and $tf.Keras.initializers.LecunNormal$ ensures that the model has high accuracy and performance while training weights. Other libraries used are *os*, *numpy*, *cv*2, and *plotly.express*.

CNN is a class of artificial neural networks that have emerged to be useful in various real-world applications. Using a variety of building pieces, including convolution layers, pooling layers, and fully connected layers, CNN is intended to automatically and adaptively learn spatial hierarchies of features through back-propagation. The application based on this paper is allied to image processing and identification. CNN is used for feature extraction from colored 3D images. The subsequent sections highlight the use of our CNN model and its behavior using different performance parameters like accuracy, loss, precision, and performance under different optimizers.

5.2 | Performance analysis

5.2.1 | Loss and validation loss

Figure 6 shows the loss versus validation loss graph while training our model. Here, the *x*-axis shows the number of epochs the model has been trained for, while the *y*-axis represents the loss. As shown in Figure 6, the loss and validation loss curves decrease with the increase in epochs. After extensive training of the framework, the loss and validation loss plot lines converge well after epoch number 10, which indicates less overfitting and a robust model.

The model is trained for 15 epochs; increasing the epochs causes overfitting and results in the divergence of loss and validation loss. Hence, an ideal number of epochs for training is chosen. During the training of the proposed CNN

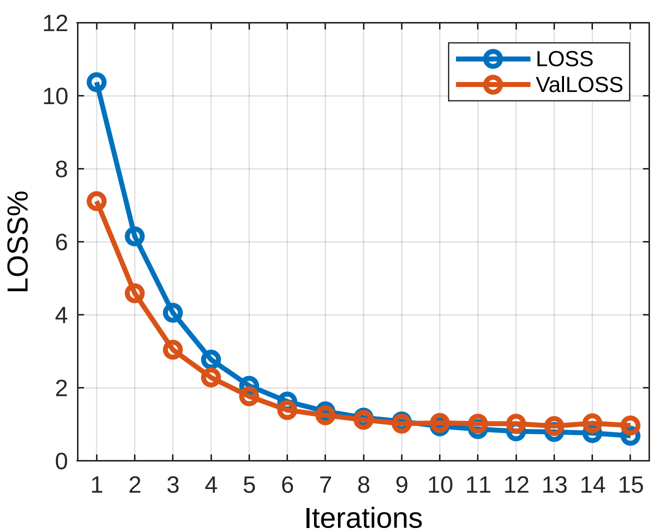

FIGURE 6 Loss versus validation loss curve while training the proposed model.

model, we can successfully tone down the loss substantially. Many methods to avoid overfitting are used during development, such as data augmentation, regularization, batch normalization, and adding dropout layers. Data augmentation is used in the initial stages of prepossessing the images; we resized the images and added many filters to them to optimally highlight the object to be detected. During training, regularization is used to help ignore the undue biases in the data and make an efficient model. We used l2 regularizers that followed ridge regression and added an l2 penalty equal to the square of the magnitude of the coefficient. l1 regularizer was not used as the penalty term; it adds the magnitude of the coefficient, which couldn't minimize the validation loss optimally.

5.2.2 | Accuracy and validation accuracy

Figure 7 shows the accuracy versus validation accuracy graph while training our model. Here, the x-axis shows the number of epochs the model has been trained for, whereas the y-axis represents the measure of accuracy. As shown in Figure 7, the accuracy and validation accuracy are increasing in value with increased epochs. After extensive training of the framework, the proposed model developed a training accuracy of around 84% and validation accuracy of 80%, which indicates less overfitting and a robust model. During the training of the proposed CNN model, various permutations of initializers and activation functions were used to maximize the accuracy such that the selected set of initializers serves their purpose of effectively increasing accuracy in an ideal way considering the chosen activation function, which performs best among all in use. The initializer used is $leCunNormal$ along with $SELU$ activation function to perfectly tone the values of weights and give a better start to the training process. Increasing the number of epochs results in overfitting, where the accuracy and validation accuracy curves diverge; hence, an ideal number of epochs is chosen for the model's training.

Figure 8 shows the comparison of accuracy versus validation accuracy for the different number of epochs training our model. Here, the x-axis shows the number of epochs the model has been trained for, whereas the y-axis represents the measure of accuracy. As shown in Figure 8, the accuracy and the validation accuracy continue to rise and converge with training; this indicates good extensive training of the proposed model. This stops after epoch number 15, after which the accuracy continues to increase, but the validation accuracy stagnates at around 75% while the training accuracy continues to increase up to epoch number 20; this indicates overfitting. Due to less number of data points as inputs for training, the model had to be trained upon a limited number of batches to classify multiple classes; thus, after several epochs, the model starts taking the previous inputs for retraining itself and thus occurs overfitting at a larger number of epochs. Overfitting is highly undesirable in the model as it would lead to false predictions; hence, the authors chose the ideal number of epochs for the model's training to be 15. Weight updating is understood to be the best for 15 epochs.

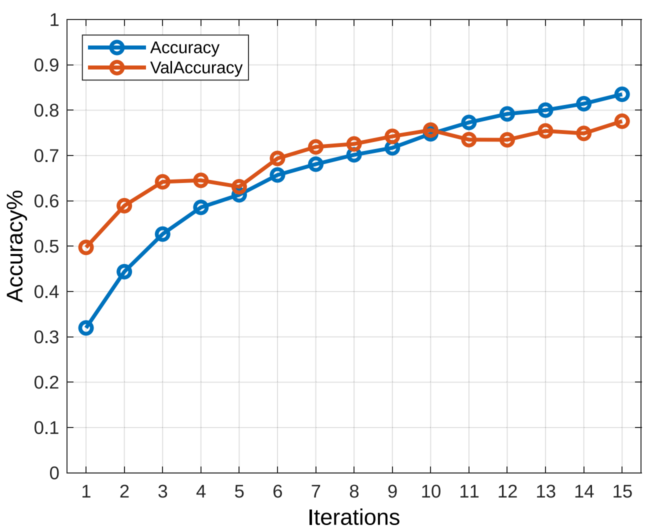

FIGURE 7 Accuracy versus validation accuracy curve while training the proposed model.

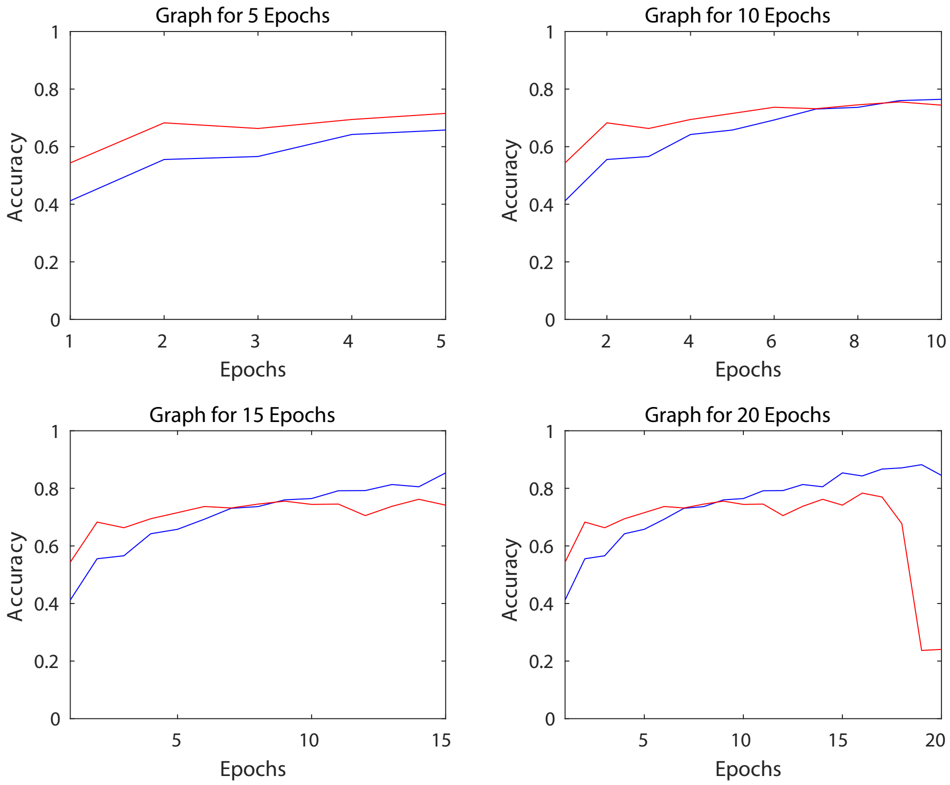

FIGURE 8 Comparison of training and validation accuracy during training of the proposed model over a different number of epochs. The red line indicates validation accuracy, and the blue line indicates training accuracy.

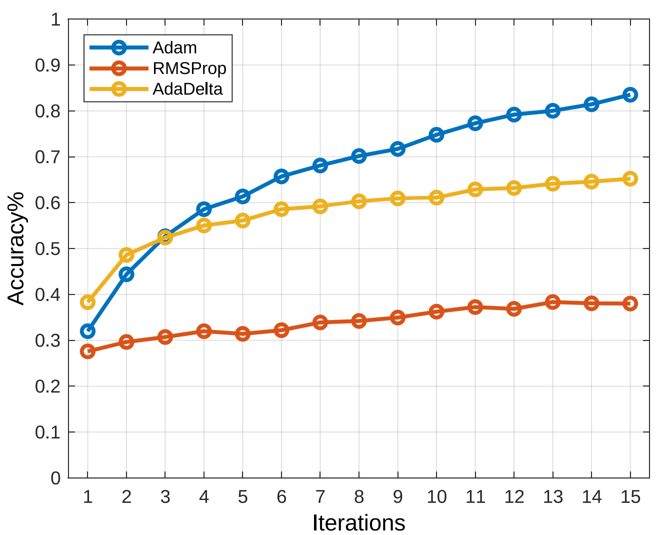

FIGURE 9 Comparison of accuracies using different optimizers while training the proposed model.

5.2.3 | Performance under different optimizers

Figures 9 and 10 show the training accuracy curve under different optimizers while training our model and the training loss curve under different optimizers while training our model. In Figure 9, the x-axis shows the number of epochs the model has been trained for, whereas the y-axis represents the measure of accuracy. In Figure 10, the x-axis shows the number of epochs the model has been trained for, whereas the y-axis represents the loss. The different colored curves

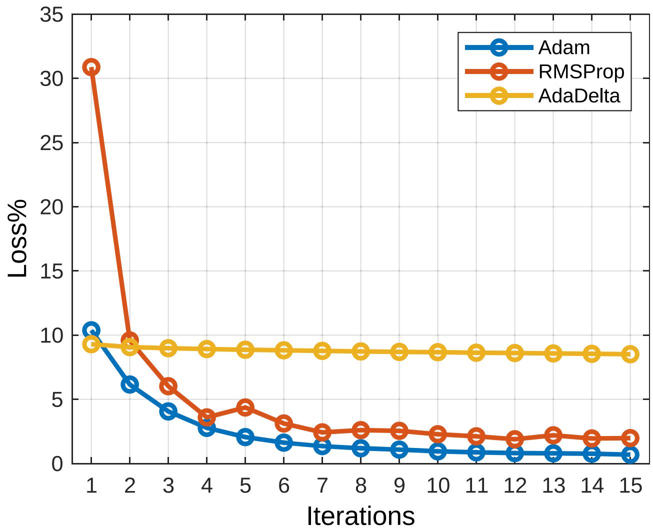

FIGURE 10 Comparison of losses using different optimizers while training the proposed model.

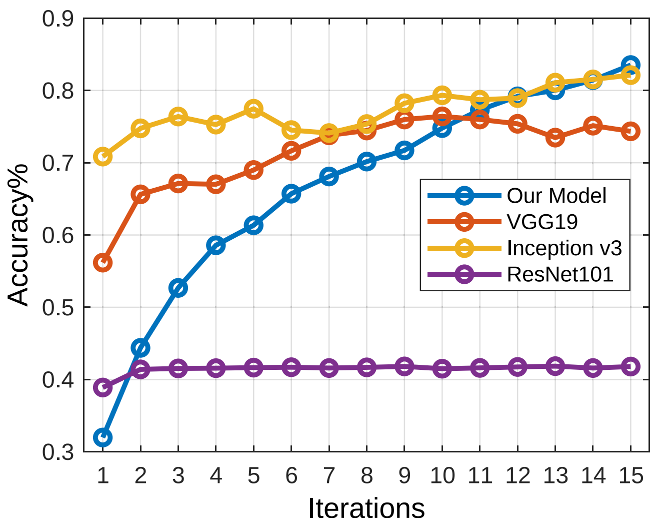

FIGURE 11 Comparison of accuracies of the proposed model with other pre-trained models.

are accuracy curves for different optimizers. The different optimizers in use are Adam, RMSprop, and AdaDelta. As shown in Figures 9 and 10, the optimizer that gives the highest accuracy and minimal loss is Adam.

Adam is an extended version of SGD and can be used in places where SGD is used to update values of parameters optimally. Adam creates an optimization technique that can handle sparse gradients in noisy situations by combining the best features of the AdaGrad and RMSProp algorithms. For example, the second moment with a decay rate to speed up from AdaGrad is used by RMSprop, whereas, in Adam, both the first and second moments are used, making Adam the superior option. AdaDelta and Adam are similar algorithms, except that AdaDelta uses the RMS of parameter updates in the numerator update rule. On the other hand, Adam inculcates bias correction and momentum to RMSprop. Our dataset includes images of low resolution, making Adam perform well than any other optimizer.

5.2.4 | Comparison between different pre-trained models and our model

Figure 11 shows the comparison of accuracy given by our model with other pre-trained models. In Figure 11, the *x*-axis shows the number of epochs the model has been trained for, whereas the *y*-axis represents the accuracy value. Different pre-trained models used for comparison with our model are VGG19, InceptionV3, and ResNet101. Figure 11 shows that the proposed model gives better accuracy than any other pre-trained models.

The other models are trained on the ImageNet dataset, whereas our model is trained on the SOHAs weapon detection dataset. The ImageNet dataset is a very extensive dataset compromising many different objects; conversely, the SOHAs weapon detection dataset emphasizes object categorization into dangerous and non-dangerous categories. Due to this, the other models cannot categorize dangerous and non-dangerous objects as efficiently as the proposed model.

6 | CONCLUSION

In this paper, we have presented a framework for identifying handheld objects and determining whether they threaten public safety. The proposed framework uses multiple UAVs for image collection and a CNN model based on DL to identify crime-related weapons and classify individuals as criminals or non-criminals. Our CNN model has been developed with high prediction accuracy and fewer computational resources than other pre-trained models, achieving an impressive accuracy of 0.8352 and a validation accuracy of 0.7758. Our model outperforms other pre-trained models, such as VGG19, Inception V3, and Resnet101, regarding accuracy, loss, number of trainable parameters, and number of layers in the CNN architecture. Moreover, our proposed framework has been designed to operate with the lowest cost and complexity possible, making it a superior option compared to other pre-trained models.

In the future, we will improvise the performance of the proposed framework by incorporating a blockchain network to confront data manipulation attacks on crime data. We also plan to perform a case study on the real-time performance of the model and improve its efficiency by implementing decentralization.

CONFLICT OF INTEREST STATEMENT

Authors declare that there is no financial or non-financial interests that are directly or indirectly related to the work submitted for publication to this Journal.

DATA AVAILABILITY STATEMENT

No data used to carry out this research.

ORCID

Rajesh Gupta https://orcid.org/0000-0003-3298-4238
Nilesh Kumar Jadav https://orcid.org/0000-0002-0717-8907
Sudeep Tanwar https://orcid.org/0000-0002-1776-4651
Amr Tolba https://orcid.org/0000-0003-3439-6413

REFERENCES

1. Jayakody A, Lokuliyana S, Dasanayaka K, Iddamalgoda A, Ganepola I, Dissanayake A. i-Police—an intelligent policing system through public area surveillance. In: 2021 IEEE 12th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON). IEEE; 2021:148-154.
2. Bang J, Lee Y, Lee Y-T, Park W. AR/VR based smart policing for fast response to crimes in safe city. In: 2019 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct). IEEE; 2019:470-475.
3. Tabassum K, Shaiba H, Shamrani S, Otaibi S. E-Cops: an online crime reporting and management system for Riyadh city. In: 2018 1st International Conference on Computer Applications and Information Security (ICCAIS). IEEE; 2018:1-8.
4. Safat W, Asghar S, Gillani SA. Empirical analysis for crime prediction and forecasting using machine learning and deep learning techniques. *IEEE Access*. 2021;9:70080-70094.
5. Patel D, Sanghvi H, Jadav NK, et al. BlockCrime: blockchain and deep learning-based collaborative intelligence framework to detect malicious activities for public safety. *Mathematics*. 2022;10(17):3195. https://www.mdpi.com/2227-7390/10/17/3195
6. Neto AJV, Zhao Z, Rodrigues JJPC, Camboim HB, Braun T. Fog-based crime-assistance in smart iot transportation system. *IEEE Access*. 2018;6:11101-11111.
7. Sultana T, Wahid KA. IoT-guard: event-driven fog-based video surveillance system for real-time security management. *IEEE Access*. 2019;7:134881-134894.
8. Li M, Lal C, Conti M, Hu D. LEChain: a blockchain-based lawful evidence management scheme for digital forensics. *Future Gen Comput Syst*. 2021;115:406-420. https://www.sciencedirect.com/science/article/pii/S0167739X1933167X

9. Tsai F-C. The application of blockchain of custody in criminal investigation process. *Proc Comput Sci*. 2021;192:2779-2788. https://www.sciencedirect.com/science/article/pii/S1877050921017853. Knowledge-Based and Intelligent Information and Engineering Systems: Proceedings of the 25th International Conference KES2021.
10. Gupta R, Kumari A, Tanwar S. Fusion of blockchain and artificial intelligence for secure drone networking underlying 5G communications. *Trans Emerg Telecommun Technol*. 2021;32(1):e4176. https://onlinelibrary.wiley.com/doi/abs/10.1002/ett.4176
11. Gupta R, Tanwar S, Kumar N. Blockchain and 5G integrated softwarized UAV network management: architecture, solutions, and challenges. *Phys Commun*. 2021;47:101355.
12. Gur A, Erim M, Karakose M. Image processing based approach for crime scene investigation using drone. In: 2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI). IEEE; 2020:1-6.
13. Karim S, Zhang Y, Laghari AA, Asif MR. Image processing based proposed drone for detecting and controlling street crimes. In: 2017 IEEE 17th International Conference on Communication Technology (ICCT). IEEE; 2017:1725-1730.
14. Ralegankar VK, Bagul J, Thakkar B, et al. Quantum cryptography-as-a-service for secure UAV communication: applications, challenges, and case study. *IEEE Access*. 2022;10:1475-1492.
15. He J, Zheng H. Prediction of crime rate in urban neighborhoods based on machine learning. *Eng Appl Artif Intellig*. 2021;106:104460. https://www.sciencedirect.com/science/article/pii/S0952197621003080
16. Hathaliya J, Parekh R, Patel N, et al. Convolutional neural network-based Parkinson disease classification using SPECT imaging data. *Mathematics*. 2022;10(15):2566.
17. Tasnim N, Imam IT, Hashem MMA. A novel multi-module approach to predict crime based on multivariate spatio-temporal data using attention and sequential fusion model. *IEEE Access*. 2022;10:48009-48030.
18. Baek M-S, Park W, Park J, Jang K-H, Lee Y-T. Smart policing technique with crime type and risk score prediction based on machine learning for early awareness of risk situation. *IEEE Access*. 2021;9:131906-131915.
19. Kshatri SS, Singh D, Narain B, Bhatia S, Quasim MT, Sinha GR. An empirical analysis of machine learning algorithms for crime prediction using stacked generalization: an ensemble approach. *IEEE Access*. 2021;9:67488-67500.
20. Han X, Hu X, Wu H, Shen B, Wu J. Risk prediction of theft crimes in urban communities: an integrated model of LSTM and ST-GCN. *IEEE Access*. 2020;8:217222-217230.
21. Wang S, Wang X, Ye P, Yuan Y, Liu S, Wang F-Y. Parallel crime scene analysis based on ACP approach. *IEEE Trans Comput Social Syst*. 2018;5(1):244-255.
22. Zhang X, Liu L, Xiao L, Ji J. Comparison of machine learning algorithms for predicting crime hotspots. *IEEE Access*. 2020;8:181302-181310.
23. Mohammadpour L, Ling TC, Liew CS, Aryanfar A. A survey of CNN-based network intrusion detection. *Appl Sci*. 2022;12(16):8162. https://www.mdpi.com/2076-3417/12/16/8162
24. Anand A, Rani S, Anand D, Aljahdali HM, Kerr D. An efficient CNN-based deep learning model to detect malware attacks (CNN-DMA) in 5G-IoT healthcare applications. *Sensors*. 2021;21(19):6346. https://www.mdpi.com/1424-8220/21/19/6346
25. Mungra D, Agrawal A, Sharma P, Tanwar S, Obaidat MS. PRATIT: a CNN-based emotion recognition system using histogram equalization and data augmentation. *Multimed Tools Appl*. 2020;79:2285-2307.

How to cite this article: Patel N, Vasani N, Gupta R, et al. Convolutional neural network and unmanned aerial vehicle-based public safety framework for human life protection. *Int J Commun Syst*. 2023;e5545. doi:10.1002/dac.5545