Indonesian Journal of Data and Science (IJODAS) ISSN: 2715-9930
Vol 4, No 1, March 2023, pp. 17-24

Detecting harmful activity in Hajj plagiarism using Deep Learning

Musa Dima Genemo [a,1]

[a] *Gumushane University, Bağlarbaşı, 29100 Gümüşhane Merkez/Gümüşhane, Turki*
musa.ju2002@gmail.com;

ARTICLE INFORMATION

Diterima : 17 – 01 – 2023
Direvisi : 19 – 02 – 2023
Diterbitkan : 31 – 03 – 2023

Kata Kunci:
Artificial Intelligence,
Computer Vision,
Classification,
Real-Time Object Recognition

ABSTRACT

CCTV surveillance is the most extensively used intelligent latest innovation. The use of surveillance cameras has risen dramatically because of the convenience of monitoring from anywhere and the reduction of crime rates in public areas. In this paper, we introduce the idea of bad vibe activity detection from live videos to enhance the security and safety of pilgrims. The proposed bad vibes activity recognition model is intended to be addressed in the most efficient manner possible using cutting-edge technologies such as TensorFlow and Keras. TensorFlow was chosen because the project could be deployed to a mobile environment in the future with the possibility of extension of other areas such as airport security, bus stain, and public areas that may deserve special attention for security checks. We choose MediaPipe Holistic for employee bad vibe recognition in the model.

I. Introduction

The use of surveillance cameras has risen dramatically because of the convenience of monitoring from anywhere and the reduction of crime rates in public areas. Hajj is one of the Five Pillars of the Islamic religion where the pilgrimage to the holy city of Mecca in the kingdom of Saudi Arabia, which takes place in the last month of the year(Hijri calendar) and which all Muslims are obligated to make at least once throughout their lifetime if they can afford it [1]. Before COVID_19 emerged, 2.5 million people would travel every year to Saudi Arabia for Hajj. Due to this, the security of pilgrims needs special attention. New cutting-edge technology is required to ensure the safety of the people and the city where the hajj imitation takes place, as well as the detection of forbidden activities and the carrying of prohibited things such as guns, flame, sharp metals, and the like. Human activity recognition (HAR) is the ability to use sensors to analyze human body indicators or motion and identify human actions or events [2]. HAR is regarded as a significant component in various scientific research settings, such as health[3], Human-robot interaction [4], and security [5].

Such technologies are in high demand during the hajj festival to safeguard pilgrims' safety. Many people have become victims of the Hajj scam in recent years, losing money, cellphones, and other valuables. Nowadays Terrorist acts pose the greatest danger to public safety [6]. Prohibited things, such as carrying a gun, hurling a bomb, deceiving people, and threatening a suicide bombing, should be checked instantly.

Figure 1. Security control of pilgrims

As a result, these challenges demand models that generate a warning or alarm. If accurate forecasts are provided in a timely manner, human lives can be saved by employing this newly introduced model. Interleaved actions, such as throwing a stone at three walls (Ramy Al Jamarat), which is also known as stoning the devil (sheytan), and running between mina and muzdalifah are a pillar of the Hajj pilgrims. The stoning of the devil may cause prediction ambiguity, by throwing stones at people or away from the road and running between mina and muzdalifah may cause prediction ambiguity with a sudden run. Recurrent Neural Network(RNN) is used to over come activity overlapping difficulties.

Figure 2. Stoning the Devil

Despite this, utilizing smart CCTV surveillance reduces labor expenses while also increasing the security and safety of pilgrims. This study proposes a deep feature extraction mechanism for forbidden motion and activity identification to address the difficulties. We proposed a new model named **L4-BRANCHED-ACTION NET**. By using this new model, we extract features from the video frame and bels the activity to their respective class like the need for special attention, or safe move. 64 layers of CNN- deep architecture are used for feature extraction. To optimize the deep features that have been obtained, an ACO feature selection technique is applied. By running convolution layers over pre-trained public data like the CIFRA-100.

II. Material and Method

The proposed model will be presented in its entirety in this section. Furthermore, this section includes details of the proposed 64-layer Classification algorithm. We used the CIFAR-100 dataset to train the proposed model, as well as feature extraction from the action recognition dataset using the proposed CNN architecture, feature selection using Ant Colony Optimization (ACO), and prediction using a variety of algorithms. For autonomous feature extraction from video frames and classifications events in the frame, a novel proposed 64-layer CNN architecture is used. The recommended L4-BranchedActionNet's physical architecture is shown in Figure 3.

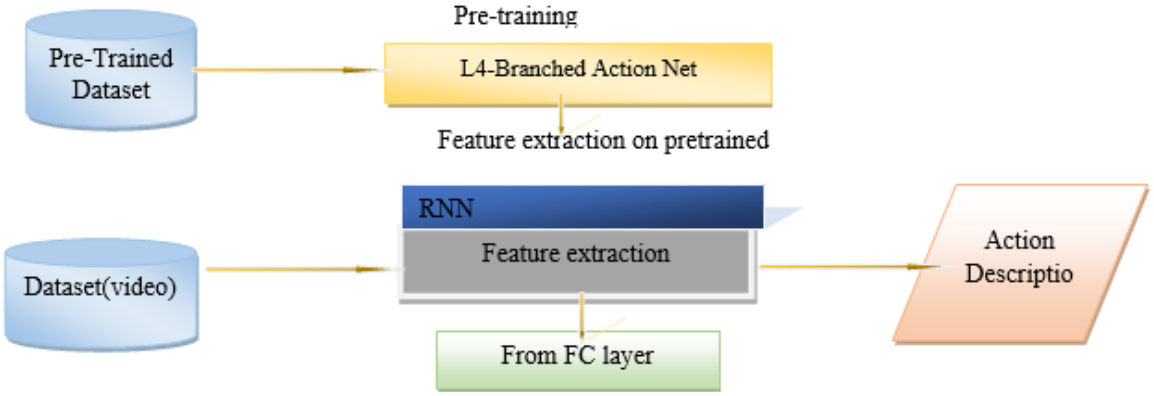


Figure 3. Structure of proposed model

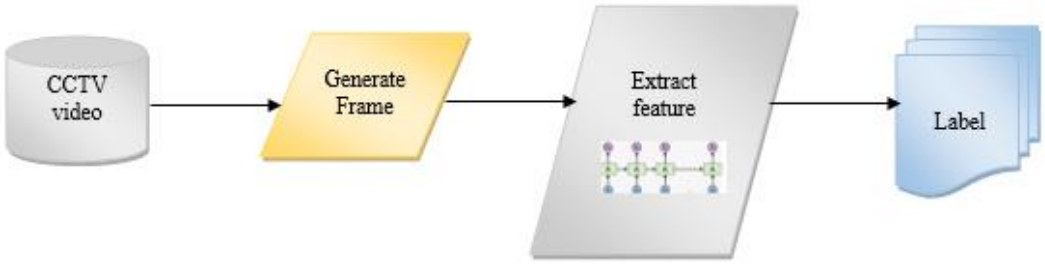


Figure 4. Video frame generation

We used AlexNet [7,8] as the foundation for the proposed architecture pipeline. The configuration of the layers, as well as all their details, are shown in Table 1. To determine numerous activities in the frame, a Recurrent Neural Network model was used. As a result, several movements are easily spotted.

Table 1. Layers configurations of l4-branched-action net

Layer	Layer Name	Feature Maps	Filter depth	Stride	Padding	Polling 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]	

Layer	Layer Name	Feature Maps	Filter depth	Stride	Padding	Polling window size/other values
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	FC_21	1 × 1 × 100	[1,1]	same		
64	Video description					

The data was collected using a script generated utilizing OpenCV and MediaPipe Holistic, as shown in Figure 4. 30 frames of data are recorded for each word caught.

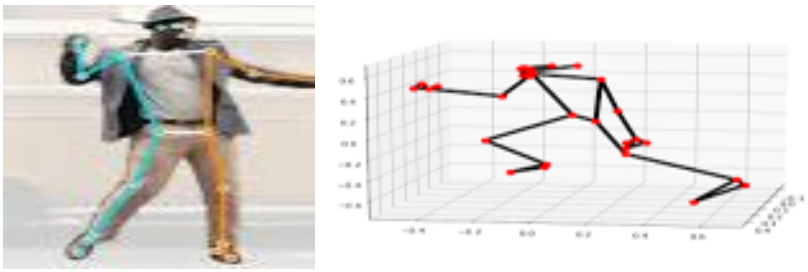


Figure 5. Key using OnePose using MediaPipe Holistic point extraction[9]

NumPy array is used instead of pictures to hold video frames. We passed three major steps to train the model. The following are the details of the new model's operations steps. The first step is conv layer; (1) In the Conv layer the input x $_{i-1}$ filter is computed using equation 1.

$$Xp^{,j} = fi\left(\sum_p fi, \mathrm{p}^{,p} \quad * \quad xp, i-1 + fp\hat{}, i\right) \tag{1}$$

where p_j input channels and $p\hat{}_j$ represent the number of output channels. j represents several layers in the mode, *fi* filter. Equation 2 is used to calculate max pool in pooling layer.

$$Xp, j, u, v = \quad maxl = 1..s, m = 1..tXp, j-1, (u+1)(v+m) \tag{2}$$

where u, v represents matrix index of frame $X_{p,\,j-1}$and l, m matrix index of the pooling window. It calculates the mean and variance in fragments. The mean is derived, and the features are separated using the standard deviation as follow.

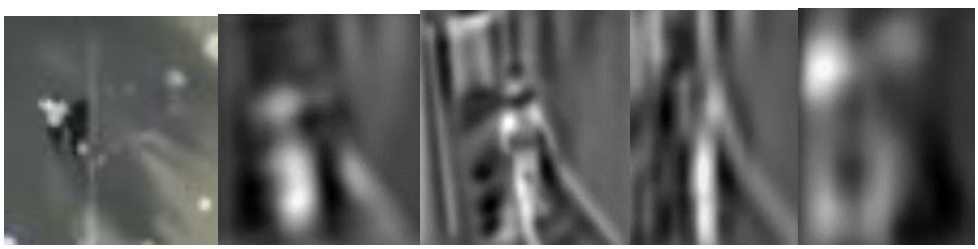
$$\mu = \frac{1}{\mathrm{w}\sum_{\mathrm{z}}^{\mathrm{w}} \mathrm{Xz}} \tag{3}$$

where *w* is the number of feature maps in a batch. We used both ReLU and Leaky_ ReLU in proposed model. All numbers less than 0 are transformed to 0 by the standard ReLU, which is stated as [10-14]:

$$Xu, v = \max(0, Xu, v) \tag{4}$$

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 [15,16-19].

The second step is feature extraction;(2) For feature extraction from video frame, the appropriate frame is retrieved. The proposed approach is intended to feature extraction from the deep-trained CNN pipeline. We trained the new model on public dataset t such as CIFAR100 [20-25] which has contained images of 1000 classes. 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 frame from the FC_18 layer. The prepared dataset contains a total of 13250 video frames. This makes the feature set dimension of all datasets 13250×4096. Figure 5 illustrates the visualizations of the strongest feature maps at various convolution layers on L4-Branched-ActionNet.



(a) (b) (c) (d) (e)

Figure 6. Image visualizations of strongest feature maps at various convolution layers (a) Conv_1, (b) Conv_2, (c) Conv_5, (d) G_Conv_8, (e) Conv_10.

And the third step is (3) after interpreting the received result the extracted features are coded by applying entropy coded ACO optimization operation [71] using equation (5).

$$e(X1 \ldots . . Xn) = \sum f1 \ldots \sum fn(p(f1 \ldots fn)LOGp(f1..fn) \quad (5)$$

Where (x1-xn) represent the feature. We used ACO for feature optimization is based on the likelihood at a given point at a certain time.

The last step is classification, in which ACO-based chosen features are at the end passed to the predictor for categorization. Several SVM and KNN versions are used to assess model performance. Cub-SVM emerges as the most effective as shown in Table 2.

Table 2. Performance of the model on

Classifier	Correctness	Sensitivity	Specificity	Precision	Measure	Percent
LSVM	74.63	83.38	72.62	39.94	52.52	77.74
QSVM	93.32	89.11	91.53	61.79	76.01	86.14
FGSVM	52.78	57.29	51.78	25.02	32.80	54.39
MGSVM	91	90,52	92.35	62.56	76.58	86.28
CGCVM	83.17	68.47	64.45	31.80	41.75	66.33
CSVM	92.99	96.33	95.59	76.61	88.08	92.99

For testing, we employed random selection using sklearn's train test function. Following that, Keras' Callback functions were used to improve the training's efficiency. The accuracy of the test data is evaluation.

We also used the public dataset ON WEIZMANN to compare our results to the current state of the art. The outcome is shown in Table 3.

Table 3. Performance evaluation On Weizmann dataset

Method reference	Year	Accuracy
DWT+KNN [21]	2020	0.93
CNN+ELM [22]	2020	0.94
Gabor-Ridgelet Transform [23]	2020	0.93
LCF + MSVM [22]	2021	0.95
ANN [24]	2020	0.80
PCANet-XY-YT [25]	2021	0.91
Ours (L4-Branched-ActionNet + EntACS + Cub-SVM)	-	0.93

III. Result and Discussion

The major goal of this study is to develop a CNN architecture that can recognize harmful actions during the Hajj festival. The, the Deep L4-BranchedActionNet Deep Network proposed here is used to extract powerful features. The pretraining is carried out using a publicly available dataset, CIFAR-100. For testing, we employed random selection using sklearn's train test function. Following that, Keras' Callback functions were used to improve the training's efficiency to complete this design, many methods such as fine-tuning, adding and removing layers and neurons were used. Finally, the 64-layer architecture was proven it is the most efficient in terms of performance. Tensor flow keras, openCV, and the numpy library were used in all the experiments in this work.

Table 4. Confusion matrix of CSVM classifier

Sudden ran	0.92021	0.00	0.01	0.02
Fighting	0.00	0.91221	0.00	0.01
Throwing	0.01	0.01	0.90021	0.00
Robbing	0.00	0.00	0.01	0.91002

	Sudden ran	Fighting	Throwing	Robbing

IV. Conclusion

The detection of harmful vibes is critical for pilgrims' safety. To detect banned actions during the hajj festival, we utilized a 64-layer CNN network called L4-Branched-ActionNet. The model is evaluated on datasets that are freely available, such as the CIFAR-100 object detection dataset. The characteristics were retrieved and subsequently reduced using an entropy coded ACO. To evaluate model performance, several SVM and KNN versions are utilized. With an accuracy of 0.91221, Cub-SVM emerges as the most effective. This work will be implemented on security personnel's mobile phones for convenient monitoring from any location in future work.

References

[1] Holy Quran (3:97)

[2] 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

[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] Konstantinidis, D., Dimitropoulos, K., & Daras, P. (2018). Sıgn Language Recognıtıon Based On Hand And Body Skeletal Data. 2018- 3DTV-Conference: The True Vision - apture, Transmission and Display of 3D Video (3DTVCON).

[6] 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.

[7] 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

[8] Genemo, M. D. (2022). Suspicious activity recognition for monitoring cheating in exams. *Proceedings of the Indian National Science Academy*, 1-10.

[9] C. A. Devine and E. D. Chin, "Integrity in nursing students: A concept analysis," *Nurse education today*, vol. 60, pp. 133-138, 2018.

[10] H. M. Abdulghani, S. Haque, Y. A. Almusalam, S. L. Alanezi, Y. A. Alsulaiman, M. Irshad, *et al.*, "Self-reported cheating among medical students: An alarming finding in a cross-sectional study from Saudi Arabia," *PloS one*, vol. 13, p. e0194963, 2018.

[11] M. A. Lewis and C. Neighbors, "An examination of college student activities and attentiveness during a web-delivered personalized normative feedback intervention," *Psychology of Addictive Behaviors*, vol. 29, p. 162, 2015.

[12] RWTH-PHOENIX-2014-T veri seti, https://wwwi6.informatik.rwth-aachen.de/~koller/RWTH-PHOENIX-
2014-T/

[13] 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

[14] 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

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

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

[17] A. Divakaran, Q. Yu, A. Tamrakar, H. S. Sawhney, J. Zhu, O. Javed, *et al.*, "Real-time object detection, tracking and occlusion reasoning," ed: Google Patents, 2018.

[18] A. Booranawong, N. Jindapetch, and H. Saito, "A system for detection and tracking of human movements using RSSI signals," *IEEE Sensors Journal*, vol. 18, pp. 2531-2544, 2018.

[19] 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.

[20] Krizhevsky and G. Hinton, "*Learning multiple layers of features from tiny* images (Technical Report)," University of Toronto, 2009
[21] D. K. Vishwakarma, "A two-fold transformation model for human action recognition using decisive pose," *Cognitive Systems Research*, vol. 61, pp. 1-13, 2020.
[22] 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.
[23] 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.
[24] 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.
[25] 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.