Specification Document

Of

Final Production Project

# Submitted by: Anan Rai

# Student Id:77356001

#### Bsc(Hons) Computing

Smart Doorbell with Enhanced Security Features

# Project Aim:

# The main objective of this project is to create an intelligent doorbell system that incorporates a facial recognition application managed for RFID access control, as well as real-time video processing. The system will also comprise an always operational secure door unlocking attempt. Future features will include break-in detection and a notification system for enhancing the security of homes.

# Objectives:

The main objective is to develop a product which:

* Develop a real-time video monitoring system.
* Implement facial recognition for secure access.
* Integrate RFID-based unlocking.
* Enable remote control and monitoring from app.
* Ensure continuous functionality with battery backup.
* Add motion detection for proactive monitoring

Secondary objectives:

* Log visitor activities
* Include tamper detection.
* Make the system weatherproof.

## Product Specification:

The product specification is categorized into functional and non-functional requirements and is described below with MoSCoW method:

|  |  |
| --- | --- |
| **Functional Requirements** | **MoSCoW** |
| Unlock the door using facial recognition | M |
| Stream real-time video to the mobile app | M |
| Enable unlocking via RFID | M |
| Maintain a log of visitor data (images and timestamps) | M |
| Trigger the camera via sensor detection | S |
| Send notifications to the mobile app | S |
| Compare more than two models at once | C |
| Measures for low-light conditions | W |
| Add future support for break-in detection | W |
| Provide remote access control for locking/unlocking | M |

|  |  |
| --- | --- |
| **Non-Functional Requirements** | **MoSCoW** |
| System works reliably with battery backup | M |
| Product is tested | M |
| User-friendly mobile app interface | S |
| Operate efficiently in low power mode | C |
| The system should be scalable for multi-door setups | W |
| Compatible with Android and iOS platforms | W |

## Research:

Object detection is a simple mechanism of identifying the locations of different objects present in an image (Lars, 2018). Neural networks are one of the leading machine learning algorithm which has been able to process, detect and optimize large datasets, also in image processing. TensorFlow is a new open source framework provided by Google which helps to model these neural networks. (Padhnis, et.al, 2018). The TensorFlow Detection API makes the process of constructing, training and deploying of the object detection model very smooth. With the API, a model is defined using a configuration files and API is responsible for structuring all other necessary elements on its own (TensorFlow, 2018). In the framework, most of the developers tend to use pre trained models to extract the features for which it is necessary to have proper experiment with different approaches. (Lars, 2018) For which this research would compare the object detection models trained specifically on MSCOCO data sets.

## Evaluation:

As defined in the product specification and objectives, the evaluation of the end product would base on the fulfillment of the objectives. For evaluation, the models would be first tested separately which means determining time, accuracy and detected classes of a single model respectively than we compare this individual result with the result we get after comparing two or more models in the product. The evaluation method would include both quantitative as well as qualitative mechanism. For instance, it would evaluate the response of the product when more than two models are given up for comparison along with their results.

## Project Planning and Methodology:

After the research, initially a particular data sets and object detection models trained on those particular data sets would be chosen. MSCOCO data sets are one of the popular training data sets which includes 80 different categories of objects and models including ssd\_inception\_v2\_coco, faster\_rcnn\_resnet101\_coco, rfcn\_resnet101\_coco and so on are trained on those data sets. After selecting the models and the datasets I would look for the way to implement any of the selected models to detect the objects in any random image. The image would consist of multiple objects present in MSCOCO data sets. Once the model has successfully detected the objects in the image, we load another model trained on the same dataset and see the results which at last are compared with the results of the previous models. The project would follow Agile methodology and the planning is further clarified using the Project Timeline and Gantt chart.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Project Timeline  Octo | **To** Nove | Dece | Janu | Feb r | Mar | Ap |
| ber | **da** mber | mber | ary | uary | ch 1 | ril |
| Start |  |  |  |  |  | Fini s |
| Mon |  |  |  |  |  | h |
| 9/24/1 |  |  |  |  |  | Sat |
| 8 |  |  |  |  |  | 4/13/ |

#### Gantt Chart:

|  |  |  |  |
| --- | --- | --- | --- |
| **Task Name** | **Duration** | **Start** | **Finish** |
| **Final Project** | **146**  **days** | **Mon 9/24/18** | **Sat 4/13/19** |
| **Feasibility Study & Planning** | **15 days** | **Mon 9/24/18** | **Fri 10/12/18** |
| Search Topic | 4 days | Tue 9/25/18 | Fri 9/28/18 |
| Research Project Scope | 6 days | Sat 9/29/18 | Fri 10/5/18 |
| Requirement gathering | 5 days | Sat 10/6/18 | Thu 10/11/18 |
| **Initiation & Planning** | **9 days** | **Fri 10/12/18** | **Wed 10/24/18** |
| Initial Research & Planning | 5 days | Fri 10/12/18 | Thu 10/18/18 |
| Develop Project Specification | 3 days | Fri 10/19/18 | Tue 10/23/18 |
| Submit  Specification Document | 1 day | Wed 10/24/18 | Wed 10/24/18 |
| **Research** | **20 days** | **Thu 10/25/18** | **Wed 11/21/18** |
| Literature Review | 10 days | Thu  10/25/18 | Wed 11/7/18 |
| Research Methodology | 10 days | Thu 11/8/18 | Wed 11/21/18 |
| **Development Phase** | **65 days** | **Thu 11/22/18** | **Wed 2/20/19** |
| Design | 14 days | Thu 11/22/18 | Tue 12/11/18 |
| Develop | 42 days | Wed 12/12/18 | Thu 2/7/19 |
| Documentation | 9 days | Fri 2/8/19 | Wed 2/20/19 |
| **Implement** | **3 days** | **Thu 2/21/19** | **Mon 2/25/19** |
| **Testing & Evaluation** | **32 days** | **Tue 2/26/19** | **Wed 4/10/19** |
| White Box Test | 10 days | Tue 2/26/19 | Mon 3/11/19 |
| Black Box Test | 10 days | Tue 3/12/19 | Mon 3/25/19 |
| Track & Solve Issues | 12 days | Tue 3/26/19 | Wed 4/10/19 |
| **Closing** | **2 days** | **Thu 4/11/19** | **Fri 4/12/19** |
| Submission Of Final Product | 1 day | Thu 4/11/19 | Thu 4/11/19 |
| Presentation | 1 day | Fri 4/12/19 | Fri 4/12/19 |

Fig: *The above Gantt chart shows the project planning along with the key dates.*

## Resources:

The list of resources (hardware and software) are listed below. All the software are free open source products and can be easily downloaded online.

### List of Software:

* + Protobuf 3.0.0
  + Python-tk
  + Pillow 1.0
  + lxml
  + Matplotlib
  + Eclipse
  + Java JDK
  + Ubuntu 16.04
  + Tensorflow (>=1.9.0)
  + Cython
  + contextlib2
  + cocoapi

### List of Hardware:

* Lenovo Y40 Laptop  AMDA Radeon Graphic

## Bibliography:

* *"Speed/accuracy trade-offs for modern convolutional object detectors." Huang J, Rathod V, Sun C, Zhu M, Korattikara A, Fathi A, Fischer I, Wojna Z, Song Y, Guadarrama S, Murphy K, CVPR 2017*
* *Tensorflow API Documenation using Python. Retreived from* [*www.tensorflow.org/api\_docs/python/*](http://www.tensorflow.org/api_docs/python/)
* *‘Objects Talk - Object Detection and Pattern Tracking Using TensorFlow’ (2018) 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT), Inventive Communication and Computational Technologies (ICICCT), 2018 Second International Conference on, p. 1216. doi: 10.1109/ICICCT.2018.8473331.*
* *Mustamo P., “Object detection in sports: TensorFlow Object Detection API case study”, University of Oulu, Degree Programme in Mathematical Sciences. Bachelor’s Thesis, 2018, 43.*
* *‘Performance of Some Image Processing Algorithms in Tensorflow’ (2018) 2018 25th International Conference on Systems, Signals and Image Processing (IWSSIP), Systems, Signals and Image Processing (IWSSIP), 2018 25th International Conference on, p. 1. doi: 10.1109/IWSSIP.2018.8439714.*
* *‘Data classification with deep learning using Tensorflow’ (2017) 2017 International Conference on Computer Science and Engineering (UBMK), Computer Science and Engineering (UBMK), 2017 International Conference on, p. 755. doi: 10.1109/UBMK.2017.8093521*
* *‘Evaluating Training Time of Inception-v3 and Resnet-50,101 Models using TensorFlow across CPU and GPU’ (2018) 2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA), Electronics, Communication and Aerospace Technology (ICECA), 2018 Second International Conference on, p. 1964. doi: 10.1109/ICECA.2018.8474878.*
* *Wilson, R. (Writer on machine learning) (2017) Machine learning for complete beginners: a visual beginners guide to machine learning with Python, data science, TensorFlow, artificial intelligence, random forests and decision trees. [S. l.] : [R. Wilson?], c2017. Available at:* [*http://ezproxy.leedsbeckett.ac.uk/login?ur*](http://ezproxy.leedsbeckett.ac.uk/login?ur)[*l=http://search.ebscohost.com/login.aspx?direct=true&d*](http://search.ebscohost.com/login.aspx?direct=true&db=cat0)[*b=cat0*](http://search.ebscohost.com/login.aspx?direct=true&db=cat0) *0621a&AN=leeds.707650&site=eds-live (Accessed: 18 October 2018).*
* *‘Object recognition in images using convolutional neural network’ (2018) 2018 2nd International Conference on Inventive Systems and Control (ICISC), Inventive Systems and Control (ICISC), 2018 2nd International Conference on, p. 718. doi: 10.1109/ICISC.2018.8398912.*