Dynamic Image Analysis for Abnormal Behavior Detection

By

MD. Ashik Ahmed (12201009)

Mushfique Ahmed Isha (12201077)

Al-Amin Ahmed (12201102)

Supervised By:

Dr. Amitabha Chakrabarty

Assistant Professor

Department of Computer Science and Engineering

BRAC University

Dhaka, Bangladesh

Thesis report submitted to Department of Computer Science and Engineering of BRAC University in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering.

Submitted on:

18TH APRIL, 2017

BRAC University, Dhaka.

Declaration

We, hereby declare that this thesis is based on results we have found ourselves. Materials of work from researches conducted by others are mentioned in the reference. This work, neither in whole nor in part, has been previously submitted for any other degree or any other publication. All the implementations and functionalities have been used are done by ourselves.
Date: 18th April, 2017.

Signature of Supervisor

Signature of Authors

____

Md. Ashik Ahmed (12201009)

____

Mushfique Ahmed Isha (12201077)

____

Al-Amin Ahmed (12201102)

____

Dr. Amitabha Chakrabarty
Assistant Professor
Department of Computer Science and Engineering
School of Engineering and Computer Science
BRAC University
Dhaka, Bangladesh

Abstract

Our world is now in such developing state where security is more of a concern rather than privacy for an individual. Nowadays, abnormal behavior detection system plays a very important role in various sectors such as, security, prison, bank etc. Abnormal behavior and its definition is different in many cases. In the vivid sense the definition of abnormal behavior, it is something deviating from the normal or differing from the typical scenario. Moreover, this abnormal behavior detection refers to the problem of finding patterns in data that do not conform to expected behavior. For a particular domain abnormal behavior can be different from the classic definition of abnormality. Detection of abnormal behavior is an important area of research in computer vision and is also driven by a wide application domains, such as dynamic image analysis from a video surveillance. Convolutional neural network made this detection and classification way easier and efficient. In this project we are prompted to detect abnormal or suspicious behavior by an individual person. Our purpose is to detect behavior which is not normal from dynamic images taken from a video surveillance. In this case we are using Convolutional Neural Network (CNN) to detect abnormal behavior. In experiments, our proposed system detected the behavior of individuals in normal scenario successfully with the accuracy of 98%. Moreover, it also detects any deviations from previous data for any new scenario from different dynamic images. Our system can be implemented in advanced security purposes.

Keywords: dynamic image; abnormal behavior; human posture; object identity verification; deep learning; convolutional neural network.

Acknowledgement

This idea was suggested to us by Dr. Amitabha Chakrabarty (Assistant Professor), Department of Computer Science and Engineering, BRAC University, Dhaka as our graduation thesis. We shared our idea with him and our desire to work with him in this topic.
We would like to express our gratitude to Almighty Allah who gave us the opportunity, determination, courage, patience in hard times and strength to progressing in our work.

We are also grateful to Toqi Tahamid Sarker, Muhaiminur Rahman Abir, Sayeed Hasan students of BRAC University for being with us till now.

We would like to thank to all our friends and classmates for supporting us and helping us by providing moral support and thank all other faculty members of Computer Science and Engineering Department from whom we gained our knowledge and helped throughout the thesis work.

Last but not least, we would like to thank our parents, our brothers and sisters, for all their enduring support and always believing in us.

Contents

Table of Figures

List of Tables and Graphs

Chapter 1

Introduction

Right now around 7.5 billion people are living on this planet. The population of cities towns is increasing day by day. With the increasing number of population the need for human safety and monitoring security has become a major concern. In the last two decades, the number of surveillance cameras installed in private and public places has increased dramatically. It is happening because of the rising fear in people about crime and terrorism. For this reason Surveillance cameras are set at important places to ensure security like home, airports, banks, city centers and major important places. Nowadays the autonomous visual analysis of surveillance videos and images is a major field of research in computer vision. The main reason is the process that influences the behavior in the monitored scene due to the different variety of situations in real life. One important necessity is to detect the situation when unexpected things happen. There is an increasing desire and need to detect abnormal behavior from live video surveillance. Due to technological advancement image processing for observing abnormal behavior detection has become desired area for research. Successful detection can prevent unwanted events and which can save human lives, assets. If we can apply machine learning, to be more precise Convolutional Neural Networks concept with this the system will become more efficient and cost effective. It is possible that, in the near future this system will be able to give the amount of security an individual expects to get rid of the rising fears about crime, terrorism and personal security. .

1.1 Objectives

- Gaining higher accuracy percentage in detection from dynamic image analysis.
- Providing high edge security system to everyone.
- Analyzing multiple dynamic images from live video surveillance.
- Providing a Robust system along with a good architecture at very low cost, easy to implement, more efficient and which gives proper detection accuracy.

1.2 Motivation

In recent years Human race are the victims of unexpected events and accidents. All of these resulted into a great loss of human lives. We wanted to build a system which will prevent these unknown and unexpected events in a reliable cost effective way. Previous works based on Neural Networks helped to classify objects, image segmentation and create models which is efficient to classify images. Moreover, in our abnormal behavior detection process we are using CNN because it provides a wide application in dynamic image analysis in the form of machine learning. First of all, we tried to implement our work by using manual algorithms which was bit costly to implement and later on we decided to go for Convolutional Neural Networks where CNN is way more cost effective and can be updated inexpensively later on. Using CNN for security and include its classification method for dynamic image analysis was our main concern. CNN consists of multi-layered based neural network where there is huge number of connections between neurons [1]. CNN is a self-learning model where the network is trained using a large dataset of dynamic images according to their weights. It helps the network to learn and then in testing phase system makes assumptions of the input appropriately and gives outputs as far as the output classes are concerned. Finally, this model will help us to build a highly advanced artificial Intelligence system which will be able to recognize abnormal behavior from a data set of multiple dynamic images for a particular domain area where abnormal behavior will be classified by the person who is in need of the security.

1.3 Scope of Thesis

In our thesis, we propose a deep convolutional neural network architecture which will effectively detect abnormal behavior for security purposes. The above mentioned terms have been elaborated later in this paper.

1.4 Features

Our system will use robust model and enriched manual dataset for accurate detection of abnormal behavior. This system will analyze the fluctuation in graph to detect abnormal behavior more precisely. Our system will give the user a clear vision by which it would be the best approach for abnormal behavior detection.

1.5 Thesis Outline

Chapter 1 is the formal introduction of the thesis. We have discussed our motivation, scope, features and objectives.

Chapter 2 is the background study we did before starting to implement the system.

Chapter 3 is focused on the design and architectural view behind our project

Chapter 4 is the detailed information about our system specification. It is focused on the tools we used to implement the system and the reasons and benefits behind them.

Chapter 5 contains experimentation processes, related results and their comparison.

Chapter 6, we mention the conclusion and we have also talked about the future aspects of our system.

References contains important citations which are used in this report.

Chapter 2

Literature Review

In this rapidly developing world ensuring security is a major concern for business authorities and law enforcement agencies. In order to prevent any kind of unexpected events it is crucial to identify any unusual or abnormal behaviors of possible suspects. Before explaining abnormal behavior some relevant keywords such as unusual, rare, suspicious are worth mentioning which were focused in previous research works [2, 3] to point out this concept. Behaviors which fall short from a threshold level are considered abnormal [4]. We can conclude behaviors detected as not usual apart from rare cases are abnormal.

Abnormal behavior detection from still and video images is now a growing research field with various types of applications. In our quest of constructing a system which facing a variety of environmental biasing factors, such as changing illuminations and backgrounds, it must be able to handle non-frontal dynamic images of all genders, different appearances, ages and races, and well enough in the presence of multiple suspects. For instance in [5], [6], [7] never mentioned solving issues arise from illuminations aspects and background noise.

For more accurate detection of any object we found neural network comes in a very helpful way. Adjacent networks can take inputs and detect by taking a particular characteristics and give an accurate output. Moreover, in recent years neural networks has been very useful for object detection. Convolutional neural networking makes this task easier.

Evaluation of neural network in terms of learning speed and generalization places CNN at the top for image recognition tasks [8]. Convolutional neural network has local connectivity concept for solving over-parameterization issue where neurons of hidden layer focused towards the nearby pixels. CNN shares weights in feature map corresponding to neurons [9]. Sub sampling concept detects object in multiple areas in image ensuring detection with the presence of distortions and translations. CNN generalizes better equated to multi-layer perceptron (MLP) or regular neural network [9].

In order to illustrate how advanced it can be for application we are highlighting a recent development. Deeper neural networks are much more difficult to train. A residual learning framework eases the training of networks [10]. On the ImageNet dataset evaluating residual nets having 152 layers comparatively 8 times deeper than VGG nets showed the novelty of CNN. An assembly of residual nets scored 3.57% error on the ImageNet test set. This achieved the 1st place on the ILSVRC 2015 classification task. Analysis on CIFAR-10 with 100 and 1000 layers is also done with it [11, 12].

An approach to identify and localize abnormal behaviors in crowd videos by Social Force model is also notable [2]. Interaction of people represents Force Flow which is calculated by the model and plotted to obtain normal behavior of the crowd using randomly selected spatio-temporal volumes of it. Interaction forces point areas of anomalies in abnormal frames which is categorized by bag of words approach. It works well with escape panic scenarios and crowd videos [2]. Applying this procedure for regular daily-life security of particular environment like office, home is yet to be tested along with the complexity of the method.

Identifying abnormality from surveillance web camera data in real-time with very low frame-rate has also been tried using image features where meaningful nearest neighbours represent usual frame or scene [3]. Algorithm adjusts to variation in the data-stream with the help of incremental learning techniques. As a result it can figure out frames with possible unusual scenes [3].

In this research we attempted a Dynamic Image analysis for abnormal behavior detection in a particular environment by using Convolutional Neural Networking by using a Theano [13] as a Machine learning library to call a number of methods. Moreover, we have also selected Keras [14] as our wrapper library. We are focusing on image data collected from video of usual security camera with very low frame-rate placed in home or office for monitoring security. Our system implemented using our own dataset consisting of around 1000 images for different procedures of varying usages during development stages. For example we split them in random ratios which were assigned for training and testing. And sometimes we manually split them purposefully to test the system.

Our system can effortlessly accommodate new data for training in the same or new environment. This makes it robust, adaptable, less burdened with complex methods and gives an upper-hand compared to other methods mentioned earlier.

Chapter 3

Architecture

When it comes to security technology along with its blessings comes first to assist. Machine learning is a very interesting and useful topic where security is the main priority to get. The scope of machine learning is bit smaller than artificial intelligence but in the vast term Machine learning is all about artificial intelligence. Machine learning is not all about what things are, it is also about what things are not. In our work we have used neural network to complete our task, where our inputs are bunch of images and as in for outputs we got categories of concepts. To define neural network, it is basically a computer program which simulates exactly like a human brain and also works like that. First of all, to use the Neural Network we had to go through few essential steps. The first step or phase is called, Training Phase. Where we are going to take pairs of input data and desired output that has been collected beforehand and learn based on that. It can follow any perimeter. For a bunch of inputs the outputs should be categories of concepts and characteristics.

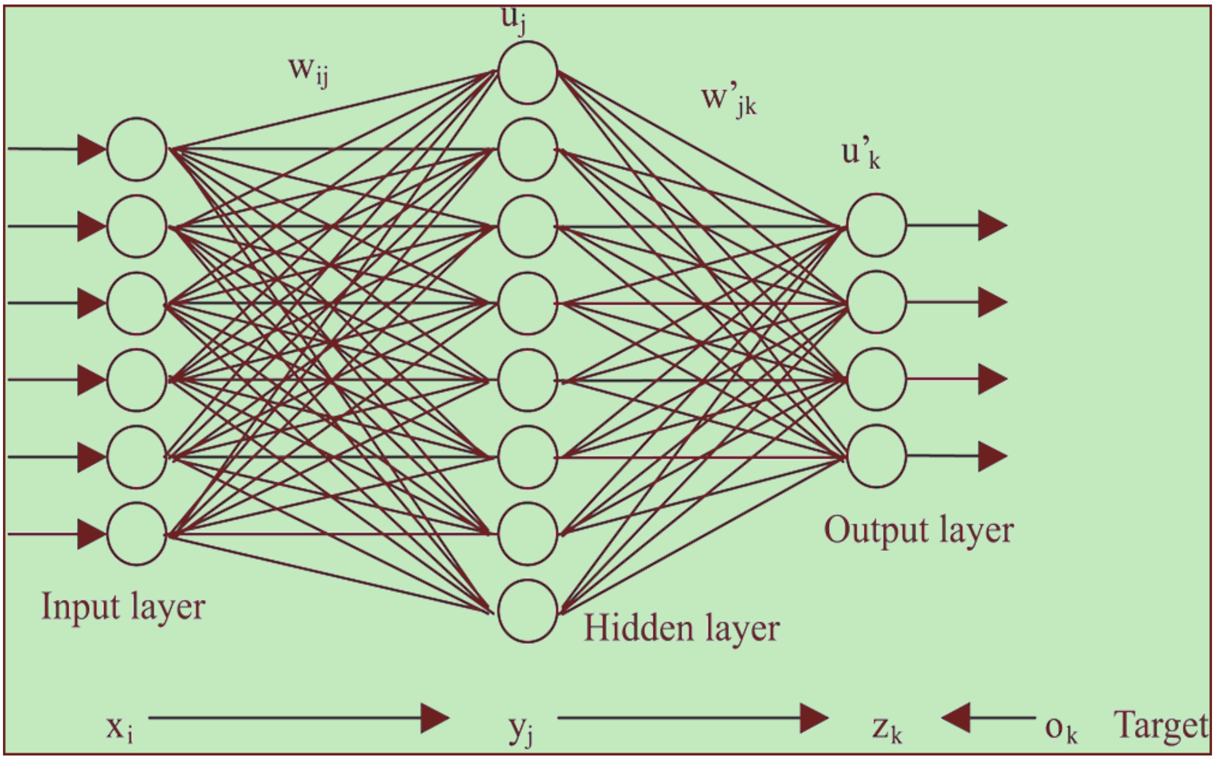


Figure 1: Adjacent Neural Network [18]

Adjacent neural networks are mainly used to clarify the training phase. Where a bunch of data are trained to give an idea about a specific domain. The inputs here would be images from our dataset, the middle layer would be characteristics of an image from the image set and the outputs would be tags for that image. Each and every single layer determines a particular characteristics for that particular image from the image data set. First, an example image will match a particular characteristics and then it will go for another if the first layer of characteristics matches. Furthermore, it will go for the next characteristics and then it will go for the next until each layer of characteristics are matched. Basically it looks at the pictures as a whole and take the major aspects of the image and determines tags based on all the tags connected to each other. To get more accurate values, we used Convolutional Neural networks. We used Convolutional Neural Networking to simplify the previous model we have already faced. Here we kind of break down the part of images and work our way through until we get variety of different tags that aren't necessarily related. To start with the Convolutional Neural Network we must go through four different definitions,

1. Local Receptive fields: A window on the input of pixels.
2. Feature Map: Mapping from input layer to hidden layer
3. Shared Weights: Positive or negative on a feature map.
4. Pooling: Simplifying the information from the feature maps.

Local receptive fields are basically a window on the input of pixels. For example let's pick an image of a person. Resize the image into NxN pixels. Selecting a particular amount of pixels form that input image or a window containing MxM pixel will be the local receptive field. Each receptive field will work as one individual neuron.

Each and every Local receptive field will be counted as a neuron. Each neurons at a certain level will create another pixel of layer by grouping by taking the neurons from the Local receptive field. Mapping from the input layer to create a new layer is called Feature Map. A feature map is a definition of one characteristic. Moreover the newly created layer will be called a hidden layer.

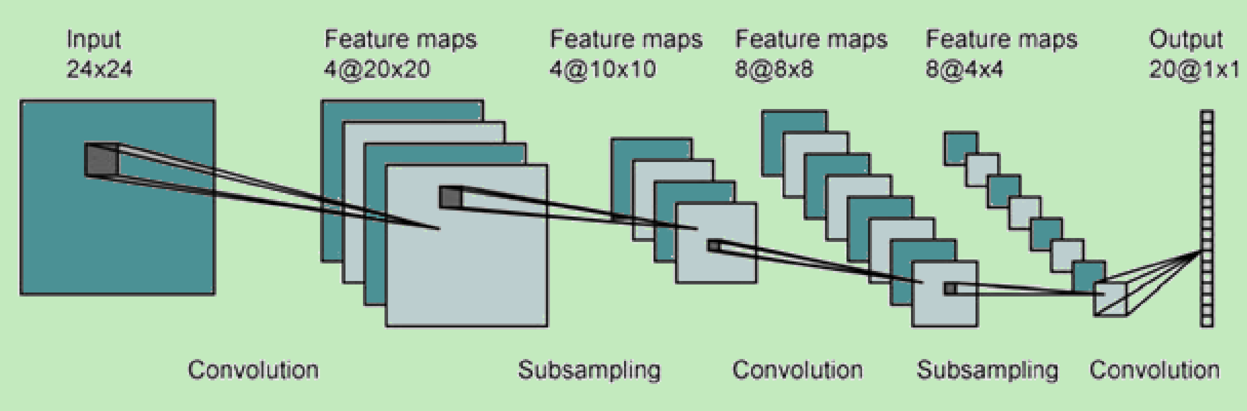


Figure 2: Convolutional Neural Networks [19]

Because of each feature map is one characteristic multiple hidden layers will be created. A complete hidden layer consists of several feature maps. Splitting the images into neurons, creating a local receptive field on those neurons and compressing those neurons into a feature map is called Convolution. We have done the exact process to detect abnormal behavior from a specific domain from a set of dynamic images taken from a CCTV footage.

The term Pooling takes the feature maps from the convolution and does the same thing. It basically makes a condensed feature map based on the feature maps. We basically do condensing the hidden neurons and then do that again for the second time then we keep breaking it down where we make feature maps and put it in the layer and then break that down and then condense it more until we find more and more detail of that input image. Mainly it's all about simplifying the information from the feature maps. We have to do the Pooling section separately for all the different characteristics for localizing things in an image.

In the overall process we can see the image is getting smaller and smaller. Because the image we are working with it's actually works really fast, significantly faster than adjacency because each characteristics we are following is basically following its own thread. So it is fully dependent on the actual size of the image it ends up being done within a short amount of time.

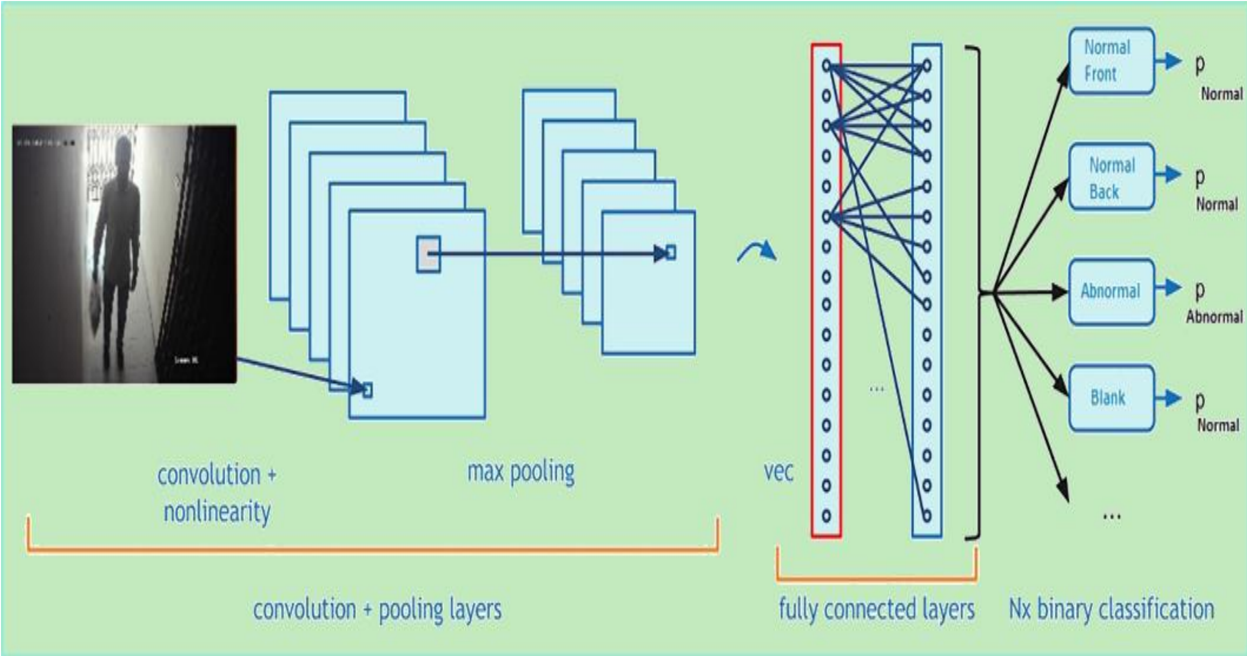


Figure 3: Convolutional neural networks with output class [20]

Convolution is mainly way easier to create neural networks because it's fast to train and multiple items outputs or tags can be found here. On the other hand adjacent neural network has no recognition of special structure but it is great and efficient for finding a single item. Whether Convolution is way more efficient in finding multiple characteristics from multiple images or items.

Chapter 4

System Implementation

Obtaining satisfactory results within limited amount of time vastly depends on how the system has been implemented. This requires proper planning and efficient use of resources available at hand. We have put our effort to utilize time and resources accordingly in building the system with the specific goals as our target.

4.1 System Implementation Details

To get our expected outcome, we used Convolutional Neural Network (CNN) architecture with our manually created Dynamic image dataset containing around 1000 images of different classes where images are taken from a CCTV footage (dynamic images) of a private property. After necessary augmentation images were put into training data folder and testing data folder of the system in file format of .jpg or .png. Images of the dataset belongs to normal behavior classes and abnormal class. Normal behavior classes contain images of front side, back side of a person, blank view which are normal and abnormal class contain images of people carrying unwanted objects which are unusual. For the first test the training images were 80% of the dataset and testing images were separated from the training images which were randomly selected covering 20% of the dataset. For the second test all training images belong to normal behavior classes were trained to test the testing images belong to abnormal behavior class. In order to implement our CNN model we used Theano [13] as the machine learning library and Keras [14] as the wrapper library. After augmentation and pre-processing our model of CNN was trained with different parameters along with optimizers like SGD [15], RMSprop [16], Adam [17] and later we used it in the tests to get various predictions. We reached to conclusion from the comparison of these results. The workflow, data divisions and samples of different classes are illustrated in the following figures and tables:

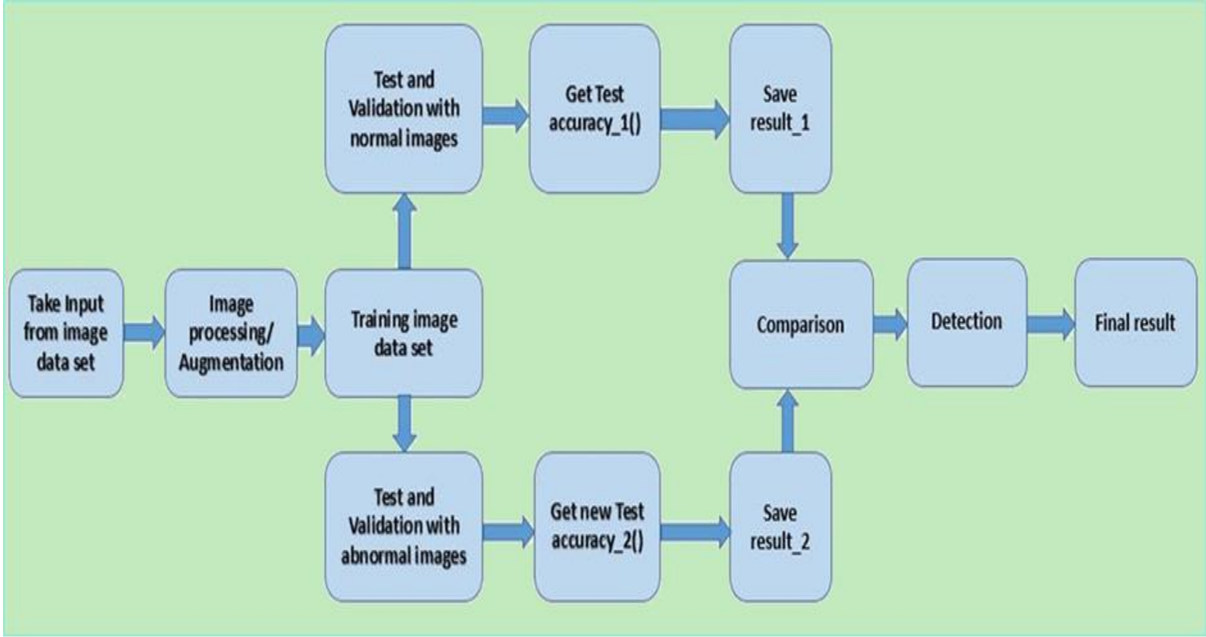


Figure 4: Work Flow

First of all, in our first approach we went for requirement analysis where we were unable to find proper image data set for our desired proposal. Therefore, we started building our own image data set, created from a CCTV footage from a private property. Our main purpose was to create a specific domain for normal and abnormal scenario. For that private property, we set the normal and abnormal behavior as,

1. One person can only enter without any objects like, bag, or any other accessories which is visible at a time.
2. One can go out without any objects like bag, or any other accessories which is visible at a time.
3. The above two scenario will represent the normal behavior classes.
4. Images with abnormal behavior will contain, a person with a bag, any weapon, accessories or any other objects.
5. More than one person at a time under the surveillance will also represent the abnormal behavior.
6. Images with abnormal postures will also be considered as an abnormal class.

The above scenario will be enough for us to implement our proposed work where abnormal behavior can be found very efficiently.

First of all, two different experiment was done to create and evaluate in between. To create a local maximum point or the local maxima we started with training images and split them. After that we tested those split images with the trained images. Therefore we got approximately 99% accuracy as in our local maximum point from the first experiment.

For the second experiment, we tested with a number of abnormal class images (20%) with the images which was trained before. As a result, the local maxima reduced and went under the threshold value. Where the accuracy difference between the first experiment and the second was more than 30%. Which is because of the images by which we started testing. Moreover, the reduced value under the threshold limit determined the amount of abnormality the image contains. Changes in graphs will send a message with the percentage of abnormality detected.

Dataset		
	Training Data	**Testing Data**
Data division for first test	Random 80%(≈640 images)	Random 20%(≈160 images)
Data division for second test	All normal behavior images	All abnormal behavior images
Total Images	1000	

Table 1: Dataset and data divisions for tests

In the first experiment we trained 3 different image classes which represents normal behavior. After that 20% split images was used to test and get the local maxima.

In the second experiment, a single class with abnormal behavior was used for testing which gave us the value which was below the threshold level and gave us the graph along with the message which detects and prints the amount of abnormality the image contains and detects efficiently the behavior which is not normal.

	Class Types			
	Normal Behavior			**Abnormal Behavior**
Sub-Type	**Front**	**Back**	**Blank**	**Abnormal**
No. of Classes	3			1

Table 2: Classes used for tests

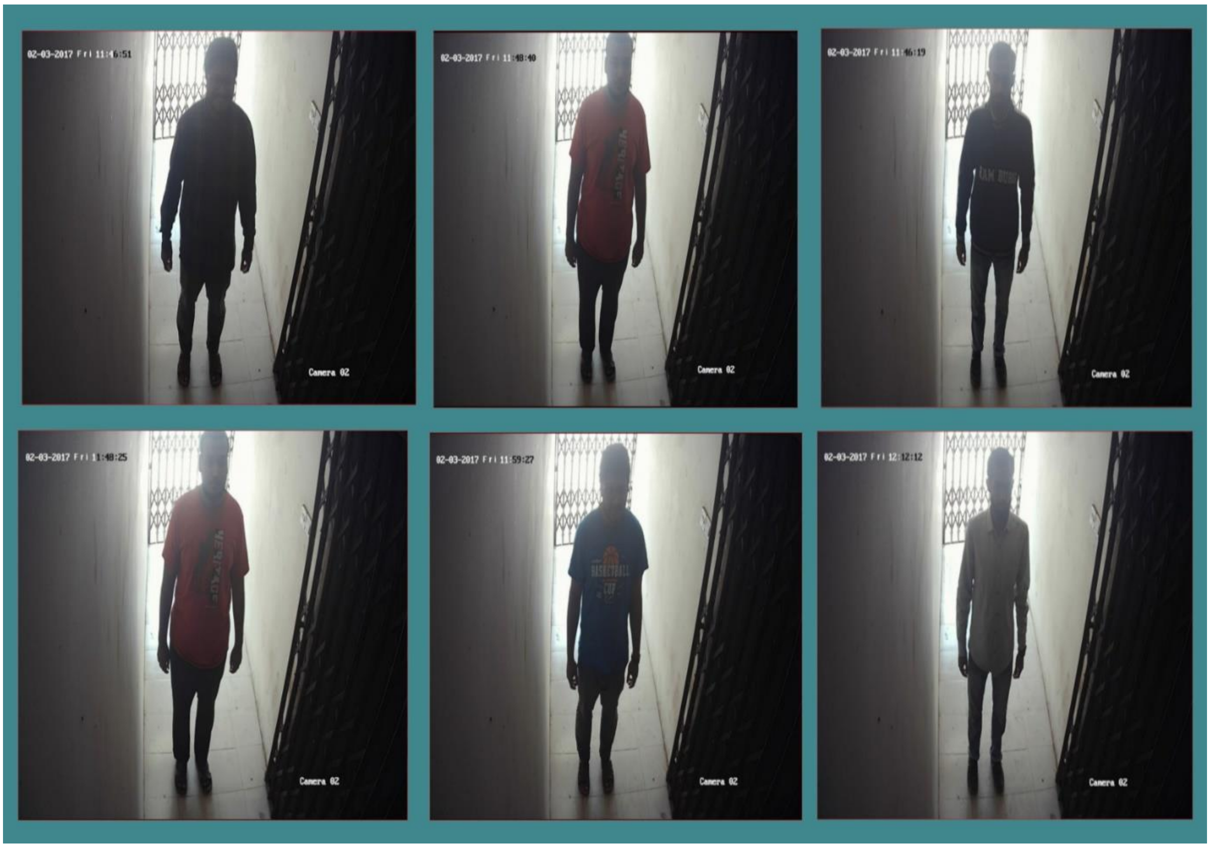


Figure 5: Image samples of Normal Behavior Front View Type

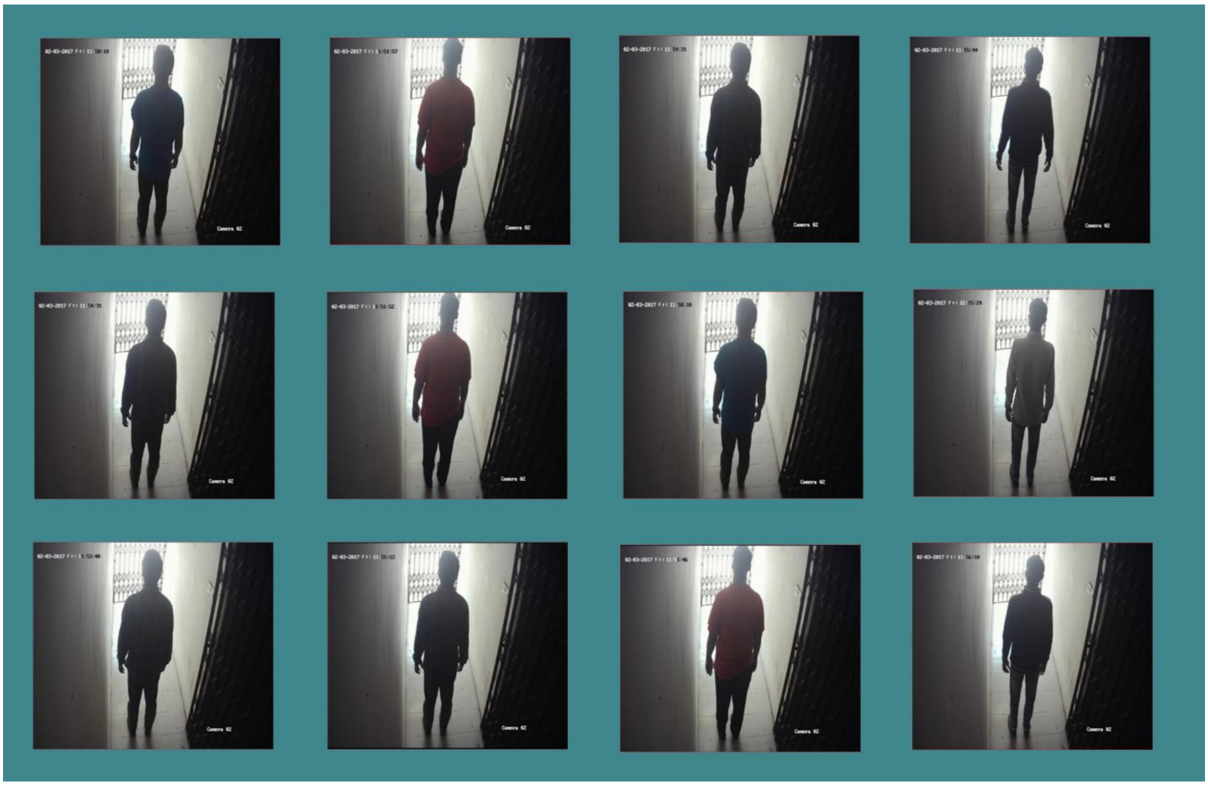


Figure 6: Image samples of Normal Behavior Back View Type

Figure 7: Image samples of Abnormal Behavior Type

4.2 Hardware and Tools

This system model is a complex deep convolutional neural network which requires high configuration GPU for faster computation. Despite that this model can be trained and tested with low configuration CPU (only) and GPU. We used computer system having high end graphics card with NVIDIA GPU (NVIDIA GTX 1060 along with cuDNN installed) which has 6GB video memory, 8GB RAM, Core i5 processor to implement the system. To conduct our work we have used:

- Windows 10 (64 bit) OS platform
- Visual Studio 2015 (C/C++ compiler part)
- Anaconda 3 (64 bit) with Python 3.5 (Open Data Science Platform by Python with useful libraries)
- CUDA 8 (64 bit) (Compiler, math libraries for NVIDIA GPUs)
- MinGW-w64

- Theano 0.9.0 (Math expression compiler)
- Keras (>1.0.0) (Wrapper library for deep learning on top of Theano)
- cuDNN 5.1-CUDA Deep Neural Network library (cuDNN) is a GPU-accelerated library build for deep neural networks.
- Numpy 1.12.1-Array processing for numbers, strings, records and objects
- Matplotlib 2.0.0-Python 2d plotting library
- scikit-learn 0.18.1– tools for data mining and data analysis in Python
- scipy 0.19.0- Scientific library for python
- Pillow 3.3.1- PIL(Python imaging library) fork
- Spyder 3.0.0-Scientific python development environment

NumPy is the fundamental package for scientific computing in Python programming language. It basically is a Python library which provides a multidimensional array object and an assortment of routines for fast operations on arrays, including logical, mathematical, shape manipulation, sorting, selecting, discrete Fourier transforms, basic linear algebra, basic statistical operations, and random simulation more. In our work, the image Dataset was taken as numpy array for processing.

Figure 8: NumPy

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and Ipython shell, web application servers and four graphical user interface toolkits.

Figure 9: Matplotlib

Scikit-learn was used for tools in order to perform actions like shuffle, split. In our first experiment the cross validation and the split part was done using Scikit-learn.

Figure 10: scikit-learn– tools for data mining and data analysis in Python

Pillow used as PIL fork which manages images. Python Imaging Library (abbreviated as PIL) (in newer versions known as Pillow) is a free library for the Python programming language that adds support for opening, manipulating, and saving many different image file formats.

Figure 11: Pillow – PIL (Python imaging library) fork

Theano is a Python library that lets you to define, optimize, and evaluate mathematical expressions, especially ones with multi-dimensional arrays. Using Theano it is possible to attain speeds rivaling hand-crafted C implementations for problems involving large amounts of data. It can also surpass C on a CPU by many orders of magnitude by taking advantage of recent GPUs.

Figure 12: Theano- Math expression compiler

We used Keras as the wrapper library[14]. Keras is a high-level neural networks API, written in Python and capable of running on top of Theano. It was developed with a focus on enabling fast experimentation. Being able to go from idea to result with the least possible delay is key to doing good research.

- Allows for easy and fast prototyping (through user friendliness, modularity, and extensibility).
- Supports both convolutional networks and recurrent networks, as well as combinations of the two.
- Runs seamlessly on CPU and GPU.

Figure 13: Keras- Wrapper library for deep learning

All programming and the system was carried out using python 3.5.

Figure 14: Python 3.5- programming language

Anaconda 3 was used as platform to implement the system with necessary Python libraries.

Figure 15: Anaconda 3-Open Data Science Platform by Python

Spyder 3 IDE was used for running the python programs along with Anaconda 3.

Figure 16: Spyder 3.0.0-Scientific python development environment

CUDA Deep Neural Network library (cuDNN) was used to train the CNN using NVIDIA GPU.

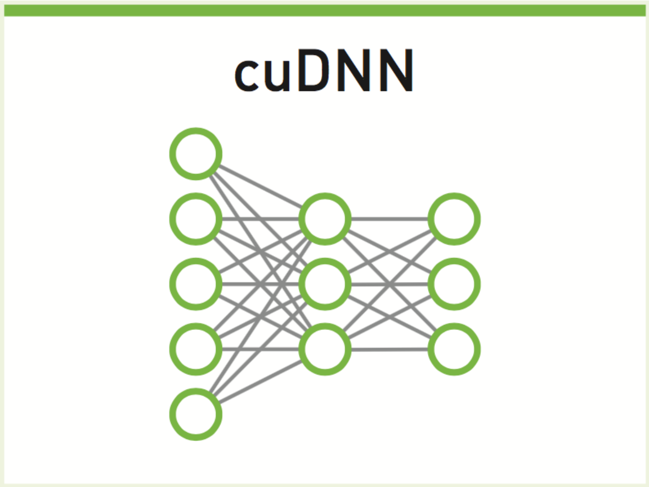


Figure 17: cuDNN-CUDA Deep Neural Network library

CUDA 8 toolkit was used for utilizing NVIDIA GPU. In our work we used NVIDIA GTX 1060 with the VRAM of 6GB along with the memory interface width of 192-bit. Which is necessary for our work. VRAM and memory interface width helps a training phase work faster.

Figure 18: CUDA 8 - Compiler, math libraries for NVIDIA GPUs

Chapter 5

Experimentation and Result Analysis

In order to evaluate our system we conducted two experiments. Our dataset was built using captured frames of CCTV video. For both experiments we needed preprocessing and augmentation of data for training and testing.

5.1 Preprocessing and Data Augmentation

Preprocessing includes resizing source images, converting to gray scale images, randomly shuffling data. Augmentation of images used for training and testing has been done by the following arguments and functions:

- Rescale: Reduces processing load by scaling RGB coefficient factors. We used 1/255 for our experiments.
- Shuffle: Shuffles the data if True.
- Shear range: Indicates shear intensity where angle is in counter-clockwise direction in radians. Parameter in Float.
- Zoom range: Method zooms in random areas of images. Parameter is Float or [lower, upper].
- Horizontal flip: Method for randomly flipping input images horizontally. Takes Boolean parameter.

5.2 Training Procedures

We trained our convolutional neural network using RMSProp [17] optimizer where learning rate of 0.001 was set along with decay rate of 0.0 for proper learning considering amount of data and time. For better convergence of the network we set momentum update as 0.9. And loss calculation of the network was done using binary cross-entropy. Preprocessing arguments with respective values are given below:

- Rescale = 1./255,
- Shear range = 0.2,
- Zoom range = 0.2,
- Horizontal flip = True.

5.3 Testing Procedures

For testing we used images from the dataset which contains abnormal behavior to pass through our trained network to predict abnormality. After final output layer we used softmax function to calculate the loss function. Only rescaling was done for source images containing abnormal behavior to get practical and fast detection. Without enhancement the system was tested for accuracy and reliability. Preprocessing argument given below:

- Rescale = 1./255

5.4 First Experiment and Results

In this experiment dataset of all normal classes were randomly split into 80% for training data and 20% for testing data. Our convolutional model has different layers, filters containing all necessary functions. The convolutional network provided loss and accuracy of train and test data. We saved the model, weights and results for comparison during the second experiment. We are using 32 filters, where it will first show us the first feature map of the first convolution layer and how our network has produced it. After getting one single input and after convolution the CNN has produced the first feature map of the first convolution layer. After that we showed the output of the output of the first activation layer. Moreover, layer 2 will show us the second convolution layer. In the second convolution layer the image size reduces and gives out put after the second convolution but the filters will be the same as always. The max pooling part was also shown in the layer 4. Here the size will again reduce and will show the activation layer after max pooling.

Layer visualizations of different convolutional layers have been listed below:

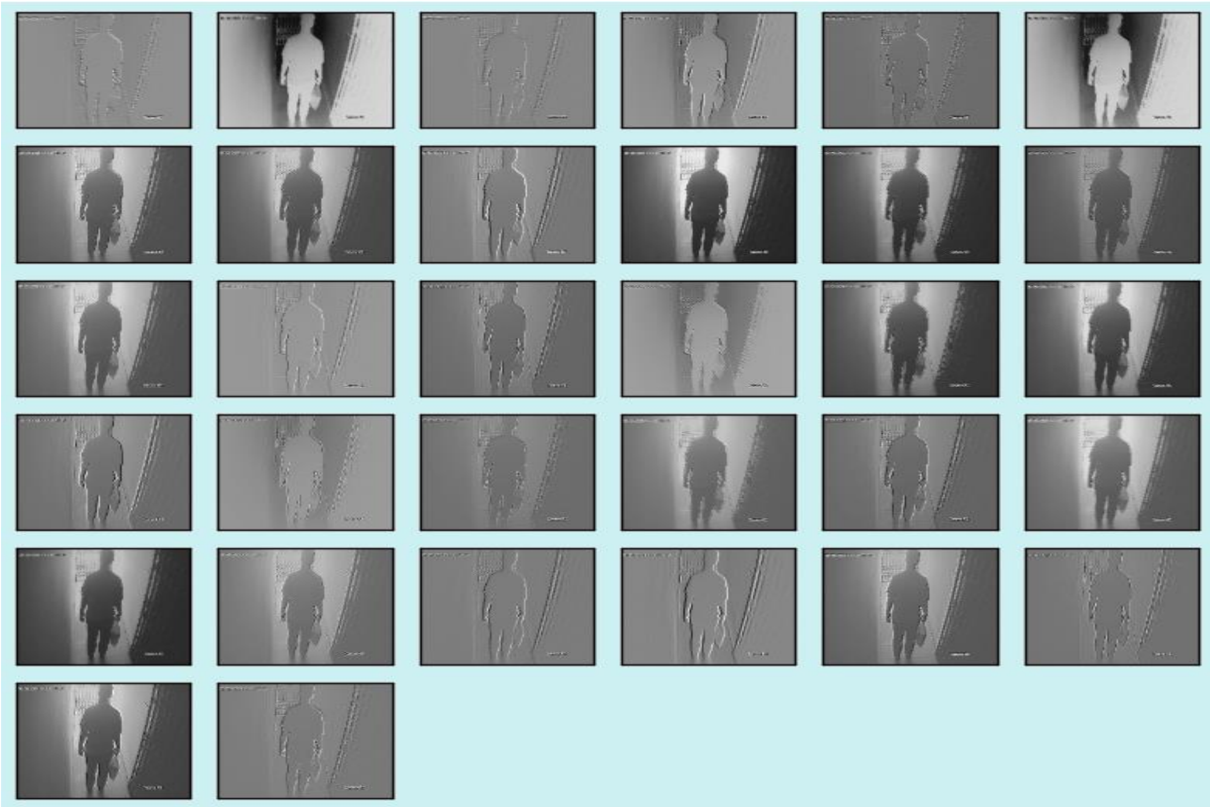


Figure 19: Layer visualization of convolutional layer 0

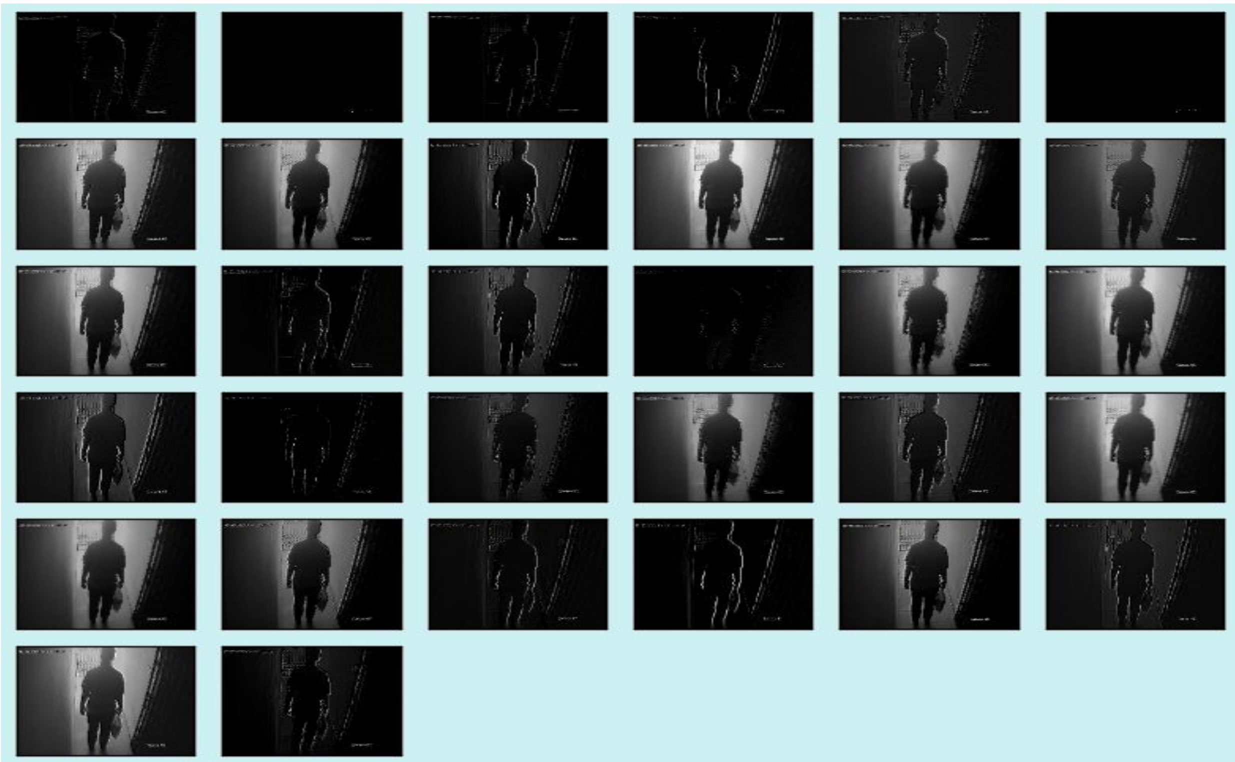


Figure 20: Layer visualization of convolutional layer 1

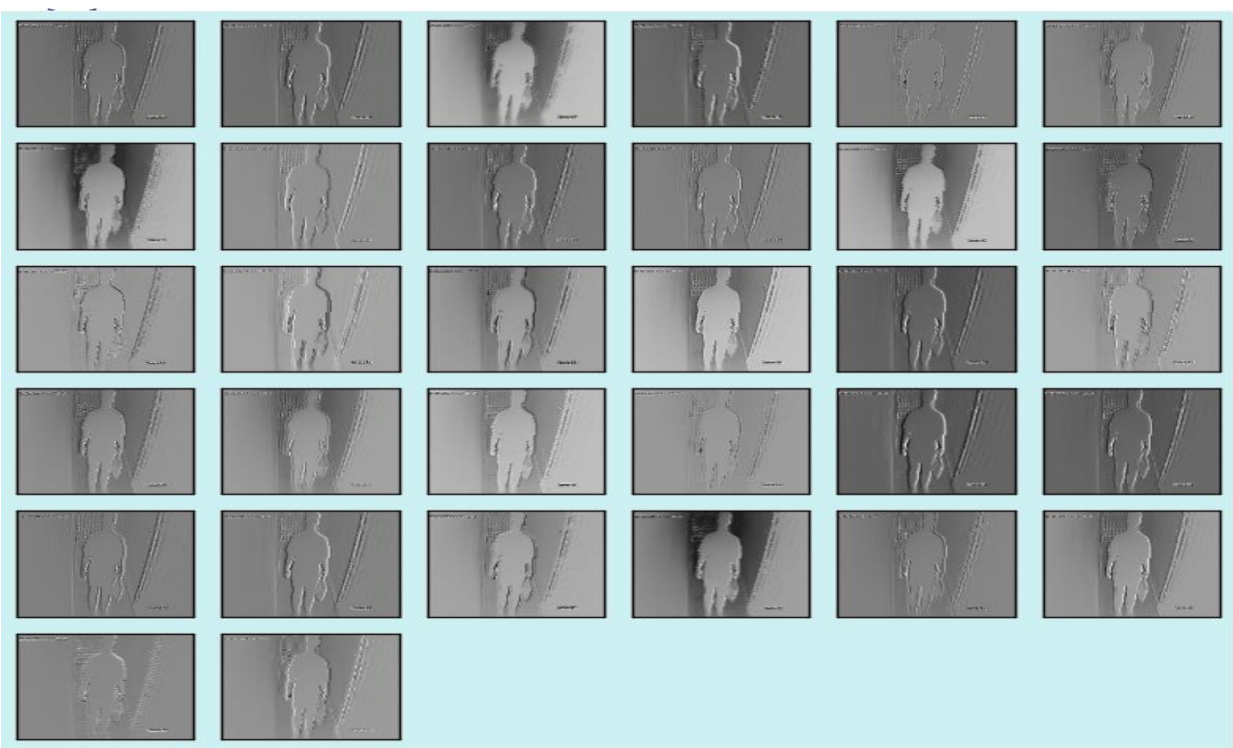


Figure 21: Layer visualization of convolutional layer 2

Figure 22: Layer visualization of convolutional layer 3

Figure 23: Layer visualization of convolutional layer 4

For normal behavior as the number of epoch increases the loss decreases and accuracy increases for both training and validation phase. Graphs illustrating results have been listed below for 10 epochs and 20 epochs respectively:

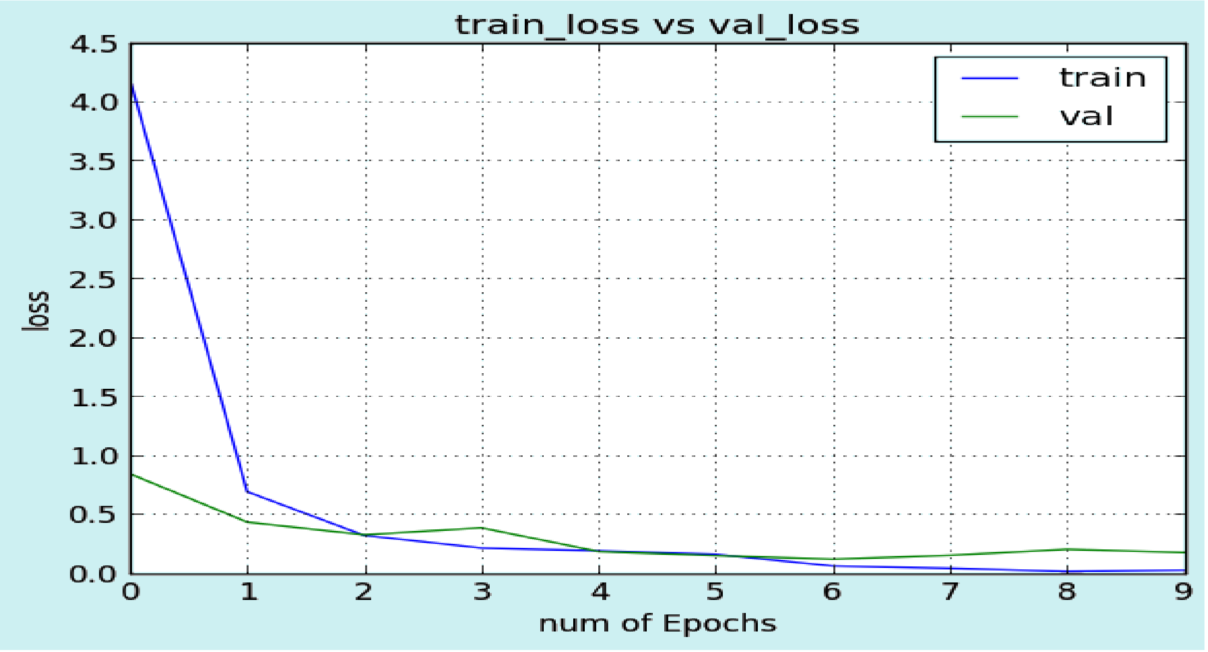


Figure 24: Train-loss vs. validation-loss of normal behavior detection for 10 epochs

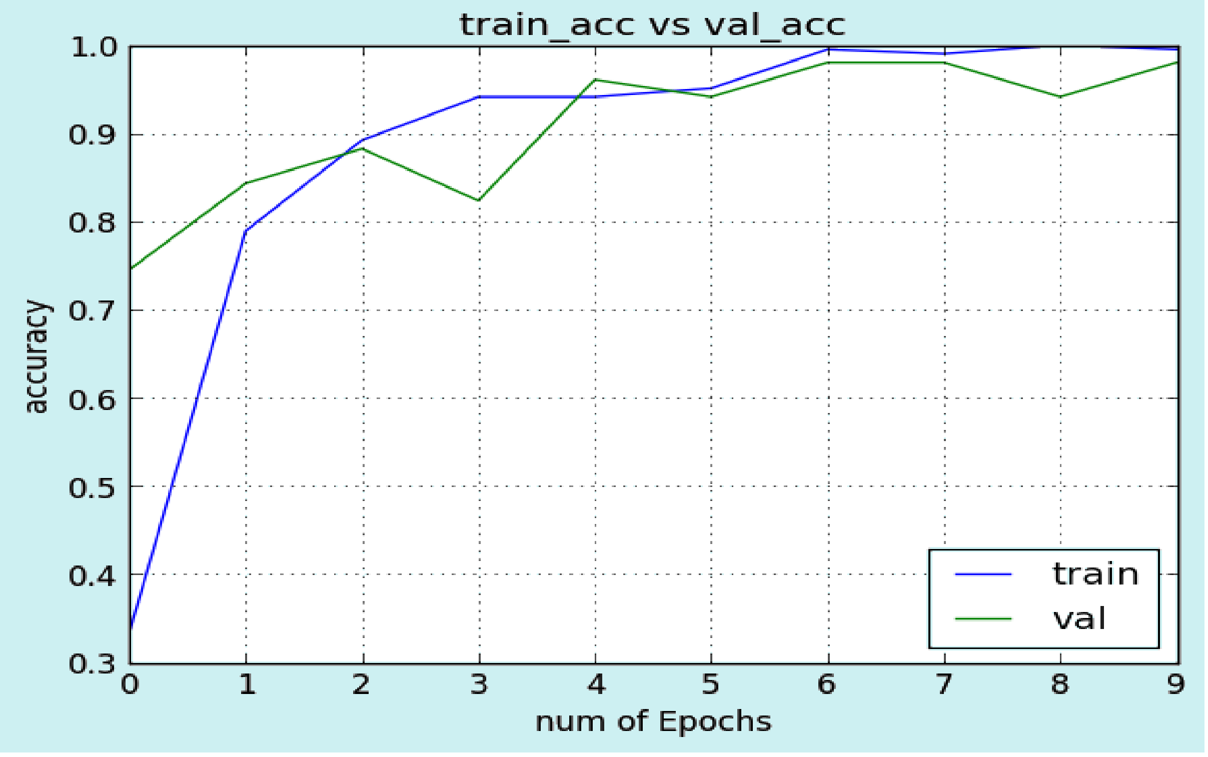


Figure 25: Train-accuracy vs. validation- accuracy of normal behavior detection for 10 epochs

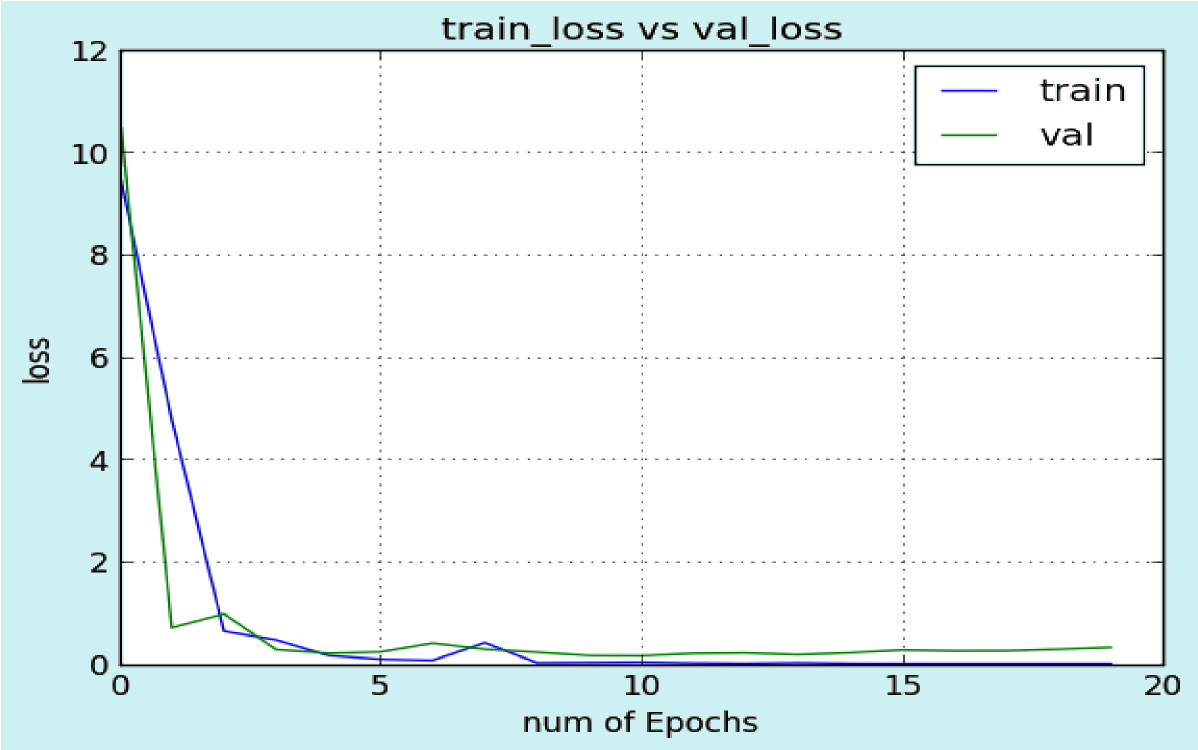


Figure 26: Train-loss vs. validation-loss of normal behavior detection for 20 epochs

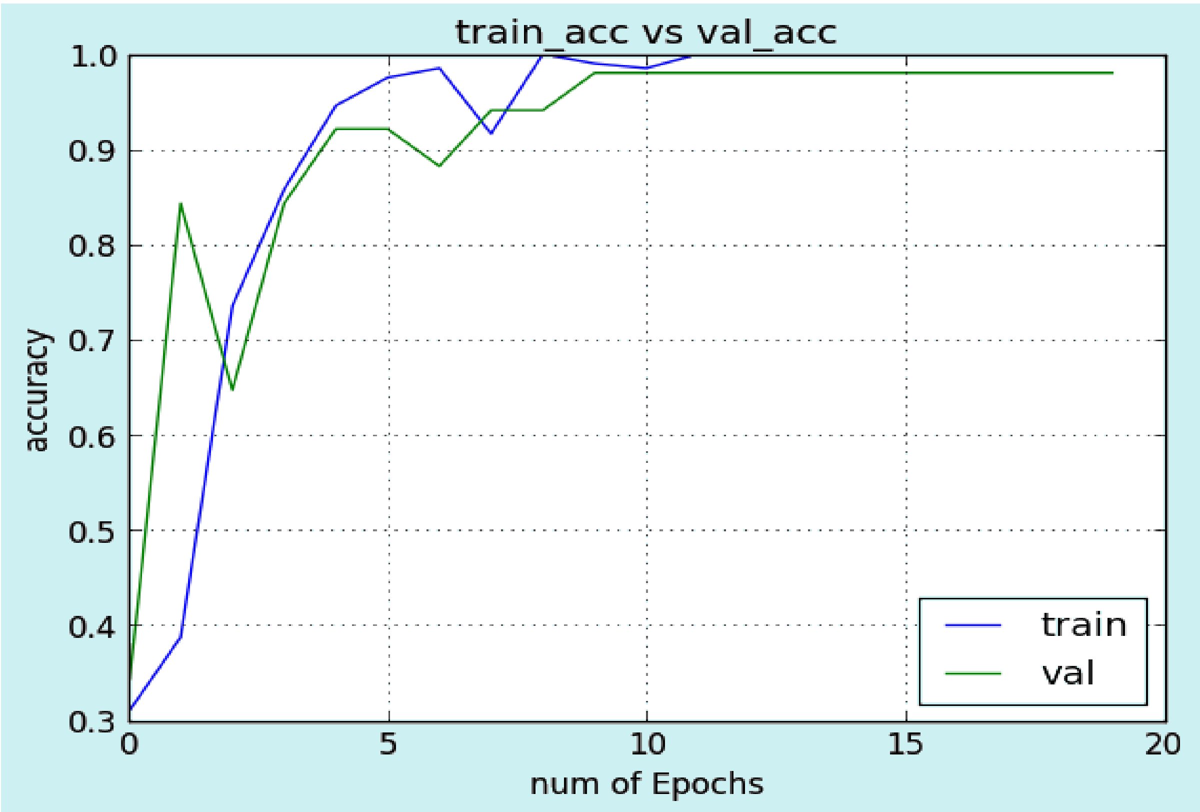
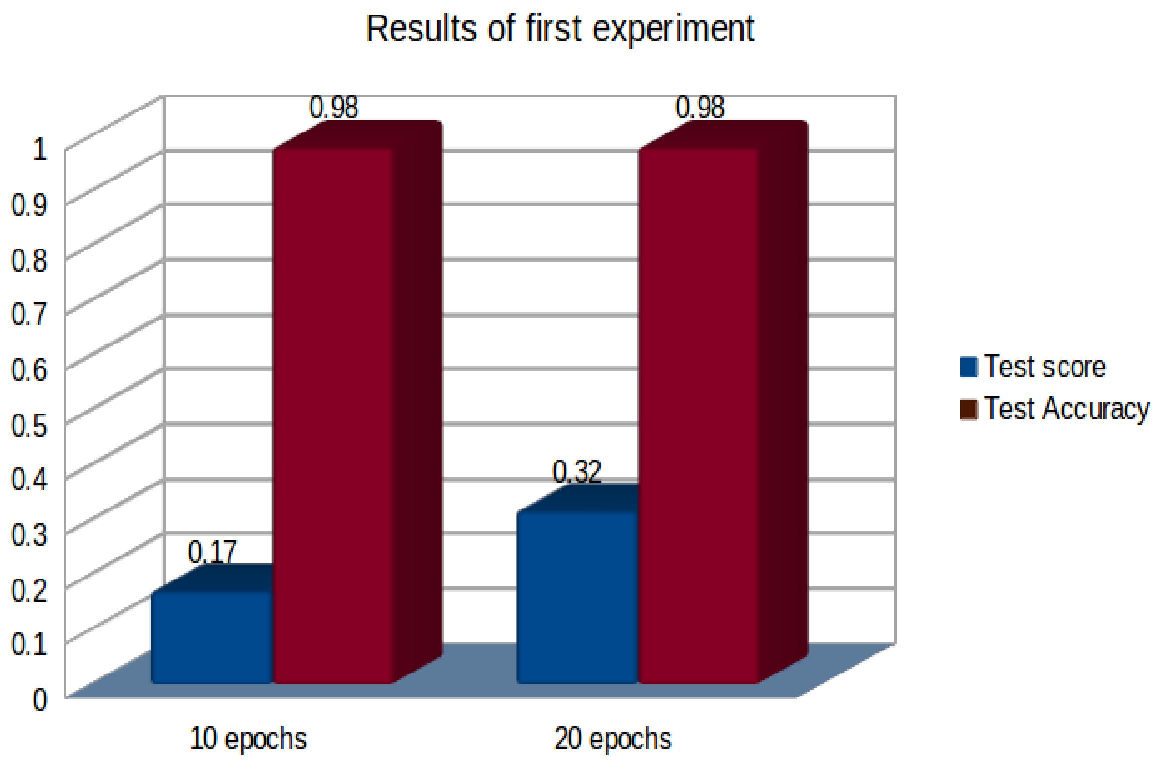


Figure 27: Train-accuracy vs. validation- accuracy of normal behavior detection for 20 epochs

For first experiment we got 98% accuracy for normal behavior detection during validation phase. Results are shown in the table below ("Test score" indicates how well trained the neural network is.):

First Experiment		
No. of Epochs	Training Phase	Validation Phase
	Test Score	Test Accuracy
10	0.1708	0.9804
20	0.3174	0.9804

Table 3: Results of first experiment



Graph 1: Results of first experiment

5.5 Second Experiment and Results

In this experiment images of all normal behavior classes from dataset were added for training data and all abnormal behavior images were added for testing data. Our convolutional network provided loss and accuracy of train and test data. We saved the model, weights and results. From comparison of second results with first results the system concluded that the images contains abnormality. A certain deviation percentage or accuracy percentage below threshold level indicates abnormal behavior has been detected.

For abnormal behavior as the number of epoch increases the loss decreases and accuracy increases for both training phase. And it rapidly reaches similar accuracy level of first experiment. On the other hand, during validation phase loss becomes negative. Compared to training accuracy validation accuracy percentages are always very small too. Graphs illustrating results have been shown below for 10 epochs and 20 epochs respectively:

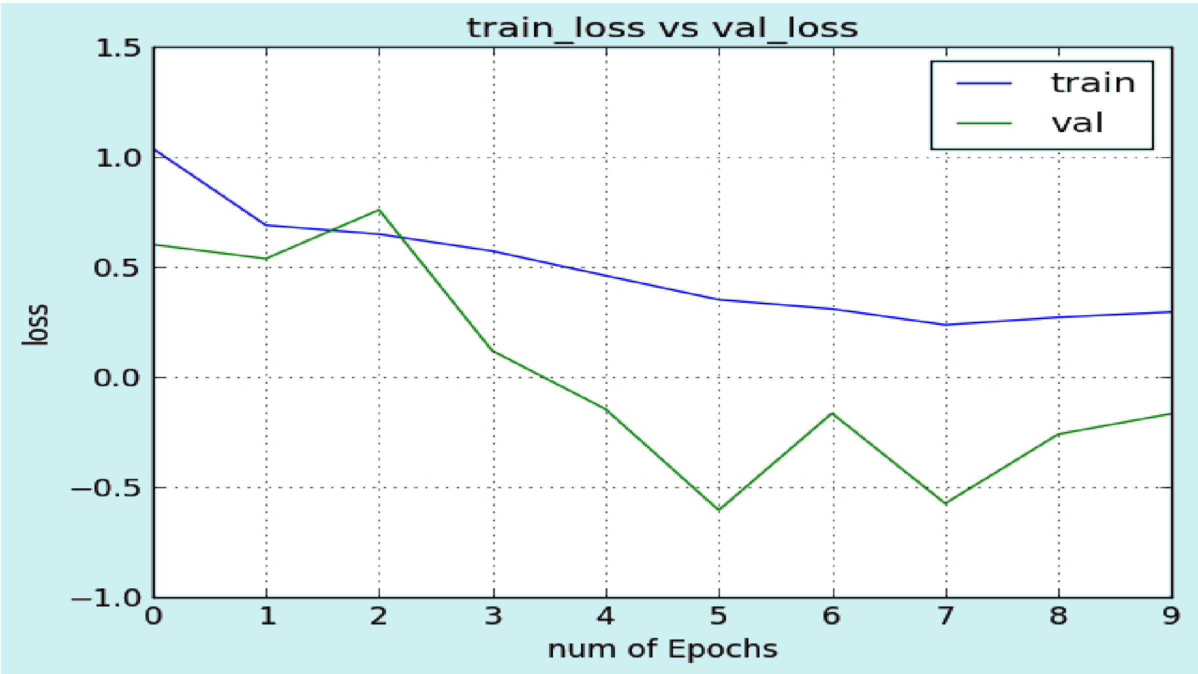


Figure 28: Train-loss vs. validation-loss of abnormal behavior detection for 10 epochs

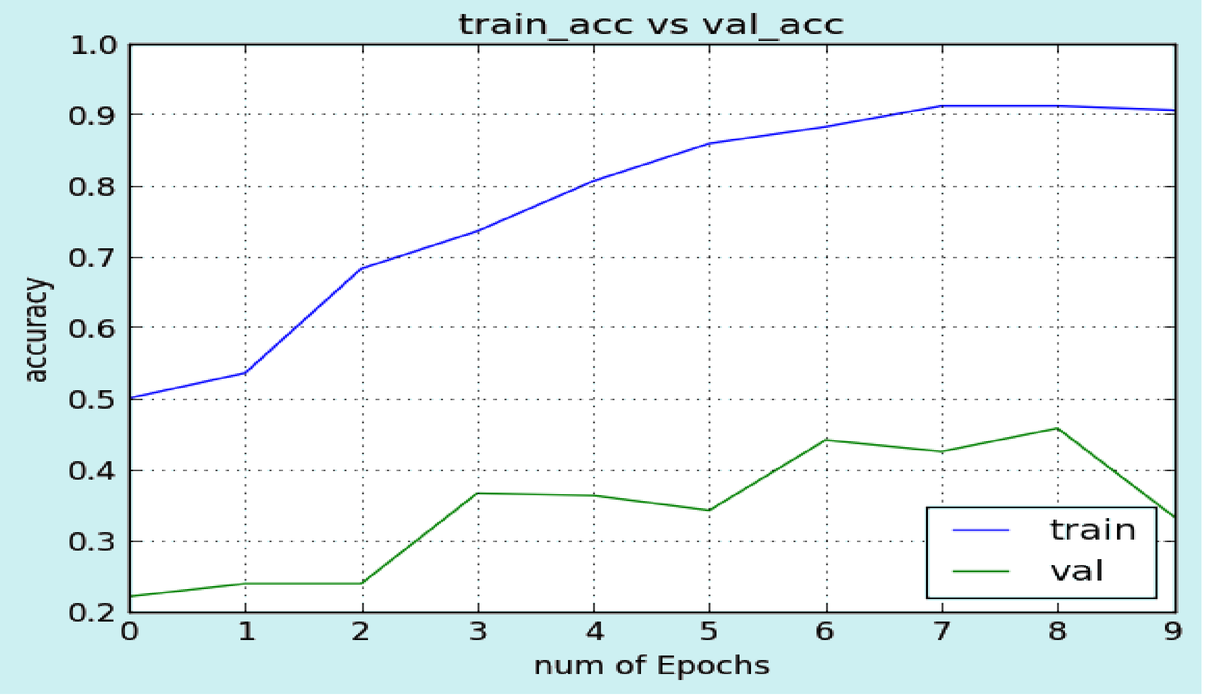


Figure 29: Train-accuracy vs. validation- accuracy of abnormal behavior detection for 10 epochs

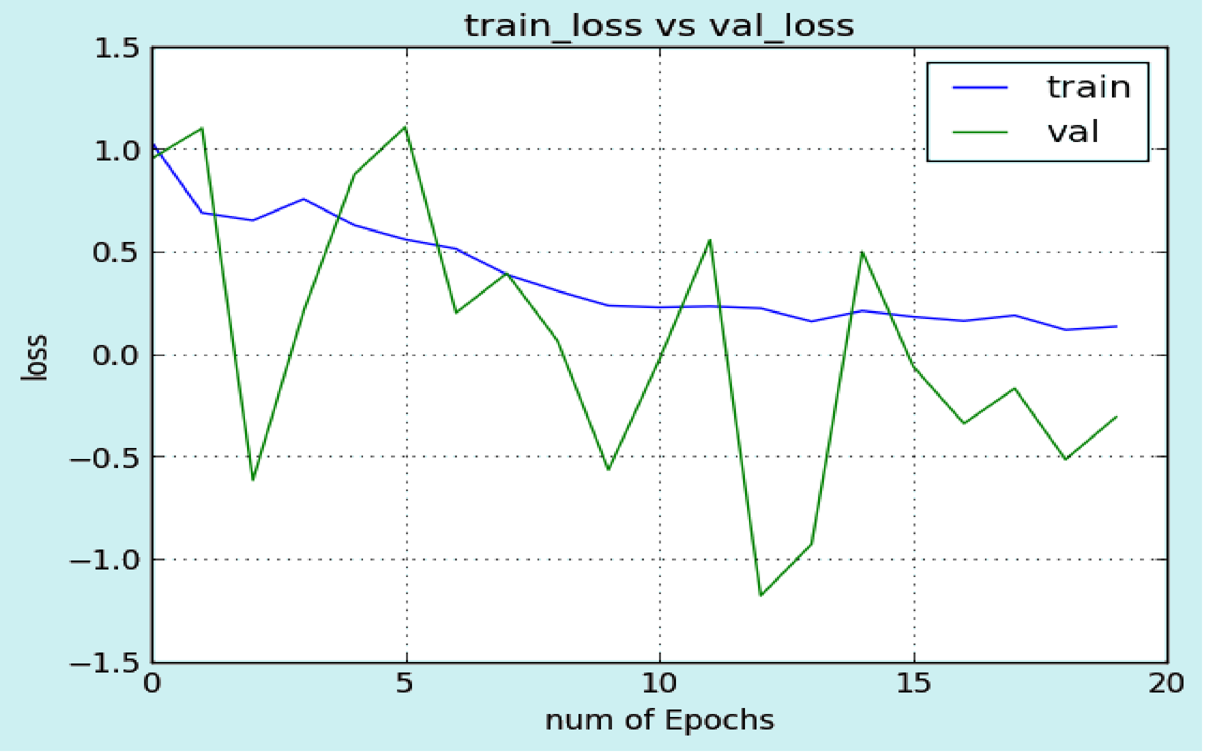


Figure 30: Train-loss vs. validation-loss of abnormal behavior detection for 20 epochs

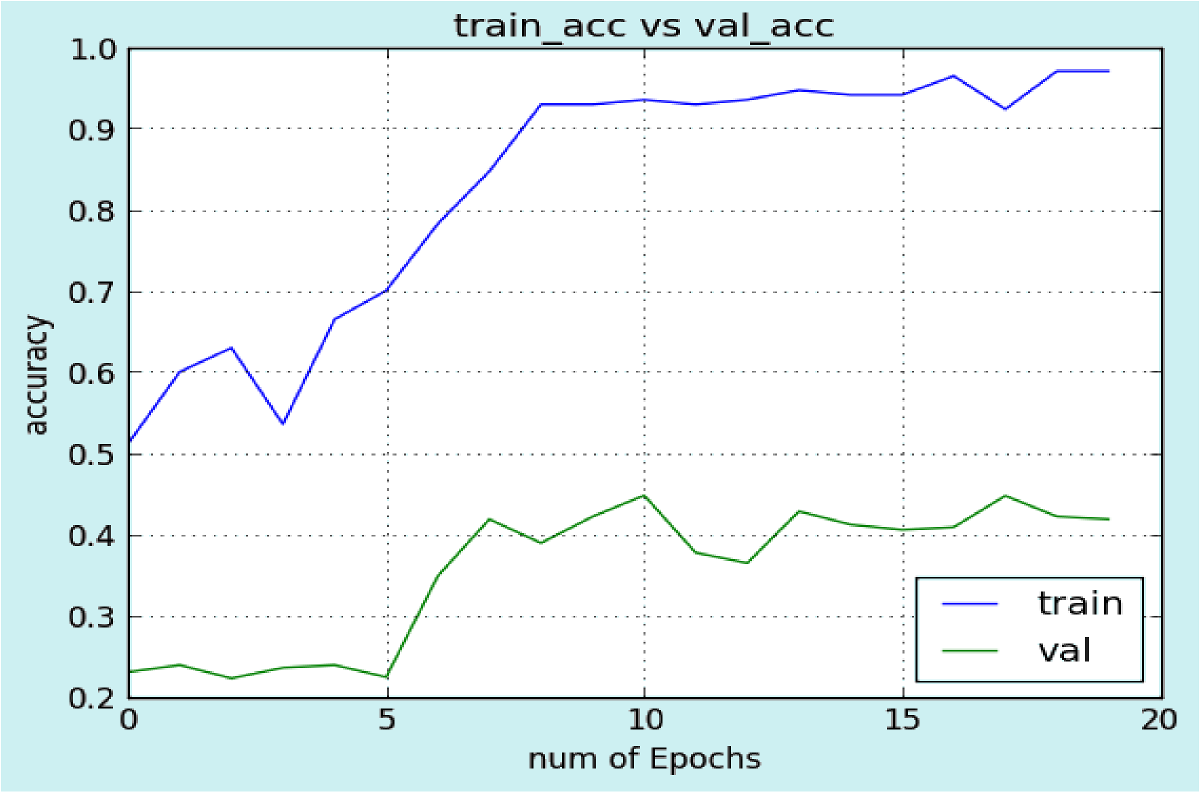
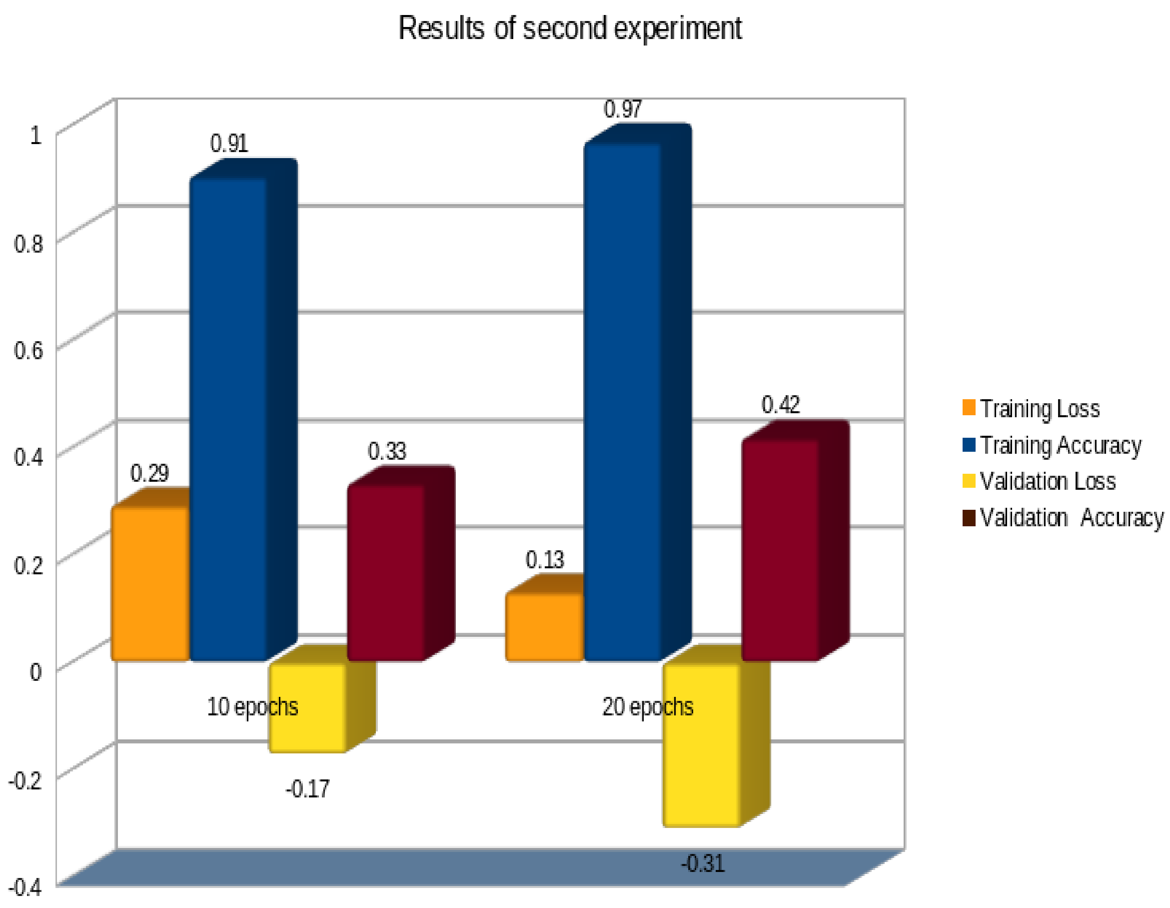


Figure 31: Train-accuracy vs. validation- accuracy of abnormal behavior detection for 20 epochs

For second experiment we got 33% and 42% accuracy for 10 epochs and 20 epochs respectively for abnormal behavior detection during validation phase which is far below the training accuracy percentage. Hence, presence of abnormality is evident. Results are shown in the table below

Second Experiment				
No. of Epochs	Training Phase		Validation Phase	
	Loss	Accuracy	Loss	Accuracy
10	0.2920	0.9059	-0.1703	0.3333
20	0.1312	0.9706	-0.3085	0.4183

Table 4: Results of second experiment



Graph 2: Results of second experiment

Our system compared results of both experiment and predicted abnormality since there is a significant difference between two accuracy levels. A threshold level can also be enabled for detection of abnormality for the trained convolutional neural network. A deviation of 3% (manually set) from normal behavior detection accuracy will be identified as abnormal behavior which is a dependable prediction for ensuring security.

Chapter 6

Conclusion

The focus of this thesis is to develop a technique by which we will be able to detect abnormal behavior efficiently. It is clear that different techniques balance certain trade-offs between computational complexity, speed and accuracy of recognition and overall practicality and ease-of-use. Convolution is now a very good efficient and developing platform for image and video classification. Classification is the basic criteria of convolutional neural networks. Rather going for direct algorithm implementation which is bit costly and inefficient we used convolution to detect abnormal behavior. The world is going through a very bad situation where security is the main priority for all. Crime, terrorism, theft and more acts are increasing with the unstable situation of the world. A simple attempt to ensure the security will not be complete but will be enough for those who is in need of the security from the dangers of the unstable world. System with our work included is easy and efficient and can be implemented easily in any environment. Moreover, our system can be updated inexpensively anytime. Furthermore, our proposed system will be adaptable in any dynamic environment. If an environment changes for our system it will not be a problem to reinitiate the system again. We tried to ensure maximum security with the help of the recent technology along with the help of machine learning platform. We strongly hope our combined experiments will provide best accuracy in normal identification and abnormal behavior detection by using Convolutional neural networks.

6.1 Future Work

We have future improvement plan regarding this project. As the availability of different reliable datasets is increasing and thus more training on images of different scenarios will significantly increase the efficiency of our system in detection of abnormal behavior. Security is the most important need of our daily life. Our project is able to give any sort of protection and in future we will try to increase the percentage of detection accuracy. Further enhancement and moderation can be brought in the architecture of our system. We would like to improve that side of our project.

- Advanced security system for Private property, Bank, Office building, Airport etc.
- Real time video analysis for more accurate detection.
- Get more accurate results by using pre-trained models and data Segmentation.

- Include predictive algorithm to detect and predict abnormality.
- Smart Security system implementation.
- Government using this system to advance the national security.
- Developing an application for mobile so that it can be used by everyone.

References

[1] J. Krause, T. Gebru, J. Deng, L.-J. Li, and L. Fei-Fei, "Learning Features and Parts for Fine-Grained Recognition," 2014 22nd International Conference on Pattern Recognition, 2014.

[2] R. Mehran, A. Oyama, and M. Shah, "Abnormal crowd behavior detection using social force model," in Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2009.

[3] M. D. Breitenstein, H. Grabner, and L. V. Gool, "Hunting Nessie – real-time abnormality detection from webcams," in IEEE International Workshop on Visual Surveillance, 2009.

[4] V. Mahadevan, W. Li, V. Bhalodia, and N. Vasconcelos. Anomaly detection in crowded scenes. In *CVPR*, 2010.

[5] Baum, L. E.; Petrie, T.; Soules, G. & Weiss, N. (1970). A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. Annals of Mathematical Statistics, vol. 41, 1970.

[6] Baum, L. E. (1972). An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes, Inequalities, vol. 3, pp. 1-8, 1972.

[7] Bicego, M.; Castellani, U. & Murino, V. (2003b). Using hidden markov models and wavelets for face recognition. Proceedings of Image Analysis and Processing 2003. 12th International Conference on, pp. 52–56, 2003.

[8] LeCun, Y. (1989). Generalization and network design strategies. Connectionism in perspective on, pp. 143-155, 1989.

[9] LeCun, Y.; Boser, B. (1990). Handwritten digit recognition with a back-propagation network. Advances in neural information processing systems, 1990.

[10] J. Krause, M. Stark, J. Deng, and L. Fei-Fei, (2013). "3D Object Representations for Fine-Grained Categorization". 2013 IEEE International Conference on Computer Vision Workshops, 2013.

[11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[12] He, K.; Zhang, X.; Ren, S. & Sun, J. (2015). Deep Residual Learning for Image Recognition. arXiv:1512.03385v1 [cs.CV], 2015.

[13] Bergstra, J.; O. Breuleux; F. Bastien; P. Lamblin; R. Pascanu; G. Desjardins; J. Turian; D. Warde-Farley; Y. Bengio (30 June 2010). Theano: A CPU and GPU Math Expression Compiler. Proceedings of the Python for Scientific Computing Conference (SciPy) 2010.

[14] Keras: Deep Learning library for Theano and TensorFlow. (2017, April 10). Retrieved from https://keras.io/
[15] L. Bottou, "Large-Scale Machine Learning with Stochastic Gradient Descent," Proceedings of COMPSTAT'2010, pp. 177–186, 2010.L. Bottou, "Large-Scale Machine Learning with Stochastic Gradient Descent," Proceedings of COMPSTAT'2010, pp. 177–186, 2010.
[16] Tieleman, T. and Hinton, G. Lecture 6.5 - RMSProp, COURSERA: Neural Networks for Machine Learning.Technical report, 2012.
[17] Ba, J., & Kingma, D.P. (2014). Adam: A Method for Stochastic Optimization. CoRR, abs/1412.6980.
[18] Minimal Data Science #3: Handwritten Digit Recognition with Convolutional Neural Network. (2017, April 14). Retrieved from http://lenguyenthedat.com/minimal-data-science-3-mnist-neuralnet/
[19] Why Convolutional Neural Networks. (2017, April 14). Retrieved from https://www.linkedin.com/pulse/why-convolutioanl-neural-networks-shamane-siriwardhana
[20] A Beginner's Guide To Understanding Convolutional Neural Networks Part 1. (2017, April 14). Retrieved from http://www.kdnuggets.com/2016/09/beginners-guide-understanding-convolutional-neural-networks-part-1.html