1 Software

1.1 Embedded Systems

Embedded systems is a branch of computer science which utilized the ability to sense and convert real-world properties (i.e temperature, pressure, light) to digital data coupled with communication protocols to enable access to this data across a network. Many modern devices like smart television, light bulb, refrigerators and even gates allow for users to access information and control these devices over a network.

1.1.1 Programs for the ESP 32

The design of a smart surveillance system requires the ability to communicate images taken over a network. As pointed out in the section above, the ESP 32 cam comes with an input bluetooth and wifi module. However, wifi provides data over a longer range and lower latency compared to bluetooth. Therefore, we use the wifi module within the AI-thinker microprocessor to communicate images taken to a user-created network.

Subsequently, the ESP 32 cam is programmed to take an image an broadcast it over the network using an accessible ip-address. ¡read on ip addresses¿. A C/ C++ program makes use of $WebServer.h$, $WiFi.h$, and $esp32cam.h$ Espressif APIs to access pin-outs for the wifi communication. The Arduino base C/C++ code makes use of the $esp32cam_objectdetection.ino$ to both capture and serve images upon request.

1.1.2 Intruder Detection

The entire intruder detection system is developed using python. Module such as openCV are utilized to perform the task of manipulating image and inferencing on the YOLO object detection model. Furthermore, the yagmail framework is used to notify users of intrusion using an SMTP protocol. Finally, the Urllib library is used to handle requests and responses to and from the ESP 32 cam on the network, while Python-Flask is used to design the wepages.

1.1.3 Requests: Urllib

Images taken with the ESP 32 camera sensor are constantly provided at the camera's ip-address. A request is made via a client machine on the network to get images taken per time. Please note that while a compilation of these images in continuous frames is actually a video. The performance of the ESP 32 cam is a close approximation to real-time. Hence, the provided data is in image frames. The requested data is obtained a data blob response which is then processed to an image object using numpy, PIL and openCV python libraries. These images can then be manipulated and feed into the object detection model.

1.1.4 Detection: YOLO

An openCV object model can be cropped and resize to fit some data requirements. In the surveillance system, a detection model is employed to identify persons within the image frame. A YOLO model which includes several CNN block and an IOU loss function is used to predict the probability of the image containing a person. Parameters such as threshold value can be used to further discriminant prediction for higher accuracy.

Additionally, the sliding window and IOU mechanism is used to localize items within the frame augmenting the information on probable element identity. In this study, we utilize the YOLO v5s model for its relative size and efficiency on limited computer power. The model is an evolution of versions developed from YOLOv1 (2015) - YOLOv5 2021. However, some concertions have to be made on the performance of the model to maximize compute power and enable higher Frames per Seconds (fps). ¡More details on the YOLO website¿

Eventually, predictions were isolated based on the desire label, which is "person" to identify intruders within the frame. This identification was observed to be improved by adjusting the threshold parameter and reducing the input image size and resolution.

1.1.5 Notification System: Yagmail

Our study recognizes an intruder using the YOLO models prediction output. However, intruders could remain in subsequent frames for a period of time. Furthermore, a non-intruder could be in the view of the camera which could lead to false detections. Hence, a simple technique is used to mitigate false alarms. First, an "enable / disable" button is place on the web interface to allow the user (property-owner) to disable within set duration of arriving in the camera's view or enable once surveillance is needed. Also, a time-duration is set from previous notification of intruder to next prediction of intruder after which another notification is sent.

Notifications are sent using an SMTP protocol configurable on most email account to permit the server to sent out notification email using a pre-determined email address. The Yagmail library in python allows for construction of email with message, subject, sender and recepient's information. The library however requires a secure token to be provided or other access to verify usage of the email account.

1.1.6 Webpage: Python-Flask

Finally, our surveillance camera requires a display interface for viewing video stream from the camera over the network. Thus, a webpage is designed using Python-Flask microframework. The framework allows for simple and efficient connection of network using HTTP protocol for resource sharing. Furthermore, It is directly scalable over the internet and other wider networks.

Python-Flask requires webpage files and their relative styles and setup a port to provide these files on the network at a particular address. As a result template files were created using html, css and Jinja syntax to design a webpage with a video streaming output and button to "enable/disable" the notification system. Along with the webpages, an "app" script was written to control serve images through the generative output of the detection module and provide them at an element end-point within the webpage.