CE903/CE913

Team 2

Software Requirements Specification

Document

Version: (1) **Date: (07/02/2019)**

Table of Contents

3.7.1 System Modes
3.7.2 User Classes
3.7.3 Interaction Diagrams
3.7.4 Feature
3.7.5 Stimulus
3.7.6 Response
3.7.7 Data Flow

3.8 Additional Comments

4. Project Management

5. Change Management Process

6. Document Approvals

7. Supporting Information

1. Introduction

This document is meant to cover the user functional and nonfunctional requirements that define what exactly the software is supposed to do for a product that is meant to classify video from a drone based upon the behaviour of the subject. The first section describes to problem domain and the scope of the project. The second section details an overview of the system and how it fits into current research into active vision (moving the camera automatically to the best position to get data), drawing attention to subjects, and classifying the behaviour of subjects. The third section goes into the functional and non-functional requirements as well as mapping some of the requirements to object interaction through sequence diagrams. The final sections are on the change management process to limit the need for changes.

1.1 Purpose

The purpose of the system is to move a drone with a camera into the best position to gather image data on the subject, and secondly can classify the human behaviour between one or more persons over sixteen classes of actions in ten different environments. In total half of these sixteen classes must be with two or more people. The system expects from the beginning of it's task that at least one person will be in view, otherwise it may fail. The dataset will use a combination of datasets of people walking in various directions and a created dataset from a vicon setup with at least six points of view.

1.2 Scope

1. Identifying at least sixteen actions classes and collecting the data as training set
   Base on the dataset that have already existed analysis which behaviors we can select as classifications, write pros and cons of each. reporting the reason why they would show the good sides of the system, how complex would be to discriminate between them.
2. Building a deep learning method (RNN) that can recognize human behaviors.
   As the train data of human behavior, which are collected as many positions of limb, are time-continuous data, recurrent neural network would be a good choice for this method. In this part, the accuracy of behavior recognition and efficiency are two important attributes. The method could be started training with one behavior like falling without environment context and does not involve more than one person.
3. Implementing a detector (Vicon or camera) that can capture body behavior from video stream
   Vicon are the premier solution with a highly accurate system that provides low latency data that is easy to use. With Vicon's turnkey approach and easy to access our

DataStream (via TCP, UDP, etc.) we can easily integrate accurate Vicon data into virtually any control system.

If we use mobile camera, we have to apply some technology like Generative Quey Network, DensePose-COCO in 3D and Visualize DensePose-RCNN to capture the human body from the environment and recognize the limb position in video stream for fitting with the interface of our DL system.

4. Conducting with a data structure that we can merge training data with generating data into same data structure.

   In the real environment, we can not use the Vicon to collect the data, so our system must be able to obtain the data of human behavior through video stream. Following we extract the data of behaviors, we have to transfer the data structure to the structure of training dataset. In this way, our human behavior recognition system could adapt all kinds of inputs. Furthermore, the final dataset creation system should be reusable by other groups and users in the future. Usability specifications as well as data management ones are, then, crucial. This also implies that the new dataset created and the tools developed for its creation should be easily extendable over the course of the project and by future users.

1.3 Definitions, Acronyms, and Abbreviations.

SRS: Software Requirements Specifications
RNN: Recurrent Neural Network
CNN: Convolution Neural Network
ML: Machine Learning
DL: Deep Learning

1.4 References

1. D. Jayaraman and K. Grauman, "End-to-end policy learning for active visual categorization," IEEE Transactions on Pattern Analysis and Machine Intelligence, 2018.
2. Y. Kong and Y. Fu, "Human Action Recognition and Prediction: A Survey," arXiv preprint arXiv:1806.11230, 2018.
3. Y. Zhu, R. Mottaghi, E. Kolve, J. J. Lim, A. Gupta, L. Fei- Fei, and A. Farhadi. Target-driven visual navigation in indoor scenes using deep reinforcement learning. In ICRA, 2017
4. Johns, E., Leutenegger, S., Davison, A.J.: Pairwise decomposition of image sequences for active multi-view recognition. In: Computer Vision and Pattern Recognition, 2016 IEEE Conference on (2016)
5. C. Trivedi, "Using Deep Q-Learning in FIFA 18 to perfect the art of free-kicks", *Towards Data Science*, 2018. [Online]. Available:

https://towardsdatascience.com/using-deep-q-learning-in-fifa-18-to-perfect-the-art-of-free-kicks-f2e4e979ee66. [Accessed: 29- Jan- 2019].
6. "New drone laws come into effect today as public demands regulation increase to ensure safer flying | UK Civil Aviation Authority", *Caa.co.uk*, 2018. [Online]. Available: https://www.caa.co.uk/News/New-drone-laws-come-into-effect-today-as-public-demands-regulation-increase-to-ensure-safer-flying/. [Accessed: 01- Feb- 2019].
7. P. Zhdanov, A. Khan, A. R. Rivera and A. M. Khattak, "Improving Human Action Recognition through Hierarchical Neural Network Classifiers," *2018 International Joint Conference on Neural Networks (IJCNN)*, Rio de Janeiro, 2018, pp. 1-7. doi: 10.1109/IJCNN.2018.8489663
8. V. Khong and T. Tran, "Improving Human Action Recognition with Two-Stream 3D Convolutional Neural Network," *2018 1st International Conference on Multimedia Analysis and Pattern Recognition (MAPR)*, Ho Chi Minh City, 2018, pp. 1-6. doi: 10.1109/MAPR.2018.8337518

1.5 Overview

The first section of the report has already covered a general outline and the scope of the project. The second section of this report details for stakeholders in the project the system's functional and non functional (e.g performance) requirements and elements such as hardware, software and network interfaces used to connect to the drone. It also compares the propose product to other products on the market, as well as any constraints on the product for memory requirements as an example. The third section elaborates in much further detail the functional requirements specified by the user and goes into design constraints and performance requirements for use by developers.

2. Overall description

This section will give an overview of the whole system. The basic functions of the system and how it interacts with other relevant systems will be examined further. It will also describe the types of participant that the system uses and show what function is available for each type of them. At last, this section will list some restrictions and assumptions for the system.

2.1 Product perspective

This system will consist of three parts: movement collecting system, data processing system and action identify application. The movement collecting system will collect the action information from the actor in the real world. In the meantime both these data and training data will be sent to the data processing system to be unified. Then these pretreated data will be set as the input of the action identification application to predict which kind of action it is.

The movement collecting system will be able to communicate with the camera system to get the movement data in the 3D space. It will also be able to record the movement and provide data for the data processing system to use.

The behavior information which is provided by the collecting system will be transferred to the data processing system. The video will be recorded in avi format that is the same format as that recorded on the drone. The data processing system will have two types of output: one is training data; the other is testing data. It shall provide a user interface for selecting training data to train from and training the solution as well as options for loading test data and testing over that test data.

As the two major components of the application are some form of “active vision” [1] and classification of the behaviour of the subject(s). For the purpose of evaluating where the research project lies in terms of other research, related research is specified here. The most comprehensive study into machine learning and “active vision” is that of [1] D. Jayaraman and K. Grauman who go into some detail of their end to end machine learning solution for active vision in contrast to “passive vision” where the system doesn’t automatically orientate itself around a subject into the best position to acquire data for testing or training purposes. It also describes similar approaches to their research such as q learning, which is used in a study on artificial intelligence in the game Fifa18 with RNNs and Q learning [5]. The research by D. Jayaraman and K. Grauman also suggests simpler models of active vision by moving the camera around the subject to preset directions such as front, left, behind and right [1], which may be sufficient for getting the best image data to be able to classify a subject’s behaviour.

Extensive research has also been made concerning action recognition. [7] proposes a hierarchical classifier, in which actions are grouped in larger classes. First, a classifier learns to predict the larger class and, for each of these, a second classifier predicts the specific action. On the other hand, the approach from [8] was to use a two-stream 3D Convolutional Neural Network with both, the RGB stream and the optical flow.

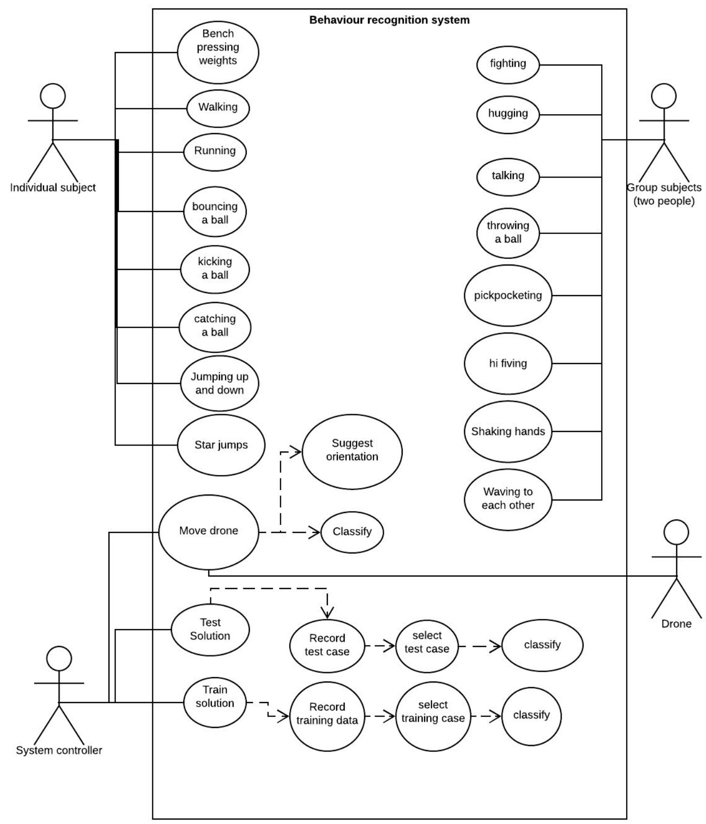
2.2 Product functions

With the identify application, the user will be able to see the illustrations of the movement and the result of the identification. The result of the application is based on the input of data. There are lots of different kind of movement and users should be able to manage which kinds of data they want to save. The saved file will show a sequence of movements provided by the application and the user can check them. The application will also provide the user a list of saved files to manage these data by checking, deleting or renaming them.

Meanwhile the movement collecting system and data processing system will collect real time data and transfer it to input data stream for this application. Ideally the user can see the real time result of the identification, but asynchronous results is also acceptable. In that case user could record the whole actions first, then send it to the application.

2.3 User characteristics

There are two types of user who need to interact with this system: the action actor and the system operator. Each of these users participate in different part of the system so they have their own requirements.



All the movements must be collected from the actor, that means that the movement collecting system must rely on some hardware system and this may need some adapting before use. The collecting system should be adaptable and should give some pointers on how to use the system. In addition, this collecting system should also have enough devices such as tracking cameras and body markers. Hence the actor should be able to wear this equipment easily and have the flexibility to move.

The training process should be performed first with the system and the system should give some user friendly way (GUI) of recording and selecting the training data and then confirm the users wish to train over that data. The test data should be gathered in real time from the camera using the movement collection system and like the training information should be recordable, selectable and testable from the user interface.

2.4 Constraints

The system is a hardware-dependent system, so all the required devices must be prepared before the system can work. That equipment may not easily to move around, so the applications should be applied in a specific space like an action collecting room.

The whole system does not need internet connection, that also means motion information can only be transferred offline, so all these components must be installed in the same computer.

2.5 Assumptions and dependencies

One assumption is this system is designed as a real time motion identification tool, so it should provide a real time feedback to the user. However, because of the uncertainty of the transport protocols, it might be difficult to pass the real time data stream between these components.

Another assumption is that all the hardware has been specifically adjusted before using. If the user does not have these devices then the system will not work. If the user has these devices but has set some unsuitable parameters, that may cause the system to get a bad performance.

2.6 Apportioning of requirement

In the case that the project is delayed, some requirements which are too complicated for this stage could be transferred to the next version of this application. Those situations will be discussed in the following section.

3. Specific Requirements

1. The solution when running in drone mode must be able to be piloted by a user to collect the "best" orientation around a subject to collect visual data from the drone's on board camera.
2. When the application is running on the drone it should suggest where to move to get the best data.
3. The solution must be able to classify sixteen kinds of behaviour from the "best" orientation on the drone. This includes:
    a. Group activities - two or more people
        i. two people hugging
        ii. two people fighting
        iii. two people talking - crossing arms
        iv. two people throwing a ball to each other
        v. one person being pickpocketed by another person from behind
        vi. two people hi fiving
        vii. two people shaking hands
        viii. two people waving at each other
    b. Individual activities: Exercising
        i. Bench pressing weights
        ii. Walking
        iii. Running

iv. bouncing a ball (basketball)
v. kicking a ball
vi. catching a ball thrown to self
vii. Jumping
viii. Star jumps

4. For all identified classes the classification process should be accurate in ten different environments.
    a. Indoor good lighting
    b. indoor dark or poor (dark) lighting
    c. Indoors poor lighting and noise in background e.g. people walking around
    d. Indoors good lighting and noise in background
    e. outdoors good lighting
    f. outdoors good lighting and noise in background.
    g. outdoors poor lighting.
    h. outdoors good lighting and noise in background
5. The project should be trainable and testable on a desktop pc or drone (excluding training)
6. When training and testing in the pc, the system should select at each step the best of a series of videos of the same action taken from different perspectives, simulating the drone's movement.
7. The software should be runnable on a drone and the drone should be pilotable by one of the users.

3.1 External Interfaces

Name of item: Drone camera
description of purpose: provides video data in avi format to system
Source of input or destination of output: Camera mounted to drone to have data taking by data processing system. All camera footage is then converted into slides of h5f format data held on the desktop machine for training.
Data formats: AVI

Name of item: Training cameras
description of purpose: multiple cameras to provide three or more way perspective recording.
Source of input or destination of output: All camera footage is then converted into slides of h5f format data held on the desktop machine for training.

Data formats: AVI

3.1.1 User interface

There will be an interface for recording, selecting, labelling and training and testing over training or test data. The system should provide a user interface to let the user know if there is a better orientation that the drone can move into in order to classify the behaviour being witnessed.

3.2 Functional requirements

The system shall be able to classify sixteen different actions (group activities - two or more people: two people hugging, two people fighting, two people talking - crossing arms, two people throwing a ball to each other, one person being pickpocketed by another person from behind, two people hi fiving, two people shaking hands, two people waving at each other; individual actions: Bench pressing weights, Walking, Running, bouncing a ball (basketball), kicking a ball,catching a ball thrown to self, Jumping
Star jumps) across ten different environments (Indoor good lighting, indoor dark or poor (dark) lighting, Indoors poor lighting and noise in background e.g. people walking around,Indoors good lighting and noise in background, outdoors good lighting, outdoors good lighting and noise in background, outdoors poor lighting,
outdoors good lighting and noise in background) and half of these will involve two or more persons or just a single person.

The system shall have a degree of redundancy so that if the wrong data is input into the system it will be recognised as faulty in some way and recover from a case of faulty input. In order to get the ‘best’ data the drone should be able to suggest to the end user where to orientate the drone to in order to get the best data.

3.3 Product Requirements

ID: FR1
TITLE: Reusable dataset
DESC: The dataset used shall be in a format which will constitute it reusable for other applications in the future.

3.4 Performance Requirements

ID: PR1
TITLE: Recognition speed
DESC: The system should be able to recognise and label human actions in almost real time. This means that the system needs to be able to either predict and classify or classify having seen the entire action. It must therefore be able to process input to the system as quickly as frames are taken or ‘FPS’ in order to provide real time classification. The vicon camera operates at 100 FPS so the program must be able to process video at this rate.

ID: PR2
TITLE: Rate of failure
DESC: The rate of failure occurring should not be above 10%.

ID: PR3

TITLE: Input stream cleaning
DESC: Once the input file stream reaches 100 MB in size, the first 50 MB should be deleted to preserve space and not cause delays in the system processing time.

ID: PR4
TITLE: Recognition performance
DESC: The system should have an accuracy of at least 60%, meaning that it should predict the correct action label 60% of the time.

3.5 Design Constraints

3.5.1 Standards Compliance

If the software is operational on a drone then it has to comply with local and national regulations this includes not flying the drone within a 1km radius of an airport or higher than 400ft [6].

ID: PR5
TITLE: Drone security compliance
DESC: Comply with the safety measures as per articles 241,94,94A,94B,95.

3.6 Software System Attributes

ID	Characteristic	H/M/L	1	2	3	4	5	6	7	8	9	10	11	12
1	Correctness	H												
2	Efficiency	L												
3	Flexibility	M												
4	Integrity/Security	L												
5	Interoperability	M												
6	Maintainability	M												
7	Portability	H												
8	Reliability	H												
9	Reusability	L												
10	Testability	H												
11	Usability	M												
12	Availability	M												

3.6.1 Reliability

ID: PR6
TAG: System Reliability
GIST: The reliability of the system.

SCALE: The reliability that the system gives the right result on action recognition.
METER: Measurements obtained from 100 actions observed during testing.
MUST: More than 90% of the actions.
PLAN: More than 95% of the actions.
WISH: 100% of the actions.

ID: PR7
TITLE: Open avi file redundancy
DESC: If the avi file couldn't be loaded for training then the user needs to be informed in that case.

3.6.2 Availability

ID: PR8
TAG: System Availability
GIST: The availability of the system when it is used.
SCALE: The average system availability.
METER: Measurements obtained from 100 hours of usage during testing.
MUST: More than 98% of the time.
PLAN: More than 99% of the time.
WISH: 100% of the time.

ID: PR9
TITLE: Connection to the camera/vicon.
DESC: The application should be able to be connected to the camera/vicon.

3.6.3 Security

ID:PR10
TITLE: drone security measures
DESC: The main concern in terms of security for the solution should be that it cannot in some way be hijacked in some way through the network interface if it is operating on the drone.

3.6.4 Maintainability

ID: PR11
TITLE: Dataset reusability
DESC: Dataset created should be easy to extend and/or reuse in another application at a future time.

3.6.5 Portability

D:PR12
TITLE: Application portability
DESC: If the application is running on the drone the software should be able to run on it to automate the data collection process.

ID: PR13
TITLE: Application portability
DESC: The application should be able to run on any Windows machine from Windows XP and onward.

ID: PR14
TITLE: File format
DESC: A common avi file shall be saved or loaded for training or testing and then it should be converted to stills in an h5f format.

ID: PR15
TITLE: File format
DESC: The state of the trained system should be saveable so that training doesn't need to be performed across all devices before testing.

3.6.6: Correctness

ID:PR16
TITLE: Correctness
DESC: The product should strive to meet the users requirements in full for both drone automation and for classification on a desktop pc.

3.6.7: Efficiency

ID:PR17
TITLE: Efficiency of product or solution on drone and pc
DESC: The program should be portable enough to work on a drone or a pc. If on a drone then that system needs to work within a constraints of the system requirements for that system.

3.6.8: Flexibility

ID:PR18
TITLE: Extra functionality
DESC: The drone should be designed in a way to have an architecture that it can be extended with extra functionality without eroding the core architecture. In the case of the product it will be developed on a personal computer and then have functionality for any given drone built on top of it.

3.6.9: Interoperability

ID:PR19
TITLE: Loose coupling
DESC: The system should be designed in a way that it is loosely coupled to give it the flexibility to have other systems built on top of the classification software.

3.6.10: Testability

ID:PR20
TITLE: Acceptance testing
DESC: The software should meet all of the specific requirements. The end user should test the software to make sure that it meets these requirements.

3.6.11: Usability

ID:PR21
TITLE: Usability of the user interface.
DESC: When the application is running on a desktop pc the user should be able to select data sets to train on or test over or record data sets. When running on a drone the drone should give visual cues on where to move in order to get the best viewpoint to classify the subject.

3.6.12: Reusability

ID:PR22
TITLE: dataset reusability.
DESC: The dataset produced for the software should be reusable elsewhere. Any data logged from the testing process should be reusable elsewhere.

3.7 Other Considerations

3.7.1 System Modes

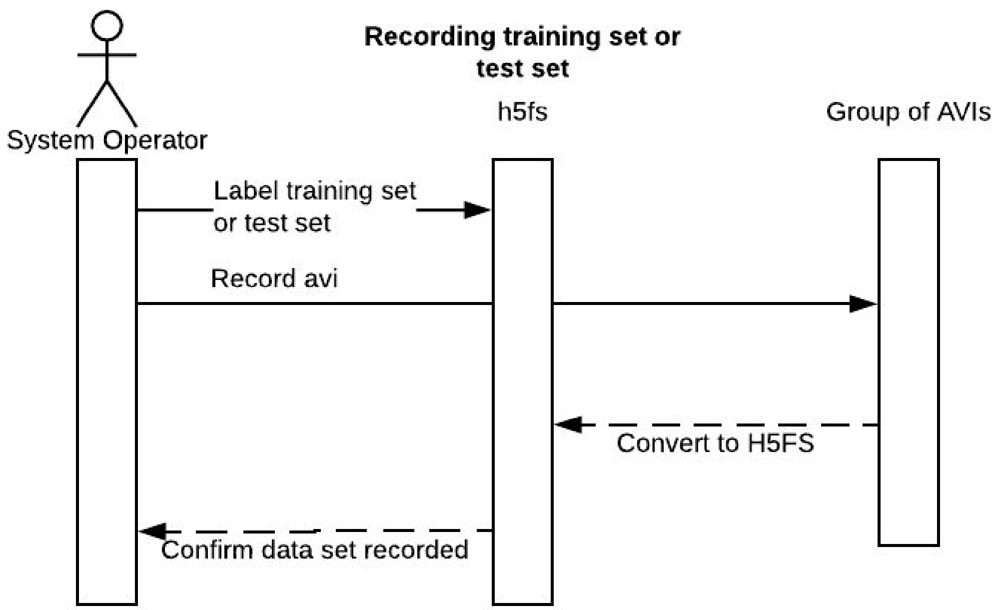
The system has two main modes of operation: training and testing; and classification with a drone. The training phase of the system takes avi input with a label, the testing phase takes a video without a label. The application can be testable or trainable on a desktop pc.

If the system is running on a drone then it should take real time video and classify on the fly.

3.7.2 User Classes
Researchers
R&D engineer
Actor or Actors
Drone system

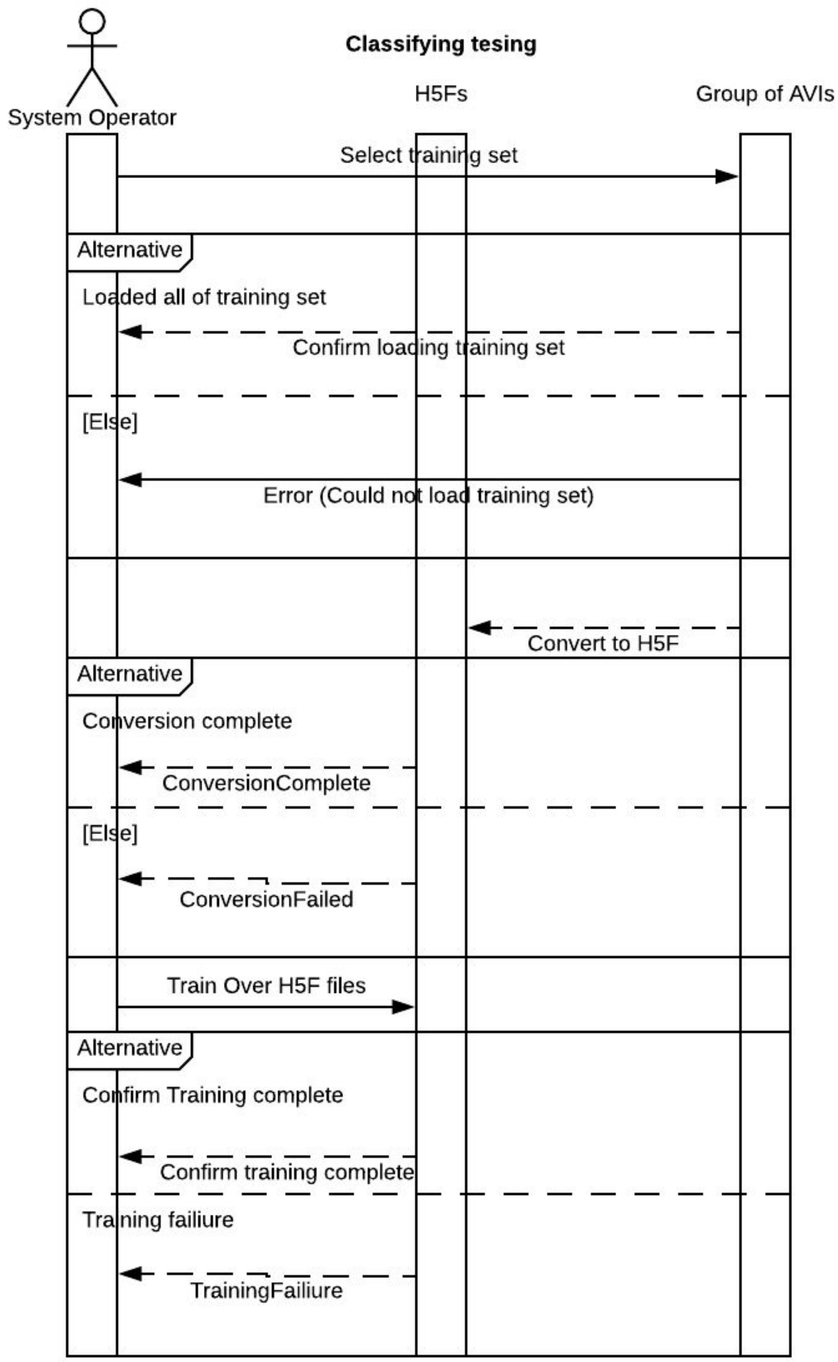
3.7.3 Interaction Diagrams

Record Dataset (testing or training):

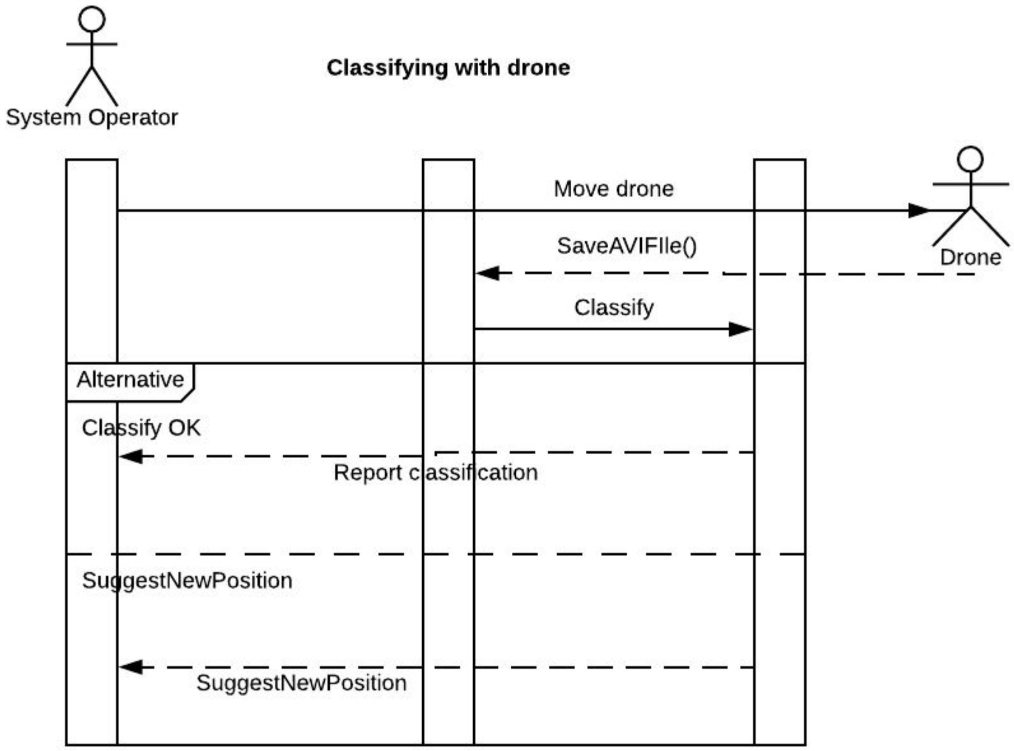


Train Network:

The diagram below describes how avi files are selected and how training is performed.



Classify on drone:



3.7.4 Feature

The movement of the drone may be considered to be an externally desired service as it's movements need to be controlled by a hardware interface separately from the system.

3.7.5 Stimulus

The system should respond to stimulus from frames recorded when a training or test set is loaded.

The system should issue suggestions on where to move in order to get the best orientation to collect data based upon current video frames.

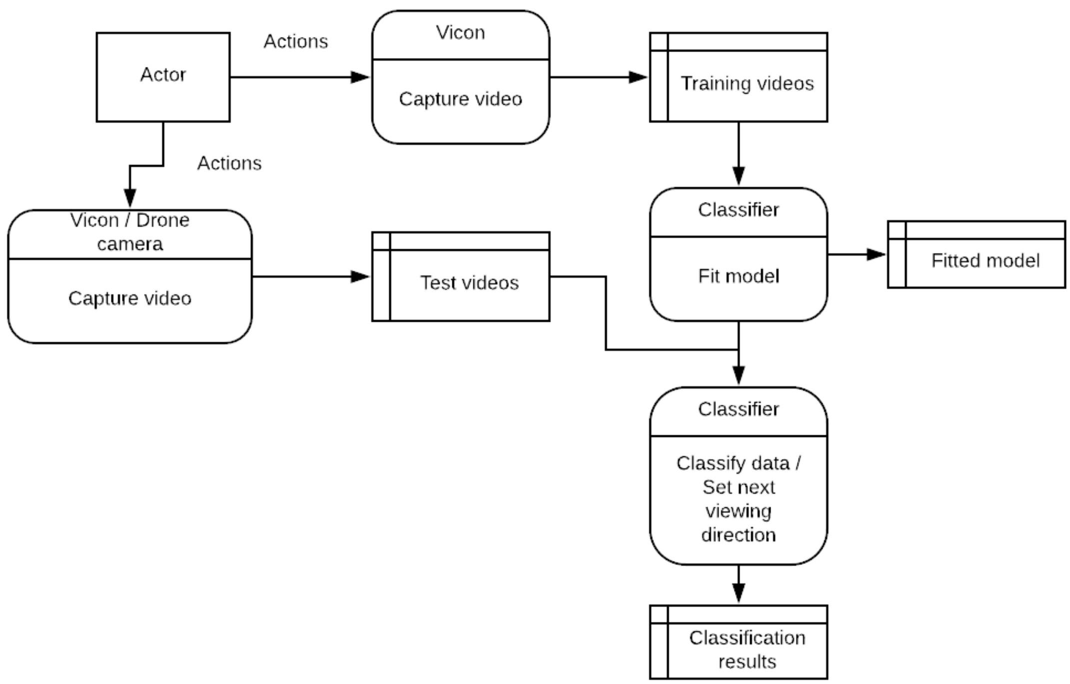
If the system detects a classification then it should inform the UI system that a classification has been made and also alert the user to the classification class found.

3. 7.6 Response

The drone system should respond to movement commands from the user. The drone should be viewed as an external system for its movement controls (hardware interface).

3.7.7 Data Flow

Data Flow



The diagram above shows the flow of data from the moment the video is recorded to the classification for the training and testing processes in a high level. As seen, the resulting outputs are the training videos, the testing videos, the fitted model, and the results of the classification, which can then be compared to the original labels.

4. Project management

A waterfall software model process was selected for the development of this project, as it allows a division of the project into well-defined sequential stages and permits a complete tracking of the progress. This implementation was possible because the requirements for the system were known in advance, fixed and understood by both sides (developing team and customer). Plus, the waterfall model suits completely with the short time for the project, and the one-run development, as no modifications can be made once the process has started because they would imply delays.

The project is divided in two big stages, in accordance to the delivery dates. The first one, consists of the development of the Software Requirements Specification, and the second one is the execution. These two stages contain the five common waterfall model phases, as shown in the chart.

SRS Stage	Requirements analysis and definition
	System and software design
Development Stage	Implementation and unit testing
	Integration and system testing
	Operation

The Gantt chart and network diagram are attached at the end of this file.

5. Change Management Process

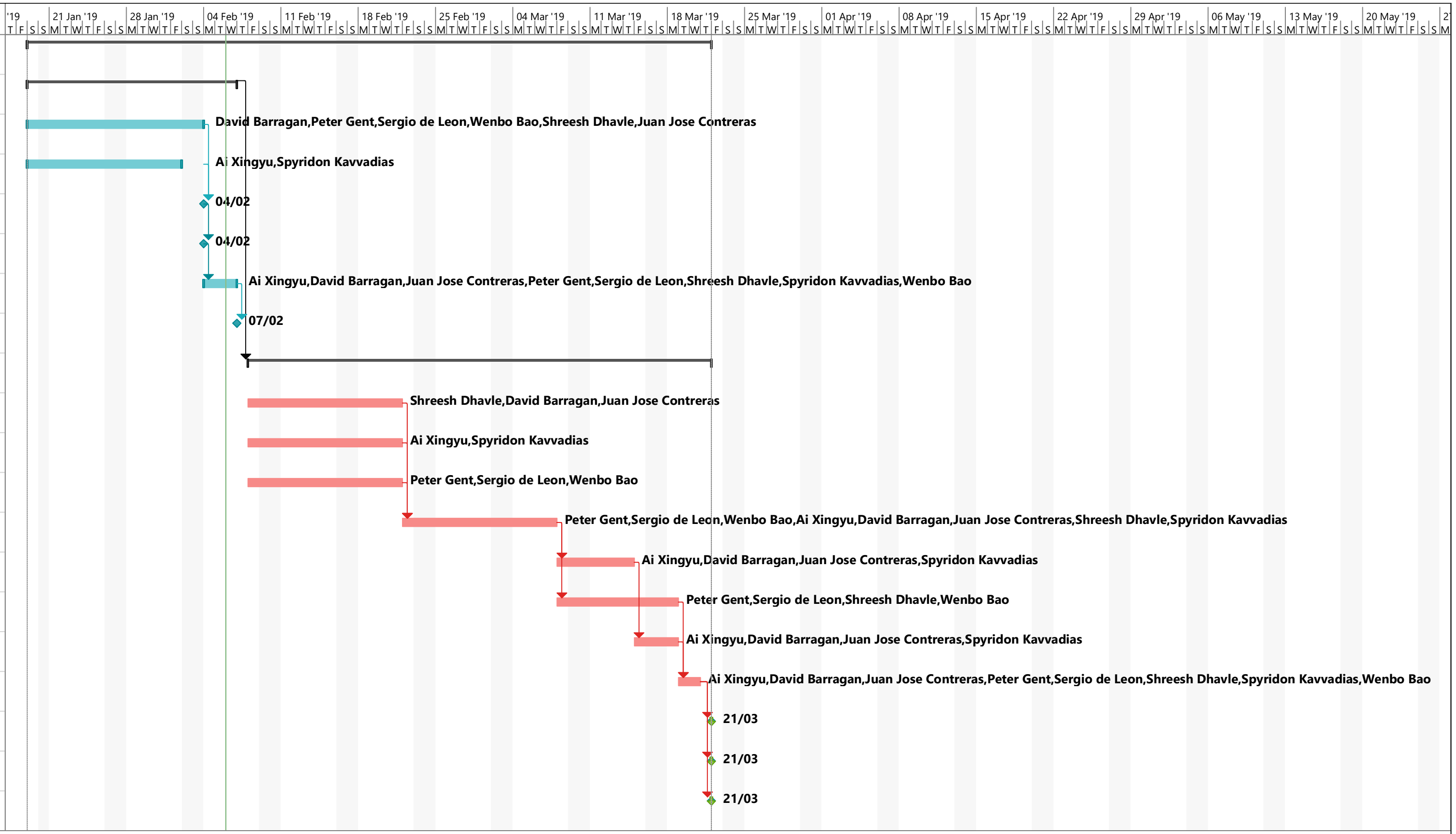
In order to make a significant change to the document the product owner has to agree to changes with the customer. The product owner in this case is Spyridon Kavvadias.

6. Document Approvals

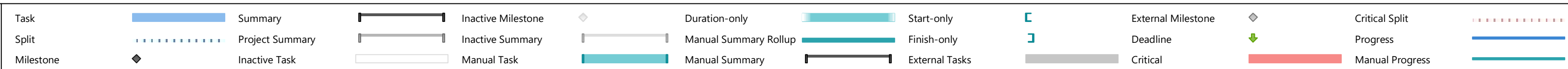
Spyridon Kavvadias
Ai Xingyu
Wenbo Bao
Sergio de León
Juan José Contreras
Dave Barragan
Shreesh Dhavle

Gantt Chart

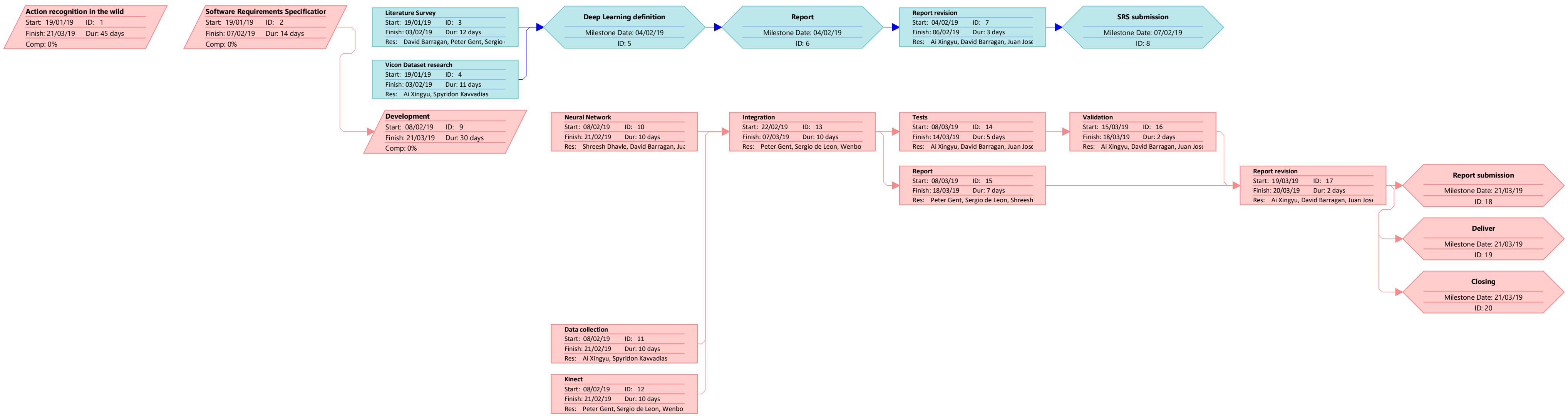
ID		Task Mode	Task Name	Duration	Start
1			**Action recognition in the wild**	**45 days**	**Sat 19/01/19**
2			**Software Requirements Specificati**	**14 days**	**Sat 19/01/19**
3			Literature Survey	12 days	Sat 19/01/19
4			Vicon Dataset research	11 days	Sat 19/01/19
5			Deep Learning definition	0 days	Mon 04/02/19
6			Report	0 days	Mon 04/02/19
7			Report revision	3 days	Mon 04/02/19
8			SRS submission	0 days	Thu 07/02/19
9			**Development**	**30 days**	**Fri 08/02/19**
10			Neural Network	10 days	Fri 08/02/19
11			Data collection	10 days	Fri 08/02/19
12			Kinect	10 days	Fri 08/02/19
13			Integration	10 days	Fri 22/02/19
14			Tests	5 days	Fri 08/03/19
15			Report	7 days	Fri 08/03/19
16			Validation	2 days	Fri 15/03/19
17			Report revision	2 days	Tue 19/03/19
18			Report submission	0 days	Thu 21/03/19
19			Deliver	0 days	Thu 21/03/19
20			Closing	0 days	Thu 21/03/19



Project: Action recognition in th
Date: Wed 06/02/19



Network Diagram



Project: Action recognition in th
Date: Wed 06/02/19

Critical		Critical Milestone		Critical Summary		Critical Inserted		Critical Marked		Critical External		Project Summary		Highlighted Noncritical	
Noncritical		Milestone		Summary		Inserted		Marked		External		Highlighted Critical			