IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.Doi Number

Suspicious Activity Recognition Using Proposed Deep L4-Branched-ActionNet with Entropy Coded Ant Colony System Optimization

Tanzila Saba[1], Amjad Rehman[1], Rabia Latif[1], Suliman Mohamed Fati[1], Mudassar Raza[2,] and Muhammad Sharif[2]

[1]Artificial Intelligence & Data Analytics Lab (AIDA), CCIS Prince Sultan University, Riyadh,11586 Saudi Arabia.
[2]Department of Computer Science, COMSATS University Islamabad, Wah Campus, 47040 Wah Cantt. Pakistan

Corresponding author: Muhammad Sharif (e-mail: muhammadsharifmalik@yahoo.com)

ABSTRACT Intelligent visual surveillance systems are attracting much attention from research and industry. The invention of smart surveillance cameras with greater processing power has now been the leading stakeholder, making it conceivable to design intelligent visual surveillance systems. It is possible to assure the safety of people in both homes and public places. This work aims to distinguish the suspicious activities for surveillance environments. For this, a 63 layers deep CNN model is suggested and named "L4-Branched-ActionNet". The suggested CNN structure is centered on the alteration of AlexNet with added four blanched sub-structures. The developed framework is first transformed into a pre-trained framework by conducting its training on an object detection dataset called CIFAR-100 with the SoftMax function. The dataset for suspicious activity recognition is then forwarded to this pretrained model for feature acquisition. The acquired deep features are subjected to feature subset optimization. These extracted features are first coded by applying entropy and then an ant colony system (ACS) is utilized on the entropy-based coded features for optimization. The configured features are then fed into numerous SVM and KNN based classification models. The cubic SVM has the highest efficiency scores, with a performance of 0.9924 in terms of accuracy. The proposed model is also validated on the Weizmann action dataset and attained an accuracy of 0.9796. The successful findings indicate the suggested work's soundness.

INDEX TERMS Threats, Classification, Machine Learning, Surveillance, Computer Vision.

I. INTRODUCTION

Human activity recognition (HAR) is a capacity to translate human body signals or movement utilizing sensors and decide human action or activity[1]. Many human day-by-day errands can be rearranged or computerized on the off chance that they can be perceived through the HAR framework. HAR is considered as a significant segment in different logical research settings for example healthcare [2], Human-robot interaction [3], and surveillance [4]. Such systems are in high demand in a country like Saudi Arabia, wherein pilgrims' security is necessary for law and enforcement agencies. Terrorist attacks, especially explosive or suicide attacks, have become a major and critical threat to public protection. Suspicious activities like carrying a weapon, long-standing of a person at a public place, sudden running, snatching a mobile phone, fighting, suspicious behavior of people or threat for potential suicide bombing require urgent attention. These activities require a clever reconnaissance framework that can create a caution or alarm consequently. Presently a-days, many assaults are considered risky actions do by terrorists. A good timely prediction can prevent terrorist attacks and loss of human lives or may significantly limit the loss. HAR can have a large impact on various areas of human lives [5]. In particular, human activities represent the accompanying problems. Some machine learning approaches can automatically recognize these activities. Some challenging HAR scenarios include (a) Simultaneous activities where individuals can complete a few activities simultaneously. For example, while exercising, people talk with other people also via mobile phone, (b) Interleaved activities where people perform one task and then perform another task, (c) Ambiguity of prediction where some activities cause ambiguous predictions, for example, throwing a stone can be related to throwing the stone to people or throwing stone away from the road, (d) Multiple people where some environments have single person. In contrast, other scenarios contain multiple people or crowds, which is difficult to handle. The automated video surveillance system is especially needed in important public places like airports, railway stations, bus stands, commercial markets, banks, institutions, etc. It tries to detect or predict suspicious activities at public places with the help of an intelligent network of smart commercial of the shelf (COTS) video cameras. The intelligent CCTV surveillance reduces the workforce cost by automated observing the events and provides the second option to the management. Apart from its importance, the performance of the completely automatic human action and suspicious object prediction framework may be deteriorated because of numerous technical challenges. Some of the prominent challenges include: (a) occlusion, (b) illumination, (c) variations in objects size, (d) variations in appearances because of varying clothing, (e) the computational time is another major challenge, and (f) the people play out an action relies upon their body postures, which makes the issue of distinguishing the hidden movement very hard to decide. Additionally, developing a model for breaking down human developments progressively with insufficient benchmark datasets for assessment is a tricky job. In this work, a deep feature extraction methodology is presented for suspicious activity recognition to tackle the above-mentioned challenges. For this purpose, 63 layers of CNN-centered deep architecture is intended for feature acquisition. The acquired deep features are optimized through a feature selection algorithm. The foremost contributions are mentioned as under:

1) A dataset of 5 suspicious activities is prepared from HMDB51 [6] and AIDER [7] datasets.
2) A 63 layers CNN network, named as L4-Branched-ActionNet, is proposed. The network is initially trained with a third-party dataset and the features of a suspicious activity recognition dataset are extracted on this pre-trained network.
3) Entropy-coded ACS is applied for feature subset selection.
4) Various classifiers are utilized to monitor the top classifier's functioning.
5) The outcomes depict the acceptable accomplishment of the intended work.

The organization of this work is as follows: After the abstract, the introduction is presented in section 1. A brief overview of some existing literature is depicted in section 2. The proposed framework, along with the description of the proposed L4-Branched-ActionNet is expounded in section 3. The results and discussion are depicted in section 4. Finally, the conclusion is written in section 6.

II. LITERATURE REVIEW

Human recognition is exceptionally valuable in activity recognition [8, 9] since we must recognize the human (doing the activities) in the observation framework. Large numbers of papers are composed for Human activity recognition. One sort of research pursues face profile [10], gait [11], and silhouette [12]. Gait Image is utilized to depict human strolling properties. Item acknowledgment utilizing optical flow is additionally an imperative field for potential research [13-15]. The assistance of valuable features achieves computerized action acknowledgment, for example, HOG [16, 17], SIFT [18], LBP [19], and profound deep learning highlights. These features are utilized in a mix of Famous cutting-edge classifiers. It is increasingly worried by perceiving occasions initially (a precedent is a man getting a firearm [20]). There are two main strategies adopted for human activity prediction i.e. (a) The traditional approaches [21] employ handcrafted feature extraction methods, (b) The automatic features (Deep learning) [22] employ automatic feature extraction methods. Some major existing works performed in human activity recognition are discussed: Navel et. al [23] present an activity prediction approach based on streaming data. The proposed technique efficiently detects the activity having any significant change. Urbano et. al [24] present a framework for daily human actions recognition. The technique employs feature extraction. Two key poses enclose each activity frame, and static with max-min features are mined. Vadim Kantorov et. al [25] use an algorithm in which they discover feature encoding by Fisher vectors and determine accurate action recognition utilizing linear classifiers. Qicong Wang et. al [26] propose an algorithm that is useful to mine deep features from small video fragments. Baochang Zhang et. al [27] introduce a less complex descriptor termed 3D histogram texture to pull out discriminant features using a series of depth maps. Allah Bux Sargano et al [28] propose an innovative approach for human activity identification using the pre-trained structure of deep CNN for mining of features and depiction pursued by a fused SVM KNN categorizer for activity recognition. Additionally, there exists plenty of tasks in human acknowledgment with an essential job in motion recognition [29, 30]. A strategy for recognizing complex human exercises is proposed, including different people with intellectual insight into the computer. It is required to distinguish the human (understudy) conduct of irregular activities [31]. Albeit numerous analysts give high consideration to HAR to guarantee human wellbeing and perceive the observation framework's human suspicious movement. In PC vision, HAR is a profoundly engaged subject for specialists. Human strange action location or conduct identification is likewise pulling in more consideration of specialists. A few frameworks present for the human security reason. A. B. Mabrouk et al. [32] propose the strategy depends on two degrees of video handling, which speaks to a wise reconnaissance framework dependent on recordings; the target of this examination is to find a fascinating occasion from the huge number of recordings to counteract hazardous circumstances all the more effectively. M. M. Hassan et al. [33]consider the use of e-wellbeing persistent recovery also. An open dataset of human action acknowledgment is utilized to investigate the presentation of action acknowledgment with their proposed calculation. Deep learning is an emerging field, but machine learning can work better than deep learning in some workspace. Machine learning methods with bags of visual words help develop human action recognition applications [34]. Besides these popular techniques, Human action recognition uses other techniques like the LSTM network, Epileptic seizure classification, deep transfer learning approach, and hybrid transfer learning model [35-38]. Some researchers use the hybrid approach by merging the old techniques with the proposed ones. For example, in work, researchers use the CNN model with SVM and KNN on UCF sports and KTH dataset [39]. In transfer learning, the previously labeled data and knowledge is utilized to understand the upcoming situations. Single RGB is a technique that uses transfer learning techniques with the invariants of human actions. Deep ensemble learning is also used for recognizing human actions [40]. Key frame-based saliency detection and real-time action with 3D deep learning are also part of HAR [41, 42].

The arrangement of HAR is created as structures to enable the constant checking and examination of human practices in various zones, for example, clever human action and conduct investigation intelligent human activity and behavior analysis [33, 43-46], sports injury detection [47], patient rehabilitation [33], monitor activity shifts amid elderly citizens that might be helpful to detect and diagnose serious illness [48, 49], monitoring children's surveillance, hospital/patient monitoring [50, 51], recognition and classification of the human usual and unusual activities[26, 52-56], human behavior recognition and human activity detection [53, 57, 58], criminal tracking system [59, 60], automatic attendance system [61, 62], etc.

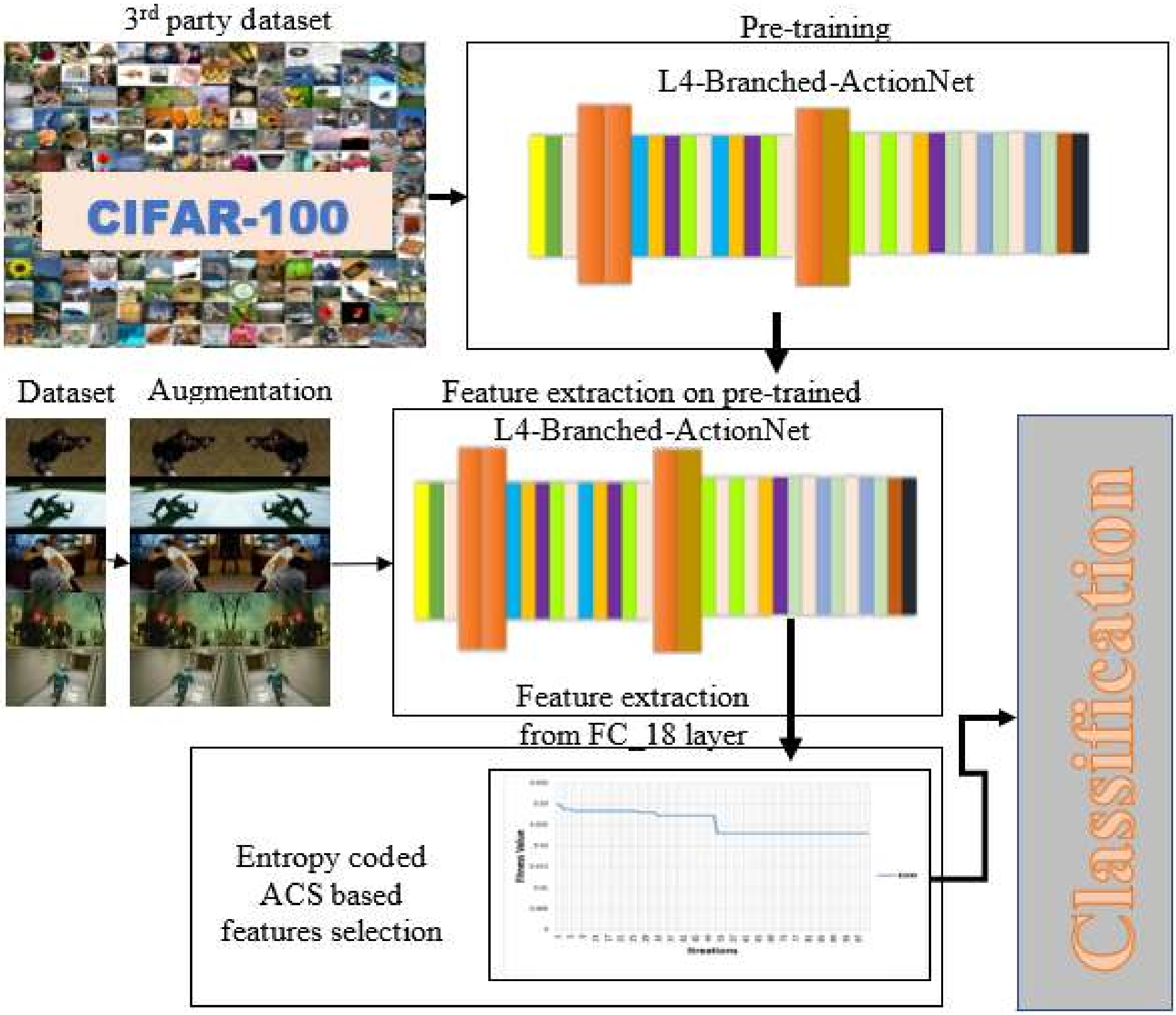


FIGURE 1. The intuition of the proposed L4-Branched-ActionNet based framework for suspicious activity recognition

III. MATERIAL AND METHODS

This segment depicts the complete sketch of the proposed framework. Also, the explanation of the suggested 63 layers CNN model is written. The framework's major steps include pre-training the proposed CNN architecture with the CIFAR-100 dataset, feature extraction of the action recognition dataset on the proposed CNN architecture, feature subset selection using (ACS) algorithm, and classification using various classifiers. Figure 1 offers a fleeting intuition of the suggested model. The different phases of the suggested framework are depicted in the text as follows:

A. L4-BRANCHED-ACTIONNET

A new CNN-based model is proposed consisting of 63 layers in an automatic features extraction and classification pipeline. The complete pipeline is termed as L4-Branched-ActionNet. The graphical arrangement of the proposed L4-Branched-ActionNet is exhibited in Figure 2. The layers' configuration and complete detail are represented in Table I. The backbone of the projected architecture pipeline is based on AlexNet [63]. AlexNet is composed of 25 layers having three different types of repeating blocks named here as T1, T2, and T3.

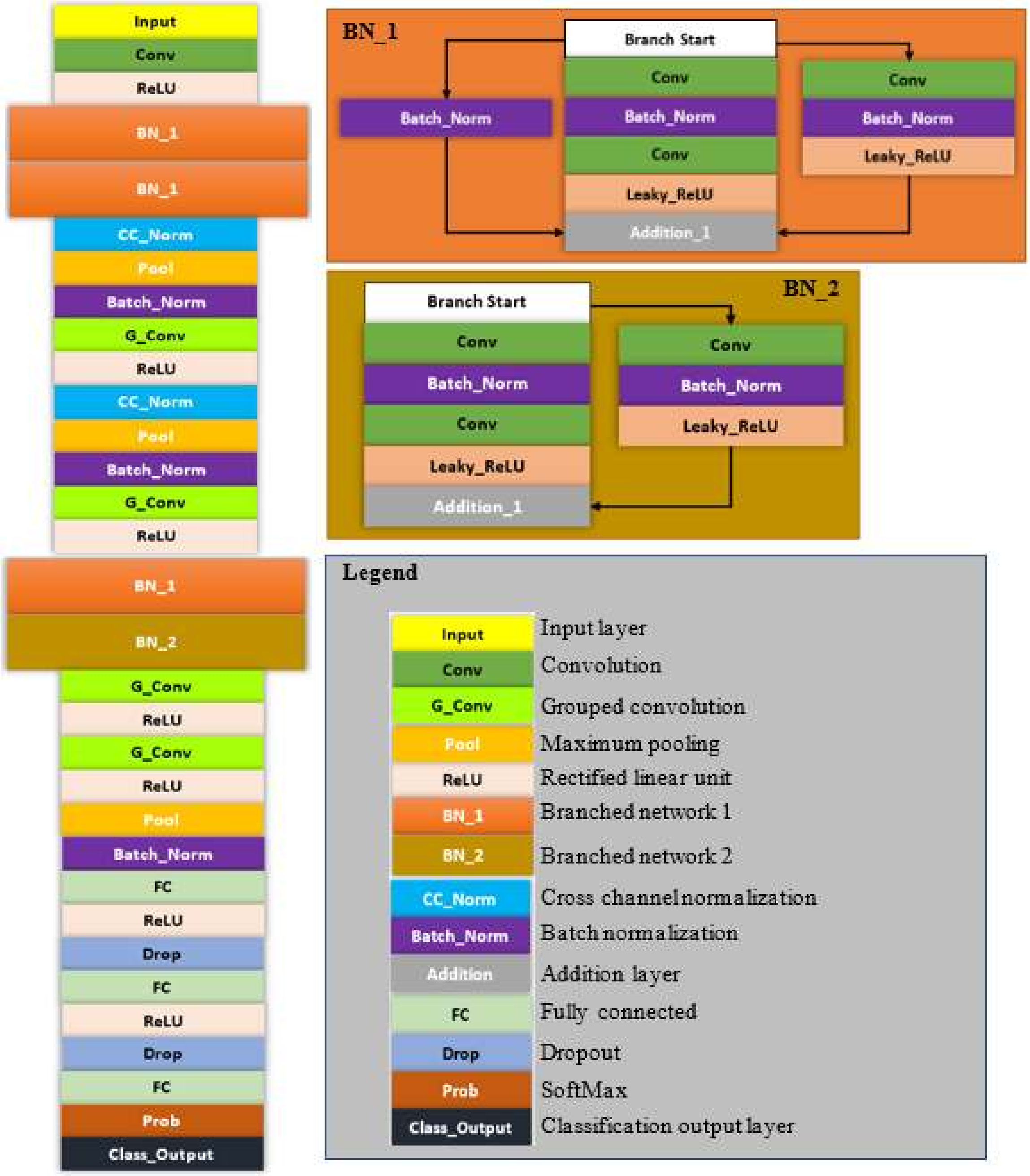


FIGURE 2. Structure of proposed L4-Branched-ActionNet

TABLE I
LAYERS CONFIGURATIONS OF L4-BRANCHED-ACTIONNET

Layer #	Layer name	Feature maps	Filter depth	Stride	Padding	Pooling window size/other values
1	Input	227 × 227 × 3				
2	Conv_1	55 × 55 × 96	11 × 11 × 3 × 96	[4 4]	[0 0 0 0]	
3	ReLU_1	55 × 55 × 96				
4	Batch_Norm_3	55 × 55 × 96				
5	Conv _4	55 × 55 × 96	5 × 5 × 96 × 96	[1 1]	Same	
6	Batch-Norm_2	55 × 55 × 96				
7	Conv_2	55 × 55 × 48	1 × 1 × 96 × 48	[1 1]	Same	
8	Leaky_ReLU_2	55 × 55 × 96				Scale 0.01
9	Batch_Norm_1	55 × 55 × 48				
10	Conv_3	55 × 55 × 96	11 × 11 × 48 × 96	[1 1]	Same	
11	Leaky_ReLU_1	55 × 55 × 96				Scale 0.01
12	Addition_1	55 × 55 × 96				
13	Batch_Norm_6	55 × 55 × 96				
14	Conv_7	55 × 55 × 96	5 × 5 × 96 × 96	[1 1]	Same	
15	Batch_Norm_5	55 × 55 × 96				
16	Leaky_ReLU_4	55 × 55 × 96				Scale 0.01
17	Conv_5	55 × 55 × 48	1 × 1 × 96 × 48	[1 1]	Same	
18	Batch_Norm_4	55 × 55 × 48				
19	Conv_6	55 × 55 × 96	11 × 11 × 48 × 96	[1 1]	Same	
20	Leaky_ReLU_3	55 × 55 × 96				Scale 0.01
21	Addition_2	55 × 55 × 96				
22	CC_Norm_1	55 × 55 × 96				
23	Pool_1	27 × 27 × 96		[2 2]	[0 0 0 0]	Max pooling 3 × 3
24	Batch_Norm_7	27 × 27 × 96				
25	G_Conv_8	27 × 27 × 256	Two groups of 5 × 5 × 48 × 128	[1 1]	[2 2 2 2]	
26	ReLU_2	27 × 27 × 256				
27	CC_Norm_2	27 × 27 × 256				
28	Pool_2	13 × 13 × 256		[2 2]	[0 0 0 0]	Max pooling 3 × 3
29	Batch_Norm_8	13 × 13 × 256				
30	G_Conv_9	13 × 13 × 384	3 × 3 × 256 × 384	[1 1]	[1 1 1 1]	
31	ReLU_3	13 × 13 × 384				
32	Batch_Norm_11	13 × 13 × 384				
33	Conv_10	13 × 13 × 192	1 × 1 × 384 × 192	[1 1]	Same	
34	Batch_Norm_9	13 × 13 × 192				
35	Conv_11	13 × 13 × 384	5 × 5 × 192 × 384	[1 1]	Same	
36	Leaky_ReLU_5	13 × 13 × 384				Scale 0.01
37	Conv_12	13 × 13 × 384	3 × 3 × 384 × 384	[1 1]	Same	
38	Batch_Norm_10	13 × 13 × 384				
39	Leaky_ReLU_6	13 × 13 × 384				Scale 0.01
40	Addition_3	13 × 13 × 384				
41	Conv_15	13 × 13 × 384	3 × 3 × 384 × 384	[1 1]	Same	
42	Batch_Norm_13	13 × 13 × 384				
43	Leaky_ReLU_8	13 × 13 × 384				Scale 0.01
44	Conv_13	13 × 13 × 192	1 × 1 × 384 × 192	[1 1]	Same	
45	Batch_Norm_12	13 × 13 × 192				
46	Conv_14	13 × 13 × 384	5 × 5 × 192 × 384	[1 1]	Same	
47	Leaky_ReLU_7	13 × 13 × 384				Scale 0.01
48	Addition_4	13 × 13 × 384				
49	G_Conv_16	13 × 13 × 384	Two groups of 3 × 3 × 192 × 192	[1 1]	[1 1 1 1]	
50	ReLU_4	13 × 13 × 384				
51	G_Conv_17	13 × 13 × 256	Two groups of 3 × 3 × 192 × 128	[1 1]	[1 1 1 1]	
52	ReLU_5	13 × 13 × 256				
53	Pool_3	6 × 6 × 256		[2 2]	[0 0 0 0]	Max pooling 3 × 3
54	Batch_Norm_14	6 × 6 × 256				
55	FC_18	1 × 1 × 4096		[1 1]	Same	
56	ReLU_6	1 × 1 × 4096				
57	Drop_1	1 × 1 × 4096				50% Dropout
58	FC_19	1 × 1 × 4096		[1 1]	Same	
59	ReLU_7	1 × 1 × 4096				50% Dropout
60	Drop_2	1 × 1 × 4096				
61	FC_20	1 × 1 × 100		[1 1]	Same	
62	Prob	1 × 1 × 100				
63	Class_Output					

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3091081, IEEE Access

After the data layer, there are two blocks of type T1 containing convolution (Conv) or grouped convolution (G_Conv), Rectified linear unit (ReLU), cross-channel normalization (CC_Norm), and maximum pooling (Pool) layers. After then, three blocks of type T2 are introduced that contain G_Conv and ReLU layers. Afterward, the pooling layer is inserted, which is proceeded by two T3 type blocks with fully connected (FC), ReLU, and dropout (Drop) layers. At last, FC, SoftMax classifier (Prob), and output (Class_Output) layers are added. In the proposed L4-Branched-ActionNet model, AlexNet is modified first by adding batch normalization (Batch_Norm) layers at the end of both type T1 blocks and the end of the third type T2 block. In addition to type T blocks, two types of branched subnetworks are introduced. The first type of branched subnetwork (BN_1) contains three branches. The primary branch entails only the Batch_Norm tier. The next branch covers Conv, Batch_Norm, Conv, and Leaky_ReLU tiers. The last branch contains Conv, Batch_Norm, and Leaky_ReLU layers. These three branches are fused with the help of an addition (Addition) Layer. In contrast, the second type of branched subnetwork (BN_2) is comprised of two branches. BN_2 is the same as BN_1, except it does not contain the branch having only the Batch_Norm layer. Two BN_1s are inserted after the ReLU layer in the first type T1 block. One BN_1 and one BN_2 subnetwork are inserted after the first type T2 block. The accompanying section contains a brief overview of the numerous types of layers used in the proposed L4-Branched-ActionNet.

In the Conv layer the input $\mathbb{I}_{j-1}$ is convolved with the filter bank. The convolution operation $*$ is mathematically represented as:

$$\mathbb{I}_{p',j} = \mathfrak{N}_j\left(\sum_p \mathfrak{F}_{j,p'p} * \mathbb{I}_{p,j-1} + \mathfrak{W}_{p',j}\right) \tag{1}$$

where p_j embody several channels of the input and p'_j portrays the number of channels that will be formed for output. The value j depicts the layer number [64]. $\mathfrak{F}$ illustrates a filter having depth p'_j and $\mathfrak{W}$ and $\mathfrak{N}$ signify nonlinear functions. In addition to conv layers, G_Conv layers are also embodied in the proposed model. A G Conv layer combines many convolutions into one. It was introduced mainly to enable the training process across clustered GPUs with low memory capacity. The filters are split into multiple divisions in a G Conv. All groups oversees a collection of 2D convolutions with a specific range. The Pool layers used are depicted mathematically as:

$$\mathbb{I}_{p,j,u,v} = \max_{l=1\ldots s,m=1\ldots t} \mathbb{I}_{p,j-1,(\mathrm{u+l})(\mathrm{v+m})} \tag{2}$$

where u, v are matrix index of image $\mathbb{I}_{p,j-1}$and l, m matrix index of the selected pooling window.

Both CC_Norm and Batch_Norm are deployed in the proposed scheme. Batch Norm [65] is a method for adjusting channel neurons over a small batch's defined amount. It calculates the mean and variance in fragments. The mean is derived, and the features are separated using the standard deviation. The mean of the batch $\mathbb{B} = \mathbb{I}_1, \ldots, \mathbb{I}_w$ is measured as follows:

$$Mean_{\mathbb{B}} = \frac{1}{w}\sum_{z=1}^{w} \mathbb{I}_z \tag{3}$$

here w represents the number of feature maps in a batch. The variance expression over the small batch is portrayed as:

$$Var_{\mathbb{B}} = \frac{1}{w}\sum_{z=1}^{w}(\mathbb{I}_z - Mean_{\mathbb{B}})^2 \tag{4}$$

Following expression is further used to normalize the features

$$\widehat{\mathbb{I}_Z} = \frac{\mathbb{I}_Z - Mean_{\mathbb{B}}}{\sqrt{Var_{\mathbb{B}} + \mathfrak{D}}} \tag{5}$$

here $\mathfrak{D}$ is the constant value used for consistency.

CC_Norm is embodied for generalization. The intention CC-Norm is to increase spatial-visual quality by using the maximal scaling factor of pixels for the previous layers locally. CC-Norm is interpreted as:

$$\overline{\overline{\mathbb{I}}}_z = \frac{\mathbb{I}_Z}{\left(\sigma + \frac{\beth * \mathfrak{I}}{\aleph}\right)^{\wp}} \tag{6}$$

where $\overline{\overline{\mathbb{I}}}_z$ is feature map acquired after CC-Norm. $\mathfrak{I}$ is the "sum of square" and $\aleph$ represents the channel size. $\sigma, \beth\ and\ \wp$ illustrate the standards applied for normalization.

The proposed CNN model employs both ReLU and Leaky_ReLU operations. The standard ReLU transforms all numbers that are lower than 0 to 0, which is expressed as [66]:

$$\mathbb{I}_{u,v} = \max(0, \mathbb{I}_{u,v}) \tag{7}$$

For values less than zero, Leaky ReLU has a small slope rather than being zero. A leaky ReLU will have v = 0.01u when u is negative.

CNN can further be learned in-depth from several works [67-69].

B. EXTRACTION OF DEEP FEATURES

The proposed approach is intended to feature extraction from the deep-trained CNN pipeline. Therefore, for pre-training, an existing third-party dataset such as CIFAR100 [70] is employed. CIFAR-100 is a repository of images with 100 classes. There are 500 learning and 100 validation images for each class. All the learning and validation images are mixed for pre-training, making 600 images in every class. The mixed dataset is supplied to the proposed CNN model for training. The trained network is then used for feature extraction on action recognition datasets and the FC_18 layer is chosen for features extraction. Total 4096 features are attained per image from the FC_18 layer. The prepared dataset contains a total of 13250 images. This makes the feature set dimension of all datasets 13250×4096. Figure 3 illustrates the visualizations of the strongest feature maps at various convolution layers on L4-Branched-ActionNet.

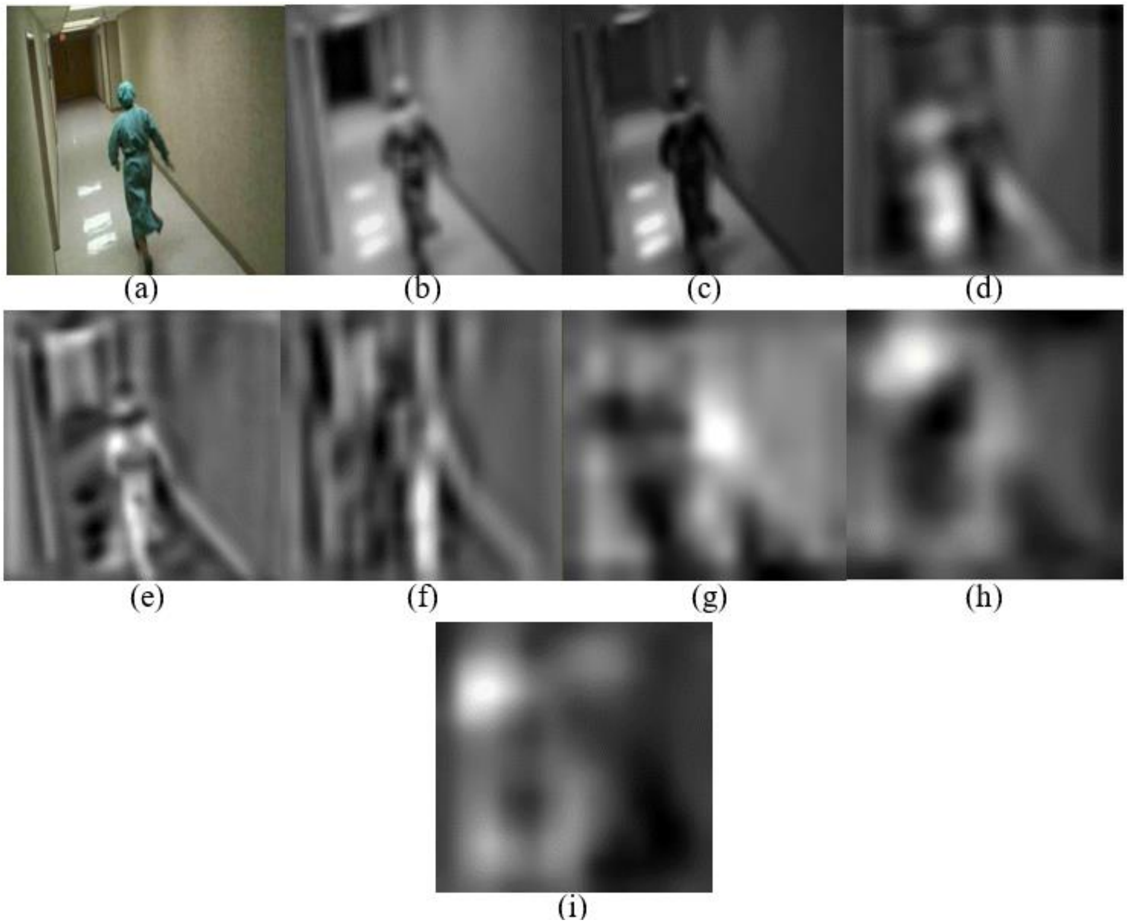


FIGURE 3. Image visualizations of strongest feature maps at various convolution layers (a) Conv_1, (b) Conv_2, (c) Conv_3, (d) Conv_5, (e) G_Conv_8, (f) Conv_10, (g) Conv_12, (h) Conv_15, (i) G_Conv_17

C. ENTROPY CODED ACS OPTIMIZATION

The extracted features are coded by applying entropy operation [71]. Entropy ℮ is famous for scoring the features. It is expressed mathematically as:

$$\text{℮}(\mathbb{I}'_1, \dots, \mathbb{I}'_n) = -\sum_{\mathfrak{f}_1} \dots \sum_{\mathfrak{f}_n} \text{þ}(\mathfrak{f}_1, \dots, \mathfrak{f}_n)\, LOG\, \text{þ}(\mathfrak{f}_1, \dots, \mathfrak{f}_n) \tag{8}$$

where $(\mathbb{I}'_1, \dots, \mathbb{I}'_n)$ illustrate the features, $(\mathfrak{R}_1, \dots, \mathfrak{R}_n)$ are embodied as the related random variable value. $\text{þ}(\mathfrak{R}_1, \dots, \mathfrak{R}_n)$ is the probability of $(\mathfrak{R}_1, \dots, \mathfrak{R}_n)$.

ACS is a learning-based feature selection approach. It becomes an embedded approach when combined with a filter-based strategy i.e. entropy-based feature scoring. The entropy-coded scores attained in the previous step are supplied to ACS (which is based on probability theory) for feature optimization.

ACS is centered on the activities and movements of ants [72]. The ants disperse an ant deposit material called "pheromone" as they move from one place to another. As time passes, the strength of this material decreases. The ants pursue the track with a strong probability of pheromone. This encourages ants to take the cheaper path. Thus, Ants move from one place to another in the same way as to move from vertex to vertex in a graph. A vertex represents a feature, while the edges between vertices indicate the choice to choose the next feature. The approach iterates to looks for the best features. When the smallest number of vertices are accessed and a freezing criterion is fulfilled, the approach comes to a halt. The vertices are linked in a mesh-like arrangement. An ant's choice of features is based on the likelihood at a given point at a certain time, which can be expressed as:

$$\text{þ}_j^m(\text{☨}) = \begin{cases} \dfrac{|\,\mathfrak{h}_j(\mathcal{T})|^{\bar{\ddot{\mathfrak{w}}}}|\,\mathfrak{S}_j|^{\mathbb{W}}}{\sum_{v\in\text{℮}(\mathbb{I}'_1,\ldots,\mathbb{I}'_n)}|\,\mathfrak{h}_v(\mathcal{T})|^{\ddot{\mathfrak{w}}}|\,\mathfrak{S}_v|^{\mathfrak{w}}} \; if\ j \in \text{℮}(\mathbb{I}'_1, \ldots, \mathbb{I}'_n) \\ 0\ otherwise \end{cases} \tag{9}$$

where $\text{℮}(\mathbb{I}'_1, \ldots, \mathbb{I}'_n)$ = entropy-coded features

$\mathfrak{h}_j(\text{☨})$ = pheromone value
$\mathfrak{S}_j$ = empirical value
$\ddot{\mathfrak{w}}$ = pheromone cost
$\mathfrak{w}$ = pragmatic knowledge
$\mathcal{T}$ = time limit

$\mathfrak{h}_j(\text{☨})$ and $\mathfrak{S}_j$ are connected to the jth feature. If the features have not yet been explored, they are considered as a part of the incomplete response.

D. CLASSIFICATION

The entropy-coded ACS-based chosen features are at the end passed to the predictor for categorization. The various SVM versions (support vector machine) and KNN (K nearest neighbors) are deployed to observe the system performance. Such versions include linear SVM (Lin-SVM), quadratic SVM (Quad-SVM), fine Gaussian SVM (Fin-Gaus-SVM), medium Gaussian SVM (Med-Gaus-SVM), coarse Gaussian SVM (Cor-Gaus-SVM), cubic SVM (Cub-SVM), cosine KNN (Cos-KNN), coarse KNN (Cor-KNN), and fine KNN (Fin-KNN). The detailed study of SVM versions can be retrieved from [73-80] while the detailed study of KNN versions can be regained from [81-85]. Observing the performance outcomes, Cub-SVM becomes the best-performed classifier for the selected action datasets. The comprehensive discussion on the tests conducted is illustrated in the subsequent section.

IV. RESULTS AND DISCUSSION

The main goal of our research is to create a CNN structure that can tackle the supplied dataset. The Deep L4-Branched-ActionNet CNN Network suggested here is solely utilized to extract powerful features following feature selection. The pretraining is performed on a third-party dataset i.e., CIFAR-100 because this research is aimed to acquire the features on given datasets using the proposed CNN. Also, the proposed 63 layers Deep L4-Branched-ActionNet CNN Network is created after extensive experimentation. Different approaches are followed to finalize this architecture. The foremost approaches include fine-tuning, adding, and removing different layers. In its final form, the 63 layers architecture is found good with the best outcomes in terms of performance. The discussion and interpretation of the outcomes of the proposed framework are described in this portion. The dataset is defined first in this section. The procedure for evaluating performance is then portrayed. Finally, the tests are well discussed. All the mentioned experiments in this manuscript are conducted on a Pentium core i-5 system containing 8 GB of memory. The training is aided with NVIDIA GTX 1070 GPU comprising 8GB RAM. The coding is performed by using MATLAB2020a.

A. DATASET

The training and performance evaluation in this work is performed on two different datasets. One data set is prepared and focused on suspicious activities. The other dataset is the Weizmann action dataset [86].

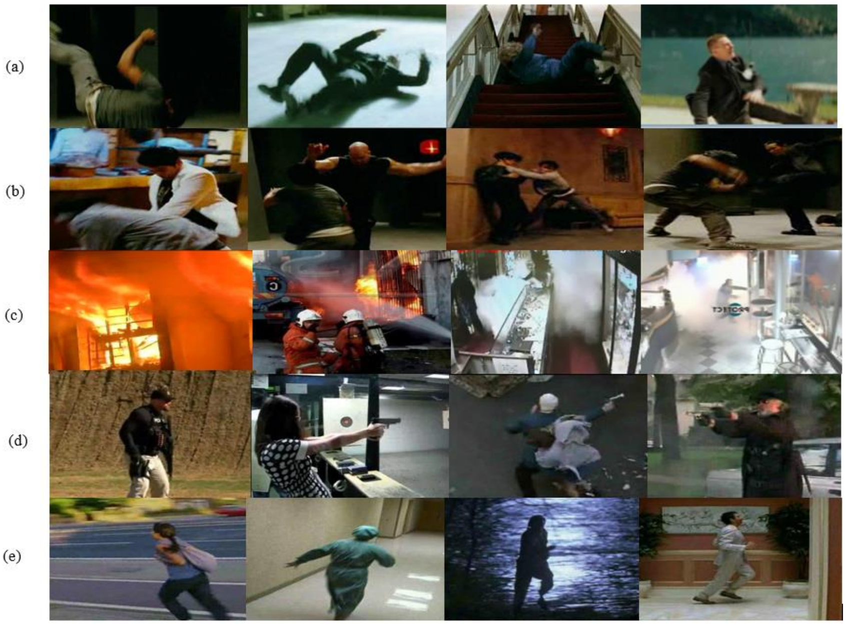


FIGURE 4. Some sample images of the prepared suspicious activity recognition dataset illustrating five classes: (a) Falling on the floor, (b) Persons fighting, (c) Fire or smoke, (d) Person firing, (e) Person running.

The suspicious activity dataset (see Figure 4 for sample images) is prepared from HMDB51 [6] and AIDER [7] datasets. HMDB51 is a huge action dataset comprised of 51 various human actions. The dataset is based on 7000 videos acquired from YouTube and numerous movie shows. Four suspicious activities, such as a person falling on the floor, persons fighting, Person firing, and person running, are selected from the HMDB51 dataset.

The videos of these activities are selected, and the images are extracted and annotated to form action classes. One class of fire or smoke is collected and annotated from the AIDER dataset. The formed dataset is then augmented by applying the mirroring of the images. Figure 5 illustrates the sample image with mirror augmentation.

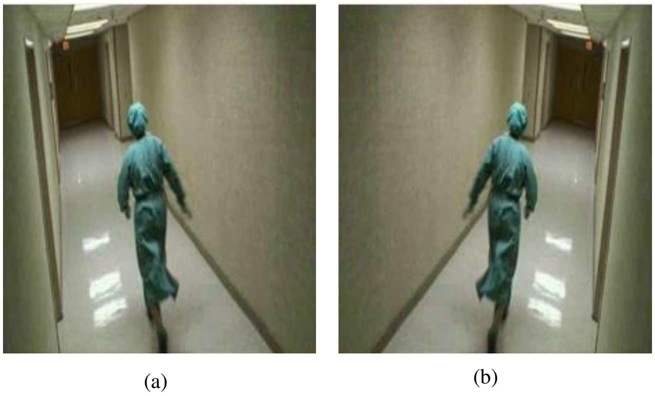


FIGURE 5. A sample image showing dataset augmentation: (a) original image, (b) flipped image

The detailed makeup of the dataset with augmentation is displayed in Table II. The Weizmann dataset is composed of 90 videos of nine people depicting ten action classes. The image composition is illustrated in Figure 6 and the sample images of the Weizmann dataset are shown in Figure 7.

TABLE II
PREPARED SUSPICIOUS ACTION DATASET DESCRIPTION

Class	Original	Augmented
Falling	1263	2526
Fighting	1398	2796
Fire	1449	2898
Firing	1373	2746
Running	1142	2284
Total	**6625**	**13250**

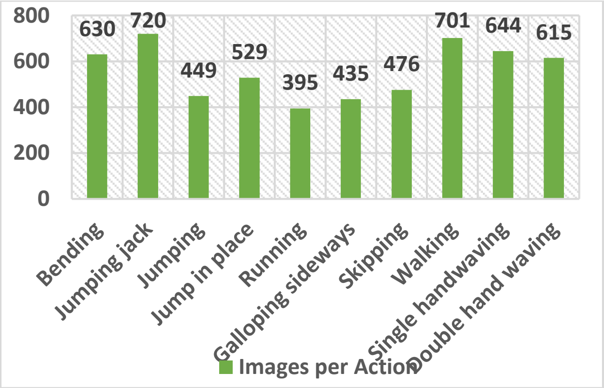


FIGURE 6. Weizmann action dataset description

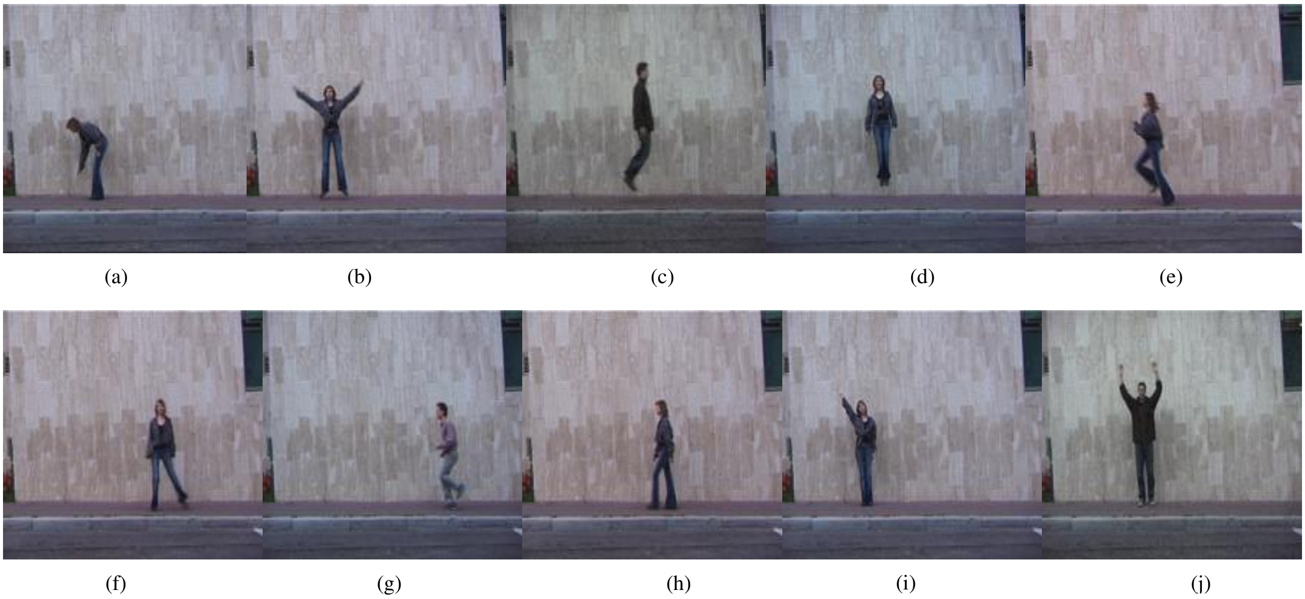


FIGURE 7. Some sample images of the Weizmann activity recognition dataset illustrating ten classes: (a) Bending, (b) Jumping jack, (c) Jumping, (d) Jump in place, (e) Running, (f) Galloping sideways, (g) Skipping, (h) Walking, (i) Single handwaving, and (j) Double hand waving

B. PERFORMANCE EVALUATION PROTOCOLS

The 5-folds mechanism for the cross-validation is employed for booth learning and assessment. To assess the efficacy of the proposed research, various evaluation assessment procedures were used in this manuscript. The confusion matrix created during the classification challenge's testing process is used in the majority of these protocols. These protocols are depicted as under:

$$\text{Accuracy (Acy)} = \frac{\text{TruePositives+TrueNegatives}}{\text{TruePositives+TrueNegatives+FalsePositives+FalseNegatives}} \tag{10}$$

$$Sensitivity\ (Sny) = \frac{\text{TruePositives}}{\text{TruePositives+FalseNegatives}} \tag{11}$$

$$Specificity\ (Spe) = \frac{\text{TrueNegatives}}{\text{TrueNegatives+FalsePositives}} \tag{12}$$

$$\text{The area under the (AUC)}= \frac{\text{TruePositives+TrueNegatives}}{\text{TruePositives+TrueNegatives+FalsePositives+FalseNegatives}} \tag{13}$$

$$Precision\ (Prc) = \frac{\text{TruePositives}}{\text{TruePositives+FalsePositives}} \tag{14}$$

$$\text{F} - \text{measure (FM)} = 2 \times \frac{Prc \times Sny}{Prc + Sny} \tag{15}$$

$$G - Mean\ (GM) = \sqrt{TruePositiveRate \times TrueNegativeRate} \tag{16}$$

5-fold cross-validation is used for ground reality class marks. It makes up 80 percent of the data for each fold is chosen at random for preparation, while the remaining 20% is chosen at random for testing.

C. EXPERIMENTS BRIEF ON THE SUSPICIOUS ACTIVITY DATASET

Extensive testing of various iterations of selected features is carried out to find the best results. A few main reviews are listed in this section. Table III provides a succinct summary of the accuracy of each test discussed.

TABLE III
A SUCCINCT SUMMARY OF TESTS PERFORMED ON THE SUSPICIOUS ACTIVITY DATASET

Experiment #	Selected Features	Best Accuracy
1	100	0.9844
2	250	0.9886
3	500	0.9915
4	750	0.9919
5	1000	0.9924

At the feature selection stage, several tests are run with various numbers of features. Table IV depicts the parameter values used for the ACS optimization approach.

TABLE IV
PARAMETERS VALUES USED FOR ACS OPTIMIZATION

Parameter name	Value
Total aunts	10
Maximum iterations	100
Empirical value	1
Pheromone	0.2
Phi	0.5

The outcomes of just five tests are shown. Figure 8 illustrates the plot of fitness values for the tests depicted in Table II. The finest fitness is found for electing 1000 features. The brief on performed tests is presented in the accompanying text. Hyperparameters values used for training of proposed CNN model is presented in Table V.

TABLE V
HYPERPARAMETERS USED FOR PROPOSED CNN MODEL

Hyperparameters	Value
Initial learning rate	0.001
Optimization	Stochastic gradient descent with momentum
Total epochs used	30
Mini batch	128
Momentum value	0.9

Table VI provides a detailed overview of the experimental tests performed with different selected features.

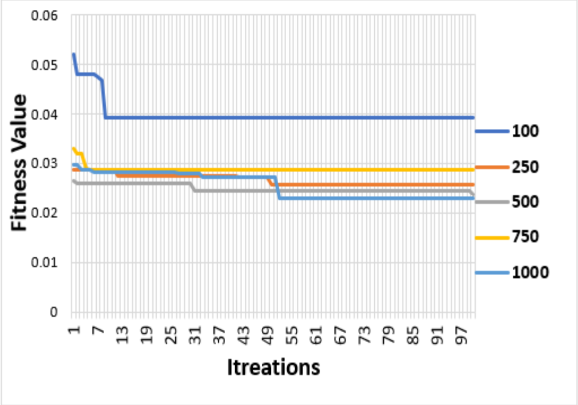


FIGURE 8. **Fitness plot various number of selected features using entropy coded ACS: (a) 100, (b) 250, (c) 500, (d) 750 and (e) 1000**

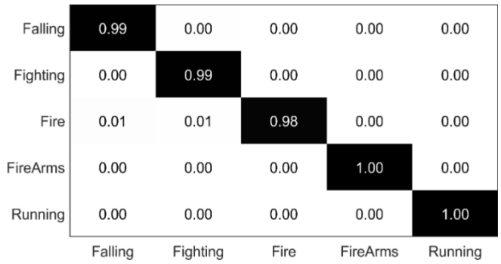
The very first experiment involves using the entropy-coded ACS function to choose 100 features. The combined feature vectors, which contain all dataset image features grow to

13250×100. This feature matrix is used for the automated marking of prediction models based on chosen classifiers. Cub-SVM achieves the overall highest performance in all performance measures having Acy, Sny, Spe, Prc, FM, and GM as 0.9844, 0.9880, 0.9812, 0.9258, 0.9606, and 0.9896, respectively in this test. Fin-KNN performance is found to be the second-best in terms of Acy, Sny, and GM only, as 0.9763, 0.9947, and 0.9833 respectively while Spe, Prc, FM are observed as second best for Quad-SVM. The second experiment tests on 250 chosen features. The joint feature vectors, which contain all dataset image features, now become 13250×250. Cub-SVM attains the overall highest performance in all performance measures having except Sny having Acy, Spe, Prc, FM, and GM as 0.9886, 0.9883, 0.9224, 0.9707, and 0.9890, respectively. Sny of Fin-KNN is found best in this test. The second-best outcomes for Acy, Sny, Spe, Prc, FM, and GM achieved are with Med-Gaus-SVM (0.9873), Cub-SVM (0.9897), Med-Gaus-SVM (0.9883), Med-Gaus-SVM (0.9517), Med-Gaus-SVM (0.9673), and Fin-KNN (0.9867), respectively.

TABLE VI
PERFORMANCE EVALUATION FOR DIFFERENT FEATURES ON SUSPICIOUS ACTIVITY DATASET (DARK BLACK AND BOLD VALUES REPRESENT BEST RESULTS, WHILE GRAY WITH BOLD VALUES DEPICT SECOND BEST OUTCOMES)

Features	Classifier	Acy (%)	Sny (%)	Spe (%)	Prc (%)	FM (%)	GM (%)
100	Lin-SVM	0.8326	0.7926	0.8420	0.5417	0.6435	0.8169
	Quad-SVM	0.9750	0.9747	**0.9751**	**0.9022**	**0.9370**	0.9749
	Fin-Gaus-SVM	0.9406	0.8939	0.9516	0.8131	0.8516	0.9223
	Med-Gaus-SVM	0.9460	0.9553	0.9438	0.8001	0.8708	0.9495
	Cor-Gaus-SVM	0.7637	0.7154	0.7751	0.4283	0.5358	0.7446
	Cub-SVM	**0.9844**	**0.9980**	**0.9812**	**0.9258**	**0.9606**	**0.9896**
	Cos-KNN	0.9656	0.9711	0.9643	0.8650	0.9150	0.9677
	Cor-KNN	0.8226	0.6956	0.8526	0.5264	0.5992	0.7701
	Fin-KNN	**0.9763**	**0.9949**	0.9719	0.8930	0.9412	**0.9833**
250	Lin-SVM	0.9089	0.8812	0.9154	0.7105	0.7867	0.8982
	Quad-SVM	0.9750	0.9747	0.9751	0.9022	0.9370	0.9749
	Fin-Gaus-SVM	0.8303	0.7106	0.8585	0.5420	0.6149	0.7811
	Med-Gaus-SVM	**0.9873**	0.9834	**0.9883**	**0.9517**	**0.9673**	0.9858
	Cor-Gaus-SVM	0.8574	0.8211	0.8660	0.5907	0.6871	0.8432
	Cub-SVM	**0.9886**	**0.9897**	**0.9883**	**0.9524**	**0.9707**	**0.9890**
	Cos-KNN	0.9728	0.9766	0.9718	0.8909	0.9318	0.9742
	Cor-KNN	0.8269	0.6968	0.8575	0.5353	0.6054	0.7730
	Fin-KNN	0.9800	**0.9976**	0.9758	0.9068	0.9500	**0.9867**
500	Lin-SVM	0.9455	0.9283	0.9496	0.8125	0.8666	0.9389
	Quad-SVM	0.9877	0.9885	0.9875	0.9491	0.9684	0.9880
	Fin-Gaus-SVM	0.7611	0.6180	0.7948	0.4149	0.4965	0.7008
	Med-Gaus-SVM	**0.9930**	0.9865	**0.9945**	**0.9769**	**0.9817**	**0.9905**
	Cor-Gaus-SVM	0.9202	0.9145	0.9215	0.7329	0.8137	0.9180
	Cub-SVM	**0.9915**	**0.9933**	**0.9910**	**0.9631**	**0.9780**	**0.9922**
	Cos-KNN	0.9758	0.9830	0.9741	0.8993	0.9393	0.9785
	Cor-KNN	0.8349	0.7403	0.8572	0.5498	0.6310	0.7966
	Fin-KNN	0.9820	**0.9980**	0.9783	0.9154	0.9549	0.9881
750	Lin-SVM	0.9602	0.9485	0.9630	0.8579	0.9009	0.9557
	Quad-SVM	**0.9882**	0.9897	0.9879	0.9506	**0.9697**	0.9888
	Fin-Gaus-SVM	0.7195	0.5435	0.7610	0.3488	0.4249	0.6431
	Med-Gaus-SVM	0.9873	0.9755	**0.9901**	**0.9588**	0.9670	0.9828
	Cor-Gaus-SVM	0.9504	0.9382	0.9533	0.8255	0.8783	0.9457
	Cub-SVM	**0.9919**	**0.9921**	**0.9919**	**0.9664**	**0.9791**	**0.9920**
	Cos-KNN	0.9765	0.9834	0.9749	0.9023	0.9411	0.9791
	Cor-KNN	0.8232	0.7067	0.8507	0.5272	0.6039	0.7753
	Fin-KNN	0.9843	**0.9988**	0.9809	0.9249	0.9604	**0.9898**
1000	Lin-SVM	0.9661	0.9592	0.9677	0.8750	0.9152	0.9635
	Quad-SVM	**0.9885**	0.9861	**0.9891**	**0.9551**	**0.9704**	**0.9876**
	Fin-Gaus-SVM	0.6943	0.5091	0.7380	0.3140	0.3884	0.6129
	Med-Gaus-SVM	0.9744	0.9470	0.9809	0.9211	0.9338	0.9638
	Cor-Gaus-SVM	0.9676	0.9620	0.9689	0.8795	0.9189	0.9655
	Cub-SVM	**0.9924**	**0.9925**	**0.9924**	**0.9683**	**0.9803**	**0.9924**
	Cos-KNN	0.9780	**0.9869**	0.9759	0.9062	0.9449	0.9814
	Cor-KNN	0.8308	0.7102	0.8592	0.5430	0.6154	0.7812
	Fin-KNN	0.8308	0.7102	0.8592	0.5430	0.6154	0.7812

A total of 500 features are selected in the third test. The pool of feature vectors here contains features of all dataset images gets the dimension of 13250×500. Cub-SVM achieves only the best results for GM only in this test. However, it attains the second-best outcome in remaining all measures. Here, Med-Gaus-SVM shows its overall dominance with Acy, Spe, Prc, and FM as 0.9930, 0.9945, 0.9769, and 0.9817, respectively. The fourth experiment is conducted on 750 features. The joint feature vector, containing all dataset image features now becomes 13250×750.



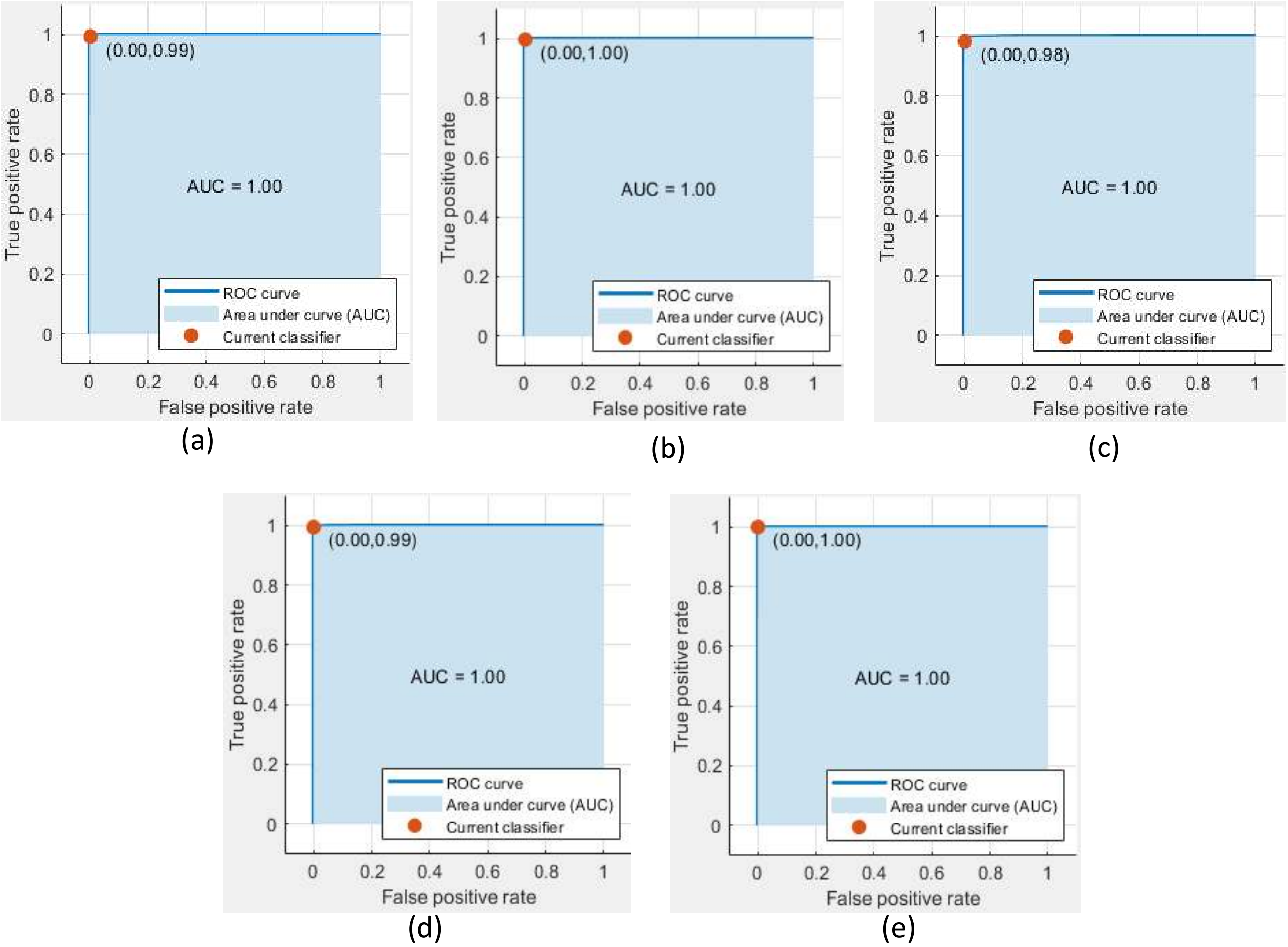
	Falling	Fighting	Fire	FireArms	Running
Falling	0.99	0.00	0.00	0.00	0.00
Fighting	0.00	0.99	0.00	0.00	0.00
Fire	0.01	0.01	0.98	0.00	0.00
FireArms	0.00	0.00	0.00	1.00	0.00
Running	0.00	0.00	0.00	0.00	1.00

FIGURE 9. **Confusion matrix of finest outcome and Cub-Svm classifier (1000 features) on suspicious activity dataset**

Again, Cub-SVM shows its prominence with Acy, Spe, Prc, and FM values as 0.9919, 0.9919, 0.9664, and 0.9791, respectively. The second-best outcomes for Acy, Sny, Spe, Prc, FM and GM achieved are from Quad-SVM (0.9882), Cub-SVM (0.9921), Med-Gaus-SVM (0.9901), Med-Gaus-SVM (0.9588), Quad-SVM (0.9697), and Cub-SVM (0.9920) respectively.

In conducting the fifth experiment, 1000 features are chosen. This test is considered best with 1000 features. The pool of feature vectors here contains features of all dataset images gets the dimension of 13250×1000. Cub-SVM is found to be the best in all performance measures with Acy, Sny, Spe, Prc, FM, and GM as 0.9924, 0.9925, 0.9924, 0.9863, 0.9803, and 0.9924. Here, Quad-SVM shows its overall worth as the second-best results. Except for Sny, Quad-SVM achieved the highest values in all performance measures. The highest accuracy on Cub-SVM using 1000 features is picked as the best outcome.

The confusion matrix of the finest result on 1000 features on Cub-SVM is illustrated in Figure 9. The classes Firearms and Running attain 100 percent accuracy while Falling, Fighting, and Fire classes achieve 99, 99, and 98 percent accuracy. The AUCs (as shown in Figure 10) portray 100 percent outcomes for all classes under best features (1000) with a Cub-SVM classifier.

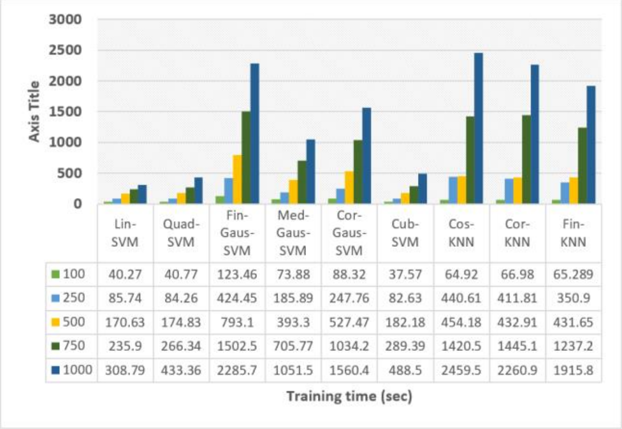


(a) (b) (c)

(d) (e)

FIGURE 10. **ROCs and AUCs of all classes having best results using Cub-Svm classifier (1000 features) on suspicious activity dataset**

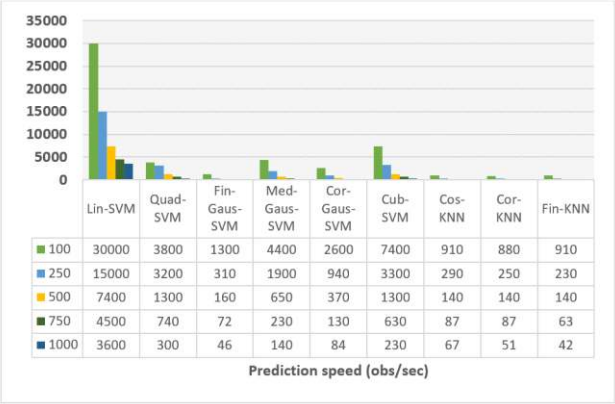
Figure 11 reveals the observations on training times on all classifiers with different feature counts.



	Lin-SVM	Quad-SVM	Fin-Gaus-SVM	Med-Gaus-SVM	Cor-Gaus-SVM	Cub-SVM	Cos-KNN	Cor-KNN	Fin-KNN
🟩 100	40.27	40.77	123.46	73.88	88.32	37.57	64.92	66.98	65.289
🟦 250	85.74	84.26	424.45	185.89	247.76	82.63	440.61	411.81	350.9
🟨 500	170.63	174.83	793.1	393.3	527.47	182.18	454.18	432.91	431.65
🟩 750	235.9	266.34	1502.5	705.77	1034.2	289.39	1420.5	1445.1	1237.2
🟦 1000	308.79	433.36	2285.7	1051.5	1560.4	488.5	2459.5	2260.9	1915.8

FIGURE 11. Training times (sec) on different selected features and different classifiers

It is noticed that the training time rises with the growth in the number of features. Overall, KNN versions of classifiers take more time as compared to SVM classifiers. Only Fin-Gaus-SVM takes a lot of time that is 2285 sec for 1000 features. Cub-SVM on the other hand takes a reasonable training time i.e., 488.5 sec. Prediction speeds can be observed in Figure 12. The prediction speed reduces with the surge in the features count. Except for Lin-SVM, Cub-SVM prediction speed is found better than the prediction speed of other classifiers.



	Lin-SVM	Quad-SVM	Fin-Gaus-SVM	Med-Gaus-SVM	Cor-Gaus-SVM	Cub-SVM	Cos-KNN	Cor-KNN	Fin-KNN
🟩 100	30000	3800	1300	4400	2600	7400	910	880	910
🟦 250	15000	3200	310	1900	940	3300	290	250	230
🟨 500	7400	1300	160	650	370	1300	140	140	140
🟩 750	4500	740	72	230	130	630	87	87	63
🟦 1000	3600	300	46	140	84	230	67	51	42

FIGURE 12. Prediction speed (obs/sec) on different selected features and different classifiers

D. EXPERIMENTS BRIEF ON THE SUSPICIOUS ACTIVITY DATASET

The experiment tests are repeated with the best-considered configuration (1000 features using entropy-coded ACS feature selection) on the Weizmann action dataset to observe the proposed framework's performance. Table VII portrays the performance results attained on the Weizmann dataset.

TABLE VII
PERFORMANCE EVALUATION FOR 10000 FEATURES ON WEIZMANN DATASET

Classifier	Acy (%)	Sny (%)	Spe (%)	Prc (%)	FM (%)	GM (%)
Lin-SVM	0.9004	1.0000	0.8878	0.5307	0.6935	0.9422
Quad-SVM	0.9668	1.0000	0.9625	0.7721	0.8714	0.9811
Fin-Gaus-SVM	0.5400	0.6952	0.5203	0.1554	0.2540	0.6015
Med-Gaus-SVM	0.9578	1.0000	0.9525	0.7275	0.8422	0.9759
Cor-Gaus-SVM	0.8400	1.0000	0.8197	0.4131	0.5847	0.9054
Cub-SVM	**0.9800**	**1.0000**	**0.9770**	**0.8468**	**0.9170**	**0.9885**
Cos-KNN	0.8311	1.0000	0.8096	0.4000	0.5714	0.8998
Cor-KNN	0.5930	0.7032	0.5790	0.1749	0.2801	0.6381
Fin-KNN	**0.9732**	**1.0000**	**0.9698**	**0.8077**	**0.8936**	**0.9848**

Again, the Cub-SVM attains best results with Acy, Sny, Spe, Prc, FM, and GM as 0.9800, 1.0, 0.9770, 0.8468, 0.9170, and 0.9885, respectively. Fin-KNN is found as the second-best in all performance measures.

Figure 13 shows the confusion matrix by using 1000 features and Cub-SVM on the Weizmann dataset. The true positives of individual classes depict that except running class, all classes have accuracy greater or equal to 0.9600.

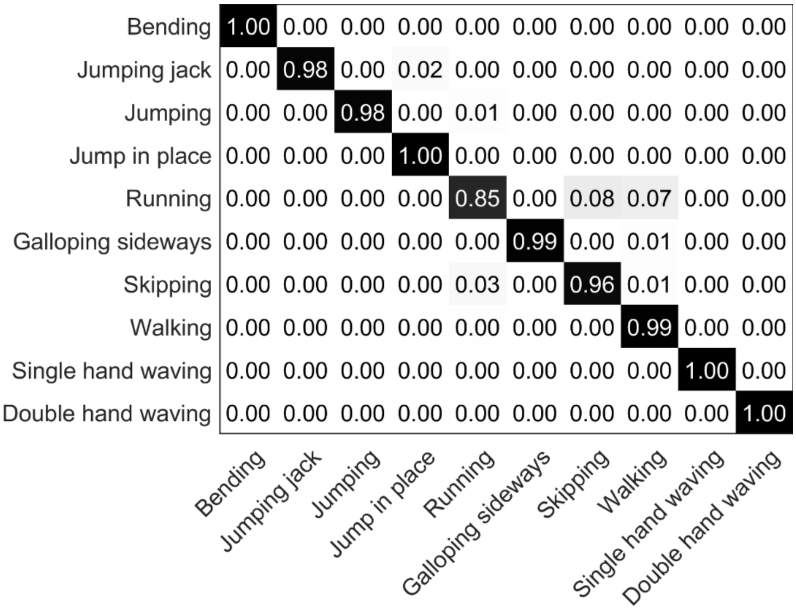
E. ACCURACY COMPARISON WITH EXISTING WORKS ON WEIZMANN DATASET

The proposed work is contrasted with some recent existing methodologies, as depicted in Table VIII.

TABLE VIII
PERFORMANCE EVALUATION FOR 10000 FEATURES ON WEIZMANN DATASET

Method reference	Year	Accuracy
DWT+KNN [87]	2020	0.9666
CNN+ELM [88]	2020	0.9870
Gabor-Ridgelet Transform [87]	2020	0.9666
LCF + MSVM [89]	2021	0.9730
ANN [90]	2020	0.8600
PCANet-XY-YT [91]	2021	0.9333
Ours (L4-Branched-ActionNet + EntACS + Cub-SVM)	**-**	**0.9800**

The result illustrates the better outcome as compared to some recent works.



	Bending	Jumping jack	Jumping	Jump in place	Running	Galloping sideways	Skipping	Walking	Single hand waving	Double hand waving
Bending	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Jumping jack	0.00	0.98	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.00
Jumping	0.00	0.00	0.98	0.00	0.01	0.00	0.00	0.00	0.00	0.00
Jump in place	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00
Running	0.00	0.00	0.00	0.00	0.85	0.00	0.08	0.07	0.00	0.00
Galloping sideways	0.00	0.00	0.00	0.00	0.00	0.99	0.00	0.01	0.00	0.00
Skipping	0.00	0.00	0.00	0.00	0.03	0.00	0.96	0.01	0.00	0.00
Walking	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.99	0.00	0.00
Single hand waving	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00
Double hand waving	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00

FIGURE 13. Confusion matrix of the best outcome and cub-SVM classifier (1000 features) on Weizmann dataset

The proposed framework is tested for the scalability issues in terms of the number of features. It is observed that increasing the number of features raises the accuracy a little bit. Also, increasing the number of features increases the computation time as well. To tackle this, feature selection is utilized. In addition, the model is also tested with two different datasets: one with four classes and the other one with ten classes. The acceptable outcomes prove the strength of the proposed framework

V. Conclusion

Suspicious activity recognition has been an important area of research in recent years. By recognizing suspicious activities automatically in a well-timed manner will help to reduce financial and human losses. This work is encompassed to classify the suspicious activities using a proposed 63 layers CNN network named L4-Branched-ActionNet. The network is pre-trained first on the CIFAR-100 object detection dataset. The dataset of five suspicious activities is then prepared and passed to the proposed pre-trained L4-Branched-ActionNet to extract features. The features are then fed to an entropy-coded ACS scheme to reduce the features. The training and testing of all datasets' selected features are performed with different variation versions of SVM and KNN categorizers. The findings are repeated on these classifiers by altering the number of features (5 experiments are mentioned in this manuscript with 100, 250, 750, and 1000 features) at the feature choice phase. The lower performance is attained on 100 features with an accuracy of 0.9844 with the Cub-SVM classifier. The best classification results are considered with 1000 features using a Cub-SVM classifier having an accuracy of 0.9924. The Cub-SVM is found to be the overall best, having better performance in all experiments. The results are also validated on the Weizmann dataset and compare with recent works. The acceptable and comparable results demonstrate the legitimacy of the suggested approach.

Feature fusion can be implemented by taking features from another CNN-based pretrained network. Existing works show superior outcomes in this regard. However, we suggest this task be explored in the upcoming future. Moreover, new deep learning building blocks and feature selection methods can be checked in this domain for a dominant performance as to future work.

ACKNOWLEDGEMENT

"This work was supported by the research project [Automated Surveillance and Detection of Human Threats based on Deep Learning Model: Saudi Arabia Case Study; Prince Sultan University; Riyadh Saudi Arabia [SEED-CCIS-2020{17}]". The authors would also like to acknowledge Prince Sultan University's support for paying the Article Processing Charges (APC) of this publication.

REFERENCES

[1] R. K. Tripathi, A. S. Jalal, and S. C. Agrawal, "Suspicious human activity recognition: a review," *Artificial Intelligence Review,* vol. 50, pp. 283-339, 2018.

[2] J. Gao, Y. Yang, P. Lin, and D. S. Park, "Computer vision in healthcare applications," *Journal of healthcare engineering,* vol. 2018, 2018.

[3] A. Tapus, A. Bandera, R. Vazquez-Martin, and L. V. Calderita, "Perceiving the person and their interactions with the others for social robotics–a review," *Pattern Recognition Letters,* vol. 118, pp. 3-13, 2019.

[4] A. Ilidrissi and J. K. Tan, "A deep unified framework for suspicious action recognition," *Artificial Life and Robotics,* vol. 24, pp. 219-224, 2019.

[5] E. Kim, S. Helal, and D. Cook, "Human activity recognition and pattern discovery," *IEEE Pervasive Computing/IEEE Computer Society [and] IEEE Communications Society,* vol. 9, p. 48, 2010.

[6] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "HMDB: a large video database for human motion recognition," in *2011 International conference on computer vision*, 2011, pp. 2556-2563.

[7] C. Kyrkou and T. Theocharides, "Deep-Learning-Based Aerial Image Classification for Emergency Response Applications Using Unmanned Aerial Vehicles," in *CVPR Workshops*, 2019, pp. 517-525.

[8] H. Mliki, F. Bouhlel, and M. Hammami, "Human activity recognition from UAV-captured video sequences," *Pattern Recognition,* vol. 100, p. 107140, 2020.

[9] G. Vallathan, A. John, C. Thirumalai, S. Mohan, G. Srivastava, and J. C.-W. Lin, "Suspicious activity detection using deep learning in secure assisted living IoT environments," *The Journal of Supercomputing,* vol. 77, pp. 3242-3260, 2021.

[10] I. Bègue, M. Vaessen, J. Hofmeister, M. Pereira, S. Schwartz, and P. Vuilleumier, "Confidence of emotion expression recognition recruits brain regions outside the face perception network," *Social cognitive and affective neuroscience,* vol. 14, pp. 81-95, 2018.

[11] M. Alotaibi and A. Mahmood, "Improved gait recognition based on specialized deep convolutional neural network," *Computer Vision and Image Understanding,* vol. 164, pp. 103-110, 2017.

[12] S. Al-Ali, M. Milanova, H. Al-Rizzo, and V. L. Fox, "Human action recognition: contour-based and silhouette-based approaches," in *Computer Vision in Control Systems-2*, ed: Springer, 2015, pp. 11-47.

[13] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool, "Temporal segment networks: Towards good practices for deep action recognition," in *European conference on computer vision*, 2016, pp. 20-36.

[14] B. Zhang, L. Wang, Z. Wang, Y. Qiao, and H. Wang, "Real-time action recognition with enhanced motion vector CNNs," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2718-2726.

[15] P. Wang, W. Li, Z. Gao, Y. Zhang, C. Tang, and P. Ogunbona, "Scene flow to action map: A new representation for rgb-d based action recognition with convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 595-604.

[16] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," 2005.

[17] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 3551-3558.

[18] M.-y. Chen and A. Hauptmann, "Mosift: Recognizing human actions in surveillance videos," 1995.

[19] R. Mattivi and L. Shao, "Human action recognition using LBP-TOP as sparse spatio-temporal feature descriptor," in *International Conference on Computer Analysis of Images and Patterns*, 2009, pp. 740-747.

[20] B. M. Peixoto, S. Avila, Z. Dias, and A. Rocha, "Breaking down violence: A deep-learning strategy to model and classify violence in videos," in *Proceedings of the 13th International Conference on Availability, Reliability and Security*, 2018, p. 50.

[21] R. Poppe, "A survey on vision-based human action recognition," *Image and Vision Computing,* vol. 28, pp. 976-990, 2010.

[22] J. Zhang and H. Hu, "Domain learning joint with semantic adaptation for human action recognition," *Pattern Recognition,* vol. 90, pp. 196-209, 2019.

[23] N. Yala, B. Fergani, and A. Fleury, "Towards improving feature extraction and classification for activity recognition on streaming data," *Journal of Ambient Intelligence and Humanized Computing,* vol. 8, pp. 177-189, 2017.

[24] U. M. Nunes, D. R. Faria, and P. Peixoto, "A human activity recognition framework using max-min features and key poses with differential evolution random forests classifier," *Pattern Recognition Letters,* vol. 99, pp. 21-31, 2017.

[25] V. Kantorov and I. Laptev, "Efficient feature extraction, encoding and classification for action recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2593-2600.

[26] Q. Wang, D. Gong, M. Li, C. Zhao, and Y. Lei, "Sparse feature auto-combination deep network for video action recognition," in *2017 13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*, 2017, pp. 712-716.

[27] B. Zhang, Y. Yang, C. Chen, L. Yang, J. Han, and L. Shao, "Action recognition using 3D histograms of texture and a multi-class boosting classifier," *IEEE Trans. Image Process,* vol. 26, pp. 4648-4660, 2017.

[28] A. B. Sargano, X. Wang, P. Angelov, and Z. Habib, "Human action recognition using transfer learning with deep representations," in *Neural Networks (IJCNN), 2017 International Joint Conference on*, 2017, pp. 463-469.

[29] A. Jalal, M. Mahmood, and A. S. Hasan, "Multi-features descriptors for human activity tracking and recognition in Indoor-outdoor environments," in *2019 16th International Bhurban Conference on Applied Sciences and Technology (IBCAST)*, 2019, pp. 371-376.

[30] N. H. Van Nguyen, M. T. Pham, N. Dai Ung, and K. Tachibana, "Human Activity Recognition Based on Weighted Sum Method and Combination of Feature Extraction Methods," *International Journal of Intelligent Information Systems,* vol. 7, p. 9, 2018.

[31] C. Dhiman and D. K. Vishwakarma, "A review of state-of-the-art techniques for abnormal human activity recognition," *Engineering Applications of Artificial Intelligence,* vol. 77, pp. 21-45, 2019.

[32] A. B. Mabrouk and E. Zagrouba, "Abnormal behavior recognition for intelligent video surveillance systems: A review," *Expert Systems with Applications,* vol. 91, pp. 480-491, 2018.

[33] M. M. Hassan, S. Huda, M. Z. Uddin, A. Almogren, and M. Alrubaian, "Human activity recognition from body sensor data using deep learning," *Journal of medical systems,* vol. 42, p. 99, 2018.

[34] M. F. Aslan, A. Durdu, and K. Sabanci, "Human action recognition with bag of visual words using different machine learning methods and hyperparameter optimization," *Neural Computing and Applications,* vol. 32, pp. 8585-8597, 2020.

[35] C. Dai, X. Liu, and J. Lai, "Human action recognition using two-stream attention based LSTM networks," *Applied Soft Computing,* vol. 86, p. 105820, 2020.

[36] T. Karácsony, A. M. Loesch-Biffar, C. Vollmar, S. Noachtar, and J. P. S. Cunha, "A Deep Learning Architecture for Epileptic Seizure Classification Based on Object and Action Recognition," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 4117-4121.

[37] H. Zhu, S. Samtani, H. Chen, and J. F. Nunamaker Jr, "Human identification for activities of daily living: A deep transfer learning approach," *Journal of Management Information Systems,* vol. 37, pp. 457-483, 2020.

[38] M. Loey, G. Manogaran, M. H. N. Taha, and N. E. M. Khalifa, "A hybrid deep transfer learning model with machine learning methods for face mask detection in the era of the COVID-19 pandemic," *Measurement,* vol. 167, p. 108288, 2021.

[39] A. B. Sargano, X. Wang, P. Angelov, and Z. Habib, "Human action recognition using transfer learning with deep representations," in *2017 International joint conference on neural networks (IJCNN)*, 2017, pp. 463-469.

[40] X. Yu, Z. Zhang, L. Wu, W. Pang, H. Chen, Z. Yu, and B. Li, "Deep ensemble learning for human action recognition in still images," *Complexity,* vol. 2020, 2020.

[41] A. Diriba, "Keyframe-Based Saliency Detection For Human Action Recognition Using Deep Learning," ASTU, 2020.

[42] A. Sanchez-Caballero, S. de López-Diz, D. Fuentes-Jimenez, C. Losada-Gutiérrez, M. Marrón-Romera, D. Casillas-Perez, and M. I. Sarker, "3DFCNN: Real-Time Action Recognition using 3D Deep Neural Networks with Raw Depth Information," *arXiv preprint arXiv:2006.07743,* 2020.

[43] A. S. Ben-Musa, S. K. Singh, and P. Agrawal, "Suspicious Human Activity Recognition for Video Surveillance System," ed: ICCICCT, 2014.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3091081, IEEE Access

[44] S. Nigam, R. Singh, and A. Misra, "Towards intelligent human behavior detection for video surveillance," in *Censorship, Surveillance, and Privacy: Concepts, Methodologies, Tools, and Applications*, ed: IGI Global, 2019, pp. 884-917.

[45] P. Schwarzfischer, D. Gruszfeld, A. Stolarczyk, N. Ferre, J. Escribano, D. Rousseaux, M. Moretti, B. Mariani, E. Verduci, and B. Koletzko, "Physical Activity and Sedentary Behavior From 6 to 11 Years," *Pediatrics,* vol. 143, p. e20180994, 2019.

[46] E. Finne, M. Glausch, A.-K. Exner, O. Sauzet, F. Stoelzel, and N. Seidel, "Behavior change techniques for increasing physical activity in cancer survivors: a systematic review and meta-analysis of randomized controlled trials," *Cancer management and research,* vol. 10, p. 5125, 2018.

[47] P. D. Chang, T. T. Wong, and M. J. Rasiej, "Deep Learning for Detection of Complete Anterior Cruciate Ligament Tear," *Journal of Digital Imaging,* pp. 1-7, 2019.

[48] A. L. Hayes, "Autism Spectrum Disorder: Patient Care Strategies for Medical Imaging," *Radiologic Technology,* vol. 90, pp. 31-47, 2018.

[49] D. A. Aristizabal, S. Denman, K. Nguyen, S. Sridharan, S. Dionisio, and C. Fookes, "Understanding Patients' Behavior: Vision-based Analysis of Seizure Disorders," *IEEE Journal of Biomedical and Health Informatics,* 2019.

[50] B. Noah, M. S. Keller, S. Mosadeghi, L. Stein, S. Johl, S. Delshad, V. C. Tashjian, D. Lew, J. T. Kwan, and A. Jusufagic, "Impact of remote patient monitoring on clinical outcomes: an updated meta-analysis of randomized controlled trials," *NPJ Digital Medicine,* vol. 1, p. 2, 2018.

[51] M. Alizadeh, S. Peters, S. Etalle, and N. Zannone, "Behavior analysis in the medical sector: theory and practice," in *Proceedings of the 33rd Annual ACM Symposium on Applied Computing*, 2018, pp. 1637-1646.

[52] P. H. Nguyen, C. Turkay, G. Andrienko, N. Andrienko, O. Thonnard, and J. Zouaoui, "Understanding user behaviour through action sequences: from the usual to the unusual," *IEEE transactions on visualization and computer graphics,* 2018.

[53] W. Sultani, C. Chen, and M. Shah, "Real-world anomaly detection in surveillance videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6479-6488.

[54] E. Hoque, R. F. Dickerson, S. M. Preum, M. Hanson, A. Barth, and J. A. Stankovic, "Holmes: A comprehensive anomaly detection system for daily in-home activities," in *2015 International Conference on Distributed Computing in Sensor Systems*, 2015, pp. 40-51.

[55] C. Lu, J. Shi, and J. Jia, "Abnormal event detection at 150 fps in matlab," in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 2720-2727.

[56] Y. Kim and T. Moon, "Human detection and activity classification based on micro-Doppler signatures using deep convolutional neural networks," *IEEE geoscience and remote sensing letters,* vol. 13, pp. 8-12, 2015.

[57] M. Sabokrou, M. Fayyaz, M. Fathy, Z. Moayed, and R. Klette, "Deep-anomaly: Fully convolutional neural network for fast anomaly detection in crowded scenes," *Computer Vision and Image Understanding,* vol. 172, pp. 88-97, 2018.

[58] C. Lu, J. Shi, W. Wang, and J. Jia, "Fast Abnormal Event Detection," *International Journal of Computer Vision,* pp. 1-19, 2018.

[59] P. L. Venetianer, A. J. Lipton, A. J. Chosak, M. F. Frazier, N. Haering, G. W. Myers, W. Yin, Z. Zhang, and R. Cutting, "Video surveillance system employing video primitives," ed: Google Patents, 2018.

[60] M. Ketcham, "CCTV Face Detection Criminals and Tracking System Using Data Analysis Algorithm," in *Advances in Intelligent Informatics, Smart Technology and Natural Language Processing: Selected Revised Papers from the Joint International Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP 2017)*, 2019, p. 105.

[61] R. Agarwal, R. Jain, R. Regunathan, and C. P. Kumar, "Automatic Attendance System Using Face Recognition Technique," in *Proceedings of the 2nd International Conference on Data Engineering and Communication Technology*, 2019, pp. 525-533.

[62] S. J. Elias, S. M. Hatim, N. A. Hassan, L. M. A. Latif, R. B. Ahmad, M. Y. Darus, and A. Z. Shahuddin, "Face Recognition Attendance System Using Local Binary Pattern (LBP)," *Bulletin of Electrical Engineering and Informatics,* vol. 8, 2019.

[63] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems,* vol. 25, pp. 1097-1105, 2012.

[64] S. Balocco, M. González, R. Ñanculef, P. Radeva, and G. Thomas, "Calcified plaque detection in IVUS sequences: Preliminary results using convolutional nets," in *International Workshop on Artificial Intelligence and Pattern Recognition*, 2018, pp. 34-42.

[65] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*, 2015, pp. 448-456.

[66] Y. Liu, X. Wang, L. Wang, and D. Liu, "A modified leaky ReLU scheme (MLRS) for topology optimization with multiple materials," *Applied Mathematics and Computation,* vol. 352, pp. 188-204, 2019.

[67] J. Bouvrie, "Notes on convolutional neural networks," *Neural Nets, MIT CBCL Tech Report,* pp. 47-60, 2006.

[68] Y. Li, Z. Hao, and H. Lei, "Survey of convolutional neural network," *Journal of Computer Applications,* vol. 36, pp. 2508-2515, 2016.

[69] J. Wu, "Introduction to convolutional neural networks," *National Key Lab for Novel Software Technology. Nanjing University. China,* vol. 5, p. 23, 2017.

[70] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images (Technical Report)," *University of Toronto,* 2009.

[71] M. Dash and H. Liu, "Feature selection for clustering," in *Pacific-Asia Conference on knowledge discovery and data mining*, 2000, pp. 110-121.

[72] A. Rashno, B. Nazari, S. Sadri, and M. Saraee, "Effective pixel classification of mars images based on ant colony optimization feature selection and extreme learning machine," *Neurocomputing,* vol. 226, pp. 66-79, 2017.

[73] Z. Liu, M. J. Zuo, X. Zhao, and H. Xu, "An Analytical Approach to Fast Parameter Selection of Gaussian RBF Kernel for Support Vector Machine," *J. Inf. Sci. Eng.,* vol. 31, pp. 691-710, 2015.

[74] P. Virdi, Y. Narayan, P. Kumari, and L. Mathew, "Discrete wavelet packet based elbow movement classification using fine Gaussian SVM," in *2016 IEEE 1st international conference on power electronics, intelligent control and energy systems (ICPEICES)*, 2016, pp. 1-5.

[75] I. Dagher, "Quadratic kernel-free non-linear support vector machine," *Journal of Global Optimization,* vol. 41, pp. 15-30, 2008.

[76] Y.-W. Chang and C.-J. Lin, "Feature ranking using linear SVM," in *Causation and prediction challenge*, 2008, pp. 53-64.

[77] W. S. Noble, "What is a support vector machine?," *Nature biotechnology,* vol. 24, pp. 1565-1567, 2006.

[78] S. Rüping, "SVM kernels for time series analysis," Technical report2001.

[79] N.-E. Ayat, M. Cheriet, and C. Y. Suen, "Automatic model selection for the optimization of SVM kernels," *Pattern Recognition,* vol. 38, pp. 1733-1745, 2005.

[80] B. Haasdonk, "Feature space interpretation of SVMs with indefinite kernels," *IEEE Transactions on pattern analysis and machine intelligence,* vol. 27, pp. 482-492, 2005.

[81] L. E. Peterson, "K-nearest neighbor," *Scholarpedia,* vol. 4, p. 1883, 2009.

[82] A. P. Singh, "Analysis of Variants of KNN Algorithm based on Preprocessing Techniques," in *2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*, 2018, pp. 186-191.

[83] A. Lamba and D. Kumar, "Survey on KNN and its variants," *International Journal of Advanced Research in Computer and Communication Engineering,* vol. 5, pp. 430-435, 2016.

[84] L. Jiang, H. Zhang, and J. Su, "Learning k-nearest neighbor naive bayes for ranking," in *International conference on advanced data mining and applications*, 2005, pp. 175-185.

[85] Y. Xu, Q. Zhu, Z. Fan, M. Qiu, Y. Chen, and H. Liu, "Coarse to fine K nearest neighbor classifier," *Pattern recognition letters,* vol. 34, pp. 980-986, 2013.

[86] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," in *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, 2005, pp. 1395-1402.

[87] D. K. Vishwakarma, "A two-fold transformation model for human action recognition using decisive pose," *Cognitive Systems Research,* vol. 61, pp. 1-13, 2020.

[88] M. A. Khan, Y.-D. Zhang, S. A. Khan, M. Attique, A. Rehman, and S. Seo, "A resource conscious human action recognition framework using 26-layered deep convolutional neural network," *Multimedia Tools and Applications,* pp. 1-23, 2020.

[89] F. Afza, M. A. Khan, M. Sharif, S. Kadry, G. Manogaran, T. Saba, I. Ashraf, and R. Damaševičius, "A framework of human action recognition using length control features fusion and weighted entropy-variances based feature selection," *Image and Vision Computing,* vol. 106, p. 104090, 2021.

[90] A. Nadeem, A. Jalal, and K. Kim, "Human actions tracking and recognition based on body parts detection via Artificial neural network," in *2020 3rd International Conference on Advancements in Computational Sciences (ICACS)*, 2020, pp. 1-6.

[91] A. Abdelbaky and S. Aly, "Human action recognition using three orthogonal planes with unsupervised deep convolutional neural network," *Multimedia Tools and Applications,* pp. 1-25, 2021.