Style Guidelines for Final Year Project ReportsDeep Fake: Voice Cloning

Final Year Project – System Design 1

Session 2020-2024

A 4th Year Student

A project submitted in partial fulfilment of the

COMSATS University Degree

of

BSc. (Hons.)BS in Computer Science / Software Engineering (CUI)



Department of Computer Science

COMSATS University Islamabad, Lahore Campus

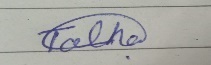
02 December 2023

**Project Detail**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Project ID (for office use) | | |  | | | | |
| Type of project | | | [ ] Traditional [●] Industrial [ ] Continuing | | | | |
| Nature of project | | | [●] **D**evelopment [ ] **R**esearch [ ] **R**&**D** | | | | |
| Area of specialisation | | | Deep Learning | | | | |
| **Project Group Members** | | | | | | | |
| Sr.# | Reg. # | Student Name | | CGPA | Email ID | Phone # | Signature |
| (i) | Fa20-bcs-007 | Muhammad Talha | | 2.94 | mt559310@gmail.com | 0322-7323128 |  |
| (ii) | Sp20-bcs-114 | Hassan Mehmood | | 2.86 | SP20-BCS-114@gmail.com | 0309-4644191 |  |

# Plagiarism Free Certificate

This is to certify that, I am **Muhammad Talha** S/D/o **Falak Sher**, group leader of FYP under registration no **CIIT/fa20-bcs-007/LHR** at Computer Science Department, COMSATS Institute of Information Technology, Lahore. I declare that my FYP proposal is checked by my supervisor and the similarity index is **7%** that is less than **20%,** an acceptable limit by HEC. Report is attached herewith as Appendix A.

Date: **2/10/2023** Name of Group Leader: **Muhammad Talha** Signature: 

Name of Supervisor: **Dr Wajahat Mehmood Qazi**

Designation: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Abstract

This project aims to develop an application that can generate voice clones based on provided text using a trained model. Many Text-to-Speech models exist, they are limited to specific voices **1** Our focus is on training a model with an English dataset to replicate a particular person's voice. Once trained, the model can produce Deepfake voices for the provided text, offering advantages in education, cost-effective advertising, and the media industry. However, it's essential to acknowledge the potential for misuse, especially in serious crimes and impersonation, necessitating subsequent efforts for detection and prevention.This project's primary objective is Deepfake text to speech creation, with ethical and security considerations in mind. **[2]**

Table of Contents

[Abstract 3](#_Toc36983214)

[Table of Contents 5](#_Toc36983215)

[List of Tables 8](#_Toc36983216)

[List of Figures 10](#_Toc36983217)

[Chapter 1: Introduction 11](#_Toc36983218)

[1 Introduction 12](#_Toc36983219)

[1.1 Introduction 12](#_Toc36983220)

[1.2 Goals and Objectives 13](#_Toc36983221)

[1.3 Problem Statement 14](#_Toc36983222)

[1.4 Assumptions and Constraints 15](#_Toc36983223)

[1.5 Project Scope 15](#_Toc36983224)

[Chapter 2: Requirement Analysis 17](#_Toc36983225)

[2 Requirement Analysis 18](#_Toc36983226)

[2.1 Literature Review 18](#_Toc36983227)

[2.1.1 Literature Review: Real-World Anomaly Detection in Surveillance Videos (Waqas Sultani *et al.*) [1] 18](#_Toc36983228)

[2.1.2 Literature Review: Anomaly Locality in Video Surveillance (Federico Landi *et al.*) [2] 20](#_Toc36983229)

[2.1.3 Literature Review: T-C3D Temporal Convolutional 3D Network for Real-Time Action Recognition (Kun Liu *et al.*) [3] 21](#_Toc36983230)

[2.1.4 Literature Review: Action Recognition with 3D ConvNet-GRU Architecture (Guangle Yao *et al.*) [4] 23](#_Toc36983231)

[2.1.5 Literature Review: Learning Spatiotemporal Features with 3D Convolutional Networks (Du Tran *et al.*) [5] 24](#_Toc36983232)

[2.2 Stakeholders list: 26](#_Toc36983233)

[2.3 Product Functions: 26](#_Toc36983234)

[2.3.1 User characteristics: 27](#_Toc36983235)

[2.4 Requirements Elicitation: 27](#_Toc36983236)

[2.4.1 Functional requirements: 28](#_Toc36983237)

[2.4.2 Non-Functional requirements: 29](#_Toc36983238)

[2.5 Use Case Description: 32](#_Toc36983239)

[2.6 Use Case Diagrams: 35](#_Toc36983240)

[2.6.1 Use Case ID: 1 35](#_Toc36983241)

[2.6.2 Use Case ID: 2 36](#_Toc36983242)

[2.6.3 Use Case ID: 3 37](#_Toc36983243)

[2.6.4 Use Case ID: 4 38](#_Toc36983244)

[Chapter 3: System Design 39](#_Toc36983245)

[3 System Design: 40](#_Toc36983246)

[3.1 Work Breakdown Structure: (WBS) 40](#_Toc36983247)

[3.2 Activity Diagram: 41](#_Toc36983248)

[3.2.1 Login: 41](#_Toc36983249)

[3.2.2 Anomaly Detection: 42](#_Toc36983250)

[3.2 Sequence Diagram: 43](#_Toc36983251)

[3.2.1 Login: 43](#_Toc36983252)

[3.2.2 Anomaly Detection 44](#_Toc36983253)

[3.3 Software Architecture: 45](#_Toc36983254)

[3.4 Collaboration Diagram: 46](#_Toc36983255)

[3.5 Project Code 46](#_Toc36983256)

[Chapter 4: System Testing 48](#_Toc36983257)

[4 System Testing 49](#_Toc36983258)

[4.1 Test Cases 49](#_Toc36983259)

[4.2 Unit Testing 53](#_Toc36983260)

[4.2.1 Bugs 53](#_Toc36983261)

[4.3 Acceptance Testing 53](#_Toc36983262)

[Chapter 5: Application Front-End 54](#_Toc36983263)

[5 Application Frontend 55](#_Toc36983264)

[5.1 Android Interface 55](#_Toc36983265)

[5.1.1 Login Screen 55](#_Toc36983266)

[5.1.2 Home Screen 56](#_Toc36983267)

[5.1.3 Alert 57](#_Toc36983268)

[5.1.4 Personal Setting 58](#_Toc36983269)

[5.1.5 Camera Settings 59](#_Toc36983270)

[5.1.6 Clip History 60](#_Toc36983271)

[5.1.7 Sign Up 61](#_Toc36983272)

[5.2 Web Interface 62](#_Toc36983273)

[5.2.1 Login 62](#_Toc36983274)

[5.2.2 Home 62](#_Toc36983275)

[5.2.3 Clip History 63](#_Toc36983276)

[5.2.4 Personal Settings 63](#_Toc36983277)

[Chapter 6: Conclusion 64](#_Toc36983278)

[6 Conclusion 65](#_Toc36983279)

[6.1 Problems Faced and Lessons Learned 65](#_Toc36983280)

[6.2 Project Summary 66](#_Toc36983281)

[6.3 Future Work 66](#_Toc36983282)

[Chapter 7: Reference 67](#_Toc36983283)

[7 References 68](#_Toc36983284)

List of Tables

[Table 1: Results of the proposed anomaly detection approach using C3D 19](#_Toc36982984)

[Table 2(User Characteristics) 27](#_Toc36982985)

[Table 3(FR01 - Log in) 28](#_Toc36982986)

[Table 4(FR02 – Anomaly Detection) 28](#_Toc36982987)

[Table 5(FR03 – Video Database) 29](#_Toc36982988)

[Table 6(NFR01 - Performance) 29](#_Toc36982989)

[Table 7(NFR02 - Usability) 30](#_Toc36982990)

[Table 8(NFR 03 - Reusability) 30](#_Toc36982991)

[Table 9(NFR 04 – Defect Maintenance) 30](#_Toc36982992)

[Table 10(NFR05 - Completeness) 31](#_Toc36982993)

[Table 11(NFR06 - Security) 31](#_Toc36982994)

[Table 12(NFR07 - Extensibility) 31](#_Toc36982995)

[Table 13(Use case - 1) 32](#_Toc36982996)

[Table 14(Use case - 2) 32](#_Toc36982997)

[Table 15(Use case - 3) 33](#_Toc36982998)

[Table 16(Use case - 4) 34](#_Toc36982999)

[Table 17(Test Case - 1) 49](#_Toc36983000)

[Table 18(Test Case - 2) 49](#_Toc36983001)

[Table 19(Test Case - 3) 50](#_Toc36983002)

[Table 20(Test Case - 4) 51](#_Toc36983003)

[Table 21(Test Case - 5) 51](#_Toc36983004)

[Table 22(Test case - 6) 52](#_Toc36983005)

[Table 23(Test Case - 7) 52](#_Toc36983006)

List of Figures

[Figure 1:The flow diagram of the proposed anomaly detection apporach 17](file:///C:\Users\Shoaib%20Kkan\Desktop\Maroof\FYP-II%20Report\GIDEON_REPORT_FYP.docx#_Toc36983021)

[Figure 2 The propsed T-C3D architecture for real-time video action recognition 21](#_Toc36983022)

[Figure 3(Use case - 1) 34](#_Toc36983023)

[Figure 4(Use case - 2) 35](#_Toc36983024)

[Figure 5(Use case - 3) 36](#_Toc36983025)

[Figure 6(Use case - 4) 37](#_Toc36983026)

[Figure 7(Work Breakdown Structure) 39](#_Toc36983027)

[Figure 8(Activity - 1) 40](#_Toc36983028)

[Figure 9(Activity - 2) 41](#_Toc36983029)

[Figure 10(Sequence- 1) 42](#_Toc36983030)

[Figure 11(Sequence - 2) 43](#_Toc36983031)

[Figure 12(Architecture) 44](#_Toc36983032)

[Figure 13(Collaboration) 45](#_Toc36983033)

[Figure 14(User Interface - Android - 1) 54](#_Toc36983034)

[Figure 15(User Interface - Android - 2) 55](#_Toc36983035)

[Figure 16(User Infercase - Android - 3) 56](#_Toc36983036)

[Figure 17(User Interface - Android - 4) 57](#_Toc36983037)

[Figure 18(User Interface - Android -5) 58](#_Toc36983038)

[Figure 19(User Interface - Android - 6) 59](#_Toc36983039)

[Figure 20(User Interface - Android - 7) 60](#_Toc36983040)

[Figure 21(User Interface - Web - 1) 61](#_Toc36983041)

[Figure 22(User Interface - Web - 2) 61](#_Toc36983042)

[Figure 23(User Interface - Web - 3) 62](#_Toc36983043)

[Figure 24(User Interface - Web - 4) 62](#_Toc36983044)

Chapter 1: Introduction

# Introduction

## Introduction

Nowadays, Intelligent Video Analytics (IVA) has become an intensive research domain, under the area of Computer Vision, Machine Learning and Artificial Intelligence. This has become possible as more and more AI accelerated hardware is becoming available. Conventional CCTV surveillance systems involve continuous human labour of visual focus on events happening and they also do not generate any automated alerts/alarms against suspicious/abnormal activity, own their own, if the human supervision is not present. Then the fact that humans get tired, cannot remain steadily active entire day long, require work breaks, may get distracted, etc. In some cases, such as homes, it is not even feasible to have human supervision available all the time, round the clock. Such windows of time are simply Down-Time or System Unresponsive.

In light of the above mentioned, **Gideon**; our Final Year Project is AI on the Edge, powered by Deep Learning frameworks and Computer Vision, capable of Real-Time Intelligent Video Analytics (IVA), trained for the detection of suspicious/abnormal activities, in context of Security Surveillance. It can be deployed in a plug-n-play manner with a currently installed CCTV system, seamlessly analyzing and extracting the exact video frames of flagged segments, logging them in system application and generating a relevant alert/intimation through its connected Mobile and Web Interfaces. This will make the entire process of Surveillance automated, eliminating the need for constant human attention to monitoring.

The recent advancements in specialized fields of Artificial Intelligence, such as Machine Learning, Computer Vision, Artificial Neural Networks and Inference based on Video Analytics (Live Stream) have been phenomenal. The nature of their computational workload is highly parallel in nature, such as vectors and tensors processing. On the other hand, the architecture of the Graphics Processing Unit (GPU) is SIMD (Single Instruction Stream, Multiple Data Streams), involving hundreds to thousands of Floating Point Units (FPUs), that NVIDIA refers to as CUDA cores.AMD refers to the same as Streaming Processors (SM).

In conjunction with the increasing density of CUDA Cores per GPU, decreasing the die-size of Processor Chips, it has now resulted in greater Computational Cores per monetary unit and volume. This phenomenon has made it practically possible, to develop and deploy Deep Neural Network-based applications that can perform Inference, on the Edge in embedded form-factor and harness the power of parallel computing to exploit them in endless possibilities. Surveillance and Security monitoring are turning out to be merely one of them.

Hence, with the development of Gideon; a theoretically and practically never-tiring, emotions-less electronic computer system, will be running a Deep Neural Network-based pre-trained model, to observe, infer and log suspicious/abnormal activities. Gideon will report a potential security breach or a threatening situation, through an automated system using its Mobile and Web interface, making the entire process of a Surveillance System, highly reliable, efficient and eliminating constant human dependency.

## 

## Goals and Objectives

* + 1. **Goals:**

1. Training 3D Deep Neural Network using UCF-Crime dataset, using Transfer learning paradigm.
2. Accurate identification of Suspicious/Abnormal activities, using Deep Learning and Computer Vision on a pre-trained model.
3. Frame extraction, labelling and logging in structured manner in an activity database of Intelligent Video Analytics (IVA) application.
4. Suspicious activity video-segment capture of the above.
5. Designing, building and deploying Mobile and Web interfaces for providing GUI of the Analytics system to user-end.
6. Alerts/trigger system design, over the already designed GUI interface, for rapid intimation to the End User of the IVA application.
   * 1. **Objectives:**
7. AI on the Edge using NVIDIA’s Jetson NANO Embedded Platform, instead of Cloud for Inference, to eliminate Network dependency, bandwidth and latency issues and recurring cost of using such resources.
8. Near Real-time response of the system, through Alarms/Alerts generation.
9. Elimination of constant human dependency to observe Anomalous Events.
10. Automatic Logging of periodic Anomalous-Events data (frames and video segments) and access for later use.
11. Providing interactive and user-friendly Mobile and Web application GUIs that are convenient to use even for a common user.

## Problem Statement

Conventional CCTV security systems require constant monitoring to detect any suspicious activities, thus making it highly dependent on human attention and focus. The main activities performed by these systems are to constantly record the scene using DVR whereas the rest of the security work such as monitoring or crime detection is expected from the person using that system. The entire reliability of such a system is limited by the attentiveness of its human supervisor which by the definition makes the system unreliable because it is a known fact that humans are not perfect. To eliminate these limitations, our project Gideon; will monitor by analysing the live video stream of the CCTV security system and generate rapid alarm against any suspicious/abnormal activity.

## Assumptions and Constraints

For our system, we have assumed the definition of suspicious activity by defining it on the bases of those 13 types (robbery, abuse, arson, arrest, accident, assault, burglary, shooting, fighting, explosion, vandalism, stealing, and shoplifting) of crimes that our dataset’s (UCF-CRIME) [1] videos are categorized. Since our system will perform binary classification, therefore, we are generalizing all those 13 types of crimes and labeling them as abnormal and labeling the 14th type (normal activity videos) as normal.

As our project will deal with live camera feed, therefore, it will produce a slight delay (of few milliseconds) informing the user about the suspicious activity because of the time consumed at preprocessing the live footage and feeding it to our inferencing model.

## Project Scope

Our main idea finds its roots in the very fact that the conventional setup and infrastructure based on a single/network of CCTV cameras deployed anywhere has to be constantly monitored by a human(s) observer. It's laborious. Also in some cases, most of them rather, it is not feasible to have a human observer monitoring all the time, such as in small homes. Or consider the case Punjab Safe Cities Authority (PSCA) where 8000+ cameras are installed already in the city of Lahore, Pakistan. With many more to be installed gradually, and it's ambit to expand to other cities in the province of Punjab as well, this will demand a huge human workforce to be deployed for such integrated monitoring and yet it will not be the feasibility to constantly monitor cameras will be covered simultaneously all the time.

So there exists a comprehensively wide window of the time that is fully exposed to going undetected for abnormal and suspicious activities. For instance, someone attempting to trespass one's property, or there is shooting happening somewhere in the city, but unfortunately, no one attentive at the CCTV Cam at that moment, to take rapid action, although the activity is being recorded. So practically there is not much use in recording such activities alone if an immediate proper action cannot be taken on them and collateral damage is avoided or controlled.

Our project, aims at providing cost feasible, computationally capable solution with AI on the Edge, by providing embedded hardware solution powered by Deep Learning and Computer Vision, that could be easily interfaced with current CCTV camera(s) in the market and provide the capability of Intelligent Video Analytics (IVA) detecting abnormal or suspicious activities. Therefore, any company that uses surveillance as part of its security system can use our system to automate the detection of suspicious activity on their premises

The embedded solution in our case is NVIDIA Jetson NANO; a 99$ Heterogeneous Computing platform with 128 Maxwell Architecture CUDA Cores for Parallel Computing, with a quad-core ARM Cortex A57 application processor @ 1.43 GHz, capable of real-time Inference with a pre-trained Neural Network application.

Therefore, infrastructure replacement will be not being required. Only an additional component(s) will be mounted. Secondly, AI on the Edge will run a pre-trained Neural Network, without the need to use a GPUs based Computer Server, remotely on Cloud. Hence, no network dependency for video analytics and inference, making surveillance system responsive in Real-Time constraints.

Chapter 2: Requirement Analysis

# Requirement Analysis

## Literature Review

### Literature Review: Real-World Anomaly Detection in Surveillance Videos (Waqas Sultani *et al.*) [1]

The paper in subject here proposes and implements a Deep Learning based approach for the detection of real world anomalies in surveillance videos, by exploiting both the Abnormal and Normal class of videos in training/testing dataset, using weakly labelled training videos at video level, instead of clip-level labels, over their own constructed diverse dataset; UCF-Crime.

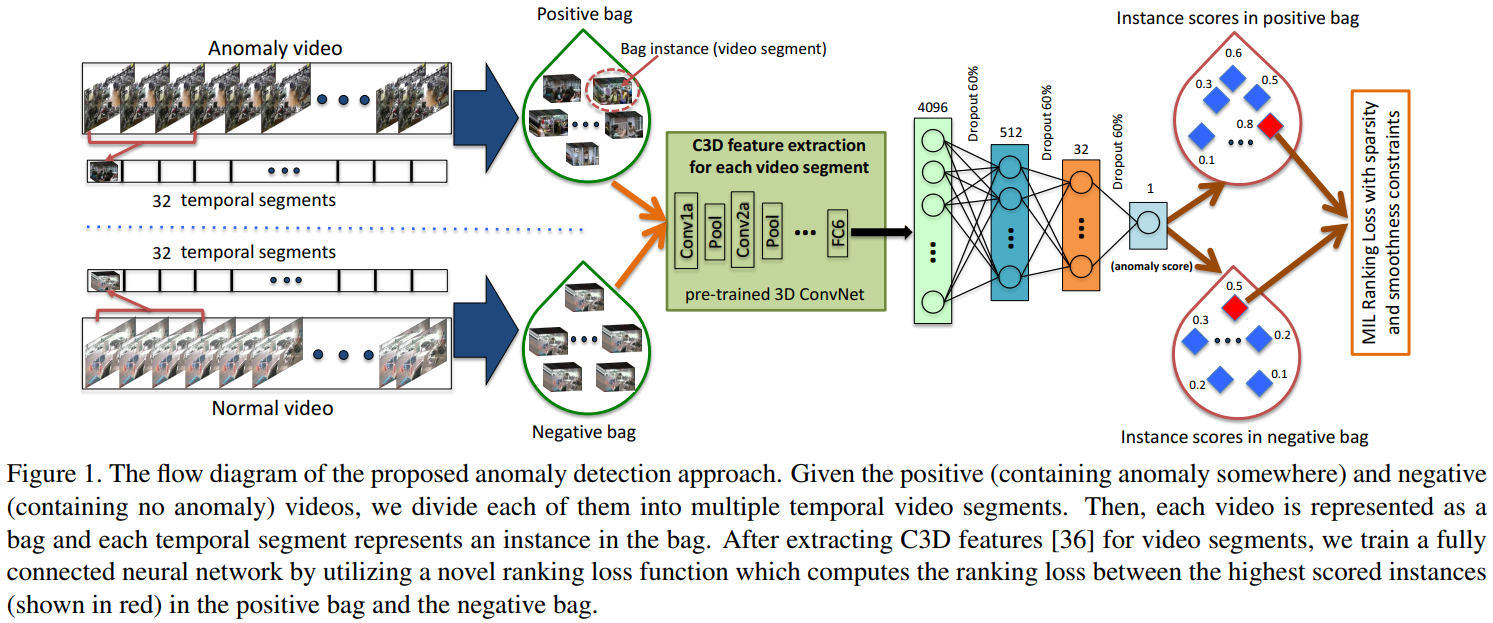
The authors propose and make use of a combination of two methodological approaches for Anomaly Detection in a real world video. Surveillance videos are divided into a fixed-size and number of segments in training phase where the segments form instances inside positive (anomalous) and negative (normal) video bags. The model is trained using (1) Multiple Instance Learning and (2) Deep MIL Ranking Model g, Loss which trains the model to automatically learn a deep anomaly ranking model that predicts high anomaly scores for anomalous video segments inside a video. For better localization of anomaly, sparsity and temporal smoothness constraints are introduced in the ranking loss function during model training.

Figure 1:The flow diagram of the proposed anomaly detection apporach

The authors introduce a new, comprehensively large-scale data-set containing 128 hours of videos, consisting of 1900 long and untrimmed videos of real-world surveillance comprising of 13 realistic anomalies such as fighting, road accidents, burglary, vandalism etc. and normal activities as well. The data-set can be used for both 2 class detection (anomalous/normal) and multi-class (class specific anomaly) detection.

For each segment of a given video, visual features are extracted from the fully connected (FC) layer FC6 of the C3D network, over re-sized video frames of 240 x 320 pixels and frame rate of 30fps, for each 16-frame segment followed by l2 normalization. The average-value of these features (4096D) is made input later to a 3-layer (512units/32units/1unit) FC neural network, with 60% dropout regularization between the layers. Activation functions ReLU and Sigmoid are used for first and last FC layers, employing Adagrad optimizer and initial learning rate of 0.001 while the sparsity and smoothness constraints in MIL ranking loss are set to λ1 = λ2 = 8 x 10-5. Frame based Receiver Operating Characteristic (ROC) curve and corresponding Area Under the Curve (AUC) metrics are used for the performance evaluation of the methodology.

The experimental results on the author’s newly constructed data-set show that their proposed Anomaly Detection approach performs significantly better than the baseline methods:

Table 1: Results of the proposed anomaly detection approach using C3D

|  |  |
| --- | --- |
| Method | AUC |
| Binary Classifier | 50.0 |
| Hasan *et a*l. | 50.6 |
| Lu *et al.* | 65.51 |
| Proposed w/o constraints | 74.44 |
| Proposed with constraints | 75.41 |

### Literature Review: Anomaly Locality in Video Surveillance (Federico Landi *et al.*) [2]

The paper in subject here proposes and explores the implementation of locality in anomaly detection of real-world videos by enriching a considerable portion of an existing dataset with spatial and temporal domains with annotations, using spatiotemporal tubes instead of whole-frame video segments. Experimental outcomes prove that the network trained this way outperforms its analogous models that are trained with only whole-frame videos.

The dataset this way, is first of its kind with bounding box supervision in both Test and Train sets. For a given input clip, it is determined whether the observed scene is normal or anomalous, levering the proposed concept of Tube Extraction for precise locality of anomaly occurrence. The model continuously outputs the probability of an unusual event taking place, with a number in range of 0 to 1. This way the granularity of video analysis changes from full-frame to spatiotemporal tubes.

The model for such consists of: (1) – A tube extraction module, (2) – A video encoder and (3) – A regression network.

Tube Extraction is implemented as a composition of crop and resize function for each frame. Input is of the form 16 frames, 224 x 224 x RGB resolution; outputs a set of 4 coordinates, marking a locality bounding box per frame.

Video Encoder implements I3D based 3D kernels to encode information from video input, to obtain visual representation of encoded action information. Also combined is; two-stream approach for appearance and motion information.

The regression network outputs anomaly score in the probability range of 0 to 1, for the investigated tube composed of input volume of encoded information for appearance and motion. 1x1 convolution is applied with 64 kernels, with output dimensions of resulting FC layers as 1024, 256, 64, 1. ReLU activation and 50% dropout regularization is adopted between FC layers.

To compensate for lacking labelled data, the authors enrich a portion of UCF-Crime dataset with Spatiotemporal annotations for 6 out of 13 anomalous categories present in the dataset, with key attention to human-based anomalies; Arrest, Assault, Burglary, Robbery, Stealing and Vandalism and 100 selected videos of each.

Experimental results show and confirm: (1) – Locality helps better accuracy in anomaly detection, (2) – the newly proposed method is more robust during testing, to different kinds of localization errors and (3) – can also provide spatiotemporal suggestions over a potentially larger set of unseen videos, that can be valuable in training a new weakly-supervised network.

### Literature Review: T-C3D Temporal Convolutional 3D Network for Real-Time Action Recognition (Kun Liu *et al.*) [3]

The paper in subject here proposes new network architecture Temporal Convolutional 3D Network (T-C3D), capable of Real-Time Action Recognition in videos, owing to its optimized network complexity and consequently much lesser computational-cost requirements. This end-to-end trainable framework exploits Temporal Encoding Method and Deep 3D-CNND to learn and train on overall temporal information of a video that improves the action recognition performance considerably.

The methodology adopted by the authors to deploy T-C3D, makes the network to learn video action representations in a hierarchical multi-granularity manner. A residual 3D Convolutional Neural Network captures complementary information of single frames, which is combined with newly formulated Temporal Encoding, to learn temporal dynamics of the entire video.



Figure 2 The proposed T-C3D architecture for real-time video action recognition

Replacement of heavier feature calculation methods (such as optical flow or IDT) with a 3D-CNN models the short-term motion and makes the framework extremely faster as it only requires RGB-Channel frames. Along with, the Temporal Encoding method characterizes the overall information of the entire video, boosting the classification accuracy drastically. T-C3D updates and optimizes its parameters through video-level score rather than the clip-level prediction, making use of aggregate functions such as Average Pooling, Max Pooling, Weighted Pooling and Attention Pooling. A key observation in this entire process is; pre-training the 3D-CNN on a rather clean but smaller data-set is more critical, than a large but a noisy data-set, to improve performance of the network.

The proposed T-C3D network was empirically evaluated on two public benchmark datasets for action recognition, (1) – UCF 101 (101 action categories, 13,320 videos of about 27 hours, 320x240 spatial resolution and avg. 25fps) and HMDB51(51 action categories, 6,766 videos). For both datasets, three standard training/testing splits are adopted and mean accuracy is reported.

Comparison with state-of-the-art methods over UCF-101 and HMDB-51 for mean-accuracy over 3 splits shows competitive performance, without requiring heavy computational cost requirements.

### Literature Review: Action Recognition with 3D ConvNet-GRU Architecture (Guangle Yao *et al.*) [4]

The paper in subject here proposes Gated Recurrent Unit (GRU) based 3D ConvNet architecture that models Temporal Dynamics of action, by imposing Res3D (on RGB) and optical flow (on clips) to extract Deep Spatiotemporal information. Then, employing GRU on the retrieved spatiotemporal features, and modelling temporal evolution for a better and enhanced action recognition.

The authors make use of Res3D, a 3D ConvNet architecture inside of residual learning framework, for the extraction of deep Spatiotemporal information from videos that achieves best performance than other 3d ConvNets so far.

The proposed approach of 3D ConvNet-GRU starts with splitting videos into non-overlapped clips, fine-tuning the Res3D with a pre-trained model. Later, 512-dimensional pool5 features are extracted instead of 50288-dimensional res5b features for action recognition, as this achieves a balance of accuracy vs dimension which is better than res5b. Max of 32 features are selected for each action and sorted with timestamp. Following that, dynamic GRU is performed and temporal evolution of actions is modelled, producing prediction for each time step using a Softmax classifier.

The optical flow carries the information relating to the motion in actions and is made input for both 2D and 3D ConvNets for action recognition. In contrast to others (Simonyan et al., Wang et al.), the authors propose fine-tune 3D ConvNet on optical flow using the knowledge learned from the RGB video dataset.

Following the extraction of deep spatiotemporal information using RGB Res3D and Optical Flow Res3D, both features are fused using summation method that happens to have lower dimension advantage over concatenation method and eventually, the GRU is performed over fused features input.

The authors evaluated their proposed method on two key and challenging action recognition datasets so far: 1) – UCF 101 (101 action categories, 13,320 videos of about 27 hours, 320x240 spatial resolution and avg. 25fps) and HMDB51(51 action categories, 6,766 videos), both datasets having 3 test/train splits.

The experimental results confirm that the modelling of the Deep Spatiotemporal information as proposed by the authors, performs better than unordered modelling and also quite comparable performance for action recognition task, than that of the contemporary state-of-the art methods discovered so far, over these datasets.

Comparison of 3D ConvNet-GRU with Res3D unordered-modelling on UCF-101 split-1 yields accuracies as: 89.3% (against 87.6%) for RGB, 88.2% (against 84.3%) for OF and 92.1% (against 91.5%) of RGB+OF.

### Literature Review: Learning Spatiotemporal Features with 3D Convolutional Networks (Du Tran *et al.*) [5]

The paper in subject addresses the challenge of learning spatiotemporal features in videos, exploiting the power of 3D ConvNets and training them over large-scale supervised datasets, following a systematic study of finding the best performing length/size of temporal kernel for 3D ConvNets.

The authors findings suggest: (1) – 3D ConvNets are better suited and effective in learning spatiotemporal information as compared to 2D ConvNets. (2) – Homogenous Architecture with a rather small 3 x 3 x 3 convolutional kernels (for all layers) is the best architecture in performance, for 3D ConvNets and lastly (3) – A simple linear classifier outperforms state-of-the-art methods on 4 different benchmarks.

This way computed and extracted; the features are compact and achieve 52.8 % accuracy on UCF-101 dataset with only 10 dimensions and owing to faster inference of ConvNets, very efficient to compute, simple and easy to train and use. Hence, C3D can model appearance and motion information simultaneously and outperforms the features extracted by the 2D ConvNet, for various video analysis tasks, using only linear classifier.

The proposed C3D’s spatiotemporal feature learning architecture has 8 #D convolution (kernel size = 3x3x3, Stride = 1 for both Spatial and Temporal dimensions), 5 max-pooling and 2 fully connected layers, followed by a softmax output layer.

Training is conducted on the Sports-1M train split, randomly extracting 5(five) of 2-second clips from every video, resized to 128x171, randomly flipped with 50% probability, utilizing SDG (Stochastic Gradient Decent) with min. size of the batch as 30 examples, initial learning rate as 0.003 divided by 2 per 150k iterations and stopping optimization every 1.9M iterations (approx. 13 epochs).

Action recognition evaluation for C3D features is conducted on UCF-101dataset, consisting of 13,320 videos of 101 categories of human-action, using 3-split.

For Object recognition evaluation is conducted on Egocentric dataset, consisting of 42 types of everyday objects.

Action similarity labelling evaluation is conducted on a dataset called ASLAN, consisting of 3,631 videos over 432 classes of action, where the task is the prediction whether the given pair of videos belongs to the same or a different action. C3D outperforms the contemporary state-of-the-art methods by 9.6% (accuracy=78.3%) and by 11.1% (area under ROC curve=86.5%).

For Dynamic Scene recognition evaluation is conducted on: (1) – YUPENN, consisting of 420 videos of 14 scene categories and (2) – Maryland, consisting of 130 videos of 13 scene categories. C3D used in combination with a simple linear SVM, clearly performs higher than the current methods on the Maryland dataset (with 87.7% accuracy) and the YUPENN dataset (with 98.1% accuracy).

## 2.2 Stakeholders list:

* Project Team
* Project Supervisor
* Security Agencies
* Data Analysts
* City Management

## 2.3 Product Functions:

Gideon consist of these following components:

* **NVIDIA Jetson Nano:** Embedded AI on the Edge Platform. The trained Anomaly Detection model will be deployed on this platform, interfaced with CCTV camera input, and will perform near real-time inference to predict anomalous/abnormal activity. The platform will also host an application server, that will communicate with its clients, for the reporting of anomalous activity inferenced.
* **Web Application:** The web application will serve as a web-based interface for the end users of Gideon, providing them with real-time monitoring and a cluster of functionalities designed for the system.
* **Android Application:** The android application will serve as an android-based interface for the end users of Gideon, providing them with real-time monitoring and a cluster of functionalities designed for the system.

Gideon provides the following functionality to the user:

* Log into the system using privileged credentials and monitor the CCTV Camera stream in real-time.
* Get alert for the anomalous activity detected, through interfaced Web and Android Interfaces of the system in real-time.
* Access and View complete archived history of Anomalous events detected by the application, in chronological order.
* Modify the contents of stored data of anomalies on the system.

### 2.3.1 User characteristics:

The types of users identified that can use this system are:

* Admin

User Characteristics

Table 2(User Characteristics)

|  |  |  |  |
| --- | --- | --- | --- |
| **User** | **Level of Computer Knowledge** | **Level of Business Knowledge** | **Frequency of Use** |
| Admin | Basic to advance knowledge | Can easily follow the system’s instructions | Constantly Using over an extended period of time |

In the above Table 2, a detailed description of the user of the system is described.

## 2.4 Requirements Elicitation:

We collected most of our basic requirements from observation and brainstorming and for our technical requirements, we read a lot of research papers and technical blogs. Furthermore, we consulted experts related to the field of Machine Learning, Data Analysts and Computer Vision. Next, we met with groups that have worked and created systems similar to ours and discussed the requirements and the improvements that can be done. Finally, we did a study on how our system can be deployed and how can organizations that are currently working in the field similar to our project can integrate our system with their own.

### 2.4.1 Functional requirements:

Functional requirements are the tasks that are expected from the system. They describe all the functions and the operations that are carried out by the system. Functional requirements can be described in different ways, sometimes the statements can be more technical or written in high-level because they are addressed to certain high-level narrow demographics whereas they can also be written in more easy and natural language so that any layperson can read them and get a good understanding of the functionality of the system.

**FR01: Login**

Table 3(FR01 - Log in)

|  |  |
| --- | --- |
| **Req. No.** | **Functional Requirements** |
| **FR01-01** | The system shall grant the admin access after verification of login credentials |
| **FR01-02** | The system shall allow the admin to change his/her username and password |
| **FR01-03** | The system shall maintain a database of all the user accounts |

In the above Table 3, the functional requirements of login are mentioned.

**FR02: Anomaly Detection**

Table 4(FR02 – Anomaly Detection)

|  |  |
| --- | --- |
| **Req. No.** | **Functional Requirements** |
| **FR02-01** | The system shall alert the user when an anomaly is detected |

In the above Table 4, the functional requirement of anomaly detection is mentioned.

**FR03: Video Database**

Table 5(FR03 – Video Database)

|  |  |
| --- | --- |
| **Req. No.** | **Functional Requirements** |
| **FR03-01** | The system shall maintain a database of anomaly clips called clip history |
| **FR03-02** | The system shall assign a unique ID to every clip in clip history |
| **FR03-03** | The system shall allow the admin to delete any video from clip history |
| **FR03-04** | The system shall allow the admin to view old clips from clip history |

In the above Table 5, the functional requirements of anomaly clips database are mentioned

### 2.4.2 Non-Functional requirements:

Non-Functional requirements set the standards for the system. They define what is expected from the system apart from its basic functionality. They provide the constraints and set restrictions on the system. Lastly, they are required as a measure of quality check during testing and cover areas such as Performance, Security, Reliability, Usability, Compatibility, Reusability, Understanding, etc.

#### NFR-01: Performance

Table 6(NFR01 - Performance)

|  |  |
| --- | --- |
| **Req. No.** | **Non-Functional Requirement** |
| **NFR01-01** | The system must alert the user no more than 10 seconds after the anomaly has occurred |
| **NFR01-02** | The start-up time of the system must not be more than 10 seconds. |

In the above Table 6, non-functional requirements of the software's performance are mentioned.

#### NFR-02: Usability

Table 7(NFR02 - Usability)

|  |  |
| --- | --- |
| **Req. No.** | **Non-Functional Requirement** |
| **NFR02-01** | The User must be able to get familiar with the interface no more than 10 minutes. |
| **NFR02-02** | The System must have minimal design with user-friendly interface |

In the above Table 7, non-functional requirements of the software’s usability are mentioned.

#### NFR-03: Reusability

Table 8(NFR 03 - Reusability)

|  |  |
| --- | --- |
| **Req. No.** | **Non-Functional Requirement** |
| **NFR03-01** | The system must allow the reuse of functions in the system in a different environment |

In the above Table 8, non- functional requirements of the software’s reusability are mentioned.

#### NFR-04: Defect Maintenance

Table 9(NFR 04 – Defect Maintenance)

|  |  |
| --- | --- |
| **Req. No.** | **Non-Functional Requirement** |
| **NFR04-01** | After implementation, the system must produce a minimum about of errors |
| **NFR04-02** | Debugging should not take more than one day |

#### In the above Table 9, non-functional requirements of the software’s defect maintenance are mentioned.

#### NFR-05: Completeness

Table 10(NFR05 - Completeness)

|  |  |
| --- | --- |
| **Req. No.** | **Non-Functional Requirement** |
| **NFR05-01** | The System must always be consistent in detecting same anomalies. |

In the above Table 10, non-functional requirements of the software’s completeness are mentioned

#### NFR-06: Security

Table 11(NFR06 - Security)

|  |  |
| --- | --- |
| **Req. No.** | **Non-Functional Requirement** |
| **NFR06-01** | The system must only allow authorized users with correct login credentials to access the system. |

In the above Table 11, non-functional requirements of the software’s security are mentioned.

#### NFR-07: Extensibility

Table 12(NFR07 - Extensibility)

|  |  |
| --- | --- |
| **Req. No.** | **Non-Functional Requirement** |
| **NFR07-01** | The system must be built in a modular form so that any extension can be easily integrated |

In the above Table 12, non-functional requirements of the software’s extensibility are mentioned

## 2.5 Use Case Description:

Table 13(Use case - 1)

|  |  |
| --- | --- |
| **Use Case ID:** | 1 |
| **Use Case Name:** | User Login |
| **Actors:** | Admin |
| **Description:** | The Admin wants to access the system |
| **Pre-Condition:** | The Admin should know login credentials |
| **Post-Condition:** | The Admin will be provided access to the system. |
| **Normal Flow of Events:** | 1. The admin opens the web/android application 2. The admin adds his/her user Id and password. 3. The admin receives access to the system |
| **Alternatives Flow:** | 1. The admin enters incorrect login information 2. Access to the system is denied |
| **Exceptions:** | None |

Table 13 shows the Use Case ID:1. In this module, the user enters the login credentials. The software compares the information with the information stored in the database and if the information is correct, the user is provided access to the system.

Table 14(Use case - 2)

|  |  |
| --- | --- |
| **Use Case ID:** | 2 |
| **Use Case Name:** | Anomaly Detection |
| **Actors:** | System, Admin |
| **Description:** | The system shall detect the anomaly, clip the video, send the video to the user and alert the user via web /android application |
| **Pre-Condition:** | There should be a camera live feed |
| **Post-Condition:** | Anomaly has been detected |
| **Normal Flow of Events:** | 1. The camera captures the live feed 2. The system preprocesses the feed 3. The system detects the anomaly 4. The system clips the relevant part 5. The system sends alerts the user and shows the video |
| **Alternatives Flow:** | 1. No anomaly detected |
| **Exceptions:** | None |

Table 14 shows the Use Case ID:2. In this module, the model is fed a live camera stream and upon happening of an anomalous event, the system generates snippet and alerts the user.

Table 15(Use case - 3)

|  |  |
| --- | --- |
| **Use Case ID:** | 3 |
| **Use Case Name:** | Watching Anomaly Clips |
| **Actors:** | Admin |
| **Description:** | Admin searches an old anomaly video from the clip history |
| **Pre-Condition:** | The availability of videos in clip history |
| **Post-Condition:** | The video is found and successfully played |
| **Normal Flow of Events:** | 1. The admin logs in to the system 2. The admin searches a video from the clip history 3. The admin clicks the video 4. The video starts playing |
| **Alternatives Flow:** | a. There are no videos in the clip history |
| **Exceptions:** | None |

Table 15 shows the Use Case ID:3. In the module, the user selects a clip from the clip database and the system starts playing the video.

Table 16(Use case - 4)

|  |  |
| --- | --- |
| **Use Case ID:** | 4 |
| **Use Case Name:** | Delete Clip |
| **Actors:** | Admin, System |
| **Description:** | Admin deletes a clip from clip history |
| **Pre-Condition:** | The availability of videos in clip history |
| **Post-Condition:** | The video is found and successfully deleted |
| **Normal Flow of Events:** | 1. The admin logs in to the system 2. The admin searches a video from the clip history 3. The admin deletes the video 4. The system removes the clip from its memory |
| **Alternatives Flow:** | a. There are no videos in the clip history |
| **Exceptions:** | None |

Table 16 shows the Use Case ID:3. In the module, the user selects a clip from database and the system deletes the clip from the database.

## 2.6 Use Case Diagrams:

### 2.6.1 Use Case ID: 1

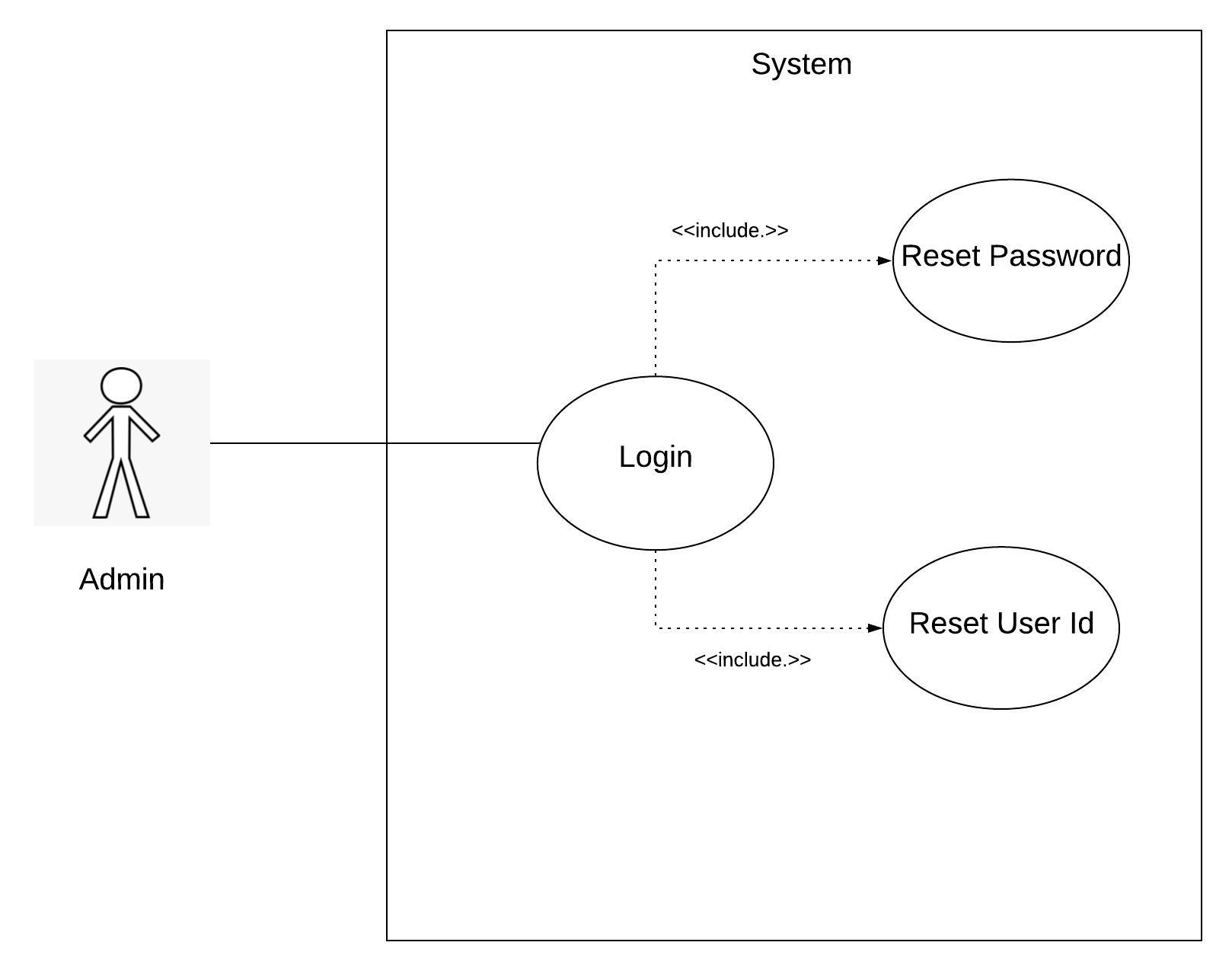
****

Figure 3(Use case - 1)

The Figure 3 above presents the description for Use Case 1, which is User Login. The actor in this use case is Admin who provides the right credentials to login into the system or the access is denied.

### 2.6.2 Use Case ID: 2

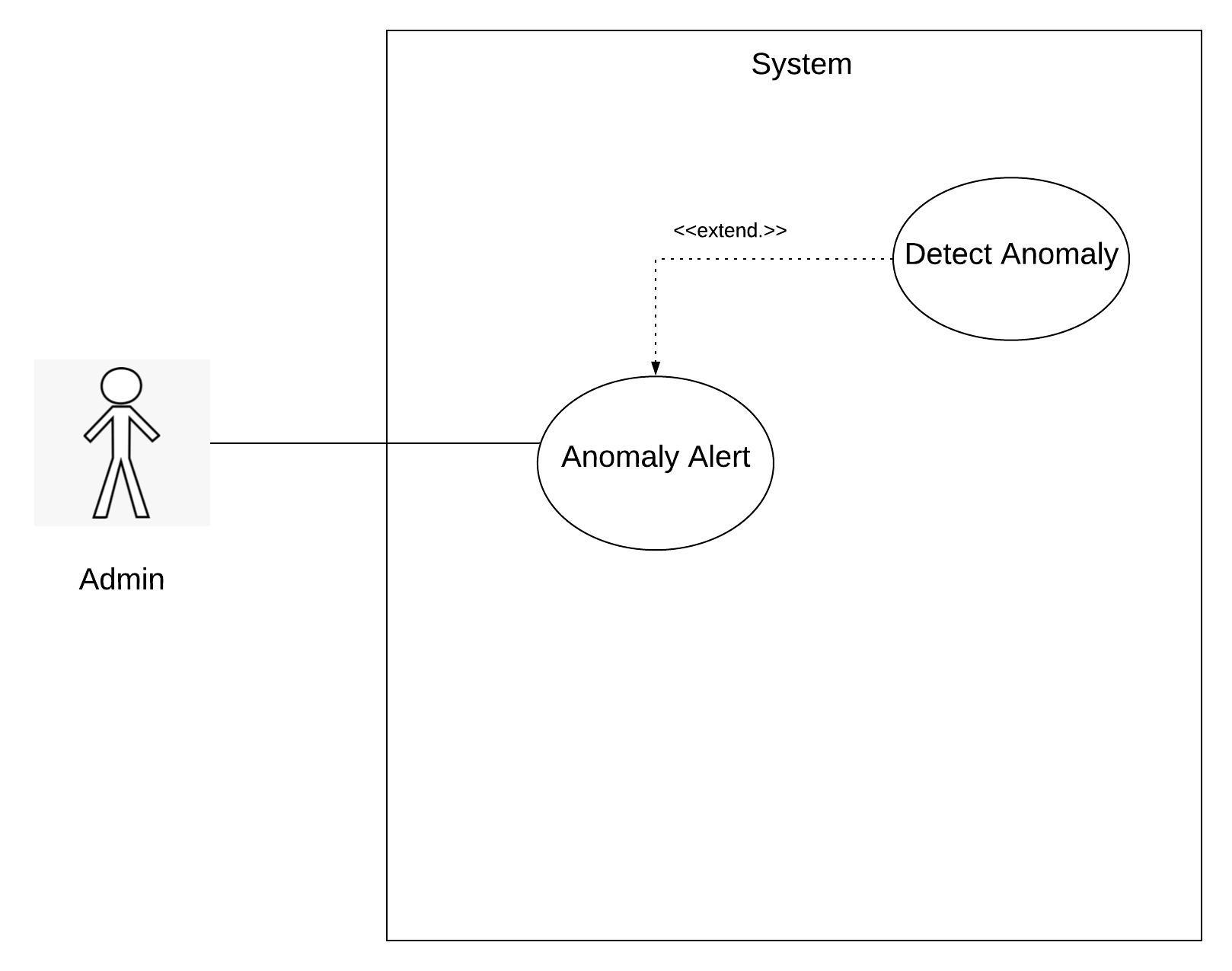
****

Figure 4(Use case - 2)

The Figure 4 above presents description for Use Case 2 which is Anomaly Detection. The Inference Engine inside the Gideon's IVA (Intelligent Video Analytics) application will detect the anomaly, extract the video segment and log it, while at the same time, will inform the system user through its dedicated Mobile and Web interfaces.

### 2.6.3 Use Case ID: 3

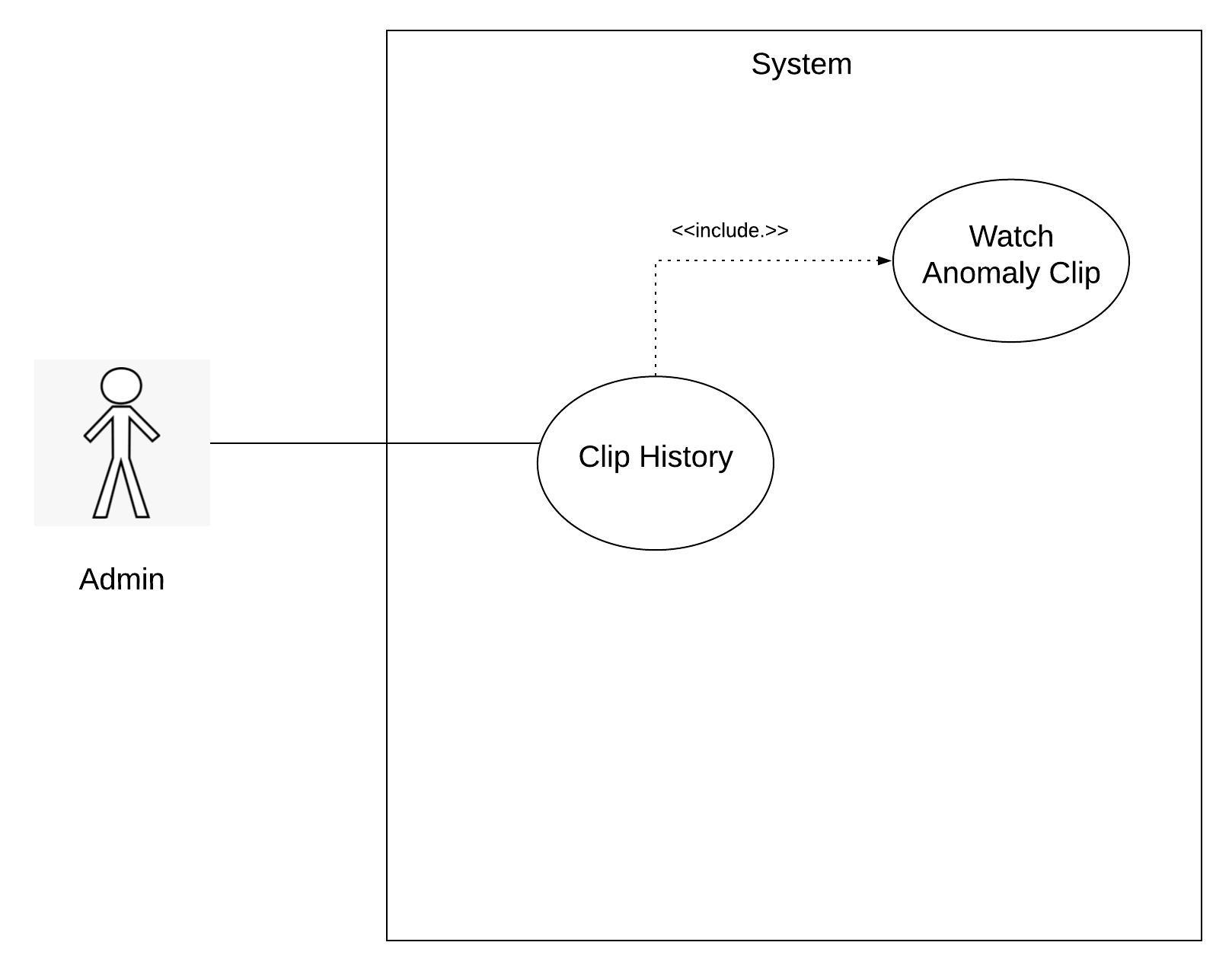
****

Figure 5(Use case - 3)

The Figure 5 above presents the description of Use Case 3 which is for viewing the extracted Anomalous Event video clips.

### 2.6.4 Use Case ID: 4

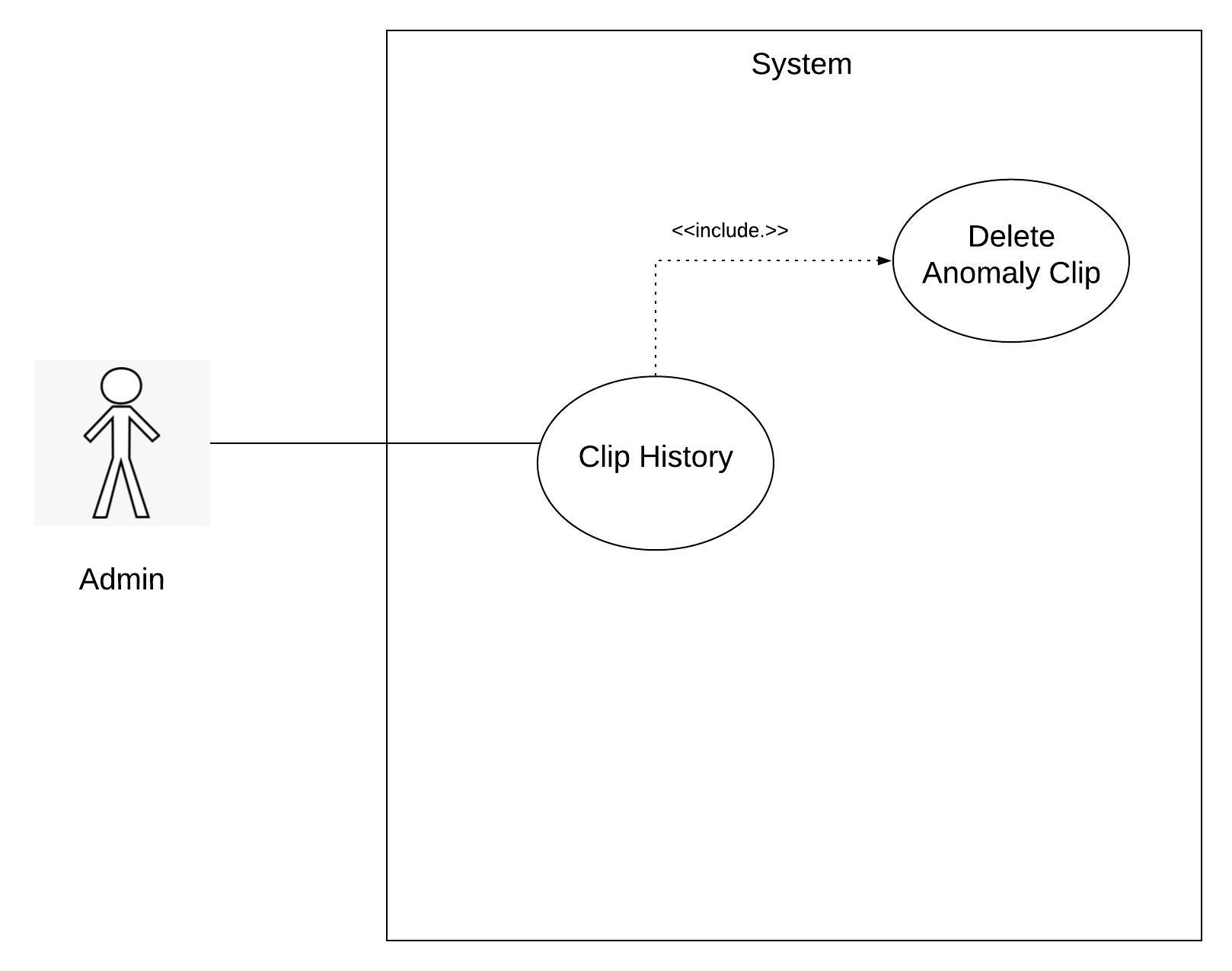
****

Figure 6(Use case - 4)

The Figure 6 above presents the description for Use Case 4 which is deleting the extracted Anomalous Event video clips, which the user does not want to retain any further in the system storage.

Chapter 3: System Design

# System Design:

## Work Breakdown Structure: (WBS)

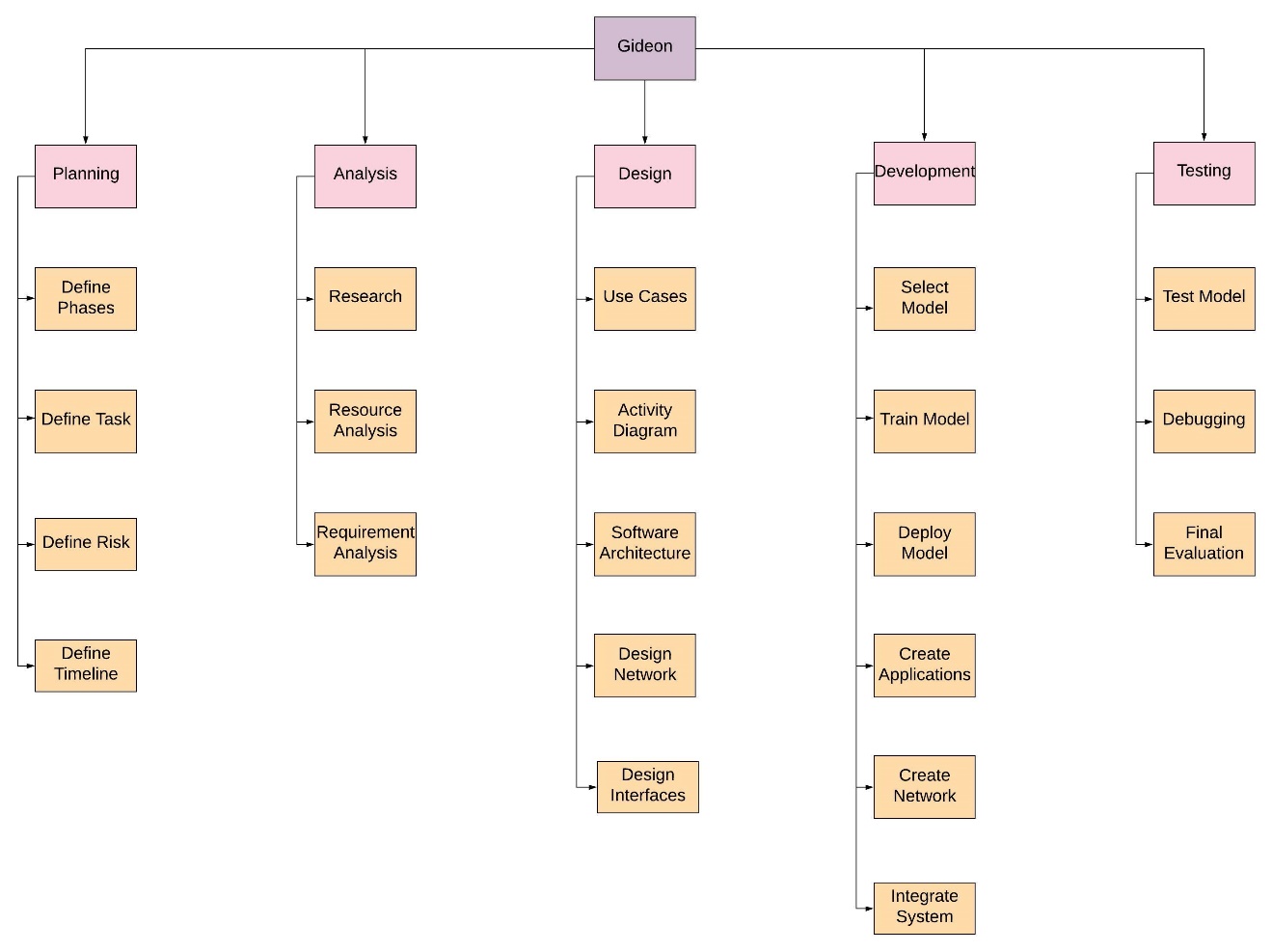


Figure 7(Work Breakdown Structure)

The Figure 7 above presents the entire Work Breakdown Structure (WBS) of the Gideon IVA (Intelligent Video Analytics) application, including all its phases for Planning, Analysis, Designing, Development and finally, the System Testing.

## 3.2 Activity Diagram:

### 3.2.1 Login:

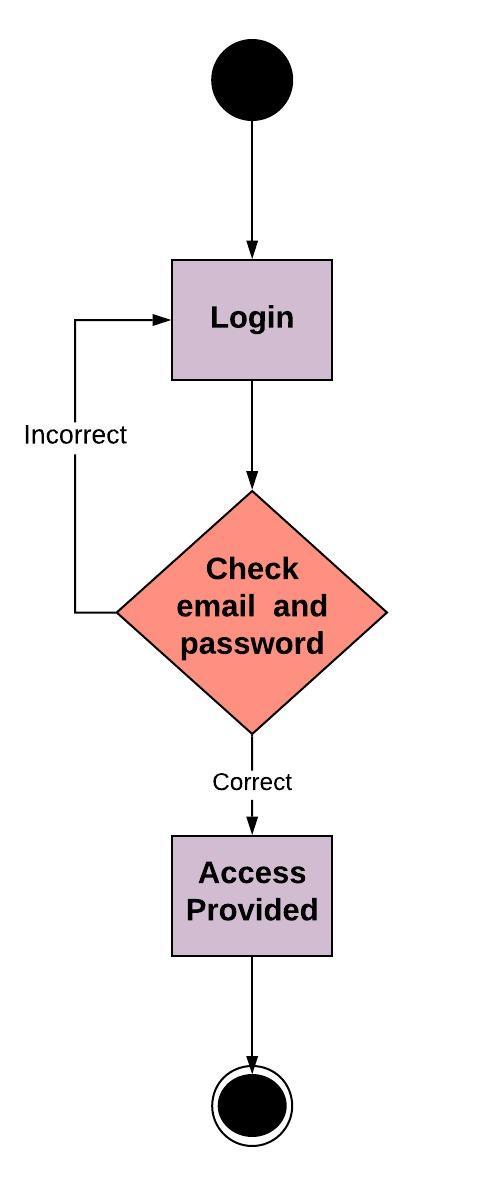


Figure 8(Activity - 1)

The Figure 8 above presents the activity sequence for system login.

### 3.2.2 Anomaly Detection:

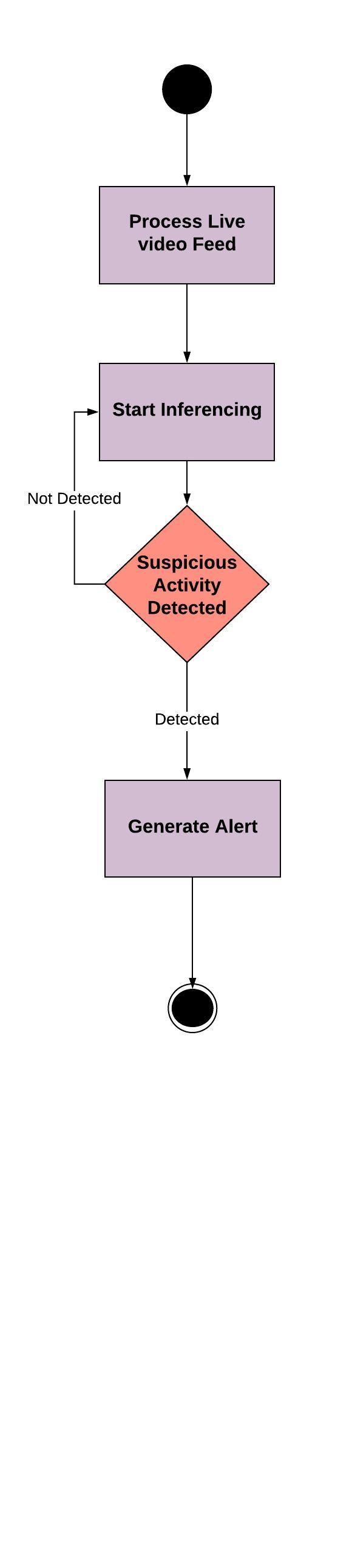


Figure 9(Activity - 2)

The Figure 9 above presents the activity sequence for Anomaly Detection and the subsequent Alert generation.

## Sequence Diagram:

### Login:

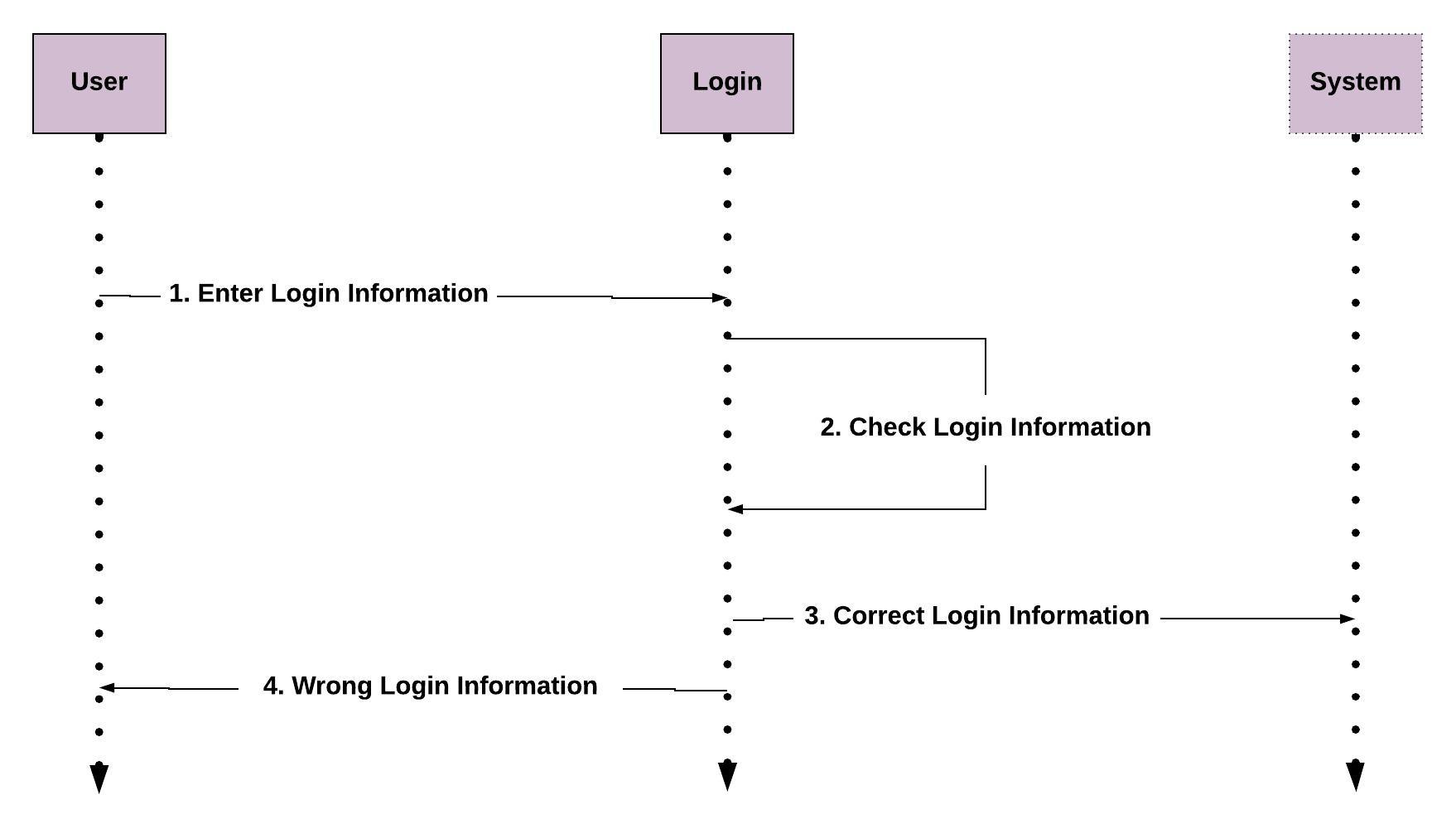


Figure 10(Sequence- 1)

The Figure 10 above describes the sequence for User Login into the Gideon IVA.

### Anomaly Detection

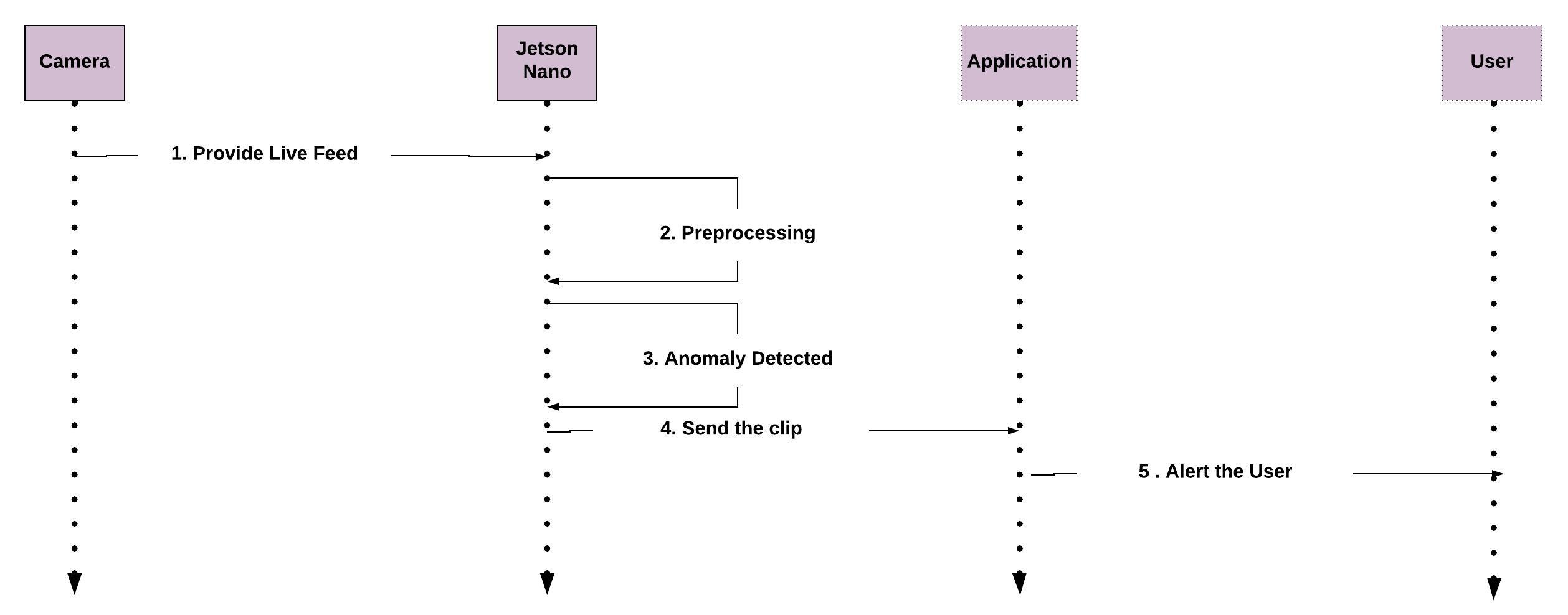
****

Figure 11(Sequence - 2)

Figure 11 above describes the sequence for the Anomaly Detection.

## Software Architecture:

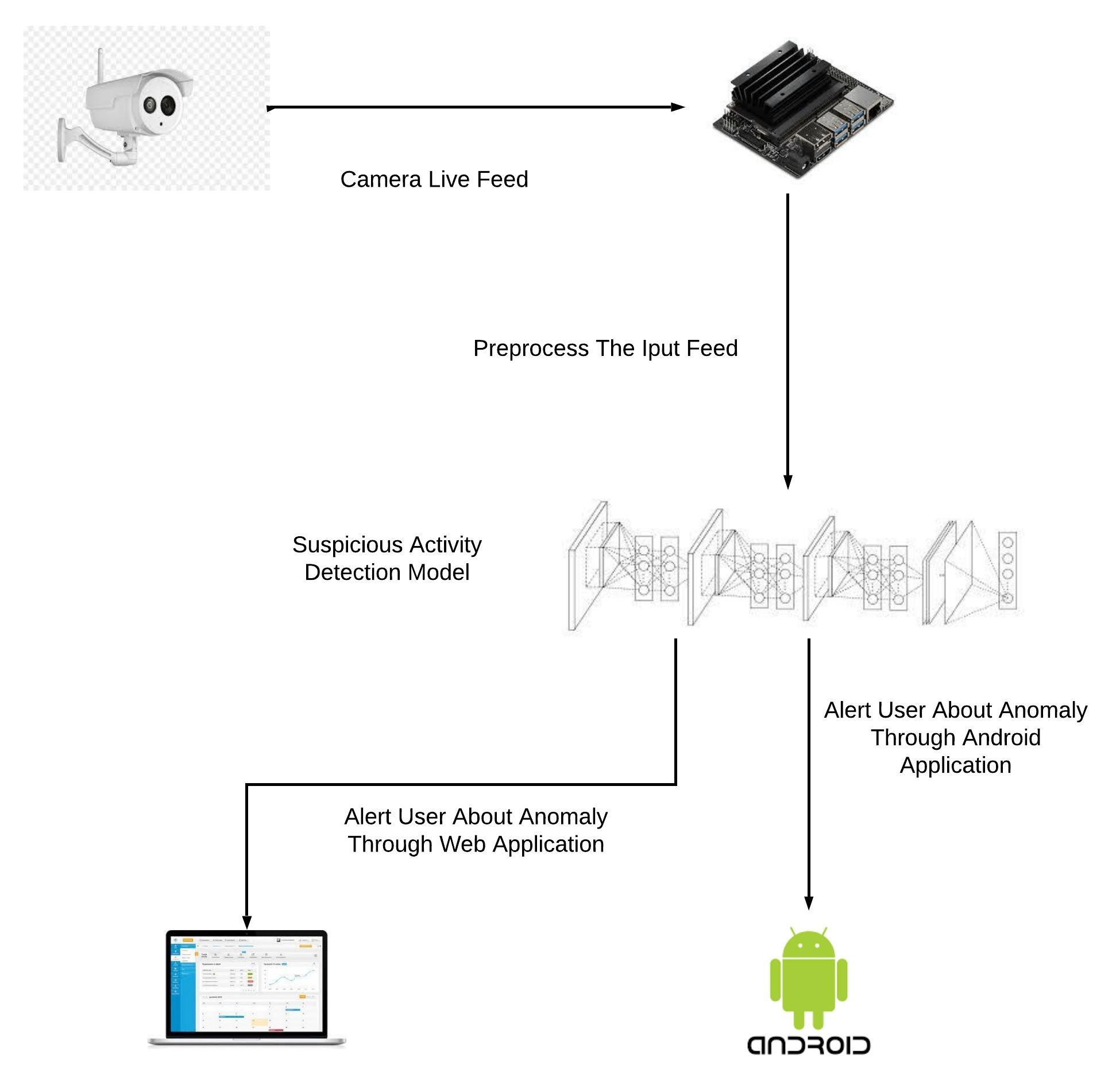


Figure 12(Architecture)

The Figure 12 above presents the System Architecture for the Gideon IVA (Intelligent Video Analytics) Application, that takes the live CCTV camera feed and exploits the AI on the Edge capability of NVIDIA's Jetson Nano Embedded Platform, to perform Anomaly Detection inference.

## Collaboration Diagram:

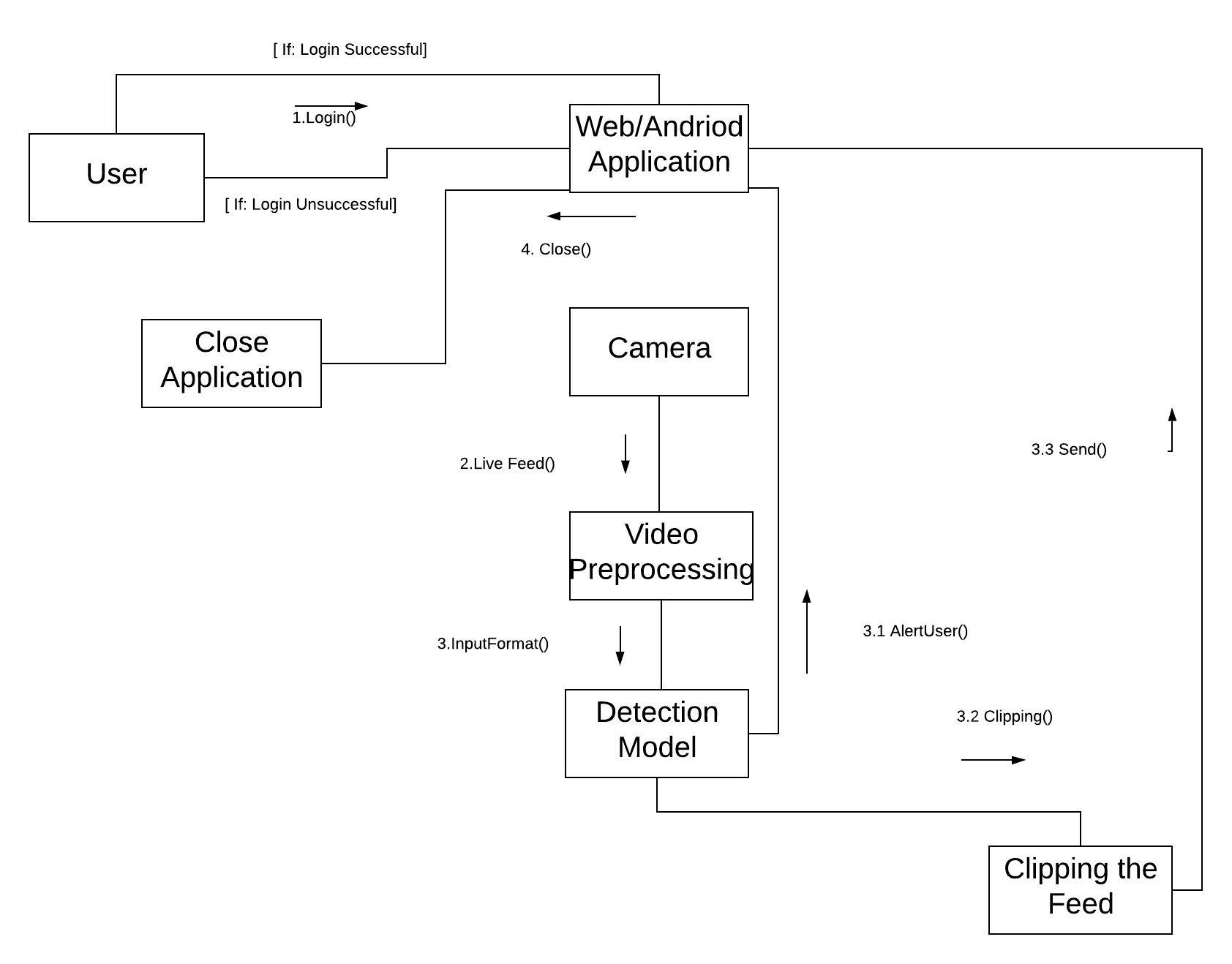


Figure 13(Collaboration)

The Figure 13 above presents the Collaboration sequence of the entire system from the Front-End interfaces along with integration with the IVA (Intelligent Video Analytics) application core running on NVIDA Jetson Nano.

## Project Code

(project code files will be placed here)

Chapter 4: System Testing

# System Testing

## Test Cases

Table 17(Test Case - 1)

|  |  |
| --- | --- |
| **Test case ID: TC-01** | |
| Application Name | Gideon:IVA |
| Use case(s) | Login |
| Input summary | User will be asked to provide their login details |
| Pre-condition | The application should be turned on |
| Post-condition | User will be navigated to the home screen |
| Output Summary  *IF SUCCESS:*  Use is provided access  *ELSE:*  Error message will be displayed about wrong credentials and user will be asked to enter again | |

In the above Table 17, a test case regarding user login is described

.

Table 18(Test Case - 2)

|  |  |
| --- | --- |
| **Test case ID: TC-02** | |
| Application Name | Gideon:IVA |
| Use case(s) | Anomaly Detection |
| Input summary | Live camera feed |
| Pre-condition | The camera is monitoring some activity |
| Post-condition | User will be alerted about the anomaly |
| Output Summary  *IF SUCCESS:*  Use is provided an anomaly snippet | |

In the above Table 18, a test case regarding the anomaly detection is described.

Table 19(Test Case - 3)

|  |  |
| --- | --- |
| **Test case ID: TC-03** | |
| Application Name | Gideon:IVA |
| Use case(s) | Change User Credentials |
| Input summary | User needs to open personal settings |
| Pre-condition | User must be logged in |
| Post-condition | New credentials will be saved |
| Output Summary  *IF SUCCESS:*  New Credentials will be saved in the database | |

In the above Table 19, a test case regarding changing username or password is described.

Table 20(Test Case - 4)

|  |  |
| --- | --- |
| **Test case ID: TC-04** | |
| Application Name | Gideon:IVA |
| Use case(s) | Attach Camera |
| Input summary | Camera should be attached to the Jetson Nano’s socket |
| Pre-condition | User must be logged in |
| Post-condition | Camera will be added |
| Output Summary  *IF SUCCESS:*  New camera will be added | |

In the above Table 20, a test case regarding attaching camera to our application is described.

Table 21(Test Case - 5)

|  |  |
| --- | --- |
| **Test case ID: TC-05** | |
| Application Name | Gideon:IVA |
| Use case(s) | Remove Camera |
| Input summary | User needs to open camera settings and add socket number |
| Pre-condition | User must be logged in |
| Post-condition | Camera will be removed |
| Output Summary  *IF SUCCESS:*  Camera will be removed | |

In the above Table 21, a test case regarding removing camera from the application is described.

Table 22(Test case - 6)

|  |  |
| --- | --- |
| **Test case ID: TC-06** | |
| Application Name | Gideon:IVA |
| Use case(s) | Delete Clip |
| Input summary | User should open clip history and select a clip |
| Pre-condition | User must be logged in |
| Post-condition | Clip will be deleted |
| Output Summary  *IF SUCCESS:*  Clip will be removed from the database | |

In the above Table 22, a test case regarding deletion of clip from the database is described.

Table 23(Test Case - 7)

|  |  |
| --- | --- |
| **Test case ID: TC-07** | |
| Application Name | Gideon:IVA |
| Use case(s) | Create Account |
| Input summary | User should open create account screen |
| Pre-condition |  |
| Post-condition | New profile will be created |
| Output Summary  *IF SUCCESS:*  New profile will be created and saved in the database | |

In the above Table 23, a test case regarding new user profile is described.

## Unit Testing

In this phase, we tested our project’s individual components that perform unit tasks and are part of the whole workflow of the project.

The limited bugs that were identified and fixed were:

## Acceptance Testing

In this phase, we checked our project to see whether it fulfils our business requirements.

In our case, we check the following functionalities of the software:

* Login
* Real Time Anomaly Detection
* Extracting Snippet
* Clip Database

Our software fulfilled the above listed requirements and is ready to be deployed.

Chapter 5: Application Front-End

# Application Frontend

This section contains screenshots of the various screens of Gideon’s Android and Web application Interface.

## Android Interface

### Login Screen

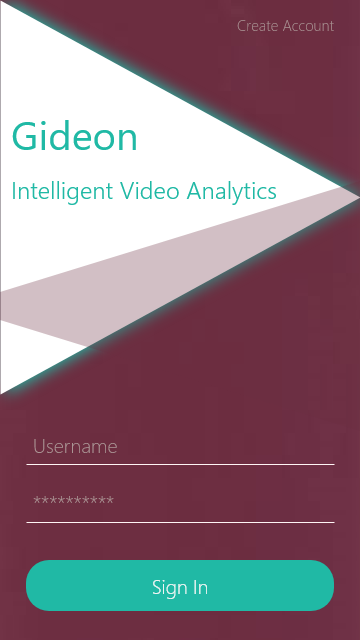


Figure 14(User Interface - Android - 1)

### Home Screen

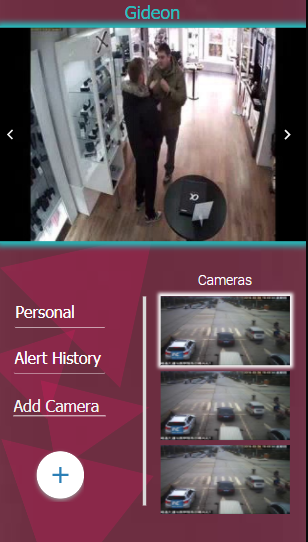


Figure 15(User Interface - Android - 2)

### Alert

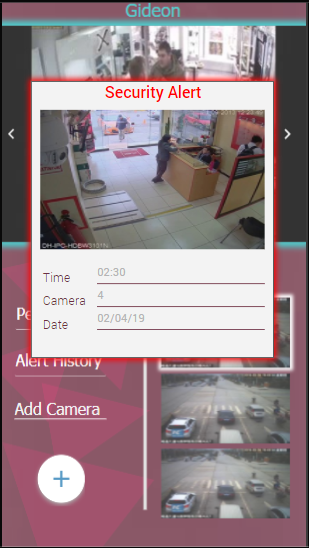


Figure 16(User Infer case - Android - 3)

### Personal Setting

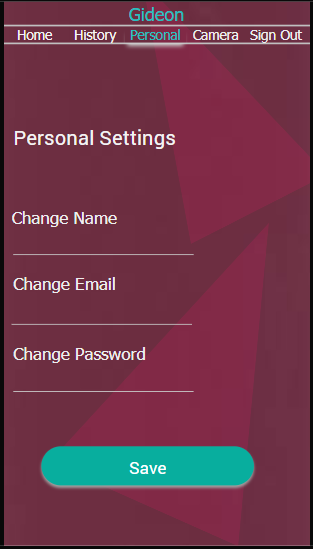


Figure 17(User Interface - Android - 4)

### Camera Settings

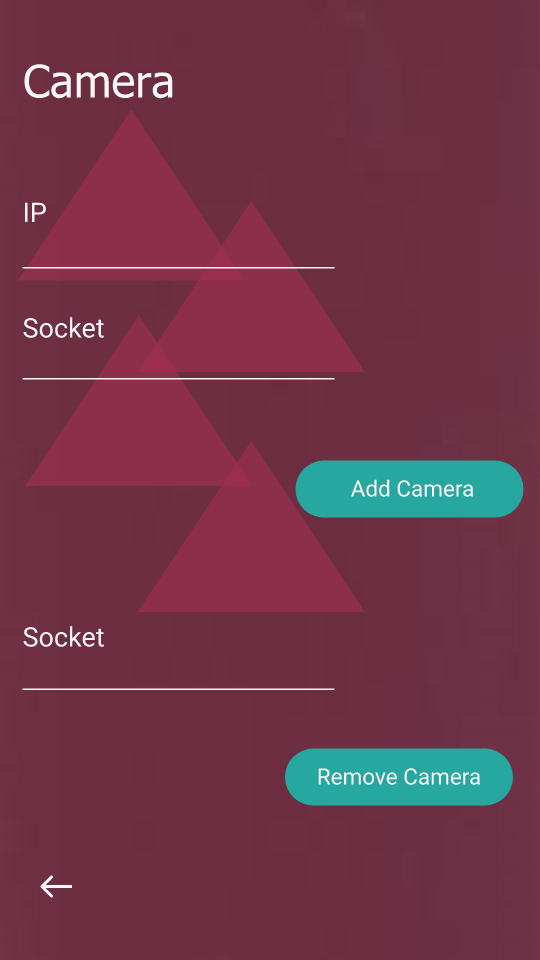


Figure 18(User Interface - Android -5)

### Clip History

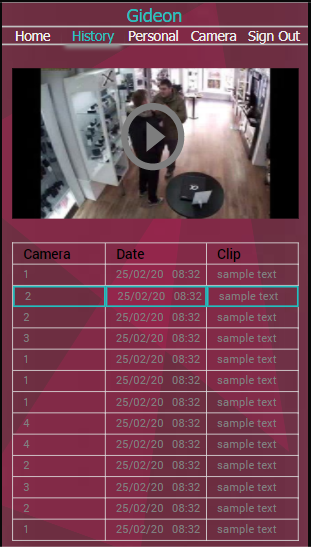


Figure 19(User Interface - Android - 6)

### Sign Up

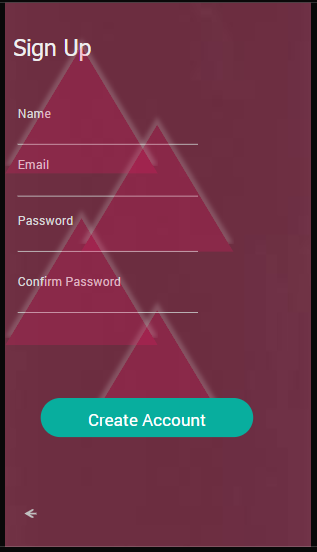


Figure 20(User Interface - Android - 7)

## Web Interface

### Login

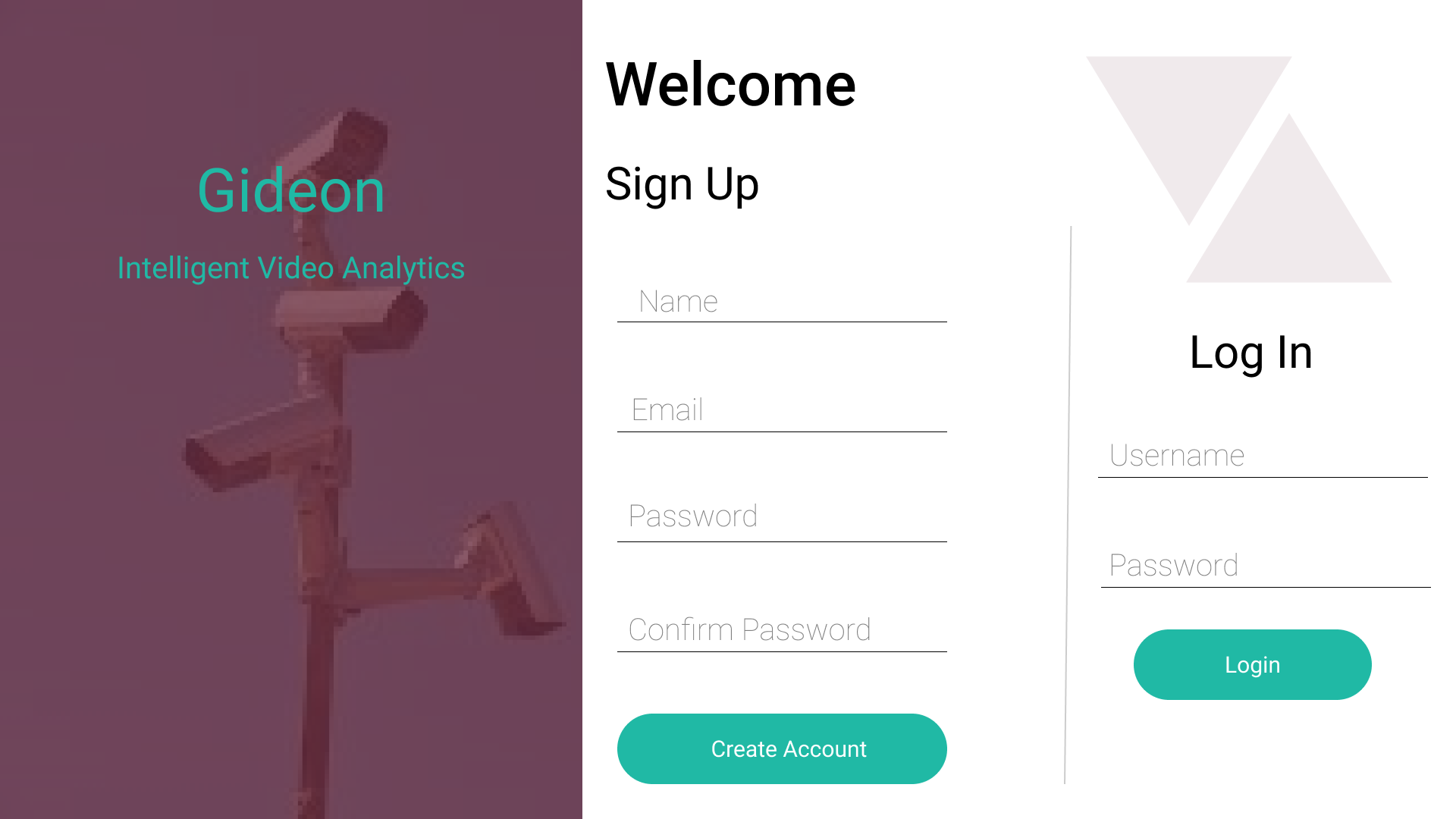


Figure 21(User Interface - Web - 1)

### Home

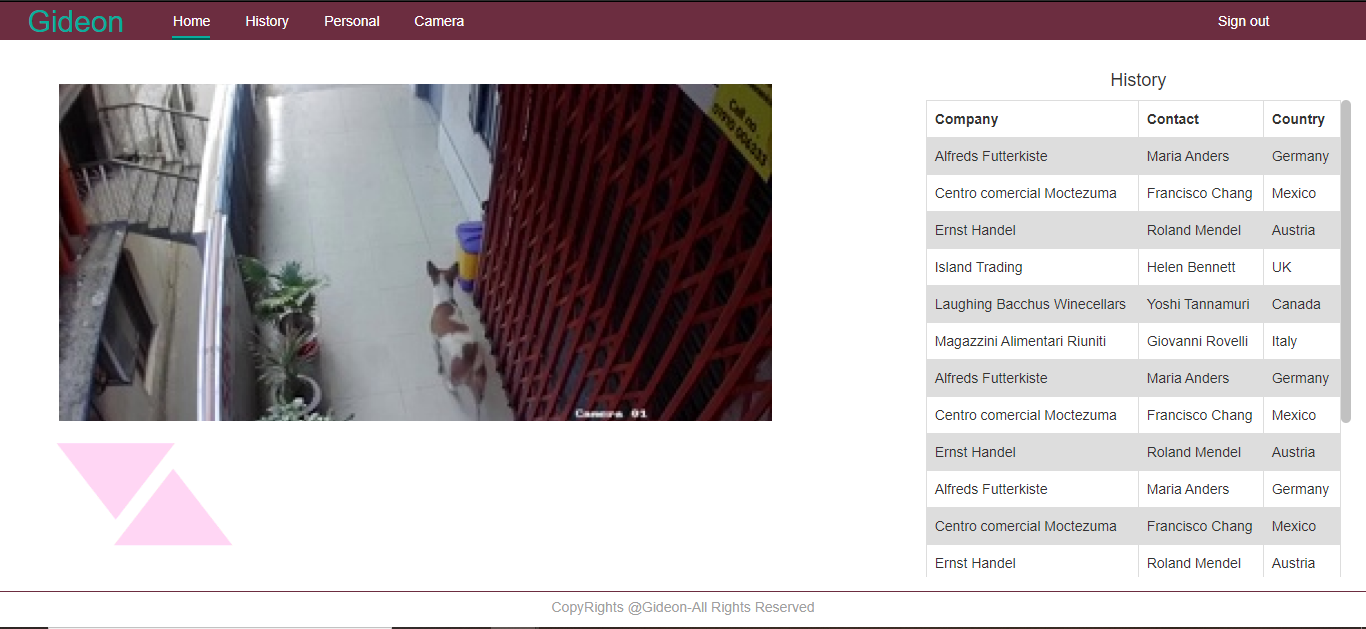


Figure 22(User Interface - Web - 2)

### Clip History

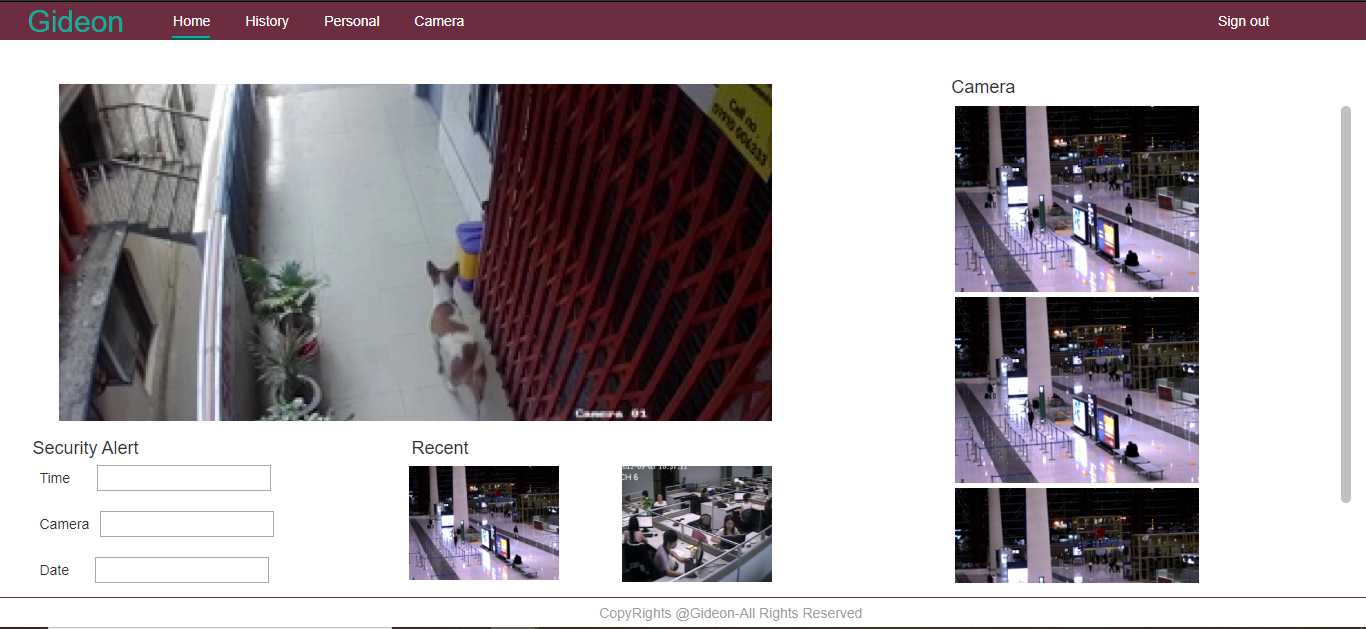


Figure 23(User Interface - Web - 3)

### Personal Settings

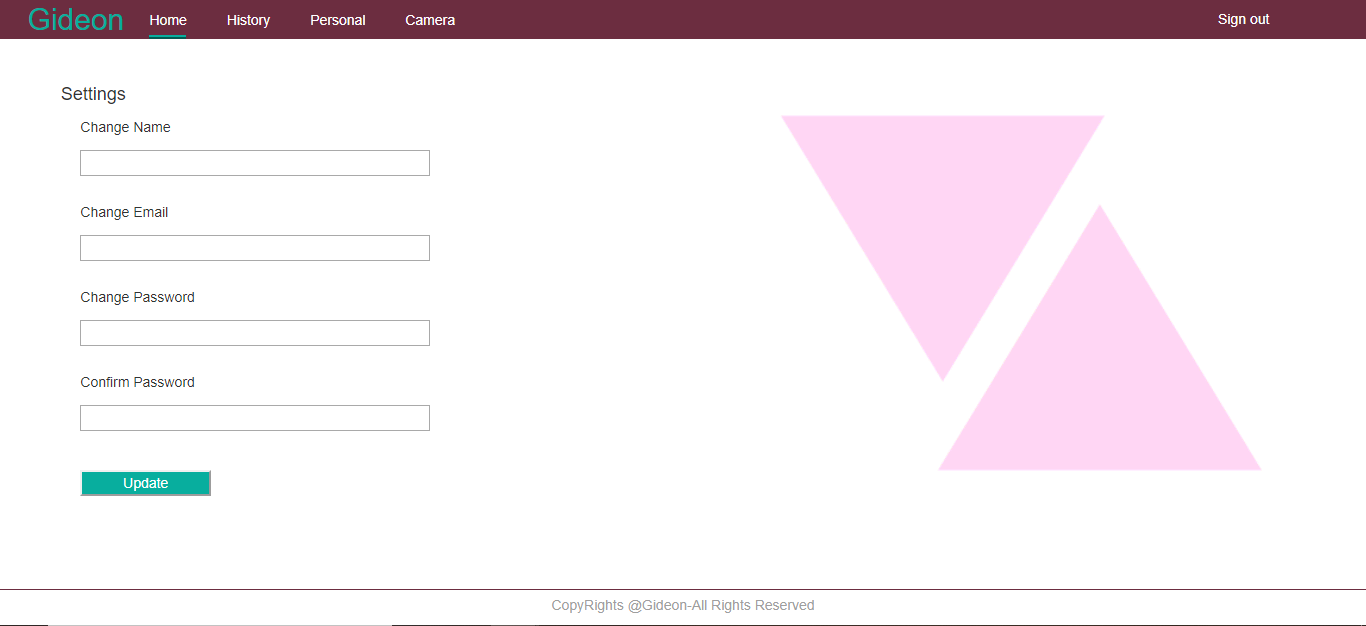


Figure 24(User Interface - Web - 4)

Chapter 6: Conclusion

# Conclusion

## Problems Faced and Lessons Learned

The first problem that we encountered in the beginning was the hardware required for the training of Neural Network. As we were dealing with complex neural networks that required huge computational power to do the learning, the facility available to us was not fulfilling our requirement. From using Google Colab to exploring Vast.ai (market platform for rental GPU service) we finally discovered Google Cloud that met our requirement and we were able to move ahead with our project.

The next problem we faced was deciding on which neural network implementation to use. We did a lot of research and tried different model architectures. Upon training we got different results from different models. Although most of them had good training accuracy but on testing, some had a large testing loss value which was never coming down while some were giving biased results as they were labelling all the videos as normal. After a lot of hit and trials, C3D(Facebook) Keras implementation was showing the best results. Although we experimented with this implementation during our initial research but due to our limited acquaintance with Neural Network training we were not able to produce the same results.

After choosing C3D ,the next challenge we faced was its deployment on NVIDIA’s Jetson Nano. As we had aimed at creating an AI on the edge portable solution, we needed our model to be small enough to be deployed on the Jetson Nano without exceeding its memory footprint. With the help of TensorRT, an optimizing tool, we were able to reduce the parameters to 16 million and were able to successfully deploy the model.

Finally, the last hurdle in our way was to create a user friendly interface so that any lay person can get familiar with our system in a short period of time. We applied a lot of user experience(UX) techniques and with iterative designing of our interactive prototype model we were able to achieve our desired Interface.

Irrefutably, we faced a lot of challenges during this project which not only increased our learning but also polished our stress management, problem solving and team working skills.

## Project Summary

This application aims to monitor by analysing the live video stream of the CCTV security system and to generate rapid alarm against any suspicious/abnormal activity. The purpose of this project was to provide a cost feasible, computationally capable solution with AI on the Edge, by providing embedded hardware solution powered by Deep Learning and Computer Vision, that can be easily interfaced with current CCTV camera(s) in the market and provide the capability of Intelligent Video Analytics (IVA) for detecting abnormal or suspicious activities.

## Future Work

Points below are some of the extensions that can be added to our project in future:

* In the future, multi-camera person tracking of the person involved in the suspicious activity can be added, thus helping authorities to catch the criminals more quickly.
* The application can be extended from a binary classification to a multi-class crime classification which would help the authorities to better respond with the necessary protocols and also it will help them to better document crime.
* Further extension is increasing the spectrum of crime detection. Currently our application can recognise anomalies based on the thirteen crimes of the UCF Crime dataset. The model can be further trained to detect more than 13 crimes.

Chapter 7: Reference

# References

[1] Waqas Sultani, Chen Chen, Mubarak Shah (2018). Real-world Anomaly Detection in Surveillance Videos.

[2] Federico Landi, Cees G. M. Snoek, Rita Cucchiara. (2019). Anomaly Locality in Video Surveillance.

[3] Kun Liu, Wu Liu, Chuang Gan, Mingkui Tan, Huadong Ma (2018). T-C3D: Temporal Convolutional 3D Network for Real-Time Action Recognition. *The Thirty-Second AAAI Conference on Artificial Intelligence*.

[4] Guangle Yao, Xianyuan Liu, Tao Lei. (2018). Action Recognition with 3D ConvNet-GRU Architecture.

[5] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, Manohar Paluri. (2015). Learning Spatiotemporal Features with 3D Convolutional Networks.

.