EFFECTIVE SMART VIDEO SURVILLANCE AND ALERTINNG

Kranthikumar Putta
EFFECTIVE SMART VIDEO SURVEILLANCE AND ALERTING

Abstract:

Machine learning models & AI are popular term in past couple of years. Almost every aspect on our daily life we are using these machine learning models & AI knowing or unknowing. The simple statement of this research paper is to prove reduction of human effort in camera surveillance by using machine learning models such as objection detection & facial recognition.

TensorFlow is leading open source tool developed and published by Google helps to build objection detection and facial recognition library. The code we about to develop can allow user to train model with his choice of images of objects and faces of persons. The user developed model must detect, track and alert user. The code written in python and OpenCV, Twilio, TensorFlow technology used makes python code works. So, this python code helps user without his presence in securing user premises

Keywords: machine learning, object detection, sport, TensorFlow Object Detection API, SSD model, Twilio account, Facial recognition.
EFFECTIVE SMART VIDEO SURVEILLANCE AND ALERTING

By

Putta Kranthi

Supervised by: Abrar Qureshi, Ph.D.
A Dissertation Submitted to the Faculty of Harrisburg University of Science and Technology in Partial Fulfillment of the Requirements for the Degree of Master of Science in Computer and Information Sciences

Computer and Information Science Program

APRIL 2020
I dedicate this project to our parents who have taught us the value of education and dedicate it to professor who have stood by us and taught us course. They all hold a dear place in my hearts, and we thank God for His love and support throughout our course.
ACKNOWLEDGMENT

I am acknowledged our sincere appreciation to the following people without whom this research work would not have been successful. It may not be possible to mention all by name, but the following were singled out for their exceptional contribution, my teachers, colleagues, siblings and parents for the support throughout this project.

I am equally grateful to our Instructor Abrar Qureshi, Ph.D. He gave me the moral support and guided me in different matters regarding the topic. He has been very kind and patient, whilst suggesting to us the outlines of this project, and correcting our doubts. We thank him for his overall support.
TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1  INTRODUCTION</td>
<td>10</td>
</tr>
<tr>
<td>2  LTERATURE REVIEW</td>
<td>10</td>
</tr>
<tr>
<td>2.1 Modern Methods in Object Detection</td>
<td>10</td>
</tr>
<tr>
<td>2.2 Object Detection in Security</td>
<td>11</td>
</tr>
<tr>
<td>3  SOFRWARE REQUIREMENT SPECEFICATION</td>
<td>13</td>
</tr>
<tr>
<td>3.1 Functional Requirement</td>
<td>13</td>
</tr>
<tr>
<td>3.2 Non-Functional Requirement</td>
<td>14</td>
</tr>
<tr>
<td>4  USE CASES &amp; DIAGRAM</td>
<td>14</td>
</tr>
<tr>
<td>4.1 Flow diagram in Object Detection</td>
<td>14</td>
</tr>
<tr>
<td>4.2 Use case diagram</td>
<td>15</td>
</tr>
<tr>
<td>4.3 User stories</td>
<td>15</td>
</tr>
<tr>
<td>4.4 Sequence diagram</td>
<td>16</td>
</tr>
<tr>
<td>5  IMPLIMENTATION</td>
<td>17</td>
</tr>
<tr>
<td>5.1 Blue Print</td>
<td>17</td>
</tr>
<tr>
<td>5.2 Explaining Code</td>
<td>17</td>
</tr>
<tr>
<td>5.2.1 Library</td>
<td>17</td>
</tr>
<tr>
<td>5.2.2 Training Model</td>
<td>18</td>
</tr>
<tr>
<td>5.2.3 Pre-testing</td>
<td>18</td>
</tr>
<tr>
<td>5.2.4 Code explanation</td>
<td>19</td>
</tr>
<tr>
<td>6  TESTING &amp; RESULT</td>
<td>20</td>
</tr>
<tr>
<td>7  ETHICAL AND SOCIATICAL</td>
<td>24</td>
</tr>
<tr>
<td>8  CODE WALKTHREW VIDEO</td>
<td>24</td>
</tr>
<tr>
<td>9  CONCLUSION</td>
<td>25</td>
</tr>
<tr>
<td>10 ANOTED BIBLIOGRAPHY</td>
<td>25</td>
</tr>
</tbody>
</table>
### LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>OD (below) and FR (above) tracking chart</td>
</tr>
<tr>
<td>2</td>
<td>OD, FR and alerting flow chart</td>
</tr>
<tr>
<td>3</td>
<td>User case diagram</td>
</tr>
<tr>
<td>4</td>
<td>Sequence Diagram</td>
</tr>
<tr>
<td>5</td>
<td>The library of OB and FR</td>
</tr>
<tr>
<td>6</td>
<td>FR library(left) and OD library (right)</td>
</tr>
<tr>
<td>7</td>
<td>Step-by-step to train model</td>
</tr>
<tr>
<td>8</td>
<td>Proves FR (right) and OD (left) working</td>
</tr>
<tr>
<td>9</td>
<td>Surveillance code detected 44th president and unknow</td>
</tr>
<tr>
<td>10</td>
<td>Get text threw Twilio saying 44th president detected</td>
</tr>
<tr>
<td>11</td>
<td>Gun detected threw live feed</td>
</tr>
<tr>
<td>12</td>
<td>Got text saying gun is detected threw Twilio</td>
</tr>
</tbody>
</table>

### LIST OF SYMBOLS

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>©</td>
<td>Copyright</td>
</tr>
<tr>
<td>♀</td>
<td>Female</td>
</tr>
<tr>
<td>®</td>
<td>Registered</td>
</tr>
<tr>
<td>≈</td>
<td>Approximately</td>
</tr>
<tr>
<td>♠</td>
<td>Spade Suit</td>
</tr>
<tr>
<td>∂</td>
<td>Partial Differential</td>
</tr>
<tr>
<td>#</td>
<td>Number Sign</td>
</tr>
<tr>
<td>¢</td>
<td>Cent Sign</td>
</tr>
<tr>
<td>TF</td>
<td>TensorFlow</td>
</tr>
</tbody>
</table>
LIST OF DEFINITIONS

The term alert list is the list created of object name by user in a program to get alert of this objects if they detect by camera.
1. Introduction

In computer vision the method of find and name the object is known as the object detection name of person is called facial recognition. From this system we can distinguish the objects in static picture and moving objects in video. This computer vision generally utilized in term of security to recognize objects in overwhelming group territory as reconnaissance. The client effectively distinguishes the objects progressively finished on the off chance that he joined criminal faces informational index to this PC vision it alarms the client when they appear before camera.

TensorFlow object detection and facial recognition API, OpenCV’s, DNN (profound neural system) library are the best open source accessible in market to recognize and distinguish objects. TensorFlow gives pre-prepared zoo models that help to identify different sort of objects with higher exactness. In this paper the video utilizing is ongoing feed from webcam to recognize and name objects.

In this paper TensorFlow object detection facial recognition API was tested on real time feed from webcam to detect objects and persons and compare to trained model to send alert to user. Twilio web service is using to send an alert to user.

2. Literature review

2.1 Modern Methods in Object Detection and Facial Recognition

With the help of modern methods in object detection and facial recognition the code able to detect objects and persons with low false accuracy. convolutional neural networks are widely used in modern object detection. Multibox single shot detector (SSD), YOLO
(you only look once), R-FCN, R-CNN are fastest model system to help object detection with higher rate.

Figure 1 Object detection (below) and facial recognition (above) tracking chart

To improve the impact of picture characterization, picture preprocessing is a significant stage. The objective picture preprocessing incorporates picture design change, picture cutting. Picture position change. Utilizing Python picture handling library PIL to accomplish the transformation of various picture designs.

2.2 Video Surveillance in Security

In video surveillance there are lot of computer vision techniques available in market such as most popular and important are object detection and facial recognition. For security guard it difficult for him to watch every object and person in live feed. Security guard don’t
know wanted person, so he/she don’t notice when they pass camera. In this scenario we need system that remembers faces and objects.

As this video reconnaissance done continuously video brilliance and chromaticity bending with RGB estimation of pixel in catch picture assists with subtracting foundation and shadow. Also, in security reconnaissance its significance to watch little pieces of picture it will done by format coordinating. Channel based following utilize the blend of appearance and movement data to follow objects. As each other detection strategy has their own significance by subtracting objects from foundation to build exactness. By this every evolved strategy and PC vision the object detection code accomplish work without human nearness to work and help client.

Per perfect security prospective in observing is create a lot of data-set of convicts or people in probation and train model and attach this data set to surveillance code and play and sit back the code will do the job in finding this guys. There will be limitation such as process power of computer and camera quality if you overcome this this is powerful program.
3. Software Requirement Specification

3.1 Functional Requirements

<table>
<thead>
<tr>
<th>ID</th>
<th>Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The program should open new window and to capture live feed threw web cam.</td>
</tr>
<tr>
<td>2</td>
<td>The user computer system must allow user to add Twilio account details to system variable.</td>
</tr>
<tr>
<td>3</td>
<td>The program allow user to choose his own choice of objects &amp; persons faces to train model.</td>
</tr>
<tr>
<td>4</td>
<td>The live window to capture feed does not supposed to close until user close it.</td>
</tr>
<tr>
<td>5</td>
<td>The program allows to send message to user once user choice of objects detected.</td>
</tr>
<tr>
<td>6</td>
<td>The program allows to send message to user once user choice of person faces.</td>
</tr>
<tr>
<td>7</td>
<td>The user allowed to choose to use webcam or external camera.</td>
</tr>
<tr>
<td>8</td>
<td>The program must use KNN_model for Facial recognition.</td>
</tr>
<tr>
<td>9</td>
<td>The program must download SSD model for object detection if user don’t have.</td>
</tr>
<tr>
<td>10</td>
<td>The new popup window able to recognize the objects &amp; persons.</td>
</tr>
</tbody>
</table>

3.2 Non-Functional Requirements

<table>
<thead>
<tr>
<th>ID</th>
<th>Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The user system needs to install TensorFlow Object Detection API.</td>
</tr>
<tr>
<td>2</td>
<td>The program is easy to use, which means taking less time for users to get familiar.</td>
</tr>
<tr>
<td>3</td>
<td>The program should be able to run on every average GPU build windows computer.</td>
</tr>
<tr>
<td>4</td>
<td>Don’t store user data</td>
</tr>
<tr>
<td>5</td>
<td>Program should not share user selected images or person faces or record video</td>
</tr>
</tbody>
</table>
4. Use Cases & Diagram

4.1 Flow Diagram:

Figure 2 Live detection, facial recognizing, and alerting flow chart.
4.2 Use Case Diagram

![Image of Use Case Diagram]

Figure 3 Use case diagram

4.3 User Stories

Scenario 1: program allow user to assign hi/her choice of images to OB or FA library to train model based on this.
- Given a user assign us presidential faces
- For FR and gun images for OD. The program
- Stores these images and allow user to train model.

Scenario 2: The program should verify installation of TF and other requirements necessary to run this program successfully.
- User run the code the code must verify
- installation of TF. The code must open
- new window to capture live feed.

Scenario 3: The Twilio account is necessary to receive alert from program
- User has to set-up Twilio account
- For SMS alert by adding to system variable.
Scenario 4: The program shall provide a for users to create alert to user from live feed.

Given a user run’s the program
The new window popups with live
Feed and detects the objects that pass
Threw feed get notify to user.

Scenario 5: User can edit the program to customize the alert.

Given a user run’s the program
When the user customizes the code
To choose the objects that user want
to get alert and ignore all other objects or persons.

4.4 Sequence Diagram

Following is the sequence diagram for program that user set-up custom alerting system.

*Figure 4 Sequence Diagram of surveillance alert*
5. Implementation

5.1 Blueprint

Mandatory step here is user must assign images of objects or person faces to work this program. Without assign images there is no model to name the objects or persons. The code detects objects or persons without model but it does not show any names of objects or persons to user window.

5.2 Explaining Code

This project about smart surveillance which cable to detect objects and persons through live window. The dependence we talk about different code files depends on each other because it’s library.

5.2.1 Libraries

As we can see there are two libraries object detection and face recognition libraries. To setup these financial we need these both libraries comes as one by clicking twillio_app.py.
5.2.2 Training Model

As we see both libraries above for FR the user need to keep his choice of images in train folder click training.py to train model for FR. To this this FR model the user can keep images in test folder to test these the model before use for main purpose. For OD the must keep images of his choice in gun-object-detection folder and click train_frcnn to train model. Once it completes the user can test by clicking test_frcnn to check is model working or not before it attached to main one. The object detection data-size is 333 images and for FR 45 images to train model. To test for OD we using random 50 images in 333 and for FR we using 10 new images which are not in 45.

5.2.3 Pre-Testing

Below are images of working model with proof. The step-by-step process from training to testing shown in following picture. The next is to be testing combination of FR and OD in live feed window.
5.2.4 Code Explanation

The first task of main code is to import OB and FR library to make work together. Next task is to test Twilio account details in system variable to SMS alert to user incase the images get positive compare model.
```
from face_recognition.testing_engine import face_recognition_knn
import gun_detection.keras_frcnn as keras_frcnn
import cv2
import os

# Set up Twilio
from twilio.rest import Client

# Twilio SID, authentication token, my phone number, and the Twil
# are stored as environment variables on my Pi so people can't se
account_sid = os.environ['TWILIO_ACCOUNT_SID']
auth_token = os.environ['TWILIO_AUTH_TOKEN']
my_number = os.environ['MY_DIGITS']
twilio_number = os.environ['TWILIO_DIGITS']

client = Client(account_sid, auth_token)
```

From above code we confirm it first task is to combine both OD and FR library then import rest necessary files.

After loading necessary models to detect and popup live window feed. The code search for gun (because we use object gun for object detection) in live window and create a rectangle box around it or else it’s shows nothing. For FR the code search for faces if detected faces come as positive it’s shows name, or it’s shows unknown rectangular box.
After detecting objects and highlight them in live window with rectangle box around object or person then next task is to send alert to user by Twilio SMS services by following code. And live video feed window doesn’t close until user closed programming window to kill program.

```python
detection_gun = True
if detection_gun:
    from gun_detection.test_frcnn_engine import tf_fit_img, class_to_color
    gun_color = (int(class_to_color['Gun'][0]), int(class_to_color['Gun'][1]), int(class_to_color['Gun'][2]))
video_capture = cv2.VideoCapture(0)
while True:
    # Grab a single frame of video
    ret, frame = video_capture.read()
    if not ret:
        break

    if detection_gun:
        all_dets = tf_fit_img(frame)
        small_frame = cv2.resize(frame, (0, 0), fx=1, fy=1)
        boxes, names = face_recognition_knn(small_frame, 0.4)
        Message = []
        for ((top, right, bottom, left), name) in zip(boxes, names):
            cv2.rectangle(frame, (left, top), (right, bottom), (0, 255, 0), 2)
            y = top - 15 if top - 15 > 15 else top + 15
            cv2.putText(frame, name, (left, y), cv2.FONT_HERSHEY_SIMPLEX, 0.75, (0, 255, 0), 2)
            if name != 'Unknown':
                Message.append(name)

        if detection_gun:
            for (real_x1, real_y1, real_x2, real_y2) in all_dets:
                top = min(real_y1, real_y2)
                left = min(real_x1, real_x2)
                y = top - 15 if top - 15 > 15 else top + 15
                cv2.putText(frame, 'Gun', (left, y), cv2.FONT_HERSHEY_SIMPLEX, 0.75, gun_color, 2)

        if len(Message) > 0:
            Message = 'Detected ' + ','.join(Message)
            print(Message)
            client.messages.create(
                body=Message,
                from_=twilio_number,
                to=my_number
            )

    cv2.imshow('Video', frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
```

After detecting objects and highlight them in live window with rectangle box around object or person then next task is to send alert to user by Twilio SMS services by following code. And live video feed window doesn’t close until user closed programming window to kill program.
6. Testing & Result

In testing this surveillance code first, we tested on facial recognition with following image.

![Figure 9](image-url)

*Figure 9 surveillance code detected 44th and me as unknown*

As we can see in above picture the code detected me as unknown because the trained model doesn’t have my picture and it detected Obama as 44-president because we trained model with Obama image naming 44-president. The next task of this code is sent alert to user because the detected person faces came out as positive.
From above picture we notice that user got text message threw Twilio saying 44-president detected. Therefore, this surveillance core is working with face recognition. Next step is test on object detection as fallows.
From above picture we can see that the program detected gun threw live feed and marked with rectangle and named it and it not detected cellphone because the code doesn’t have it image to detect. The next task is to give alert to user via SMS alert saying gun is detected.

From below picture we can confirm that program working with live detection and alert. The next step is to run code and keep FR and OD together to test the code can send alert both at once.
7. Ethical and Societal Effect

As we are in age of information it’s very important to secure data. The data type we are using in this project are photos of objects and faces of persons so, it’s necessary to secure that data and don’t use for illegal activity and accept privacy.

In surveillance technology it’s ethical to keep limitation in observing and tracking i.e it ok to observe with camera but it’s not ethical to attach scanners(to scan objects what inside their cloths or bags) to camera in public places. The people may feel insecure when they in surveillance like they thought someone watching them they might don’t like it so, important to keep limitations to user let program do it job. The users may feel secured if they use in secure house or private property. It’s user job to mention on wall or in banner about that your (people pass through area) under surveillance.

8. Code Walkthrough Video

Code explain and how-to-setup
https://www.youtube.com/watch?v=9OF9JxIPVhU&feature=youtu.be
9. Conclusion

After I decided to work on this topic it’s difficult because I didn’t find almost any material on this concept. I did find object detection and face recognition tutorials, but I didn’t find them as together so, it very changeling for me to write code that uses both library at once. During development I did learn lot about machine learning models how to choose as per requirement.

I have some difficulties during development especially the code must read detected object or person name to send SMS to user via Twilio. My future work about this is convert to GUI interface so user can use this product by clicking some buttons.

10. Annotated Bibliography

TensorFlow Installation available at:
https://www.tensorflow.org/install

In machine learning TensorFlow is most popular and commonly used platform. It’s open source end-end communication platform published by google. This platform written in python, CUDA, C++.

This open software library helped my object detection & facial recognition program works to detect. This software is base core step of my program.

How to train your own Object Detector with TensorFlow’s Object Detector API:
This tutorial has the detail explanation of how-to set-up TensorFlow with Object Detection API with explanation of how to install TensorFlow and how to train your own Object Detection.

By this tutorial I understand the basics of TensorFlow object detection API, how it’s going to work and learned how to train own object detection with your own images, but my project uses pre-defined model. This tutorial also explains errors faces while running program.

https://www.cs.cmu.edu/~efros/courses/LBMV07/Papers/viola-IJCV-01.pdf

The authors in this journal discuss about object detection in computer vision and the methods that make fast to detect objects the fastest method is R-FCN. These author makes three contribution first one is “integral image” which compute quickly, second is “Adaboost” based learning algorithm which select small number of critical visual feature and last one is “cascade” helps to detects objects from background easily.

By this journal I learned about modern methods algorithm in Facial recognition and how it works in live session.


In this tutorial the author explained hot to set-up facial recognition library with TensorFlow and requirements installation and basic function.
From this I understand how to setup this facial recognition and requirements to makis library works. I understand from this how should train model and its limitation.

Twilio
https://www.twilio.com/sms

This website provides virtual mobile number with SMS services able to send from using internet.

As my project about detecting and alerting this Twilio SMS service need to send alert to user mobile.

MOVING OBJECT DETECTION, TRACKING AND CLASSIFICATION FOR SMART VIDEO SURVEILLANCE by Yi˘githan Dedeo˘glu.

The author demonstrates the object detection, tracking and classification in a smart surveillance and explains what user can do this advanced computer vision technique.

From this paper it made me think what I can do with my object detection program resulting I created an alerting SMS program inside of live object detection.

Retrieve Nvidia Tool Kit at:

Retrieve Protocol Buffers at:
https://github.com/protocolbuffers/protobuf/releases
This file is necessary for my TensorFlow project helps to set paths one code to another code.

**Live Object Detection**
https://towardsdatascience.com/live-object-detection-26cd50ccef7d

This tutorial clearly states how to set up live object detection with webcam using open source software OpenCV. The author of this tutorial explains how to install OpenCV and explanation of live detection code.

From this tutorial I learned about OpenCV and how it’s works especially he explains about code from this I make a variable that can store name of detected objects passing through live feed.

**Raspberry Pi Pet Detector Camera using TensorFlow Object Detection API by Evan Juras**

This author explains how to set-up alert system to user from object detection about pets want to come in or out of door and introduces Twilio SMS servers. His code also states how to read, and store detected object names.

From this author tutorial I understand how to use Twilio SMS and how to make a code that able to read and store the detected object names and matches with user created list to send SMS.

**How to Develop a Face Recognition System Using FaceNet in Keras**
The author focuses on Face Recognition with FaceNet Model in Keras and how to detect face recognition and developing its classifications.

By this paper I understand how to loan FR model with Keras and how to combine with object detection library.