Research Article

Journal of Sensor Networks and Data Communications

Deep Surveillance System

Pritam Naharwal[1*], Sudhir Singh Mushuni[2], Himanshu Joshi[3] and Shalini Goel[4]

[1,2,3]*Student, Dept. Of Computer Science and Engineering, HMR Institute of Technology and Management, Hamidpur, Delhi, India*

[4]*Associate Professor, Dept. Of Computer Science and Engineering, HMR Institute of Technology and Management, Hamidpur, Delhi, India*

[*]Corresponding Author
Pritam Naharwal, Student, Dept. Of Computer Science and Engineering, HMR Institute of Technology and Management, Hamidpur, Delhi, India.

Submitted: 2023, Aug 21; **Accepted**:2023, Sep 22: **Published**: 2023, Sep 28

Citation: Naharwal, P., Mushuni, S. S., Joshi, H., Goel, S. (2023). Deep Surveillance System. *J Sen Net Data Comm, 3*(1), 93-106.

Abstract

This study has been undertaken to investigate and implement multiple detection systems into a single surveillance system and check whether the input videos may comprise and capture a variety of realistic anomalies or not. In this paper, we propose to learn various anomalies by exploiting both normal and anomalous videos and implemented it to new model. Real time object detection is a vast, vibrant and sophisticated area of computer vision aimed towards object identification and recognition. Object detection detects the semantic objects of a class objects using Open source Computer Vision, which is a library of programming functions mainly trained towards real time computer vision in digital images and videos. The main aim behind this real time object detection is to help the peoples to overcome their difficulty. Real time object detection finds its uses in the areas like tracking objects, video surveillance, pedestrian detection, people counting, self-driving cars, face detection, tracking in sports and many more. This is achieved using Convolution, Probabilistic Neural Networks, etc. which are a representative tool of Deep learning. This project acts as an aiding tool for peoples who wants to take care of everything inside, outside, and around their house just for their full security expectations. Surveillance is a must for small houses to large-scale industries as they fulfil our safety aspects because theft and burglary have always been a problem. By combining this Surveillance idea to IoT and some Machine Learning stuff this will be a major product. The proposed project is a single autonomous surveillance system, based on analysis and detection technology. The proposed system is capable of monitoring all actions at once and alerts the concerned officials immediately and precisely.

Keywords: Smart Video Surveillance System, Human & Object Tracking & Detection, Real Time Video Surveillance, Deep Learning Surveillance Techniques, Video Recognition, Automatic Surveillance with Alert System.

1. Introduction

Surveillance system plays a key role in maintaining the security in today's life. But, fail to provide the feature of avoiding the unfortunate happenings. This work comes up with an idea of smart surveillance system designed such a way that the user is notified when upcoming dangers are detected. In this work, motion of the intruders and the presence of a human faces are detected using image processing algorithms and various tracking and detection models included. As the motion detected in the locality of the surveillance area, a short video clip is recorded and sent to the user along with an alert message through the email server. Live streaming video from the camera is accessed on any Internet enabled device, that's the basic idea behind the following system called deep surveillance system using deep learning techniques.

During these recent years, applications of video surveillance have attracted more and more researchers. Consequently, various types of modeling, as well as several techniques of analysis and detection of human activities, are suggested. Particularly, many pieces of research are involved in the recognition and detection of human activities in general and especially abnormal activities. One important application is the supervision of elderly and disabled people at home in care centers, or hospitals. Recognition of human activities is a recent field that is interested to provide techniques and methods allowing the detection and classification of human activities, and extended now to recognize normal or abnormal activities. The motivation behind the latter is to provide an immediate intervention to preserve the lives of individuals or to ensure them some services they are unable to do by themselves. Being recent and interesting, this field has attracted the attention of several researchers who try to find solutions to the problems faced in studying such types of activities. However, the proposals made

for this until now are those used for the recognition of normal human activities with minor modifications. These proposals are still very restricted because of the very limited number of works and surveys in this field. Moreover, they are not efficient and suffer from several limitations and technical difficulties. To this end, we propose in this paper an overview and an analysis of the existing works, to offer the researchers a general view of what exists in this field and to provide them with a tool being helpful to them propose new approaches. For this, the manuscript is organized as follows. In the second section, we present a definition of the abnormal activities, their various types, as well as some examples of abnormal activities of a group or a single person. We then discuss in the third section the motivations that led to the advent of this research axis and the development of techniques allowing the analysis and recognition of human activities in general and abnormal activities in particular. The fourth section is devoted to the proposed approaches in the literature for the detection of abnormal activities. In this section, we present for each proposal, the purpose for which it is set up, its different stages, and the means used for its validation. Subsequently, we discuss some aspects affecting or influencing the effectiveness and credibility of the classification of human activities. The sixth section presents the three modes of automatic learning (supervised, unsupervised, and semi supervised). Thereafter, we enumerate the encountered limitations to be taken into consideration to improve the systems of recognition and identification of abnormal activities. Finally, we finish with a conclusion where we summarize our study.

2. Literature Survey

Surveillance cameras are increasingly being used in public places e.g. streets, intersections, banks, shopping malls, etc. to increase public safety. However, the monitoring capability of enforcement agencies has not kept pace [1]. The result is that there is a glaring deficiency in the utilization of surveillance cameras and an unworkable ratio of cameras to human monitors. One critical task in video surveillance is detecting anomalous events such as traffic accidents, crimes or illegal activities. Generally, anomalous events rarely occur as compared to normal activities [2,3]. Therefore, to alleviate the waste of labor and time, developing intelligent computer vision algorithms for automatic video anomaly detection is a pressing need. The goal of a practical anomaly detection system is to timely signal an activity that deviates normal patterns and identify the time window of the occurring anomaly [4]. Therefore, anomaly detection can be considered as coarse level video understanding, which filters out anomalies from normal patterns. Once an anomaly is detected, it can further be categorized into one of the specific activities using classification techniques [2,5,6].

A small step towards addressing anomaly detection is to develop algorithms to detect a specific anomalous event, for example violence detector and traffic accident detector [7,8,9]. However, it is obvious that such solutions cannot be generalized to detect other anomalous events, therefore they render a limited use in practice. Real-world anomalous events are complicated and diverse. It is difficult to list all of the possible anomalous events. Therefore, it is desirable that the anomaly detection algorithm does not rely on any prior information about the events. In other words, anomaly detection should be done with minimum supervision. Sparse coding based approaches are considered as representative methods that achieve state-of-the-art anomaly detection results [10,11]. These methods assume that only a small initial portion of a video contains normal events, and therefore the initial portion is used to build the normal event dictionary. Then, the main idea for anomaly detection is that anomalous events are not accurately reconstruct able from the normal event. However, since the environment captured by surveillance cameras can change drastically over the time, these approaches produce high false alarm rates for different normal behaviors. Although the above mentioned approaches are appealing, they assume that any pattern that deviates from the learned normal patterns would be considered as an anomaly. However, this assumption may not hold true because it is very difficult or impossible to define a normal event which takes all possible normal patterns/behaviors into account [1].

More importantly, the boundary between normal and anomalous behaviors is often ambiguous. In addition, under realistic conditions, the same behavior could be a normal or an anomalous behavior under different conditions. Therefore, it is argued that the training data of normal and anomalous events can help an anomaly detection system learn better. To formulate a weakly-supervised learning approach, we resort to multiple instance learning [12,13]. Specifically, we propose to learn anomaly through a deep MIL framework by treating normal and anomalous surveillance videos as bags and short segments/clips of each video as instances in a bag. Based on training videos, we automatically learn an anomaly ranking model that predicts high anomaly scores for anomalous segments in a video. During testing, a long untrimmed video is divided into segments and fed into our deep network which assigns anomaly score for each video segment such that an anomaly can be detected. We propose a MIL solution to anomaly detection by leveraging only weakly labeled training videos. We propose a MIL ranking loss with sparsity and smoothness constraints for a deep learning network to learn anomaly scores for video segments. To the best of our knowledge, we are the first to formulate the video anomaly detection problem in the context of MIL. We introduce a large-scale video anomaly detection dataset consisting of hundreds of real-world surveillance videos of different anomalous events and normal activities captured by surveillance cameras. Experimental results on our new datasets show that our proposed method achieves superior performance as compared to the state-of-the-art anomaly detection approaches.

Anomaly detection is one of the most challenging and long-standing problems in computer vision [14,15,3,16]. For video surveillance applications, there are several attempts to detect violence or aggression in videos [17,18,6]. Datta et al. proposed to detect human violence by exploiting motion and limbs orientation of people. Kooij et al. employed video and audio data to detect aggressive actions in surveillance videos. Gao et al. proposed violent flow descriptors to detect violence in crowd videos [19].

More recently, Mohammadi et al. proposed a new behavior heuristic based approach to classify violent and non-violent videos [6]. Beyond violent and non-violent patterns discrimination, authors in proposed to use tracking to model the normal motion of people and detect deviation from that normal motion as an anomaly [20,5]. Due to difficulties in obtaining reliable tracks, several approaches avoid tracking and learn global motion patterns through histogram-based methods, topic modeling , motion patterns, social force models, mixtures of dynamic textures model, Hidden Markov Model (HMM) on local spatial-temporal volumes, and context-driven method [21,3,22,23,16,24,25]. Given the training videos of normal behaviors, these approaches learn distributions of normal motion patterns and detect low probable patterns as anomalies. Following the success of sparse representation and dictionary learning approaches in several computer vision problems, researchers in used sparse representation to learn the dictionary of normal behaviors [10,26]. During testing, the patterns which have large reconstruction errors are considered as anomalous behaviors. Due to successful demonstration of deep learning for image classification, several approaches have been proposed for video action classification [27,28]. However, obtaining annotations for training is difficult and laborious, specifically for videos. Recently, used deep learning based auto-encoders to learn the model of normal behaviors and employed reconstruction loss to detect anomalies [15,29]. Our approach not only considers normal behaviors but also anomalous behaviors for anomaly detection, using enriched labeled training data.

All deep ranking methods require a vast amount of annotations of positive and negative samples. In contrast to the existing methods, we formulate anomaly detection as a regression problem in the ranking framework by utilizing normal and anomalous data. To alleviate the difficulty of obtaining precise segment-level labels (i.e. temporal annotations of the anomalous parts in videos) for training, we leverage multiple instance learning which relies on weakly labeled data (i.e. video-level labels – normal or abnormal, which are much easier to obtain than temporal annotations) to learn the anomaly model and detect video segment level anomaly during testing.

3. Methodology

Software:

♦ Python

It is a high-level, flexible, simple coding programming language. it uses an interpreter and widely used for generalpurpose programming. This language can support structural and object-oriented programming, imperative, functional programming, and procedural styles. Python uses whitespace indentation to delimit code blocks which allows programs to be coded in fewer lines of code. It is very flexible, because of its ability to use modular components that were designed in other programming languages like c++, java etc. It has a large number of libraries like NumPy, SciPy, and Matplotlib etc with specialized libraries such as Biopython and Astropy.

♦ OpenCV

It was created by Intel to accelerate commercial applications of computer vision with computational efficiency and a strong focus on real-time applications in mind. It's an open-source computer-vision which is free for commercial, public & academic use, and its libraries can greatly simplify computer vision programming. OpenCV can take advantage of multicore processing and has so many advanced capabilities like face detection, face tracking, face recognition, Kalman filtering, and a variety of artificial intelligence (AI) methods in plug and play form. OpenCV is a multi-platform framework which supports both Windows, Linux, IOS, Android and Mac OS X and has C++, C, Python and Java interfaces.

♦ Haar Cascade Classifier

It uses Haar-like features for object detection. Haar-like features are digital image features used in object recognition. The detection algorithm is based upon an approach for human upright facial detection introduced by Viola and Jones. This algorithm designs a system by giving input as a huge number of positive pictures and negative pictures and train a classifier to detect the object. It consists of four main steps :-

• **Haar Features:** Haar features or digital features like shown in above images are used. Each feature is compared with image and a single value obtained by subtracting the sum of pixels under white rectangle from the sum of pixels under black rectangle.

• **Integral Image:** An integral image is defined as two dimensional looked up tables in the form of a matrix with the same size of the original image. This is calculated for all features on all images by using values of neighbouring features and finds the best threshold to find positives and negatives. We select the features with minimum error rate. Haar features are calculated all over the image which may have many features per image.

• **Adaboost:** Summing up the entire image pixel and then subtracting them to get a single value is not efficient in realtime applications. This can be reduced by using Ada boost classifier. Ada boost reduces the redundant features. Here instead of summing up all the pixels, the integral image is used. Adaboost classifies relevant features and irrelevant features. After identifying relevant features and irrelevant features the Adaboost assigns a weight to all of them. It constructs a strong classifier by combining many Weak classifiers.

• **Cascading:** This strong classifier is used to create a cascading sheet which consists of all the mathematical calculations required to detect the targeted animals from the given training dataset. This cascading sheet is in an XML format which is used by OpenCV to detect objects in real time.

♦ K-Nearest Neighbours Algorithm

The K-nearest neighbours (KNN) algorithm is a classifier under supervised learning of algorithms that is often used to classify complex data. Here we give a labeled training dataset consisting of a relationship between 2 points xx and yy. it learns a function h: X→Y, h: X→Y that predict yy using xx.

This consist of two learning algorithms namely :

• **Non-Parametric:** It makes no assumptions about the function h, avoiding the dangers of modeling the underlying distribution of the data.

• **Instance-Based:** The algorithm doesn't learn any model. Instead, it chooses to memorize the training instances for the prediction phase. dist.((x, y), (a, b)) = $\sqrt{(x - a)^2 + (y - b)^2}$

This algorithm uses the Euclidean distance formula to calculate the nearest neighbour around our target and boils down to forming a majority vote between the K most similar instances to a given "unseen" observation. The K value should be larger to classify the object more effectively.

Deep learning, a subset of machine learning which in turn is a subset of artificial intelligence (AI) has networks capable of learning things from the data that is unstructured or unlabeled. The approach utilized in this project is Convolutional Neural Networks (CNN). It uses the Haar-cascade classifiers which help us in the detection of objects.

• CNN:

The convolutional neural network, or CNN for brief, could also be a specialized kind of neural network model designed for working with two-dimensional image data, although they're going to be used with one-dimensional and three-dimensional data. Central the convolutional neural network is the convolutional layer that gives the network its name. This layer performs an operation known as "convolution". In the context of a convolutional neural network, a convolution may be a linear operation that involves the multiplication of a group of weights with the input, very similar to a standard neural network. as long as the technique was designed for two- dimensional input, the multiplication is performed between an array of input file and a two-dimensional array of weights, called a filter or a kernel. The filter is smaller than the input file and therefore the before the sort of multiplication applied between a filter-sized patch of the input and the filter may be a scalar product. A scalar product is that the element-wise multiplication between the filter-sized patch of the input and filter, which is then summed, always leading to one value. Because it leads to 1 value, the operation is conventionally represented and mentioned because the "scalar product". Using a filter smaller than the input is intentional because it allows an equivalent filter (set of weights) to be multiplied by the input array multiple times at distinct points on the input. Specifically, the filter is applied systematically to every overlapping part or filter-sized patch of the input file, left to right, top to bottom.

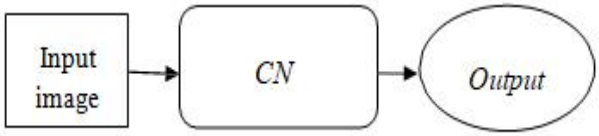


Figure 1: Block Diagram Indicating Image Processing Flow Using CNN

This systematic application of an equivalent filter across a picture may be a powerful idea. If the filter is meant to detect a selected sort of feature within the input, then the appliance of that filter systematically across the whole input image allows the filter a chance to get that feature anywhere within the image. This capability is usually represented and mentioned as translation in variance, e.g. the total altogether concerns in whether the feature is present instead of where it should had been present.

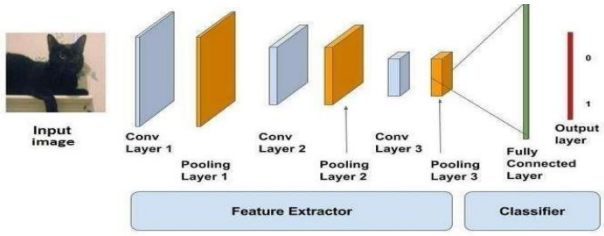


Figure 2: Image Classification Using CNN

• Training the Data Set

The data set is typically the gathering of knowledge . the info set could also be collection of images or alphabets or numbers or documents and files too. the info set we used for the thing detection is that the collection of images of all the objects that are to be identified. Several different images of every and each object is typically present within the data set. If there are more number of images like each object within the datasets then the accuracy are often improved. The important thing that's to be remembered is that the info within the data set must be labeled, there'll be actually 3 data set. they're the training data set, the validation dataset and therefore the other one is testing data set. The training data set will usually contain around 85-90% of the entire labeled data. This training dataset are going to be training our machine and therefore the model is obtained by training the info set. The validation data set consists of around 5-10% of the entire labeled data. this is often used for the validation purpose. the opposite data set is that the testing dataset and it's wont to test the performance of our machine.

• Developing a Real Time Object Detector

For developing a true time object detector using deep learning and OpenCV we'd like to access our webcam during a really effective way then the thing detection is to be applied to each and every frame. we should always install OpenCV in our systems.

The deep neural network module should be installed. Firstly, we should always import all the specified packages:

- From imutils.video we'll import VideoStream
- From imutils.video we'll import FPS
- We'll import numpy as np
- We'll import argparse
- We'll import imutils
- We'll import time
- We'll import cv2

The next step is to construct the argument parse then we should always parse the arguments.
--prototxt: provide path to the Caffe prototxt file.
--model: provide path to the pre-trained model.
--confidence: minimum probability threshold to filter weak detection.

The next step is to initialize CLASS labels and corresponding random COLORS. Each object when it's detected, it's surrounded by a box with some predefined colour. Thus, we assign each object a specific color. After that we'll load our model and that we will provide the regard to our prototxt and also to our model files. With the assistance of imutils we'll read the video and that we will set the amount of frames per second. Now with this some predefined number of frames are going to be loaded per second. Each frame is analogous to the image. Now these images are going to be given because the inputs to the model. The model will process the input image and produces the output image which consists of labels. in additional practical sense the input raw image is given to the model. Now the model processes the input image. within the output image all the thing s are identified and every object is surrounded by an oblong box and therefore the name of the object is additionally displayed. we'll be only observing the output video stream but not the input video stream.

Object detection is a Computer technology that is used to identify different objects in digital images like humans, animals etc. There are many applications depending upon this Technology like robotics, security, face detection, medical to name a few. Object detection algorithms mainly used to extract features to recognize instances of an object. Detecting object like an animal in surveillance videos is a challenging task due to their different appearances and variety of poses they can adopt.

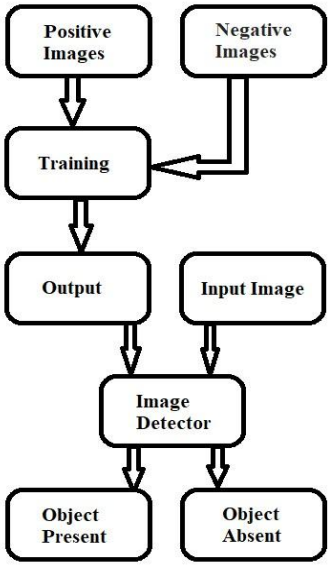


Figure 3: Object Detection Process

To extract these features, we first train the program to detect required objects and it gives us an XML file of the objects required features. For this we give a huge set of positive images with the object is present and negative images with the object not present to the algorithm. The algorithm uses various detection techniques like haar features, neural networks etc., to find a set of usable features. To get these features the algorithm scans the images thousands of features and come out with a set of useful features in form of an XML file. Then we use these features to scan a digital image and identify if the object is present or not. We do it by scanning the image to identify if the output features are present or not. If they are present we take the images positive i.e. object is present, or else we take it as a negative i.e object is absent. This is how an object is detected in an image.

4. Implementation

Here are some of the project implementation of object tracking, detection and analysis.

♦ Face Detection and Recognition

Face detection perhaps be a separate class of object detection. We wonder how some applications like Facebook, Faceapp, etc., detect and recognize our faces. this is often a sample example of object detection in our day to day life. Face detection is already in use in our lifestyle to unlock our mobile phones and for other security systems to scale back rate.

♦ Human & Object-Tracking

Object detection is additionally utilized in tracking objects like tracking an individual and his actions, continuously monitoring a ball within the game of Football or Cricket. As there's an enormous interest for people in these games, these tracking techniques enables them to know it during a better way and obtain some additional information. Tracking of the ball is of maximal importance in any ball-based games to automatically record the movement of the ball and adjust the video frame accordingly.

♦ Self-Driving Cars

This is often one among the main evolution of the planet and is that the best example why we'd like object detection. so as for a car to travel to the specified destination automatically with none human interference or to form decisions whether to accelerate or to use brakes and to spot the objects around it, this needs object detection.

♦ Emotions Detection

This permits the system to spot the type of emotion the person puts on his face. the corporate Apple has already tried to use this by detecting the emotion of the user and converting it into a respective emoji within the smart phone.

♦ Biometric Identification through Retina Scan

Retina scan through iris code is one among the techniques utilized in high security systems because it is one among the foremost accurate and unique biometric.

♦ Smart Text Search and Text Selection (Google lens)

In recent times, we've encountered an application in smart phones called google lens. this will recognize the text and also images and search the relevant information within the browser without much effort.

Types of an anomaly to detect object or behavior some are as follows :

- Video-based abnormal human behavior recognition.

Figure 4: Example of Difference from Walking or Jogging

This technique only focuses on updating anomalous human activity detection. The hidden Markov Model (HMM) and Dynamic Bayesian Network Model (DBNM) are using to detect suspicious behavior as shown in Fig. 1

- Motion detection, tracking, and classification for automated video surveillance.

Figure 5: Tracking of Moving Object

Existing System

- In the existing system, the video surveillance system is designed for human operators to observe protected.
- Space or to record video data for further detection.
- But watching surveillance video is a labor intensive need to be controlled.
- It is also a very tedious and time-consuming job and human observers can easily lose attention.

Proposed System

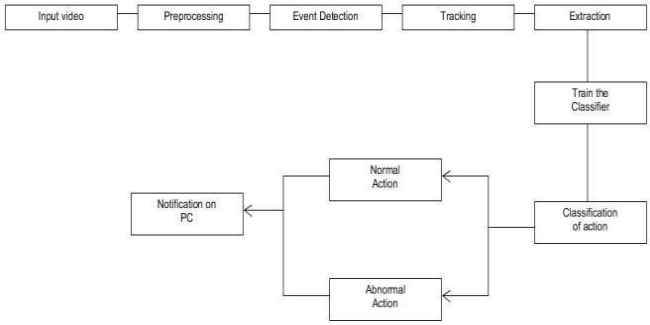


Figure 6: The Proposed Activity Recognition Framework for Surveillance Applications

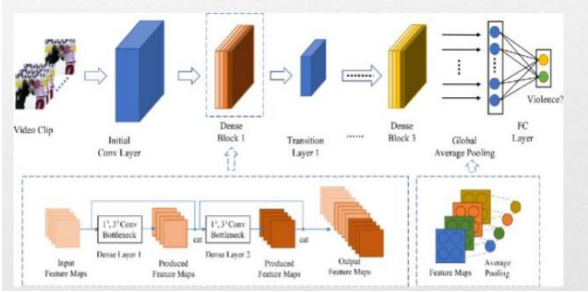


Figure7: Architecture Diagram

♦ Capturing Video

OpenCV is an open-source python library that contains various functions for image and video operations. With OpenCV, we can capture a video from the camera. cv2.VideoCapture () method is defined to get a video capture object for camera. Create an infinite loop and use the read () method to read the frames using above created object. Cv2.imshow () method is used to show the frames in the video. Loop will break when the user clicks a specific key.

♦ Motion Detection

In the proposed work, Motion detection is performed by using OpenCV and Pandas library. Captured videos are treated as a stack of pictures called frames. Different frames are compared to the static frame which has no movements. We compared two images by comparing the intensity value of each pixel.

Firstly, we convert a color image into a grayscale image, then a gray-scale image is converted to GuassianBlur so that change can be easily found. After that difference between the static background and the current frame is found out. If we found to change between them is greater than 30 it will show white color. Then contour of the moving object.

♦ Feature Extraction

Feature extraction is the process of identifying the important features of the data. It reduces an initial set of raw data to more manageable groups for processing. So here, we will start with reading colored images, using the imread () method. Using the shape function, the shape of the image is found out. Suppose the shape for the image is 375*500.

So the number of features will be 187500. If you want to change the shape of the image that is also can be done by using reshape function from NumPy where we specify the dimension of the image.

For this scenario, the image has a dimension (375, 500, and 3). This three represent the RGB value as well as the number of channels.

Now we will use the previous method to create the features. The total number of features will be for this case 375*500*3 = 562500. This colored image has a 3D matrix of dimension (375*500 * 3) where 375 denotes the height, 500 stands for the width and 3 is the number of channels.

To get the average pixel values for the image, we will use a for a loop. Now we will make a new matrix that will have the same height and width but only 1 channel. To convert the matrix into a 1D array we will use the Numpy library. CT is found out.

♦ Classifier

An important step for surveillance activity recognition is to detect, localize, and track each individual throughout the video stream. This task is not feasible with object detectors that are trained on general categories of data. For this purpose, we fine-tuned a light-weight CNN model for human detection with new data and enabled it to work in a changing surveillance environment. It is superior to state-of-the-art methods, its effectiveness is verified from experiment. This architecture makes our system able to achieve LSTM-level accuracy while being more efficient than the LSTM.

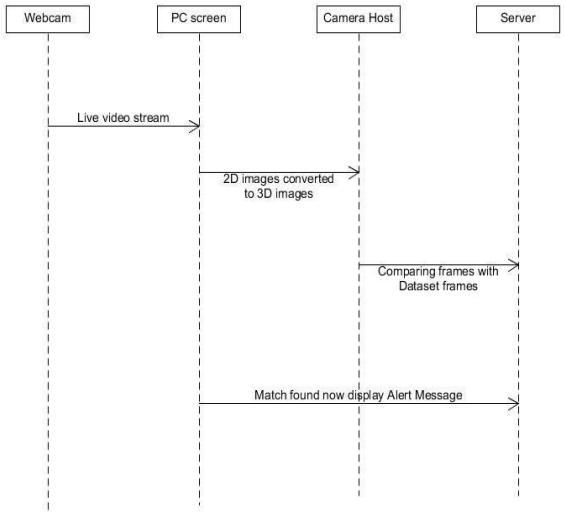


Figure 8: Sequence Diagram

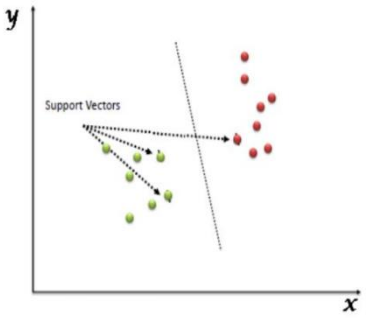


Figure 9: Support Vector Machine (SVM)

• R-transform is used to extract periodic, scale & translation invariant features.
• Different algorithms are used as a non-linear technique to overcome the similarities among different classes of activities.
• SVM (Support Vector Machine) Algorithm is used for training & recognition of activities due to its suitability for time dependent sequential data in this research.

We briefly review the existing video anomaly detection datasets. The UMN dataset consists of five different staged videos where people walk around and after some time start running in different directions. The anomaly is characterized by only running action. UCSD Ped1 and Ped2 datasets contain 70 and 28 surveillance videos, respectively. Those videos are captured at only one location. The anomalies in the videos are simple and do not reflect realistic anomalies in video surveillance, e.g. people walking across a walkway, non pedestrian entities (skater, biker and wheelchair) in the walkways. Avenue dataset consists of 37 videos. Although it contains more anomalies, they are staged and captured at one location. Similar to videos in this dataset are short and some of the anomalies are unrealistic (e.g. throwing paper). Subway Exit and MIL Ranking Loss with sparsity and smoothness constraints Subway Entrance datasets contain one long surveillance video each. The two videos capture simple anomalies such as walking in the wrong direction and skipping payment. BOSS dataset is collected from a surveillance camera mounted in a train. It contains anomalies such as harassment, person with a disease, panic situation, as well as normal videos. All anomalies are performed by actors. Overall, the previous datasets for video anomaly detection are small in terms of the number of videos or the length of the video. Variations in abnormalities are also limited. In addition, some anomalies are not realistic.

Due to the limitations of previous datasets, we construct a new large-scale dataset to evaluate our method. It consists of long untrimmed surveillance videos which cover 13 real world anomalies, including Abuse, Arrest, Arson, Assault, Accident, Burglary, Explosion, Fighting, Robbery, Shooting, Stealing, Shoplifting, and Vandalism. These anomalies are selected because they have a significant impact on public safety. We compare our dataset with previous anomaly detection datasets in Table 1. To ensure the quality of our dataset, we train ten annotators (having different levels of computer vision expertise) to collect the dataset. We search videos on YouTube and LiveLeak using text search queries (with slight variations e.g. “car crash”, “road accident”) of each anomaly. We remove videos which fall into any of the following conditions: manually edited, prank videos, not captured by CCTV cameras, taking from news, captured using a hand-held camera,

and containing compilation. We also discard videos in which the anomaly is not clear. With the above video pruning constraints, 950 unedited real-world surveillance videos with clear anomalies are collected. Using the same constraints, 950 normal videos are gathered, leading to a total of 1900 videos in our dataset.

For our anomaly detection method, only video-level labels are required for training. However, in order to evaluate its performance on testing videos, we need to know the temporal annotations, i.e. the start and ending frames of the anomalous event in each testing anomalous video. To this end, we assign the same videos to multiple annotators to label the temporal extent of each anomaly. The final temporal annotations are obtained by averaging annotations of different annotators. The complete dataset is finalized after intense efforts of several weeks.

	# of videos	Average frames	Dataset length	Example anomalies
UCSD Ped1 [27]	70	201	5 min	Bikers, small carts, walking across walkways
UCSD Ped2 [27]	28	163	5 min	Bikers, small carts, walking across walkways
Subway Entrance [3]	1	121,749	1.5 hours	Wrong direction, No payment
Subwa Exit [3]	1	64,901	1.5 hours	Wrong direction, No payment
Avenue [28]	37	839	30 min	Run, throw, new object
UMN [2]	5	1290	5 min	Run
BOSS [1]	12	4052	27 min	Harass, Disease, Panic
Ours	**1900**	**7247**	**128 hours**	**Abuse, arrest, arson, assault, accident, burglary, fighting, robbery**

Table 1: A Comparison of Anomaly Datasets

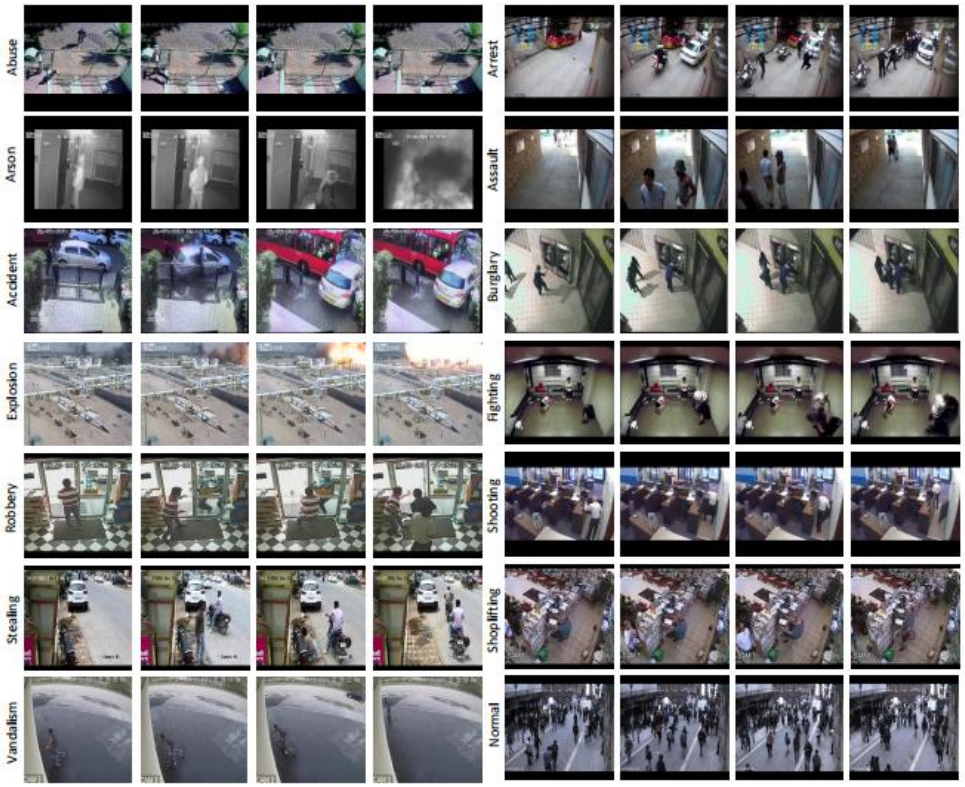
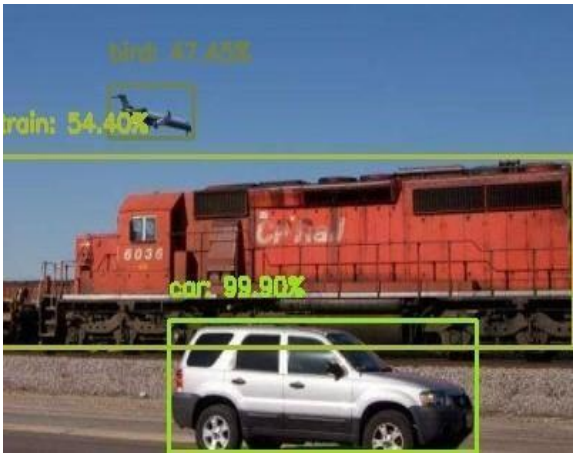


Figure 10: Examples of Different Anomalies from the Training and Testing Videos in our Dataset

5. Target

Here, in this project we've considered around 15 to 20 objects to be detected during the training. Some of those include 'person', 'car', 'weapon', 'train', 'bird', 'sofa', 'dog', ''plant', 'aeroplane', 'bicycle', 'bus', 'motorbike', etc.

The output of this project displays the objects detected with a rectangular box around the object with a label indicating its name and therefore the exactness with which the object has been detected on the top of it. It can dig out any number of objects existing during a single image with certainty.



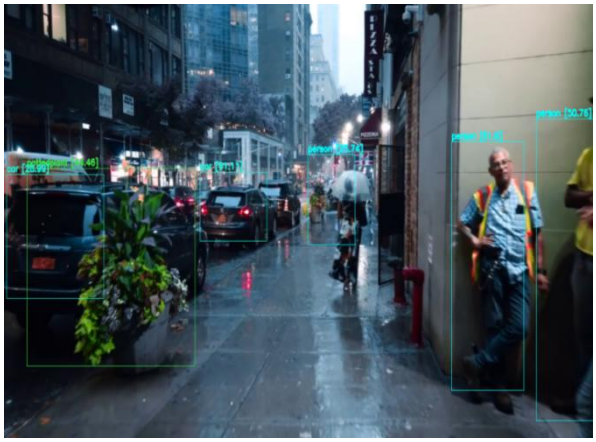


Figure 11: Expected Outcomes (Includes All Declared Objects)

6. Result

Deep Surveillance System consists of front-end which is for the user to access the system easily, and also back-end where the main processing takes place or we can say maximum work held into this phase. It includes all of the sub-systems for detection purposes by just a click on it, as below we have taken some detection cases for result and output purposes.

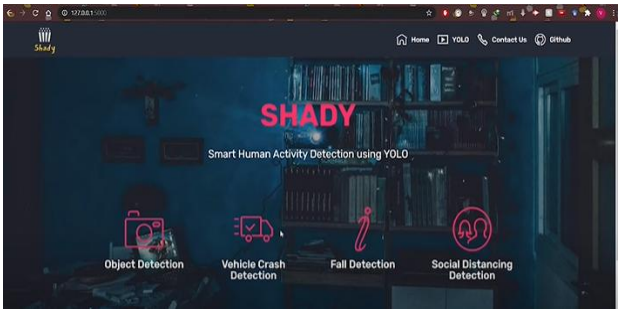


Figure 12: Front-End of System

♦ Cases

1. Distance between cars < particular value
2. Rectangle boxes of cars intersect = Red Box

♦ Alert

Frames (Detection) = 20

♦ Output

1. Display the message “Crash Detected.”
2. Otherwise “Crash Not Detected”

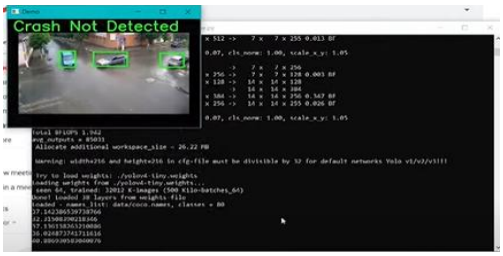


Figure 13: Check Cases for Detection (Back-End)

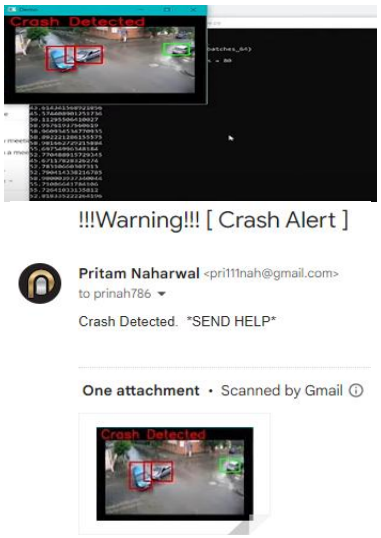


Figure 14: Abnormal Activity Detected and Alert Send to User's Email with Warning

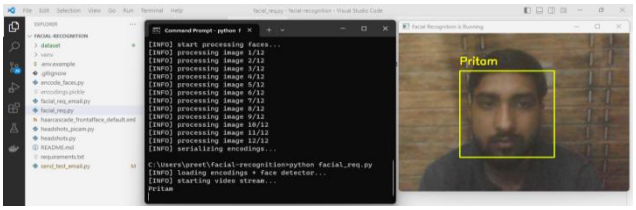


Figure 15: Facial Recognition Sub-System Procedure

Figure 16: Alarm Related Outcomes with Respect to Cases

7. Future Scope

• The proposed system aims to open a new door in the field of video surveillance and provide the result on the basis to detect abnormal activities. It will help the user to monitor any abnormal activities or suspicious events. It's been very difficult to monitor abnormal activities in various fields like security, crime prevention, traffic monitoring. It will help the user by sending an alert message when an abnormal condition is identified.

• The number of parameters currently included is an attempt to cover all the basic aspects of video surveillance and other overlooked parameters which deserve recognition. This work is interested in the recognition of abnormal human activities by providing a brief analysis of the recent research tasks in this field of video surveillance. The scope of this project is to minimize the theft happening in future and maximize the protection of a data in highly confidential region. This will be more essential in industry areas. This will minimize the theft happening. The administrator will be notified at the time of theft happening. The data will be protected with high security in the confidential region. The administrator will be notified regarding the abnormal activity in his/her place by a means of short message service or mail system using pop3 configuration. The process is fast and highly securable.
• In future, we will add more detection sub-systems to make the surveillance system more convenient and more updates will be added with respect to new features which definitely lead into artificial intelligence area.
• Finally, through this analysis of the recent research tasks in this field of video surveillance, we concluded enrich skills and knowledge and provide the final output and result.

8. Conclusion

Deep-learning based object detection has been a search hotspot in recent years. This project starts on generic object detection pipelines which give base architectures for other related tasks. With the assistance of this the 3 other common tasks, namely object detection, face detection and pedestrian detection, are often accomplished. We accomplished this by combing 2 things: Object detection with deep learning and OpenCV and Efficient, threaded video streams with OpenCV. The camera sensor noise and lightening condition can change the result because it can create problem in recognizing the objects. generally, this whole process requires GPU's rather than CPU's. But we've done using CPU's and executes in much less time, making it efficient. Object Detection algorithms act as a mixture of both image classification and object localization. It takes the given image as input and produces the output having the bounding boxes adequate to the number of objects present within the image with the category label attached to every bounding box at the highest. It projects the scenario of the bounding box up the shape of position, height and width.

The proposed system aims to open a new door in the field of video surveillance and provide the result on the basis to detect abnormal activities. It will help the user to monitor any abnormal activities or suspicious events. It's been very difficult to monitor abnormal activities in various fields like security, crime prevention, traffic monitoring. It will help the user by sending an alert message when an abnormal condition is identified. The number of parameters currently included is an attempt to cover all the basic aspects of video surveillance and other overlooked parameters which deserve recognition. This work is interested in the recognition of abnormal human activities by providing a brief analysis of the recent research tasks in this field of video surveillance. We have implemented the CNN to detect abnormal activities. Finally, through this analysis of the recent research tasks in this field of video surveillance, we provide the result on the basis to detect abnormal activities. Security is essential where there is a more secured data. To secure large amount of data or highly securable data in confidential region security plays a major role. For establishing security for a data, it must be protected from an unauthorized person. The camera will capture the video and deep learning concepts will provide the features of facial analysis. When facial features are recognized it must be validated with authorized data sets in a database. The validated face is examined to find whether it is an authorized person or not.

This concept will reduce the theft happening in real world areas and the person who attempt to enter the area in a unproper manner will be notified and caught. This paper reduces the theft happening or unauthorized way of entering [30-50].

References

1. Arunnehru, J., Chamundeeswari, G., & Bharathi, S. P. (2018). Human action recognition using 3D convolutional neural networks with 3D motion cuboids in surveillance videos. Procedia computer science, 133, 471-477.
2. Tzelepis, C., Galanopoulos, D., Mezaris, V., & Patras, I. (2016). Learning to detect video events from zero or very few video examples. Image and vision Computing, 53, 35-44.
3. Wei, X., Du, J., Liang, M., & Ye, L. (2019). Boosting deep attribute learning via support vector regression for fast moving crowd counting. Pattern Recognition Letters, 119, 12-23.
4. Huang, W., Ding, H., & Chen, G. (2018). A novel deep multi-channel residual networks-based metric learning method for moving human localization in video surveillance. Signal Processing, 142, 104-113.
5. Yuan, Y., Zhao, Y., & Wang, Q. (2018). Action recognition using spatial-optical data organization and sequential learning framework. Neurocomputing, 315, 221-233.
6. Hanocka, R., Fish, N., Wang, Z., Giryes, R., Fleishman, S., & Cohen-Or, D. (2018). Alignet: Partial-shape agnostic alignment via unsupervised learning. ACM Transactions on Graphics (TOG), 38(1), 1-14.
7. Potok, T. E., Schuman, C., Young, S., Patton, R., Spedalieri, F., Liu, J., ... & Chakma, G. (2018). A study of complex deep learning networks on high-performance, neuromorphic, and quantum computers. ACM Journal on Emerging Technologies in Computing Systems (JETC), 14(2), 1-21.
8. Wu, G., Lu, W., Gao, G., Zhao, C., & Liu, J. (2016). Regional deep learning model for visual tracking. Neurocomputing, 175, 310-323.
9. Tian, Y., Lee, G. H., He, H., Hsu, C. Y., & Katabi, D. (2018). RF-based fall monitoring using convolutional neural networks. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, 2(3), 1-24.
10. Zhou, T., Tucker, R., Flynn, J., Fyffe, G., & Snavely, N. (2018). Stereo magnification: Learning view synthesis using multiplane images. arXiv preprint arXiv:1805.09817.
11. Pathak, A. R., Pandey, M., & Rautaray, S. (2018). Application of deep learning for object detection. Procedia computer sci-

ence, 132, 1706-1717.
12. Ribeiro, M., Lazzaretti, A. E., & Lopes, H. S. (2018). A study of deep convolutional auto-encoders for anomaly detection in videos. Pattern Recognition Letters, 105, 13-22.
13. Fakhar, B., Kanan, H. R., & Behrad, A. (2018). Learning an event-oriented and discriminative dictionary based on an adaptive label-consistent K-SVD method for event detection in soccer videos. Journal of Visual Communication and Image Representation, 55, 489-503.
14. Shao, L., Cai, Z., Liu, L., & Lu, K. (2017). Performance evaluation of deep feature learning for RGB-D image/video classification. Information Sciences, 385, 266-283.
15. Perez, M., Avila, S., Moreira, D., Moraes, D., Testoni, V., Valle, E., ... & Rocha, A. (2017). Video pornography detection through deep learning techniques and motion information. Neurocomputing, 230, 279-293.
16. Mammadli, R., Wolf, F., & Jannesari, A. (2019). The art of getting deep neural networks in shape. ACM Transactions on Architecture and Code Optimization (TACO), 15(4), 1-21.
17. Huang, H., Xu, Y., Huang, Y., Yang, Q., & Zhou, Z. (2018). Pedestrian tracking by learning deep features. Journal of Visual Communication and Image Representation, 57, 172-175.
18. Hassan, M. M., Uddin, M. Z., Mohamed, A., & Almogren, A. (2018). A robust human activity recognition system using smartphone sensors and deep learning. Future Generation Computer Systems, 81, 307-313.
19. Najva, N., & Bijoy, K. E. (2016). SIFT and tensor based object detection and classification in videos using deep neural networks. Procedia Computer Science, 93, 351-358.
20. Ahmed, S. A., Dogra, D. P., Kar, S., & Roy, P. P. (2018). Surveillance scene representation and trajectory abnormality detection using aggregation of multiple concepts. Expert Systems with Applications, 101, 43-55.
21. Guraya, F. F. E., & Cheikh, F. A. (2015). Neural networks based visual attention model for surveillance videos. Neurocomputing, 149, 1348-1359.
22. Xu, M., Qian, F., Mei, Q., Huang, K., & Liu, X. (2018). Deeptype: On-device deep learning for input personalization service with minimal privacy concern. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, 2(4), 1-26.
23. Fan, Z., Song, X., Xia, T., Jiang, R., Shibasaki, R., & Sakuramachi, R. (2018). Online deep ensemble learning for predicting citywide human mobility. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, 2(3), 1-21.
24. Yu, Z., Li, T., Yu, N., Pan, Y., Chen, H., & Liu, B. (2019). Reconstruction of hidden representation for robust feature extraction. ACM Transactions on Intelligent Systems and Technology (TIST), 10(2), 1-24.
25. Yao, S., Zhao, Y., Shao, H., Zhang, A., Zhang, C., Li, S., & Abdelzaher, T. (2018). Rdeepsense: Reliable deep mobile computing models with uncertainty estimations. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, 1(4), 1-26.
26. Wang, C., Yang, H., & Meinel, C. (2018). Image captioning with deep bidirectional LSTMs and multi-task learning. ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM), 14(2s), 1-20.
27. Nasir, M., Muhammad, K., Lloret, J., Sangaiah, A. K., & Sajjad, M. (2019). Fog computing enabled cost-effective distributed summarization of surveillance videos for smart cities. Journal of Parallel and Distributed Computing, 126, 161-170.
28. Lovering C, Lu A, Nguyen C, Nguyen H, Hurley D, Agu E. Fact or fiction. Proc ACM Hum-Comput Interact. 2018;2:111.
29. Hedman, P., Philip, J., Price, T., Frahm, J. M., Drettakis, G., & Brostow, G. (2018). Deep blending for free-viewpoint image-based rendering. ACM Transactions on Graphics (ToG), 37(6), 1-15.
30. Kardas, K., & Cicekli, N. K. (2017). SVAS: surveillance video analysis system. Expert Systems with Applications, 89, 343-361.
31. Wang, Y., Shuai, Y., Zhu, Y., Zhang, J., & An, P. (2019). Jointly learning perceptually heterogeneous features for blind 3D video quality assessment. Neurocomputing, 332, 298-304.
32. Fakhar, B., Kanan, H. R., & Behrad, A. (2018). Learning an event-oriented and discriminative dictionary based on an adaptive label-consistent K-SVD method for event detection in soccer videos. Journal of Visual Communication and Image Representation, 55, 489-503.
33. Luo, X., Li, H., Cao, D., Yu, Y., Yang, X., & Huang, T. (2018). Towards efficient and objective work sampling: Recognizing workers' activities in site surveillance videos with two-stream convolutional networks. Automation in Construction, 94, 360-370.
34. Tsakanikas, V., & Dagiuklas, T. (2018). Video surveillance systems-current status and future trends. Computers & Electrical Engineering, 70, 736-753.
35. Wang, Y., Zhang, D., Liu, Y., Dai, B., & Lee, L. H. (2019). Enhancing transportation systems via deep learning: A survey. Transportation research part C: emerging technologies, 99, 144-163.
36. Pang, S., del Coz, J. J., Yu, Z., Luaces, O., & Díez, J. (2017). Deep learning to frame objects for visual target tracking. Engineering Applications of Artificial Intelligence, 65, 406-420.
37. Xu, M., Fang, H., Lv, P., Cui, L., Zhang, S., & Zhou, B. (2019). D-STC: Deep learning with spatio-temporal constraints for train drivers detection from videos. Pattern Recognition Letters, 119, 222-228.
38. Pouyanfar, S., Sadiq, S., Yan, Y., Tian, H., Tao, Y., Reyes, M. P., ... & Iyengar, S. S. (2018). A survey on deep learning: Algorithms, techniques, and applications. ACM Computing Surveys (CSUR), 51(5), 1-36.
39. Roy, P., Song, S. L., Krishnamoorthy, S., Vishnu, A., Sengupta, D., & Liu, X. (2018). Numa-caffe: Numa-aware deep learning neural networks. ACM Transactions on Architecture and Code Optimization (TACO), 15(2), 1-26.
40. Ben-Hamu, H., Maron, H., Kezurer, I., Avineri, G., & Lipman, Y. (2018). Multi-chart generative surface modeling. ACM Transactions on Graphics (TOG), 37(6), 1-15.

41. Ge, W., Gong, B., & Yu, Y. (2018). Image super-resolution via deterministic-stochastic synthesis and local statistical rectification. ACM Transactions on Graphics (TOG), 37(6), 1-14.
42. Sundararajan, K., & Woodard, D. L. (2018). Deep learning for biometrics: A survey. ACM Computing Surveys (CSUR), 51(3), 1-34.
43. Kim, H., Kim, T., Kim, J., & Kim, J. J. (2018). Deep neural network optimized to resistive memory with nonlinear current-voltage characteristics. ACM Journal on Emerging Technologies in Computing Systems (JETC), 14(2), 1-17.
44. Liu, D., Cui, W., Jin, K., Guo, Y., & Qu, H. (2018). Deeptracker: Visualizing the training process of convolutional neural networks. ACM Transactions on Intelligent Systems and Technology (TIST), 10(1), 1-25.
45. Yi, L., Huang, H., Liu, D., Kalogerakis, E., Su, H., & Guibas, L. (2018). Deep part induction from articulated object pairs. arXiv preprint arXiv:1809.07417.
46. Zhao, N., Cao, Y., & Lau, R. W. (2018). What characterizes personalities of graphic designs?. ACM Transactions on Graphics (TOG), 37(4), 1-15.
47. Tan, J., Wan, X., Liu, H., & Xiao, J. (2018). QuoteRec: Toward quote recommendation for writing. ACM Transactions on Information Systems (TOIS), 36(3), 1-36.
48. Qu, Y., Fang, B., Zhang, W., Tang, R., Niu, M., Guo, H., ... & He, X. (2018). Product-based neural networks for user response prediction over multi-field categorical data. ACM Transactions on Information Systems (TOIS), 37(1), 1-35.
49. Yin, K., Huang, H., Cohen-Or, D., & Zhang, H. (2018). P2p-net: Bidirectional point displacement net for shape transform. ACM Transactions on Graphics (TOG), 37(4), 1-13.
50. Yao, S., Zhao, Y., Shao, H., Zhang, C., Zhang, A., Hu, S., ... & Abdelzaher, T. (2018). Sensegan: Enabling deep learning for internet of things with a semi-supervised framework. Proceedings of the ACM on interactive, mobile, wearable and ubiquitous technologies, 2(3), 1-21.