An ensemble framework augmenting surveillance cameras for detecting intruder clusters as potential mobs

Omobayo Ayokunle Esan[1], Isaac Olusegun Osunmakinde[2]
[1]School of Computing, College of Science, Engineering and Technology, University of South Africa, Florida, South Africa
[2]Department of Computer Science, College of Science, Engineering and Technology, Norfolk, United States

Article Info

Article history:

Received Nov 6, 2023
Revised May 9, 2024
Accepted Jun 1, 2024

Keywords:

Deep learning
Image processing
Intruder cluster
Machine vision
Surveillance

ABSTRACT

Many developing nations around the world curtail crimes through video surveillance technology, but the crime rate is still high. This is compounded by short-staffed security operatives and a deficiency of security infrastructure to assist security operatives with knowledge-driven decision support systems in the low-resource constraint environment. In a public environment, it is challenging to detect intruder clusters accurately as potential mobs for early warning. Previous research investigated some classical techniques, but their recommendations were insufficient. This research develops a machine learning 3-tiers ensemble framework, which integrates gray level co-occurrence matrices (GLCM) principles to enhance the capabilities of surveillance cameras and security operatives to effectively discern and respond to potential mob formations. The University of California San Diego (UCSD) pedestrian datasets that are publicly available were used for the experiments. With an improved overall average precision of 0.98, recall of 0.98, and accuracy of 98.52% on the UCSD dataset, the suggested framework outperforms the widely used methods for the detection of intruder clusters. The reduction in computational time on processors showcases the framework's significant advancements as a promising solution for robust real-time threat assessment applications.

Corresponding Author:

Omobayo Ayokunle Esan
School of Computing, College of Science, Engineering and Technology, University of South Africa
Florida, South Africa
Email: 58525483@mylife.unisa.ac.za

1. INTRODUCTION

Surveillance systems is a monitoring and detection technology used by governments and organizations to protect homes, businesses, and communities from external threats [1]. These technologies are installed in many private and public areas such as schools, airports, and malls to monitor and track potential intruders that can cause security threats to people’s lives and properties [1]. Intruders can be defined as a potential individual or group of individuals performing activities that do not conform to normal activity in a public environment [2]. Intruder's behaviour has been noted as one of the key causes of anomalies, which have consequently led to crime worldwide. Despite the increasing availability of surveillance, much of it is not used optimally to support real-time security operatives’ detection decision-making in a public environment.

The detection of intruder behaviour in a public environment is a complex problem due to some hidden malicious behaviour patterns that are embedded in the environment [3]. These hidden patterns are due to inconsistencies and complexities in individual behavioural patterns which are often integrated into the ever-changing environment. Unintentional errors made by security staff are the main source of inconsistencies in surveillance data.

The intruder-based security system detects deviation by analyzing the current behavioural patterns with a predefined normal pattern. However, it is affected by the issue of false alarms and delays in time response in real-world implementation [4], which makes the applications not sufficient to support the detection of intruders in crowded environments. In recent years, machine learning has been used in various fields to resolve issues related to a high false alarm and low detection rates [5]. These include artificial neural network (ANN) [6], support vector machine (SVM) [7], random forest (RF) [8], k-nearest neighbour (KNN) [9], decision tree (DT) [10], naïve bayes (NB) [11], convolutional neural network (CNN) [12], rule-based [13], and long short-term memory (LSTM) [14]. Most of these machine learning techniques require the feature extraction of the input image data which is done with various parameter tuning techniques which makes the detection process difficult and time-consuming.

To detect intruder clusters the possible potential criminals and create early warning about the crime before its occurrence, an optimal extraction of the input image feature with gray level co-occurrence matrices (GLCM) applied to the ensemble method is proposed. To accomplish this task, in this research work, the proposed GLCM statistically extracts features from the input image and passes these features to machine learning classifiers for detection purposes. Several machine learning classifiers are used to train the statistically extracted features and the three classifiers with the better result are taken into consideration. Hence, the method is suitable for the detection of possible intruders in a public environment. Most of the prior research focused on using image texture or patched images directly on machine learning which is difficult, inaccurate, and time-consuming. In addition to that, no approaches have dealt with the intricacies of quick detection of potential intruder mob formations hampered by the computing efficiency of real-time surveillance systems.

Drawing upon the provided background information, this study raises the following inquiry: How can a new 3-tier machine learning ensemble framework, integrating modified GLCM for intruder cluster detection be developed and optimize computational efficiency on both CPU and GPU in real-time surveillance systems? The proposed method is developed with the consideration of the above-mentioned gaps and the associated research question, which led to the following contributions,

- New machine learning ensemble framework: The development of a new framework that integrates modified GLCM principles with 3-tier machine learning models. This offers an innovative approach to enhance the performance of detecting intruder clusters as potential mob formations. It thereby enhances the overall security measures of surveillance systems.
- Optimized computational efficiency: A method for reducing computational time on both central processing units (CPU) and graphic processing units (GPU), thereby addressing a critical aspect of real-time surveillance systems, and enabling more responsive threat assessment is introduced.
- Comparative performance evaluation: Detailed experimental evaluations of the proposed framework were conducted on publicly available crime datasets extracted from image analysis, and benchmarked with state-of-the-art detection techniques. This research proves the proposed framework's superiority over the alternatives by demonstrating its improved performance and dependability.

The remaining sections of this paper are organized as follows: section 2 offers an overview of the current detection model as well as the chosen theoretical foundation of the proposed model. An ensemble-based 3-tiers model is explained in detail in section 3, and various experiments and model evaluations are covered in section 4. Section 5 discusses the paper, and section 6 contains the closing remarks.

2. RELATED WORKS

Various studies on the detection of anomalies in surveillance have been published in the literature. Table 1 provides an overview of the current state of crime prediction techniques, including information on the problem being solved, the approach taken, the outcome attained, and any drawbacks. One can see that the existing literature regarding intruder cluster detection within surveillance systems has become apparent from the limitations. Previous research has contributed immensely but often lacked comprehensive and advanced approaches to deal with the intricacies of detecting potential mob formations. Also, a quick assessment of these risks of mob formations is hampered by the computing efficiency of real-time surveillance systems. Through a variety of noteworthy contributions like those listed above, this research greatly strengthens solutions to these weaknesses. It develops an ensemble-based 3-tier model, which integrates principles of feature engineering to reveal early warning of potential public violence information to security operatives.

2.1. Selected theoretical techniques

2.1.1. Artificial neural network

ANN is composed of three layers: an input layer for data storage, an output layer for information computation, and a hidden layer for interconnecting the input and output layers [6]. The weighted sum of an input vector transferred by a transfer function is essentially what makes up a neuron. An ANN is trained via feedforward propagation.

Table 1. Summary of related works on anomalous activities detection in surveillance systems

Citations	Problem Addressed	Method Used	Result Obtained	Limitations
[11]	Anomalous event detection (AED) in urban surveillance on the appearance of objects and their environment	The utilization of CNNs and generative adversarial networks (GANs) was observed.	Their method's outcome demonstrates that, within a given time frame, an anomalous event can be precisely and successfully detected in a crowded scene.	The method's implementation necessitates domain experts' knowledge due to its computational complexity
[12]	An anomaly in surveillance video that involves the temporal localization of anomalous events in unannotated video sequences.	Rule-based dynamic threshold algorithm (DTA).	The experiment results show that accuracy of 0.877, recall of 0.994, precision of 0.824, and score of 0.901.	Due to the rule-based approach used in the experiment, the approach can sometimes be biased.
[13]	The problem of video surveillance anomaly detection was discussed.	Long-short-term memory (LSTM).	The outcome of the experiment demonstrated that the method could successfully reconstruct images and recognize abnormal behaviours.	The approach requires a lot of memory to run the simulation and is much harder to implement
[6]	Detection of anomalous in surveillance images.	using a deep neural network.	According to the experiment's results, the method's accuracy was 97.7%.	The technique is computationally intensive.
[8]	The problem of inaccurate performance in the current anomalous detection system	Median filtering and the KNN technique were used.	The experimental result revealed that the model outperformed others with an accuracy of 85.15%.	Detailed experimental analysis of how features are extracted for k-NN for detection was not shown

2.1.2. Support vector machine

For classification problems, supervised algorithms like SVM can be utilized [7]. To create a linear boundary between the classes and identify a suitable region containing most of the data from an unknown probability distribution (non-linear class problem), SVM is utilized. For every class, the maximum distance should be found between the boundary and the closest data point.

2.1.3. Random forest

An ensemble of DT is used in the RF-supervised learning algorithm to create a forest [8]. To create training sets, RF applies the bootstrap method. Next, divide the nodes and branch features of each training set into a DT using entropy and information gain.

2.1.4. K-nearest neighbour

KNN is a supervised algorithm for machine learning that can be applied to regression issues as well as classifications [9]. Nonetheless, KNN is employed in this study to solve the classification issue. In this step, the minimum distance is typically calculated using the Euclidean distance of lower-dimensional space. The KNN technique has the benefit of being resilient to data samples that have never been seen before.

2.1.5. Decision tree

Typically using a top-down greedy method, a DT classifier offers a quick and efficient way to classify data instances [10]. Training datasets are recursively divided into smaller subsets by DT until every set is part of a single class. Information theory is used iteratively by the DT algorithm as a means of selecting attributes.

2.1.6. Naïve bayes

The NB is a popular classifier method that has been applied to several domains such as mage and patterns recognition, detection, and weather forecast [11]. The NB classification algorithm estimates the class-conditional probability by assuming that the attributes are conditionally independent, given class label C. This implies that the NB classifier allows each feature to contribute towards the classification decision both equally and independently of other features.

2.1.7. Convolutional neural network

An example of an ANN is CNN, which filters inputs using a convolutional layer to extract meaningful data for the network, like edges, shapes, and patterns [12]. Rectified linear unit (ReLU), pooling, convolutional, fully connected, and other types of repeating layers and activation functions are common components of CNNs, as Figure 1 illustrates.

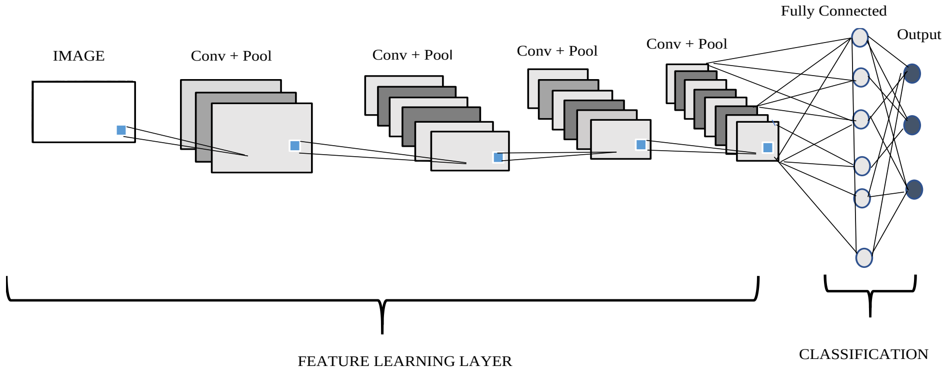


Figure 1. CNN network adopted from [12]

2.1.8. Convolutional neural network-based tensorflow

A CNN family developed on the TensorFlow platform to lower the cost of deep learning is represented by CNN-based TensorFlow [15]. TensorFlow, a CNN-based architecture, consists of two convolutional layers, two pooling layers, one fully connected layer, one output layer, and dropout, which is inserted between the fully connected layer 1 and layer 2 to prevent overfitting of the data.

2.1.9. Gradient boost machine

An effective machine learning method for solving regression and classification issues involving ensemble weak prediction models is called gradient boost machine (GBM) [16]. The GBM approach optimizes an arbitrary differentiable loss function to generalize models that are constructed step-by-step. The next section discussed the methodology used for the implementation of this research.

3. METHOD

The investigation in this research employs an experimental research design. This study makes use of experimental data to back up positive claims about gaps and contributions in section 1. The following sections show the experimental procedures used for the implementation. In this study, MATLAB R2017 was the implementation software used. The University of California, San Diego (UCSD) dataset repository supplies 36 intruder videos and 34 non-intruder videos, from which the image frames used in the implementation are taken [17].

3.1. Experimental procedure

In this study, the image data is acquired from publicly available UCSD data which contains both a mixture of normal activities and suspicious activities. The image is passed through image data processing to remove noise or any unwanted artefacts. The pre-processed image is fed to the feature extraction stage where modified GLCM is used to statistically extract features from the image. The extracted features are passed to the classifiers for training and detection purposes. For the reader to comprehensively understand, the stages used in the experimental procedures for the detection of the intruder clusters as potential mobs are illustrated in Figure 2.

3.1.1. Stage 1: image acquisition

In this experiment, the image used is obtained from publicly available UCSD dataset. This dataset contains intruder clusters (anomalous activities) and non-intruder clusters (normal activities) behavioural patterns [17]. To improve the performance of the proposed method, the acquired image is directed to image pre-processing stage where unnecessary noise and other artefacts are removed.

3.1.2. Stage 2: image pre-processing

Image pre-processing has become a regular operation in image processing for computational efficiency. To perform the pre-processing stages used in this research, different steps are implemented such as image resizing, image annotation, image noise removal, image augmentation, and background subtraction. The steps in the pre-processing stage used for the implementation in this research are discussed in the following:

– Image resizing and annotation of targets

The original image from the camera used in this study is 1920×1080. This image is resized by 512×512 using a bilinear interpolation algorithm to lower the computational complexity of the image data. The image frames were annotated with bounding boxes to indicate the presence of intruder clusters and reduce false detection errors during implementation. During the implementation, the annotated image with the bounding boxes is those that contain intruders as a cluster as identified as a potential mob and is manually labelled '1' in the dataset while the image with non-intruder activities is labelled as '0'.

– Noise removal

The image frames that are noisy (due to factors like direction, smoke, background lighting, and light conditions) as indicated in the second layer from the top of Figure 2, are fed to the noise removal subsystem to improve the quality and consistency of the data. As shown in (1), mean filtering is used in this study to eliminate noise from the image. Where $f(i,j)$, is a noisy image, $g(x,y)$ is the enhanced image, S is a neighborhood of (x,y), and N is the number of pixels in S.

$$g(x,y) = \frac{1}{N}\sum_{(i,j)\in S} f(i,j) \tag{1}$$

– Images augmentation

The filtered image is passed to image augmentation where preprocessing methods like rotation and flipping are applied to address the class imbalance between intruder clusters and non-intruder clusters to improve machine learning models' performances. The augmented image is passed to the background subtraction for further processing.

– Background subtraction

Immediately after image augmentation is image background subtraction where the current image background $I(x,y,t)$ at the time (t) is subtracted from the previous image frame $I(x,y,t-1)$ at a time (t-1) using the frame differencing technique [4], as in (2). The foreground is the region of interest in this research where behavioural activities are taking place. Where Thr is the threshold value which ranges from 0-255 and the output of the foreground image is passed to the feature engineering stage for further processing.

$$Foreground = |I(x,y,t) - I(x,y,t-1)| > Thr \tag{2}$$

3.1.3. Stage 3: feature extraction

To reduce feature redundancy and improve classification results, the modified GLCM [18] which is a statistical feature method of extraction is used in this research; this includes correlation, contrast, variance, mean, entropy, skewness, kurtosis, homogeneity, as shown in Table 2. These features are extracted from the image and fed into an ensemble-based 3-tier model for improved detection. These are more beneficial to this research in terms of improved sensitivity to certain texture patterns, enhanced discrimination between simila texture, and reduced computational efficiency. For better understanding of readers, these modified GLCM are shown in Table 2 with the features, descriptions, and equations.

3.2. Model training, testing, and evaluation metrics

The dataset is divided into training, validation, and testing groups. A cross-validation technique is used to test the model using the validation dataset after it has been trained on the annotated dataset, with 90% of the dataset designated for testing and the remaining 10% for training. This is because training and testing datasets need to be appropriate representations of a potential mob for intruder identification. Overfitting and bias were prevented by repeating this procedure. The best machine-learning techniques were determined by testing the trained model on unknown (unobserved) image frames.

3.3. Building model by ensemble methods with algorithmic and mathematical analysis

After applying each classification (C) algorithm to the extracted data (x) separately, this research computes the result of each of the model classification results from each test instance, and the final output is detected as computed in (3). Where x is the extracted input data and C is the assigned classifier.

$$y = Max\{C_1(x), C_2(x), \ldots., C_n(x)\} \tag{3}$$

Figure 3 displays the pseudo-code used to create an ensemble-based 3-tier model implementation. After converting the picture frames into numerical vectors, each classifier is applied to the data to identify potential mobs by feeding and reading the data as a CSV file. The best classifiers with the best detection capability were ultimately selected to form the suggested 3-tiers ensemble model after the majority vote was applied.

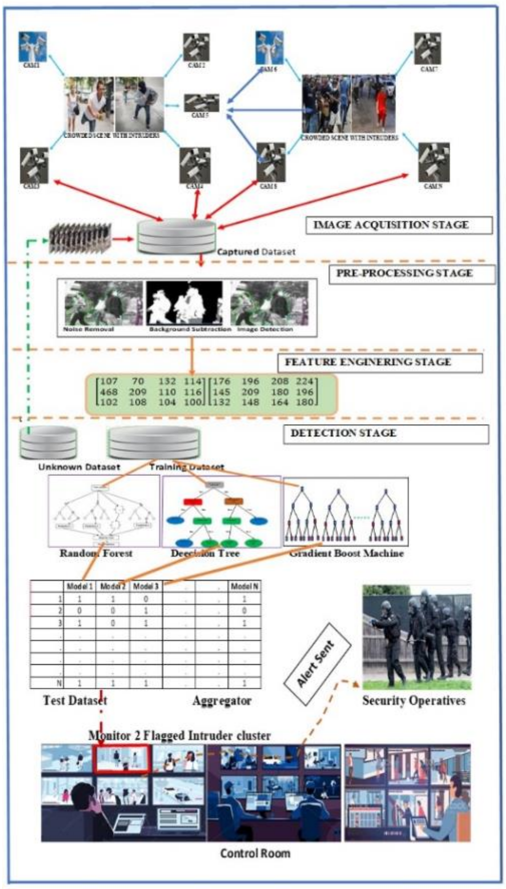
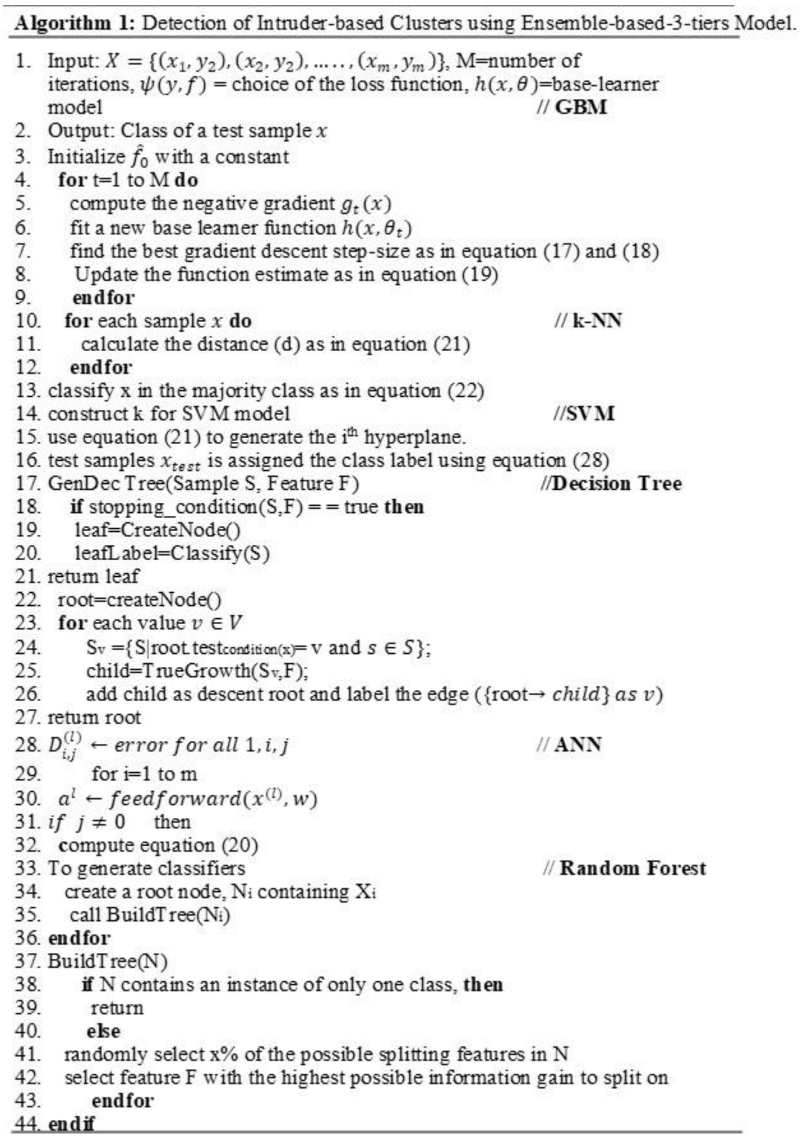


Figure 2. Flow chart for detection of intruder clusters as potential mobs

Table 2. Modified GLCM features and descriptions

	Features	Description	Equation
[i]	Correlation	This indicates the degree of correlation between a pixel and its surrounding pixels in an image.	$correlation = \sum_{i,j=0}^{N-1} p_{i.j} \frac{(1-\mu)}{\sigma^2}$
[ii]	Contrast	This yields an intensity contrast value across an image between a pixel and its neighbour.	$contrast = \sum_{i,j=0}^{N-1} p_{i,j} (1-j)^2$ In an image size MXN, p (I, j) represents the pixel value at point (i, j).
[iii]	Variance	This is a distribution measure around the mean intensity level of neighbouring pixel pairs.	$\sigma^2 = \sum_{i,j}^{N} p_{i,j} (i-\mu_i)^2$
[iv]	Mean	This is calculated to represent the grey distribution of the image and is the average of all the pixels in the image matrix.	$\mu = \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} p(i,j)$ Where p (i, j) is the gray value of an image pixel at a point (i, j), and M and N are the sizes of an image (i, j).
[v]	Entropy	This is used to quantify the degree of pixel randomness in an image and describes the texture of the image.	$E = -\sum_{i=0}^{255} p_i log_2 p_i$ Where p_i represents the likelihood of falling between [0, 255].
[vi]	Skewness	This statistical characteristic describes how asymmetrically distributed the pixels are within the given window to their mean value.	$S_{sk} = \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} [(p(i,j)-\mu)/\sigma]^3$ Where μ and σ represent the mean and standard deviation, and p (i, j) is the image pixel value at a point (i, j).
[vii]	Kurtosis	This quantifies the image's distribution's peak or flatness with a normal distribution.	$k = \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} [p(i,j)-\mu]^4 - 3$ Where μ represents the mean value of the pixel and p (i, j) is the image pixel value at a point (i, j).
[viii]	Homogeneity	This represents the degree to which the elements in the image are distributed closely.	$Homogeneity = \sum_{i,j=0}^{N-1} \frac{P_{i,j}}{1+(i-j)^2}$. Where N is the number of levels, and p (i, j) is the image pixel evaluating a point (i, j).



```
Algorithm 1: Detection of Intruder-based Clusters using Ensemble-based-3-tiers Model.
Input: $X = \{(x_1, y_2), (x_2, y_2), \ldots\ldots, (x_m, y_m)\}$, M=number of
iterations, $\psi(y, f)$ = choice of the loss function, $h(x, \theta)$=base-learner
model                                                    // GBM
Output: Class of a test sample $x$
Initialize $\hat{f}_0$ with a constant
  for t=1 to M do
    compute the negative gradient $g_t(x)$
    fit a new base learner function $h(x, \theta_t)$
    find the best gradient descent step-size as in equation (17) and (18)
     Update the function estimate as in equation (19)
     endfor
  for each sample $x$ do                                 // k-NN
     calculate the distance (d) as in equation (21)
    endfor
classify x in the majority class as in equation (22)
construct k for SVM model                                //SVM
use equation (21) to generate the i^th hyperplane.
test samples $x_{test}$ is assigned the class label using equation (28)
GenDec Tree(Sample S, Feature F)                         //Decision Tree
    if stopping_condition(S,F) = = true then
     leaf=CreateNode()
     leafLabel=Classify(S)
return leaf
 root=createNode()
 for each value $v \in V$
       Sv ={S|root.testcondition(x)= v and $s \in S$};
       child=TrueGrowth(Sv,F);
       add child as descent root and label the edge ({root→ $child\} \; as \; v$)
return root
$D_{i,j}^{(l)} \leftarrow error \; for \; all \; 1, i, j$            // ANN
        for i=1 to m
 $a^l \leftarrow feedforward(x^{(l)}, w)$
$if \;\; j \neq 0$    then
 compute equation (20)
To generate classifiers                                  // Random Forest
  create a root node, Ni containing Xi
   call BuildTree(Ni)
endfor
BuildTree(N)
     if N contains an instance of only one class, then
       return
      else
  randomly select x% of the possible splitting features in N
  select feature F with the highest possible information gain to split on
        endfor
endif
```

Figure 3. Ensemble-based 3-tier model pseudo-code for detection of intruder clusters as potential mobs

3.4. Evaluation metrics

This section presents the evaluation metric used in this research for implementation of the proposed objectives. These metrics include hold-out cross-validation technique, the receiver operating characteristics (ROC) curve, and the confusion matrix as explained in [19]. However, since this research involves videos and images that could demand computational times, the following formulas and terms are defined to assess the machine learning models on both CPU and GPU, as in (4)-(6).

$$\text{Total Computational Time=Preprocessing Time+Training Time+Inference Time} \quad (4)$$

$$\text{Total\_CPU=T\_prep+T\_train\_CPU + T\_inference\_CPU} \quad (5)$$

$$\text{Total\_GPU=T\_prep + T\_train\_GPU + T\_inference\_GPU} \quad (6)$$

Where T_prep is the preprocessing time, T_train_CPU is the training time (CPU), T_train_GPU is the training time (GPU), T_inference_CPU is the inference time (CPU), and T_inference_GPU is the inference time (GPU). The next section discussed the experimental result and discussion of the proposed method with other baseline methods used for intruder detection on image frames.

4. EXPERIMENTAL RESULTS AND DISCUSSION

4.1. Data description and experimental settings

For this study, MATLAB R2017 was the implementation software used. The UCSD dataset repository contains 36 intruder videos and 34 non-intruder videos, from which the image frames used in the implementation are taken [17]. A manual label of "1" is applied to the image containing the intrusion during implementation, while "0" is applied to the image containing non-intrusion activity. Figure 4 displays a snapshot of a few examples of image frames that are accessible to the public and used in this study for implementation.

Figure 4. Snapshot of a different intruder as a potential mob in the publicly available UCSD pedestrian dataset [17]

4.1.1. Experiment 1: detecting intruder clusters as potential mobs using popular CNN approaches

The intention here is to evaluate the proposed model's robustness and reliability in the detection of intruder clusters using the UCSD pedestrian dataset. The image is used for the qualitative experiment. Figures 5(a) to 5(c) noisy image frames, Figures 5(d)to 5(f) enhanced image frames, Figures 5(g) to 5(i) image augmentation output, Figures 5(j) to 5(l) foreground image, and Figures 5(m) to 5(o) bounding box indicating the annotated images show the performance of each of the proposed pre-processing stages, and Figure 6 shows the detection stage using CNN. To evaluate the efficacy of the proposed approach, CNN and CNN-dependent TensorFlow deep learning techniques are utilized for the identification of intruder clusters. Table 3 displays the CNN and CNN-based TensorFlow configuration parameters.

The CNN takes the original image of size 512×512 with 1×1 kernel size and 1 filter produced 512×512×1 output. The output is passed as input to the convolution layer 1, where a convolution operation is performed on the image with the 3×3 kernel size and filters of 24 to obtain 256×256×24 as the output. This output is passed as input to the convolution layer 2 as input, where a convolution operation is performed on the image with the 3×3 kernel size and filters of 48 to obtain 128×128×48 as the output. Then the output is passed to the convolution 3 with the pooling layer of 2×2, with the 3×3 kernel size and filter of 48 to produce 64×64×48 output. The ReLU activation function is applied to increase the non-linear properties of the decision function in the neural network. Thereafter the Softmax function is implemented to classify the pattern in the image as intruder or non-intruder. During the implementation, a cross-validation technique of 90% for the training dataset and 10% for the testing dataset. The confusion matrix for the implementation is shown in Figure 7.

Figures 7(a) and 7(b), the confusion matrix compares the detection values like true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) class of intruders and non-intruders. Figure 7(a) shows the class intruder and non-intruder are correctly detected as 748 and 464 respectively with CNN. Figure 7(b) shows that the class intruder the non-intruder is correctly detected as 653 and 323 by the CNN-based TensorFlow method. Furthermore, we utilized the ROC curves for the model’s performance comparative analysis as shown in Figure 8.

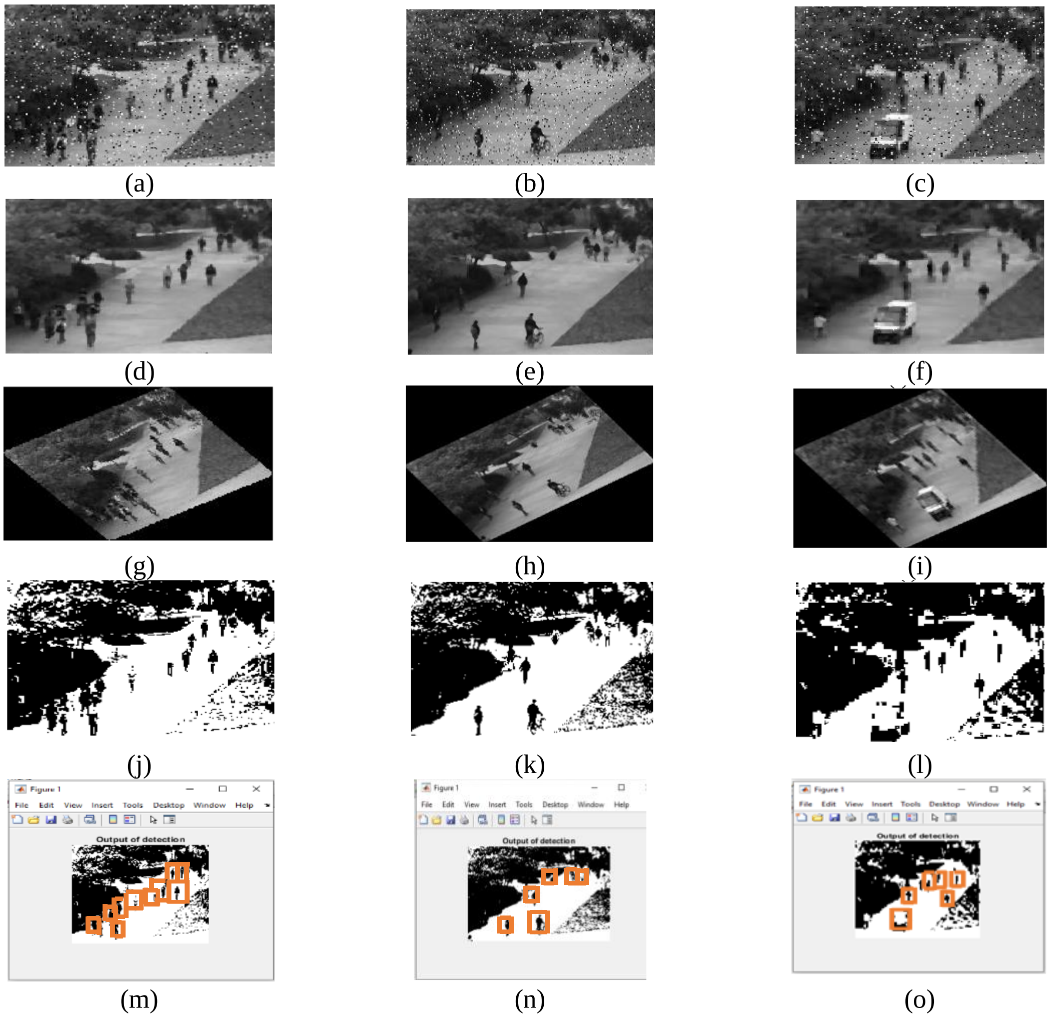
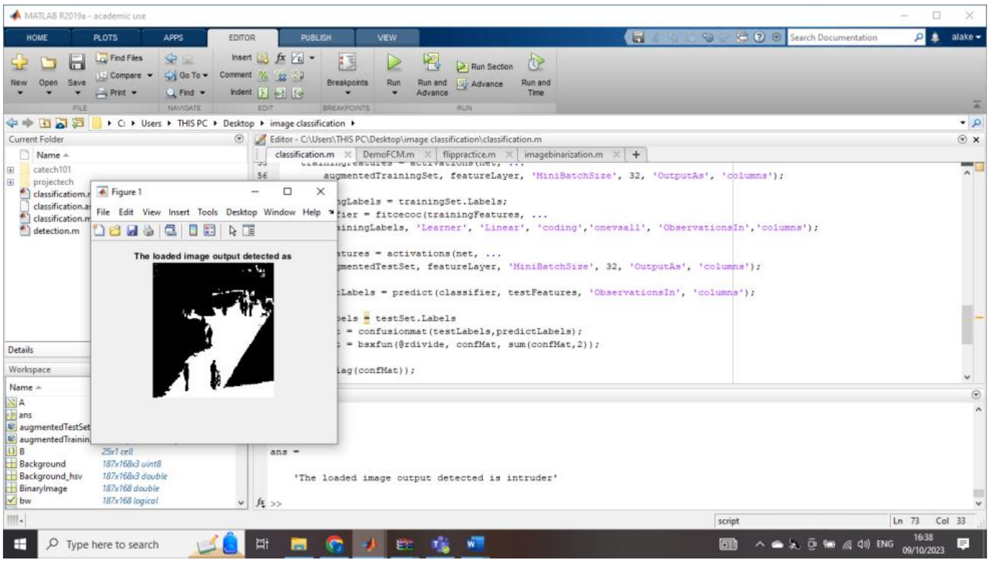


Figure 5. Detection result using the CNN in (a)–(c) noisy image frames, (d)–(f) enhanced image frames, (g)–(i) image augmentation output, (j)–(l) foreground image, and (m)–(o) bounding box indicating the annotated images show the performance of each of the proposed pre-processing stages



```
    augmentedTrainingSet, featureLayer, 'MiniBatchSize', 32, 'OutputAs', 'columns');

gLabels = trainingSet.Labels;
ier = fitcecoc(trainingFeatures, ...
iningLabels, 'Learner', 'Linear', 'coding','onevsall', 'ObservationsIn','columns');

tures = activations(net, ...
mentedTestSet, featureLayer, 'MiniBatchSize', 32, 'OutputAs', 'columns');

Labels = predict(classifier, testFeatures, 'ObservationsIn', 'columns');

els = testSet.Labels
 = confusionmat(testLabels,predictLabels);
 = bsxfun(@rdivide, confMat, sum(confMat,2));

ag(confMat));
```

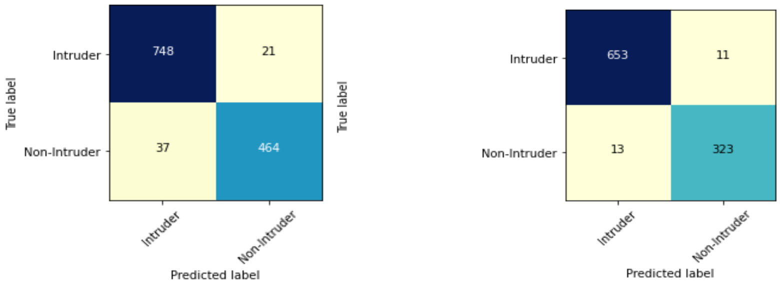
```
ans =

    'The loaded image output detected is intruder'
```

Figure 6. The detection stage using CNN

Table 3. Configuration of CNN architecture

CNN						CNN-based TensorFlow		
Layers	Kernel Size	Filters	Input	Output Shape	Activation Function	Layers	Input	Output Shape
Original image	1×1	1	512×512×1	512×512×1	–	Original image	512×512×1	512×512×1
Convolutional layer 1 + Pooling (2x2)	3×3	24	512×512×1	256×256×24	ReLu activation	Convolutional layer 1 + Pooling (2×2)	512×512×1	256×256×24
Convolutional layer 2 + Pooling (2x2)	3×3	48	256×256×24	128×128×48	ReLu activation	Convolutional layer 2 + Pooling (2×2)	256×256×24	128×128×48
Convolutional Layer 3 + Pooling (2x2)	3×3	48	128×128×48	64×64×48	ReLu activation	Convolutional Layer 3 + Pooling (2×2)	128×128×48	64×64×48
dense 1	-	-	64×64×48	24	ReLu activation	dense 1		24
dense 2	-	-	64×64×48	12	Softmax activation	dense 2		12



(a) (b)

Figure 7. Confusion matrix for detection of intruder clusters (a) CNN and (b) CNN with tensorFlow

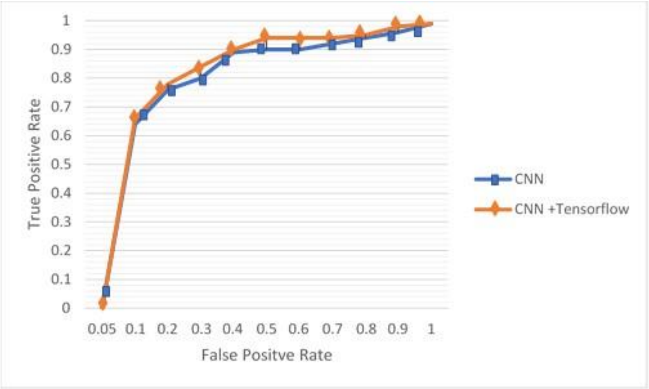


Figure 8. ROC-curve for the detection of intruder clusters

From Figure 8, we can see that the CNN-based TensorFlow provides better performance compared to the CNN model. The summary of CNN and CNN-based TensorFlow models' performances in terms of recall, precision, F1-score, and accuracy are shown in Table 4. From Table 4, one can observe that the CNN-based TensorFlow model provided satisfactory detection performance. However, computing pixel value directly on the voluminous surveillance image dataset is susceptible to errors and unreliable in real-life practices due to much computational time required for the data to be processed and learned by the model which consequently leads to delays in sending quick messages to the security operatives in the control room on the monitors with intruders as potential mobs' activities for them to take appropriate actions.

Table 4. Summary of performance metrics on CNN techniques

Models	Recall	Precision	F1-Score	Accuracy (%)
CNNs	0.93	0.92	0.92	92.54
CNN-based TensorFlow	0.93	0.93	0.94	93.86

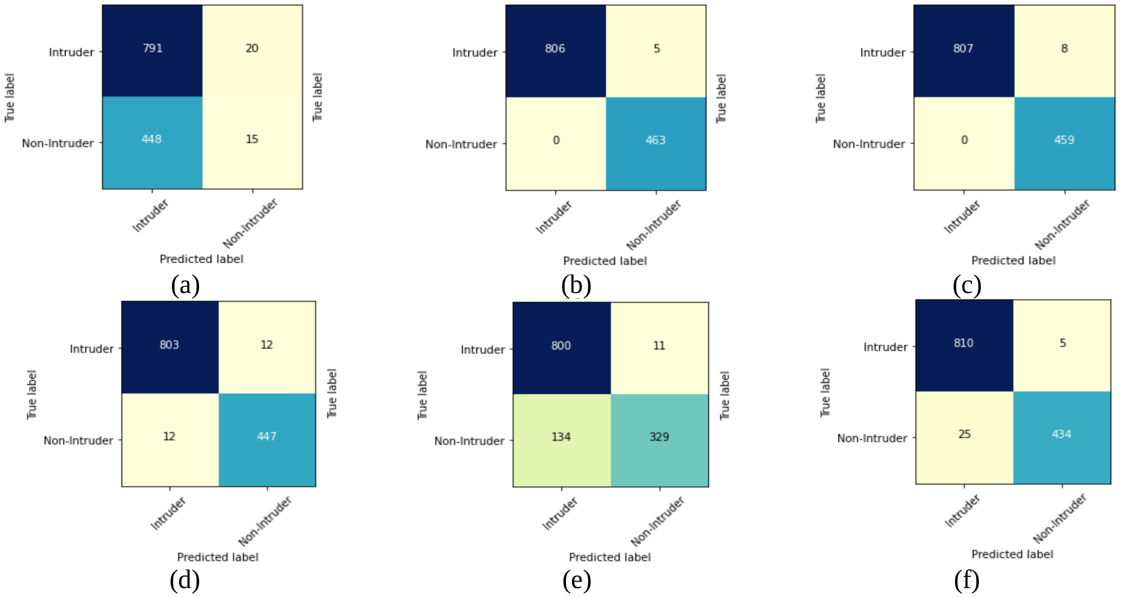
4.1.2. Experiment 2: detecting intruder clusters as potential mobs using proposed ensemble framework on publicly available UCSD dataset

The purpose of this experiment is to investigate the robustness of the proposed ensemble framework using an image extracted from the modified GLCM on the publicly available UCSD pedestrian dataset. The features were extracted from the image using the feature engineering process explained. Quantitative experiments are conducted on the extracted data with all the models selected in this research using a cross-validation technique like Experiment 1. The detected result for the intruder cluster is shown in the right column of Table 5. Table 5 shows the feature extracted from the image frames and the predicted results from NB, KNN, SVM, DT, RF, and GBM. We can observe the differences between the actual detection result and prediction results of the six models used for the detection of intruder clusters on image pixel values, from this result the DT, RF, and GBM show better prediction results. The performance of all selected models is further compared using the confusion matrix as shown in Figure 9.

Table 5. Detection of intruder clusters with modified GLCM features extraction

Image	Homgeneity	Entropy	Skewness	Kurtosis	Mean	Cont-rast	Variance	Correlation	Actual	Predicted					
										NB	SVM	k-NN	DT	RF	GBM
152	0.0688	0.0398	0.8937	0.109	0.918	0.0332	12.481	0.8798	0	0	0	0	0	0	1
279	0.0608	0.0309	0.9176	0.104	0.939	0.0474	13.211	0.7404	0	1	0	0	0	0	0
332	0.0139	0.1254	0.9527	0.215	0.977	0.0187	8.4148	0.0853	1	0	0	1	1	1	1
523	0.0749	0.0178	0.9246	0.137	0.952	0.0554	6.6776	0.8961	1	0	0	0	1	1	1
702	0.0749	0.0367	0.8933	0.098	0.917	0.0274	15.299	0.4297	1	1	1	1	1	1	1

Figure 9(a) presents the confusion matrix of the detection result obtained from NB, Figure 9(b) is confusion matrix of detection of intruder clusters using SVM, Figure 9(c) presents the confusion matrix of detection of intruder clusters using KNN, Figure 9(d) is confusion matrix of detection of intruder cluste with DT, Figure 9(e) is confusion matrix of detection of intruder clusters using RF, and Figure 9(f) is confusion matrix of detection of intruder clusters using GBM. The ROC curves which show the graph of true positive rate (TPR) against false positive rate (FPR) with varied thresholds are further used for selected model performance comparison as shown in Figure 10. From this graph, one can see the performances of each model in the detection of intruder clusters. Other performance metrics used are summarized in Table 6.



(a) (b) (c)

(d) (e) (f)

Figure 9. Confusion matrices capturing the performances of the implemented models: (a) NB, (b) SVM, (c) KNN, (d) DT, (e) RF, and (f) GBM

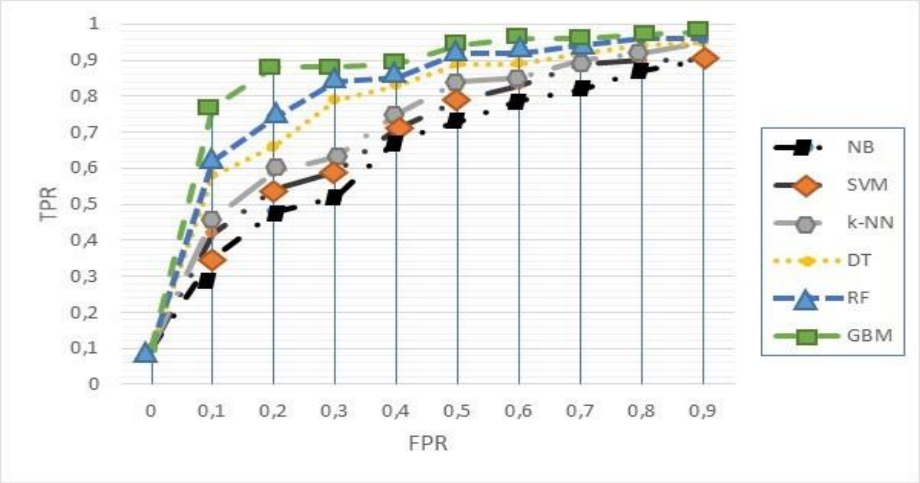


Figure 10. ROC-curve of six classifiers for the detection of intruder clusters with feature engineering

The best three classifiers are chosen using a voting method in (3), as indicated in Table 6 because one of the goals of this study is to identify the best three classifiers with strong predictive results to form an ensemble model as shown in Table 7. According to Table 6, the proposed model has an accuracy of 98.52%, an F1-score value of 0.98, a recall of 0.98, and an overall average precision of 0.98. From these findings, the proposed ensemble-based 3-tiers model performs better than Experiment 1, and this is because statistical feature extraction from the images reveals information that is otherwise hidden.

Table 6. Performance metrics

Model	Performance			
	Precision	Recall	F1-Score	Accuracy (%)
NB	0.92	0.92	0.91	92.17
SVM	0.94	0.93	0.91	93.39
k-NN	0.97	0.96	0.97	97.62
DT	0.98	0.97	0.97	98.35
RF	0.97	0.98	0.97	98.37
GBM	0.98	0.98	0.98	98.83
Proposed ensemble-based 3-tiers (DT +RF + GBM)	0.98	0.98	0.98	98.52

Table 7. Detection test results via 3-tier ensemble model

Image	Homogeneity	Entropy	Skewness	Kurtosis	Mean	Contrast	Variance	Correlation	Actual	Predicted			
										DT	RF	GBM	Av (DT+RF+GBM)
152	0.0688	0.0398	0.8937	0.1098	0.918	0.0332	12.4814	0.8798	0	0	0	1	0
279	0.0608	0.0309	0.9176	0.1048	0.939	0.0474	13.2111	0.7404	0	0	0	0	0
332	0.0139	0.1254	0.9527	0.2105	0.977	0.0187	8.4148	0.0853	1	1	1	1	1
523	0.0749	0.0178	0.9246	0.1337	0.952	0.0554	6.6776	0.8961	1	1	1	1	1
702	0.0749	0.0367	0.8933	0.0968	0.917	0.0274	15.2991	0.4297	1	1	1	1	1

4.2. Comparison evaluation of the proposed method and convolutional neural network methods with computational time

To determine the CPU and GPU computational time with the publicly available UCSD data on the intruder detection framework, experiments were conducted using publicly available image frames like Experiment 1 and 2, the process was repeatedly done (3 runs) and the total average computational time for both the CPU and GPU systems for the proposed models is taken as visualized as in Figure 11. By contrast with

CPU and GPU, when looking at Figure 11 we can observe that the processing time of the CNN models on CPU increases time executions. The results show that using the modified GLCM with an ensemble-based 3-tier model learning process is faster than CNN models used, and this is also applicable to GPU thereby addressing a critical aspect of real-time surveillance systems and enabling more responsive threat assessment is introduced. Although the processing time is high for real-time surveillance intruder detection systems, the processing time is better compared to those detection models used as a baseline in this study.

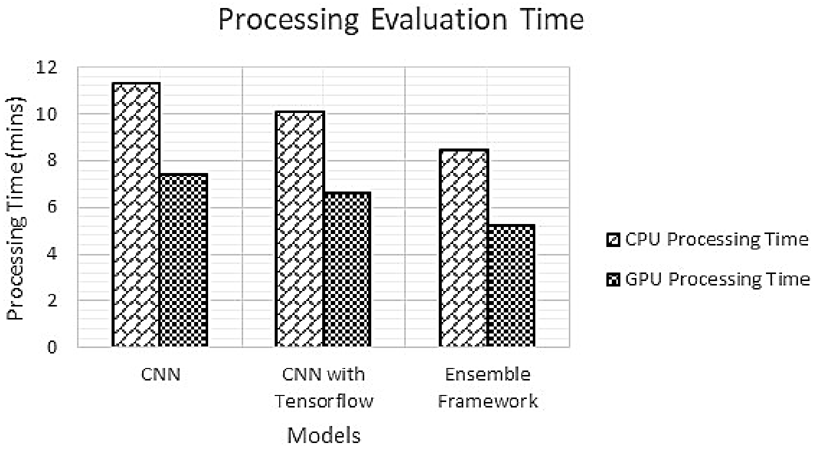


Figure 11. Comparison of model computational time on CPU and GPU on UCSD pedestrian datasets

4.3. Comparison evaluation of performance of proposed method with other methods

This section compares the performance of the proposed model with other state-of-the-art detection models that are currently in use on UCSD ped datasets regarding the following: precision, recall, F1-score, accuracy, anomalies detected, and features extraction model used, as shown in Table 8. Table 8 indicates that, when compared to other state-of-the-art models used in this study, the proposed ensemble-based 3-tiers model performs significantly better on the UCSD ped1 dataset, with an accuracy of 98.52%. The model is appropriate for real-time applications because of its accuracy.

Table 8. Performance comparison among proposed ensemble-based 3-tier method with existing methods

Ref.	Detected Anomalies	Feature Extraction Method	Model	Precision	Recall	F1-score	Accuracy (%)
[20]	Spatio-temporal abnormal behaviour detection in massive crowds	CNN	ResNet-50 CNN and RF	64.72	89.31	75.05	75.72
[21]	Anomalous in crowded scenes	Optical flow	2D CNN	0.81	0.82	0.81	81
[22]	Regular activities as well as clusters in video	Handcrafted Spatio-temporal	Convolutional Autoencoder (Con-AE)	0.864	0.95	-	-
[23]	Bicyclist and cars moving on pedestrian paths	CNN	CNN-LSTM	0.947	0.943	-	95.47
[24]	Violent activities in surveillance systems	CNN-LSTM	MobileNet v2 classifier	0.96	0.96	0.96	96
[25]	Detection of Anomaly in Video (VAD)	Generative Adversarial Network (GAN)	Hybrid model {3D-CNN, GAN, and AE)	0.86	0.944	0.902	91
Proposed method	Intruder clusters as potential mobs	Modified GLCM	Ensemble-based 3-tiers model	0.98	0.98	0.98	98.52

5. DISCUSSION

This study investigated the detection of intruders in public environments using modified GLCM to statistically extract features from the image frames and trained the extracted features with an ensemble-based 3-tiers method while earlier studies have explored different methods such as textural, patch, shape, and edge-based features extraction on image frames and train with the machine learning to detect intruder activities, they have not explicitly addressed it the intricacies of detecting potential mob formations in public environment hampered by the computing efficiency of real-time surveillance systems. From the experiment conducted, we found that the modified GLCM approach was able to statistically extract features from the UMN image data used in the implementation. The GLCM-based features extracted method showed the statistical variations of hidden information embedded in each image frame and the extracted features are trained with different classifiers as shown in Table 6. The best three classifiers that give optimal detection results are the DT, RF, and GBM which are then used as the proposed ensemble method in this research. Furthermore, the processing time of the proposed method was 5 minutes on GPU and 8 minutes on CPU which correlates with the research in [26] on evaluation of the processing time. The proposed method used in this study tended to have an inordinately higher proportion of true detection accuracy of 98.5% with a lower false alarm of 0.015.

In comparison of the proposed method with other suspicious detection methods used in literature as shown in Table 8, the study suggests that higher accuracy is not associated with poor-quality image frames. The proposed method may benefit from the image noise removal method and statistical feature extraction without adversely impacting the performance accuracy of the proposed method in the detection of intruders in crowded environments. Limitations, this study explored comprehensive and advanced approaches to deal with the intricacies of detecting potential mob formations in addition to a quick assessment of these risks of mob formations hampered by the computing efficiency of real-time surveillance systems in public environments. However, further, and in-depth studies may be needed to confirm its suspicious detection performance on voluminous image frames using two or more feature extraction methods to improve the detection model processing time.

6. CONCLUSION

In the study, we have described the theories and illustrated the application of modified GLCM and ensemble-based 3-tier technology for the detection of intruder clusters as possible mobs on publicly available surveillance detection image datasets obtained from crowded environments. This contributes to an effective autonomous surveillance system and handles the issue of hidden information embedded in surveillance images in public environments. The results of the six classifiers on the publicly available dataset show that the new ensemble-based 3-tiers method gives improved performances when compared with conventional methods. With the improved performance obtained, the observations suggest that the proposed method could simply be used by security operatives in public environments to detect the possibility of intruder clusters as potential mobs in surveillance systems before it leads to crime. The findings provide conclusive evidence that this phenomenon is associated with the revealing of unobserved behavioural patterns in the image frames due to statistical feature extraction and the ensemble-based 3-tiers method used for intruder detection in the implementation. Interestingly, integrating this proposed model into surveillance security modules (e.g., security alerts and planning) would result in diverse real-life problems (such as school criminal accidents) being solved with intelligent surveillance security systems. Implications for future research, this study demonstrates that utilizing the proposed method is more resilient than the other state-of-the-art methods used in this study. Future studies may explore the use of a combination of texture and statistical-based feature extraction methods with feasible ways of producing robust suspicious surveillance detection in image frames.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the financial support made available by the University of South Africa and resources made available by Norfolk State University, USA.

REFERENCES

[1] E. Varghese, J. Mulerikkal, and A. Mathew, “Video anomaly detection in confined areas,” *Procedia Computer Science*, vol. 115, pp. 448-459, 2017, doi: 10.1016/j.procs.2017.09.104.

[2] O. A. Esan and I. O. Osunmakinde, “A computer vision model for detecting suspicious behaviour from multiple cameras in crime hotspots using convolutional neural networks,” *International Conference on Practical Applications of Agents and Multi-Agent Systems*, vol. 1678, pp. 197-209, 2022, doi: 10.1007/978-3-031-18697-4_16.

[3] V. A. Kotkar and V. Sucharita, “A comparative analysis of machine learning based anomaly detection techniques in video surveillance,” *Journal of Engineering and Applied Sciences*, vol. 12, no. 12, pp. 9376-9381, 2017.

[4] U. R. M. Castro, M. W. Rodrigues, and W. C. Brandao, “Predicting crime by exploring supervised learning on heterogenous data,” *In Proceeding of the 22nd International Conference on Enterprise Information Systems (ICEIS2020)*, vol. 1, pp. 524-531, 2020.

[5] M. I. Sarker, C. L. -Gutiérrez, M. M. -Romera, D. F. -Jiménez, and S. L. -Sánchez, “Semi-supervised anomaly detection in video-surveillance scenes in the wild,” *Sensors*, vol. 21, no. 12, 2021, doi: 10.3390/s21123993.

[6] V. Singh, S. Singh, and D. P. Gupta, “Real-time anomaly recognition through CCTV using neural networks,” *International Conference on Smart Sustainable Intelligent Computing and Applications*, vol. 173, pp. 254-264, 2020.

[7] B. Prabha, N.R Shanker, M. Priya and E. Ganesh, “Human anomalous activity detection: shape and motion approach in crowded scenes,” *Journal of Physcis*, vol. 3, pp. 1-9, 2021, doi: 10.1088/1742-6596/1921/1/012074.

[8] L. Zhu, X. Zhou, and C. Zhang, “Rapid identification of high-quality marine shali gas reservoirs based on the oversampling method and random forest,” *Artificial Intelligence in Geosciences*, vol. 2, pp. 76-81, 2021.

[9] D. Esan, P. A. Owolawi, and C. Tu, “Anomalous detection in noisy image frames using cooperative median filtering and KNN,” *IAENG International Journal of Computer Science*, vol. 49, no. 1, 2022.

[10] D. M. Fand, L. Zhang, C. M. Rahman, M. A. Hossain, and R. Strachan, “Hybrid decision tree and naive bayes classifiers for multi-class classification tasks,” *Expert Systems with Applications* vol. 41, pp. 1937-1946, 2014.

[11] A. Ahammed, B. Harangi, and A. Hajdu, “Hybrid adaboost and naïve bayes classifier for supervised learning,” *Conference on Information Technology and Data Science*, vol. 1, no. 2874, pp. 1-18, 2020.

[12] R. Chauhan, K. K. Ghanshala, and R. C. Joshi, “Convolutional neural network (CNN) for image detection and recognition,” *2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC)*, vol. 2, pp. 1-7, 2018.

[13] K. Dohun, K. Heegwang, M. Yeongheon, and P. Joonki, “Real-time surveillance system for analyzing abnormal behavior of pedestriansopen access,” *Applied Sciences-Basel*, vol. 11, no. 13, 2021, doi: 10.3390/app11136153.

[14] J. T. Zhou, J. Du, H. Zhu, X. Peng, Y. Liu, and R. S. M. Goh, “AnomalyNet: an anomaly detection network for video surveillance,” *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 10, pp. 2537 - 2550, 2019.

[15] L. Yu, B. Li, and B. Jiao, “Research and implementation of CNN-based on tensorflow,” *IOP Conference Series: Materials Science and Engineering*, vol. 490, no. 4, 2019, doi: 10.1088/1757-899X/490/4/042022.

[16] K. Dohun, K. Heegwang, M. Yeongheon, and P. Joonki, “Real-time surveillance system for analyzing abnormal behavior of pedestriansopen access,” *Applied Sciences*, vol. 11, no. 13, 2021, doi: 10.3390/app11136153.

[17] A. B. Chan and N. Vasconcelos, “Modeling, clustering, and segmenting video with mixtures of dynamic textures,” *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 30, no. 5, pp. 909-926, 2008.

[18] Y. Hu and Y. Zheng, “A GLCM embedded CNN strategy for computer-aided diagnosis in intracerebral hemorrhage,” *arXiv-Computer Science*, pp. 1-9, 2017, doi: 10.48550/arXiv.1906.02040.

[19] O. A. Esan and I. O. Osumakinde, “Application of machine learning in predicting crime links on specialized features,” *International Conference on Computer and Communication Engineering*, pp. 143-157, 2023, doi: 10.1007/978-3-031-35299-7_12.

[20] T. Alafif *et al.*, “Hybrid classifiers for spatio-temporal abnormal behavior detection, tracking, and recognition in massive hajj crowds,” *Electronics*, vol. 12, no. 5, pp. 1-19, 2023, doi: 10.3390/electronics12051165.

[21] K. K. Aastveit, “Deep learning for crowd anomaly detection,” *Master Thesis*, Department of Engineering and Sciences, University of Agder, Kristiansand, Norway, 2022.

[22] M. Hasan, J. Choi, J. Neumann, A. K. R. -Chowdhury, and L. S. Davis, “Learning temporal regularity in video sequences,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 733-742, 2016, doi: 10.1109/CVPR.2016.86.

[23] O. A. Esan, D.O. Esan, M. Mbodila, F.A. Elegbeleye, and K. Koranteng, “A surveillance detection of anomalous activities with optimized deep learning technique in crowded scenes,” *Bulletin of Electrical Engineering and Informatics*, vol. 12, no. 3, pp. 1674-1683, 2023, doi: 10.11591/eei.v12i3.4471.

[24] S. Leela, K.V. S. Likhita, D. Kumar, A. Abhiram, and V. Keerthika, “suspicious human activity recognition and alarming system,” *International Journal of Research in Applied Science & Engineering Technology (IJRASET)*, vol. 10, no. 7, pp. 1-15, 2020.

[25] W. Shin, S.-J. Bu, and S.-B. Cho, “3D-convolutional neural network with generative adversarial network and autoencoder for robust anomaly detection in video surveillance,” *International Journal of Neural Systems*, vol. 30, no. 6, pp. 3000-3008, 202.

[26] B. Ya-Meng, W. Yang, and W. She-Shen, “Detection of abnormal behaviour in video images based on hybrid approach,” *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 13, no. 11, 2022.

BIOGRAPHIES OF AUTHORS

Omobayo Ayokunle Esan is a Ph.D. student in the Department of Computer Science at the University of South Africa (UNISA). His research interests include image processing, machine learning, computer vision, cybersecurity, and the internet of things (IoT). He can be contacted at email: 58525483@mylife.unisa.ac.za.

Isaac Olusegub Osunmakinde received his Ph.D. Degree in Computer Science from the University of Cape Town, South Africa. He is an Associate Professor of Computer Science at Norfolk State University (NSU) in Virgnia, USA. He has a track record of accomplishment in supervising students and authoring prestigious articles in accredited refereed journal, book chapters, international conferences, and practical research interest in emerging areas of computational intelligence and deep learning applications, data science, cyber-intelligence, and IoT smart systems. He can be contacted at email: ioosunmakinde@nsu.edu.