**International Journal of Electrical and Computer Engineering (IJECE)**
Vol. 11, No. 2, April 2021, pp. 1656~1665
ISSN: 2088-8708, DOI: 10.11591/ijece.v11i2.pp1656-1665 

# Searching surveillance video contents using convolutional neural network

**Duaa Mohammad, Inad Aljarrah, Moath Jarrah**
Department of Computer Engineering, Jordan University of Science and Technology (JUST), Jordan

**Article Info**

***Article history:***

Received Feb 24, 2020
Revised Aug 4, 2020
Accepted Nov 4, 2020

***Keywords:***

Convolutional neural networks
Key frames
Sobel detector
VGG-16
Video content analysis

**ABSTRACT**

Manual video inspection, searching, and analyzing is exhausting and inefficient. This paper presents an intelligent system to search surveillance video contents using deep learning. The proposed system reduced the amount of work that is needed to perform video searching and improved the speed and accuracy. A pre-trained VGG-16 CNNs model is used for dataset training. In addition, key frames of videos were extracted in order to save space, reduce the amount of work, and reduce the execution time. The extracted key frames were processed using the sobel operator edge detector and the max-pooling in order to eliminate redundancy. This increases compaction and avoids similarities between extracted frames. A text file, that contains key frame index, time of occurrence, and the classification of the VGG-16 model is produced. The text file enables humans to easily search for objects of interest. VIRAT and IVY LAB datasets were used in the experiments. In addition, 128 different classes were identified in the datasets. The classes represent important objects for surveillance systems. However, users can identify other classes and utilize the proposed methodology. Experiments and evaluation showed that the proposed system outperformed existing methods in an order of magnitude. The system achieved the best results in speed while providing a high accuracy in classification.



***Corresponding Author:***

Moath Jarrah
Department of Computer Engineering
Jordan University of Science and Technology
Computer and Information College, P.O. Box 3030, Irbid, 22110, Jordan
Email: mjarrah@just.edu.jo

## 1. INTRODUCTION

Video content analysis (VCA) is the method of analyzing video streams to detect and determine temporal and spatial events to find what a video represents and the type of information it has [1]. It is used by different applications that require high security to detect intruders and abnormal events especially fraud actions [2]. Airports, hotels, banks, and other public places need VCA to ensure a secure environment for clients and staff. Market owners can improve their productivity by understanding their customer reactions, needs, and desires. Moreover, VCA is of high importance in subway stations to detect dangerous situations and secure the areas [3]. Transportation systems rely on VCA to ensure passengers security, vehicle control, and better tracking methods [4]. Furthermore, video analysis is used to detect underwater objects [5]. Hence, video content searching systems must be able to search the video in a fast way. However, the amount of data that is reported by surveillance cameras is huge, which puts a challenge on the search process. It is important to organize and search the contents of videos in a way where users can find objects of interest within a short time.

Deep convolutional neural networks (CNNs) are used in different domains such as image classification [6, 7], object detection [8, 9], and natural language processing [10, 11]. The networks are computationally intensive tasks. However, they provide high accuracy in object detection. CNNs contain three types of layers which are: convolutional layers, pooling layers, and fully connected layers. In some applications, an activation function (i.e. rectified linear unit (ReLU)) layer follows the convolutional layers. There have been successful applications of CNNs for image classification such as AlexNet model which achieved 15.3% top-5 test error rate in ILSVRC-2012 [6]. In addition, ZF Net [12] and GoogLeNet model [13] achieved excellent performance. VGG-16 model and ResNet model were designed by He *et al*. for object detection [14, 15]. CNNs are also used in face recognition such as the work by Li *et al*. [16]. Li *et al*. proposed a multi-resolution CNN cascade for fast face detection. Furthermore, Sun *et al*. proposed two deep neural network architectures, DeepID3, and achieved 99.53% accuracy in LFW face verification and 96.0% LFW rank-1 face identification [17]. In addition, CNNs have been used for medical image analysis [18, 19]. In 1994, the CNNs were used to detect the micro-calcifications in digital mammography [20].

Much of the past and on-going research aims to analyze video contents using different methods. Lao *et al*. proposed a system for semantical analysis of human behaviors in a monocular surveillance video captured by a consumer camera [21]. The authors incorporated a trajectory estimation method besides human-body modeling to comprehend the semantic analysis of human activities and events in video sequences. Zhao and Cai employed a short-time memory model to segment a given video and to specify the scene importance for key frames extraction [22]. Bertini *et al.* presented a framework for event and object extraction of soccer videos [23]. The authors applied semantic transcoding to the frames that contain events and human faces. Four classes of events were detected in their framework. Kolekar introduced a probabilistic approach for video analysis and indexing, based on bayesian belief network (BBN) [24]. They used a hierarchal classification framework to extract features from videos and then the BBN assigns the semantic label for each event in video clips. Furthermore, Chen and Zhang proposed a video content analysis system using autoregressive (AR) modeling to model the feature sequence of frames over time [25]. Sun *et al.* introduced a video analysis method that depends on color distributions between frames [26]. The distributions are used to search the video frames.

Furthermore, Sharif *et al*. proposed a detection system using entropy measure to partition a video into small spatial-temporal patches [27]. However, their system measures the background features only and does not assess the behaviors of individuals and moving objects. Cernekov *et al*. extracted key frames using mutual information and joint entropy for ease of search of video contents [28]. Zeng *et al*. applied a block-based markov random field (MRF) model to segment the moving objects obtained from video frames to analyze video contents; and used backtracking to select the key frames [29]. Zhou *et al*. proposed a non-uniform sampling method as well as a simple uniform sampler (Uni) for summarizing long video content [30]. Afterward, the proposed sampling method extracts important features and produce a short video where users can search it faster. Their system takes a second to retime each video and ten seconds to render each frame. On the other hand, Bai *et al*. introduced a video semantic content analysis framework that depends on domain ontology [31]. The authors used low-level algorithms to extract both high level and low-level features in the videos. The video event detection is performed manually. Foggia *et al*. introduced a fire detection system analysis for surveillance videos [32]. Their proposed system relies on color, shape variation, and motion analysis to detect a fire. They used YUV color space and the scale-invariant feature transform (SIFT) descriptors for blobs movements' detection. Then the multi-expert system (MES) produces the prediction. The system achieved a considerable rate of false positives.

This paper proposes a system for searching surveillance video contents (SSVC). SSVC system uses CNN for object recognition and classifications. Specifically, it uses the VGGNet model which was developed by Oxford's visual geometry group (VGG) [14]. The model scored the first place in image localization and the second place in image classification [33]. There are different configurations of the VGGNet such as VGG-16 and VGG-19. VGG-16 has 13 convolutional layers and 3 fully connected layers. SSVC system uses VGG-16. It generates a text file that contains different classes of the detected objects and the time of appearance of each object in the video, as well as the frame index. To improve the performance, SSVC system processes only a specific number of frames that hold most of the information (key frames). Furthermore, a matching process is performed to eliminate redundant key frames. As a result, VGG-16 eliminates many of the extracted frames, which improves the speed of the system. VIRAT and IVY LAB datasets are used in the experiments. Results show that SSVC system outperforms previously proposed methods in terms of speed. The rest of this paper is organized as follows. Section 2 presents the methodology of the proposed system and the different techniques that are employed to enhance video analysis. Additionally, the experiments are presented to evaluate each technique. Section 3 concludes the paper.

## 2. METHODOLOGY AND EXPERIMENTAL RESULTS

MATLAB is used for the implementation and training of the pre-trained CNN model. In addition, FFmpeg is used for key frames extraction like the work in [34]. FFmpeg supports different video and audio formats. For CNN re-training, NVIDIA Tesla K80 is used which has 2496 CUDA cores.

### 2.1. Methodology

In this work, the VGG-16 is re-trained using a dataset that contains 48000 images. The VGG-16 model was not trained from scratch for time saving purposes. In addition, training the VGG-16 model on a relatively small dataset causes network overfitting. Hence, a transfer learning is used for a VGG-16 model that was trained on a very large dataset (ImageNet, which contains 1.2 million images with 1000 categories). The last fully-connected layer of the trained model was removed and tailored for the 128 categories of our system. Also, appropriate training parameters values were selected. The values were chosen manually with trial and error while monitoring the performance and the accuracy of the VGG-16 model.

The training dataset is chosen according to the probability of appearances in surveillance videos. Table 1 lists the categories that were selected in our VCA system. These objects are of interest in surveillance videos. Nevertheless, users can add or remove objects as they prefer and still follow the methodology of the proposed system. The system considers 128 different objects. Some of the selected objects are considered dangerous and may harm humans such as fire, smoke, guns, revolvers, nails, and drill tools. In addition, animals like scorpions, snakes, and spiders are considered. The dataset also has normal objects that are mainly used by humans like cars, motorbike, bags, cameras, mobile phones, computers, ATMs, iron, and vacuum cleaner. Some examples of objects that were excluded are sea objects, clothes, and musical instruments. After the dataset has been prepared, each image is labeled and used in the training of the pre-trained VGG-16 model.

Table 1. The selected object categories in the dataset

| The categories that were selected in our VCA system | | | | | | | |
|---|---|---|---|---|---|---|---|
| Packet | Airplane | Toaster | Alligator | Ambulance | Apiary, bee house | Apple | Ashcan, garbage can |
| Printer | Ball | Vending machine | Banana | Bannister, banister | Barrel, cask | Bassinet | Bicycle |
| School bus | Cab, taxi, taxicab | Pop bottle | Calculator | Camera | Can opener | Candle, taper, | Car |
| Snake | Cassette player | Moving van | Cat | Cellular phone | Chain | Chair | Child, kid |
| Tiger | Computer mouse | Gasmask | Confectionery, candy store | Corkscrew, bottle screw | Crane | Crash helmet | Crutch |
| Tram, trolley car | Drill | People | Eagle | Electric fan | Espresso maker | Fence, fencing | Fire |
| Smoke | Flowerpot | Racing car | Forklift | Fountain | Frying pan | Garbage truck | Garden cart |
| Orange | Hatchet | Screw | Bag | Grocery store | Hair dryer | Hammer | Hamster |
| Fire truck | Book | Spider | Bookcase | Hen | Horse | iPod | Iron |
| Pen | Carton | Tractor | Cash machine, ATM | Knife | Ladle | Lamp | Laptop |
| Rabbit | Combination lock | Water bottle | Computer | Lighter | Limousine | Macaw | Mailbox, letterbox |
| Scorpion | Dial telephone | Saw | Dog | Microwave oven | Milk can | Monitor | Motorbike |
| Sofa bed | Fish | Microphone, mike | Flower | Mug | Mushroom | Nail | Necklace |
| Strawberry | Lemon | Assault rifle, | Cart | Jeep, land rover | Perfume, essence | Pipe, tube | Plate |
| Table | Vacuum cleaner | Automatic washer | Bird | Clock | Refrigerator, icebox | Remote control | Revolver |
| Traffic light | Train, railroad train | Street sign | Turtle | Syringe | Screwdriver | Sheep | Shovel |

Appropriate training parameters are assigned and used during the training of the VGG-16 model. Table 2 shows the assigned values for the most important training parameters of the VGG-16 model. The assigned values were tested and verified to provide the best results through trial and error. These parameters are important to fit the training images in the memory and to avoid overfitting of the CNN network.

The script is executed 300 times Table 2. At each iteration, 160 images are chosen at random from the training set. The network layers are trained over these images and the prediction is compared against the ground truth. Figure 1 shows the training results. The network reaches an accuracy of 88.91%. Figure 2 and Figure 3 shows different testing examples of the trained network where the network was able to correctly identify objects.

Table 2. Training parameters

| Parameters | Specification |
|---|---|
| Number of classes | 128 different categories |
| Training images | 48,000 |
| Validation images | 14,000 |
| Minibatch size | 160 |
| Number of iterations | 300 |
| Initial learning rate | 0.0001 |
| Number of epochs | 30 |

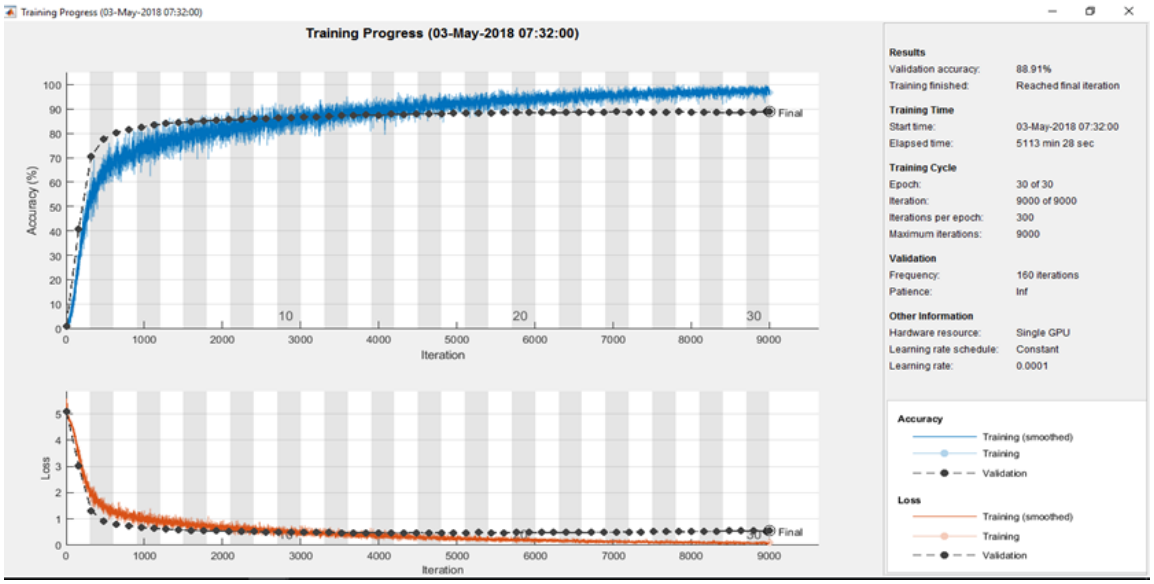



Figure 1. The accuracy of VGG-16 against the number of iterations

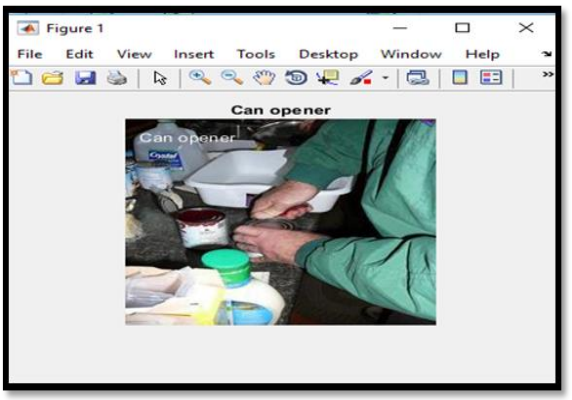



Figure 2. A can opener is detected in the frame

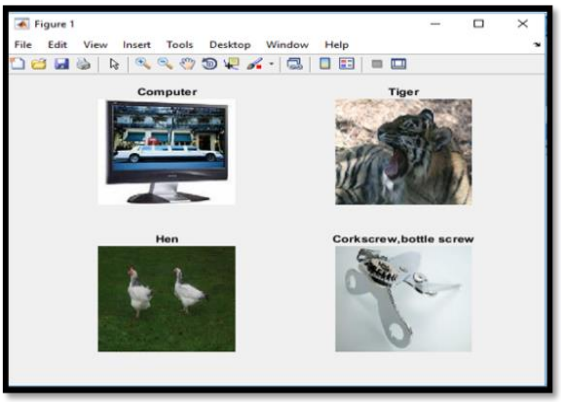



Figure 3. Examples of objects that were correctly detected by the VGG-16

### 2.2. Speeding up the VCA system

To enhance the speed of the VCA system, three techniques were used to improve the performance. The techniques are extraction key frames, sobel edge detector, and Max-pooling as follows.

#### 2.2.1. Key frames

In a video, the entire information exists in the (I) frame, which is called also the key frame. The set of key frames is similar to having a short representation of a video. To increase the speed of the proposed system, only key frames were used. FFmpeg program is used to extract the frames. The extraction is performed for different frame rates such as 25, 30, and 60 frames per second. Moreover, Different surveillance

videos for indoor and outdoor activities can have redundancy in the key frames because of small transitions. Similarities in different sequences of the key frames were found. Hence, the sobel detector is used for enhancement to gain more discrimination of the frames.

### 2.2.2. Sobel edge detector

The second step in speeding up the proposed system is using the sobel operator edge detector to determine most useful representation of the key frames. Sobel operator detects the edges in the extracted key frames, which is used later to find matchings between consecutive frames. If the sequences of frames have matchings according to a threshold, then the system eliminates the second frame because there is no need to classify it by VGG-16 network. Sobel operator has a high accuracy in detecting edges of images. It uses two 3×3 kernels. The first one is for the horizontal and the second one is for the vertical differences [35]. Figure 4 shows sobel edge detector filters.

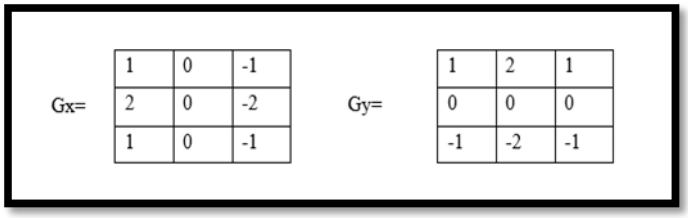



Figure 4. Sobel edge detector filters

The $G_x$ and $G_y$ filters are convolved with the image and result in a gradient of a magnitude that is computed using (1). The gradient direction is computed using (2) [35]. The sobel edge detector is applied to the extracted key frames and the edges are then computed. The output image has the same size as the original one (i.e. 224×224).

$$G = \sqrt{Gx^2 + Gy^2} \quad (1)$$

$$\Theta = atan\frac{Gy}{Gx} \quad (2)$$

### 2.2.3. Max-pooling

After applying the sobel edge detector, a 3×3 max-pooling sliding window is applied to find the maximum number of the area. The extracted features are compared to make the matching decision more accurate. Similar frames are ignored. This method speeds up the system since the analysis and classification are performed on a reduced number of key frames. Max-pooling sliding window of size 3×3 and a frame size of 224×224 was used. The matching decision is based on the number of matching pixels relative to the total number of pixels as shown in (3).

$$Total\ matching\ percentage = (matching\ pixels/total\ pixels)x100\% \quad (3)$$

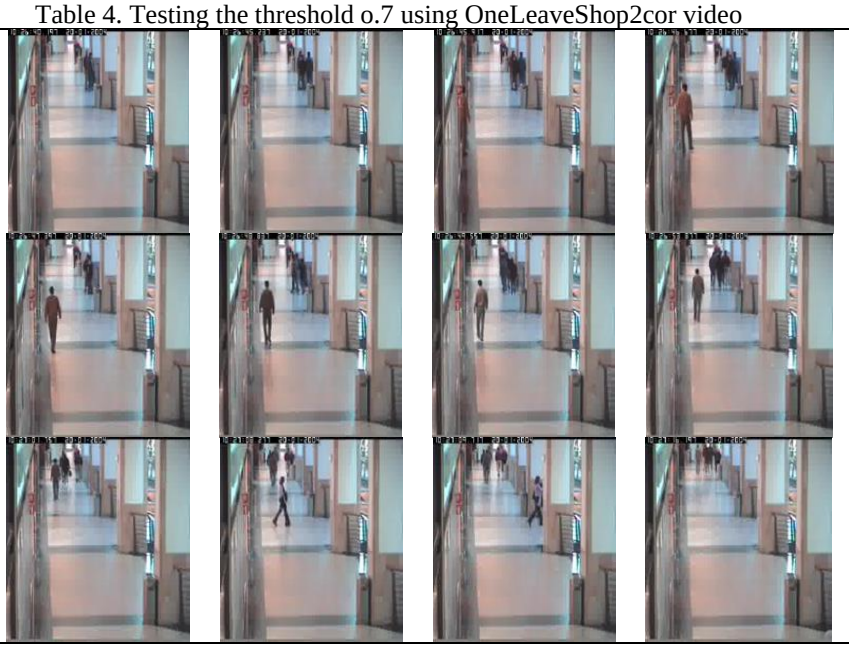
A threshold of 0.7 is used for the matching percentage. If the total match percentage is less than 0.7, then the frame is different from the next frame and needs to be analyzed. Table 3 shows different threshold values that were tested. A value of 0.7 was found to be relatively better in producing accurate results for the key frames. To validate the selected threshold value, OneLeaveShop2cor video from CAVIAR dataset [36] is used. The video contains 63 key frames. However, there are similarities between different frames. Table 4 shows sample images using the threshold value of 0.7. When using the values 0.9 or 1.0 for the threshold, the similarity becomes less than the threshold and all of the frames will be selected and classified.

The performance of eliminating redundant frames is applied to an outdoor surveillance video with a length of 68 seconds. One extracted frame takes 0.135359 seconds to be classified by the VGG-16 network. The proposed system takes 16.804304 seconds to analyze the video without the application of the sobel detector and max-pooling methods. On the other hand, it takes 10.504688 seconds when the enhanced methods are used. This shows how the sobel detector and max-pooling enhance the system speed without altering accuracy. The different steps of the proposed system are summarized as shown in Figure 5.

Table 3. Threshold testing

| Threshold | # of Key Frames | # of Eliminated Key Frames | # of Analyzed Frames |
|---|---|---|---|
| 0.1 | 63 | 63 | 0 |
| 0.2 | 63 | 63 | 0 |
| 0.3 | 63 | 60 | 3 |
| 0.4 | 63 | 55 | 8 |
| 0.5 | 63 | 53 | 10 |
| 0.6 | 63 | 52 | 11 |
| 0.7 | 63 | 50 | 13 |
| 0.8 | 63 | 8 | 55 |
| 0.9 | 63 | 0 | 63 |
| 1 | 63 | 0 | 63 |

Table 4. Testing the threshold o.7 using OneLeaveShop2cor video

| | |
|---|---|
| 0.7 |  |

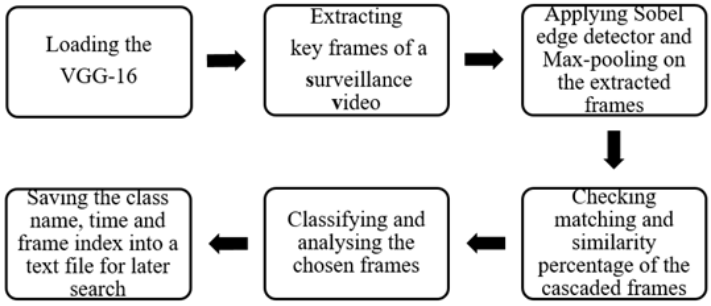



Figure 5. The framework of the proposed system

### 2.3 Testing and evaluation

The video content analysis system is tested on a variety of surveillance videos including different objects, movements, and events. The selected videos are taken from the image video system lab (IVY) dataset [37] and video image retrieval analysis tool (VIRAT) video dataset [38]. The experimental results are shown in Table 5. The proposed system is tested using nine videos. These videos are different from each other in time, frame rate, and the number of key frames. The execution time includes the loading of the trained network, extracting frames, applying sobel detector, applying max-pooling, the matching stage, and saving of the results into a text file. The experiments were performed using Intel i7 PC and Tesla K80 GPU card. The VGG-16 network requires 11 ms on average to classify one frame. Videos numbers 7, 8, and 9 have originally 519, 612, and 31 key frames, respectively. However, using our mechanisms of matching and

ignoring similar frames, the proposed system classified 29, 11, and 12 frames, respectively. Inspecting the videos reveals that they were recorded by fixed cameras and the scene of the site has little information with small transitions. Hence, the proposed system is useful for searching video contents.

Table 5. Experimental results for nine videos

| Video # | Length (Sec) | Total No. of frames | No. of Key frames | Classified frames | Frame rate (Frame/Sec) | CPU Execution Time (Sec) | GPU Execution Time (Sec) |
|---|---|---|---|---|---|---|---|
| Video 1 (ATM) | 29 | 725 | 61 | 15 | 25 | 8 | 0.666 |
| Video 2 (Hallway) | 22 | 660 | 56 | 12 | 30 | 7.80 | 0.71 |
| Video 3 (Stair) | 24 | 720 | 59 | 6 | 30 | 7.70 | 0.625 |
| Video 4 (junction) | 2653 | 66325 | 5527 | 129 | 25 | 1177 | 99.68 |
| Video 5 (parking lot) | 689 | 19981 | 83 | 58 | 29 | 92.415 | 11.089 |
| Video 6 (outdoor) | 302 | 8758 | 37 | 18 | 29 | 43.071 | 3.872 |
| Video 7 (parking lot) | 505 | 15150 | 519 | 29 | 30 | 55.703 | 4.759 |
| Video 8 (parking lot) | 244 | 7076 | 612 | 11 | 29 | 100 | 8.65 |
| Video 9 (ATM) | 59 | 1711 | 31 | 12 | 29 | 9.25 | 0.72 |

In addition, using Tesla K80 speeds up the system 12X than using CPU only. The output text file for video 2 is listed below. Video 2 contains 56 key frames (I) that represent two children walking in the hallway. The frames that were analyzed and classified were 12 as a result of applying the sobel filter and max-pooling. The proposed system was able to analyze the frames in 0.71 seconds using the GPU and 7.80 seconds using the CPU. As seen from the text file of video 2, there was no useful information before time 17.20 seconds. Then, the children appear in the video. This is an efficient way to find objects of interest videos instead of searching manually and spending a lot of time. A user can search within the text file to find a time of interest and then fast forward the video (e.g. to time 17.2 secs). The text file for video 9 (ATM machine) is shown below. Video 9 contains 31 original key frames (I) that represent the area of the ATM machine. The classified frames were only 12 frames. Again, our system was able to analyze the video very fast (0.72 seconds on GPU and 9.25 seconds on CPU) when compared to the length of the video (59 seconds).

Contents of the Text File for Video 2

D:\HALLWAY_A-01.jpeg 0 0.000000 Ashcan, garbage can, score 0.79
D:\HALLWAY_A-44.jpeg 516 17.200000 Child, kid, score 0.79
D:\HALLWAY_A-45.jpeg 528 17.600000 Child, kid, score 0.71
D:\HALLWAY_A-46.jpeg 540 18.000000 Child, kid, score 0.42
D:\HALLWAY_A-47.jpeg 552 18.400000 Child, kid, score 0.63
D:\HALLWAY_A-48.jpeg 564 18.800000 Child, kid, score 0.66
D:\HALLWAY_A-49.jpeg 576 19.200000 Child, kid score 0.44
D:\HALLWAY_A-50.jpeg 588 19.600000 Child, kid, score 0.65
D:\HALLWAY_A-51.jpeg 600 20.000000 Child, kid, score 0.80
D:\HALLWAY_A-52.jpeg 612 20.400000 Child, kid, score 0.59
D:\HALLWAY_A-53.jpeg 624 20.800000 Child, kid, score 0.86
D:\HALLWAY_A-54.jpeg 636 21.200000 Cash machine, ATM, score 0.96

Contents of the Text File for Video 9

D:\ATM105.jpeg 240 8.008000 Flowerpot, score 0.79
D:\ATM106.jpeg 300 10.010000 Flowerpot, score 0.81
D:\ATM110.jpeg 524 17.484133 People, score 0.80
D:\ATM112.jpeg 644 21.488133 Child, kid, score 0.84
D:\ATM113.jpeg 704 23.490133 Child, kid, score 0.81
D:\ATM117.jpeg 928 30.964267 Child, kid, score 0.68
D:\ATM118.jpeg 988 32.966267 Bannister, banister score 0.54
D:\ATM121.jpeg 1168 38.972267 Child, kid, score 0.68
D:\ATM122.jpeg 1228 40.974267 People, score 0.54
D:\ATM125.jpeg 1392 46.446400 People, score 0.66
D:\ATM129.jpeg 1632 54.454400 Bannister, banister score 0.8
D:\ATM130.jpeg 1692 56.456400 Bannister, banister, score 1.00

### 2.4. Comparison with the state of the art

Table 6 shows a comparison between our system and three systems that were proposed in the literature. Lee *et al*. proposed a video content method using a regression model that detects important objects and organizes them as a sequence of images [39]. The system in [39] needs 1 second for each frame to be analyzed with an accuracy of 68.75%. Meghdadi and Irani introduced an analysis system that processes 3 frames per second with an accuracy of about 69% [40]. The system in [40] works by extracting multiple frames based on motion and then visualizing trajectories of the objects in surveillance streams. Zhou *et al*. proposed an approach for video analysis using a space-time saliency method and a faster re-timing method to tackle long video indexing and summarization [30]. The system in [30] requires 11 seconds to process one frame. Table 6 shows that the proposed system outperforms the related work with an order of magnitude. The proposed system processes 6 frames per second. The number of frames of the matching process can reach up to 40 frames per second.

Table 6. Computational time comparison

| Related work | Computational time |
|---|---|
| Lee *et al*. [39] | 1 Frame per second |
| Meghdadi and Irani [40] | 3 frames per second |
| Zhou *et al*. [30] | 1 frame per 11 second |
| The proposed SSVC system | 6 frames per second |

## 3. CONCLUSION

A new system for video content analysis has been proposed. The system uses VGG-16 deep convolutional neural network for object detection and identification. The number of different classes of objects that were considered for security and surveillance systems is 128. However, users can define more classes of their interest. In addition, three techniques were used to reduce the amount of information that needs to be processed by the VGG-16 network. The three techniques are extraction of key frames using FFmpeg, sobel edge detector, and Max-pooling. The sobel detector and max-pooling are used in order to further reduce the number of frames that are needed for processing. The output of the system is a text file that is readable by humans so they can search in it for their objects of interest. The text file contains the objects and movements that occur in a video. Instead of manual video inspection for long periods, the presented system provided users with an easy and simple way for video content searching. The time to produce the text file is negligible compared with the size and time of the analyzed video. The results have showed that the proposed system outperformed existing methods in an order of magnitude.

## REFERENCES

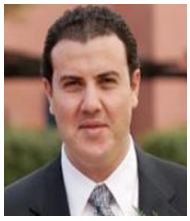
[1] Aljarrah I. and Mohammad D., "Video content analysis using convolutional neural networks," *2018 9th International Conference on Information and Communication Systems (ICICS)*, Irbid, 2018, pp. 122-126.
[2] S. Fakhar A. G., et al., "Development of a portable community video surveillance system," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 9, no. 3, pp. 1814-1821, 2019.
[3] Krausz B. and Herpers R., "MetroSurv: detecting events in subway stations," *Multimedia Tools and Applications*, vol. 50, no. 1, pp. 123-147, 2010.
[4] Prakoso, P. B., and Sari, Y., "Vehicle detection using background subtraction and clustering algorithms," *TELKOMNIKA (Telecommunication, Computing, Electronics and Control)*, vol. 17, no. 3, pp. 1393-1398, 2019.
[5] Prabowo, M. R., Hudayani, N., Purwiyanti, S., Sulistiyanti, S. R., Setyawan, F. A., "A moving objects detection in underwater video using subtraction of the background model," in *2017 4th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*, Yogyakarta, 2017, pp. 1-4.
[6] Krizhevsky A., et al., "ImageNet classification with deep convolutional neural networks," *Advances in neural information processing systems*, pp. 1097-1105, 2012.
[7] Asil, H., and Bagherzadeh, J., "Proposing a new method of image classification based on the AdaBoost deep belief network hybrid method," *TELKOMNIKA (Telecommunication, Computing, Electronics and Control)*, vol. 17, no. 5, pp. 2650-2658, 2019.
[8] Erhan D., et al., "Scalable object detection using deep neural networks," *2014 IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, OH, 2014, pp. 2147-2154.
[9] Yuan, Y., Mou, L. and Lu, X., "Scene recognition by manifold regularized deep learning architecture," *IEEE transactions on neural networks and learning systems*, vol. 26, no. 10, pp. 2222-2233, 2015.
[10] Collobert R and Weston J., "A unified architecture for natural language processing: Deep neural networks with multitask learning," *The 25th international conference on Machine learning*, 2008, pp. 160-167.
[11] Santos, C. N. D., Xiang, B., and Zhou, B., "Classifying relations by ranking with convolutional neural networks," *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, vol. 1, 2015, pp. 626-634.

[12] Zeiler M. D. and Fergus R., "Visualizing and understanding convolutional networks," *European conference on computer vision*, vol. 8689, 2014, pp. 818-833.
[13] Szegedy C., et al., "Going deeper with convolutions," *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, 2015, pp. 1-9.
[14] Simonyan K. and Zisserman A., "Very deep convolutional networks for large-scale image recognition," *arXiv:1409.1556*, 2014.
[15] He K., et al., "Deep residual learning for image recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, 2016, pp. 770-778.
[16] Li H., et al., "A convolutional neural network cascade for face detection," *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, 2015, pp. 5325-5334.
[17] Sun Y., et al., "Deepid3: Face recognition with very deep neural networks," *arXiv:1502.00873*, 2015.
[18] Basheer, N. M., and Mohammed, M. H., "Classification of breast masses in digital mammograms using support vector machines," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 3, no. 10, pp. 57-63, 2013.
[19] Shadeed, G. A., Tawfeeq, M. A., Mahmoud, S. M., "Deep learning model for thorax diseases detection," *TELKOMNIKA (Telecommunication, Computing, Electronics and Control)*, vol. 18, no. 1, pp. 441-449, 2020.
[20] Zhang W, et al., "Computerized detection of clustered microcalcifications in digital mammograms using a shift-invariant artificial neural network," *Medical Physics*, vol. 21, no. 4, pp. 517-524, 1994.
[21] Lao W., et al., "Automatic video-based human motion analyzer for consumer surveillance system," *IEEE Transactions on Consumer Electronics*, vol. 55, no. 2, pp. 591-598, 2009.
[22] Zhao Z. C. and Cai A. N., "Extraction of semantic keyframes based on visual attention and affective models," *2007 International Conference on Computational Intelligence and Security (CIS 2007)*, Harbin, pp. 371-375, 2007.
[23] Bertini M., et al., "An integrated framework for semantic annotation and adaptation," *Multimedia Tools and Applications*, vol. 26, no. 3, pp. 345-363, 2005.
[24] Kolekar M. H., "Bayesian belief network based broadcast sports video indexing," *Multimedia Tools and Applications*, vol. 54, no. 1, pp. 27-54, 2011.
[25] Chen W. E., and Zhang Y. J., "Video segmentation and key frame extraction with parametric model," *2008 3rd International Symposium on Communications, Control and Signal Processing*, St Julians, 2008, pp. 1020-1023.
[26] Sun Z., et al., "Video key frame extraction based on spatial-temporal color distribution," *2008 International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Harbin*, 2008, pp. 196-199.
[27] Sharif, M. H., and Djeraba, C., "An entropy approach for abnormal activities detection in video streams," *Pattern recognition*, vol. 45, no. 7, pp. 2543-2561, 2012.
[28] Cernekova Z., et al., "Information theory-based shot cut/fade detection and video summarization," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 1, pp. 82-91, 2006.
[29] Zeng W., et al., "Robust moving object segmentation on H. 264/AVC compressed video using the block-based MRF model," *Real-Time Imaging*, vol. 11, no. 4, pp. 290-299, 2005.
[30] Zhou F., et al., "Time-mapping using space-time saliency," *2014 IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, OH, 2014, pp. 3358-3365.
[31] Bai L., et al., "Video semantic content analysis based on ontology," *International Machine Vision and Image Processing Conference (IMVIP 2007)*, Kildare, 2007, pp. 117-124.
[32] Foggia P., et al., "Real-time fire detection for video-surveillance applications using a combination of experts based on color, shape, and motion," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 9, pp. 1545-1556, 2015.
[33] Begeja L., et al., "Vidcat: an image and video analysis service for personal media management," *Multimedia Content and Mobile Devices*, vol. 8667, 2013.
[34] "IMAGENET," 2020. [Online]. Available: http://www.image-net.org/challenges/LSVRC/2014/results.
[35] Kittler J., "On the accuracy of the Sobel edge detector," *Image and Vision Computing*, vol. 1, no. 1, pp. 37-42, 1983.
[36] "CAVIAR project," 2020. [Online]. Available: http://groups.inf.ed.ac.uk/vision/CAVIAR/CAVIARDATA1.
[37] "IMAGE and Video SYSTEMS LAB," 2020. [Online]. Available: http://ivylab.kaist.ac.kr/default/.
[38] "VIRAT video dataset," 2020. [Online]. Available: http://www.viratdata.org/.
[39] Lee Y. J., et al., "Discovering important people and objects for egocentric video summarization," *2012 IEEE Conference on Computer Vision and Pattern Recognition*, Providence, RI, 2012, pp 1346-1353.
[40] Meghdadi A. H. and Irani P., "Interactive exploration of surveillance video through action shot summarization and trajectory visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, pp. 2119-2128, 2013.

## BIOGRAPHIES OF AUTHORS

**Duaa Mohammad** received her B.S. degree from the Department of Computer and Network Engineering at Al-Balqa Applied University (BAU) in 2014. She is currently a Master student at the Department of Computer Engineering at Jordan University of Science and Technology. She is also working as a security network engineer at the Land Transport Regulatory Commission in Amman, Jordan. Her research interests include artificial intelligence and machine learning.

**Inad Aljarrah** is an Associate Professor at the Department of Computer Engineering at Jordan University of Science and Technology. He received his B.S. in Electrical Engineering from Jordan University of Science and Technology in 1999. He received his Masters and Ph.D. degrees in Electrical and Computer Engineering from Ohio University, Athens, Ohio, USA in the years 2002 and 2006 respectively. His research interests are computer vision, image processing and analysis, artificial intelligent systems. Currently he is a visiting professor at the Electrical and Computer engineering department at North Carolina State University Raleigh, NC.



**Moath Jarrah** is an Associate Professor at the Department of Computer Engineering at Jordan University of Science and Technology (JUST). He has received his Master and Ph.D. degrees from the Department of Electrical and Computer Engineering at the University of Arizona, Tucson, USA, in 2005 and 2008, respectively. He received the B.S degree from the Department of Electrical and Computer Engineering at JUST in 2002. Jarrah worked a research fellow at Rolls-Royce corporate lab at Nanyang Technological University (NTU), Singapore; and as a scholar researcher at the Department of Creative IT Engineering and at POSTECH University, South Korea in the period of 2014 to 2016. He has co-authored and published many papers in top-tier journals and conferences. His research interests include modeling and simulation, distributed systems, cloud computing, smart grid, machine learning, and optimization.