Grenze International Journal of Engineering and Technology, July Issue

Artificial Intelligence based Video Monitoring System for Security Applications

Abhinav Kumar Mallick[1], Animesh Verma[2], Utkarsh Sahay[3], Kumar Shubham[4] and Jayanna H S[5]
[1-5] Dept. of Information Science & Engineering, Siddaganga Institute of Technology, Tumakuru, Karnataka, India
[1-4]Email: {abhinavmallick831, vanimesh007, kumar.1si16cs132, utkarsh.sahay}@gmaill.com
[5]Email: jayannahs@sit.ac.in

***Abstract*—In this paper, an artificial intelligence based video monitoring system for security applications is being presented. The inspiration driving this undertaking work is to enhance the present security frameworks. The present system is human checking which has to be eradicated by developing automated tools. The present security administrations depend a great deal on closed circuit television (CCTV) cameras and video filming. This innovation has altered security segments with innovative results. The checking administrations require steady observing by an individual. It is squandering human resources and is inclined to make mistakes. These imperfections leave the framework powerless. The proposed arrangement gives a self-sufficient framework to screen the previously mentioned exercises. The framework will have the option to consistently screen the video which is taken. The principle thought is to utilize Graphics Processing Unit (GPU) quickened Deep Learning techniques to prepare a convolutional neural network (CNN) to distinguish ill-conceived acts. The work done in this paper shows the effectiveness of the method that is being proposed.**

***Index Terms*— Security, CCTV, Neural Network, GPU, automation, CNN, Video, Surveillance.**

I. INTRODUCTION

The research work targets the building up a self-governing framework to screen any ill-conceived action as arranged and characterised by the client. The thought is to utilise profound learning strategies to prepare a Neural Network that can distinguish for ill-conceived actions. At the point when the video is being recorded, the neural system will continually screen for ill-conceived exercises [1]. When any such action is recognised, an alarm will be created and the hour of ready age will be signed into the framework alongside a couple of moments of video film around the hour of that in the examples of movement. The clients can allude to this log record anytime to get the insights regarding the alarm produced [2]. This will help in diminishing human exertion of going through the full video film. This will take out human mistakes. By and large, the framework will help improve the response time towards these exercise components, incorporating the applicable criterion that follows [3]. The main objective of this work is to improve security frameworks by evacuating the requirement for consistent checking of observation recordings by people and to help in better use of Human Resources (HR) [4]. Many people have worked on the crowd counting in low resolution crowded scenes using region-based artificial intelligence (AI) and machine learning [5]. In order to fill this

Grenze ID: 01.GIJET.6.2.11
© Grenze Scientific Society, 2020

principal objective, this work distinguished the accompanying targets for the proposed work and time will be saved instead of using HR. We do this by first preparing the neural system on a conventional dataset containing around 40,000 pictures taken from kaggle.com ordered as '*Anomaly and Normal*' [6]. At that point conveying the model on a raspberry pi appended to the pi camera which is going to take the continuous info and order the scene as '*Anomaly or Normal*' [7]. A brief portrayal of the execution and the tests done are expressed in further paragraphs.

The remainder of the paper is organized as follows: The Section II gives a brief review of the related work. A brief insight into how the problem statement was developed and is being initiated was presented in Section III. The section IV gives the implementation procedure using a standard software tool. The paper concludes with the scope for future work in the section V followed by the references.

II. LITERATURE SURVEY

The following literature survey contains a survey of a number of research papers that were referred towards the development of this research. The papers are from various domains but mostly we discuss here the video surveillance with the help of machine learning, deep learning and convolutional neural networks. Also, some research papers that were referred discusses the deployment of deep learning models on IoT devices such as the Raspberry PI and how complex computations can be carried out on devices with limited computational power. In this section, just an exhaustive comprehensive review of the ongoing works done by different researchers over the year is being presented.

Jing Wang & Zhijie Xu in [8] worked on the crowd anomaly detection for automated video surveillance problems and produced novel results after carrying out a brief investigation. An innovative spatio-temporal texture model was proposed in their research work for its rich crowd pattern characteristics. Spatio-temporal Texture (STT) statistical model was also developed. They showed that STT is sensitive to the changes of crowd motion and could be used for monitoring crowd activates in real-time environment. They mathematically modelled the process using spatio-temporal volume and its slices located at highly dynamic crowd area and defined the spatio-temporal volume (STV) in a 3D cartesian space denoted by *X*, *Y*, and *T* (time) axes using the average field flow function by using

$$W = \sum_i W_i \;\&\; W_i = \begin{cases} 1 & |h_i|^2 \geq mean(|h_i|^2) \\ 0 & otherwise \end{cases}$$

$$V\left(x|_{\theta=d_i}\right) = \frac{1}{2\pi I_0(m_i)} \exp\left[m_i \cos(x-\mu_i)\right]$$

$$0 < x < 2\pi; 0 < \mu_i < 2\pi \;\&\; 0 < m_i$$

which was used for the simulation purposes to obtain the dynamic information of crowd's movement in a video scene.

The authors, Chibloun *et.al.* [9] carried out preliminary investigations on the technique of detecting unusual crowd behaviour in a video sequence using probability models of speeds and directions. They used the optical flow to extract velocities at each image frame, which were then reduced to speed and motion orientations. Using expectation maximization algorithm, the authors constructed a mixture model of von-mises distribution describing the set of directions &a mixture model of normal distribution related to the speed set. Further, the authors also focused on association of each frame of the video with 2 probability densities - one characterizing the speed of the crowd and the other characterizing its direction given by the mathematical model,

$$p_s\left(x|_{\theta_s}\right) = \sum_{i=1}^{k_s} w_{si}\, G\left(x, \theta_{si}\right)$$

$$p_d\left(x|_{\theta_d}\right) = \sum_{i=1}^{k_d} w_{di}\, G\left(x, \theta_{di}\right)$$

Where p_s is the probability model of speeds and p_d is the probability model of direction with the Gaussian densities having the form of

$$G\left(x|_{\theta_{si}}\right) = \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left[-\frac{(x-\mu_i)^2}{2\sigma_i^2}\right]$$

$$x \in R\;;\; \mu_i \in R\;; 0 < \sigma_i$$

& used the Bhattacharya distance given by : $D_b(p,q) = -ln\{BC(p,q)\}$

to find out the distance between the frames, thus producing very fast computational results.
Qing-Ge Ji *et. al.* in [10] worked on the anomaly detection and localisation in the crowd scenes using a block-based social force model, where they proposed approach detects anomaly both in pixel and block levels. In pixel-level anomaly detection, a Gaussian mixture model was used to detect pixel-level anomalies. In block level, the crowd was segmented into blocks according to pedestrian detection, then anomalies were detected and localised using a block based social force model, which had a probabilistic nature making the model to learn the parameters automatically by using component weights, means & the variances of the video frames.

III. Problem Definition

Today's security services rely a lot on closed circuit television (CCTV) cameras and video footage. This technology has brought about a revolution in modern markets, shops, facilities and basically any place that needs constant monitoring for security. It is a great innovation but with one major flaw. There is a necessity for a person to constantly sit by the screen and keep watch. This method is utter waste of the human resource and moreover, has a lot of room for errors. These flaws, in turn, leave the system vulnerable.

A. Existing System

Existing security surveillance systems rely on a network of CCTV cameras connected to a single monitor where a security personnel watches the live stream continuously and alerts the admin in case of some illegitimate activity, thus, making the current system vulnerable to many human errors.

B. Proposed System

Developing an autonomous system to monitor any illegitimate activity as configured and defined by the customer is an issue. The idea is to use Graphics Processing Unit (GPU) accelerated deep learning methods to train a neural network that can identify for illegitimate activity to tackle the problem effectively. The CCTV network will be fully connected to the system running this trained neural network, so that every unwanted activity is recognised on-the-spot. When the video is being recorded, the program will keep a constant track for any illegitimate activity. As soon as any such activity is detected, the program will generate an alert and save the activity information along with its date and time stamp into a log file. These files will also be synced online with the user account on our portal. The customer can refer to this log file anytime from anywhere by simply logging in if they are away from their machines and get the information regarding the activities, providing them the particular time and date of the illegitimate activity so they can watch the footage related to that particular time, thus saving a lot of time. A short clip (approx. 10 mins) will also be stored on the local machine so that they can just watch this video rather than the entire footage.

C. How our solution is better than the existing solution?

The user can also refer to the clips and log files synced online when they are away from their local machine. They just need to log in to their registered account and switch on the notification feature which will send notification as soon as an alert is generated by the program. This way, they will stay updated with the security status of the shop or home or any place the program is being used for. The user can fully configure the program remotely from the dashboard of the account on online portal. They can also read the log files and view clips online. Thus, the time and human resource will both be better utilised in our proposed system.

IV. Implementation Process

A. Requirements and Implementation

The first step towards the development of the proposed work was deciding the objectives first. Since the work deals with surveillance activities and uses artificial intelligence in an effort toward improving the system, we decided that following objectives must be achieved after the development:

1. The system should be able to identify illegitimate activities as configured and defined by the user.
2. It should reduce the amount of storage required by the system.
3. It should eliminate human errors& make automatic.

After the objectives of the system were defined, it was time to develop a plan of action for the development of the proposed research work. These activities come under the planning framework activities. The first stage of planning was to elicit all the requirements for the project. This is the most crucial stage of a planning process as the determination of wrong requirements can lead to a decrease in the overall quality of the project which may lead to failure in achieving the objectives of the project.

As the project deals with AI and implements GPU accelerated deep learning algorithms, the first obvious requirement was that of a GPU. The GPU is a specialized electronic circuitry designed to rapidly manipulate and alter memory to accelerate the creation of images in a frame buffer intended for output to a display. Many GPUs are available in the market, but the two leading companies providing GPUs are AMD and NVIDIA. Due to monetary reasons, it is decided to work with a single NVIDIA GTX 1080. The NVIDIA GTX 1080 is a powerful piece of hardware and it fits our budget allocated for this project. It follows the NVIDIA's pascal architecture, with a 10 Gbps memory speed and 8 GB GDDR5X memory RAM and frame buffer. It has a highest clock speed of 1733 Mhz. The GPU engine supports 2560 CUDA cores. In brief, it is suitable for a prototype of the project proposed in our research work.

After the GPU, we needed a powerful CPU to support the GPU. We went with a i7 7^{th} generation processor and 16GB of RAM. Ubuntu was used for all programming purposes. For deployment of the system, we decided to use an IoT device called the Raspberry PI. Using a Raspberry PI, it was easier to provide portability to the end product and allowed it to connect easily to an array of other independent systems. An auto manoeuvrable drone was also used to test the system, once it was finished.

As hardware requirements were settled upon with, the software requirements were coined. The first software requirement was a framework library that will allow training deep learning models fast and efficiently. We decided to setup a virtual environment on our ubuntu machine. The virtual environment was created by installing Anaconda. It is a python data science platform and supports over 1500 packages. It is easy to install and is widely supported. It also has a large community of users. The packages installed on this installation of Anaconda were TensorFlow and Keras.

TensorFlow is a free and open-source software library for dataflow and differentiable programming mainly used for machine learning applications like neural network. It was developed by Google. Keras, on the other hand, is TensorFlow's high-level API for building and training deep learning for fast prototyping, research and production. OpenCV has also been used for training and testing the model. OpenCV stands for Open Computer Vision library and is a default package for computer vision programs. Python was used as the primary programming language.

After we decided on the software and the programming languages, we had to decide the functions and features of the system. The main purpose of this project was to monitor one or more than one video feeds, analyze them constantly and identify if there is anomalous activity or illegitimate activity taking place in those frames. This leads to a unique problem. There can be several different categories of anomalies from kick, push to murder, theft & other criminal activities.

The challenge with such a vast category is that of a powerful cluster of hardware requirement. Second is the availability of images that can be used to train a TensorFlow model. Since, we have decided on using only one GPU, it was settled that the instead of classifying the frames into several categories and increasing the complexity of the system and the training time and the resources required we would classify the frames as containing normal or anomalous activity.

To train this model being created, a dataset would be required as the input. This dataset should contain images from different classes of anomalies. We used the fight dataset available freely on Kaggle.com and UCSD (Univ. of California, San Diego) dataset named anomaly detection dataset for the same purpose. The fight dataset is a collection of over 38000 images of people fighting. The images have been classified into push, kick, hit, kill, etc. The UCSD's dataset contains short video clips of CCTV footages.

It was also decided to keep the model re-trainable so that in future it could be trained to identify an even wider variety of activities. We decided to store 10 minutes of footage around the occurrence of an anomalous activity to reduce the server space required to store all the video footage. Also, it was decided to maintain a log file to store the timestamp of these activities for fast & easy future reference to these and also concerned authorities.

To summarize, here is the list of requirements that were decided upon at the end were hardware (7^{th}generation processor, NVIDIA GTX 1080 GPU, Raspberry PI 3, Auto manoeuvrable Drone) & the software (Anaconda – A data science platform for python, Python 3.7, Tensor Flow, Keras) & the image Dataset (kaggle.com - fight dataset, Anomaly Detection dataset – UCSD from USA).

The identification of stakeholders was the next step in this process. We identified the group of students developing the project and the people guiding us as the stakeholders for this project. Security companies, shopkeepers and homeowners were identified as the intended users of the end product and also as a stakeholder. After completing all these steps, we were done with the planning framework activity of our process model. The next framework activity is modelling process using the deep learning packages before which the architecture of our proposed work would be discussed.

B. Architecture of the Proposed Security System

After identifying the requirements and the stakeholders, it was time to plan the build of the actual system. The first step towards building the system is build an architecture for the security based monitoring system. Good system architecture was detailed and contained all information about the modules separately and also information about how these modules interact with each other, communicating important data's. The pipeline for the communication between these modules and how the specified requirement is fulfilled. There are 3 separate modules in the architecture of this system.

The first module exclusively handles inputs from video cameras as video inputs. This becomes a complicated task as the number of cameras, to be monitored, increases. As this number increases, the amount of computing power required to handle the input from the cameras also increases. To handle the inputs from cameras, the cameras input feed from their existing systems will be passed on directly to the Raspberry PI. To improve this module and to handle the amount of traffic coming from multiple cameras a first in first out queue can be added to handle the overflow of frames and act as a frame buffer for the actual processing and classification processes.

The second module contains the deep learning model trained for identification and classification works. This module takes frames from the input module and then classifies them as normal or anomalous. The third module is used to handle the output. This displays the camera feeds on screen and then labels them using the feedback from the system. It is also responsible for creating the log file, which maintains the history of the anomalous activities recognized along with their timestamp. It is also responsible for the storage of 10 minutes of footage around every identified anomaly.

The Fig. 1 shows the complete overview of how the system will operate. It starts with the training of the model. A convolutional neural network will be trained using the datasets mentioned before. A reinforcement learning technique is followed by deep learning algorithms to train the network. The trained network model will then be deployed on a Raspberry PI. This model will then receive input images from the Raspberry PI camera buffer. These are then classified and then a reinforcement signal is provided to the neural network model to improve the accuracy of the model.

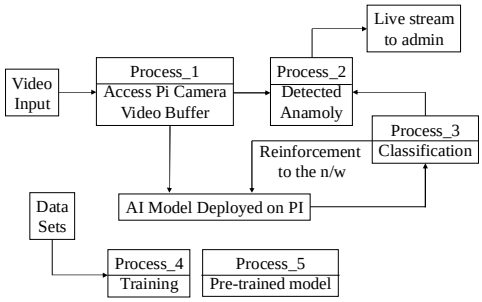

Fig. 1: Flowchart of the system

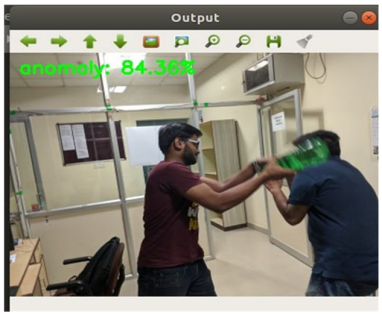

Fig. 2: GUI showing a frame used for testing the system in the simulation

C. Deep Learning Packages

As discussed earlier, there were several packages to be installed before we could start with the coding for the AI model. These packages were TensorFlow, Anaconda and Keras. Here, a detailed description on setting up the environment required for the project has been given. The first software to be installed was the Anaconda. As stated before, Anaconda is a data science platform. It comes with support for 1500 data science packages. Installing Anaconda on a Linux platform is a lengthy process compared to its counterpart Windows. We went with the version of Anaconda compatible with python 3.7. Here are the steps involved for installing Anaconda on Ubuntu 18.04.

1. First, go to anaconda.com & download the Anaconda installer for Linux.
2. Enter the following command to install Anaconda for python 3.7 Bash ~/Downloads/filename.sh
3. Follow through the steps prompted by the installer.
4. Close and open the terminal window for the installation to complete.

Once Anaconda is installed on the system, we go on to install the dependencies required. Those are TensorFlow and keras. We can install them in two ways. Anaconda provides "*conda install*" and "*pip*

install", but, before we go on installing these packages, we need to create a virtual environment. The steps to create a virtual environment are done before using the software tool for simulation purposes. After installation of pip on the virtual environment once it is completed, we can move on to install the TensorFlow and Keras packages. The command for installing TensorFlow is "*pip3 install –upgrade tensorflow*" and the command for installing keras is "*pip install keras*".

D. User Interface

After completing the modelling, the requirements and the architectural design, we still had to come up with a user interface for the application. We decided to go with two different types of user interfaces. One would be the command line interface and the other one would be a graphical user interface. The command line interface would be used for accessing the log files and other related things. The GUI would be used for displaying the camera feed and whether the frames in the feed are classified as normal or anomalous. This completed the modelling framework activity for the process model. The GUI showing a frame used for testing the system in the simulation is shown in Fig. 2.

V. CONCLUSION & FUTURE ENHANCEMENT

A detailed description of the procedure, processes and functions involved in the implementation of an artificial intelligence based video monitoring system was discussed in this paper. Also, the requirements for the project and explanation of the tools used were discussed. In this conclusive section, we look at the results obtained after completion of the project. Conclusions based on this project and provide a series of tasks that can be carried out in future to improve the developed system were drawn.

A. Future Enhancements

Due to the limitation in time and resources, this research does not provide a ready to-use system. Future research can focus on developing the system and proving its performance.

To conclude, we showed the proof-of-concept of a system which is able to emulate operators and can potentially outperform a human being. Once the system knows what is considered suspicious behaviour, it can be automatically detected. This brings us one step closer to an automatic video surveillance system that can be used to prevent incidents and keeping us safe in this digital world as shown in the simulate results.

REFERENCES

[1] S. Ali and M. Shah, "Floor fields for tracking in high density crowd scenes", *Computer Vision–ECCV ed: Springer*, pp. 1-14, 2008.

[2] S. Ali and M. Shah, "A lagrangian particle dynamics approach for crowd flow segmentation & stability analysis," *IEEE Conf. on Computer Vision & Pattern Recogn.*, 2007. CVPR'07., pp. 1-6, 2007.

[3] B. Zhan, D.N. Monekosso, P. Remagnino, S. A. Velastin& L.-Q. Xu, "Crowd analysis: a survey,"*Machine Vision and Applications*, vol. 19, pp. 345357, 2008.

[4] R. Mehran A., Oyama & M. Shah, "Abnormal crowd behavior detection using social force model," *IEEE Conf. on CVPR Comp. Vision & Pattern Recogn.*, pp. 935-942, 2009.

[5] M. Saqib, S. D. Khan, N. Sharma and M. Blumenstein, "Crowd Counting in LowResolution Crowded Scenes Using Region-Based Deep Convolutional Neural Networks," *IEEE Access*, vol. 7, pp. 35317-35329, 2019.

[6] G. Chandan, A. Jain, H. Jain and Mohana, "Real Time Object Detection and Tracking Using Deep Learning and OpenCV,"*2018 Int. Conf. on Inventive Res. in Computing Applns.* (ICIRCA), Coimbatore, pp. 1305-1308, 2018.

[7] M. Nazariani and A. A. Barforoush, "Dynamic Weighted Majority Approach for Detecting Malicious Crowd Workers,"*Canadian Journal of Electrical and Computer Engineering*, vol. 42, no. 2, pp. 108-113, Spring 2019.

[8] J. Wang and Z. Xu, "Crowd anomaly detection for automated video surveillance,"*6th Int. Conf. on Imag. for Crime Prev.& Detec.* (ICDP-15), London, pp. 1-6. 2015.

[9] Chibloun S., El Fkihi H., Mliki M., Hammami & R. Oulad Haj Thami, "Abnormal Crowd Behavior Detection Using Speed and Direction Models," *2018 9th International Symposium on Signal, Image, Video and Communications* (ISIVC), Rabat, Morocco, pp. 197 – 202, 2018.

[10] Q. Ji, R. Chi & Z. Lu, "Anomaly detection and localisation in the crowd scenes using a block-based social force model," *IET Img. Procg.*, vol. 12, no. 1, pp. 133-137, 2018.