



Generate Navigations to Guide and Automate Endotracheal Intubation Process

R.M.R. Wanigasekara

Index No : 14001502

S.D.M.H. Siyambalapitiya

Index No : 14001411

S.D.S.H. Dissanayake

Index No : 14000271

Supervisor: Dr. K.L.Jayaratne

Co-Supervisor: Dr. M.I.E Wickramasinghe

January 2019

Submitted in partial fulfillment of the requirements of the
B.Sc(Hons) in Software Engineering 4th Year Project (SCS4123)



Declaration

I certify that this dissertation does not incorporate, without acknowledgment, any material previously submitted for a degree or diploma in any university and to the best of my knowledge and belief, it does not contain any material previously published or written by another person or myself except where due reference is made in the text. I also hereby give consent for my dissertation, if accepted, be made available for photocopying and for interlibrary loans, and for the title and abstract to be made available to outside organizations.

Candidate Name: Mr. R.M.R. Wanigasekara

Signature of Candidate:..... Date:.....

Candidate Name: Ms. S.D.M.H. Siyambalapitiya

Signature of Candidate:..... Date:.....

Candidate Name: Ms. S.D.S.H. Dissanayake

Signature of Candidate:..... Date:.....

This is to certify that this dissertation is based on the work of Mr. R.M.R. Wanigasekara, Ms. S.D.M.H. Siyambalapitiya, Ms. S.D.S.H. Dissanayake under my supervision. The thesis has been prepared according to the format stipulated and is of an acceptable standard.

Supervisor's Name: Dr. K.L. Jayaratne

.....

Signature of Supervisor

Date:

Co- Supervisor's Name: Dr. M.I.E. Wickramasinghe

.....

Signature of Supervisor

Date:

Abstract

In the medical field, Endotracheal Intubation is an emergency procedure that's performed on people who are unconscious or who cannot breathe on their own. The endotracheal tube which is put through the mouth or nose serves as an open passage through the upper airway. The purpose of Endotracheal Intubation is to permit air to pass freely to and from the lungs to ventilate the lungs.

Many complications can occur in the Nasotracheal Intubation process such as bleeding, infections, damages to the voice box (larynx), thyroid gland, vocal cords and windpipe (trachea), or pharynx. To overcome the consequences, we propose a system to guide and automate endotracheal intubation by providing the physician with necessary navigations for moving the intubation tube. This study is a proof of concept to prove that generating navigations using a trained model based on Convolution Neural Network has practical potential.

In the proposed system, different anatomical landmarks along the path from nose to carina are identified, captured images are preprocessed and use a Convolution Neural Network based on image classification to predict the class of each anatomical landmark. The proposed system has been trained using the training dataset of 9753 images, and the performance of the system was evaluated using the testing dataset of 2438 images.

The results, obtained using the pre-trained model, show that the system correctly classified 2385 out of 2438 (97.8%).

Acknowledgment

The authors of the thesis take this opportunity to gratefully acknowledge the assistance and contribution of supervisors and advisors who have been helping throughout the project.

First and foremost we would like to express gratitude to the Supervisor Dr. K. L. Jayaratne and the Co-supervisor Dr. M. I. E. Wickramasinghe for their guidance, encouragement and immense knowledge to do a better quality project.

We are grateful to Professor Aruna Munasinghe for giving this idea of this research project and for providing us with basic knowledge related to the research area.

Finally, we would like to thank Dr. S. Geethal Perera (Consultant Pulmonologist), Sri Jayewardenepura General Hospital, Dr. R. De S. Karunatilake (Consultant Respiratory Physician), National Hospital Sri Lanka and Dr. R. Asha Samaranayake (Residential Pulmonologist) National Hospital Sri Lanka, for their valuable commitment for supporting us with necessary resources.

Table of Contents

Declaration	i
Abstract	ii
Acknowledgment	iii
Table of Contents	iv
List of Figures	vi
List of Tables	viii
List of Acronyms	ix
Chapter 1 - Introduction	1
1.1 Background to the Research.....	1
1.2 Problem Definition	2
1.3 Objectives.....	3
1.4 Motivation	3
1.5 Delimitation of Scope.....	4
1.6 Methodology	4
1.7 Justification for the project.....	4
1.8 Outline of the Dissertation	5
Chapter 2 - Background Study.....	6
2.1 KIS (Kepler Intubation System).....	6
2.2 Endotracheal Tube Position Confirmation System	7
Chapter 3 - Analysis and Design	8
3.1 Problem Analysis	8
3.2 Functional Requirements.....	10
3.3 Non-functional Requirements	10
3.4 System Overview	11
3.5 High-level Architecture	12
3.6 System modeling	15
Chapter 4 - Implementation	16
4.1 Software Development Process Model	16
4.2 Tools and Technologies	16
4.3 Justifications for tools and technologies	17
4.4 Graphical User Interfaces (GUI)	18

4.5 Methodology	24
4.5.1 Analysis of the path from the nose to the carina	24
4.5.2 Image Capturing and Preprocessing	27
4.5.3 Convolutional neural network for classification	27
4.5.4 Prediction of locations	34
Chapter 5 - Result and Analysis.....	43
5.3 Grayscale Intensity Histogram Analysis	43
5.4 Grayscale Local Binary Pattern(LBP) Histogram Analysis.....	45
5.5 Convolution Neural Network Approach (CNN)	46
6.1 Test items	47
6.2 Test approach	49
6.3 Test tools	51
6.4 Test meetings.....	54
6.5 Evaluation.....	55
Chapter 7 - Conclusion	57
7.1 Limitations	57
7.2 Future work	57
References.....	58
Appendix A: Diagrams	60
A.1 System modeling	60
A.1.1 Use Case Diagram	60
A.1.2 Use Case Narratives.....	61
A.1.3 Class Diagram.....	67
A.1.4 Activity Diagram	68
A.1.5 Sequence Diagram	69
Appendix B: Code Listings.....	71
B.1 pytest script.....	71
B.2 Feature Matching.....	71
B.3 Homography generation	72
Appendix C	73
C.1 Contribution.....	74
C.2 Recommendations.....	76
C.3 User Survey Questionnaire for the Endotracheal Intubation System.....	78

List of Figures

Figure 2.1: Kepler Intubation.....	6
Figure 3.1: Training and testing dataset.....	8
Figure 3.2: System overview	11
Figure 3.3: High-level architecture	12
Figure 4.1: Login Interface	19
Figure 4.2: Mode selection interface	20
Figure 4.3: Manual Intubation	21
Figure 4.4: Save snapshot and oprn file.....	30
Figure 4.5: Automatic Intubation.....	23
Figure 4.6: Anatomical Location 1	25
Figure 4.7: Anatomical Location 2	25
Figure 4.8: Anatomical Location 3	25
Figure 4.9: Anatomical Location 6	26
Figure 4.10: Anatomical Location 5	26
Figure 4.11: Anatomical Location 4	26
Figure 4.12: Anatomical Location 7	26
Figure 4.13: Anatomical Location 8	26
Figure 4.14: Anatomical Location 9	26
Figure 4.15: Anatomical Location 10	26
Figure 4.16: Standard convolutional layer.....	31
Figure 4.17: Depthwise separable layer.....	31
Figure 4.18: Smoothed graph training accuracy vs number of steps	33
Figure 4.19: Smoothed graph validation accuracy vs number of steps	33
Figure 4.20: Sample set of key points generated by SURF key point detectors.....	37
Figure 4.21: Sample set of quadrilateral projection in the images of videos.....	38
Figure 4.22: Sample set of pointer projection on the images of videos.....	39
Figure 4.23: Pipeline process of sub tasks	41
Figure 5.1: The sample set of intensity histograms of the images at anatomical location 1.....	44
Figure 5.2: The sample set of intensity histograms of the images at anatomical location 4.....	44

Figure 5.3: The sample set of intensity histograms of the images at anatomical location 1.....	44
Figure 5.4: The sample set of LBP histograms of the images at anatomical location 2	45
Figure 5.5: The sample set of LBP histograms of the images at anatomical location 2	45
Figure 5.6: The sample set of LBP histograms of the images at anatomical location 5	46
Figure 6.1: Incorrect pointer prediction	48
Figure 6.2: Correct pointer prediction	48
Figure 6.3: pytest test results	51
Figure 6.4: Pylint test results	52
Figure 6.5: Integrate Pylint with PyCharm	53
Figure 6.6: Ambiguous error notifications.....	53
Figure 6.7: Final Pylint rate for the corrected code	54
Figure A.1.1: Use case diagram	60
Figure A.1.3: Class diagram	67
Figure A.1.4: Activity diagram	68
Figure A.1.5.1: Send operation Details.....	69
Figure A.1.5.2: Save snapshots.....	70

List of Tables

Table 3.1: Training dataset	9
Table 3.2: Testing dataset	9
Table 4.1: Tools and technologies	17
Table 4.2: Pre-trained model for classification.....	29
Table 4.3: Set of actions per-formed at each location	35
Table 5.1: Accuracies in pretrained models.....	46
Table 6.1: Measures and Metrics	54
Table 6.2: Efficiency evaluation	56
Table C.1: Results of usability test	56

List of Acronyms

EI	-	Endotracheal Intubation
CNN	-	Convolutional Neural Network
CO ₂	-	Carbon Dioxide
OLI	-	One Lung Intubation
AI	-	Artificial Intelligence
KIS	-	Kepler Intubation System
GLCM	-	Gray Level Co-occurrence Matrices
CMOS	-	Complementary Metal-Oxide Semiconductor
ANN	-	Artificial Neural Network
IEEE	-	Institute of Electrical and Electronics Engineers
VGG	-	Visual Geometry Group
GUI	-	Graphical User Interface
LBP	-	Local Binary Patterns
SIFT	-	Scale Invariant Feature Transform
GIL	-	Global Interpreter Lock

Chapter 1 - Introduction

1.1 Background to the Research

Endotracheal Intubation is a relatively common procedure provided by trained physicians in emergency and scheduled conditions in which the patient ceases or cannot breathe on their own. It is the placement of a flexible tube into the trachea (windpipe) of a patient to maintain an open airway in patients. This tube can be inserted into the respiratory system through the mouth or the nose. Orotracheal Intubation is the process where the tube is inserted through the mouth while Nasotracheal Intubation inserts the tube through the nose for the same purpose. This project is aimed at automation of Nasotracheal Intubation process.

Currently, inserting the endotracheal tube into the trachea is done by the physicians with their practice. For this purpose, tools are used such as “laryngoscope” which assist in the process. However, the current approach is not very effective because there is no camera configured with the tube to see inside the body after the tube is inserted through the nose. Therefore this has to be done by an experienced and well-trained physician to avoid any difficulties that may occur due to the lack of visuals inside the body.

Nasotracheal intubation is performed on patients undergoing maxillofacial surgeries (treating many diseases, injuries, and defects in the head, neck, face, jaws and the hard and soft tissues of the mouth and face region) or dental procedures or when oro-tracheal intubation is not feasible (e.g., patients with limited mouth opening, patients who have serious damages due to accidents). In these cases, insertion of the tube manually could be dangerous because if the tube is not in the right area of the trachea, the patient may not get the oxygen he needs. Decreased levels of oxygen can lead to permanent organ and brain damage, and death. On rare occasions, the anatomy of the patient does not allow easy insertion of the Endotracheal Tube, and consequently, the tube might be incorrectly positioned in the patient’s respiratory system.

Several methods have been proposed to verify the correct positioning of the tube. For many years auscultation of breath sounds over the lungs using a stethoscope was the most widely used technique as it utilizes very accessible tools. This technique

requires high attention, lack of environmental noise and it suffers considerable false positive and false negative rates which significantly affecting its reliability.

Recently measurement of CO₂ (Carbon Dioxide) in the air has become the standard practice. Since one of the main functions in the lung is to eliminate CO₂ by exhalation the substantial measuring amount of CO₂ in the exhaled air proves that the endotracheal tube is located in the airways. This method suffers however from significant shortcomings. It is not capable of differentiating proper positioning from a too deep one. Additionally, it is unreliable for EI confirmation in many medical emergencies such as cardiac arrest.

1.2 Problem Definition

As mentioned in the above section, the main problem of the current Intubation approach is that it does not provide sufficient visuals inside the body. A critical point which can be identified along the path is the point where the Intubation tube meets the esophagus opening (food pipe) and trachea (windpipe). The tube must not go into the food pipe because if this occurs, the patient may vomit and food or fluid may enter his lungs which can cause a lung infection.

Unrecognized esophageal Intubation carries fatal risks as the stomach is ventilated rather than the lungs, resulting in severe oxygen deficiency in the vital organs. In cases of one lung intubation (OLI), only one lung is ventilated which might cause serious complications such as the collapse of the non-ventilated lung, yielding reduction in oxygen tension in the blood or lung infection. OLI happens when the tube passes the carina (the point where the respiratory system divides into two lungs). Therefore, the physician has to be very careful not to pass the tube this critical point.

Therefore, a proper method must be suggested to do Intubation more accurately and effectively.

1.3 Objectives

As explained in the description, Nasotracheal Intubation is currently done by the physicians with their practice and the experience. Since this approach is almost a blind approach, it could lead to serious complications. Therefore, the objective of the proposed system is to automate the Endotracheal Intubation process.

The developed system focuses on Nasotracheal Intubation and generates navigations to guide the physician through an interface on where the endotracheal tube should be moved according to the current tube location. This study is a proof of concept to prove that guiding the physician through a device that predicts navigations using neural network methodologies has practical potential. The route for Nasotracheal intubation is the path from nose to the trachea and stops just above the carina (between the divisions of the two main airways to the lungs). The trachea is the target structure in path detection for endotracheal intubation process.

1.4 Motivation

This study is motivated by an incident where a snake moves into a snake burrow. When the snake wants to move inside, it places its head near the burrow hole and pulls its body into the burrow smoothly by bending the body accordingly because it already knows the exact path inside the burrow. It knows where to move exactly because it has been through this path earlier and the path is in its brain.

According to the professional opinion, the same concept can be applied to move the intubation tube along the path from nose to carina without any human interaction. First, guide the tube by training a model to recognize several anatomical landmarks and move along a predefined path. After training, when the tube is placed inside the nose, the tube can move into the body by analyzing current tube location and deciding which direction to move.

(Discussion with Prof. Aruna Munasinghe at National Hospital Sri Lanka, February 2018)

1.5 Delimitation of Scope

The ultimate goal of this project is to build an AI to automate endotracheal intubation process without any human interactions fully. However, this study only focuses on guiding the physician through the process by giving navigations on where to move the tube especially when the tube meets trachea (windpipe) and the esophagus. This system only considers the path from nose to the carina. It does not consider the path after passing this point which means the suggested path is not extended up to the lungs. Another point to be noted is, this study is done as a proof of concept to prove that training a model to recognize a predefined path has practical potential. The system contains no hardware device because it is out of software engineering project scope.

1.6 Methodology

This system is focused on generating navigation details to guide the physician based on a pre-trained model based on image processing and neural network methodologies. First, the path from nose to carina is analyzed. The path is segmented into different regions which can easily be differentiated. These different regions are known as anatomical landmarks. Identified anatomical landmarks will play an important role in indication correct tube position. Then, images of each anatomical landmark are obtained and train a model to classify these images into their classes. Then each image is annotated with the navigation detail where the tube should be moved depending on the current location of the tube. The system is developed mainly using python, TensorFlow and OpenCV.

1.7 Justification for the project

Software engineering project focuses on the development of software products with the main intention of doing an innovation rather than an invention. From the facts that were gathered from the expertise, there is no Artificial Intelligent system to automate the endotracheal intubation process fully. Hence this will be an innovative idea.

The concept mentioned above has a research component and a development component as well. Image analysis and feature extraction from those images is a complex part that was addressed. Based on those features the best classification method for classifying the images was found out. To depict the results of those research

components, a desktop application is implemented which takes videos as inputs and outputs navigations that need to move the intubation tube according to the current tube location. Hence the system can be named as a research-based software engineering project.

Software engineering is the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software (IEEE Standard Glossary). After considering the overall characteristics and capabilities of the members and the time constraints, an agile methodology was adopted as the software engineering process model. Version controlling and code reviews (peer reviews) was carried throughout the project to maintain best coding structures and increase the quality of coding. This project did a feasibility study and background research on similar products before starting of the development.

To maintain the quality of this application quality assurance principles are applied for the process from the beginning of the project. Hence, this study could reach its goal with a quantifiable approach. Since a particular software product to a real-world application is developed, this project falls into the product-based software engineering project category. As the study follows the above software engineering techniques and principles, this is a software engineering project and to be more specific; this is a product-based software engineering project.

1.8 Outline of the Dissertation

The dissertation contains six main chapters each dedicated to an important aspect of the system. Chapter 1 provides an overview of the system from problem definition to the conclusion. Chapter 2 discusses the related work carried out regarding the Endotracheal Intubation. Chapter 3 gives a detailed description of problem analysis and system design along with UML diagrams. Chapter 4 describes the tools and technologies used in the implementation process. Chapter 5 discusses the approach taken, issues identified in each approach and selected approach in detail. Chapter 6 explains how the testing was carried out and how the system was evaluated. Chapter 7 provides the conclusion as the final chapter of the body of the dissertation.

Chapter 2 - Background Study

This chapter discusses approaches and systems related to this project. According to the findings, currently ‘Endotracheal Intubation’ is done manually by doctors and there is no software oriented mechanism to do this process. Though the complete process is not available, some approaches that have been implemented in automation manners which have been developed for different purposes.

2.1 KIS (Kepler Intubation System)

Kepler Intubation System is a project that has been done to develop a robotic intubation system. The intubation process is controlled by a robot arm and which was controlled by a center joystick. KIS consist of joystick, JACO robotic arm, a video laryngoscope, and a software control system. The process in this system is that the video laryngoscope provide real-time video and it is visible for the physician on a screen. From that, the physician controls the robotic arm by moving the joystick, and the software control system manages the actions of robot arm corresponding to the joystick movements. In here an assistant is needed to open and control the mouth while performing robotic intubation. This Kepler Intubation has performed for 12 patients and intubation in all was successful except one, and that is due to fogging of the video laryngoscope.



Figure 2.1: Kepler Intubation

The KIS is a semi-automated approach because the controlling process and the identification of the human body parts are done by the practice of the physician. The drawback in this system compared to the proposed system is the semi-automated principle. The goal of the proposed system is a proof of concept which shows that intubation can be fully automated. This study proves this by developing a software system that can give navigations to this type of robotic arm. KIS can be used as an extension of this project which the actions predict by this system can be used to control the KIS.

2.2 Endotracheal Tube Position Confirmation System

This system was developed to verify correct endotracheal tube positioning. The approach is based on direct visual cues such that identification of specific anatomical landmarks as indicators of correct or incorrect tube positioning. It is developed and evaluated using animal and human tissue models. The system is based on identifying the carina in the acquired video image and other anatomical structures. This system is composed by an artificial neural network which is used to identify images using the texture based features of the images. In particular, features based on gray level co-occurrence matrices (GLCM) are used in this approach. This approach has designed and assembled a system which resembles an intubating stylet with its camera at the tip. The tip comprises a miniature CMOS sensor. The inner part of the stylet contains wires and a narrow lumen. The image sensor is connected to a processor with an integrated image acquisition component. The neural network process those images, classify accordingly and provide the likelihood score and then confirm the carina.

This tube positioning confirmation system satisfies the requirement of identifying carina, and that process has been done in fully automated. The gap between this approach and the proposed system is that the proposed system can identify a full path from nose to the carina.

Chapter 3 - Analysis and Design

This chapter provides a brief description of problem analysis and overall description of functional requirements, non-functional requirements, software engineering principles in detail. Further, it discusses architectural design along with any design assumptions and model diagrams both static and dynamic.

3.1 Problem Analysis

Although the ultimate goal of this research concept is to automate the Endotracheal Intubation process fully, this study only focuses on generating navigation details using a pre-trained model.

Dataset

Since there are not enough saved Nasotracheal Intubation images to train a model, the system had to acquire saved videos of the intubation process to obtain images. In Sri Lanka, there is no sophisticated equipment which can record the intubation process. Therefore, it was suggested by the project advisor that the best alternative is to use Bronchoscopy videos to obtain images and they were collected from the Sri Jayewardenepura General Hospital. Although those two are performed for two separate purposes, both go on the same path.

Initially, the whole dataset is divided as 80% and 20% for training and testing purposes respectively.

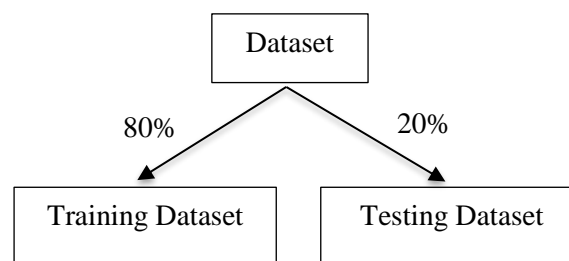


Figure 3.1: Training and testing dataset

Training Dataset

According to the dividing factor, the training dataset consists of 9753 images. Table 3.1 illustrates the arrangements of the training dataset.

Table 3.1: Training dataset

Class	Number of images	Percentage
Anatomical Landmark 1	880	9.02%
Anatomical Landmark 2	929	9.52%
Anatomical Landmark 3	1001	10.26%
Anatomical Landmark 4	951	9.75%
Anatomical Landmark 5	1012	10.37%
Anatomical Landmark 6	1036	10.6%
Anatomical Landmark 7	924	9.47%
Anatomical Landmark 8	1010	10.35%
Anatomical Landmark 9	1046	10.72%
Anatomical Landmark 10	964	9.88%

Testing Dataset

According to the dividing factor, the testing dataset consists of 2438 images. The important fact is that this testing data is never being in the training dataset. Table 3.2 illustrates the arrangements of the testing dataset.

Table 3.2: Testing dataset

Class	Number of images	Percentage
Anatomical Landmark 1	220	9.02%
Anatomical Landmark 2	232	9.51%
Anatomical Landmark 3	250	10.25%
Anatomical Landmark 4	238	9.76%

Anatomical Landmark 5	254	10.41%
Anatomical Landmark 6	259	10.62%
Anatomical Landmark 7	231	9.47%
Anatomical Landmark 8	252	10.33%
Anatomical Landmark 9	261	10.7%
Anatomical Landmark 10	241	9.88%

3.2 Functional Requirements

Following are the main two functionalities of this system.

- Verify correct endotracheal tube positioning.
- Generate navigations to guide the physician.

This product is an independent application which performs its operations without being integrated with other software systems. It is a self-contained python application which only has interfaces to the physician. Following is the flow of actions in the system.

- Run the application.
- Connect the camera to the computer.
- Select the mode either Manual or Automatic.
- If the mode is manual, then observe the application annotations and move device accordingly.
- If the mode is automatic then, pointer indications can be used to send as input signals for a hardware device.

3.3 Non-functional Requirements

3.3.1 Accuracy

Since this system is a critical one which is implemented to automate a medical procedure the accuracy is highly needed. If wrong navigation is provided, it will cause to damages and bleeding. Accuracy is a major concern when the intubation tube meets trachea and esophagus because the tube must not go into the esophagus.

3.3.2 Usability

The main actor of this system is a doctor (physician), and hence the system should be very user-friendly. It should be easy to learn the system which supports all the expected requirements of the physician

3.3.3 Reliability

Software reliability is the ability of a computer program to perform its intended functions and operations in a system's environment, without experiencing failure (system crash). In this system, the failure rate must be extremely low because it will be really dangerous if the system fails during the procedure.

3.3.4 Performance

Performance requirements describe response time, resources required and the survivability. Since the system processes real-time data, it must be interactive, and the delays involved must be less.

3.4 System Overview

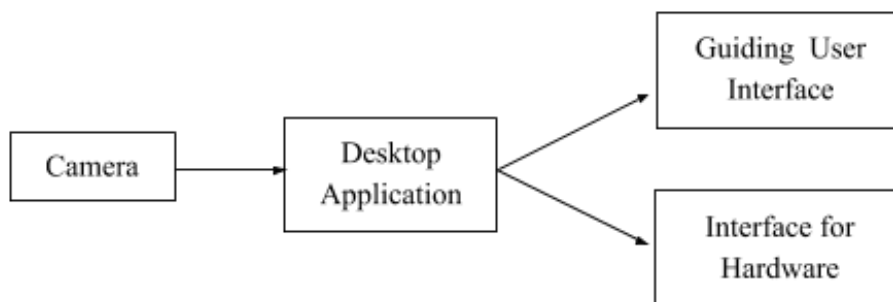


Figure 3.2: System overview

This figure shows the main components and the flow of information through the system. The endotracheal tube has attached a camera, and the camera feed is obtained. Then those images are preprocessed, classified and finally, navigations are generated and displayed through a separate interface designed for the physician. As a further improvement, generated navigations can be fed to a device to fully automate the Endotracheal Intubation Process.

3.5 High-level Architecture

This provides navigations to guide and automate endotracheal intubation process based on a layered architecture with each layer providing a clear set of functions to the layer above. Fig.1 depicts the high-level design of the system.

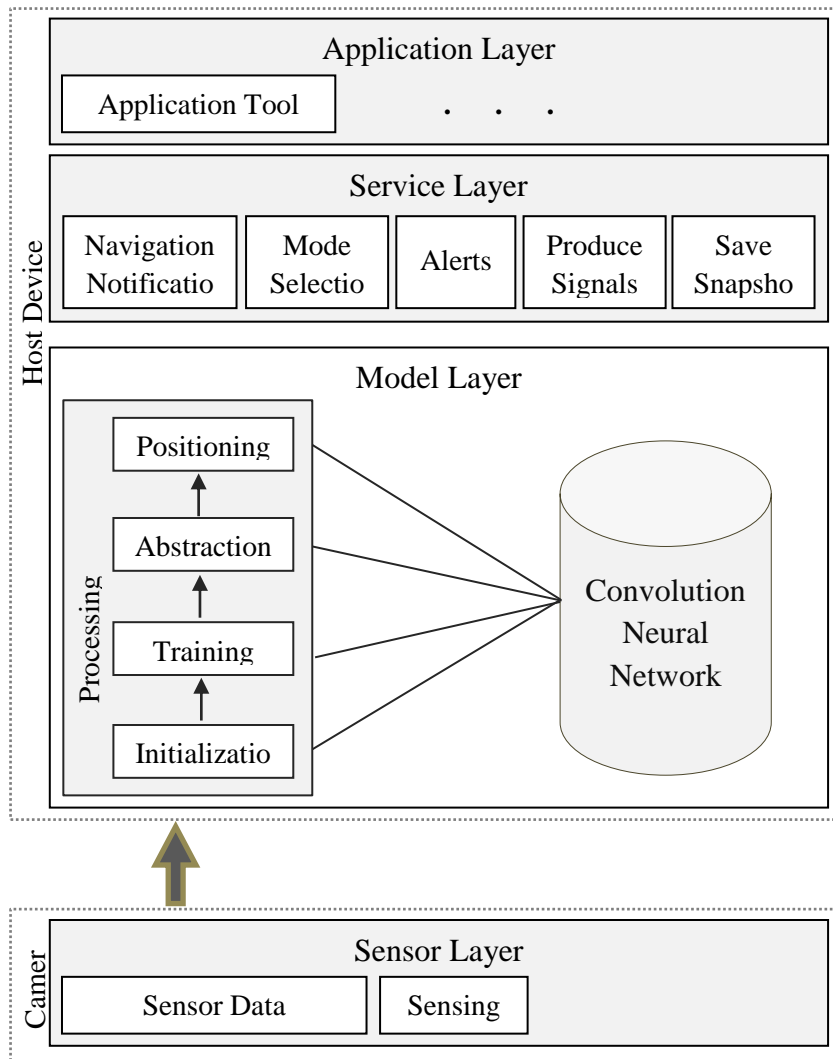


Figure 3.3: High-level architecture

The overall design of the project is composed of four layers: the sensor layer, the model layer, the service layer, and the application layer.

3.5.1 Sensor Layer

The sensor layer collects data from the configured camera as an input stream to the layer above.

1. Data Collection: The relevant camera inputs the live display of the intubation process. Frames of the video are considered as data to be input to the convolutional neural network.

3.5.2 Model Layer

The model layer feeds data provided by the sensor layer through a data processing pipeline and through the convolutional neural network it classifies data to the relevant classes. The pipeline involves initialization, training, abstraction, and positioning.

1. Convolutional Neural Network: For classification of images with great accuracy it is planned to use a convolutional neural network (CNN). As the number of data that can be collected is limited, it is better to use a pre-trained model for classification so that it can give a higher accuracy in the classification with less number of images. Out of the most used models such as AlexNet, Inception, Inception v2, Inception v3, MobileNetV2 and VGG 16, this system uses MobileNetV2.

In this system, the CNN is used in two different aspects. One is to train the model using training data set, and the other aspect is making predictions using the trained data. For training purpose, the model MobileNetV2 is used in the TensorFlow Hub and this MobileNetV2 model is a pre-trained model that has been trained from the ImageNet database. ImageNet database contains millions of image data that follows the WordNet (large lexical database in English) hierarchy. The nodes in this model are already trained in feature extraction. So it needs to be trained this model for classification.

After training the model for classification, the graph with weight for a model is saved. In the prediction, component predictor loads the graph and get the input image frame from the application and classify accordingly.

2. Data Processing Pipeline: The processing pipeline for the convolutional neural network consists of five stages, with the first one operating on the data provided by the sensor layer, and the following ones each operating on the output of the previous stage.

Initialization: The initial pipeline stage generates a new neural network at the beginning. It continuously reads frames from the input video. Each frame represents a matrix. The data it contains is extracted and added to the neural network.

Training: From a given set of ground truth, a convolutional neural network is trained for predictions.

Abstraction: The third stage of pipeline performs various abstractions on the quantitative output of the previous stage in order to generate qualitative information.

Positioning: The positioning stage operates on all sensor information available in the neural network and produces new position and orientation. After completion of this stage, the current neural network remains stored.

3.5.3 Service Layer

The service layer provides different applications with access to the convolutional neural network. This happens directly through alerts, navigation notification, and query mechanism.

3.5.4 Application Layer

In the application layer, user-level programs utilize services in the layer below. Applications can read specific attribute values associated with output nodes, and they can get a copy of the entire neural network. The six services available are navigation notification, alerts, mode selection, snapshot saving and produce signals.

3.6 System modeling

3.6.1 Use Case Diagram

The main actor of this system is the physician. The physician can select the intubation mode either manual or automatic. According to the selected mode, the relevant camera must be configured. Other use cases of the physician are observe the tube moving path, obtain navigations to move the tube, save snapshots of the current view, get notifications to stop the tube insertion process when the final anatomical landmark is reached and receive an alert when the tube passes the carina. (Appendix A.1.1 and Appendix A.1.2)

3.6.2 Class Diagram

The class diagram shows the relationships between classes namely application GUI, physician, Training, CNN, prediction, Service along with their attributes and operations. (Appendix A.1.3)

3.6.3 Activity Diagram

The activity diagram shows how the system achieves its goal and how data flows within the system. After uploading the video, navigation details are generated during the tube inserting process. When the tube meets final anatomical landmark, generate navigation details to move the tube outside the respiratory path. (Appendix A.1.4)

3.6.4 Sequence Diagram

Sequence diagrams depict the objects and classes involved in the generating navigation details and save snapshots. Further, it shows the sequence of messages exchanged between the objects needed to carry out the functionality of these two scenarios. (Appendix A.1.5)

Chapter 4 - Implementation

This application runs on a computer and one interface is available to integrate a camera, and another interface is available to send input signals to hardware which can be developed in future up to a fully automated system. This chapter provides implementation details, development tools, testing tools, and implementation tools with justification for selecting each tool.

4.1 Software Development Process Model

The iterative and incremental development process was followed at the implementation stage of the project. At first, video frames were captured and pre-processed. Then a neural network was trained with those images. When the new videos are obtained, the model had been trained over and over again. That means the process is iterative. As the other requirements were implemented, the system is incremented by one functionality. Main increments were the identification of anatomical landmarks, training of a model, generating navigation details and user interfaces integration. Hence the process is incremental.

4.2 Tools and Technologies

- Operating system: Windows 7 or above
- Database: No database system will be used, and the TensorFlow framework will store its knowledge in weights
- Development IDE: PyCharm
- Development language: python 3.6
- Application framework: PyQt5
- User Interface Design: Qt Designer
- TensorFlow Framework
- Image Processing tasks: OpenCV
- Visualization Tool: TensorBoard
- Version Controlling: GitHub

4.3 Justifications for tools and technologies

Table 4.1: Tools and technologies

Technology	Purpose	Reason	Version
python	For implementing neural network and do image processing.	<ul style="list-style-type: none"> • Many libraries for image processing and neural network exist for python which will facilitate the development process. 	python 3.6
PyCharm	IDE for developing python application.	<ul style="list-style-type: none"> • Ability to integrate with GitHub • Ability to integrate with pytest testing framework. 	PyCharm 2018.1.4 [community Edition]
TensorFlow	Framework for developing and using Neural networks.	<ul style="list-style-type: none"> • Availability of existing neural network models. • Availability of functions necessary for operation in neural networks development • Free and open source 	TensorFlow 1.10.0
OpenCV	For image processing purposes.	<ul style="list-style-type: none"> • Possess a variety of image processing techniques. • Have much fast image processing techniques. • Free and open source 	3.4.0
PyQt	Application framework for developing the desktop GUI	<ul style="list-style-type: none"> • Rapid prototyping tool • Posses a Qt Designer, a graphical user interface designer 	PyQt5

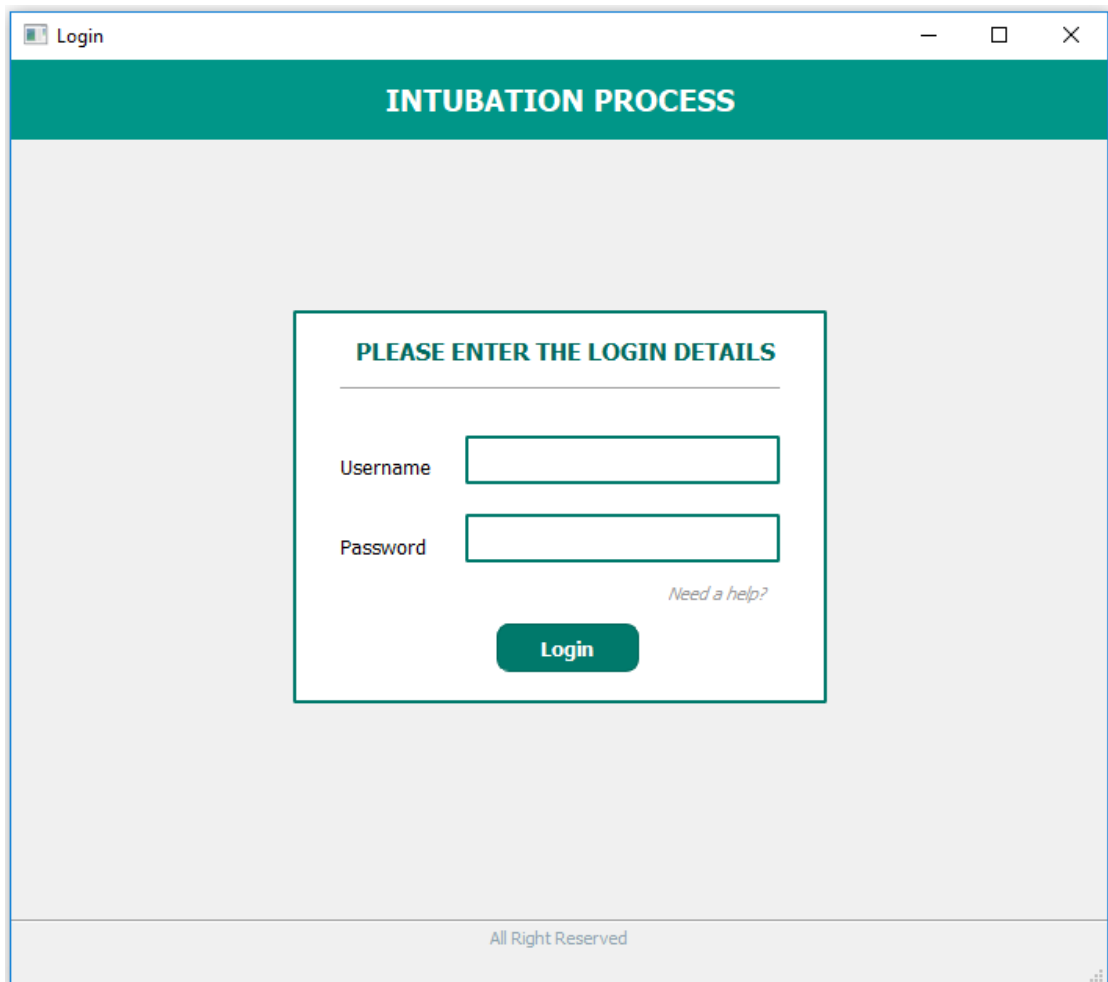
Qt Designer	GUI designer for Qt applications	<ul style="list-style-type: none"> ● Drag and Drop functionality ● Can include stylesheet to each user interface component ● Can convert to .py file easily 	5.9.1
pytest	For unit testing	<ul style="list-style-type: none"> ● Much faster because of less verbosity 	4.1.0
Pylint	For source code testing	<ul style="list-style-type: none"> ● Checking code length ● Checking if variable names are well-formed ● Checking if declared interfaces are truly implemented. 	2.2.2
GitHub	For version controlling the application	<ul style="list-style-type: none"> ● Easy and familiar version controlling software ● Easy to integrate with PyCharm 	Git 2.6.3

4.4 Graphical User Interfaces (GUI)

Medical procedures are always complex and critical to perform. Hence, the user interfaces play a major role in computerized medical processes. Well-designed user interfaces lay out all the functionalities in front of the user. This system provides the physician with a separate GUIs with several functionalities.

Login

Login interface is provided for the physician so that details of the physician who performed intubation on each time can be recorded.



The image shows a web browser window titled "Login". The page has a teal header with the text "INTUBATION PROCESS". Below the header is a light gray background. In the center, there is a white box with a teal border containing the following elements:

- A heading: "PLEASE ENTER THE LOGIN DETAILS"
- A horizontal line separator.
- A label "Username" followed by a text input field.
- A label "Password" followed by a text input field.
- A link: "Need a help?"
- A teal "Login" button.

At the bottom of the page, there is a footer that says "All Right Reserved".

Figure 4.1: Login Interface

Select Intubation Mode

After logging into the system, the physician can choose one of the two modes either manual or automatic. Manual mode can be selected if the physician wishes to insert the tube by himself with the help of guidance provided by the system. Automatic mode can be selected if this system is extended to send navigation details to a hardware device.

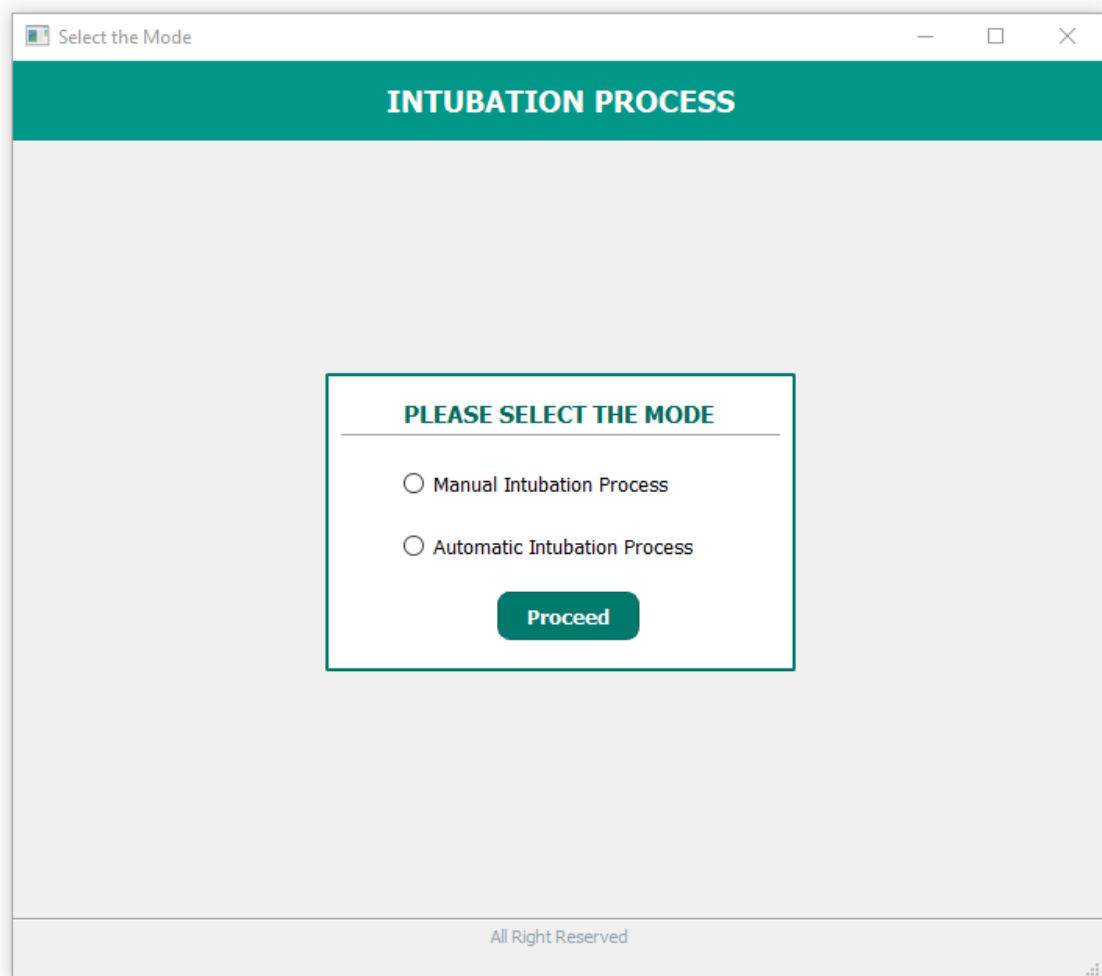


Figure 4.2: Mode selection interface

Manual Intubation Mode

In manual intubation mode, camera feedback is obtained, annotated with relevant navigation details and displayed in the graphics view so that the physician can move the tube according to the given guidance provided by the system.

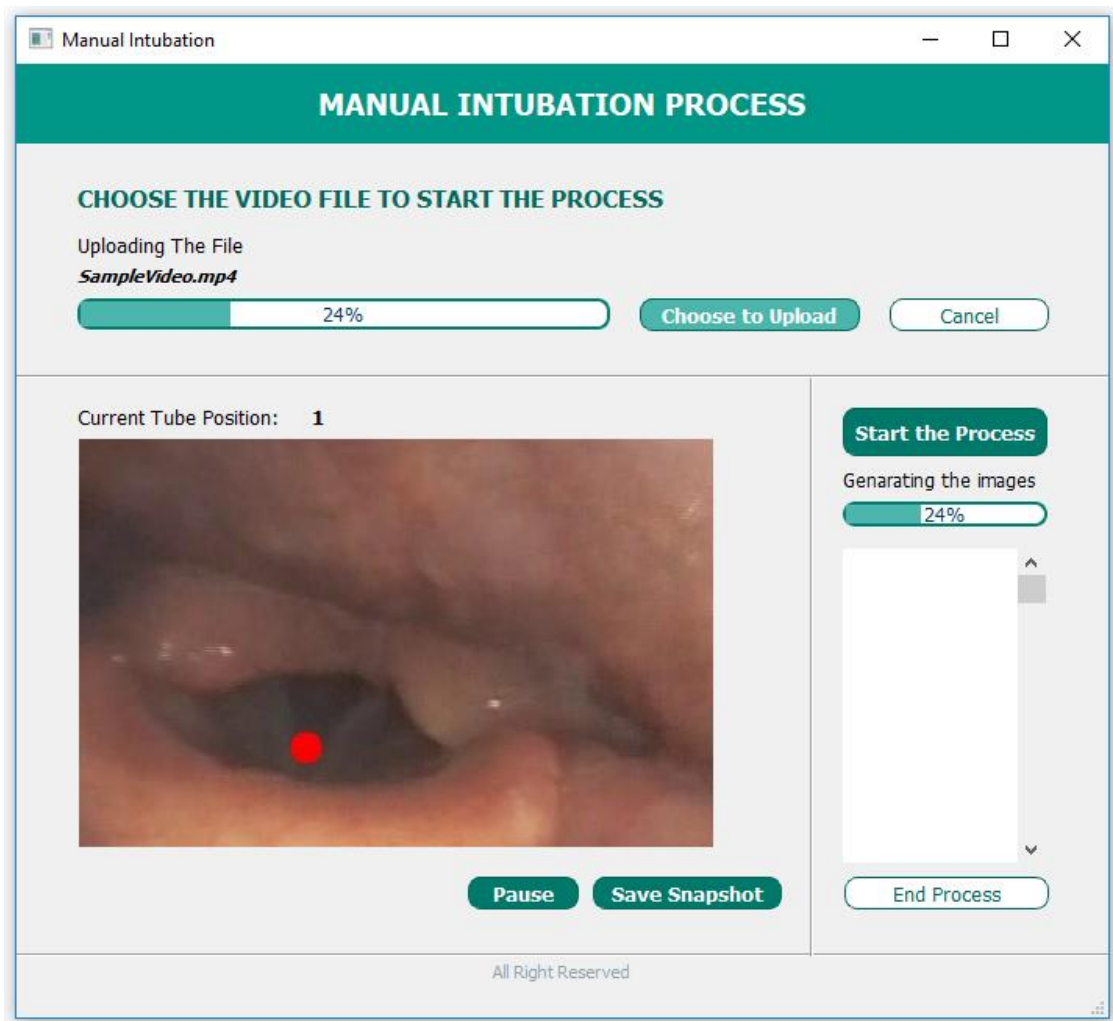


Figure 4.3: Manual Intubation

If he wants to be waiting in a particular point, then the pause button can be clicked and further examine the current location.

Further, he can save a snapshot of the current view to study later using save snapshot button.

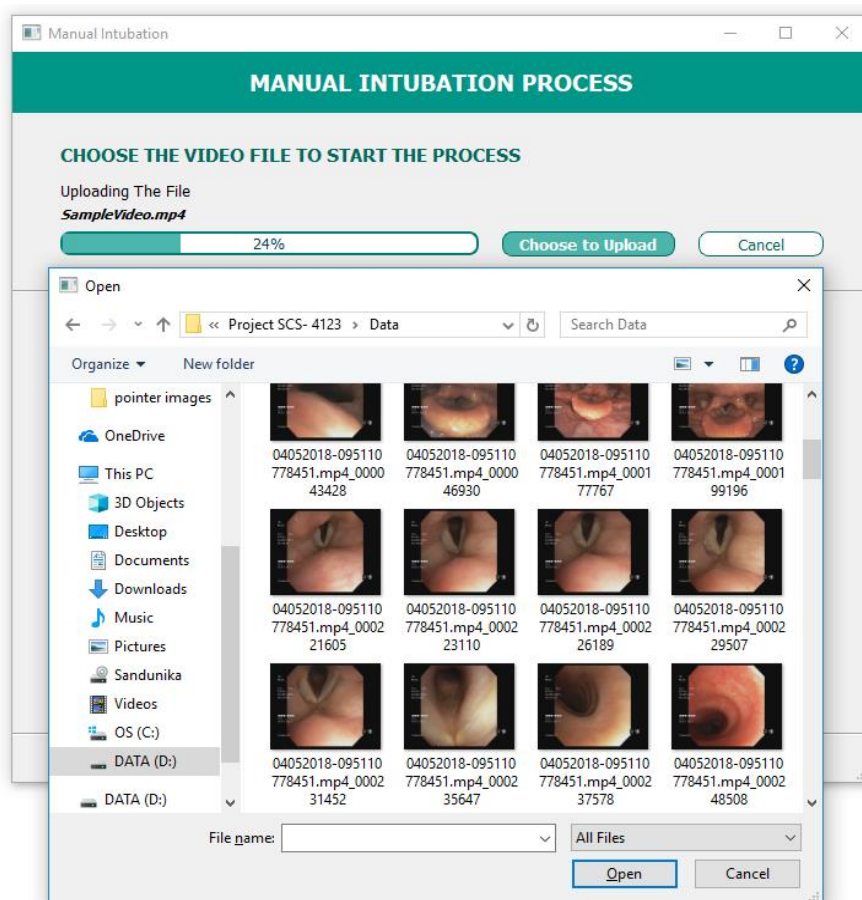
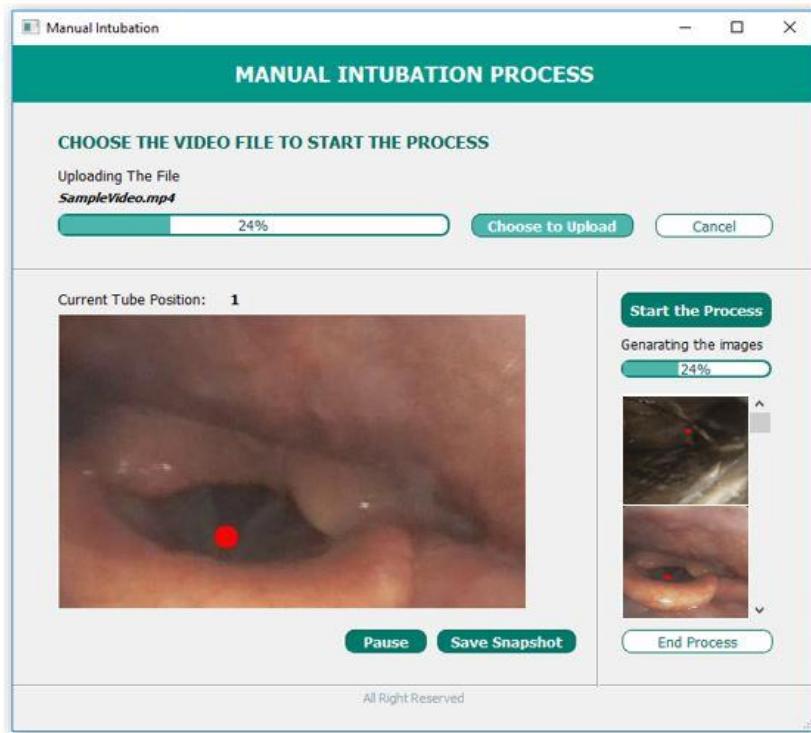


Figure 4.4: Save snapshot and Open file

Automatic Intubation Mode

Automatic mode is designed for the hardware device if this system is extended up to a hardware device and it provides almost the same functionality as in manual mode. Here, the physician can start the process and allow the system to insert the tube automatically while examining the procedure.

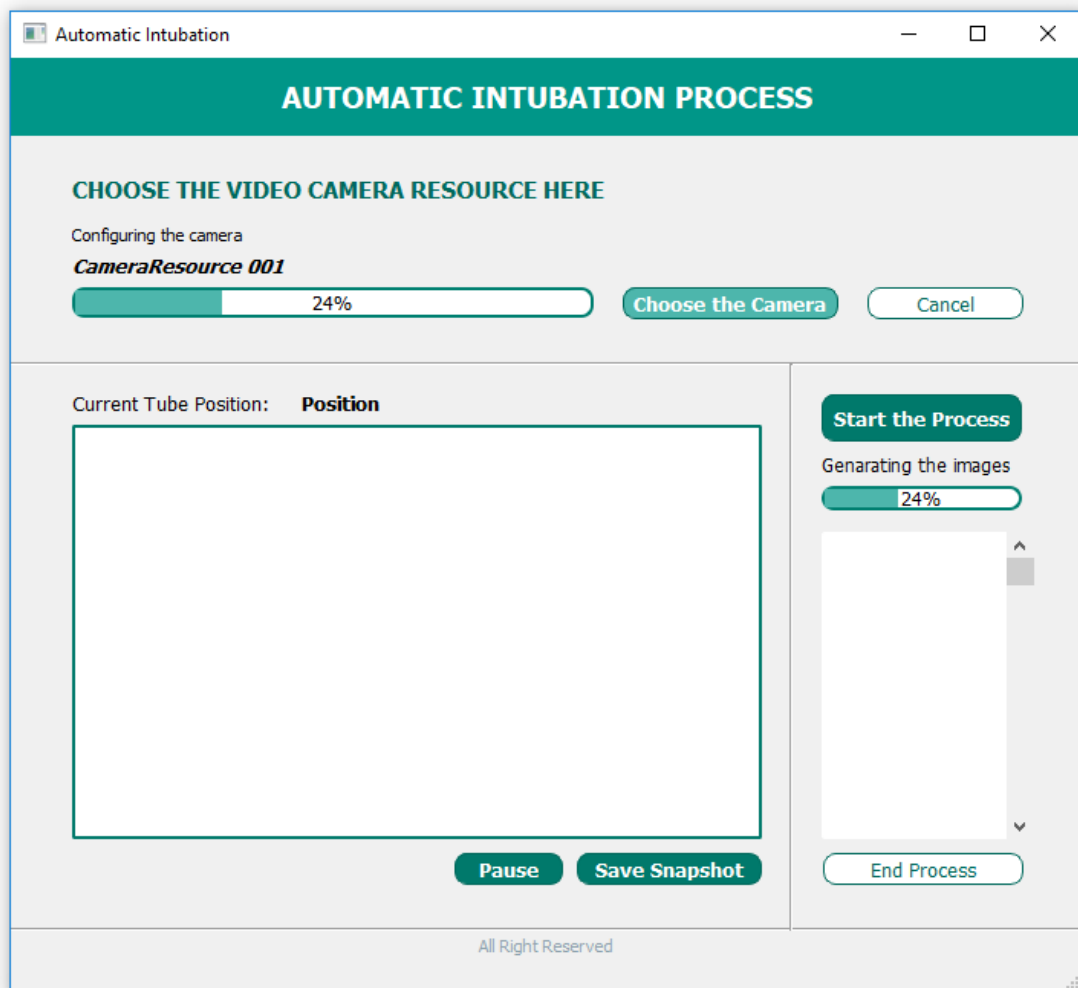


Figure 4.5: Automatic Intubation

4.5 Methodology

4.5.1 Analysis of the path from the nose to the carina

For the analysis of the path from nose to the carina, the path was divided into different regions. These regions are identified by markers which are anatomical locations. When identifying different locations, the criticality of the point was considered with the guidance of a physician.

Critical Locations and their difficulties

Tracheal Opening

- Tracheal opening separates windpipe and food pipe which is trachea and esophagus. Misjudging these two will lead the tube to enter into the food pipe.
- Hitting the walls of tracheal opening even with less pressure will damage the vocals.

Carina

- The tube should be stopped and placed 2 cm above the carina to perform the intubation.
- Passing the carina will send the tube towards bronchus and will lead for a lung failure if one lung intubation happens.

Nasal Opening

- Since the nasal opening is comparatively small a device which is capable of doing nasal intubation should pass through the nasal opening carefully.

Therefore the identified locations should include these special landmarks of the path. To identify these landmarks separately, the system must be able to classify them precisely. Therefore the set of landmarks should be selected in a way that the number of anatomical landmarks would increase the accuracy of the classifier. Since the exact number of anatomical locations were not known beforehand, a different number of locations and different regions of the path were analyzed. After a trial and error method, the best number of landmarks were obtained, so that images from one landmark can be

easily distinguished from another landmark. When identifying landmarks, feasibility to capture images due to distinguishability was also considered.

It was clear that increasing the number of points will not always increase the efficiency of the system. Parts of the respiratory path are hard to recognize due to many reasons. Not having enough light to obtain quality images is the main difficulty. Nearby points are almost similar in color and shape. Since the capturing frames from the video were done by hand, when there were very close points that look similar, capturing them was difficult, and they were classified into the same class instead of two separate classes. The validation accuracy was very low.

Having three or four points was not appropriate because predicting navigations between far away points did not give better accuracy in the overall system. Therefore it was needed to identify an average number of anatomical landmarks along the path that satisfies the above concerns.

After experimenting a different number of points, it was shown that ten anatomical landmarks are sufficient to identify the path. One problem arose here was the slowness of the classification process.

Following are the identified anatomical landmarks:



Figure 4.6: Anatomical Location 1



Figure 4.7: Anatomical Location 2



Figure 4.8: Anatomical Location 3



Figure 4.9: Anatomical Location 4

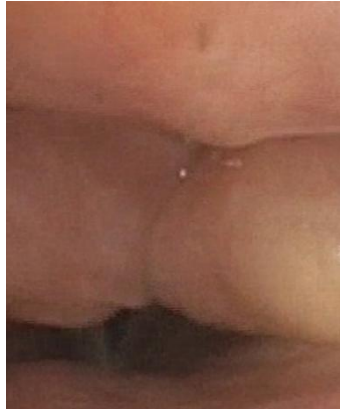


Figure 4.10: Anatomical Location 5



Figure 4.11: Anatomical Location 6



Figure 4.12: Anatomical Location 7



Figure 4.13: Anatomical Location 8



Figure 4.14: Anatomical Location 9



Figure 4.15: Anatomical Location 10

From these set of locations, images were captured and made sets of images for each location.

4.5.2 Image Capturing and Preprocessing

The first step of image preprocessing is important because noisy images may lead to erroneous results in the neural network. Before training the CNN, captured images are preprocessed using several image processing techniques and python libraries.

First, images are cropped to remove unwanted (noise) parts such as printed letters and borders. Then, they are resized to reduce the size to obtain a higher speed in processing. Then images are smoothed using Gaussian smoothing to reduce image noise and reduce detail in batch mode using python scripts. (Appendix B.4) Finally, a python library named Augmentor which is designed to aid the augmentation (Appendix B.5) and artificial generation of image data for machine learning tasks is used to image preprocessing. It is primarily a data augmentation tool, but will also incorporate basic image pre-processing functionality. Augmentor is used to increase the data set. In the image gathering process, a number of images that could collect is limited. The Augmentor provides different functionalities to make changes in the images and obtain multiple copies from them. The operations of the Augmentor used in this process are the rotation and zooming. The reason for doing rotations is that in the intubation process the camera may be inserted in different angles. Therefore, using rotate operations would not cause any damage to data set since it is a natural phenomenon. The zooming operations are used as some locations may be visible in variable distances. Some points in the path can be seen as close images and distant images from the zoom operator.

4.5.3 Convolutional neural network for classification

4.5.3.1 Justification for selecting a CNN over other techniques.

A Convolutional Neural Network is a network architecture for deep learning. It consists of three concepts which lead to the correct image analysis. First one is the Local receptive fields. In here, only a small region of input layer neurons connect to the neurons in hidden layer. These regions are called local receptive fields. Local receptive field is translated across an image to create a feature map from the input layer to the hidden layer neurons. Convolution can be used to implement this process.

The second one is the Shared weights and biases. In here weights and bias values are same for all hidden neurons in a given layer. All hidden neurons are detecting the same features such as an edge etc. on different regions of the image. This makes network tolerant to the translation of images. When considering SIFT features, they are invariant for scaling and slightly rotations but not for substantial rotations. Therefore the use of CNN may be advantageous as they are robust to imaging angles. This property really importance to the task in hand, because during the intubation the tube may be inserted in different angles and directions, depending on the technique employed by the person performing the procedure.

The third one is Activation and Pooling. In here, transformation to the output of each neuron is done by using activation functions. Then it can further transform the output of the activation step by applying a pooling step. Pooling reduces the dimensionality of the feature map by considering the output of small regions of neurons into a single output.

Among the three ways to use a CNN for image analysis, the used methodology in this project was Transfer Learning. It is based on the idea that the knowledge of one type of problem can be used to solve a similar problem.

Other learning algorithms or models can also be used for image classification. However, CNN has emerged as the model of choice for multiple reasons. These include the multiple uses of the convolution operator in image processing. The CNN architecture implicitly combines the benefits obtained by a standard neural network training with the convolution operation to classify images efficiently. Further, being a neural network, the CNN (and its variants) are also scalable for large datasets, which is often the case when images are to be classified.

Other Machine Learning algorithms like KNN, SVM, logistic regression, etc..., can also be used. But these algorithms learn less when comparing to CNN because transfer learning does not happen in Machine learning algorithms. But in deep learning, transfer learning happens, then it will learn more, and also reduces the error.

4.5.3.2 Transfer Learning

In this methodology, a pre-trained CNN is used which follows the concept of transfer learning. It is based on the idea that the knowledge of one type of problem can be used to solve a similar problem. The usage of transfer learning facilitates the classification in many aspects. Pre trained models use in transfer learning provide high accuracy as they are fine-tuned models for millions of image classifications. These models composed of feature extraction procedure which do not need to implement from scratch. Due to this feature extraction ability, the system can be trained using less training samples.

CNN Models for Transfer Learning

Different CNN models were examined and tested to find the best pre-trained model for the classification.

Table 4.2: Pre-trained models for classification

Network	Accuracy rate
AlexNet	89.6 %
Inception v1	89.6 %
Inception v2	91.8 %
Inception v3	93.9 %
ResNet v1 50	92.2 %
VGG 16	89.8 %
MobileNetV1	92.5 %

From the above models, the models which have high accuracy are tested for the data set of anatomical landmarks and the accuracies of them compared. Then the computation time for the prediction of images from each model are compared, and the best model is used in the implementation. Almost all models that were used gave high accuracy. From these models, MobileNetV2 model showed high accuracy and used less computation time. Therefore MobileNetV2 is used in the training process.

MobileNetV2 model over other CNN models.

From the above models, VGG models are specialized for facial recognition. VGG models are more expensive in computation time. They contain multiple nonlinear layers inside its network. Although that leads to learning more complex features, it significantly increases the computation time. As this system classify images real-time, it needs a fast computation mechanism. Therefore VGG models were not used in this system as a neural network.

Considering the above accuracy rates, MobileNetV2 gives a large amount of accuracy and less error rate. In MobileNetV2 model the depth of neurons is less that fires during classification. The width of convolution layers and the number of convolutional layers are reduced in this model compared to AlexNet and ResNet model. Therefore computation time is reduced in the MobileNet model.

4.5.3.3 MobileNetV2 Model

MobileNet is a lightweight neural network architecture which is designed for mobile devices. MobileNet which has pre-trained weights trained on ImageNet has been developed by Google. Although MobileNet is mostly used for object detection, it is not just limited to object detection. Tasks of the MobileNet spans a variety of tasks such as facial attributes, landmark recognition, fine-grain classification.

MobileNet models are based on depth wise separable convolutions which consist of factorized convolutions and 1x1 pointwise convolutions. In MobileNets the existence of depthwise convolution provides a filter for each input channel. Then the 1x1 pointwise convolution provides a means to combine the outputs and the depthwise convolution. In a scenario of standard convolution filtering and combining inputs to outputs occur in one step. In MobileNetV2 the depthwise separable convolution provides a methodology to splits this process into two layers. One layer is for filtering, and another separate layer is for combining. This process is the reason for the effect of reduced computation and model size. The activation function used in this model is the ReLU6.

$$\text{ReLU6 function : } f(x) = x^+ = \min(\max(x, 0), 6)$$

Where x is the input to a neuron. Relu is half rectified. It is mostly used in mobile machine learning cases. It is used in google MobileNet architecture. The issues in ReLU is, neglecting the negative inputs generated by images. To overcome this, the model has used high bias values. In this model batch normalization is used as in standard convolution layers so that it would normalize the input layers by adjusting and scaling the activations.

Standard Convolutional Layer

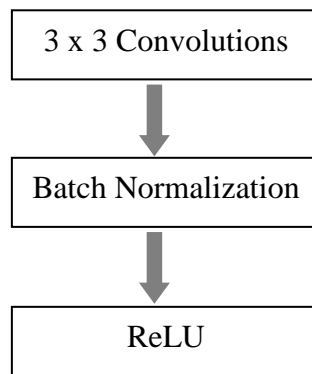


Figure 4.16: Standard Convolution layer

Depthwise separable layer

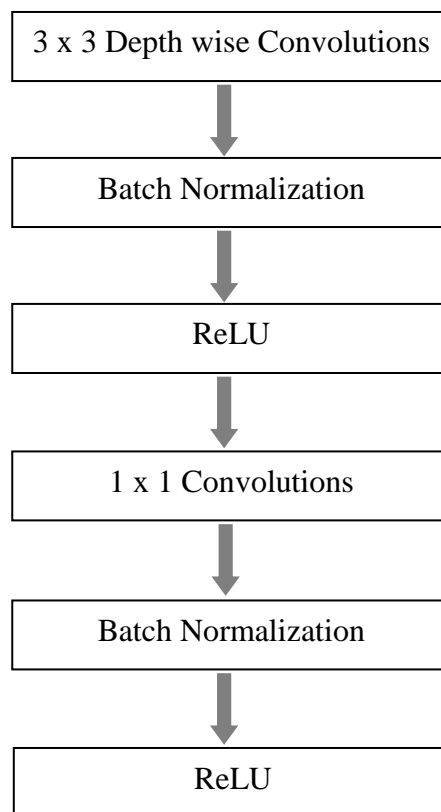


Figure 4.17: Depthwise separable layer

Using the MobileNet model the data set was trained and the parameters used in the training process is as follows.

4.5.3.4 Training Technicalities

Training network parameters

Learning rate = 0.01

Testing percentage = 20 %

Validation percentage = 20%

Training batch size = 100

Training steps = 4000

Image parameters in the model

Flip left right = True

Random crop = False

Random scale = 20%

Random brightness = 20%

When the flip left-right parameter set to true, the images used in training process will perform different rotations. Random crop set to false, because the cropping will lead to misclassification in some anatomical locations. Anatomical location 3 is a distant view of anatomical location 4. If the cropping occurs in anatomical location 3, it may show a similar view of anatomical location 4. Therefore the random crop is set to false. Training accuracy growth with the number of training steps

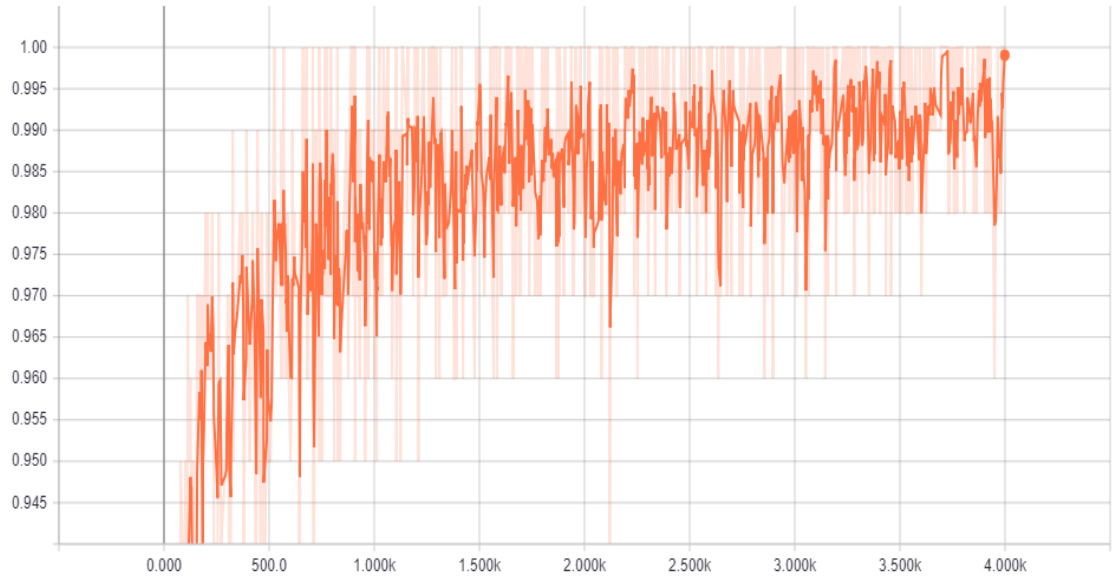


Figure 4.18: Smoothed graph training accuracy vs number of steps

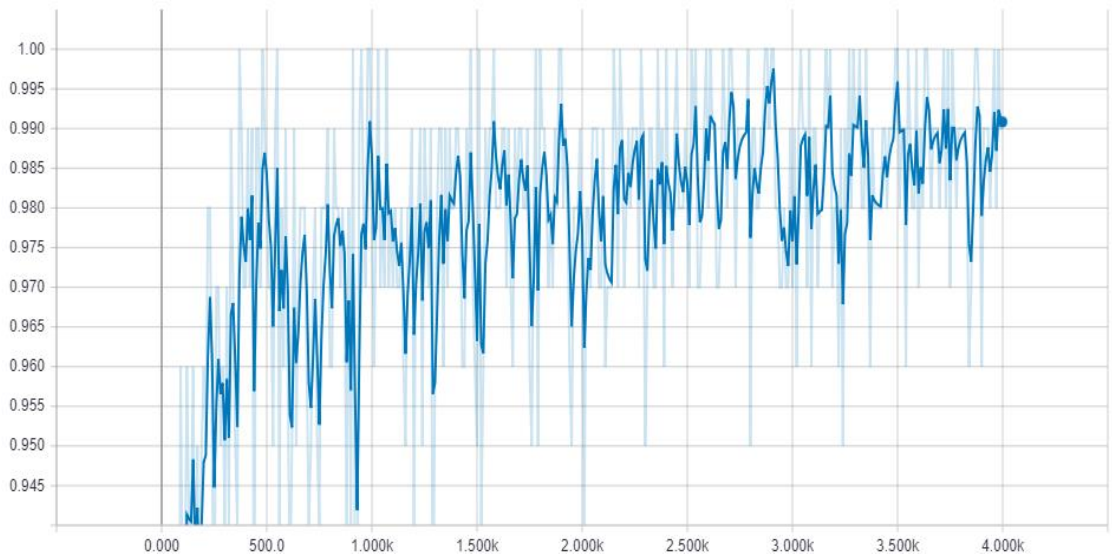


Figure 4.19: Smoothed graph validation accuracy vs number of steps

4.5.4 Prediction of locations

The trained model is used in the prediction process. Prediction component in this system composed of three phases, reading and processing phase, predicting phase, and visualizing phase. After the training process, the outputs of training which is the output graph is used for predicting the results.

4.5.4.1 Reading and processing phase

The functionality of this phase is to capture the frames from the video feed or a video file. The video is allowed to read from video file, or a camera input. Python OpenCV libraries were used to obtain frames from videos. The images are subjected to preprocess operations like cropping and resizing in order to read tensors from the image.

4.5.4.2 Predicting phase

After the image is fed into the prediction model, the model with the use of its softmax layers generates different percentage values for the occurrence of each class. From those percentages, only five maximum values are given as outputs from the softmax layer. Out of that five values, maximum percentage values are considered, and if the maximum value is less than 85%, then that prediction is neglected. As the prediction is performing for a video stream the sequence of prediction classes (anatomical locations) must be continuous or nearly continuous. If the prediction for a frame is class 3, then the next prediction can be same as class 3, class 4, class 5 (when the tube is moving forward in the intubation path) or they can be class 2, class 1 (when the tube is moving backward in the intubation path). This phenomenon is also taken as an advantage to increase the accuracy of prediction. If the model predicted a class which far away from the current location, then that prediction also can be neglected.

4.5.4.3 Visualization phase

Visualization is the final task in the system which shows output to the user which consists of three main components. These components are Action Analysis, Navigation Pointer Generation, and Displaying Figures.

Action analysis

In the action analysis phase, there are a set of predefined actions that need to be performed when moving the tube around each anatomical location. The set of actions are as follows.

Table 4.3: Set of actions performed at each location

Anatomical Location	Action
1	Move forward
2	Move forward with a downward thrust
3	Move forward
4	Wait for few seconds till location 4 become stable
5	Stop, Move 1cm back
6	Move forward
7	Move forward with a downward thrust
8	Move center pixel toward the black hole
9	Move forward with extreme slow
10	Stop move 2cm back

The actions are determined by the opinions of experts in this field. In the locations 2 and location 7 the reason for giving downward thrust while moving forward is due to the curves in the nasal cavity and trachea. In location 4, the tracheal opening is subjected to rounds of frequent opens and closes due to the high sensitivity in that region. Therefore the tube should not move forward directly and should be waiting until the open state of trachea become stable. After that, the tube can move forward. The anatomical location 5 is the closed state of tracheal opening. In that case, the tube should stop moving immediately and should move backward around 1cm otherwise damages may cause for the epiglottis. The epiglottis is the thin muscular valve which controls the opening of food pipe and windpipe. It can open trachea while closing food pipe and open food pipe

while closing trachea. At location 8 the instruction given was to move the tube towards the black hole so that it would tend to reduce the touching of tube walls. Therefore it has less tendency for damaging and bleeding. Since location 9 is the close region where the tube should be stopped to perform the intubation, it is instructed to move the tube forward with extremely slow speed so that tube will not go beyond that critical region. When the location 10 is set as the current tube location that means the tube has gone more than the required depth and hence instructed to move 2cm backward. All these instructions are provided graphically for the user by annotating them in the video.

Navigation Pointer Generation

In order to guide a physician apart from giving verbal instructions, it is necessary to depict a pointer so that physician can move the tube towards the navigation of that pointer. Further, an automated device which is a future work for of this approach can be moved with the aid of that pointer. To mark the pointer with correct placement, it is necessary to identify the correct places where the endotracheal tube should be pointed. Series of images of such places were identified for each anatomical location. To obtain the features of those images SURF local feature detector is used. Before obtaining features, all images are converted to grayscale images. Python OpenCV library contains surf features detector functions which can detect the key point of the images. Key points are interest points of an image. Though the image is subjected to rotations, shrinks, expands, translations or distortions the same set of key points can be found from the image. Feature descriptors were computed using those key points. Feature descriptors are vectors which encode interest information of images. The task is to match these features with the frames in the ongoing video stream.

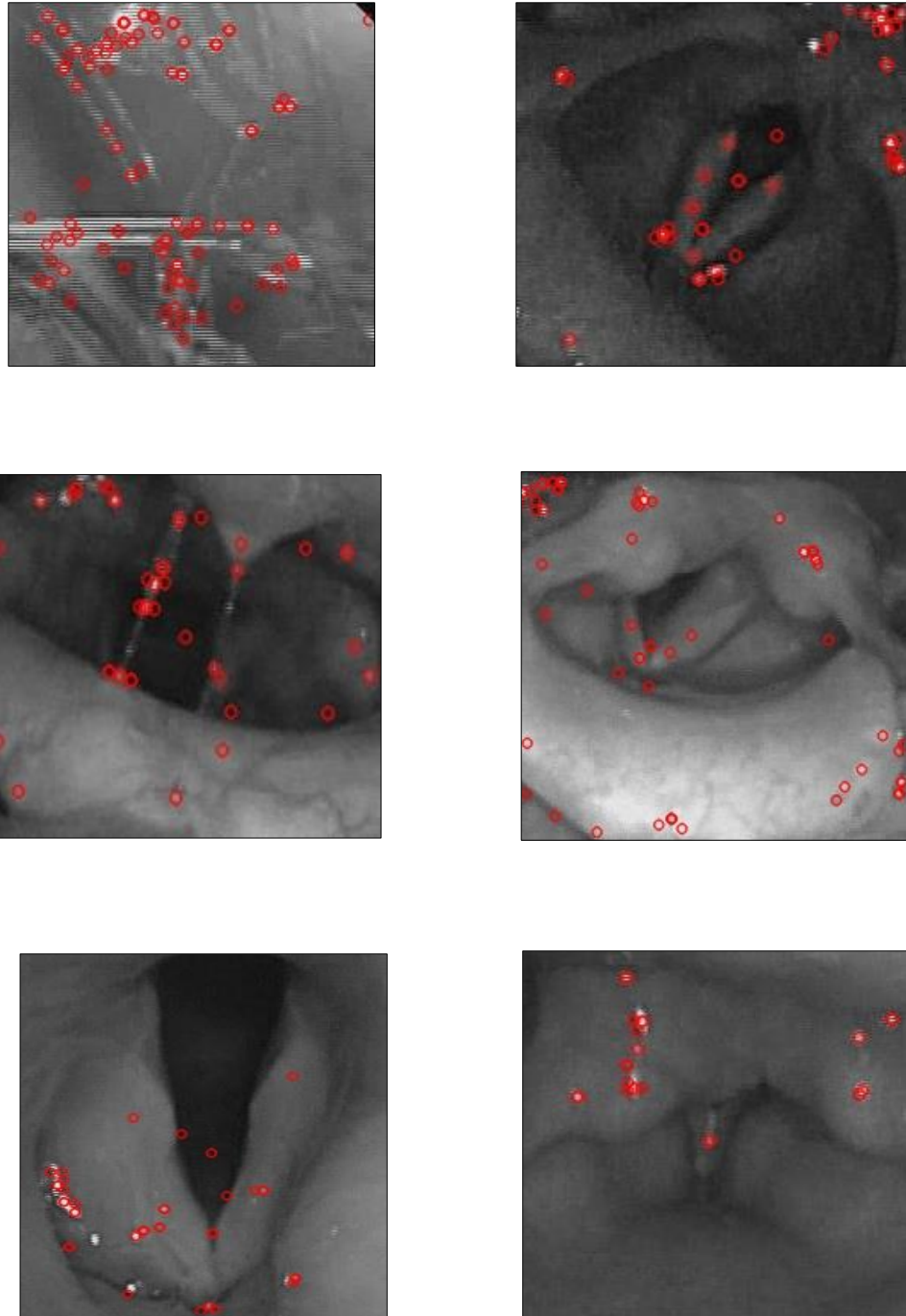


Figure 4.20: Sample set of key points generated by SURF key point detectors

After obtaining features of both image sets, they need to be matched, and the matches are computed using FLANN based matcher. FLANN stands for “Fast Library for Approximate Nearest Neighbors” and it consists of algorithms optimized for computing matches in features of images with high speed. The reason for choosing the FLANN

matcher is that FLANN is capable of performing fastly for large data sets when compared with the BFMatcher. BFMatcher stands for Brute Force Matcher which computes matchings between features of images by brute force methodology via distance calculations. From all the matches only good matches are considered, and if the count of good matches is greater than 25, then it is considered that the two images are matching. The count was set to 25 by doing several trials and finding the best possible count value. This feature matching is implemented using python OpenCV libraries (Appendix B.2)

After matching images are found, although the images are in different angles, homographic operations are performed so that matching points in the image set can be displayed in the video frame images. Then a function in OpenCV called perspective transform is used to obtain the pixel coordinates of the quadrilateral which bound the matching features points obtain from homography. Homography matching functionalities are provided in python OpenCV libraries(Appendix B.3)

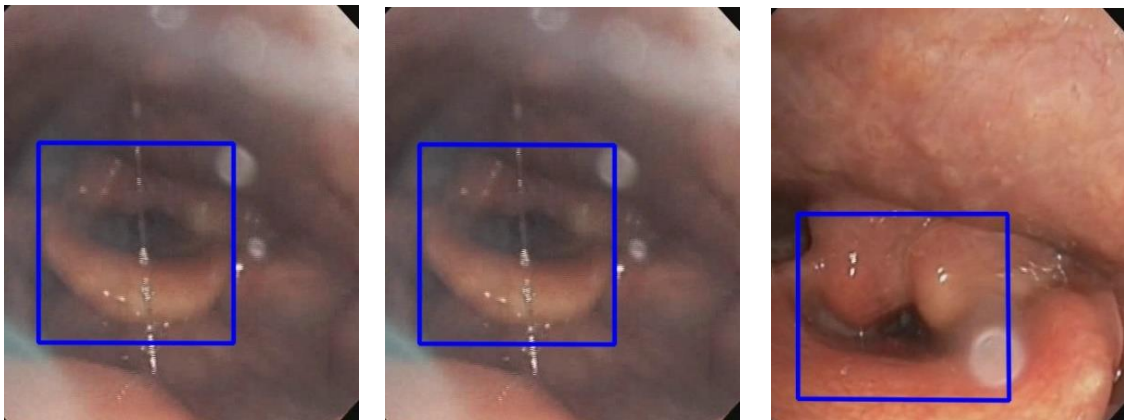


Figure 4.21: Sample set of quadrilateral projection in the images of videos.

After obtaining the quadrilateral coordinates, the central points of these quadrilaterals are calculated, and these points are depicted in the frame of the image as follows by a red dot. This feature matching task tends to slow down the system further due to calculations operate in the OpenCV functions. The complete process of navigation pointer mentioned above is a series of pipeline tasks which need to be performed sequentially and does not support simultaneous processing.

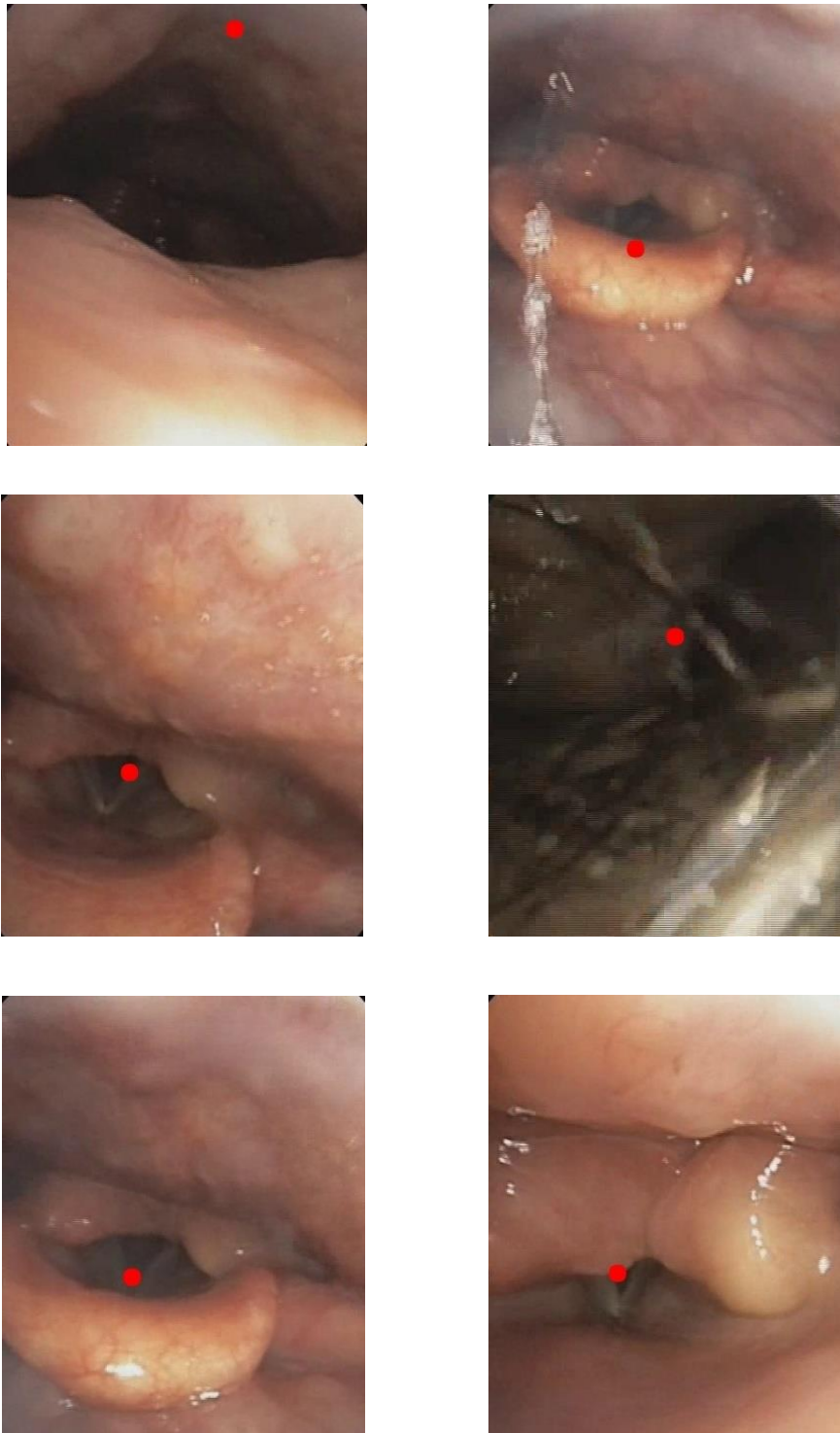


Figure 4.22: Sample set of pointer projection on the images of videos.

Displaying Figures

The instruction given at each anatomical location should be annotated into the video with the identified anatomical location. The reason for annotating the identified anatomical location is to user training purposes. These annotations are made to each frame in the video stream which has already perform the pointer projection.

4.5.4.4 Drawbacks in the prediction

The main drawback in the prediction phase is that the slowness in the prediction. As the system has to obtain every frame from the video and predicts each frame, it needs much computational power. 20-30 frames have to be predicted within one second, and before the prediction, the images needed to be captured, preprocessed and also final visualization needed to be done. These tasks cause for lagging in the prediction phase. Therefore optimizations were carried out to improve efficiency.

4.5.4.5 Optimization techniques

Among the several tasks in this system, the predicting of a frame which is obtained from the video is the part that shows a slowness. To overcome that issue, different approaches were taken. Multitasking is the main approach that was experimented. To perform multitasking, the main tasks of the system were identified first.

The main tasks of the systems are as follows

- Reading and processing frames from the video
- Predicting the frames obtained from the video
- Visualizing the output

4.5.4.5.1 Multithreading

The three subtasks mentioned above was assigned to three threads apart from the main thread. The subtasks work according to a pipeline process as follows.

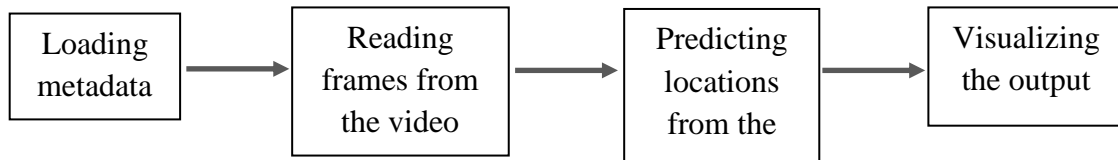


Figure 4.23: Pipeline process of sub tasks

Although loading metadata is a one time process other tasks are running throughout the program execution time. The output from reading data is obtained by the predicting task, and that output is then obtained by the visualizer. Since the output from one subtask is essential for the processing of other subtasks, in a multi-threading approach if one thread fails to produce an output other thread, have to wait. Therefore a queue is used in the process so that first thread will enqueue data to a queue, and others will dequeue if the queue is not empty.

The result of Multithreading Approach

Slowness was increased due to the threading approach. Although threading lead to run the system without errors, slowness happens due to the issues in python threading. As our system is mainly based on python programming language threading causes troubles in the system.

GIL effect in python (Global Interpreter Lock)

The Python Global Interpreter Lock or GIL is simply a mutex (or a lock) that allows only one thread to hold the control of the Python interpreter. Python interpreter is the program which read and execute the python code. Python threads lock its interpreter when it executes, and hence other threads cannot execute. This means that only one thread can be in a state of execution at any point in time. The impact of the GIL is not visible in single-threaded programs, but it can be a performance bottleneck in CPU-bound and multi-threaded code. Since our system is CPU-bound multi-threaded program this GIL effect cause to lose its speed of execution.

4.5.4.5.2 Multiprocessing

The methodology to solve this issue is to have different interpreters for each subtask. Python has a multitasking strategy same as multi-threading which is called multiprocessing. Multiprocessing keeps a different interpreter for each process. And also each process gets its own address space. This allows a program to leverage multiple processors on a given machine. Therefore above three subtasks were assigned to three processes and yet the queue is used since still the pipeline mechanism prevails.

Results of multiprocessing

A bit speed in the processing can be achieved, but that is not still enough. Before multiprocessing and multithreading, about two frames were predicted per second. After multiprocessing, it improved up to about three frames per second. This enhancement does not support in the overall system performance. The reason for this lagging is the bottleneck caused by the predicting task. Reading frames from the video is the task that happens at a great speed and visualizing the frames also not causing any lagging. But the predicting process takes 0.25-0.3 seconds to predict a single frame. Hence this process blocks the whole system flow, and multiprocessing is not helpful in obtaining the speed in processing. A mechanism should be made to speed up the predicting mechanism.

These optimization techniques still unable to give the required efficiency, and the system efficiency is the future objective of the project.

Chapter 5 - Result and Analysis

The goal of the project is to produce guidance for the intubation process hence, the path from nose to carina should be analyzed. Identifying the structure of this path by a computer is necessary. Hence images of the path should be captured and preprocessed to use in the analysis. Different approaches were carried out to find a classification mechanism and to find the optimal solution. Grayscale intensity histograms were obtained and analyzed to find a pattern. Then local binary pattern histograms were obtained and analyzed. Then SIFT and SURF feature matching mechanisms were examined. Since each of them had drawbacks, a Convolution Neural Network was used as the optimal solution for the classification task.

5.3 Grayscale Intensity Histogram Analysis

The first approach was Grayscale Intensity Histogram Analysis where it generates intensity histogram for each anatomical landmark to differentiate among them. Following are the observations of grayscale intensity histogram analysis.

Observation 1: Images in the same class (anatomical location) show dissimilarities in the histogram. (Figure 5.1)

Observation 2: Images in the different classes tend to show similar histograms. (Figure 5.2 and Figure 5.3)

The justification for observation 1

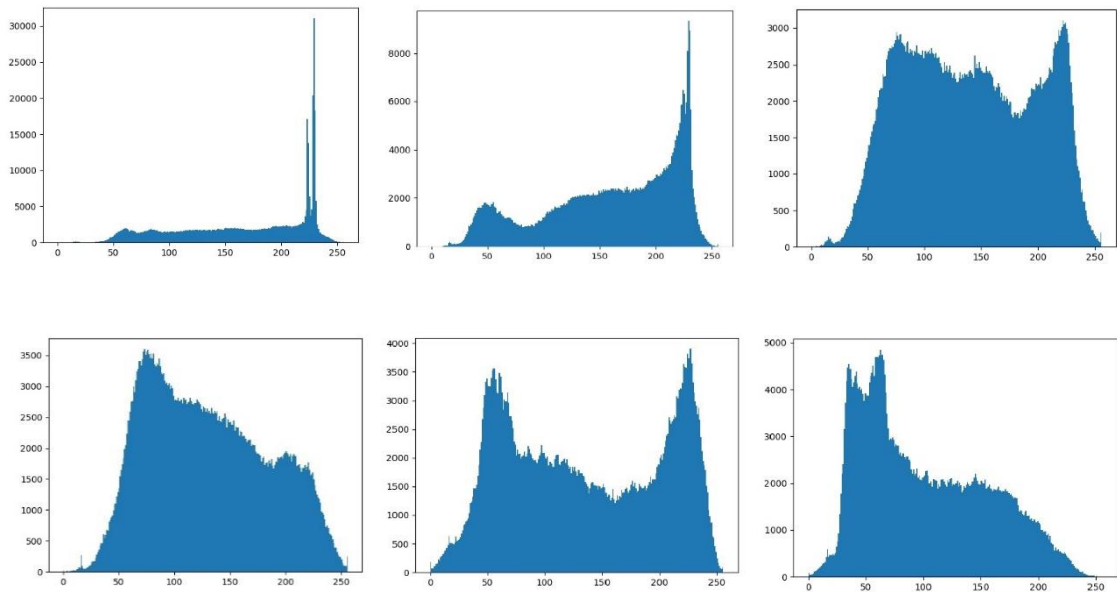


Figure 5.1: The sample set of intensity histograms of the images at anatomical location 1

The justification for observation 2

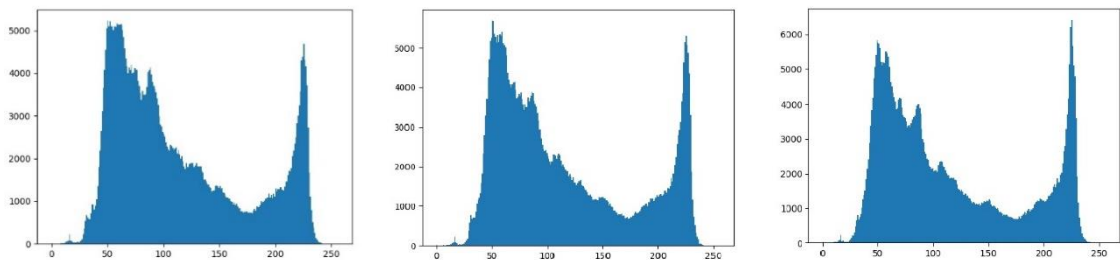


Figure 5.2: The sample set of intensity histograms of the images at anatomical location 4

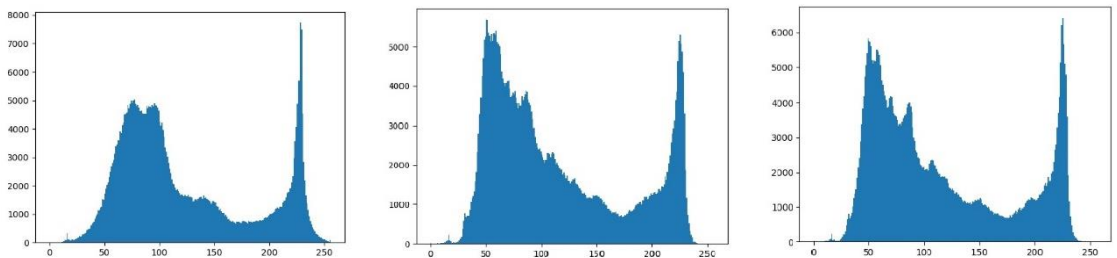


Figure 5.3: The sample set of intensity histograms of the images at anatomical location 1

Therefore, the study concludes that grayscale Intensity Histogram analysis method cannot be used to categorize the images. Hence it was rejected.

5.4 Grayscale Local Binary Pattern(LBP) Histogram Analysis

The second approach was Grayscale Local Binary Patterns Histogram (LBP) Analysis which is combined with the histogram of gradients to enhance the detection process further. Following are the observations obtained.

Observation 1: Images in the same category shows similar histograms (Figure 5.4)

Observation 2: Images of different categories also shows almost similar histograms (figure 5.5 and figure 5.6)

The justification for observation 1

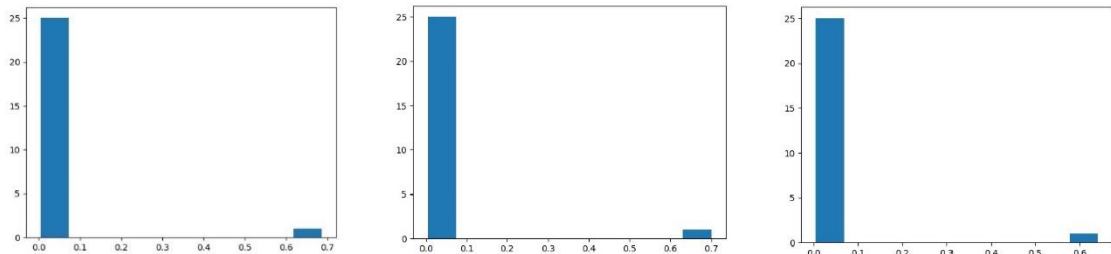


Figure 5.4: The sample set of LBP histograms of the images at anatomical location 2

The justification for observation 2

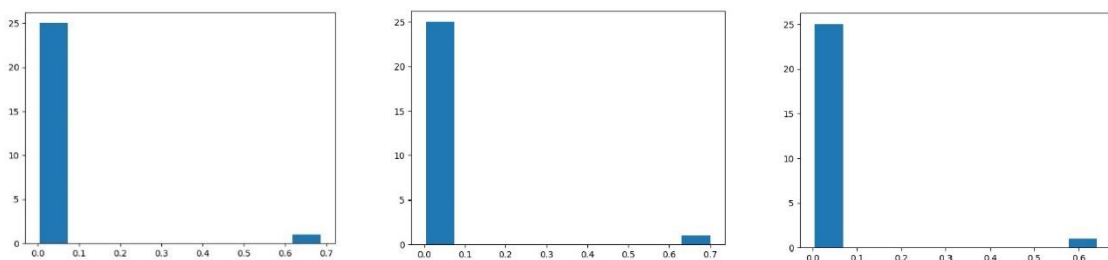


Figure 5.5: The sample set of LBP histograms of the images at anatomical location 2

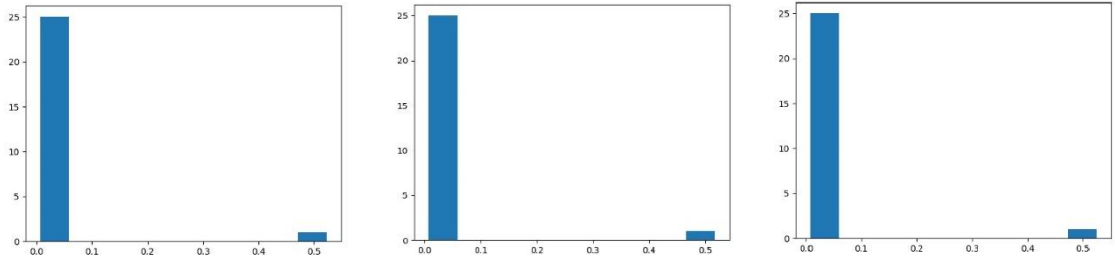


Figure 5.6: The sample set of LBP histograms of the images at anatomical location 5

Therefore, LBP was rejected concluding that it cannot categorize images accurately.

5.5 Convolution Neural Network Approach (CNN)

The third approach was the CNN. CNN provides a mechanism to classify the anatomical landmarks into different classes. For different models, different accuracies were obtained, and they are as follows.

Table 5.1: Accuracies in pretrained models

Network	Accuracy Rate
Inception v3	97.7 %
MobileNetV2	97.8 %
Resnet v1 50	95.2 %
Inception v1	92.7 %

Though Inception v1 and MobileNetV2 provide the highest accuracy, the efficiency of the system reduces when using Inception v1 model. Therefore, MobileNet model performs well in the classification process.

Chapter 6 - Testing and Evaluation

6.1 Test items

- **Preprocessing Images**

The preprocessing mechanism followed were cropping, resizing, and Gaussian smoothing. Python scripts were used for these operations in batch processing mode. Since all images are in the same size, the cropping applied similarly to all images. Hence cropped images were tested to check the loss of necessary image detail during the cropping. The images were obtained as random samples from all anatomical locations and tested. Gaussian smoothing was applied for only selected images so that images will not tend to blur unnecessarily. After all the image preprocessing operations, the resulting images were tested manually.

- **Neural Network Training**

The neural network was tested with different image datasets with different dimensionalities for accuracy and efficiency. Parameters in the pre-trained model were changed accordingly to get the optimal output from the training model. Different pre-trained models were tested and evaluated to obtain the optimal model for the purpose as well.

- **Predictions generated by the trained model**

The predictions were the output obtained from the trained convolutional neural network. The predictions were tested automatically by the model itself during the training process. Apart from that different image frames were run in the prediction component manually by getting the percentage values output from the softmax layer. Then their percentages were examined. Around 100 images were obtained randomly from different anatomical locations which were not used for the training process. Out of 100, 89 images were identified as correctly classified images since they gave above 90% in the softmax output layer. This testing result was considered to change the accepted softmax layer accuracy

level to 85% so that 95 images could be classified correctly out of the 100 images previously tested. Three series of these image sets were tested, and the average accuracy in these tests was approximately 93%.

- **Visualizing navigations**

Navigation visualizing is separated into different sections, and the testing was performed for different sections, and their integration tested. The different sections are as follows.

Pointer generation

The pointer is shown in the video so that a device which is a future work can lead to moving the device head to wards the pointer in the video. The pointer generation was tested by examining frames generated during the video. When the inaccurate pointer generations are visibly high, pointer generation components were refactored to obtain the optimal pointer generation. In this test section, it was clarified that pointer stability is in a very weak state.



Figure 6.1: Incorrect pointer prediction



Figure 6.2: Correct pointer prediction

Verbal navigation details

Depend on the current anatomical location in the video; the video is annotated with navigation details. This annotation process was tested in two approaches. They are forward tube moving and backward tube moving. There are different navigation detail sets for different approaches, and the results were 100% accurate. Since these verbal navigation details depend on the average prediction accuracy, verbal navigation accuracy is almost stable.

Integrating navigating details and pointer with video

Navigation details and the pointer coordinates should be synchronized correctly with the corresponding video frame. Otherwise, the overall output of the system fails. This was tested and the no issues were encountered during the integration process of these components. This process is further tested with the expert ideas in the evaluation process to test the accuracy of all navigation detail and the pointer with the current video

- **Sending operation details as an input signal to the hardware**

This process was not tested since it is not feasible to provide signals for a device which was not implemented. And such an approach will increase the complexity in the current system.

6.2 Test approach

Unit testing

Individual functionalities (test items) were tested and discussed the issues founded in them. Then necessary changes were made, and issues were resolved before integration. The neural network was trained with every video gathered and tested the label given for every anatomical landmark by the trained model before generating navigation details.

Integration testing

Each tested components which had been refined in the unit tests were integrated one by one and tested the dependencies between each component so that integrated components will function properly. After training the model with considerable accuracy, generating navigation component was integrated and then tested to verify that the system still classifies images with the same accuracy and generated navigations are accurate. User interface integration with the prediction models was a major aspect of the integration process and this integration was tested for responsiveness of user interfaces with the output in visualizing subcomponent in major prediction component.

System testing

After integrating all the system components and user interfaces, system testing was performed and checked whether the system fulfills the functional requirements intended by the requirements.

Regression testing

Performed regression testing to tests the issues in the system that will cause due to bug fixes and changes such as user interface changes, and new trained data in order to ensure that system will not be prone to bugs due to those changes.

Alpha testing

Acceptance testing was performed with the project advisor by giving him a full demonstration of the approach the project took, and his feedback was taken into account to improve the accuracy of navigation details generation.

Performance testing

The performance was not tested thoroughly as this project was carried out as a proof of concept. If this system is deployed in the client environment, performance testing must be done to verify that the system can efficiently process real-time data.

Image preprocessing results was tested because incorrectly preprocessed images may lead to erroneous results in the neural network. Therefore, image preprocessing

programming codes were tested and each image was tested to maintain the reliability of the system.

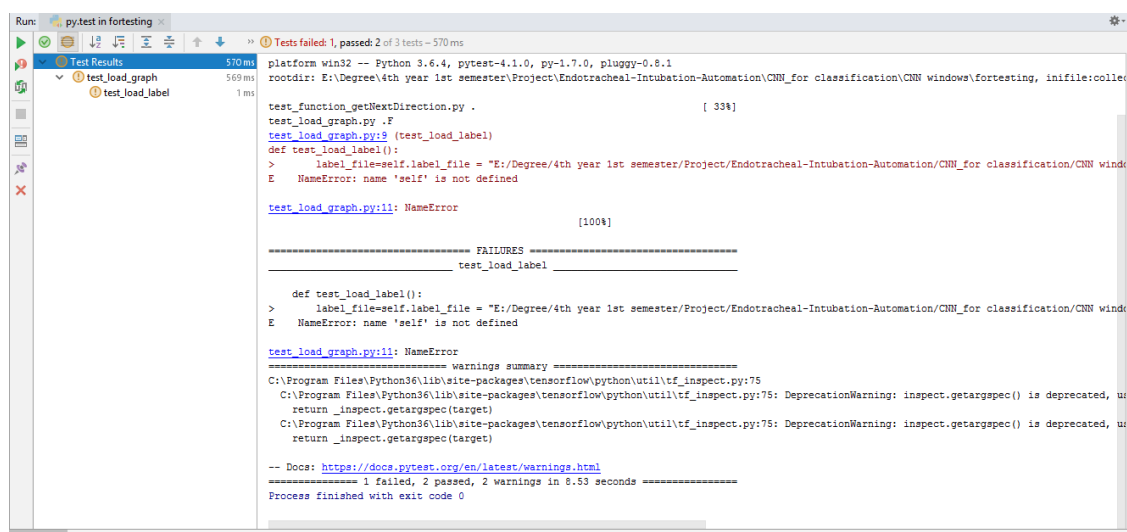
Neural network training was visualized using the TensorBoard and tests in the training process were examined by using the trained network for predictions. These tests were carried in batch mode because training the network takes a considerable time. Sending operations details as an input signal to hardware device should be tested in a specific manner as there is no hardware developed yet.

6.3 Test tools

6.3.1 pytest

All the unit tests in the python application were intended to test using pytest. The pytest was used over other test tools like "nose" and "unittest" due to different capabilities of pytest like fast testing mechanism, highly customizable, simple scripting mechanism and less verbosity. It gives much information on the failure. Pytest can be integrated with PyCharm and test each unit. Python scripts were written to perform the testings. (Appendix C.1)

Test Results



```
Run: py.test in fortesting
Test Results 570 ms
  test_load_graph 569 ms
  test_load_label 1 ms
  Tests failed: 1, passed: 2 of 3 tests - 570 ms

platform win32 -- Python 3.6.4, pytest-4.1.0, py-1.7.0, pluggy-0.8.1
rootdir: E:\Degree\4th year 1st semester\Project\Endotracheal-Intubation-Automation\CNN_for classification\CNN windows\fortesting, infile:colle

test_function_getNextDirection.py . [ 338]
test_load_graph.py .F
test_load_graph.py:9 (test_load_label)
def test_load_label():
> label_file=self.label_file = "E:\Degree\4th year 1st semester\Project\Endotracheal-Intubation-Automation\CNN_for classification\CNN windo
E NameError: name 'self' is not defined

test_load_graph.py:11: NameError [100%]

===== FAILURES =====
_____ test_load_label _____

def test_load_label():
> label_file=self.label_file = "E:\Degree\4th year 1st semester\Project\Endotracheal-Intubation-Automation\CNN_for classification\CNN windo
E NameError: name 'self' is not defined

test_load_graph.py:11: NameError

===== Warnings summary =====
C:\Program Files\Python36\lib\site-packages\tensorflow\python\util\tf_inspect.py:75
C:\Program Files\Python36\lib\site-packages\tensorflow\python\util\tf_inspect.py:75: DeprecationWarning: inspect.getargspec() is deprecated, use
return _inspect.getargspec(target)
C:\Program Files\Python36\lib\site-packages\tensorflow\python\util\tf_inspect.py:75: DeprecationWarning: inspect.getargspec() is deprecated, use
return _inspect.getargspec(target)

-- Docs: https://docs.pytest.org/en/latest/warnings.html
===== 1 failed, 2 passed, 2 warnings in 0.53 seconds =====
Process finished with exit code 0
```

Figure 6.3: pytest test results

From these test results, each unit can be tested and rework the component before integrating. The system could obtain 100% test cases pass rate after reworking.

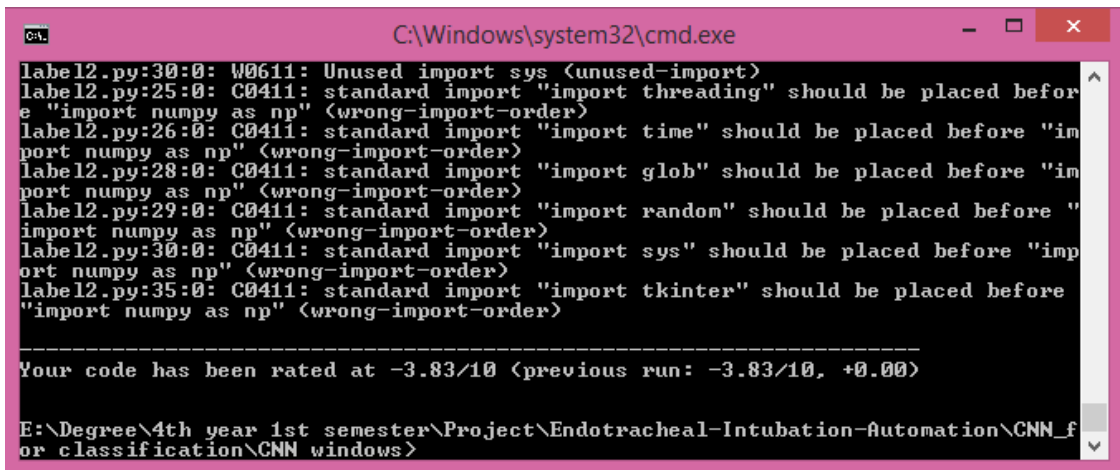
6.3.2 Pylint

The source code was tested using Pylint. Pylint provides different functionalities to maintain a good code structure and code quality. The features that Pylint provide are, coding standards, error detection, refactoring, IDE integration, etc. Since the project was developed using PyCharm, Pylint was integrated with PyCharm. The source code can be tested using Pylint itself with the command line interface. It gives an overall rating of the source code out of ten.

Test Results

Each source code was tested, and the obtained results are as follows in the initial state of testing.

Source code rate for -3.83 out of 10



```
C:\Windows\system32\cmd.exe
label2.py:30:0: W0611: Unused import sys (unused-import)
label2.py:25:0: C0411: standard import "import threading" should be placed before
e "import numpy as np" (wrong-import-order)
label2.py:26:0: C0411: standard import "import time" should be placed before "im
port numpy as np" (wrong-import-order)
label2.py:28:0: C0411: standard import "import glob" should be placed before "im
port numpy as np" (wrong-import-order)
label2.py:29:0: C0411: standard import "import random" should be placed before "
import numpy as np" (wrong-import-order)
label2.py:30:0: C0411: standard import "import sys" should be placed before "imp
ort numpy as np" (wrong-import-order)
label2.py:35:0: C0411: standard import "import tkinter" should be placed before
"import numpy as np" (wrong-import-order)

-----
Your code has been rated at -3.83/10 (previous run: -3.83/10, +0.00)

E:\Degree\4th year 1st semester\Project\Endotracheal-Intubation-Automation\CNN_f
or classification\CNN windows>
```

Figure 6.4: Pylint test results

Each code warnings and errors can be inspected easily via PyCharm.

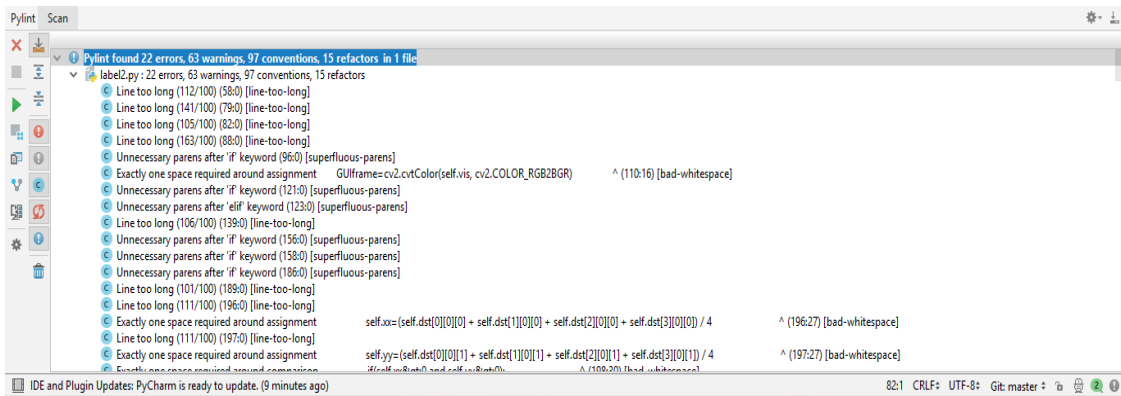


Figure 6.5: Integrate Pylint with PyCharm

After inspecting errors, these code segments were refactored and solved the issues. It is not necessary to solve all the issues since their different coding styles. Further, it suggests alterations for comments as well.

There were ambiguous error notifications which show that OpenCV does not have members called 'cvtColor', 'COLOR_RGB2BGR', 'imread' etc and they are as follows.

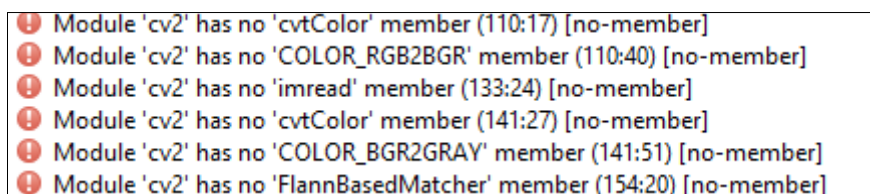


Figure 6.6: Ambiguous error notifications

These errors were neglected. And other most conventions, errors, and standards were corrected and the final rating for the code obtained as 8.93 as follows.

```

C:\Windows\system32\cmd.exe
style <invalid-name>
label2.py:182:16: C0103: Variable name "h" doesn't conform to snake_case naming
style <invalid-name>
label2.py:182:19: C0103: Variable name "w" doesn't conform to snake_case naming
style <invalid-name>
label2.py:214:22: C0103: Variable name "g" doesn't conform to snake_case naming
style <invalid-name>
label2.py:342:16: C0103: Variable name "im" doesn't conform to snake_case naming
style <invalid-name>
label2.py:358:16: C0103: Variable name "t" doesn't conform to snake_case naming
style <invalid-name>
label2.py:536:0: C0103: Constant name "run" doesn't conform to UPPER_CASE naming
style <invalid-name>

-----
Your code has been rated at 8.93/10 (previous run: 8.93/10, +0.00)

E:\Degree\4th year 1st semester\Project\Endotracheal-Intubation-Automation\CNN_f
or_classification\CNN windows>

```

Figure 6.7: Final Pylint rate for the corrected code

6.4 Test meetings

During development time after the completion of each unit, the team gathered and discussed the test that needs to be carried out for each component.

Table 6.1: Measures and Metrics

Metric	Measure
Accuracy of classification	Score percentage
Responsiveness of camera and application	Seconds
Speed of prediction	Seconds

6.5 Evaluation

6.5.1 Model accuracy evaluation

The trained model in the system has the accuracy of 97.8%. This accuracy varies with the type of model used in the training process. Pre-trained models were used, and the accuracies were compared. The models MobileNetV2, Inception v3, Resnet v1 50, Inception v1 showed accuracies 97.8, 97.7, 95.2, 92.7. Inception v3 model and the MobileNetV2 showed the highest accuracy. Therefore the use of the MobileNetV2 model contributes to the overall reliability system. The system which shows a similar approach (section 2.2) have the accuracy of 96%. Hence this system has improved accuracy.

6.5.2 Pointer generation precision evaluation

Pointer generation precision differs upon the good matches count value and the images used for feature matching. Images with high details contribute to the higher precision. When the good matches count is in higher value 40-50, the pointer precision is around 95-100 %. But that will lead for neglecting most of the matching pairs. Hence a pointer cannot obtain continuously in the process. Therefore a good match count should be less value. The value 20-25 provide average precision accuracy around 80-90 %. That will lead to continuous generations of the pointer.

6.5.3 Efficiency evaluation

The efficiency depends mainly on the prediction model. Prediction is the main bottleneck in the system, and multithreading was implemented to improve efficiency, and it degrades the efficiency due to the python GIL effect. Multiprocessing in the python improved the efficiency in milliseconds.

Table 6.2: Efficiency evaluation

Mode of implementation	Frame generation time(ms)
Sequential mode	334.4
Threading mode	412.3
Multiprocessing mode	280.6

6.5.4 Approach evaluation

The system was first evaluated by giving a full demonstration to the project advisor. He was satisfied with the 96% accuracy in the model and asked to improve further. He stated that the approach this project took to achieve the research goal, is correct. Then the system was evaluated by two other respiratory physicians. They said that this is a very valuable research idea, this can be extended to detect the cancer cells in lungs and hence, developing this type of device will do greater good for Sri Lankan medical field.

Then system was evaluated by final year medical students from faculty of medicine, University of Peradeniya. Their feedback was collected through a survey and summary of the feedback is shows in usability test statistical data table.

All of the satisfaction letters are attached in Appendix C.

Chapter 7 - Conclusion

This project was undertaken to generate navigation details in Endotracheal Intubation process to overcome the major issue of the current approach which is not having visuals inside the body and to reduce complications. A neural network-based approach used to train a model which can predict navigation details according to the current tube location with 97.8% accuracy. The physicians who evaluated the system concluded that the developed system had met the goal with a correct approach.

7.1 Limitations

Several limitations of this system need to be acknowledged. First one is the limited number of videos that were used in the model training process. To make the predictions more accurate system must be trained with more images. The second limitation is the lack of images of trauma patients. Since the anatomy of every patient is not the same, the system should be trained with the images of unhealthy people as well. Instability of the navigation pointer due to breathing movement is a drawback of the system that needs to be addressed. When considering the performance, slowness of the system and high usage of system memory are major issues that should be solved when improving the overall system performance.

7.2 Future work

It is recommended that further research can be undertaken in the following areas. This system can be developed to an AI to automate the Endotracheal Intubation Process fully. According to the professional opinion, this approach can be adapted to detect cancer cells in the lungs by providing navigation details to reach cancer cells in CT (Computed Tomography).

References

- [1] T. M. Hemmerling, R. Taddei, M. Wehbe, C. Zaouter, S. Cyr and J. Morse, “First robotic tracheal intubations in humans using the Kepler intubation system”, *British Journal of Anaesthesia(BJA)*, vol. 108, pp. 1011–1016, 2012
- [2] “OpenCV documentation.” [Online]. Available: <https://docs.opencv.org/3.0-beta/index.html>
- [3] “TensorFlow-Slim image classification model library” [Online]. Available: <https://github.com/tensorflow/models/tree/master/research/slim>
- [4] Teoh WH, Shah MK and Sia AT, “Randomised comparison of Pentax AirwayScope and Glidescope for tracheal intubation in patients with normal airway anatomy”, *Anaesthesia*, vol. 64, pp. 1125-9, 2009
- [5] “How to Retrain an Image Classifier for New Categories” [Online]. Available: https://www.tensorflow.org/hub/tutorials/image_retraining
- [6] “Continuous video classification with TensorFlow, Inception and Recurrent Nets” [Online]. Available: <https://blog.coast.ai/continuous-video-classification-with-tensorflow-inception-and-recurrent-nets-250ba9ff6b85>
- [7] “Numpy and Scipy Documentation” [Online]. Available: <https://docs.scipy.org/doc/>
- [8] “Faster video file FPS with cv2.VideoCapture and OpenCV” [Online]. Available: <https://www.pyimagesearch.com/2017/02/06/faster-video-file-fps-with-cv2-videocapture-and-opencv/>
- [9] Kaiming He, Xiangyu Zhang, Shaoqing Ren and Jian Sun, “Deep Residual Learning for Image Recognition”, Cornell University, 2015
- [10] “Fully Connected Layer-Artificial Intelligence.” [Online]. Available: https://leonardoaraujasantos.gitbooks.io/artificial-intelligence/content/fc_layer.html
- [11] “Convolutional Neural Networks - CS231n Convolutional Neural” [Online]. Available: <http://cs231n.github.io/convolutional-networks/>
- [12] “ResNet, AlexNet, VGGNet, Inception: Understanding various architectures of Convolutional Networks” [Online]. Available: <https://cv-tricks.com/cnn/understand-resnet-alexnet-vgg-inception/>

- [13] “Image Classification Transfer Learning with Inception v3” [Online]. Available: <https://codelabs.developers.google.com/codelabs/cpb102-xf-learning/index.html?index=..%2F..%2Findex#1>
- [14] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens and Zbigniew Wojna, “Rethinking the Inception Architecture for Computer Vision”, Cornell University, 2015
- [15] “Running Inception on Cloud TPU, Cloud TPU, Google Cloud.” [Online]. Available: <https://cloud.google.com/tpu/docs/tutorials/inception>
- [16] “WordNet - A Lexical Database for English” [Online]. Available: <https://wordnet.princeton.edu/>
- [17] “Image Recognition - TensorFlow” [Online]. Available: https://www.tensorflow.org/tutorials/images/image_recognition
- [18] “Top 10 Pretrained Models to get you Started with Deep Learning” [Online]. Available: <https://www.analyticsvidhya.com/blog/2018/07/top-10-pretrained-models-get-started-deep-learning-part-1-computer-vision/>
- [19] “Comparing pre-trained deep learning models for feature extraction.” [Online], Available: <https://tech.zegami.com/comparing-pre-trained-deep-learning-models-for-feature-extraction-c617da54641>
- [20] Kaiming He, Xiangyu Zhang, Shaoqing Ren and Jian Sun, “Deep Residual Learning for Image Recognition”, Cornell University, 2015

Appendix A: Diagrams

A.1 System modeling

A.1.1 Use Case Diagram

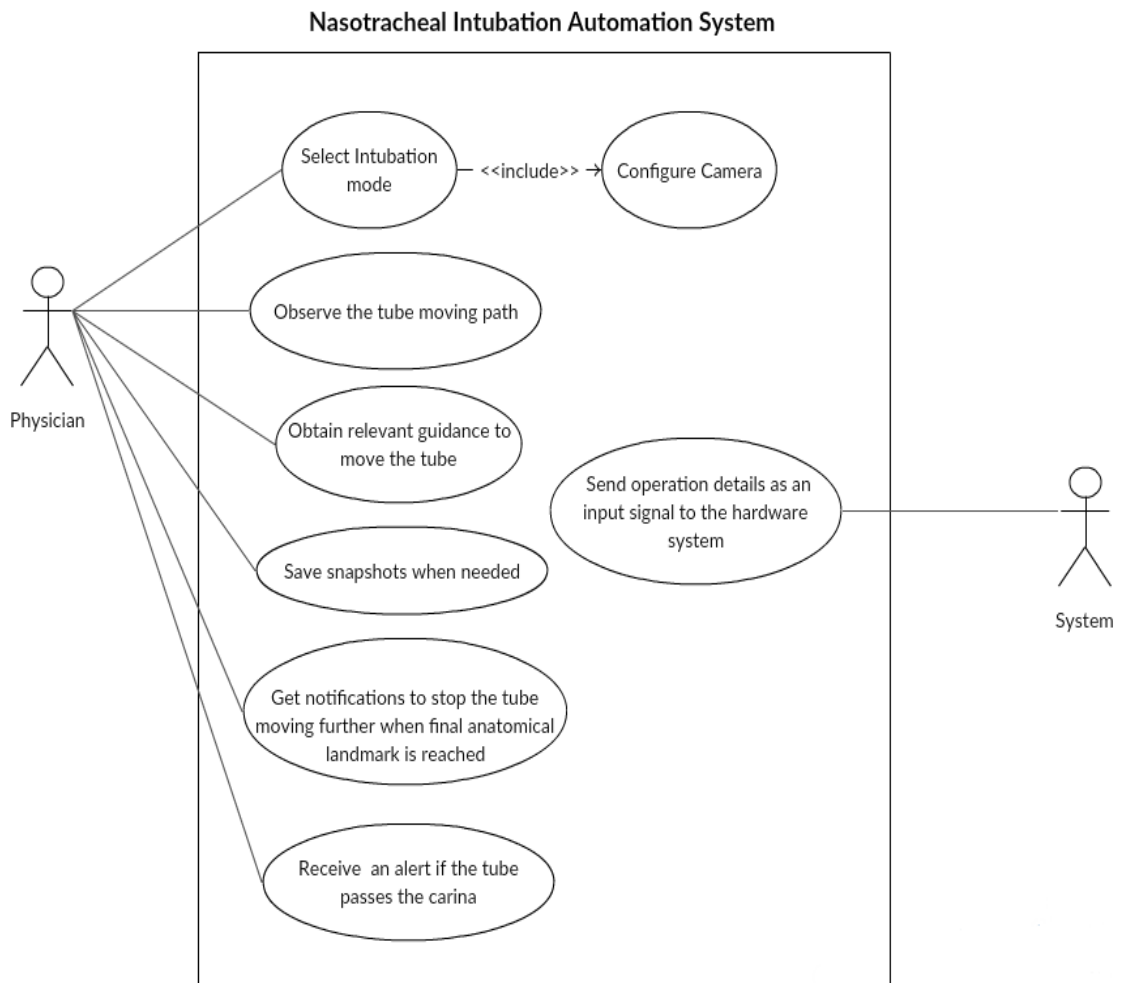


Figure A.1.1: Use case diagram

A.1.2 Use Case Narratives

Use case 01: Select the intubation mode

Use case ID	01
Use Case Name	Select the intubation mode
Primary Actors	Physician
Other Actors	None
Description	At the beginning of the process, the doctor has to select the mode either manual or automatic
Priority	Low
Precondition	None
Trigger	This use case initiated when the user needs to use the application for the first time
Normal Course	<ol style="list-style-type: none">1. Launch the application2. The system shows “Click on preferred mode” Manual or Automatic3. The user selects one of the two modes4. System asks “Are sure you want to switch to selected mode”5. The user selects Yes or No6. The system switches it mode accordingly
Alternative Courses	None
Post conditions	The system has been switched to the selected mode

Use case 02: Configure the relevant camera

Use case ID	02
Use Case Name	Configure the relevant camera
Primary Actors	Physician
Other Actors	None
Description	According to the selected mode, the relevant camera should be then configured before starting intubation

Priority	Medium
Precondition	Either manual or automatic mode should have been selected
Trigger	This use case is initiated at the beginning of the intubation process
Normal Course	<ol style="list-style-type: none"> 1. If the mode is manual then configure the camera attached to the intubation tube 2. Otherwise, configure the camera attached to the hardware device
Alternative Courses	None
Post conditions	Properly configured camera ready to be used in the intubation process

Use case 03: Observe the tube moving path through a screen

Use case ID	03
Use Case Name	Observe the tube moving path through a screen
Primary Actors	Physician
Other Actors	Supporting physicians
Description	This feature can be implemented on both manual and automatic mode to provide the physician with the visuals inside the body where the tube is passing.
Priority	High
Precondition	Intubation mode should have been selected
Trigger	This use case is initiated when the intubation tube starts moving into the body
Normal Course	<ol style="list-style-type: none"> 1. Show inside the body on the computer screen 2. Show inside the body in the extended screen
Alternative Courses	None
Post conditions	The path that the tube is being moved should be displayed on both screens

Use case 04: Obtain relevant guidance to move the tube

Use case ID	04
Use Case Name	Obtain relevant guidance to move the tube
Primary Actors	Physician
Other Actors	None
Description	The physician should be provided with relevant navigation details to move the tube.
Priority	High
Precondition	Screens should be configured to display tube moving path and relevant operation details
Trigger	This use case is triggered when the tube starts moving inside the body
Normal Course	1. Provide navigations details 2. Provide operation details when to stop or to move
Alternative Courses	None
Post conditions	None

Use case 05 : Save snapshots when needed

Use case ID	05
Use Case Name	Save snapshots when needed
Primary Actors	Physician
Other Actors	None
Description	Any abnormal or unexpected findings are recorded by saving a snapshot of the live display during the intubation process.
Priority	Medium
Precondition	The camera should have been configured to capture the snapshots with appropriate speed
Trigger	This use case is initiated when the physician press save button

Normal Course	1. Physician press save button 2. The system saves the snapshot into a predefined location
Alternative Courses	None
Post conditions	The system must save the snapshot to the relevant location

Use case 06: Get notifications to stop the tube moving further when the final anatomical landmark is reached

Use case ID	06
Use Case Name	Get notifications to stop the tube when the final anatomical landmark is reached
Primary Actors	Physician
Other Actors	None
Description	Identify the final anatomical landmark to stop the procedure and notify
Priority	High
Precondition	Intubation tube should have reached near the final anatomical landmark
Trigger	This use case will be initiated when the tube is about to pass the final anatomical landmark
Normal Course	1. Intubation tube is moved to the final anatomical landmark 2. Send notification to stop the procedure
Alternative Courses	None
Post conditions	Stop the tube moving further into the body

Use case 07: Receive an alert if the tube passes the carina

Use case ID	07
Use Case Name	Receive an alert if the tube passes the carina
Primary Actors	Physician
Other Actors	None
Description	The tube must not pass the carina and enters a lung. Therefore an alert must be sent in case the tube passes the carina
Priority	Medium
Precondition	Intubation tube should have reached the carina
Trigger	This use case is triggered when the tube passes the carina
Normal Course	<ol style="list-style-type: none"> 1. Intubation tube is moved to the carina 2. The system sends a notification to stop the tube moving 3. The system sends operation details to move the tube backward
Alternative Courses	None
Post conditions	The tube should be moved to backward

Use case 08: Send operation details as an input signal to the hardware system

Use case ID	08
Use Case Name	Send operation details as an input signal to the hardware system
Primary Actors	System
Other Actors	None
Description	In automated mode, hardware system should be provided with relevant operation details
Priority	Low

Precondition	The system should have generated navigation details for the current position of the tube
Trigger	This use case is triggered when the automatic mode starts
Normal Course	<ol style="list-style-type: none"> 1. The hardware device is received relevant operation details 2. The tube is moving according to the receiving signals
Alternative Courses	None
Post conditions	The tube should stop moving when it reaches the final anatomical landmark

A.1.3 Class Diagram

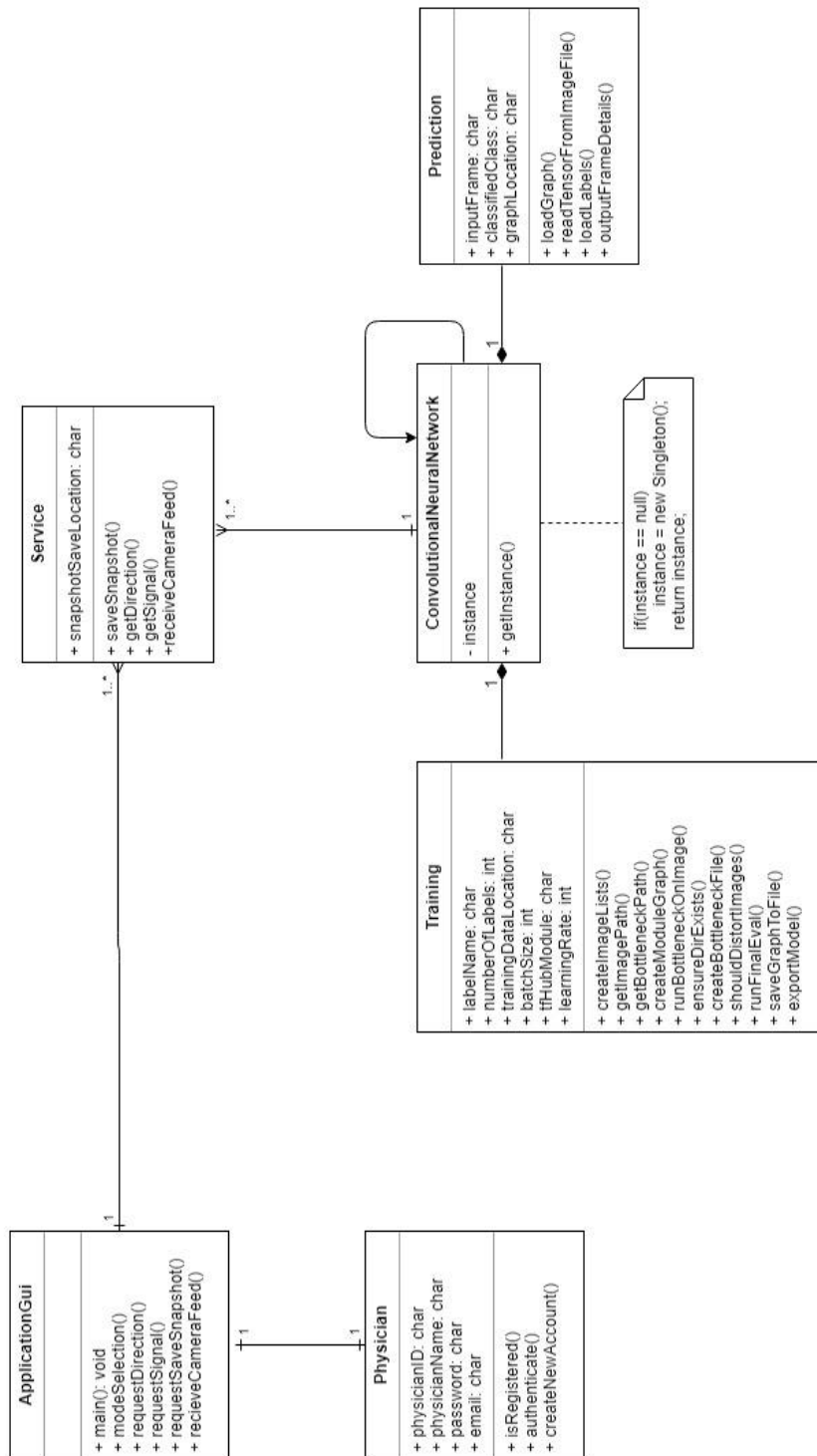


Figure A.1.3: Class diagram

A.1.4 Activity Diagram

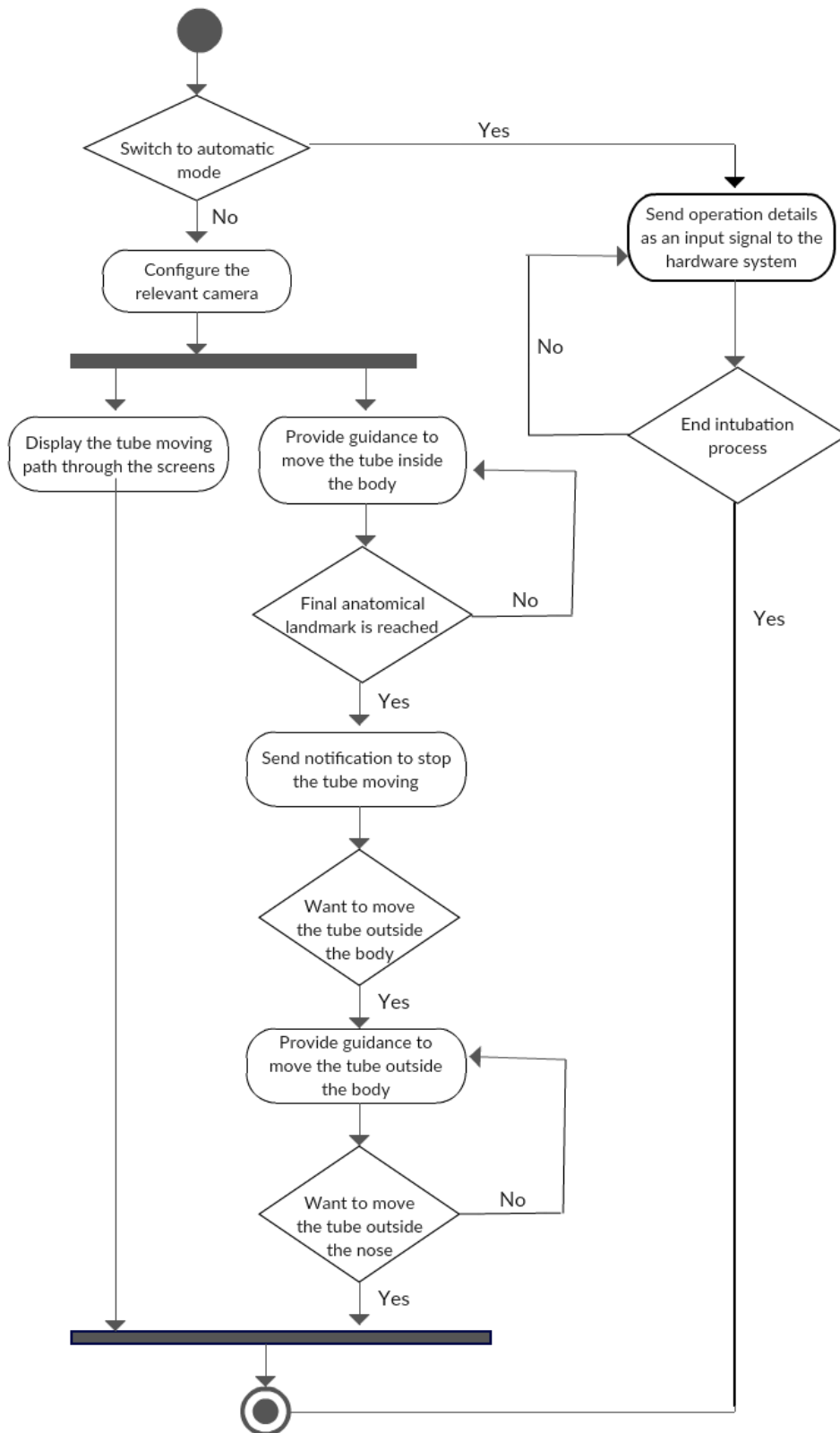


Figure A.1.4: Activity diagram

A.1.5 Sequence Diagram

Send operation details

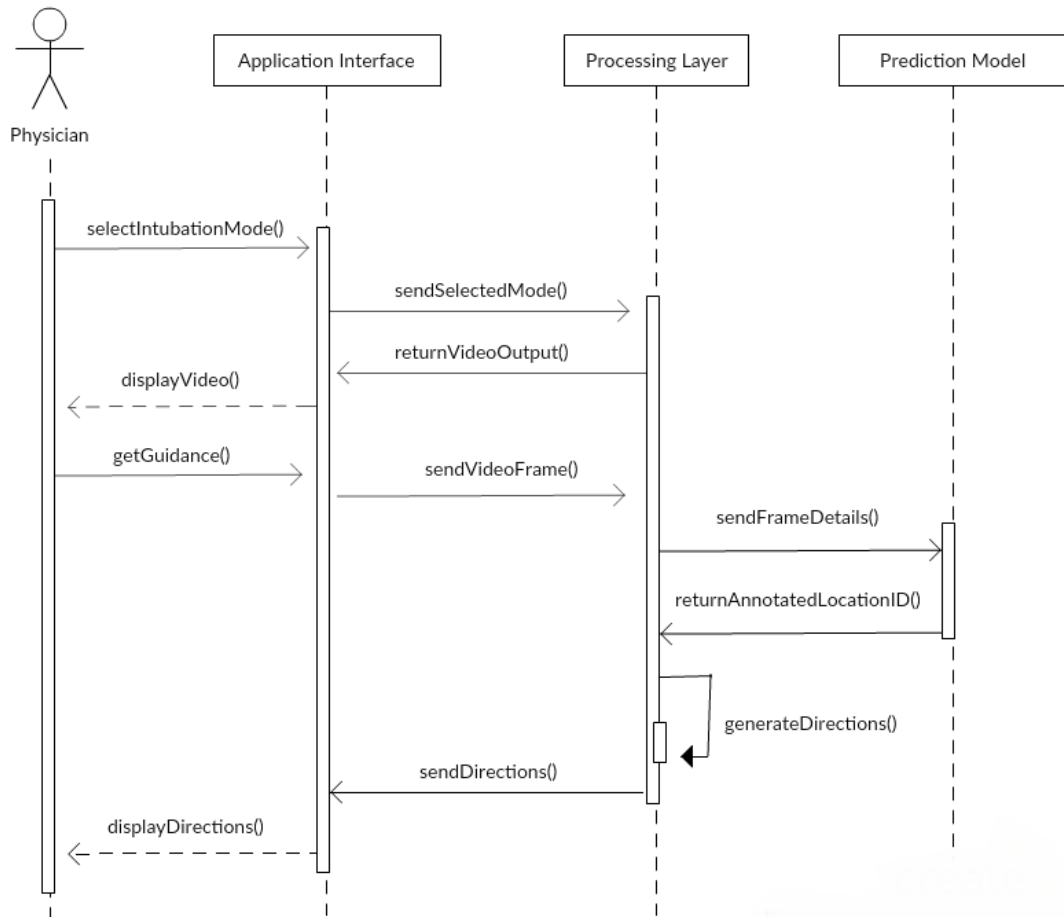


Figure A.1.5.1: Send operation Details

Save Snapshots

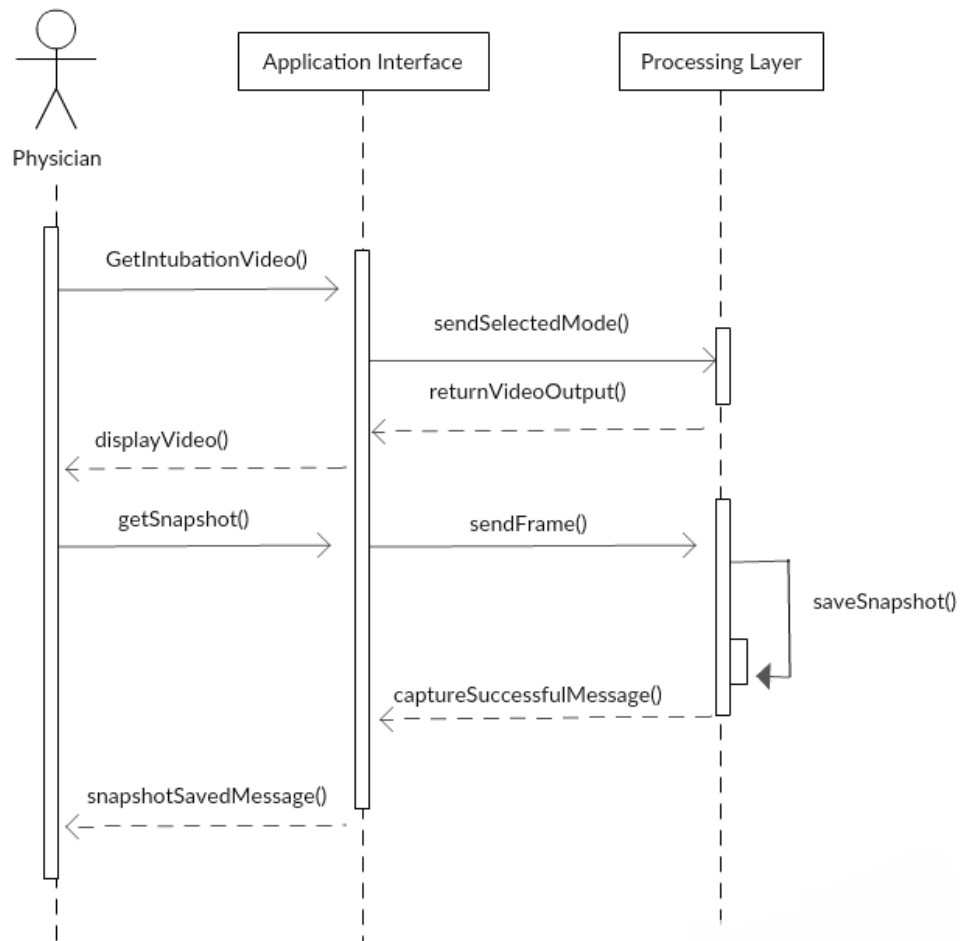


Figure A.1.5.2: Save snapshots

Appendix B: Code Listings

B.1 pytest script

```
import load_graph
model_file = "E:/Degree/4th year 1st semester/ Project/
Endotracheal-Intubation-Automation/CNN_for classification/CNN
windows/tmp/output_graph.pb"

def test_load_graph():
    val=load_graph.load_graph(model_file)
    assert val is object

def test_load_label():
    label_file = self.label_file = "E:/Degree/4th year 1st
semester/Project/Endotracheal-Intubation-
Automation/CNN_for classification/CNN windows/tmp/
output_labels.txt"
    val = load_graph.load_label(label_file)
    assert val == ['anatomical location 1', 'anatomical
location 10', 'anatomical location 2', 'anatomical
location 3', 'anatomical location 4', 'anatomical location
5', 'anatomical location 6', 'anatomical location 7',
'anatomical location 8', 'anatomical location 9']
```

B.2 Feature Matching

```
match_img = cv2.imread(filename, 0)
kp1, des1 = self.surf.detectAndCompute(match_img, None)
temp_out_img = cv2.cvtColor(self.img2, cv2.COLOR_BGR2GRAY)
kp2, des2 = self.surf.detectAndCompute(temp_out_img, None)

FLANN_INDEX_KDTREE = 0
```

```

index_params = dict(algorithm=FLANN_INDEX_KDTREE, trees=5)
search_params = dict(checks=50)
flann = cv2.FlannBasedMatcher(index_params, search_params)

if des2 is None:
    break
if len(des2) <= 1:
    break

matches = flann.knnMatch(des1, des2, k=2)
good = []

for m, n in matches:
    if m.distance < 0.99 * n.distance:
        good.append(m)

if len(good) > 20:
    self.first_pointer = True
    src_pts = np.float32([kp1[m.queryIdx].pt for m in
        good]).reshape(-1, 1, 2)
    dst_pts = np.float32([kp2[m.trainIdx].pt for m in
        good]).reshape(-1, 1, 2)
    M = []

```

B.3 Homography generation

```

M, mask = cv2.findHomography(src_pts, dst_pts, 0, 5.0)

h, w = temp_out_img.shape
pts = np.float32([[0, 0], [0, h - 1], [w - 1, h - 1], [w - 1,
0]]).reshape(-1, 1, 2)

if isinstance(M, np.ndarray):
    self.dst = cv2.perspectiveTransform(pts, M)

```

Appendix C

C.1 Contribution

In accordance to the problem definition in this dissertation, all the three team members were equally contributed in finding an approach to achieve the proposed goal. Different components in the project were implemented by different members and some components were implemented collectively by all the members.

Individual contribution for the project

Contributor 1: R.M.R Wanigasekara

Transfer learning approach was examined and neural network training process and the pre-trained model configuration was implemented. From the trained model a predicting component was implemented which output predicted class percentages. A pointer generation model which use SURF feature matching was developed. Prediction accuracy was tested using different image sets to find the applicable percentage value of the accuracy in the softmax layer. The architecture of the system was revised and contributed in designing. Analysis of path from nose to carina and capturing the images of each anatomical location was carried out collectively.

Contributor 2: S.D.M.H Siyambalapitiya

Different SIFT, SURF mechanisms are evaluated for image classification. Integration of three models pointer generation model, and navigation detail generation and video output was performed. Major component of the architecture of the system which having a layered architecture was designed. The main user interface component which is video viewing section was designed and then integrated with the overall output in the prediction model. Further functionalities of login and save snapshot was developed. Image preprocessing tasks which are cropping, resizing were performed. Unit testing was carried using pytest and changes were made to pass the test cases. Analysis of path from nose to carina and capturing the images of each anatomical location was carried out collectively.

Contributor 3: S.D.S.H Dissanayake

Different network models were evaluated to obtain the optimal pre-trained model to use in the system. Image preprocessing task to reduce noise which is gaussian smoothing was performed. Source code testing and refactoring operations using pylint was performed. Designing of user interface and building up the functionalities of file upload, snapshot and intubation modes was developed. Architecture of the system was revised and contributed in re designing. Analysis of path from nose to carina and capturing the images of each anatomical location was carried out collectively.

C.2 Recommendations

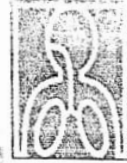


Prof. Aruna Munasinghe is evaluating the project on 2019.01.08 at University of Colombo School of Computing



Central Chest Clinic

Medical Research Institute Premises (MRI)



TO WHOM IT MAY CONCERN,

This is to express our satisfaction on virtual bronchoscopic endo-bronchial navigational software done by below group of students.

These undergraduate students are from University of Colombo School of Computing, Sri Lanka. Namely, R.M.R. Wanigasekara, S.D.M.H. Siyambalapitya, S.D.S.H. Dissanayake.

We wish index software utility will aid future life saving as well as reduce complications in difficult intubations.

Thank you.

1. Dr. Ravini Karunathilake


.....

Dr. R. De S. Karunathilake
MBBS, MD, MRCP
Consultant Respiratory Physician
National Hospital of Sri Lanka &
Central Chest-Clinic Colombo.

2. Dr. R. Asha Samaranayake


.....

Dr. R. Asha Samaranayake (MBBS,MD)
GMC UK 7504570 SLMC 13654
Residential Pulmonologist
NHSL- Central Chest Clinic-Colombo

C.3 User Survey Questionnaire for the Endotracheal Intubation System

Thank you for the contribution given by participating in the user evaluation survey of the project “Automate Endotracheal Intubation”. The project goal is to generate directions to guide and automate endotracheal intubation process. The main concerns in the project are generate navigation details, pointer generation in the video, and interface handling. This evaluation is used as a feedback of the system and will helpful to improve the overall performance of the system.

Please complete the following questionnaire with specific regard to the above enquiry, by placing a CROSS in the appropriate box.

		strongly agree	agree	neither agree nor disagree	disagree	strongly disagree
1.	It is simple to use this system.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2.	This system performs its intended task.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3.	The graphical user interface of this system is pleasant.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4.	It was easy to learn the system usage.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
5.	I am satisfied with the navigation details.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
6.	The information provided with the system is easy to understand.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
7.	I am satisfied with the performance of the system.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
8.	This system provide a clear view of the intubation path.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
9.	The organization of information on the system screen is clear.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
10.	Overall, I am satisfied with this system.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

User Survey Questionnaire for the Endotracheal Intubation System

Thank you for the contribution given by participating in the user evaluation survey of the project “Automate Endotracheal Intubation”. The project goal is to generate directions to guide and automate endotracheal intubation process. The main concerns in the project are generate navigation details, pointer generation in the video, and interface handling. This evaluation is used as a feedback of the system and will helpful to improve the overall performance of the system.

Please complete the following questionnaire with specific regard to the above enquiry, by placing a CROSS in the appropriate box.

		strongly agree	agree	neither agree nor disagree	disagree	strongly disagree
1.	It is simple to use this system.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2.	This system performs its intended task.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3.	The graphical user interface of this system is pleasant.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4.	It was easy to learn the system usage.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
5.	I am satisfied with the navigation details.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
6.	The information provided with the system is easy to understand.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
7.	I am satisfied with the performance of the system.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
8.	This system provide a clear view of the intubation path.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
9.	The organization of information on the system screen is clear.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
10.	Overall, I am satisfied with this system.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

User Survey Questionnaire for the Endotracheal Intubation System

Thank you for the contribution given by participating in the user evaluation survey of the project “Automate Endotracheal Intubation”. The project goal is to generate directions to guide and automate endotracheal intubation process. The main concerns in the project are generate navigation details, pointer generation in the video, and interface handling. This evaluation is used as a feedback of the system and will helpful to improve the overall performance of the system.

Please complete the following questionnaire with specific regard to the above enquiry, by placing a CROSS in the appropriate box.

		strongly agree	agree	neither agree nor disagree	disagree	strongly disagree
1.	It is simple to use this system.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2.	This system performs its intended task.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3.	The graphical user interface of this system is pleasant.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4.	It was easy to learn the system usage.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
5.	I am satisfied with the navigation details.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
6.	The information provided with the system is easy to understand.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
7.	I am satisfied with the performance of the system.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
8.	This system provide a clear view of the intubation path.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
9.	The organization of information on the system screen is clear.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
10.	Overall, I am satisfied with this system.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

User Survey Questionnaire for the Endotracheal Intubation System

Thank you for the contribution given by participating in the user evaluation survey of the project “Automate Endotracheal Intubation”. The project goal is to generate directions to guide and automate endotracheal intubation process. The main concerns in the project are generate navigation details, pointer generation in the video, and interface handling. This evaluation is used as a feedback of the system and will helpful to improve the overall performance of the system.

Please complete the following questionnaire with specific regard to the above enquiry, by placing a CROSS in the appropriate box.

		strongly agree	agree	neither agree nor disagree	disagree	strongly disagree
1.	It is simple to use this system.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2.	This system performs its intended task.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3.	The graphical user interface of this system is pleasant.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4.	It was easy to learn the system usage.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
5.	I am satisfied with the navigation details.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
6.	The information provided with the system is easy to understand.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
7.	I am satisfied with the performance of the system.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
8.	This system provide a clear view of the intubation path.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
9.	The organization of information on the system screen is clear.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
10.	Overall, I am satisfied with this system.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

User Survey Questionnaire for the Endotracheal Intubation System

Thank you for the contribution given by participating in the user evaluation survey of the project “Automate Endotracheal Intubation”. The project goal is to generate directions to guide and automate endotracheal intubation process. The main concerns in the project are generate navigation details, pointer generation in the video, and interface handling. This evaluation is used as a feedback of the system and will helpful to improve the overall performance of the system.

Please complete the following questionnaire with specific regard to the above enquiry, by placing a CROSS in the appropriate box.

		strongly agree	agree	neither agree nor disagree	disagree	strongly disagree
1.	It is simple to use this system.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2.	This system performs its intended task.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3.	The graphical user interface of this system is pleasant.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4.	It was easy to learn the system usage.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
5.	I am satisfied with the navigation details.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
6.	The information provided with the system is easy to understand.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
7.	I am satisfied with the performance of the system.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
8.	This system provide a clear view of the intubation path.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
9.	The organization of information on the system screen is clear.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
10.	Overall, I am satisfied with this system.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

User Survey Questionnaire for the Endotracheal Intubation System

Thank you for the contribution given by participating in the user evaluation survey of the project “Automate Endotracheal Intubation”. The project goal is to generate directions to guide and automate endotracheal intubation process. The main concerns in the project are generate navigation details, pointer generation in the video, and interface handling. This evaluation is used as a feedback of the system and will helpful to improve the overall performance of the system.

Please complete the following questionnaire with specific regard to the above enquiry, by placing a CROSS in the appropriate box.

		strongly agree	agree	neither agree nor disagree	disagree	strongly disagree
1.	It is simple to use this system.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2.	This system performs its intended task.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3.	The graphical user interface of this system is pleasant.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4.	It was easy to learn the system usage.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
5.	I am satisfied with the navigation details.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
6.	The information provided with the system is easy to understand.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
7.	I am satisfied with the performance of the system.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
8.	This system provide a clear view of the intubation path.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
9.	The organization of information on the system screen is clear.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
10.	Overall, I am satisfied with this system.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Usability test statistical data

Table C.1: Results of usability test

		strongly agree	agree	neither agree nor disagree	disagree	strongly disagree
1.	It is simple to use this system.	12	5	3	-	-
2.	This system performs its intended task.	9	9	1	1	-
3.	The graphical user interface of this system is pleasant.	15	4	1	-	-
4.	It was easy to learn the system usage.	16	3	-	1	
5.	I am satisfied with the navigation details.	17	3	-	-	-
6.	The information provided with the system is easy to understand.	12	7	1	-	-
7.	I am satisfied with the performance of the system.	7	2	6	3	2
8.	This system provide a clear view of the intubation path.	10	7	3	-	-
9.	The organization of information on the system screen is clear.	14	4	2	-	-
10.	Overall, I am satisfied with this system.	18	2	-	-	-