

LITERATURE SURVEY ON SMART SURVEILLANCE SYSTEM

Rishabh Paunikar
Computer Engineering
Bharati Vidyapeeth College of Engineering
Navi Mumbai

Prof. B.W. Balkhande
Computer Engineering
Bharati Vidyapeeth College of Engineering
Navi Mumbai

Shubham Thakare
Computer Engineering
Bharati Vidyapeeth College of Engineering
Navi Mumbai

Utkarsh Anuse
Computer Engineering
Bharati Vidyapeeth College of Engineering
Navi Mumbai

Abstract—Surveillance Systems using CCTV cameras or any other surveillance devices record the footage all the time they are active. Most of the Data recorded is idle data where no activity takes place. When an activity which has occurred is supposed to be viewed the user has to go through all of the footage to check when and what had happened. This system eases this time consuming task. It uses Action Recognition to filter the idle movement data and trims the part where activity has been recorded using movement detection and various actions. It has a prominent scope for utilization in making the process of examining the footage from surveillance systems simple. This project could be implemented at public places and could enable respective concerned departments to investigate the data at an amazingly rapid rate. Thus, in an era of increasing crimes rates, this project could be a breakthrough in untangling the evidence data and eventually assist in increasing the rate of crime resolution.[1]

Keywords—Surveillance System, Surveillance, Machine Learning, Neural Networks, Action Recognition, Image detection

I. INTRODUCTION

In our previous paper we have discussed our theoretical approach towards making a surveillance system using action recognition. In a nutshell we planned on implementing the YOLO neural network made by Joseph Redmon and Ali Farhadi [2]. We planned on using this neural network[5] which is used for object detection, but with a twist. The object detection is based on probabilities of bounding boxes and it is a brute force method of scanning every image only once and processing it to get results. The trained datasets is stored as in weights and cfg files. So we thought, what if when humans are detected, we make the same network load files based on human activity , that is comparing the resulting human found to be rescanned by the same system but with different weights. So we set out to implement this system with a goal to process raw footage files and detect any actions or activity that is going to occur

II. TRAINING OUR DATA

For testing purposes we decided to create our own training set by using a python script to download images from Google. We downloaded pictures of human walking. These images were kept in a separate folder for ease of access.

After we had about 200 images it was time to make a weights file and a configuration file to use it. We made another script to generate xml files for each image. These xml files contained the exact area (or bounding area) in rectangle where the subject appeared to be performing its activity. The application was simple and every bounding box was required to be drawing over each image to generate the xml files

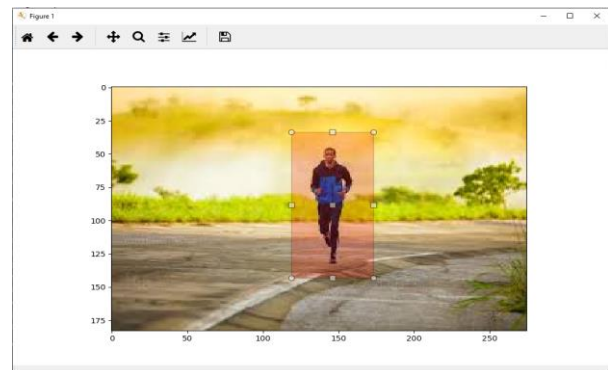


Fig 1. Diagram of the application to draw individual Bounding boxes for the action to be recorded

After the xml files were generated we used a tiny Yolo [3][4]-test-voc file available on the official Yolo[3][4] website. This small file is suggested to be used for training of custom object detection, using their CLI commands we trained the cfg file to detect our custom objects (i.e. activities of humans, walking in this case) the training of dataset was thus complete and it was then time to program the working of this.

III. IMPLEMENTING ACTION RECOGNITION

In the program the same network had to be fed 2 different configuration files. A recursive method was used

for this. In our code we have 2 separate options for initiating the model.

By using a second option we would use the same network with different configuration to process the activity in the function.

We then code the program to search for the bounding box in the image. In a single image the various objects are detected like person.

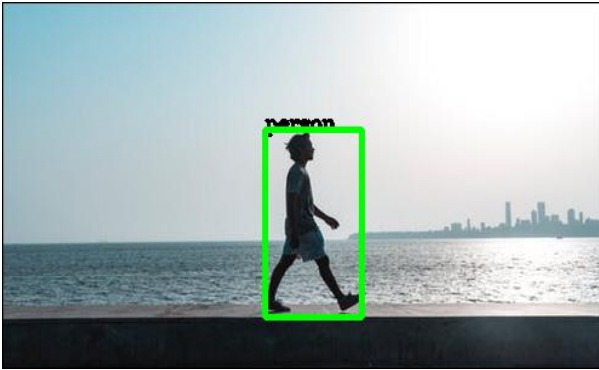


Fig 2. Person being Recognized by Yolo[3][4]

As a person is detected we can then call a subroutine or another function which will use the bounded area to detect what activity is done here. The difference between the two models is that the YOLO [3][4] searches for what object is in the picture while this second function will try to understand what exactly lies in the marked area. As we see in the above image a human is detected. We send this sub image from the image into the next function which uses the same Yolo [3][4] but with different weights and compares the activity dataset to recognize what activity is being done. Since we have trained our dataset on images of running currently we try sending this image from the next function and append the activity to the existing label.



Fig 3. Activity of the recognized person being recognized by the same model with different weights

IV. BUILDING THE GUI

A good application is one that is easy to use and not too complicated. It is something that any person should be able to use and understand. We designed the frontend of our project on PyQt5 which is a GUI framework for python which lets us build an application like interface.

The idea behind the frontend was to keep things minimal and maximize the functionality. 4 buttons were added. The first button was to select the input video which is needed to be processed. After selecting the video to be processed the application will show the file directory of the selected video. This can be changed anytime before any function is executed.

The second button is the save detection button which is useful if you wish to save the videos with the bounding boxes, this is useful when you have too many objects to keep track of.

The third button is a create log button which gives us a text file in log format. This text file contains the information of each frame and at what time it occurs. This is the fastest and best way to generate descriptive information of the video

The final button is the Preview button. It will display the key activities which occur in the video with high threshold values. this can be used to check if any major activity occurs in an idle footage video like shopping malls at night etc to check if any activities occur at odd timings. The preview allows us to understand if the video has any useful activities or it is useless.

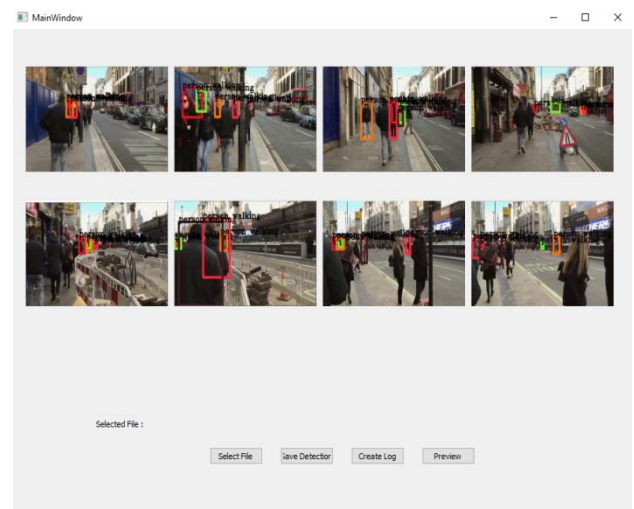


Fig 4. The preview button in the software displaying various different images of the video where the threshold is crossed

The preview shows if the video contains any noticeable activity. A blank preview or a preview without any bounding boxes indicates that no noticeable activity is being done.

V. APPLICATIONS

This software can particularly used at almost any place which requires processing of Surveillance footage however here are a few particular examples:

- To process footage of security malls, to see if any particular individual or group are trespassing the mall afterhours and if yes then at what timing, to pinpoint the timing and reduce the effort taken to take images and screen captures.



- A modification of software can allow for real-time processing which offers for more possibilities like alerts when certain objects are detected or persons are detected at non visiting hours.
- grocery stores which usually don't have a budget to afford expensive security cameras can use a normal cheap camera and also this software for processing , which can save a lot storage space and budget.

VI. CONCLUSION

This paper demonstrates and evaluates the usage of long-term temporal convolutions (LTC)[1][5] and YOLO[2][3][4] for pattern matching, object detection and action recognition. Using space-time convolutions and YOLO[2][3][4] over a large number of video frames, we obtain bounding boxed that detect the object or human in frame. With consequently larger training dataset, the media output will be much more efficient.

ACKNOWLEDGEMENT

This paper was supported by our guide Prof. B.W. Balkhande. We would like to thank our Project Coordinator Prof. Kanchan Doke. We would like to thank our Head of Department, Dr. D.R. Ingle.

VII. REFERENCES

- [1] Varol G., Laptev I., and Schmid C., "Long-Term Temporal Convolutions for Action" Recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence
- [2] Redmon J., Divvala S., Girshick R., "You Only Look Once: Unified, Real-Time Object Detection" Ali Farhadi University of Washington , Allen Institute for AI , Facebook AI Research
- [3] Redmon J., Farhadi A., "YOLOv3: An Incremental Improvement"
- [4] Redmon J., Farhadi A. "YOLO9000: Better, Faster, Stronger " University of Washington , Allen Institute for AI
- [5] Saha S., Rodriguez J., "Convolutional Neural Network" Intel student ambassadors for AI