ORTA DOĞU TEKNİK ÜNİVERSİTESİ
KUZEY KIBRIS KAMPUSU

CNG 491

First Iteration Report
Intelligent Security System for Abandoned Luggage

Team Members:

Damla Gül Altunay – 2015881
Naciye Karademir – 201609

Table of Contents

1. INTRODUCTION

This section includes the information about Intelligent Surveillance System for Abandoned Luggage (SSAL). Following parts are divided into three. In the first part, what kind of Surveillance System is going to be designed and its usage is explained. After that, scope and importance of the system is given. Finally, details regarding the development process of the project is outlined.

1.1 *What kind of Surveillance System is SSAL?*

SSAL is an intelligent surveillance system that combines Computer Vision and Machine Learning techniques in its implementation. The main purpose is to find the abandoned luggage/s in the scope of the indoor environment of the airport, and track its owner/s by checking previous frames.

1.2 *Why is this project important?*

Due to security concerns in the last decades, implementing an intelligent video surveillance system in order to help the human actor in monitoring public places has become a necessity. Unattended object detection is one of the most important and critical tasks in such systems. Most of the systems requires a security personnel to track all dangerous situations and checking illegal items of people. Besides that, they also need to keep track of abandoned objects and its owner to prevent an unexpected accident such as the bombing of a public space. It is not the most sufficient way to control public safety. SSAL will take this job from security stuff and provide a better secure environment for people in airports.

1.3 *How will the process be?*

Software implementation will consist of two parts. In the first part, by using computer vision algorithms, static objects in each frame of the camera will be detected. Detected objects will be sent to the second part of the program which is recognizer. In this part, it will be decided whether or not the object is luggage with the help of trained model of luggage dataset. If luggage, it will check short-term memory to find its owner, otherwise luggage will be considered as abandoned object and the alarm triggers for personnel.

2. Background, Aims and Organization

With rapidly increasing deployment of surveillance cameras in public areas, it is required to design a reliable system for keeping citizens safe. As Dahivelkar et al. [1] stated in their paper published in 2017, as a result of terrorism and global security, people are feared to take public transportation with the attacks in their mind.

The detection of suspicious (dangerous) items is one of the most important applications. These items can be grouped into two main classes, dynamic suspicious behaviors (e.g., a person attempting to attack others) and static dangerous objects (e.g., luggage or bomb abandoned in public places) [2]. The scope of this project falls into the latter category and the environment is airport as a public place.

Our aim is to design a system that detects abandoned luggage which is left in public places such as airports, and then define these objects if it becomes a threat. These threat issue will be held by doing event recognition so that we can find the owner of the luggage and evaluate its threat. This will be the main feature to distinguish this system from existing systems.

The system should be able to detect luggage and also its owner from the coming frame so that if the luggage is abandoned, we can find its owner. Abandoned object is defined as an entity which is absolutely static in the scene for more than a time period T and the owner is not present around it. To sum up, main objectives are:

- Fast detection of baggage that has been abandoned,
- Fast identification of the individual who left the baggage, and
- Fast determination of owner after moving away from the luggage

This project is initiated as a Senior Project in Middle East Technical University and carried by two students and two advisors. Project organization is handled with division of labor. Naciye is working on Computer Vision part which is the first part for the flow of the program. Damla is working on Machine Learning which is the second part. The flow of the program which is the solution for the formulated problem above will be explained in detail in the following parts.

3. REQUIREMENTS

3.1 System Requirements Specification

3.1.1 Functional Requirements

Desktop Application	
DA -01	Administrator of the system should log in for management operations, otherwise there is no need to login for security personnel.
DA -02	Administrator should be able to add security personnel to the system, also s/he can edit profile of personnel and delete information from the system.
DA -03	Administrator can also add another admin to the system and do the same operation as can be done for security personnel.
DA -04	Administrator can connect a new camera to the system by entering its IP addresses.
DA -05	Administrator can remove any camera from the system by entering its IP address & name.
DA -06	Administrator can activate a camera which is already connected but not activated.
DA -07	Administrator can deactivate a camera which is connected and in use.
DA -08	Administrator and Security personnel can control which camera will be displayed on screen by selecting them on left-side menu.
DA -09	Desktop Application will keep track of the event log, which is created for detected luggage and its owners in order to show time and description.
DA -10	In Event log, detected abandoned luggage and its owner will have the same ID to associate each other.
DA -11	Every data in event log should be added to the database.
DA -12	After detecting abandoned luggage, it will be surrounded by a bounding box which refers that it could be a danger, and after waiting sometime, if owner still does not arrive, then it will give warning signal.
DA -13	Desktop Application will include a “Help” section which gives detail information about the usage of the system.

Camera	
C -01	Camera should be connected to the Internet.
C -02	Minimum FPS should be 20, and pixel resolution should be 640*380 for better detection

3.1.2 Non-Functional Requirements

In the following sessions, non-functional requirements of the system are divided into usability, reliability, robustness, performance, supportability, and security.

i) Usability

The system should be easy to learn for desktop application users: Admin and security personnel. There will be a help page and complete user documentation which will explain how to achieve common tasks. Error messages should give specific instructions for recovery. The help system will explain all functions and how to achieve common tasks.

ii) Reliability

The reliability of the wearable device essentially depends on the software tools (OpenCV, Tensorflow etc.) and hardware tools (camera) used for the system development. In order to handle this issue, system will always check for dependencies (tools, libraries) and functionality of camera. If there is any problem, then it will directly report administrator of the system.

iii) Robustness

Problems in data delivery can occur anywhere with the communication between data sources in our case camera and the surveillance system. We take robustness as not only that the system continues to run in the wake of data failures or delays, but also that it continues to provide meaningful information to its users. Our main priority should be this in order to handle the problem. Since if the user aware of what happened, then it could find a solution. Besides that, system is going to check every error and failure to find a solution for them.

iv) Performance

Image data transfer through Internet connection and live streaming makes performance measures critical. For desired performance, image capturing, transferred data size, speed of connection, response time, processing speed must be considered. System should work real-time which means that data should not be faced any serious delays. Because of this, IP cameras that is going to be used in this system requires high value for FPS and minimum 640x380 resolution pixel to recognize objects from each frame.

The database should have sufficient memory for incoming data from camera. Also, there will be another record for event log, in which detected abandoned luggage and its owner images, ID and time will be stored.

v) Security

All data for camera and event log is going to be stored in database. To access this service, user should have an administrator username and password so that s/he can login to the system. In this way, we can protect our data.

3.2 System Models

3.2.1 Use Case Diagram

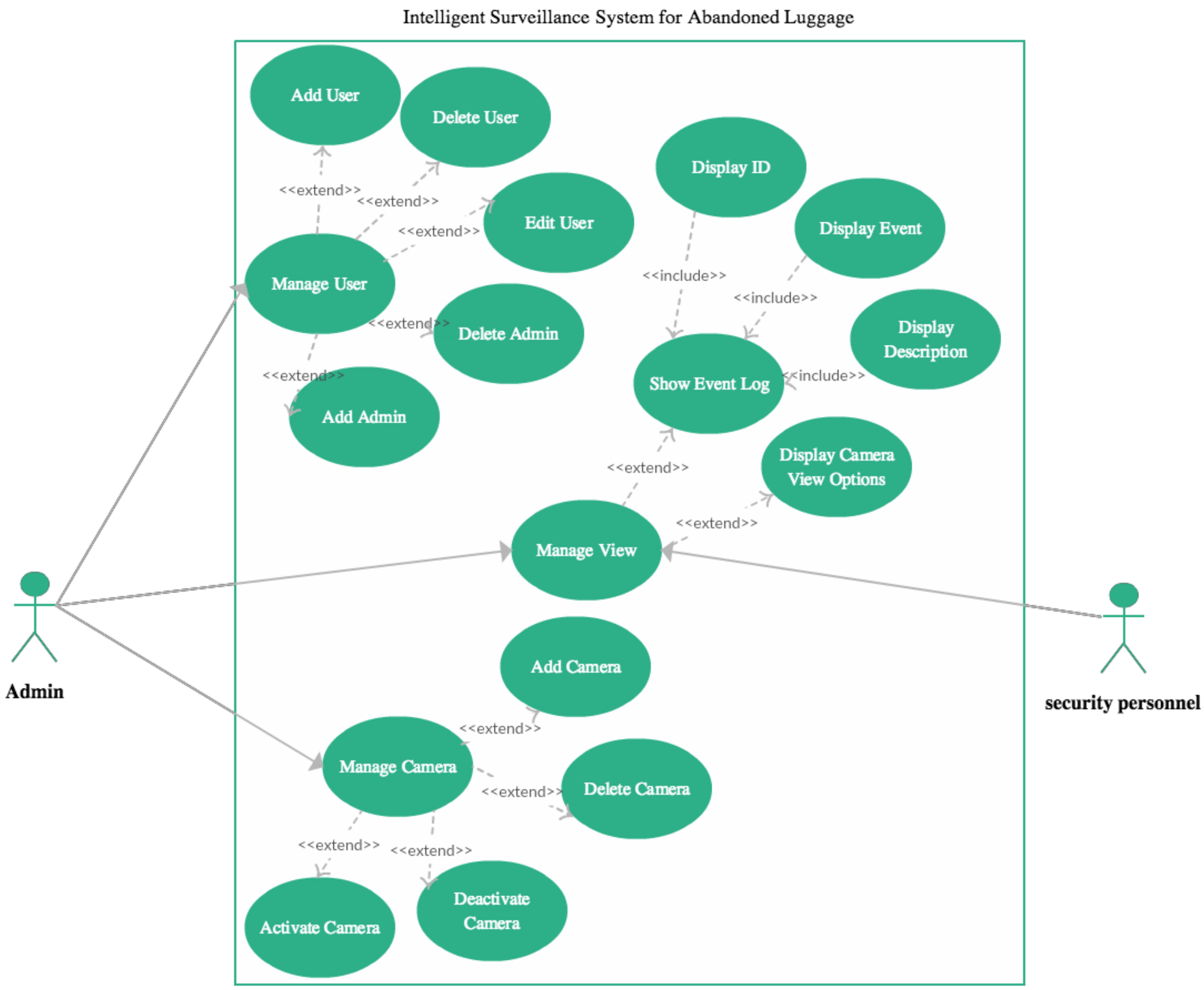

Figure 1: Use Case Diagram

3.2.2 Sequence Diagram

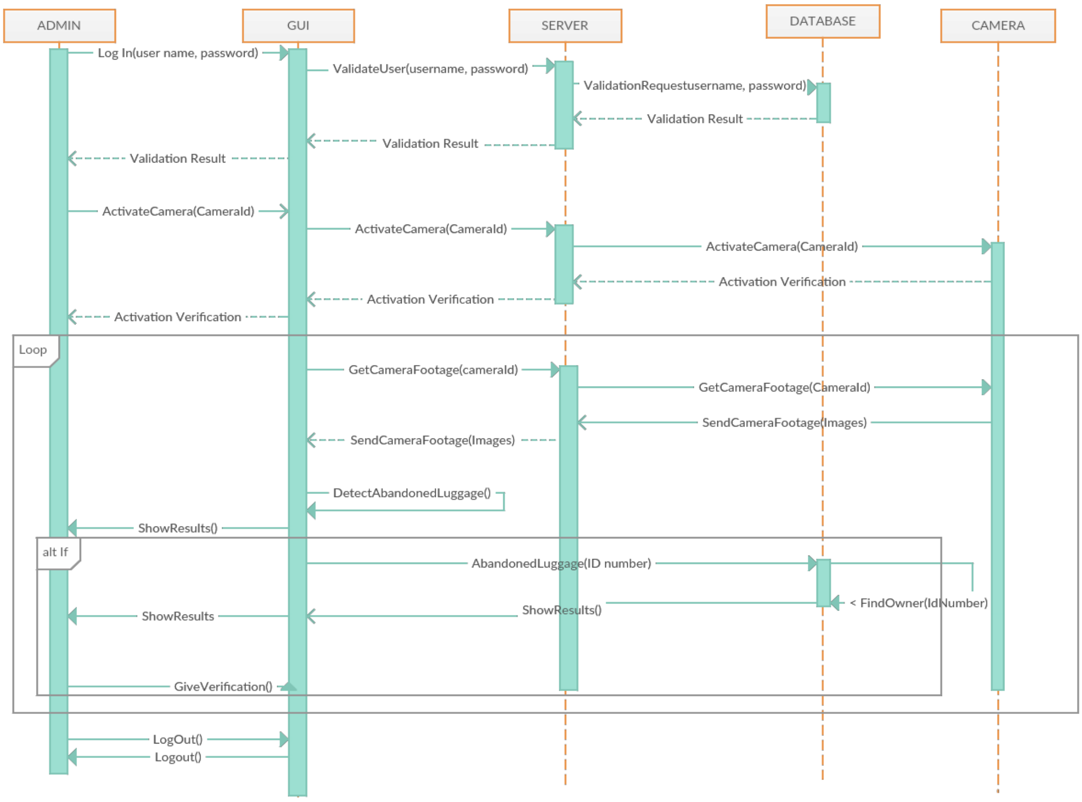

Figure 2: Sequence diagram

3.2.3 Database ER Diagrams

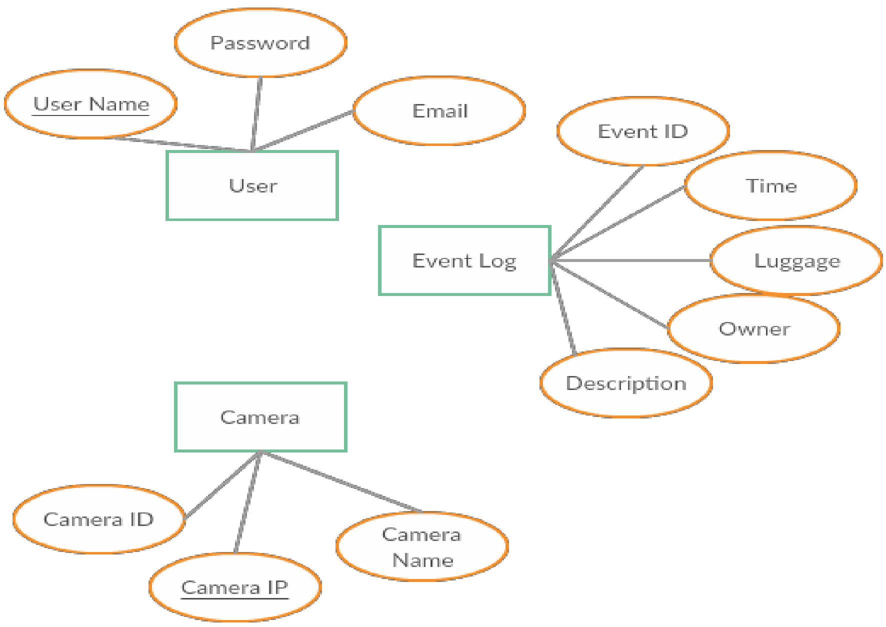

Figure 3: Database ER diagrams

3.2.4 Context Diagram:

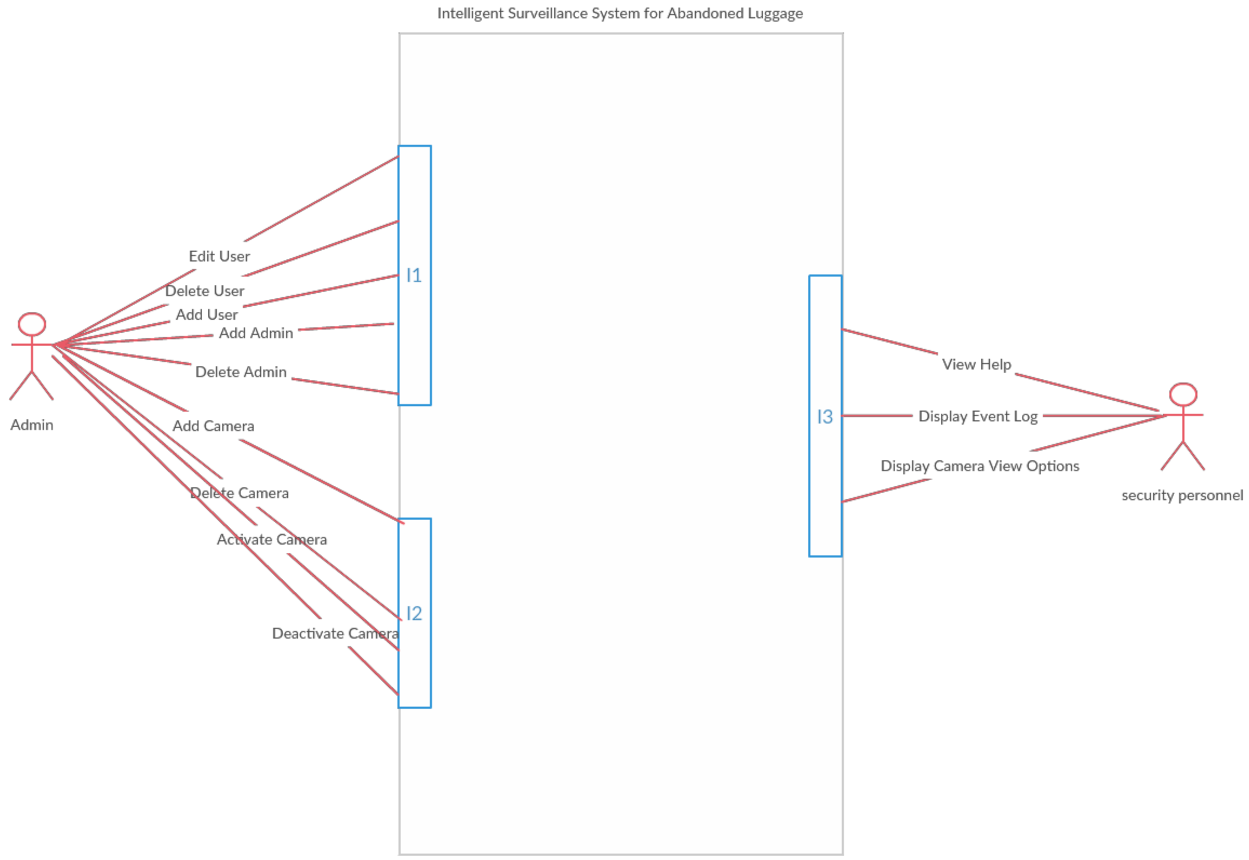

Figure 4: Context Diagram

3.2.5 Architectural diagram level-1

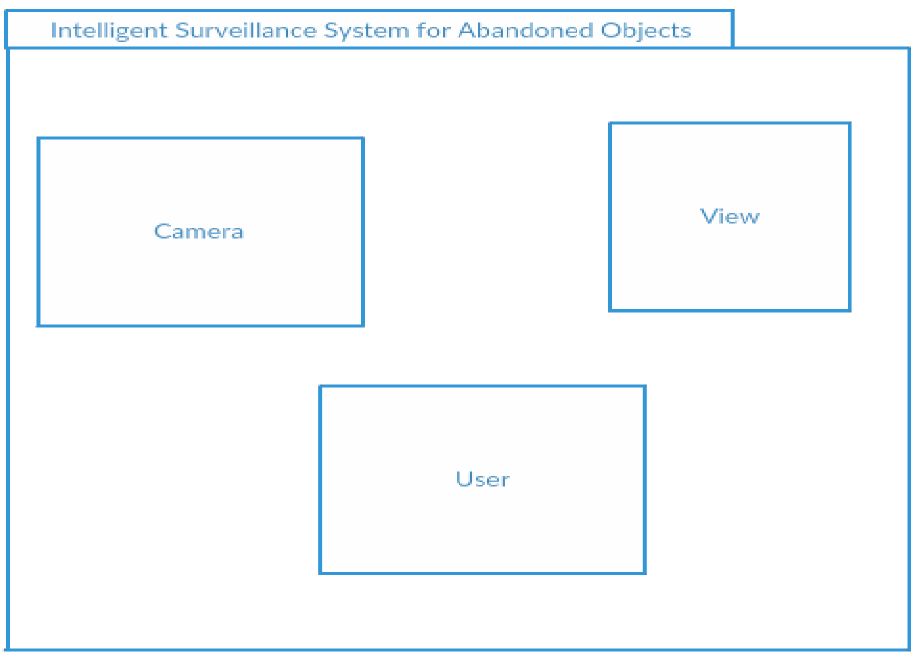

Figure 5: Architectural diagram

3.2.6 Process Model

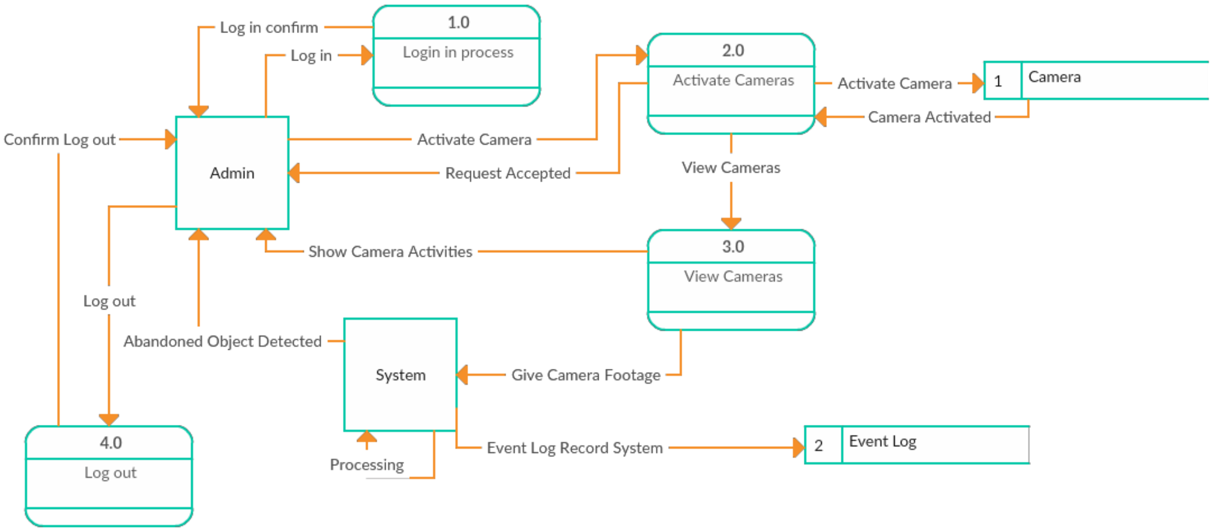

Figure 6: Process Model

4. Utilization of Engineering/Problem Solving Skills

4.1 Problem Solution

Besides existing researches, our contribution will be

- Output of the system will not only detect the abandoned object but also find its owner, evaluate threats and give alarms for it.
- At the end of this project, most popular and fastest algorithms for objects detection which YOLO and Faster R-CNN will be used, that means we are going to show usability of 'deep neural network' for these applications.

While detecting static objects in each frame, we are using image processing algorithms, these are done by taking as a baseline of algorithm implementations of OpenCV library but in more complex way (more explanation will be in Computer Vision sub-section). Detected static objects are sent to machine learning part of the program in which objects are classified as luggage or non-luggage (any other static object in the frame) by using trained model of created luggage dataset.

If it is a luggage, it looks for its owner back from the archive video frames, which is done with short-term and long-term foreground modelling, and tracks them. By checking time and distance with its owner, it is decided whether luggage is abandoned or not after leaving as a static object in the environment.

Below is the flow of the program that currently we are working on:

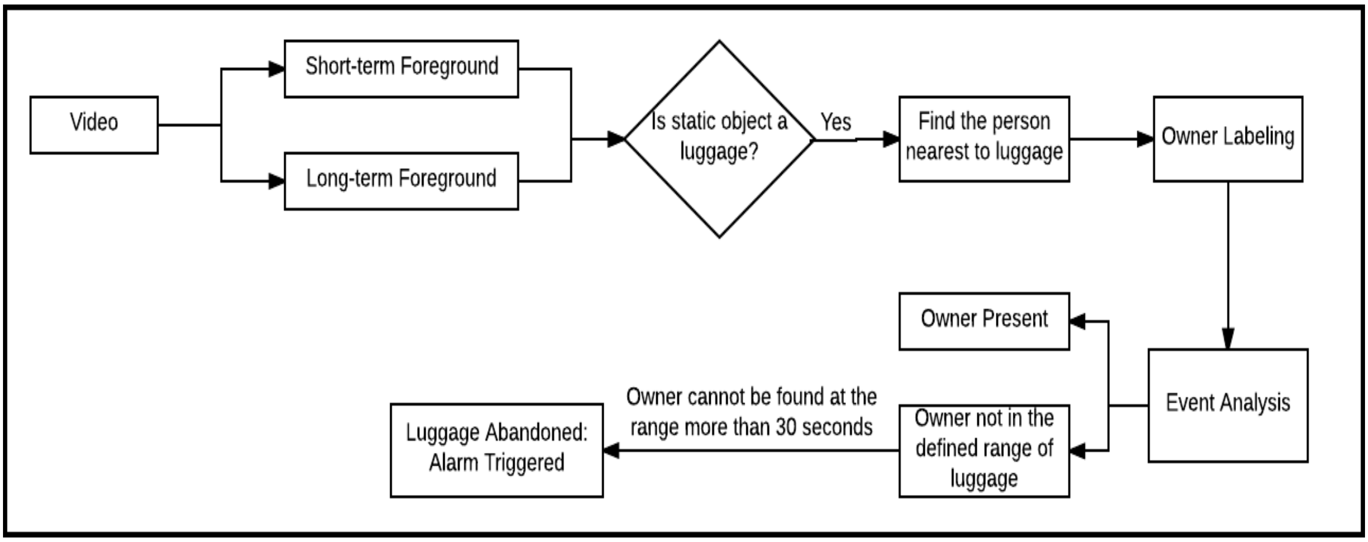

Figure 7: Flowchart of the program

Every person in the frame is detected using YOLO, and then by using short-term foreground modeling, we start to check every frame if there is a static object that did not appear on the previous images. If there is a new static object/s, we check these are luggage or not using same algorithm (YOLO). At the same time we label the nearest person to the object as its owner. Now, we have a luggage and owner of it. After that we start to analyze the event. If the owner is not in the area which is bordered with the defined distance range around the luggage, then we keep time (30 seconds). After 30 seconds, if the person is not still in the frame in other words not around the luggage, then we trigger the alarm and show the luggage with a bounding box and name it as “abandoned luggage”.

4.1.1 Computer Vision (Background Modeling)

Since the system will be used for real-time video surveillance we use computer vision techniques and this is one of the main parts. These techniques are used to make our system faster, more reliable and more robust.

In SSAL time for image processing is very important because we are working on real-time videos. Using only machine learning without pre-processing in the frames leads to time-consuming calculations that cause delays that we don't want to have in our system. According to Liao (2017), using computer vision techniques in the first step of their methodologies made their system faster and more robust for luggage detection. We are following their steps also.

Machine Learning techniques are not slow but without reducing the number of objects that are possible abandoned luggage, the system will try to identify all of the objects in a frame. This process takes time and slows the system down. So, to reduce the number of objects, we are applying for following technique.

Since looking all objects in the video to decide whether it is abandoned luggage or not takes a lot of time, we aimed to find less number of objects for recognizer (where we will decide if it is a luggage or not). We will detect static objects in the video frames, and send the recognizer. To detect static objects, a dual background modeling is applied for the system. Dual background modeling is specifically used to detect static objects in videos. Flow for dual background modeling is below.

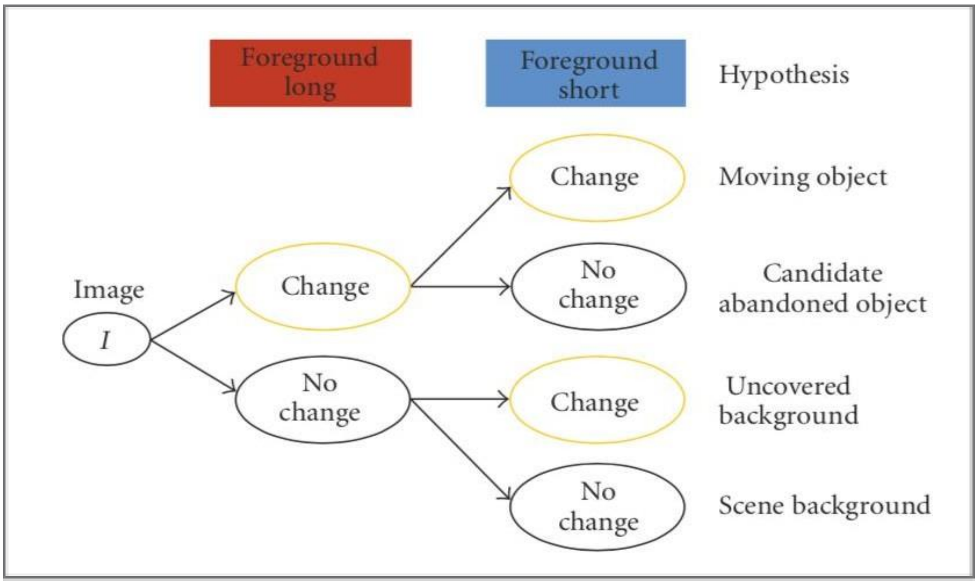

Figure 8: Hypotheses on long- and short-term foregrounds. [5]

So dual background usage basically is reducing the number of objects that should be checked by the recognizer. By this method, we are able to detect static objects first and then look at it if it is an abandoned luggage or not. To illustrate, dual background modeling steps are:

- Apply generic background modeling where we only can differentiate between static background and moving foreground.
- Differentiate between moving objects and static background in the frame.
- Keep two backgrounds in the system which are long-term and short-term backgrounds to compare background models later on. The main difference between long-term and short-term backgrounds is that system updates short-term background more than a long-term background.
- Liao (2017) states that they updated long-term background every 40 seconds and short-term background every 2 seconds. Then they compare long-term and short-term backgrounds to find which moving object in the long-term background is static now in the short-term background. We are also following their research with different parameters for the system.
- Send detected objects to the recognizer where we will understand if the object is a luggage or not by using the machine learning techniques. Liao (2017) argues that in their original work this method reduced the number of objects that proposed for abandoned luggage significantly. Their work is below.

Figure 9: An example of static foreground detection in PETS 2007 dataset. The time point in the second row is 270 frames after the one in the first row. (a) shows the person/object

detection (bounding box) and owner labeling (green line). The red lines are the tracking traces of detected person. (b) and (c) are the foregrounds which are extracted from the long-/short-term model respectively. (d) shows the static foreground. The place of detected bag of interest is indicated by red bounding boxes. [4]

We decided to use static cameras for our detection. Porikli (2007) argues that results for dual background modeling are better in terms of accuracy if the camera is static. Also if abandoned luggage belongs to the original background, luggage will not be detected. So, luggage should have brought to the scene after we initialize first long-term background.

Generally, this method implemented by using Gaussian mixture modeling (GMM). We will also use GMM since we need to decide if a pixel belongs to moving foreground or static background. Deciding to parameter is a challenge because we need to take right results with an optimum time. So steps for the solution is below:

- For background subtraction, we need to apply GMM.
- Make a pixel zero if it belongs to the static background and one if it is belonging to the moving foreground. To understand the difference, we apply GMM that we are using suitable parameter for the SSAL.
- After construction our first background models then we keep it as long-term background.
- Keep the same background also as the short-term background.
- Update long-term every 40 second and short-term every 2 seconds.
- Then compare two backgrounds continuously. To compare, if a pixel belongs to long-term background say it is LB, and if it belongs to short-term background say it is SB.
- Comparison:
    - LB(x, y) = 1, SB(x, y) = 1 → moving foreground.
    - LB(x, y) = 1, SB(x, y) = 0 → possible abandoned object.
    - LB(x, y) = 0, SB(x, y) = 1 → background pixel that was occluded before.
    - LB(x, y) = 0, SB(x, y) = 0 → static background.
- When a static object found, we are sending bounding box results of the object to the recognizer where we check if it is a luggage or not.

To summarize, we will use computer vision techniques in our project since we are working on real-time videos. We will do background modeling to detect static object at each frame to do our operations later on. After detecting static objects, system will send bounding box information of each object to recognizer. So these techniques will help us to detect static object, make our system more robust, reliable, and it will help to make the system much faster.

4.1.2 Machine Learning Part (Training for dataset)

The modern history of object recognition goes along with the development of Convolutional Neural Networks (ConvNets), which was all started here in 2012 when AlexNet won the ILSVRC 2012 by a large margin. AlexNet bases on the decades-old LeNet, combined with data augmentation, ReLU, dropout, and GPU implementation. It proved the effectiveness of ConvNets and started a new era for computer vision.

Since 2012, many methods/algorithms have been published, and almost in every new algorithm, the results in terms of speed, accuracy and the computational cost are getting improved. For this project, we started to compare two algorithms: You Only Look Once (YOLO) and Faster Region-based Convolutional Neural Networks (Faster RCNN) for training our luggage dataset. The aim is to find out the best algorithm for our application, after getting the results for both algorithms, we are going to decide which algorithm is going to be used at the end.

Here is the steps before starting to training, we should have a dataset of objects that is recognized (in our case, it is only luggage).

- The first job is to create a dataset by collecting luggage images. There is not only one type of luggage, so we collected as much as possible different types of it. Now, we have 7,000 images in our dataset.
- Secondly, each image is labeled and an xml file which includes the position of luggage in the image is created.
- Next, annotated images are separated into two categories: train set and test set. Test set is chosen from 10% of the whole dataset.

For YOLO algorithm, as Redmon et al. [6] did for their research, we used DarkNet framework to train our dataset. Below is the process:

- A configuration file in which property of each layer, number of layers, number of classes, batch sizes are define is created.
- A pre-trained model is chosen in order to have an initialization for weights.
- Until the loss reaches 0.06 or below, the program continues its training. This threshold is given by DarkNet.

We did not train our dataset for Faster RCNN, but process will not be different from as we did in YOLO.

- After training, we have our own model to use in detection program. By using Tensorflow and OpenCV, a detection program is written and tested with images, videos and cameras.

In overall program, this part is going to be used after background modeling. After getting static object in each frame, this objects are sent to recognition (detection program) and decide if the object is luggage.

4.1.3 Owner Labeling and Event Recognition

After finding out left luggage or in other words the luggage which is not moving by anyone, the person who is the closest to the luggage is defined as owner of it.

For now, we defined the events to check if the luggage is certainly unattended. It is one of the possible case that owner wants a break put his bag on a table and then go to somewhere else far away from luggage for a while, and come back. This case satisfies the rules for abandonment, but the bag is not abandoned. We aim to find out a solution for this cases.

As shown in Figure 10, a luggage item is unattended when the owner is further from the luggage. The border of yellow part is called as “warning zone” [7], where the luggage is neither attended to nor left unattended. Also, after moving away from “red zone” which is almost 3 meters from the object, then alarm is waiting for 30 seconds to trigger.

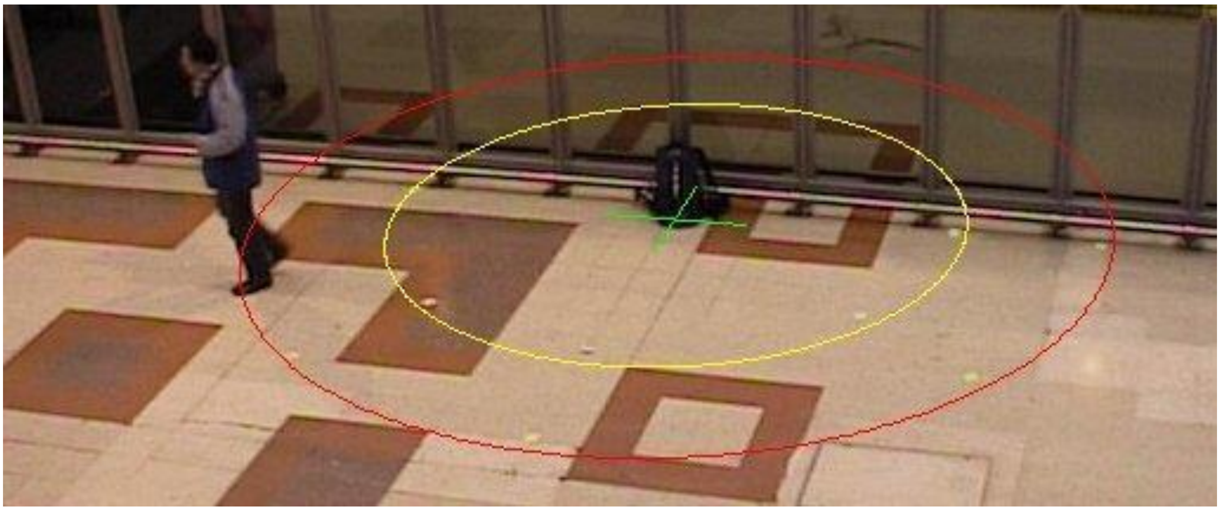

Figure 10: Distance around luggage

At that time, like Liaoa et al. [4] did their paper by using the approach proposed by Xiao et al. in 2016 [8]. The method is to extract deep features for person and called as "Person Re-identification". To find the person in the camera footage after leaving the red zone, we use his/her features and every time the person in the scene we show it by bounding box with the same color of bounding box of abandoned luggage.

4.2 Assumptions

We included a reasonable amount of assumptions to the SSAL so that our system could be more useful in a variety of surveillance systems. All assumptions are stated below.

1) Assumptions on Abandoned Object: There are various definitions of what abandoned object is. Tian (2008) gives the definition of an abandoned object as to be a stationary object that has not been on the scene before. We also assume that it is left by someone with a bad intention. Also, PETS dataset assumption is that an object has not its owner in 3 meters for more than 30 seconds. Based on these papers we assume that abandoned objects are left by its owner more than 30 seconds and its owner is not in the 3 meter and we will look if that object is a luggage or not.

2) What is Luggage: Dictionaries define luggage as suitcases, containers and other bags which hold personnel belongings while traveler is in transit. Since SSAL planed to be used in airports we assume that luggage is every bag can be threatened if it is left intentionally by its owner while traveling.

3) Indoor Environment: To make SSAL more useful our another assumption is that it is not used for an outdoor environment. Indoor and outdoor systems both have usually similar features but outdoor systems should be able to compatible every kind of weather and lightning. This is why we assume that the environment is considered for only indoors.

4) Use of Static Camera: We assumed that camera that we add to the system should always remain static. This assumption relates to background modeling in our system. Porikli (2007) indicates background modeling is stronger under static cameras. The reason is that we are trying to find static objects and if the camera moves it is harder to detect static objects from it and usually gives wrong results.

5) First Frame of Camera: As we stated above we will use dual background modeling in the implementation. Since dual background modeling compares two frames to decide if there is an abandoned object, we assume that in the first frame that we got from camera shouldn't contain any abandoned object. Porikli (2007) says that an abandoned object should not be the part of the original area of scene. Because of this reason we assumed that the first frame is the original scene and nothing is abandoned.

5. Extension of Knowledge

5.1 Prior Concepts

There are lots of researches and projects about this kind of systems. Although techniques that is used can vary from one to another, generally solutions are based on only computer vision algorithms. Each have its own advantages and limitation on parameters of accuracy and computation.

There are mainly two types of methods in the literature. The first type is based on tracking [10-12]. The second type of methods use background model to detect suspected region [14-16]. The tracking-based methods encounter the problems of merging, splitting, entering, leaving, occlusion, and correspondence [13]. These problems are not easy to solve in many cases. Especially in crowded situations, it is hard to track all the objects. On the contrary, methods based on background subtraction, using an appropriate threshold procedure on the difference between each image of the sequence and a model image of the background can provide the best compromise between performance and reliability.

In addition to that, only using background subtraction method for detection requires a dedicated size of object. In other words, object should be in a defined shape, size, not necessarily to be exact but should satisfy a given threshold. It is almost impossible to consider any luggage in one type, there are lots of colors, shapes and sizes. For example, we call a musical instrument bag as luggage as well as we do for shoulder bag. In these studies, an instrument bag which is abandoned in a public space do not have a risk to become a threat. But, in 2001, a Jerusalem café was attacked by a suicide bomber carrying a bomb built into a guitar case [17]. So, it is possible for these type of luggage to be risky. Hence, this reduces the reliability of the whole system.

Besides that, machine learning is a very new topic in the field of Computer Vision. When looked at the references that we give in this report, many of them are published in last 1 or 2 years.

There are two baseline for us to solve these problem. First one's paper was written in 2017 by Liaoa et al., and other one was written by Sidyakin et al. Both papers are about real-time abandoned object detection. In Sidyakin's research, abandoned luggage is detected using Convolutional Neural Networks (CNNs), they designed their own architecture for CNN. But the precision and recall results are not satisfying to consider as a robust system. In addition to that they are not interested in finding owner of the object. In Liao's paper, the main objective is to recognize and analyze events which are related with abandoned/left objects. They did not specify their objects as luggage like we do, and also their events are consisted of "Taken by owner", "Moved by un-owner" and "Theft" cases which are different from ours. So, in this paper, main issue is not to detect abandoned objects and their threat, it is about identifying the future of the abandoned objects with the given in cases. We are planning to use these paper while associating luggage and its owner.

5.2 Innovations

Our contributions will be in the part of machine learning. We propose a method which uses deep learning convolutional neural networks in detection of abandoned luggage. Apart from that, YOLO algorithm has not been used in any paper for this purpose.

In addition to that, instead of using only YOLO and looking for every possible object if they are luggage, we decrease the region of interest with background modeling. This will help for the elimination of false positive bounding boxes which came as an output from YOLO.

6. Project Management

In the management of our project, we follow below structure to make sure that we track every step in the project.

Management activities:

While progressing in SSAL, project management is an important step of progress. Project management includes 3 steps where we can follow our progress, circumstances, project cost in terms of time and money. These steps are where we do planning, scheduling and risk management.

Project Costing:

Cost for this project is decided before we start to build the project. Project cos usually includes labor cost which is 9 months of 2 people for this project since it is an undergraduate senior project. Also, we have an equipment cost that includes only one IP camera for the project. The project doesn't include any material, travel, rent, benefit or salary cost that any software project usually includes.

Project planning

Project planning activity usually takes the most of our time while doing management of the project because it includes scheduling, schedule updates, sprint decision details and more. Steps are stated below:

- Establishing limitations for the project
- Making initial assumptions about the project
- Defining milestones
- Every sprint:
    - Making project schedule
    - Starting with the activities according to schedule
    - Reviewing progress every week
    - Revising estimations and assumptions
    - Re-negotiating project

The project is divided into monthly sprints that we can do the planning and building of the project. After defining the milestones and making the schedule we start our sprint. In a sprint, every week we have a meeting that we review our progress and revise assumptions.

Doing changes in the project:

Since we are 2 people who are working this project, we haven't used a project management tool for now. As I stated above we have weekly meetings with our advisors to discuss all our progress and decide to milestones. If we do a change, what we do is writing to a paper and discuss it in weekly meetings. So, we keep the records for changes.

We arrange a meeting in that week and discuss changes if one will be made. This meeting between project members is the main discussion on how to progress and moreover how to do changes.

Besides that, we are planning to do this changes in a formal way soon with a report, which includes the information of who did the change/s, when, how, why, so that we follow changes clearly.

Milestones of the project:

We tried to divide the project into two for each term so that we can schedule each term separately. We decided to 8 milestones for this term. Milestones determine the scheduling of the project. 8 milestones for this term are:

- Preparation
- Data set creation
- Training
- Dual background modeling
- Owner Labeling
- Luggage detection
- Event analysis
- Alarm decision

Project Scheduling

Timeline of the project is where we did the scheduling. In our schedule, we have only the milestones of the project. Below we can see the time line of the project for the first term. Also time we decided to spend on each mile stone is also indicated.

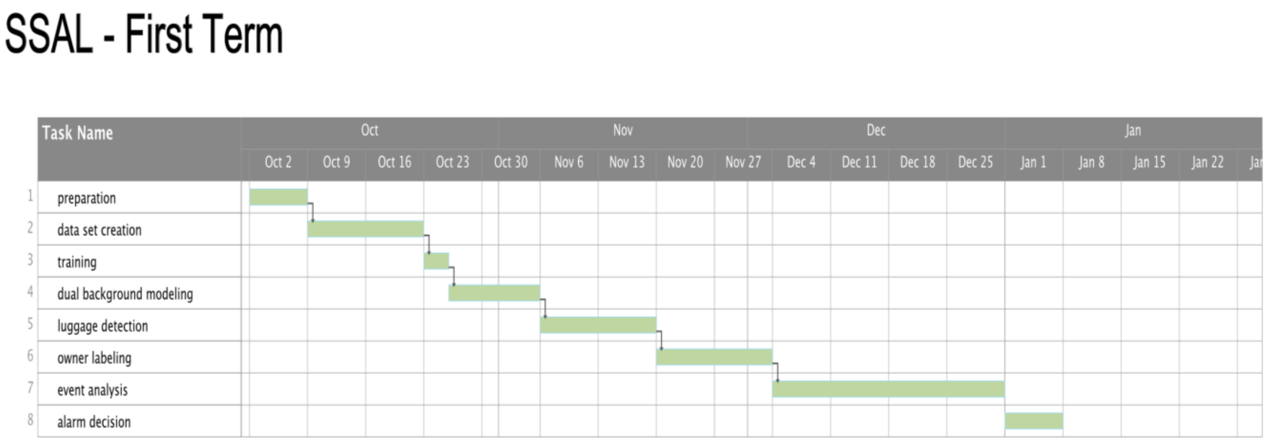

Figure 11: Gantt chart of Project

Risk Management

1) Hardware unavailability:

Identification: Hardware which is essential in our case is 1 IP camera might not be delivered on time. Also we can count that if even it is delivered, equipment has a chance to be broken.

Analysis: It is highly possible since equipment order is made by school and delivery is made by cargo.

Risk planning: We have a webcam of our own which is not an IP camera but it can be used in case equipment delivery is not made.

Risk monitoring: We still have this possibility since we didn't get the camera yet.

2) Time line estimation (Size underestimate or overestimate):

Identification: There is a possibility of stay behind of the schedule on the project.

Analysis: Risk of time line estimations to be wrong is medium chance. We sometimes have problems that we cannot see before. Also, there is chance to be faster than we think.

Risk planning: We always update the schedule, and discuss our progress every week in scrum meetings.

Risk monitoring: If faster, do estimation again and give time to tasks might take longer than you think. If slower, then give take time from tasks that might take shorter time than you think.

3) Requirements change:

Identification: There might me large amount of changes to the requirements than anticipated at the beginning.

Analysis: Chance for this is very low, since we did all our research and requirement analysis at the beginning.

Risk planning: Avoid large number of changes at the project.

Risk monitoring: We always discuss progress and even minor changes in the project development. No big changes occurred yet and chance to be occurred is still very low.

7. SCRUM DETAILS

While running this project, we come together every two days and have a meeting for evaluating our progress about what we have done so far and what we are going to do for next meeting. We think that this makes project easy to control and manage. Each meeting lasts maximum 40 minutes.

Our platform is defined as Ubuntu 16.04 since while we were using Windows, we had to add dependencies of the program for each compilation so it wasted our time. Instead, in Ubuntu it is not required, just adding libraries into program is enough to compile successfully. On the other hand, for our first attempt detection algorithm - You Only Look Once (YOLO), we planned to train our model in Darknet framework. It did not work on Windows properly because of dependencies (OpenCV and Tensorflow).

7.1 Sprint Backlog

Sprint Backlog		
Sprint Term	Description	Effort in Hours
SPRINT 1 (01.10.17 – 01.11.17)	Installation of Ubuntu 16.04 Virtual Machine on Windows and Mac (team members' laptops)	0.5 hours
	Installation of OpenCV & Tensorflow libraries to team members' laptops	1.5 hours
	Setting up cuDNN and CUDA Development Toolkit on Lab Computer	2 hours
	Installation of OpenCV & Tensorflow libraries to Lab PC	1 hour
	Creating a new luggage dataset (15000 images) from ImageNET	3 hours
	Extracting the images with the background	2 hours
	Annotation of 2500 images in the dataset	5 hours
	Installing Darknet framework and adjusting annotation file (xml) to train in YOLO algorithm	0.5 hours
	First training with small part of dataset (300 images)	16 hours

	Second training with all annotated images (2500 images)	45 hours
	Writing a luggage detection program with created model in training by using Tensorflow	3 hours
	Writing a background subtraction program	0.5
	Writing human detection program by using Support Vector Machine & Histogram of Gradient Descents algorithms	5 hours
SPRINT 2 (01.11.17 – 01.12.17)	Background Subtraction with Gaussian Mixture Model	1 hour
	Create short-term background subtraction using Zivkovic parameters for Mixture of Gaussian	1 hour
	Crate long-term background subtraction using Zivkovic parameters for long-term Mixture of Gaussian	0.5 hour
	Extracting moving foreground from each backgrounds	5 hours
	Compare proposals from background extract static object proposals	3 hours
	Testing background modeling with videos and webcam	1 hour
	Creating dataset with the images that have one-colored background	5 hours
	Annotation of new dataset	2.5 hours
	Preparing xml files to train in Faster R-CNN	1 hour
	Literature review about comparison of Faster R-CNN and YOLO	4 hours

7.3 Sprint Review

Within Sprint #2, we focused on existing projects and their implementations. While coming to end of this sprint, we are able to detect luggage in the coming frame and do background subtraction. Goal of this sprint was to analyze and criticize these projects since in order to implement a better application, we should consider their work. As promised, we looked at different approaches. This sprint makes the program flow of the project clear to follow.

7.4 Sprint Retrospective

Due to workload in other courses, we had limited time for the implementation part. However, in the previous sprint, we were able to do more than we planned, this will not cause any problem according to our plans. Naciye finished literature review for background modeling, but she will continue checking new papers or implementations. For the next sprint, we will be able to extract static objects from the frame, and send them to recognizer part. Damla continue to train the dataset with different number of images and algorithms for the next sprints.

5 ESTIMATION

Function Point Estimation

Function Points			
Program Characteristic	Low Complexity	Medium Comple	High Complexity
Number of inputs	1 x 3	1 x 4	1 x 6
Number of outputs	2 x 4	0 x 5	2 x 7
Inquiries	1 x 3	2 x 4	1 x 6
Logical internal files	2 x 7	0 x 10	0 x 15
External interface files	0 x 5	0 x 7	0 x 10
Function Point Estimation			
Unadjusted total of function points	66		
Influence multiplier	1.07		
Adjusted total of function point	70.62		

Table 4: Sprint backlog table for MTS

Calculation of Influence Multiplier:

	Rating		
1- data communications	2	11- installation ease	4
2- Distributed Functions	1	12- operational ease	5
3- Performance	3	13- Multiple sites	1
4- Heavily used configurations	1	14- Facilitate change	3
5- Transaction Rate	3	TDI	42
6- online data entry	4		
7- end user efficiency	5		
8- online updates	5		
9- complex processing	2		
10- reusability	3		
11- installation ease	4		
12- operational ease	5		

IM/ VAF = TDI * 0.01 + 0.65 = 42 * 0.01 +0.65 = 1.07 <= Influence Multiplier

ATFP = UTFP * IM = 66 * 1.07 = 70.62

Since we train our model in C, C for the Language Unit Size (LUS), which concludes us with 128 =>: most common value .

Line of Codes (LOC) = LUS * ATFP = 128 * 70.62 = 9039.36

COCOMO (Constructive Cost Model)

KDSI = Thousand delivered source instructions (KLOC)

KDSI = ATFP * Language unit size /1000 = 70.62 * 128 / 1000 = 9.03936

Since the LOC is only applicable on Basic COCOMO, we took Basic COCOMO's development mode as Semi-Detached. Hence, the following values of a, b and c were used:

a= 3.0 b = 1.12 c = 0.35

PM (Person-Month) = a * $(KDSI)^{b}$

PM = 3.0 * $(9.03936)^{1.12}$

PM = 35.318

TDEV (Total Development Time) = 2.5 * $(PM)^{c}$

TDEV = 2.5 * $(33.8755)^{(0.35)}$

TDEV = 8.70 months

6 GUI

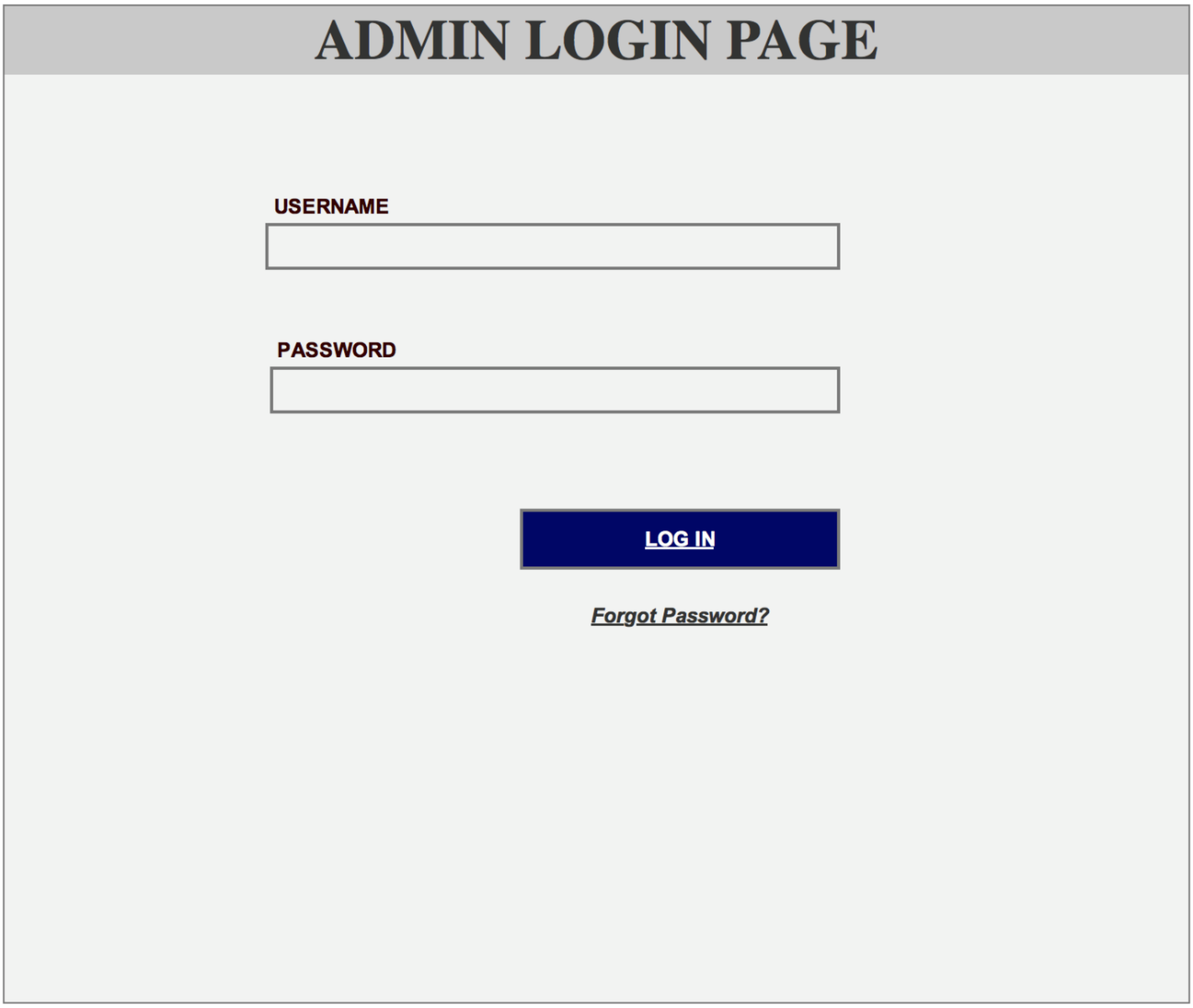

Figure 7: Login Page for Admin

GUI design for our surveillance system desktop application is planned to have multiple users. Firstly, we have should have an administrator to manage the application. An administrator should log in the system first. The page above is planned to use in the first step of our application. Admin will have a username and password to activate the system. If s/he tries to enter the system without a username and/or password, system will show an error message to the administrator. If the admin forgot the password and clicked the "Forgot Password?" button, it will send an email to registered email which is in the system. After administrator logs into the application, main page of it will show up.

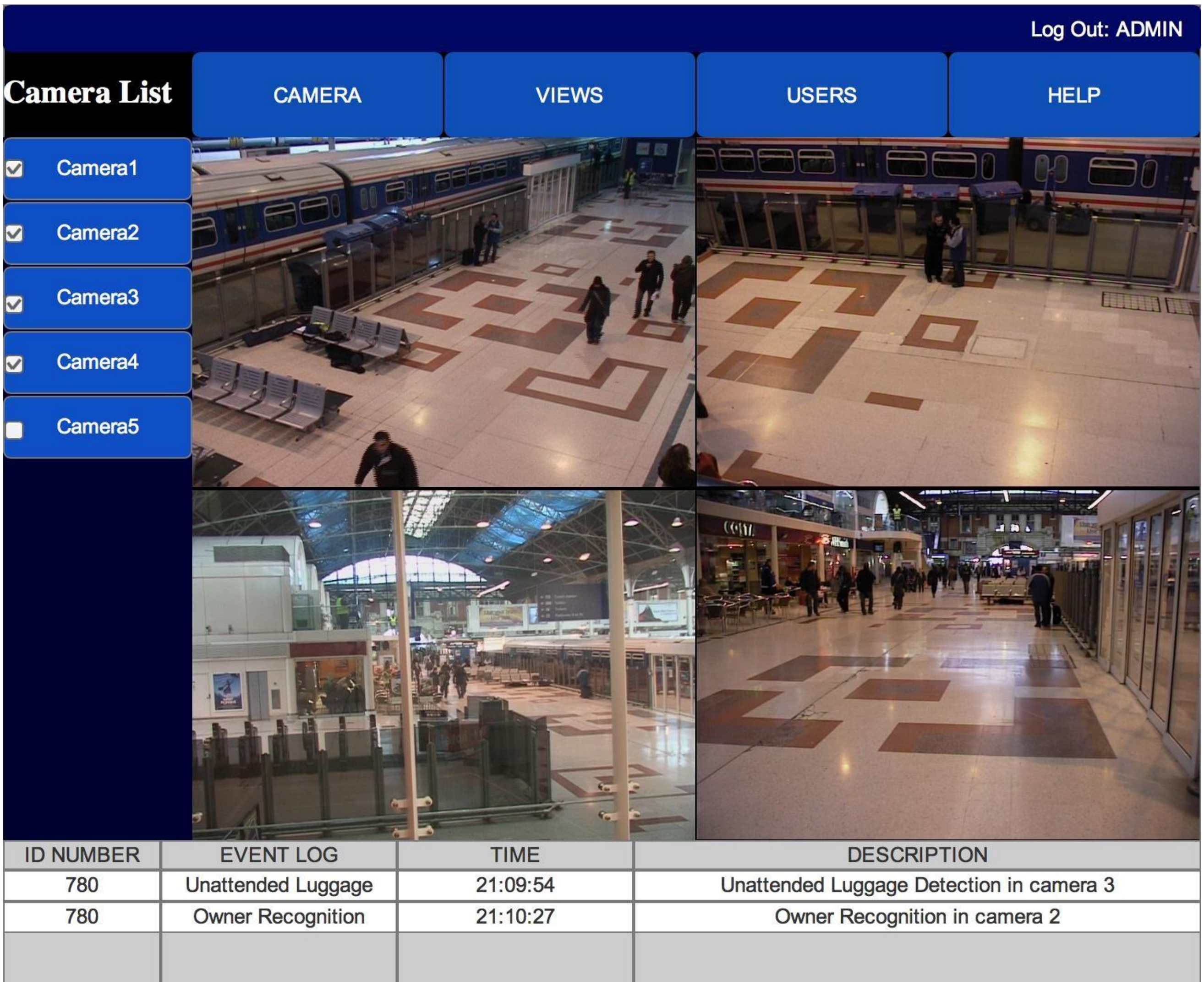

Figure 8: Main Page of Application [1]

Above figure shows the planned design for the main page of the application. In the left admin will be able to see cameras activated in the system. So, camera list basically shows the activated cameras for the security. Cameras which has a tick sign will be shown on the main page. Admin will be able to choose which cameras should be displayed.

In the below of the page, there will be event log flow. Our system will be able to detect any unattended luggage and then it will be able to find the owner also. In the figure, there is a small example to show how it will do and show this.

This flow also will assign an ID Number for luggage and its owner. Also, administrator can see the time of detection and description for the event. In the above of the main page, there will be multiple buttons for admin. Administrator will able to go camera page where s/he can handle camera management system. Below we can see camera page.

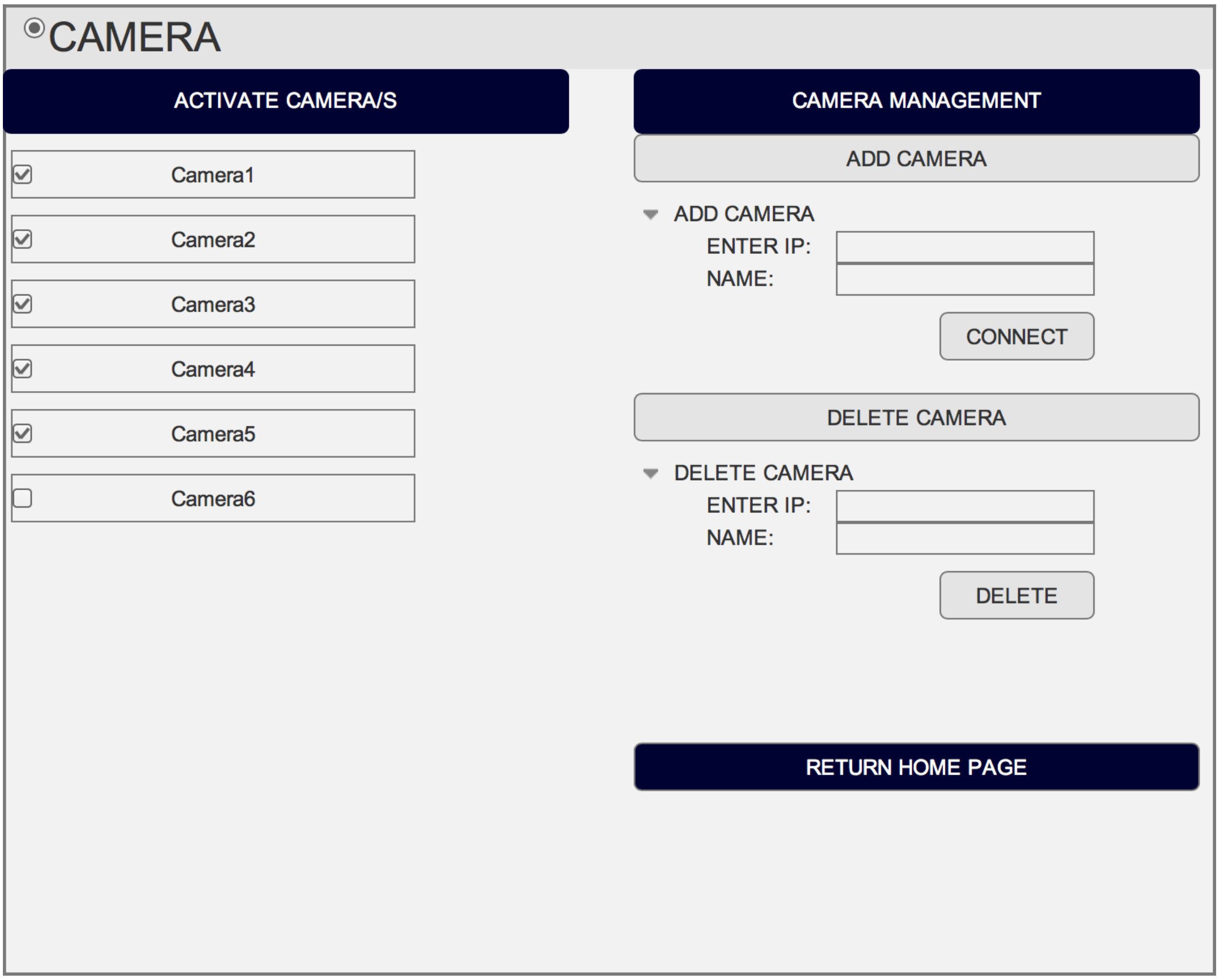

Figure 9: After clicking "Camera" link from Main Menu

In the camera management page, the administrator will be able to activate or deactivate any camera that they have. Also, s/he will be able to add a camera for the system also by IP address of the camera. Furthermore, delete camera option will be available on this page so admin also can remove any camera from the system. Admin can return the main page by clicking to the return home page button below the page.

In the main page admin will have 3 more button choices. One of them is view option. View option is to make admin able to manage camera viewing options. Admin can choose to view only one camera as well as view multiple cameras.

If admin clicks to “USERS” button, s/he will go to user management page in the application. In the user management page, application will have options to add a user. This user will not be admin; they will be security personnel who will follow the system in front of the camera. Also, admin will be able to delete any user from the system. To be clear, these users will only be able to see the video system. Since users will be actually security personnel, they will not have access to the user management or camera management pages like admins have. In addition to this, the administrator will be able to add or delete another administrator from this page also. This change will require an email verification from the new administrator. Two requirements will be requested from administrators, to choose a username and enter a password. We assume that at least one admin should be registered in the system. So if one admin will be deleted, there should be at least one more administrator other than the admin will be deleted.

Help button also available the right sight of the page. This help page will show the contact information and general information about how to use application of the system.

At the top most, there will be logged out page for the administrators. By clicking to this button, the admin will close the application and will be able to just leave.

REFERENCES

[1] Dahivelkar, S. and Borse, S. (2017). Abandoned Object Detection Based on Statistics for Labeled Regions. International Journal of Engineering Research in Electronics and Communication, 4(8), pp.2-4.

[2] Hui Kong, Audibert, J. and Ponce, J. (2010). Detecting Abandoned Objects With a Moving Camera. IEEE Transactions on Image Processing, 19(8), pp.2201-2210.

[3] Sidyakin, S. and Vishnyakov, B. (2017). Real-time detection of abandoned bags using CNN. Automated Visual Inspection and Machine Vision II, [online] 10334, pp.1-13. Available at: http://SPIEDigitalLibrary.org/conference-proceedings-of-spie [Accessed 7 Oct. 2017].

[4] Liao, W., Yang, C., Ying Yang, M. and Rosenhahn, B. (2017). SECURITY EVENT RECOGNITION FOR VISUAL SURVEILLANCE. ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences, IV-1/W1, pp.19-26.

[5] Porikli, F., Ivanov, Y., & Haga, T. (2007). Robust abandoned object detection using dual foregrounds. EURASIP Journal on Advances in Signal Processing, 2008(1), 197875.

[6] Redmon, J., Divvala, S., Girshick, R. and Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. arXiv preprint arXiv:1612.08242. [online] Available at: https://arxiv.org/pdf/1506.02640.pdf [Accessed 9 Sept. 2017].

[7] PETS 2006. (2017). PETS 2006 Benchmark Data. [online] Available at: http://www.cvg.reading.ac.uk/PETS2006/index.html#workshop-organisation
[Accessed 2 Nov. 2017].

[8] Xiao, T., Li, H., Ouyang, W. and Wang, X., 2016. Learning deep feature representations with domain guided dropout for person re-identification. arXiv preprint arXiv:1604.07528.

[9] Tian, Y. L., Feris, R., & Hampapur, A. (2008, October). Real-time detection of abandoned and removed objects in complex environments. In The Eighth International Workshop on Visual Surveillance-VS, 2008.

[10] M. Valera, S.A Velastin, "Intelligent distributed surveillance systems: a review", IEE Proceedings - Vision, Image and Signal Processing, 152(2), pp. 192-204, 2005.

[11] M.S. Nagmode, A. J. Madhuri, M. S. Ashok, “A Novel approach to Detect and Track Moving Object using Partitioning and Normalized Cross Correlation”, ICGST-GVIP Journal, 9(4), Aug. 2009.

[12] S. Ferrando, G. Gera, C. Regazzoni, “Classification of Unattended and Stolen Objects in Video Surveillance System”, Proc. of IEEE Intl. Conf. on Advanced Video and Signal-Based Surveillance (AVSS’06), pp.21, Nov. 2006.

[13] T. T. Zin, P. Tin and T. Tori, "A Probability-based Model for Detecting Abandoned Objects in Video Surveillance Systems," in Proceedings of the World Congress on Engineering, London, U.K, 2012.

[14] Auvinet, E., Grossmann, E., Rougier, C., Dahmane, M., Meunier, J. “Left-luggage detection using hymnographies and simple heuristics.” In: Proceedings of IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS), New York, pp. 51–58 (2006).

[15] Chitra, M., Geetha, M. and Menaka, L. (2013). Occlusion and Abandoned Object Detection for Surveillance Applications. International Journal of Computer Applications Technology and Research, 2(6), pp.708-713.

[16] Bhandari, P., Ratanpara, T. and K.H, D. (2017). Abandoned Object Detection Using Dual Background Model from Surveillance Videos. International Journal of Engineering and Technology, 9(3S), pp.215-223.

[17] Aman, M. (2007). Preventing terrorist suicide attacks. Sudbury, MA: Jones and Bartlett, p.21.