TABLE OF CONTENTS

Abstract:

Nowadays, there has been a rise in the amount of disruptive and offensive activities that have been happening. Due to this, security has been given principal significance. Public places like shopping centers, avenues, banks, etc. are increasingly being equipped with CCTV to guarantee the security of individuals. Subsequently, this inconvenience is making a need to computerize this system with high accuracy. Since constant observation of these surveillance cameras by humans is a near-impossible task. It requires work forces and their constant attention to judge if the captured activities are anomalous or suspicious. Hence, this drawback is creating a need to automate this process with high accuracy. Moreover, there is a need to display which frame and which parts of the recording contain the uncommon activity which helps the quicker judgment of that UN ordinary action being unusual or suspicious. Therefore, to reduce the wastage of time and labor, we are utilizing deep learning algorithms for Automating Threat Recognition Systems. Its goal is to automatically identify signs of aggression and violence in real-time, which filters out irregularities from normal patterns. We intend to utilize different Deep Learning models (CNN and RNN) to identify and classify levels of high movement in the frame. From there, we

can raise a detection alert for the situation of a threat, indicating the suspicious activities at an instance of time

CHAPTER 1

INTRODUCTION

Presently, there has been an increase in the number of offensive or disruptive activities that have been taking place these days. Due to this, security has been given utmost importance lately. Installation of CCTV for constant monitoring of people and their interactions is a very common practice in most of the organizations and fields. For a developed country with a population of millions, every person is captured by a camera many times a day. A lot of videos are generated and stored for a certain time duration. Since constant monitoring of these surveillance videos by the authorities to judge if the events are suspicious or not is nearly an impossible task as it requires a workforce and their constant attention. Hence, we are creating a need to automate this process with high accuracy. Moreover, there is a need to show in which frame and which parts of it contain the unusual activity which aids the faster judgment of that unusual activity being abnormal or suspicious. This will help the concerned authorities to identify the main cause of the anomalies that occurred, saving time and labor required in searching the recordings manually. Anomaly Recognition System is defined as a real-time surveillance program designed to automatically detect and account for the signs of offensive or disruptive activities immediately. This work plans to use different Deep Learning models to detect and classify levels of high movement in the frame. In this work, videos are categorized into segments. From there, a detection alert is raised in the case of a threat, indicating the suspicious activities at an instance of time. In this work, the videos are classified into two categories: Threat (anomalous activities) and Safe (normal activities). Further, we recognize each of the 12 anomalous activities - Abuse, Burglar, Explosion, Shooting, Fighting, Shoplifting, Road Accidents, Arson, Robbery, Stealing, Assault, and Vandalism. These anomalies would provide better security to the individuals. To solve the above-mentioned problem, deep learning techniques are used which would create phenomenal results in the detection of the activities and their categorization. Here, two Different Neural Networks: CNN [3] and RNN [4] have been used. CNN is the basic neural network that is being used primarily for extracting advanced feature maps from the available recordings. This extraction of high-level feature maps alleviates the complexity of the input. To apply the technique of transfer learning, we use InceptionV3- a pre-trained model. The inceptionV3, pre-trained, is selected by keeping in view that the modern models used for object recognition consider loads of parameters and thus take an enormous amount of time to completely train it. However, the approach of transfer learning would enhance this task by considering initially the previously learned model for some set of classified inputs e.g., Image-net; which further can be re-trained based on the new weights assigned to various new classes. The output of CNN is fed to the RNN as input. RNN has one additional capability of predicting the next item in a sequence. Therefore, it essentially acts as a forecasting engine. Providing the sense to the captured sequence of actions/movements in the recordings is the motivation behind using this neural network in this work. This network has an LSTM cell in the primary layer, trailed by some hidden layers with appropriate activation functions, and the output layer will give the final classification of the video into the 13 groups (12 anomalies and 1

normal). The output of this system is used to perform real-time surveillance on the CCTV cameras of different organizations to avoid and detect any suspicious activity. Hence, the time complexity is reduced to a great extent.

CHAPTER 2

2.1 LITERATURE SURVEY:

J. Kooij, M. Liem, J. Krijnders This paper proposes an approach to use the sensor systems which can alert on the occurrence of any suspicious activity Although Sensors account for events, it does not provide any information about them. By analyzing the captured video, information about the threat and cause can be obtained very quickly and accurately to take mitigating actions. The main disadvantage is hard ware not working smartly.

S. Mohammadi, A. Perina, H. Kiani proposed a way to deal with the classification of violent and peaceful recordings utilizing behavior heuristic-based approach Apart from violent and normal pattern identification, some authors proposed to utilize tracking to recognize the anomaly and characterize peculiarity as a deviation from that normal movement. But here this approach got less efficient.

W. Li, V. Mahadevan, detected aggressive actions by employing video and audio data from surveillance videos. However, to avoid tracking several methodologies are proposed and implemented. Here the disadvantage is it has obtained only 75% efficiency.

X. Cui, Q. Liu, M. Gao the automated video surveillance system has risen as a significant research topic in the field of public security. A lot of work has been reported, addressing the movement recognition and tracking of an object. But it was difficult to track the device.

T. Hospedales, S. Gong, this paper proposes an approach to use the sensor systems which can alert on the occurrence of any suspicious activity. information about the threat and cause can be obtained very quickly and accurately to take mitigating actions. In this research they have used social force models.

Y. Zhu, I. M. Nayak Thus, the use of CCTV camera and sensor systems, independently or jointly, may not be sufficient for real-time detection of undesired events. So, they designed a system that will detect a threat in time under different lighting conditions using a camera and sensor networks.The system that has been designed was assisted with an intelligent, measurable, nifty and unswerving algorithm.an article, for implementing a Deep Learning based surveillance framework using Object Detection.

Biryukova EV, Roby-Brami A, From the so far literature review, it has been observed that the maximum number of researchers have designed methodologies for learning distribution of ordinary movements from the training done using available

recordings. histogram-based techniques**.** Some have proved that the use of sparse matrices for representation are more effective while solving the problems.

Junseok Kwon Deep learning has resulted in being best for image classification and hence, is found suitable for video activity classification. A background- based estimation and body-based detection were performed to analyze the human outline and capture the human motion using various edge detection algorithms. Some patterns which are resulting in enormous restoration errors are considered as anomalous.

2.2 **Deep Surveillance with Deep Learning – Intelligent Video Surveillance**

Surveillance security is a very tedious and time-consuming job. In this tutorial, we will build a system to automate the task of analyzing video surveillance. We will analyze the video feed in real-time and identify any abnormal activities like violence or theft.

There is a lot of research going on in the industry about video surveillance among them; the role of CCTV videos has overgrown. CCTV cameras are placed all over the places for surveillance and security.

In the last decade, there have been advancements in deep learning algorithms for deep surveillance. These advancements have shown an essential trend in deep surveillance and promise a drastic efficiency gain. The typical applications of deep surveillance are theft identification, violence detection, and detection of the chances of explosion.

We have generally seen deep neural networks for computer vision, image classification, and object detection tasks. In this project, we have to extend deep neural networks to 3-dimensional for learning spatio-temporal features of the video feed.

For this video surveillance project, we will introduce a spatio_temporal autoencoder, which is based on a 3D convolution network. The encoder part extracts the spatial and temporal information, and then the decoder reconstructs the frames. The abnormal events are identified by computing the reconstruction loss using Euclidean distance between original and reconstructed batch.

Intelligent Video Surveillance with Deep Learning

we will use spatial temporal encoders to identify abnormal activities.

The dataset for abnormal event detection in video surveillance:

Following are the comprehensive datasets that are used to train models for anomaly detection tasks.

CUHK Avenue Dataset:

This dataset contains 16 training and 21 testing video clips. The video contains 30652 frames in total.

The training videos contain video with normal situations. The testing videos contain videos with both standard and abnormal events.

UCSD pedestrian Dataset:

This dataset contains videos with pedestrians. It includes groups of people walking towards, away, and parallel to the camera. The abnormal event includes:

- Non-pedestrian entities
- Anomalous pedestrian motion patterns

Summary:

In this deep learning project, we train an autoencoder for abnormal event detection. We train the autoencoder on normal videos. We identify the abnormal events based on the Euclidean distance of the custom video feed and the frames predicted by the autoencoder.

We set a threshold value for abnormal events. In this project, it is 0.0068; you can vary this threshold to experiment getting better results.

2.3 Smart Video Surveillance System Based on Edge Computing

New processing methods based on artificial intelligence (AI) and deep learning are replacing traditional computer vision algorithms. The more advanced systems can process huge amounts of data in large computing facilities. In contrast, this paper presents a smart video surveillance system executing AI algorithms in low power consumption embedded devices. The computer vision algorithm, typical for surveillance applications, aims to detect, count and track people's movements in the area. This application requires a distributed smart camera system. The proposed AI application allows detecting people in the surveillance area using a Mobile Net-SSD architecture. In addition, using a robust Kalman filter bank, the algorithm can keep track of people in the video and also provide people counting information. The detection results are excellent considering the constraints imposed on the process. The selected architecture for the edge node is based on a UpSquared2 device that includes a vision processor unit (VPU) capable of accelerating the AI CNN inference. The results section provides information about the image processing time when multiple video cameras are connected to the same edge node, people detection precision and recall curves, and the energy consumption of the system. The discussion of results shows the usefulness of deploying this smart camera node throughout a distributed surveillance system.

Nowadays, deep learning has shown great advantages in several research fields, for example finance [1], medicine [2], automatic modulation classification in cognitive radios [3], and many others. In particular, computer vision was the first research field in which deep learning took place [4]. New processing methods based on deep learning are replacing traditional computer vision algorithms relying on physical representations, models, functions with some level of meaning deep learning vs. traditional computer vision [5]. The more advanced systems are able to process huge amounts of data in large computing facilities. Current challenges are related to training the machine learning system with enough information which requires the

labeling of those data. On the other hand, in this paper, we focused our attention on smart camera nodes distributed along a surveillance area which are far from that approach.

The modernization of technologies at both the software and hardware level has allowed the design of smart video systems capable not only of managing the video feeds from a closed camera circuit but also analyzing and extracting information in real time from the video streams.

These embedded systems are installed in both public and private locations, being able to control the number of people in an area, crowds' movement, detecting anomalous behaviors, etc. Most of these systems are centralized, executing the computer vision algorithms in one single location. This central processing system receives and processes the information from all the camera networks. Old systems did not process video information and required the operator's analysis, who was duty bound to carefully monitor any camera and whose analysis efficiency may decrease due to fatigue and boredom [6,7].

The modern video processing centralized systems [8,9] store and process the video information retrieved from the camera network. Only some alerts or video-clips are shown to the central console/operator without requiring a high level of attention into the management of the surveillance system.

On the other hand, the emergence of the Internet of Things (IoT) and the computing on the edge nodes [10], has led to the appearance of many research works that propose distributed video-surveillance systems based on this concept [11]. Hence, the intelligence of the system is distributed in multiple nodes, where each one can include a camera and a processing system that performs simple tasks before sending the information to the operator, facilitating its work.

For more complex tasks, current detection algorithms for computer vision include deep neural networks (DNNs). To execute DNNs, a high-end hardware system with high computing power, such as graphical process units (GPUs), is usually required. Apart from being expensive, these hardware modules have a high energy consumption, and it can be difficult to embed them in the distributed smart camera nodes.

In this context, our paper presents a smart video-surveillance system for detecting, counting and tracking people in real time, in an embedded hardware system using new vision processing units (VPUs) hardware modules. The paper describes both the hardware architecture used in the embedded system and the developed computer vision algorithm for detecting, tracking and counting people in real time. This system can be easily installed and configured to work as a smart camera edge node in a distributed video-surveillance system.

The proposed system is based on a low-cost embedded platform UpSquared2 [12], that includes a VPU, the Myriad-X [13]. This allows the parallelization of the developed algorithms to allow them to work in real time, with a reduced power consumption.

A Mobile Net-SSD architecture [14] has been selected for the task of tracking and detecting people. Furthermore, a bank of Kalman filters allows people tracking and counting. We evaluated the performance of the system, making a comparison with other algorithms and extracting the values of the edge node related to performance, computer power and consumption, making a pipelined architecture capable of processing up to 12 video streams simultaneously.

CHAPTER 3

3.1 Embedded Processing System: Hardware and Software Components

Since one of the requirements of the proposed systems is its portability and flexibility for deployment, the selection of the hardware embedded platform was one of the main tasks addressed in the research work. Below, the hardware components characteristics and the software framework used later are briefly explained.

The embedded platform selected for the smart node is the UpSquared2 system [12], a specialized hardware characterized by: an Intel Atom x7-E3950 microprocessor, an 8 Gigabyte (GB) RAM memory, a 64GB embedded Multimedia Card (eMMC) ROM. This also includes the deep learning module of Intel Movidius Myriad X VPU [13], a System-On-Chip (SoC) that can be used for accelerating AI inference with a low-power footprint.

The objective of the Myriad-X device is to process DNNs' inference at high speed and with the lowest possible power consumption. According to the description given by the manufacturer, the architecture of the device allows it to perform more than 4 trillion operations per second (TOPS). This number of FLOPS is achieved thanks to the combination of the neural compute engine and the 16 128-bit VLIW SHAVE (streaming hybrid architecture vector engine) processors that make up the device. In addition, the system is composed of two LEON4 Cores CPUs (RISC; SPARC v8). As previously mentioned, this device enables focusing on achieving high processing speeds in the inference with low power consumption, resulting as perfect for developing an inference in the limit and in a battery-powered embedded system.

To execute AI algorithms in the VPU, the Open Vino [51,52] framework has been used, which eases the optimizing and deployment of CNNs. The Open Vino framework includes two different tools, as shown in Figure 1: the model optimizer and the inference engine. These tools allow optimizing the model and executing it in different hardware platforms such as Intel CPUs, VPUs or GPUs, reducing the execution time. One of the advantages of this framework is that it can be installed in any device that meets the minimum requirements. This also allows the installation of these two modules separately, being able to have the model optimizer in the PC used to train the network and the inference engine in the embedded system.

The Model Optimizer is an application that runs on the command line and that allows to adjust and optimize the neuronal models to achieve an acceleration in the inference of the system. The Open Vino model optimizer can work with different libraries such as TensorFlow, Pytorch or Caffe. The model optimizer provides the intermediate representation (IR) files. These files are one with an extension .xml which defines the layers, sizes and connections of the architecture and a file .bin, which defines the weights of each parameter of the architecture.

Regarding the inference engine, as mentioned above, this Open Vino module can be installed on any device, independently of the model optimizer. This module loads the IR files and runs the inference on the hardware selected by the plugging. It can be run on CPU, VPU or GPU. In addition, the inference engine is in charge of balancing the inference load so as not to overload any device. Thus, the load is distributed between the CPU cores or between a set of VPUs.

3.2 AI Image Processing to Detect and Track People at the Edge Node

One of our research goals was to design an AI application which could detect, track and count people in an embedded system. The proposed architecture (Figure 2) was based on the parallelization of several processes in order to use the hardware modules available in the most efficient way. The analysis was divided into two processes, which communicate through the use of independent buffers.

In the first process, the so-called data analytics process, the preprocessing of the image and the post-processing of the information returned by the AI inference engine were carried out. Before the preprocessing of the image, different algorithms such as noise reduction, image edge detector [53] or any image enhance low-level pixel processing could be applied. As this low-level preprocessing is dependent on the real scenario, and due to the extra computational cost this preprocessing would add, the preprocessing has been reduced to a minimum: ROI selection, image resize to 300 × 300 pixels and (−1.0, +1.0) range normalization of pixel values, in order to adapt the data to the requirements for the input of the first Mobile Net-SSD layer.

The post-processing consists of the reorganization of the data generated by the Mobile Net-SSD network, which contain the bounding boxes of where it has predicted that people are located and then that information is analyzed by a Kalman filter bank that predicts movements and future overlaps, returning a more optimal and reliable result. This first process was entirely executed on the CPU of the system. The second process that confers the system is the one that performs the network inference. The inference is designed to be executed at the edge using a VPU. However, there is also the option of running it on a CPU. Once the Mobile Net-SSD performs the inference, it returns and stores the bounding boxes in a common buffer.

Figure 2 shows a schematic of the flows that compose the main system. By using elements such as the VPU, more than one inference could be implemented at the same time, being able to process more than one video stream. In the case of using a CPU, the capacity to perform more than one inference is determined by the CPU and the number of cores it contains.

Despite the fact that these processes are pipelined for one single video feed, several video streams might also be considered in the process's pipelining. In the case of the Mobile Net-SSD inference, when executed by the VPU, it allows the inference of up to four video streams in an efficient way. This process is explained in more detail in later sections. Depending on the hardware limitations, a higher or lower level of parallelization can be achieved, which is able to interleave the different processes that

make up the system to enable the processing of multiple video streams at the same time.

In the following sections, we discuss the people detection with a Mobile Net-SSD and the use of tracking and counting through a Kalman filters bank.

3.3. People Detection Using a Mobile Net-SSD

This section explains how it works and why this network has been chosen for this system. The chosen network is a Mobile Net-SSD, a Mobile Net architecture [14], which uses the SSD [45] method for object detection. The architecture that forms this network is very similar to the one used in the VGG-16 [54]. The main difference is the replacement of the VGG-16 module by the Mobile Net module. The reasons for using this architecture are two-fold: firstly, because a fast architecture was needed, which could be implemented with selected algorithms such as SSD; and secondly, the resource consumption was low, because the devices used in this project are portable and its hardware is not as powerful as a high-performance PC. For this reason, Mobile Net architecture was chosen, because its main feature is the speed of computing and the use of a type of convolutional layers that allow the use of fewer resources.

The architecture used is shown in Figure 3. It can be seen that at the beginning of the network there is the Mobile Net module, composed of 35 convolutional layers, which are responsible for the feature extraction. A set of five layers of these 35 are in charge of carrying out the classification of objects applying an SSD.

this architecture reduces the intensity in data processing due to the use of separable depth wise and pointwise convolutional layers.

Thus, we compared the computational cost of a standard convolution used in any type of architecture such as the VGG-16 with respect to the operations required in the Mobile Net convolutions.

To measure the computational cost of a standard convolution, first it is necessary to know the size of the kernels that compose it. Standard convolution kernels can be expressed graphically , where the number of channels in the image is M, the number of kernels needed is N, which corresponds to the number of output channels, and the size of the kernels is $D_k \cdot D_k$.

The sequence of work performed for the Kalman filter bank is as follows:

Tracker creation: when an object is first detected in the scene for N (this number is configurable) consecutive frames, the system must assign a Kalman filter to that new object;

Object association: the system must be able to correctly identify the detections made by the detection system with the already assigned Kalman filters. This means that for each object detected, its Kalman filter must be associated;Kalman filter iteration: once the measurements of each object have been delivered to each Kalman filter, the iteration of each one of the filters must be done to fulfill the prediction and correction stages. If a previously identified object does not appear in the image due to an occlusion or a failure of the detector, the Kalman filter can use the estimated values until the object appears again in the scene;

This tracking module will receive information directly from the Mobile Net-SSD in the form of the bounding boxes of the detected objects. For each detected object with the Mobile Net-SSD and exceeding a detection threshold, a Kalman filter will be assigned.

3.4. Pipeline Operation

The pipeline operation was designed to parallelize each of the elements that form the system. The main elements that form the architecture are the preprocessing, the Kalman filter and the Mobile Net-SSD inference. This parallelization allows the processing of more than one video stream at the same time, since the inference is separated from the CPU processing. This inference is processed through the use of VPUs. These devices permit the processing of more than one inference at a time and also in a more optimal way than through CPUs. shows the structure of the processes that make the system up, a structure which can be parallelized as much as the hardware allows. This permits not only the processing of more video streams but also a more efficient use of the hardware systems that make the device up.

3.5. RESULTS

The main objective of the contribution presented in the paper is to propose a portable and fast embedded system for detection, tracking and counting people.

When talking about portability, we mean two goals: firstly, that it can be easily installed anywhere, as shown in and secondly, that it can be maintained and correctly operated without a connection to mains power, i.e., it can be powered by portable batteries.

The results section is divided into two parts. The first part describes the experimental setup, the datasets used and the edge hardware modules. The second part shows the comparison of obtained results with other similar research works. As mentioned below, the target features a portable system with robust detection and capable of running in real time.

CHAPTER 4

4.1. EXPERIMENTAL SETUP

For the system evaluation, the EPFL dataset was used. The characteristics of this dataset fit a possible real scenario in which there are overlaps, large numbers of people and changes in the illumination. This dataset consists of two environments: the first one is a laboratory, which is a large space in which there is a set of four people overlapping each other and changing their position in the image, moving away from and towards the camera. The second scenario is a university corridor, in which there are up to eight people in a small space, at different distances from the camera, with changes in lighting between the different videos and with a high number of overlapping.

Conclusions:

The paper proposes a portable video surveillance system with AI CNN processing at the edge, which can detect and track people in a robust and reliable way. The novel