INFORMATICS INSTITUTE OF TECHNOLOGY

In Collaboration with

UNIVERSITY OF WESTMINSTER

Caught in a Lie:

Automated Deception Detection System Using Human Facial Cues and Facial Chirality

A Dissertation by

Ms. Chamudi Vidanagama

Supervised by

Ms. Abarnah Kirupananda

Submitted in partial fulfilment of the requirements for the BEng (Hons) Software Engineering degree at the University of Westminster.

May 2022

Declaration

I hereby certify that this project report and all the artifacts associated with it is my own work and it has not been submitted before nor is currently being submitted for any degree program.

Full name of student: Chamudi Gayanika Vidanagama

Registration No: 2017298 / w1715750

Signature: …………………………… Date: 05/05/2022

Abstract

Deception detection is an important task carried out in high skate situations like suspect interrogations. Criminal investigations are solely dependent on a suspect's truthful confession for a case to be solved. The only way to make a suspect confess is by conducting a suspect interrogation with evidence. Detectives face difficulties in detecting deception manually due to various reasons such as excessive time requirement, not specialized in deception detection, judgements become human biased, necessity of more detective resources. Due to these reasons manual deception detection has become a hard process to conduct. Therefore, researchers have identified different automated deception detection approaches. These approaches also have many limitations which leads to inaccurate and invalid results.

There is a necessity of an accurate, valid and high performing automated deception detection system. For this reason, **Caught in a Lie**: an automated deception detection system using human facial cues and facial chirality has been introduced. The system takes nonverbal indicators of humans such as facial cues and facial chirality into account and detect deception of interrogation videos. Human face is a display of emotions people feel. Hence, it is hard to hide or change non-verbal indicators. That is the main reason non-verbal indicators were focused in deception detection on the system **Caught in a Lie**. To the best of the author's knowledge, a Facial Chirality deep learning model has not been used for any automated deception detection systems up to this point. However, manual deception detection focuses on facial chirality which gives accurate results.

The proposed system detects deception of interrogation videos accurately and efficiently by outputting the deceptive and truthful percentage of the suspect as well as which indicators caused the deceptiveness or truthfulness of the suspect. Therefore, the **Caught in a Lie** system is a novel automated deception detection system which is the first to focus on facial chirality when detecting deception. This facial chirality model will assist future researchers of the domain or different domains which require facial chirality.

Keywords: Deception Detection, Facial Chirality, Facial Cues, Interrogations, Non-Verbal Indicators, Criminal Investigations, Deep Learning Model

Acknowledgement

First and foremost, I would like to express gratitude to my supervisor, Ms. Abarnah Kirupananda, for guiding me through the whole project and encouraging to complete the project on time. Her guidance and encouragement throughout the past 8 months were a great support for the successful completion of the project.

I owe many thanks to domain experts from Criminal Investigation Department, Sri Lanka Police for participating in the interviews and providing necessary information for the requirement elicitation and evaluation of the project. I wish to express my appreciation for police officers who provided domain information which assisted throughout the whole project.

Besides I would like to thank all the lecturers for supporting and guiding me through academic years which assisted in completing the project. Also, I thank all the batch mates who were supporting and encouraging me for the past four years in completion of the degree program along with the final year project.

Finally, I would like to give my special thanks to my family for assisting and facilitating me with necessities for the successful completion of the project.

Table of Contents

List of Figures

List of Tables

List of Glossaries

Abbreviation	Acronym
AI/ML	Artificial Intelligence Markup Language
API	Application programming Interface
ASCII	American Standard Code for Information Interchange
AWS	Amazon Web Services
CID	Crime Investigation Department
CNN	Convolutional Neural Network
EEG	Electroencephalography
FAU	Facial Action Unit
FBI	Federal Bureau Investigation
FLBD	Facial landmark blink detector
fNIRS	Functional Near-Infrared Spectroscopy
FR	Functional Requirement
GUI	Graphical User Interface
IDE	Integrated Development Environment
LO	Learning Outcome
LSTM	Long-Short Term Memory
ML	Machine Learning
NFR	Non-Functional Requirement
OIC	Officer In-Charge
OOAD	Object-Oriented Analysis and Design
OS	Operating System
RBF	Radial Basic Function
RGB	Red Green Blue
RLT	Real-Life Trial
RNN	Recurrent Neural Network
ROS	Robot Operating System
SLEP	Social, Legal, Ethical and Professional
SVM	Support Vector Machine
VSC	Visual Studio Code
WFM	Weighted Fusion Model

CHAPTER 1: INTRODUCTION

1.1 Chapter Overview

The domain studied in this project is primarily focused on suspect interrogations in criminal investigations where deception detection is carried out as one of the key acts in a suspect interrogation. This chapter outlines the problem discussing the problem domain, problem definition and problem statement. Research gap is identified along with research contributions, aims and objectives. A brief explanation of existing work is carried out to discuss the related work in the domain. Finally, project scope and resource requirements were discussed.

1.2 Problem Domain

Crimes happen all over the world. It has a huge impact on citizens' day to day lives. "A crime is an offence that merits community condemnation and punishment, usually by way of fine or imprisonment" (What is a Crime?, 2021). Criminal investigation is an ensemble of methods which study crimes and apprehend criminals. Criminal investigators discover the identity of criminals and victims, methods used for the crime, motives of the criminal and, also search for witnesses and conduct interrogations accordingly (Criminal investigation, 2021). Criminal investigations are carried out by trained criminal investigators. Criminal investigators use tools to establish facts and develop evidence. These tools are information, interview, interrogation and instrumentation. Investigative interviews of the accused, the accuser and witnesses can be identified as significant tasks included in the criminal investigation process. Goal of these investigative interviews is to gather accurate, reliable and actionable information. Deception detection plays a crucial role in interrogations. Investigative interviews can be categorized in to two sections according to the outcome of the process, which are confrontational interviews and interrogations. In this study the author focuses on interrogation process which is carried out to extract confessions from the suspect of the criminal investigation.

1.2.1 Deception Detection

Deception detection task is executed in the interrogation process to identify the truthfulness and deceptiveness of a suspect. According to Deception Detection | How to Tell If Someone is Lying (2021) "Deception detection refers to the investigative practices used to determine a person's truthfulness and credibility". Fight or flight, cognitive complexity and perception management are three different deception detection approaches (Former FBI Agent Explains Criminal Profiling | Tradecraft | WIRED, 2021). These approaches are done by analysing the key indicators of deception which are verbal, non-verbal and non-linguistic communication of the subject. Identifying what exactly the truth or deceit is associated with both psychological background and body language. In the investigation process, a lie is a statement intended to make people believe something that the speaker does not believe to be true, or not the whole truth (The Definition of Lying and Deception (Stanford Encyclopedia of Philosophy), 2021). There is an assumption that liars will exhibit stress-based cues due to the fear of being caught and guilty feeling about lying. Criminal investigators should identify the deception of a criminal when carrying out a criminal investigation in order to bring justice to the innocent and punish offenders (Lakshan et al., 2019). Currently, the deception detection process is done manually in most countries including Sri Lanka. However, manual deception detection has major limitations as specified below and remains a difficult task.

a. Task requires excessive analysis time
b. Lack of highly trained and skilled detectives specialized in deception detection
c. Judgements can be human biased
d. Poor classification of deception

Therefore, in order to get better and accurate results from the manual deception detection in interrogation, at least two skilled detectives who are specialized in human deception detection are needed to interrogate a single suspect.

1.2.2 Crime & Deception Detection in Sri Lanka

According to Sri Lanka 2020 Crime and Safety report, recent crime statistics indicate an upward trend for serious crime in the country overall (Working Together to Protect U.S. Organizations Overseas, 2021). Even though there is a considerable number of detectives assigned to work on criminal investigations in Sri Lanka, the number of detectives specialized in deception detection are limited. The interrogation process is a time-consuming task specially when detecting deception as a part of the process to obtain accurate results. As the crime rate is high in Sri Lanka and the number of specialized detectives available are limited, time spent on and the detective allocation to each case has become less. Hence manual deception detection has become ineffective due to limitations such lack of time and limited number of detectives specialized in deception detection.

1.2.3 Automated Deception Detection

In order to overcome limitations found in manual deception detection, researchers have identified different ways of automating deception detection. One of the mostly used automated deception detection techniques is polygraph. According to lie detector | Definition, Accuracy, & Facts, (2021), polygraph is an instrument for recording physiological phenomena such as blood pressure, pulse rate and respiration of a human subject as he answers questions put to him by an operator; these data are then used as the basis for making a judgment as to whether the subject is lying. Even though polygraph is used for deception detection, there are many limitations in polygraph. Polygraph has a direct physical contact with the subject and as polygraph test depends on pulse rate and blood pressure, even innocent suspects can fail the test if that person is nervous or has any health issues. Therefore, liability of the polygraph is still in question. Even though polygraph tests are currently available in some countries, there is still a necessity of the same amount of time and detectives in order to interrogate a suspect accurately.

Accordingly, if an accurate automated deception detection system which can be used to analyse the features of a suspect without any physical contact as detectives specialized in detecting deception, it will be more useful and effective for the interrogation process.

1.3 Problem Definition

Deception Detection is one of the key tasks carried out in a suspect interrogation. Accurate deception detection helps the criminal investigations to solve crimes easily. Manual deception detection is carried out by highly trained skilled detectives specialized in deception detection who can analyse human facial cues and posture and conclude if the suspect is lying or telling the truth. This task requires lots of time and specialized detectives. Due to the high crime rates and lack of specialized detectives, manual deception detection has become an ineffective task in interrogation. Therefore, researchers have discovered different techniques of detecting deception such as using polygraph tests, EEG tests and emotion detection. However, the problems with some of these techniques are that these techniques have a physical interaction with the subject which can make the subject feel uneasy and nervous. This deception detection conclusion could lead the interrogation to a wrong direction. According to the above discussed concerns relevant to currently available deception detection processes, there is a necessity of having an automated deception detection system which can act as a trained detective to identify deception of a subject of an interrogation by analysing the human facial cues and behaviour of the subject and suggest whether the subject shows deceptiveness or truthfulness.

1.3.1 Problem Statement

Currently available deception detection processes are time consuming, ineffective and less accurate which require highly trained and specialized detectives in order to perform the interrogation process effectively and accurately. Introducing an accurate automated deception detection system which behaves as a trained detective and detect deception by analysing human facial cues and facial chirality will help to improve the process.

1.4 Research Motivation

Crimes happen daily all over the world. Criminal investigations are carried out in order to solve crimes and serve justice. Author is an egalitarian who believed that all the people are equal and deserve equal rights. Therefore, author is concerned about the fairness in a criminal investigation where justice being served for the victims and punishments being sentenced to the responsible criminals. The author's interest in the domain of criminal investigation and got an opportunity to identify one of the gaps in the current criminal investigation processes. Identified area is deception detection which is one of the key tasks carried out in interrogation process. It's proven that the manual deception detection process requires a large amount of time, skilled detectives specialized in deception detection and these results can be human biased. Due to high crime rates and lack of detectives specialized in deception detection, interrogation processes have become hard for the detectives and ineffective. Moreover, the currently available deception detection systems lack accuracy. These limitations cause inaccurate results in interrogations which may lead to unfair judgements such as punish innocent or release guilty. To overcome all these shortcomings and effectively continue the interrogation processes with the help of an accurate automated deception detection system which analyses human facial cues and facial chirality will have a great impact on the interrogation process in criminal investigations.

1.5 Existing Work

Deception detection is an important task carried out in the interrogation process. Deception detection is mostly done manually by detectives who are specialized in the area. Limitations in manual deception detection are lack of skilled detectives specialized in deception detection, require an excessive amount of time and decisions can be human biased. Therefore, researchers have identified new techniques to detect deception. Polygraph test is the most common deception detection technique which has been used. The limitations of polygraph are the accuracy rate is low and even innocent subjects can fail the test when they are nervous, require sophisticated equipment setup, overt in nature and require trained operators to use, results can differ due to peoples' health issues. Please find the existing work in Appendix A.

1.6 Research Gap

Researchers have identified different approaches to detect deception. But all these techniques have different limitations. Polygraph tests, EEG tests (Lakshan et al., 2019), fNIRS (Zhang et al., 2021) which have physical contact with the subject in order to collect information have limitations such as information depend on people's health issues, people can be nervous with physical contact and it may lead to inaccurate information. Most of the approaches associated with facial cue analysis focus on a single aspect of the face, for example, blink detection, eye gazing and facial expressions. These individual analyses provide insufficient information to come to a conclusion in detecting deception. This insufficient information led to inaccurate results. Limitations with the multimodal approach where video, audio, heart rates, EEG were taken into account are overfitting due to lack of data, low performance, too much functionality and low accuracy. The proposed system detects deception by analysing human facial cues and facial chirality by identifying comfort and discomfort of the suspect. To the best of the knowledge of the author, this system is the first to analyse facial chirality to detect deception.

1.7 Contribution to the body of knowledge

1.7.1 Domain Contribution

Deception detection has been a challenging task for the detectives through the interrogation. Even though there are different methods proposed to detect deception, detectives still struggle in deception detection due to limitations in proposed methods. Proposing a deception detection system which acts as a supportive detective by analysing human facial cues and facial chirality in order to identify comfort and discomfort of the subject and suggest whether the subject is truthful or deceitful. This deception detection system will be a great help for the detectives who interrogate the subject and will be able to attain accurate results in the interrogation process. To the best of the knowledge of the author, facial chirality analysis can be identified as a new approach of automated deception detection.

1.7.2 Technological Contribution

In order to develop the proposed solution in automating deception detection, author will be providing a refined dataset suitable for the system. A model which specially identifies facial chirality of a suspect and using that facial chirality information and human facial cues to detect deception will be done in the system. Facial Chirality model will be newly introduced hence it will be the main technological contribution of the research.

1.8 Research Challenge

The main challenge of the research can be identified as obtaining the required domain knowledge in order to identify the problem background and analyse difficulties in the currently available deception detection techniques used in the interrogation process. Information on how deception detection is currently handled and the limitations of those systems were gathered for the identification of the research gap which was challenging for the author. Facial chirality analysis is very new to automated deception detection, therefore author faced challenges in finding out the best approach for facial chirality analysis. Author has to receive the domain knowledge from different sources to have a better understanding on the approaches and techniques. Another challenge the author faced was finding a publicly available video dataset which is suitable for the research area. The refinements of the dataset and mapping the dataset with the research findings in order to build a novel solution was challenging.

1.9 Research Questions

1. What effect does deception detection have on interrogation process carried out in a criminal investigation
2. What is the contribution of the currently available automated deception detection systems to the investigation process?
3. How does human facial cues and facial chirality analysis identify comfort and discomfort of the subject in order to detect deception

1.10 Research Aim

The aim of this research is to design, develop and evaluate an automated deception detection system which acts as a deception detection detective that analyses human facial cues and facial chirality to identify subject's truthful or deceptiveness in order to detect deception in the interrogation process.

Furthermore, the outcome of the above stated project aim will produce more accurate and reliable results in deception detection.

1.11 Research Objective

Research Objective	Explanation	Learning Outcome
Problem Identification	Identify a domain which is relevant to analyse significant research problem in this phase	LO1
Literature Review	Analyse extensive information related to following techniques of deception detection. ● Analyse deception detection to gain more knowledge on the domain. ● Identify limitations, strengths and future research gaps on the existing automated deception detection systems. Identify available algorithms, technologies and other tools which are required for the development phase of the project.	LO4

Data Gathering and Analysis	Gather requirements related to the following areas of the research ● Interview detectives to identify the problems in current deception detection system and the necessity for an automated deception detection system ● Identify the necessary features for the user-friendly GUI that needed to be developed to interact with the detectives Gather feedback from domain and technical experts for the further improvements of the system	LO2, LO7
Research Design	Design the proposed solution ● Develop the facial identification algorithm ● Design a model to extract facial cues and identify facial chirality ● Design the model for deception detection from the results of identified facial cues and facial chirality6 Identify testing and evaluation plans required in the latter part of the project	LO5
Implementation	Develop the proposed deception detection system based on the identified project resource requirements. Develop the facial cue extraction and facial chirality analysis model Develop the deception detection system Develop a user-friendly graphical user interface for the detectives' use	LO5, LO6
Testing and Evaluation	Test and evaluate the developed system ● Evaluate the proposed system with the manual deception detection system ● Evaluate the proposed system with the existing automated deception detection systems and benchmark against performance and accuracy Evaluate the performance against requirements	LO8

Table 1 - Research Objectives Table

1.12 Project Scope

This proposed project will be a research effort towards automating the deception detection process by analysing the facial cues and facial chirality to identify the comfort and discomfort of the subject to detect deception. Important and core features have been defined as the project scope.

1.12.1 In-Scope

- Human facial feature recognition
- Identification of comfort and discomfort of the subject by facial cue and facial chirality analysis
- Detect deception by analysing the comfort and discomfort level of the subject for each interrogation question
- User-friendly graphical user interface for the detectives to detect deception

1.12.2 Out-Scope

- Take other verbal and non-verbal communication into account such as human posture analysis, vocal variations, cognitive complexity and perception management in order to increase performance
- Make the proposed system a real-time deception detection system

1.12.3 Prototype Feature Diagram

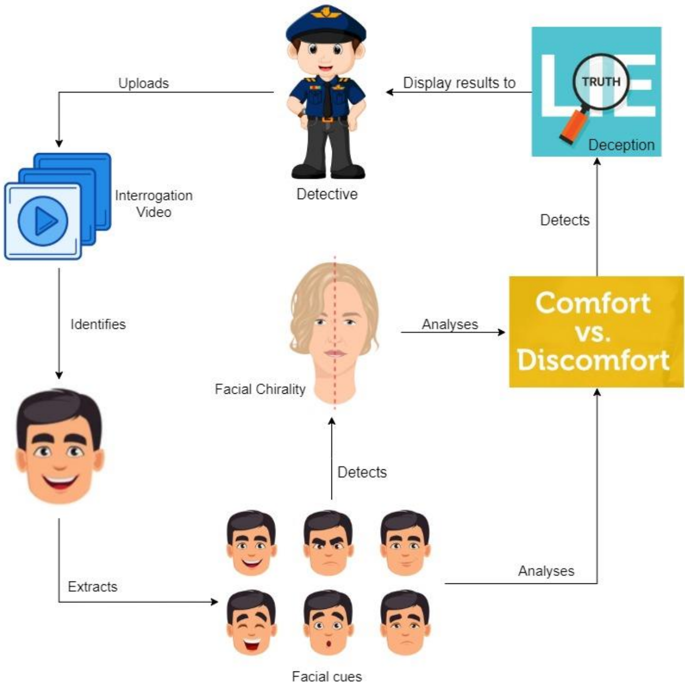

Figure 1 - Prototype Feature Diagram (Self-composed)

1.13 Resource Requirements

Identified resource requirements of the proposed solution according to the research deliverables are as follows. Resource requirements can be categorized into hardware requirements, software requirements, skill requirements and data requirements.

1.13.1 Hardware Requirements

Device	**Specifications**	**Alternatives**
Laptop	OS – Windows 64 bit Core i7 or higher 16GB RAM or higher Disk space of 80GB Nvidia GEFORCE RTX3050	MacOS, Ubuntu 64 bit

Table 2 - Hardware Requirements Table

1.13.2 Software Requirements

Requirement	**Alternatives**	**Purpose**
Operating System Windows 64 bit	Mac OS Ubuntu	Implement project solution and process data for the deep learning approach
Python	R	Algorithm and model implementation
PyCharm	Atom Jupytor Notebook	To make the development efficient
Scikit Learn	TensorFlow	Implementation architecture and development
Mendeley	Zotero	Manage, organize research papers and generate bibliographies for scholarly articles
ReactJs	AngularJs	Develop front-end of the system
Ms Word	Google Docs	Documentation purposes
Google Drive	One Drive	Documentation and project file backup
GitHub	Bit bucket	Project maintenance and update new development versions

Table 3 - Software Requirements Table

1.13.3 Skills Requirements

Skills	Usage
Programming skills	Development of the prototype
Domain knowledge	Understanding of the domain for the research purpose
Mathematical Knowledge	Understanding of mathematical relations to AI concepts
Machine Learning Skills	Understanding and Development of the models and architecture
SVM / CNN Algorithmic Knowledge	Development of new algorithmic approaches to find the best fitting algorithm for the project
Analytical skills	Identification of the research gap, analysis of performance and accuracy levels of the gap

Table 4 - Skill Requirements Table

1.13.4 Data Requirements

Real life trial dataset and FACS coded an emotion labelled human facial dataset for the facial cue and facial chirality extraction and analysis.

1.14 Chapter Summary

The first chapter which is the introduction chapter conveys a vivid picture of the project. This explains the domain of the problem, project background, the research aim and objectives of the completion of the project. Author explains the research gap identified and how the author will be contributing to the research domain as well as the field of software engineering in order to fill the identified research gap. The next chapter will be Software Requirement Specification chapter which includes the requirement elicitation methods and findings as well as the functional and non-functional requirements.

CHAPTER 2: LITERATURE REVIEW

2.1 Chapter Overview

The chapter reiterate the gathered information from existing research articles and publications to justify the improvements, approaches, techniques and technologies of the research topic. This chapter is a great impact to identify and understand the improvements and limitations of existing researches to identify the best approach to implement the proposed solution of deception detection system for the mentioned problem in the foregoing chapter.

2.2 Concept Map

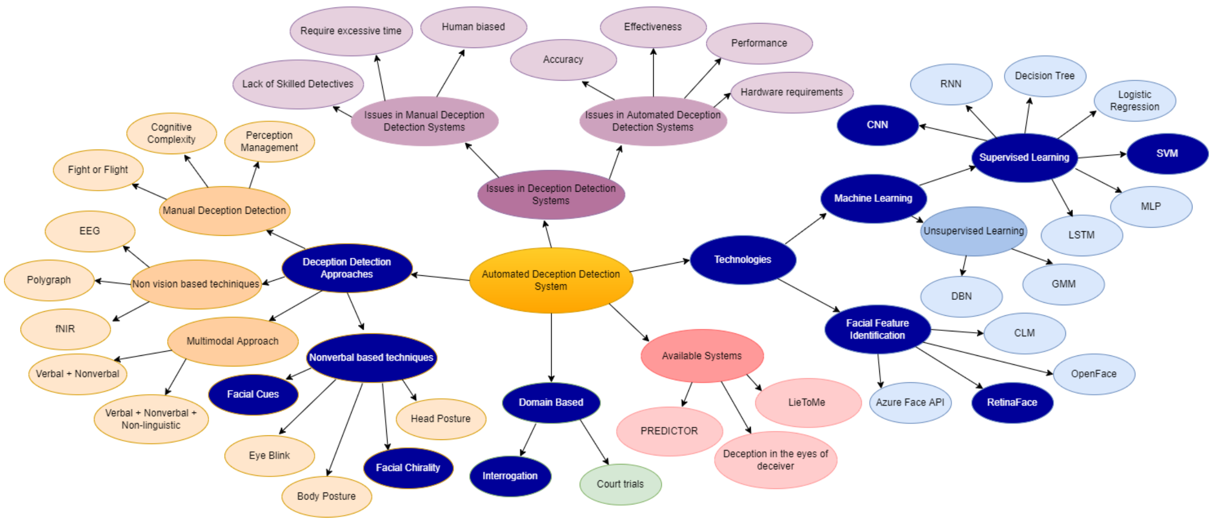

Figure 2 - Concept Map (Self-composed)

2.3 Background

2.3.1 Introduction to Crime

Crime is defined as "the intentional commission of an act usually deemed socially harmful or dangerous and specifically defined, prohibited and punishable under criminal law" stated in the (Crime | law, 2021). Crimes can take place anytime, anywhere in this world. There are four main types of crime which are felonies, misdemeanors, inchoate offenses and strict liability offenses (Criminal Law, 2021). Felonies such as robbery, rape, child abuse, murders and drug dealing have been rapidly expanding and spreading all over Sri Lankan society with the current crime wave. "Crimes which are differed to each society has a big impact on contravention of the collective conscious that helps the positive existence in the society" stated (Jayathunga, 2011). Mostly the violent crime impact on human lifestyle is major, that it makes citizens question the safety of the surroundings they live and the threat to their lives. Crimes can be identified as a major norm of regulating development in a country. According to Sri Lanka 2020 Crime and

Safety report, recent crime statistics stipulate an upward trend for serious crime in the country overall (Working Together to Protect U.S. Organizations Overseas, 2021). Criminal investigation processes are needed to take necessary actions to solve these crimes and punish criminals by the law. The court (1000 Culprits Can Escape but, One Innocent Should Not be Punished: Madras H.C Raises Questions on Perfunctory Investigation | Lawsisto Legal News, 2021) stated *"One of the basic principles of the criminal justice system is that the benefit of doubt must always be extended in favor of the accused. 1000 culprits can escape, but one innocent should not be punished".* Therefore, criminal investigations must carry out well in order to punish the guilty.

2.3.2 Criminal Investigation

According to (Criminal investigation, 2021) “Criminal investigation is an ensemble of methods by which crimes are studied and criminals apprehended”. Criminal investigation is a complicated and difficult task. There are many aspects to consider when examining the crime scene, the methods of deduction and what constitutes a crime. The evidence examination was done firstly to identify the key details of the crime such as fingerprints, motive and weapons in order to determine which of these pieces of information is most important throughout the investigation. Once the evidence is gathered and the suspects are identified, investigative interviews are carried out. Investigative interviews are one of the key processes carried out in the criminal investigation. These are done mainly for the accused, the accuser and witnesses. Goal of these investigative interviews is to gather accurate, reliable and actionable information. Investigative interviews consist of two sections which are non-confrontational interviews and interrogations. Confrontational interviews are conducted to gather objective facts and information by questioning with open ended questions while allowing the eight witnesses to supply evidence. Interrogations are designed to extract confessions from the suspect to the crime when the law enforcement already have gathered evidence (Police Interviews versus Interrogation: ‘Fairbanks Four’ Case Illustrates Important Distinction - Innocence Project, 2021). This study focuses on the suspect interrogation process of the criminal investigation.

2.3.2.1 Suspect Interrogation

Interrogation may seem like an intimidating venture for the uninformed; however, interviewing suspects is no different than any other facet of life involving interpersonal contact. If one is able to maintain their composure throughout the process, they will be able to achieve success with relative ease. Despite popular belief, there are no hard and fast rules concerning how to interrogate a suspect; it all depends on what would work best in each situation. Usually, suspect interrogations are carried out in interrogation rooms where a single detective or two detectives interrogate the suspect while another two detectives examine the suspect's behaviour and facial features from the outside in order to identify the truthfulness and deceptiveness of the suspect's behaviour. This is one of the approaches of manual deception detection done by the highly trained skilled detectives in suspect interrogation.

2.3.3 Deception Detection

Deception detection can be defined as "the investigative practices used to determine a person's truthfulness and credibility" according to (Deception Detection | How to Tell If Someone is Lying, 2021). As human beings, sometimes people think human actions never make mistakes. Individuals feel like they are always in control of what happens and what is going on around them. But even the most cautious individual can be influenced by their surroundings. In particular, these individuals can become targets for deception with psychological tactics such as guilt-tripping or manipulation. In order to prevent people from being manipulated by these approaches, investigators must have a set of tools that they can use when interviewing possible suspects. In this way, suspects can be questioned about their stories in order to determine if they are being deceptive. Deception detections is used on many occasions in law enforcements. They can be identified as suspect interrogations in criminal investigations, testimonies in court trials and alibies in investigative interviews. Accordingly, deception detection plays a crucial role in these domains.

2.3.3.1 Manual deception detection process

Investigators manually detect deception of the suspect using three main approaches. They can be identified as fight or flight, cognitive complexity and perception management. According to former FBI agent's explanation (WIRED, 2021), fight or flight is noting the physiological changes that occur when someone's under stress. What happens in fight or flight is that the suspect's primitive brain wants to either stand-up and fight or run away from the situation. Therefore, body provides extra energy to the limbs and that makes people do unnecessary body movements, grooming gestures and gross body movements. Accordingly, the human body posture is analysed as an important indicator of deception detection. Cognitive complexity means when people lie, usually they try to keep it as simple as possible. They typically lie only about two senses which are what they hear and what they see. They don't add spatial relationships or interactions with other people. Therefore, when the questions are asked deeply after drifting down, lies fall apart. Perception management is when a suspect tries to behave in a way, they think the interviewer would expect from somebody who is giving a truthful statement. For example, unusual stillness can be identified as an indicator that they are lying. Verbal, nonverbal and non-linguistic verbal communication are the key indicators of deception detection. To capture deception in verbal communication in English language, detectives focus on the words such as like, sort of, almost, kind of which depicts that the teller does not have confidence on what they deliver. Verbal indicators are different from language to language therefore it is hard to have a universal approach to detect deception using non-verbal communication indicators. Non-verbal indicators include facial feature changes, emotions or body movements. Non-verbal indicators are more reliable as they are universal and don't change with caste, culture, creed, religion and region. "At any particular instance of time, the emotion felt by a person can only be deciphered through the expression put up by that person" (Krishna et al., 2014). Non-linguistic verbal indicated by the increase in the pace of talking, increase or decrease in the volume and a change in pitch. When someone is telling facts they must simply state them but when they are telling a lie, they have to convince the people that what they are about to say is the truth. This makes people put extra effort when they are lying. All of the above-mentioned techniques are manual ways of detecting deception using highly trained and skilled detectives.

2.3.3.2 Issues in Manual Deception Detection

Manual deception detection approaches require excess amount of time and skilled detectives specialized in deception detection. Due to high crime rates and a limited number of detectives, the time allocated to each case has become limited. Also, there are very few numbers of detectives who are specialized in deception detection. Most importantly human decisions of deception detection are human biased. The accuracy level of manual deception detection has the average of 54% (Burgoon, 2015) which is only slightly above from the decision of a coin toss. Because of these limitations, deception detection has not been carried out well in the interrogation process which leads to inaccurate end results.

2.3.3.2 Automated Deception Detection Process

As solution for this problem researchers have identified automated deception detection systems. These automated deception detection systems can be categorized into three approaches which are facial and behavioural analysis approach, physiological phenomena analysis approach and multimodal approach. Author will be focusing on the facial and behavioural analysis approach in this research. These facial and behavioural analysis approaches are based on different aspects of the human behaviour. Some of them are human facial expression analysis, facial action unit analysis, blink rate analysis, emotion analysis, eye gaze analysis, body posture analysis. Mostly image processing and video analysis technical approaches are used to analyse above mentioned. Random forest algorithm, multiple instances learning algorithm, deep neural networks, convolutional neural networks, k nearest neighbor, transfer learning, recurrent neural networks are some of the algorithms and machine learning models used to the research projects based on facial and behaviour analysis for deception detection. Author will discuss in detail on the automated deception detection approaches, the algorithmic approaches and models used in the next few topics.

2.4 Automated Deception Detection Approaches

Critical evaluation of the existing automated deception detection approach based similar researchers will be explained in this section of the literature review. Researchers have identified different approaches for automated deception detection. Author will state the improvements and limitations of the currently available research to justify the novelty and uniqueness of the proposed solution. Deception detection research areas can be mainly divided into three types based on the approach. They are,

1. Nonverbal indicator analysis approach
2. Physiological phenomena analysis approach / non-vision-based techniques
3. Multimodal approach

2.4.1 Physiological Phenomena Analysis Approach

Researchers have done physiological phenomena-based research around the area of deception detection. Polygraph test, EEG test, fNIRS test can be identified as three main research areas under physiological phenomena-based deception detection systems. The most used system can be named as Polygraph test. The polygraph test detects deception by taking physiological factors such as blood pressure, heart rate and respiration levels into consideration. Polygraph tests are used for detecting deception in interrogations but due to the identified limitations, validity of the polygraph tests is in question. Limitations of the polygraph test can be identified as follows,

- Sensitive to health issues
- Indicate emotions, not the knowledge of deception (Lakshan et al., 2019)
- Nervousness affects the accuracy of the test results
- Requirement of sophisticated equipment setup
- Trained specialists are needed to operate the equipment and perform the test

Due to above mentioned limitations, law enforcement has not yet considered polygraph test as a credible deception detection technique.

EEG test is another system used to detect deception. It is a "test that detects electrical activity in your brain using small metal discs named electrodes attached to your scalp" according to (EEG (electroencephalogram) - Mayo Clinic, 2021). These experiment neuron signals captured using EEG machine has been analysed to determines if a subject is telling a lie or truth along with their emotional state and the attention level. (Lakshan et al., 2019). **PREDICTOR** is an EEG test-based research done to detect deception detection (Lakshan et al., 2019). This system consists of three main parts of detection which are lie detection, emotion detection and attention detection. In order to capture EEG records, researchers have used MUSE 2. EEG was captured in all the states of lie state, emotion state and attention state in order to perform a classification algorithm to detect deception using the EEG records. Classification algorithms used can be identified as, binary classification which is used for lie detection and attention detection and multi-class classification which are used to classify emotions. K Nearest Neighbors, Logistic Regression, Convolutional Neural Network and Random Forest are used to identify best-separated classes in the research. Random Forest classification algorithm is chosen for the classification task as the accuracy level is above 80% according to the study. Even though this system has a high accuracy, there are some limitations identified in the study when applying to the real-world interrogation process. The limitations are,

- Suspects should be free from health problems as the system acts differently for people with health issues
- Suspects should avoid movements and excessive facial muscle movements throughout the process
- Suspects should wear the MUSE 2 headband throughout the process

Due to these limitations, practicality of this system PREDICTOR is still in question.

fNIRS can be identified as another deception detection system under physiological phenomena analysis approach. Functional near-infrared spectroscopy known as fNIRS is an optical neuroimaging tool that can provide the quantitative hemodynamic information including oxyhemoglobin (HbO) and deoxyhemoglobin concentration changes (Zhang et al., 2021). Main advantages of the fNIRS are, fNIRS can be operated in a portable, comfortable and quiet way with fewer body constraints. In this system, fNIRS was used to inspect the neural mechanism of deception by using the measure of power, built based on the Welch power spectrum estimation. Welch algorithm was used as it can break down the original signal into L overlapped segments. Study has identified that the natural deceptive behaviours exhibited significantly higher power than the controlled behaviour. The analysis results demonstrates that the power can be an effective neuromarker to reveal the complex neural mechanism associated with deception. But the limitations of this approach can be identified as follows,

- Act differently on people with health issues
- Require sophisticated equipment setup
- Require trained operators to use the equipment

Due to these limitations this system's accuracy and the validity has not been proven with better results. After the evaluation of the improvements and limitations of the systems under physiological phenomena-based approach, author has decided to go ahead with a more accurate approach than the physiological phenomena-based approach for deception detection.

2.4.2 Multimodal Approach

Detectives focus on verbal, non-verbal and non-linguistic communication indicators of the suspect when detecting deception manually. Multimodal approaches are made to ease the deception detection process by analysing more than one deception indicator considered in manual deception detection. Most of the multimodal approaches have combined verbal and non-verbal communication indicators. Intention of the creation of the multimodal approach is to increase accuracy, applicability and generalizability. In the research (Belavadi, et al., 2020), researchers have proposed a multimodal deception detection system which is accurate, applicable and generalizable. This system explores the feasibility applying AI/ML techniques to detect lies in video using multiple datasets. The research was conducted to examine if deception can be caught using multimodal aspects such as facial expressions and movements, audio cues, video cues. Three different datasets with varying degrees of deception were used to experiment the feasibility of this multimodal approach.This study has used OpenFace API (OpenFace API Documentation - OpenFace API Docs 0.1.1 documentation, 2021) for the facial feature extraction. OpenFace API returns a CSV file which contains rich data regarding the location of the facial landmarks, the confidence of the face recognition algorithm, gaze, head pose tracking and the action unit data for 18 action units when the video is given to the API. Another feature considered was Eye Aspect Ratio (EAR) which provides a simple and elegant approach to tracking eye dynamics. CNN method was applied for deception detection using video frames extracted with ffmpeg. The study has experimented two transfer learning techniques which are feature extractor and model finetuning and the better accuracy was provided from model finetuning. But this research study has identified considerable number of limitations. According to the study using a multimodal data for deception detection does not appear to be generalizable and multimodal approach has low performance. The accuracy was increased due to overfit to the given dataset and that may not generalize to other datasets according to the study. Hence new data will provide a low accuracy to the system. In conclusion, this research study reveals that there is a lack of reasonable evidence that AI-based deception detection is generalizable over different scenarios of deception such as deliberate deception, deceptive under duress and being deceptive through half-truths.

Another research area of multi-modal deception detection approach can be identified as The Truth and Nothing but the Truth: Multimodal Analysis for Deception Detection (Jaiswal, Tabibu and Bajpai, 2016). This research study is based on a data driven method for automatic deception detection in real-life trial data using visual and verbal cues. OpenFace API was used to facial action unit recognition and to analyse the movements of the facial features. Eyebrows, eyes and mouth changes were mainly focused on the visual feature extraction. Lexical analysis was performed on the spoken words, emphasizing the use of pauses and utterance breaks. Bag of words model was used for verbal feature extraction. Audio feature extraction was done using OpenSMILE which extracts basic features like Mel-frequency coefficient, harmonics-to-noise ratio, jitter which is a proven indicator of nervousness, which is one of the categories of lying. SVM was used to feed all data to test deceit or truthful prediction. Limitations of this multimodal approach can be identified as follows,

- Acoustic feature extraction performed the worst - low accuracy as most of the videos accounted for more than one person speaking
- These inaccurate data lead to false results when combining visual and acoustic features in multimodal approach
- Vocal features cannot be considered universal due to language differences

Due to limitations found out in multimodal approach, author has decided to go ahead with a single approach which will be helpful in detecting deception in a more accurate and a reliable way where the researcher will be considering universal usage of the system.

2.4.3 Nonverbal Indicator Analysis Approach

This approach focuses on a single aspect of deception detection where only one of the deception indicators are analysed in order to detect deception. Studies based on non-verbal indicator analysis solely based on facial features. The reason can be identified as the facial features are universal and do not change with caste, creed, religion and region (Krishna et al., 2014). Facial features play a prominent and resolute role in people's daily lives. The exact emotions felt by a person depicts from that person's facial features without him/her being able to control that. Even when people try to hide the real emotion, they felt by covering it up with forced facial features, the real emotions appear in the face. That is when facial chirality can be detected. Chirality can be simply defined as asymmetricity where it looks like it is going to fold over perfectly but in fact it does not. Facial chirality happens when the face depicts more than one emotion or facial expression at a single time. Specially, when someone forces a facial expression in their face, but the real facial expression also appears automatically. That is why facial features have identified as a prominent factor in the process of deception detection. Researchers have done studies focusing on the facial features in order to detect deception.

This research can be named as a facial expression-based imagination index and a transfer learning approach to detect deception (2019 8th International Conference on Affective Computing and Intelligent Interaction (ACII), n.d.). This is a framework build to automatically differentiate between imagining vs remembering associated facial expressions while answering a question. Transfer learning approach was used for the process. OpenFace API was used to analyse the facial expression of the witness. Facial Action Coding system was used to extract FAU features. The feature affective extracts and analyses affective features like joy, fear, disgust, sadness, anger, surprise, contempt, valence, engagement, smile. Contribution of the research can be identified as follow,

- Development of a novel facial expression-based imagination index to distinguish the level of imagining vs remembering using neural network
- Introduction of a transfer learning-based framework for applying the imagination index for deception detection in an interrogation based dyadic communication game

Baseline Knowledge Model was created using two LSTMs differentiate between imagining vs remembering. Weighted Fusion Model and Support Vector Machine was used as the Deception Detection Models. This model was able to achieve an accuracy of 60%. The human performance for the same task is 51% so the model does not contain a better accuracy which was identified as the main limitation of this system.

Deception in the eyes of deceiver: A computer vision and machine learning based automated deception detection (Khan, et al., 2021) can be identified as another research study based on non-verbal indicators for deception detection. This study proposes a real-time deception detection approach which was developed utilizing advanced computer vision and machine learning approaches to model the non-verbal deceptive behaviour. Focus of this research is to identify the fine-grained eye movement and facial micro-movements which are significant clues for the automated deception detection. Different classification algorithms which are artificial neural networks, random forests, support vector machines and multi-layer perception were used to classify the deceptiveness and truthfulness of the video's each timeslot. Eye movements are significant compared to facial micro- movements according to the clustering-based attribute analysis in the study. Silent Talker system was used for feature extraction and micro gesture interactions to identify the deceptive behaviour. HAAR Cascade algorithms was used to encode local appearance of objects using haar like features and haar functions. The mentioned limitation of the study can be identified as the necessary improvement of the classification accuracy in order to get better results for the deception detection task.

Automatic Deception Detection in RGB videos using Facial Action Units (Avola, Cinque, Foresti and Pannone, 2019) can be identified as another automated deception detection research. The dataset of the research contains RGB video frames which include subject interviews. Facial action unit extraction and classification was done to identify the truthfulness and deceptiveness of the subject. This method extracts facial action units automatically from RGB video frames. The algorithm used to classify the extracted facial action units is Support Vector Machine algorithm. Classification is used to identify the deceptive or truthful behaviour. Experiments were performed on real court trials data to reveal the effectiveness of the system. Convolutional Experts Constrained Local Model (CE-CLM) is used for facial landmark point detection. The individual appearance of each facial landmark was detected by CLM local detectors and constrained optimization shape model. Main limitation of the system can be identified as lack of robustness of the system.

This research study can be identified as the First Attempts in Deception Detection in HRI by using Thermal and RGB-D cameras (Iacob & Tapus, 2018). HRI also known as Human Robot Interaction is a natural and an efficient robot which has the ability to understand the messages sent by its human interlocutors. This process is expeditious but accurate as a human being. The robot is able to comprehend the nonverbal and paraverbal message components sent by humans. This ability has become obliging to identify the honesty and deceptiveness of a person by evaluating the physiological states of the person. Thermal and RGB-imaging are the non-invasive techniques used for the process. This study focuses on developing non-invasive techniques to allow HRI to detect deception based on physiological manifestations. Nonverbal cues which are the position of eyes and the eye behaviour, head posture and orientation can be identified as some of the physiological parameters used. The cardiovascular and respiratory were also taken into consideration when detection deception. Non-invasive sensors which are RGB-D and thermal cameras were used to monitor the physiological parameters. The system has used ROS packages to acquire video streams. A program developed in C++ using ROS and OpenCV was used to process the videos. CLM face tracker to track 66 facial landmark points. The aim of this research is to develop lie detection techniques which are purely non-invasive and can be integrated in HRI easily without interfering with the interaction fluidity. The identified limitations of the study are as follows,

- Perceived level of stress during the interrogation was slightly above average
- Lower accuracy of the non-invasive measurements
- Measurement errors induced by the measurement techniques and the hardware limitations
- Errors due to unrestricted natural movements

Deception Detection by 2D-to-3D Face Reconstruction from Videos (Ngô et al., 2021) is another research study in the area of deception detection. This research proposes a novel method which extracts reliable facial features, head postures, expressions illumination of the video, skin reflectance to achieve a convert, reliable and non-invasive deception detection. Recurrent Neural Network was used to model the proposed solution. The dataset used for the system was the Real-life Trial (RLT) dataset which contains truthful and deceptive video recording in a high skate courtroom. The limitations of facial action units can be identified as error prone due to environmental condition such as viewpoint, illumination and occlusion. This research works on reconstructing the face to get an effective and a reliable deception detection. The proposed face reconstruction method is a fully unsupervised end-to-end deep architecture. Results reveal that the proposed deception detection method has improvements than the existing state of art methods also this method outperforms manual deception detection using facial attributes. Development of the system was done using RNN to capture temporal relations between facial representation of each video's frame. OpenFace was used to extract facial features. Decision Tree and Random Forest classifies were used to classify the facial features with mentioned default parameters. CNN was used for time series classification. All together the RNN constructs a powerful baseline with the use of the facial features. Limitation identified in the study is the low accuracy which could have been better in detecting deception as it is a crucial task.

LieToMe is an ensemble approach for deception detection from facial cues (Avola et al., 2020). This research proposes an RGB video-based deception detection approach which focuses on facial features and stack generalization ensemble. Both static and dynamic facial characteristics have captured as face presents meaningful deception cues. System was implemented and tested based on two datasets which are real-life trial dataset and the bag of lies dataset. The effectiveness and performance of the proposed method has shown improvements on the real-life trial dataset. Base-level classifiers were used to analyse different facial feature combinations and the ensemble technique was used to stack into a more robust meta-level classifier. Dynamic Ensemble Learning (DEL) method was used for the image registration and image type classification which are heterogeneous tasks. OpenFace was used to detect, align and mask the face of the subject. Histogram of oriented gradients (HOG) and local binary pattern (LBP) were used to extract the micro and macro expressions. SVM, logistic regression, extreme gradient boosting, decision tree and MLP were classifiers tested in the proposed method. Out of these classifiers, MLP has resulted in a more robust meta-level classifier. Future enhancement of the system is to test automatic feature extractors like 3D CNN to improve the performance by comparing available features directly. Except for lack of performance, the proposed system has made a great impact on the automated deception detection.

Different researchers have carried out different researches in the area of deception detection to have an accurate, reliable and efficient automated deception detection system. These researches have made impacts on deception detection in positive ways. This study is to introduce a new deception detection approach by specifically considering the facial chirality of a person to identify the comfort and discomfort of a person when detecting deception.

2.5 Technology Comparison

Automated deception detection approaches have used different tools and technologies in detecting deception. The technologies used can be broken down into different sections according to the use of the tools and techniques. Supervised, unsupervised and semi supervised machine learning algorithms can be used for the classification purpose of the system. SVM, Logistic regression, Decision trees, CNN, MLP, RNN, LSTM, DBN, GMM can be identified as some of the classification algorithms used for similar systems. Facial feature identification and analysis process is another key process in the system. OpenFace, Azure Face API, CLM was used for facial feature recognition, analysis processes. Detailed descriptions of these algorithms and toolkits are stated below.

2.5.1 Machine Learning Algorithms

2.5.1.1 Support Vector Machine

Support Vector Machine also known as SVM is a supervised machine learning algorithm which is used for classification, regression problems (1.4. Support Vector Machines, 2021). Most of the automated deception detection systems have used SVM for the classification purpose. There are some advantages as well as limitations found in SVM (SVM | Support Vector Machine Algorithm in Machine Learning, 2021).

Advantages	**Disadvantages**
Effective in high dimensional space	Do not directly provide probability estimates
Memory efficient	Complexity
No distribution requirements	Memory intensive
Versatile	
High Accuracy	

Table 5 - SVM Advantages and Disadvantages

2.5.1.2 Logistic Regression

Logistic Regression is a supervised machine learning algorithm which is used for the classification problem. This algorithm is a predictive analysis algorithm based on concept of probability. Logistic sigmoid function was used to return a probability value.

Advantages	**Disadvantages**
Easy implementation and interpretation	Applicable only for linear solutions
Fast training	Assumes the input residuals to be normal
Space complex solution	Assumes input features to be mutually independent
Many ways to regularize the model	Limited expressive power
	Hard to make incremental

Table 6 - Logistic Regression Advantages and Disadvantages

2.5.1.3 Decision Tree

Decision tree is a supervised machine learning algorithm used for classification purposes. It is one of the prominent and early discovered algorithms. The data classification of decision trees outcome a tree like structure.

Advantages	**Disadvantages**
Easy to interpret	Require excessive amount of time for large dataset training
Quick to learn	Complex calculations
Distribution requirements are not needed	Accuracy can vary a lot on presented data
Normalization and scaling of data are not required	Inadequate for regression application and continuous value prediction (Top 5 advantages and disadvantages of Decision Tree Algorithm, 2021)

Table 7 - Decision Tree Advantages and Disadvantages

2.5.1.4 Convolutional Neural Network

Convolutional Neural Network also known as CNN is a supervised deep learning algorithm used for data analysis and classification purposes. CNN algorithm has the ability to recognize, classify, segment and process images and it is designed specifically to process pixel data.

Advantages	**Disadvantages**
Computationally efficient	Do not encode object position and orientation
Less preprocessing dependency	Require large amount of training data
Easy to understand	Lack of ability to be spatially invariant to the input data (Difference between ANN, CNN and RNN - GeeksforGeeks, 2021)
Fast to implement	Slower due to some operations
Weight Sharing	Overfitting
Higher accuracy compared to image prediction algorithms (Convolutional Neural Network (CNN) questions, 2021)	
Human supervisions are not needed as the important features are detected automatically	

Table 8 - CNN Advantages and Disadvantages

2.5.1.5 Multi-Layer Perception

Multi-layer Perception also known as MLP is a supervised learning algorithm. It generates set of outputs from a set of inputs which can be identified as a feedforward artificial neural network

(What is a Multilayer Perceptron (MLP)? - Definition from Techopedia, 2021). MLP is a deep learning method (1.17. Neural network models (supervised), 2021). MLP classifier uses backpropagation for network training.

Advantages	**Disadvantages**
Ability to work well with large input datasets	Functioning depends on the quality of training
Can apply to non-linear complex problems	Too many parameters due to full connection
Fast predictions	Redundant
Less time for training	Inefficient
Same accuracy ratio can be achieved even with smaller datasets	Difficult computations (Akkaya and Çolakoğlu, 2019)
Adaptive learning	

Table 9 - MLP Advantages and Disadvantages

2.5.1.6 Recurrent Neural Network

Recurrent Neural Network also known as RNN are semi supervised learning algorithms. RNNs are a type of ANNs which are used in speech recognition and natural language processing. RNN can be identified as the state-of-the-art algorithm for the sequential data. It is a robust and powerful neural network (A Guide to RNN: Understanding Recurrent Neural Networks and LSTM Networks, 2021).

Advantages	**Disadvantages**
Ability to process lengthy inputs	Slow computation
Useful in time series predictor	Exploding and vanishing gradients
Weights are shared across time steps	Difficult in training
Model size does not increase even when the input size is large	Difficult in processing long sequences when using tanh/relu as activation functions (RNN), 2021)
Use internal memory for processing arbitrary inputs	

Table 10 - RNN Advantages and Disadvantages

2.5.1.7 Long Short-Term Memory

Long short-term memory also known as LSTM is a memory extension for RNN. LSTM is an unsupervised learning method. LSTM can be identified as building blocks for RNN layers.

Advantages	Disadvantages
Ability to learn from a large range of parameters	Require excessive time to train
Deal with vanishing gradient problem in RNN	Overfitting
Relative insensitive to gap length	Harder to implement dropout
Well suited for classification, processing, prediction of time series given time lags of unknown durations	Sensitive to different random weight initializations

Table 11 - LSTM Advantages and Disadvantages

2.5.1.8 Gaussian Mixture Modal

Gaussian Mixture Modal also known as GMM is an unsupervised learning algorithm specialized for pattern recognition. GMM is a clustering algorithm which can be used to perform soft clustering on query data or hard clustering

Advantages	Disadvantages
Powerful technique with a better accuracy	Too many parameters to fit
Do not require subpopulation which belongs a data point / Learn subpopulation automatically	Require high amount of data
Allows for most flexible identification	Better results need many interactions

Table 12 - GMM Advantages and Disadvantages

2.5.1.9 Deep Belief Network

Deep Belief Network also known as DBN is an unsupervised probabilistic deep learning algorithm. DBN is a generative hybrid graphical model. It is made of multi-layer of stochastic latent variables (Deep Learning—Deep Belief Network (DBN), 2021). DBN is used for image, video sequences and motion-capture data recognition, clustering and generation.

Advantages	Disadvantages
Cannot suffer from the training data fragmentation problem	Require complete and high temporal frequency datasets
Reduce the over smoothing problem	Generated speech quality is degraded
Require few parameters and low parameter uncertainty	Stationary DBN do not capture time-variable gains (Ockenden et al., 2017)

Table 13 - DBN Advantages and Disadvantages

2.5.2 Face Identification and Feature Extraction

2.5.2.1 OpenFace

OpenFace is an open-source toolkit which can be used to facial landmark detection, head pose estimation, facial action unit recognition and eye gaze estimation (Understanding Open-Source Facial Recognition Through OpenFace, 2021). FaceNet algorithm which is a unified embedding for face recognition and clustering was used for automatic facial identification in OpenFace toolkit (OpenFace, 2021). OpenFace allows for easy integration with other applications and devices through a lightweight messaging system (Baltrusaitis, Robinson and Morency, 2016).

Advantages	**Disadvantages**
Does not require many human resources	Sensitive to illumination
Impressive performance on the labeled faces in the wild benchmark	Difficulty in detecting faces in occlusion, different poses and complex backgrounds
Open source	
Better accuracy	
Capable of real-time performance	
Ability to run from a simple webcam without specialist hardware	

Table 14 - OpenFace Advantages and Disadvantages

2.5.2.2 Constrained Local Model

Constrained Local Model also known as CLM is a method used for facial landmark detection (Zadeh, Baltrušaitis and Morency, 2021). CLM generates the response images applying normalized correlation with a local patch, where the model patches are modified to fit the current face but constrained by a global texture model (Constrained Local Models, 2021).

Advantages	**Disadvantages**
Ability to locate facial landmarks	Inability to model complex individual landmarks
	Affected by illumination, expression, facial hair, makeup, and accessories.

Table 15 - CLM Advantages and Disadvantages

2.5.2.3 Azure Face API

Azure Face API detects, recognizes and analyses human faces in images using pre-trained machine learning models that have been developed by Microsoft. Azure Face APIs are used to create facial recognition systems (Transparency Note: Azure Cognitive Services Face API, 2021).

Advantages	**Disadvantages**
Advanced facial recognition	Only support still images
Easy to use	No built-in anti-spoofing countermeasures such as depth or motion detection
Secured	Can only be applied to darker skin tones
Privacy guaranteed	

Table 16 - Azure Face API Advantages and Disadvantages

2.5.2.4 RetinaFace

RetinaFace is a face detection tool with facial landmarks developed using python. RetinaFace can be used for facial detection and 2D and 3D facial alignment. This can be identified as a well performing tool.

Advantages	**Disadvantages**
Advanced facial detection	High storage requirement
Easy installation	
Detection of many faces at once	
Facial alignment feature	

Table 17 - RetinaFace Advantages and Disadvantages

2.6 Chapter Summary

Literature review chapter included the through explanation of the domain of deception detection following a critical evaluation of existing systems and research studies in the area of deception detection. This chapter was concluded with a technological comparison which consists of the tools and techniques which are currently being used for the process of automated deception detection. Author has created a concept map for the study to ensure all the key aspects are covered through the literature review. Domain based, currently available deception detection approaches, issues in deception detection systems, currently available systems for deception detection, tools and techniques were covered in the literature following the concept map. The next chapter the project management section will include the methodologies and approaches of the project **Caught in a Lie.**

CHAPTER 3: METHODOLOGY

3.1 Chapter Overview

This chapter details on the methodology followed in working on the project **Caught in a Lie**. Chapter covers the research methodology, development methodology, project management methodology in detail. Process model, design methodology, evaluation methodology will be discussed under development methodology. Project management method, deliverables, Gantt chart and risk management will be explained under project management methodology.

3.2 Research Methodology

Research Philosophy	**Pragmatism** research philosophy was used because it can integrate the use of multiple research methods such as qualitative, quantitative which are used in this research project. (Research Philosophy - Research Methodology, 2021)
Research Approach (Chetty, 2021)	**Deductive approach** was chosen as the best match for the project as abundant source availability, short time available to complete the study and to avoid risk.
Research Strategy	The author chose **quantitative surveys** to gather requirements. Also, **qualitative interviews** will be carried out for problem identification for requirement gathering phase and feedback gathering for testing and evaluation phase. Domain experts will be interviewed.
Research Choice	**Mixed method** was chosen as the research choice as the author uses both qualitative and quantitative data which will be gathered through quantitative surveys and interviews.
Time zone	**Cross sectional** time zone has been selected instead of Longitudinal, as the author will be gathering data at one specific point in time.

Table 18 - Research Methodology Table

3.3 Development Methodology

3.3.1 Process Model

Iterative process model will be used as the process model for the proposed project. The iterative process is suitable for this research project because requirements are defined clearly and easy to understand. This process also allows accessing previous phases, in which the changes are made accordingly. (Iterative Model: Advantages and Disadvantages |Professionalqa.com, 2021)

3.3.2 Design Methodology

As an analysis and design approach, the author selected the Object-Oriented Analysis Design. The OOAD approach was chosen because it will be helpful to the user in reducing the complexity of the system and developing the system in an effective and efficient way. This approach increases the readability of the code that was written. Basically, it provides an organized system to the user.

3.3.3 Evaluation Methodology

3.3.3.1 Confusion Matrix

The confusion matrix of this project includes classes which are identified as follows,

- Positive - Deceptive
- Negative - Non deception
- True Positive - Actually deceptive, detected deceptive
- False Negative- Actually deceptive, detected non deceptive
- False Positive - Actually non deceptive, detected deceptive
- True Negative - Actually non deceptive, detected non deceptive

3.3.3.2 Benchmarking

Benchmarking can be identified as comparing a system with specific industry and domain standards to test and identify objectives that are attained with quantitative metrics. The proposed system has better accuracy and performance compared to the existing systems. This system can be identified as the first to consider analysis of facial chirality to identify human comfort or discomfort in order to detect deception.

3.4 Project Management Methodology

3.4.1 Project Management Method

Project management method for this research project was chosen considering the factors which are the ability to take the risks, flexibility and timeline. PRINCE2 was chosen as the project management methodology. This is a project management which has the ability to suit any project. PRINCE2 is based on generic best practices which makes the projects flexible. PRINCE2 has the ability to take risks, it emphasizes on risk management therefore this approach is a great match for this research project. Due to the less time limit to complete the project, the time management aspect of the PRINCE2 approach will be useful.

3.4.2 Project Plan

Please refer **Appendix A** for Gantt chart and **Appendix A** for Work Breakdown Structure.

3.4.3 Deliverables

Deliverables	Date
Project Initiation Document (PID)	4th November 2021
Literature Review Document (LR)	21st October 2021
Software Requirement Specification (SRS)	25th November 2021
Prototype Version 1	6th December 2021
Interim Progress Report (IPR)	27th January 2022
Prototype Version 2	21st February 2022
Project Specification Design and Prototype (PSPD)	3rd March 2022
Test and Evaluation Report	17th March 2022
Final Project Report	28th April 2022
Final Viva	16th May 2022

Table 19 - Deliverables Table

3.4.4 Risk and Mitigation

Risk Item	Severity	Frequency	Mitigation Plan
Laptop Crash	5	5	Upload project documents frequently to the google drive, have a backup in an external hard drive and commit changes daily to git
Issues in meeting Domain experts for interviews due to covid situation	3	1	Arrange virtual meetings with the domain experts by explaining the situation
Wide domain knowledge to gain	5	1	Allocate time to understand the domain and define the scope according to realistic terms of domain
Dataset unavailability	2	1	Check the availability of an open dataset before starting the project
Incomplete work by the deadline	4	2	Time allocation and management, have a self-made deadline before the

			actual deadline

Table 20 - Risk Management Table

3.5 Chapter Summary

This chapter assisted the author in identifying the suitable methodologies in the areas of research, development and project management. Iterative model was chosen for as the process model, OOAD was chosen as the design approach of the project and PRINCE2 was selected as the project management methodology. Gantt chart was created to demonstrate the tasks and time allocation of each task separately. Finally, risk management was discussed to identify and reduce potential risks.

CHAPTER 4: SOFTWARE REQUIREMENT SPECIFICATION

4.1 Chapter Overview

The software requirement specification chapter mainly focuses on identifying the significant stakeholders of the system **Caught in a Lie** and the requirements and findings using selected requirement elicitation techniques. Author will also discover the functional and non-functional requirements in the system by exploring the gathered requirements. Additionally use case diagrams along with use case descriptions will be included in order to depict the system-user interactions. Finally, domain model and a summary of this chapter will be included.

4.2 Rich Picture Diagram

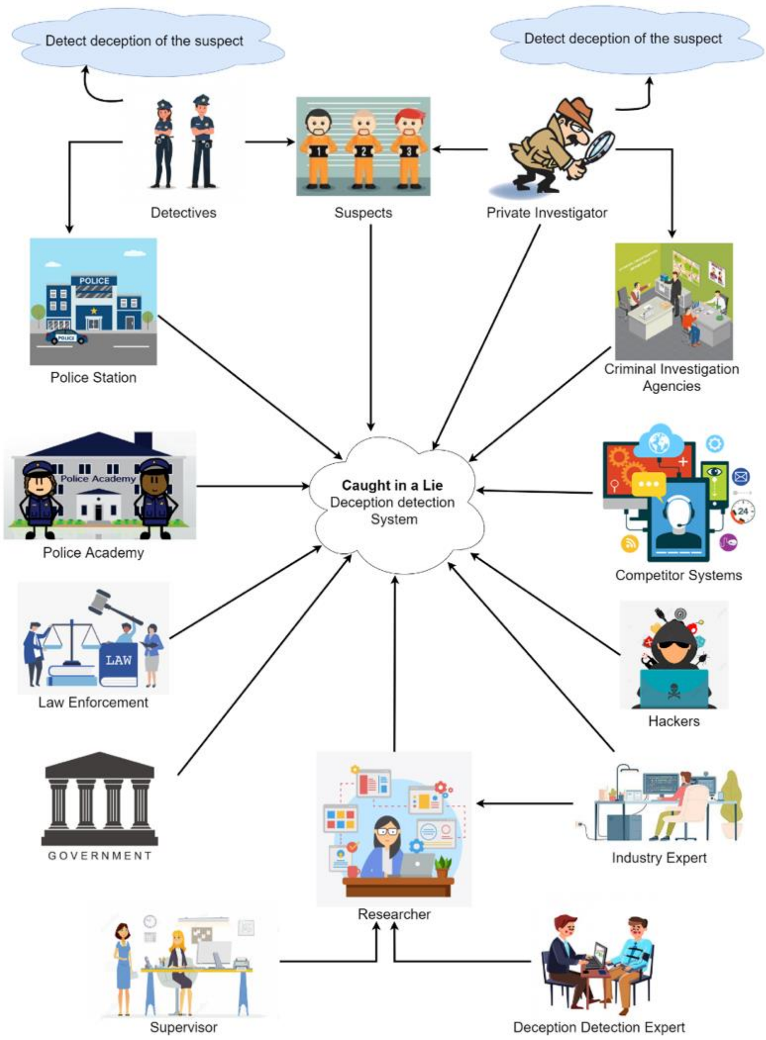

Figure 3 - Rich Picture Diagram (Self Composed)

Figure 3 illustrates the Rich Picture Diagram of the **Caught in a Lie** deception detection system. This includes the bigger picture of the system and system interactions with different parties.

4.3 Stakeholder Analysis

4.3.1 Onion Model

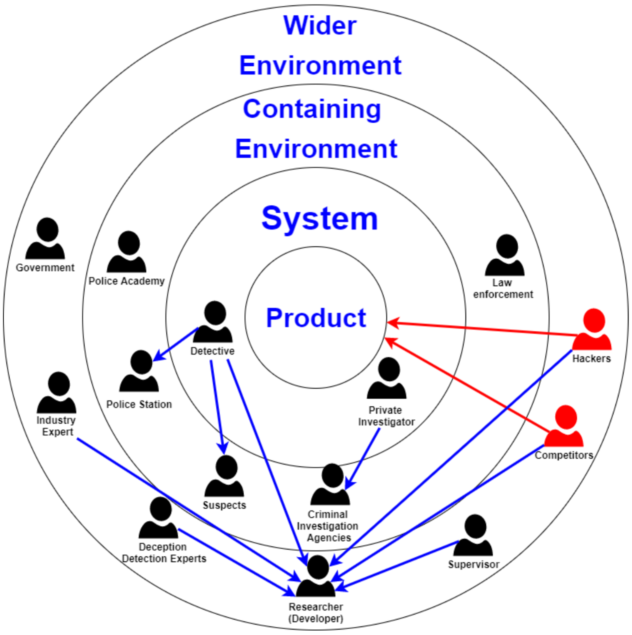

Figure 4 - Onion Model (Self Composed)

Above figure 4 depicts the Onion model of the **Caught in a Lie** deception detection system. Below table consists of the detailed descriptions of the stakeholder and the roles of the stakeholders.

4.3.2 Stakeholder Descriptions

Stakeholder	Role	Description
Detectives	Operational Beneficiary	Will use the system to detect deception of a suspect in an interrogation
Private Investigators	Operational Beneficiary	Will use the system for the deception detection process in interrogation carried out in the criminal investigation process
Police Station	Functional beneficiary	Will benefit from the system as the system will help in solving crimes efficiently and accurately
Criminal Investigation Agencies	Functional beneficiary	Will have the ability to use the system for their criminal investigation processes

Police Academy	Social beneficiary	Will aid police academy in training policemen in the learning process of deception detection to make them detectives skilled in detection deception
Suspects	Social beneficiary	Will be taken as the participants of the system and may want to confirm the truthfulness of their interrogation answers if the suspects are innocent
Law enforcement	Social beneficiary	Will get the suspect's interrogation results as the evidence for the criminal case where they can use the facts to sentence the guilty suspects and release innocent suspects
Researcher	Managerial Beneficiary	Does research on the domain in order to implement and improve the system
Deception Detection Experts	Expert	Will review the functionality of the system and help the researcher in identifying key aspects of deception detection
Industry Experts	Expert	Will give advice to the researcher in order to implement the system by choosing the most appropriate tools and techniques
Supervisor	Advisory	Supervise the researcher by guiding throughout the research process in order to propose the solution.
Government	Political Beneficiary	Will indirectly help to reduce the high unsolved criminal case rates in the country by spending less time on cases
Competitors	Negative stakeholder	Will look for the current system features and try to build a system with better performance and accuracy level
Hackers	Negative stakeholder	Will disrupt and damage the data and functionality of the system

Table 21 - Stakeholder Descriptions

4.4 Requirement Elicitation Techniques

4.4.1 Literature Review

Author will use the literature review requirement elicitation methodology to explore the similar systems in the domain and find out the similarities and differences by carrying out a comparison of the **Caught in a Lie** system and the similar systems. The goal of the literature review method is to identify the improvement and limitations of the similar existing systems and use the gathered information to improve our system. Literature review findings of the **Caught in a Lie** system reveals that law enforcement has not yet found a valid automated deception detection system.

4.4.2 Observation

Observations are carried out to observe the similar deception behaviors of the suspects specially to understand the key point which can be considered in the system. Detectives' manual deception detection techniques are observed to identify the features detectives take into consideration in detecting deception manually which can be used for the system. One of the limitations of the observations is that it is time consuming to observe from the scratch (Top 10 Most Common Requirements Elicitation Techniques, 2022).

4.4.3 Brainstorming

Author conducted brainstorming with different people in order to gather different perspectives and opinions of each individual. First the brainstorming was conducted with the supervisor to finalize the idea and select approaches, then with colleagues to get a general opinion on the system. Brainstorming session was carried out with a detective in order to understand the current crime states and manual deception detection approaches which gave more understanding of the domain. Limitations of this method is that due to different opinions and viewpoints, requirements might lead to contrary directions (Top 10 Most Common Requirements Elicitation Techniques, 2022).

4.4.4 Prototyping

Prototyping method will be used to design the understanding of the requirements in order to provide user better user experience. When prototyping is done in prior to the actual implementation it can be useful to get reviews from the stakeholders and improve the system accordingly. This was developer will be able to implement the system with less confusions and errors at the development stage.

4.4.5 Interview

Interviews are conducted with the detectives who will be the main users of the system and for the deception detection experts in order to get reviews of the system functionality. Detective interviews will be useful to get an understanding of the necessity of an automated deception detection system and the domain expert reviews will be useful for the improvements of the system. Limitations of this method can be identifying as interviews need more preparation and has to review question structure before the interviews.

4.4.5.1 Interview Designs

Interview with Police Officer	
Aim	• To identify how Sri Lanka police conduct interrogations • To identify the difficulties of current deception detection process • To identify the usage of automated deception detection methods
Interviewees	Mr. Rohan Wishwanath Minor Complaint Officer In Charge – Maharagama Police Station
Findings	• Sri Lanka Police doesn't have specific interrogation rooms for the interrogations • Police officers face difficulties in manual deception detection due to high time consumption and lack of specialized police officers • Automated deception detection methods have never been used by Sri Lanka Police

Table 22 - Interview Design 1

Interview with Criminal Investigation Department Officer	
Aim	• To identify the interrogation process done in CID • To find the current deception detection process in CID • To confirm the necessity of an accurate deception detection system
Interviewees	Mr. B. R. S. R. Nagahamulla Deputy Inspector General of Police - Former Director of CID
Findings	CID uses interrogation rooms and equipment for interrogations CID mostly does deception detection manually due to fewer technological facilities Accurate automated deception detection methods are necessary for CID in order to gather evidence and solve crimes at ease

Table 23 - Interview Design 2

Interview with Police Officer	
Aim	• To find the crime rates in Sri Lanka • To identify the availability of police officers for area coverage

	• To find the time allocations for interrogations in each case • To identify the importance of deception detection in interrogation process
Interviewees	Mr. Ashoka Mahinda Weerakkody Senior Superintendent of Police - Former Crime OIC
Findings	• Crime rates are high in Sri Lanka mainly in urban areas • Area coverage is difficult due to limited number of police officers • Interrogation time for the first time differs due to the type of case e.g. : Normal Case - 24 hours Drugs related Case – 72 hours • Deception detection is very important in interrogation process in order to find accurate evidence efficiently and solve the crime

Table 24 - Interview Design 3

4.4.5.2 Interview Question Design

Question	What is the current crime rate in Sri Lanka?
Aim of Question	To identify the current crime rate in Sri Lanka whether the crime rate is high or low and how it differs from area to area
Findings	Mostly the crime rate is high in urban areas of the country. This rate varies from 50% to 75% in different areas. There are different types of crimes which are violent crimes, thefts, burglary, fraud and altogether the rate of these crimes happening is high up to 75%

Table 25 - Interview Question Design 1

Question	Are there sufficient detectives to cover all areas and handle the cases without any delays?
Aim of Question	To find out the police officers' allocation to areas in order to solve the cases efficiently
Findings	One officer per thousand population is the current allocation in Sri Lanka if the population is considered. Urban areas have a high number of allocations while the rural areas have a low number of allocations. But according to the statistics, police officers are limited.

Table 26 - Interview Question Design 2

Question	What is the average time allocated for interrogation?
Aim of Question	To identify how much time can be spent on an interrogation
Findings	For normal case, maximum time suspects can be kept in interrogations is 24 hours and for drugs related cases, time is 72 hours to 7 days for severe cases

Table 27 - Interview Question Design 3

Question	How often are interrogations conducted in a criminal investigation?
Aim of Question	To find out the number of times interrogations are conducted
Findings	Police officers can conduct interrogations as many times as necessary. For direct cases there are fewer interrogations while complicated cases require many interrogations.

Table 28 - Interview Question Design 4

Question	How many detectives are required for a single suspect interrogation in order to interrogate while detecting deception?
Aim of Question	To identify the availability of police officers are enough for the interrogations
Findings	General police stations use maximum two police officers to interrogate suspects in normal cases while CID uses many police officers depending on the severity of the case.

Table 29 - Interview Question Design 5

Question	Is deception detection an important task carried out in a suspect interrogation? Explain the importance?
Aim of Question	To identify the importance of deception detection in suspect interrogation
Findings	Deception detection is an important task which most of the police officers fail to do due to lack of training and less time. Therefore, police officers try to skip deception detection and move on to evidence gathering from suspects' alibi which leads to inaccurate results

Table 30 - Interview Question Design 6

Question	How does deception detection help in solving crimes?
Aim of Question	To find the role of deception detection in criminal investigation
Findings	Deception detection helps in getting accurate information from the suspect which leads detectives to find accurate evidence efficiently and solve crime

Table 31 - Interview Question Design 7

Question	How is deception detected in the current interrogation process?
Aim of Question	To identify the current deception detection process
Findings	Currently Sri Lankan Police and CID both mainly use manual deception detection process

Table 32 - Interview Question Design 8

Question	Please explain the manual deception detection approaches used?
Aim of Question	To find out the manual deception approaches
Findings	Police officers focus on the facial expressions, body language and the features like confidence of the suspect and the corporation. Thorough analysis of these features helps in detecting deception.

Table 33 - Interview Question Design 9

Question	What is the validity of manual ways of detecting deception?
Aim of Question	To know the validity of manual deception detection
Findings	Deception experts' results can be identified as valid results but since manual deception detection results can be human biased the court does not consider them as valid results

Table 34 - Interview Question Design 10

Question	What role does nonverbal indicators play in deception detection?

Aim of Question	Importance of nonverbal indicators in deception detection
Findings	Non- verbal indicators are one of the most important factors in deception detection because these indicators hard to control by humans

Table 35 - Interview Question Design 11

Question	Are there detectives who are specialized in manual deception detection? Please specify in percentage.
Aim of Question	To find out specialized deception detection detectives are limited
Findings	Police stations do not have specialized detectives for deception detection while CID has specialized officers.

Table 36 - Interview Question Design 12

Question	How hard is it to detect deception manually?
Aim of Question	To know the difficulties in manual deception detection
Findings	Manual deception detection is so hard that only deception specialized detectives can do it accurately. But that can also be mistaken sometimes

Table 37 - Interview Question Design 13

Question	What automated deception detection approaches are used in Sri Lanka?
Aim of Question	To find the automated deception detection approaches in Sri Lanka
Findings	Currently neither Sri Lanka Police nor CID uses automated deception detection approaches

Table 38 - Interview Question Design 14

Question	Explain the effectiveness and validity of currently available automated deception detection systems
Aim of Question	Identify the effectiveness of automated deception detection systems
Findings	As automated deception detection systems are not available in Sri Lanka, the effectiveness of these cannot be discussed

Table 39 - Interview Question Design 15

Question	Will it be helpful to have an automated deception detection system which acts as a detective specialized in deception detection and help the interrogation detective by identifying deception through deceptive facial indicators?
Aim of Question	To identify the necessity of s system like **Lie To Me**
Findings	This will be very helpful and will make the interrogation process more effective and efficient. This system will help in carrying out an interrogation with a single police officer while assisting the detective to detect deception. This system will not violate any Human Rights Law as there will not be any physical contact with the suspect. System will be very helpful to the police officers.

Table 40 - Interview Question Design 16

Question	How useful will that system be in solving crimes accurately and efficiently?
Aim of Question	To identify the impact of the system for solving crimes
Findings	This system will help in getting truthful statements and alibis from the suspect which leads the police officers to find accurate evidence from the suspects' statements without wasting time from following deceptive leads. This will help police officers to solve crimes efficiently and accurately.

Table 41 - Interview Question Design 17

4.5 Summary of Finding

Findings	**Literature Review**	**Observation**	**Brainstorming**	**Prototyping**	**Interviews**
Crime rates are high in Sri Lanka	√	√			√

Criminal Investigation process is a time-consuming task which require number of detectives		√			√
Interrogation is a key task carried out in the criminal investigation process	√	√	√		√
Importance of deception detection in suspect interrogation and solving crimes	√	√			√
Current deception detection approaches used in Sri Lanka	√		√		√
How manual deception detection is carried out in the interrogation	√		√		√
Limitations of manual deception detection	√	√	√		√
Availability of expert deception detectors in Sri Lanka for the manual deception detection process	√	√	√		√
Usage of automated deception detection systems in Sri Lanka		√	√		√
Validity of currently available automated deception detection systems	√	√			√
Necessity of an automated deception detection system which will help detectives by acting as a specialized deception detector by identifying deception through deceptive facial indicators			√		√
Ability to consider the system as the first to take facial chirality indicator in automated deception detection		√	√		
System with better user interfaces for the detectives to use the system easily without confusion				√	
Usefulness of the system in solving crimes accurately and efficiently			√		√

Table 42 - Summary of Findings Table

According to the summary of findings, it is identified that Sri Lankan crime rates are higher and with the limited detective allocations and time allocations, investigation and interrogation processes have become difficult tasks. It has become a time-consuming task. Therefore, instead of the currently available manual deception detection system which lack validity, it has become a necessity to have an accurate, valid deception detection system for the detective to ease the interrogation process.

4.6 Context Diagram

Below figure 3 depicts the context diagram of the **Caught in a Lie** system. System and the system interactions are demonstrated in the diagram and the goal of the context diagram is the definition and clarification of the system boundaries.

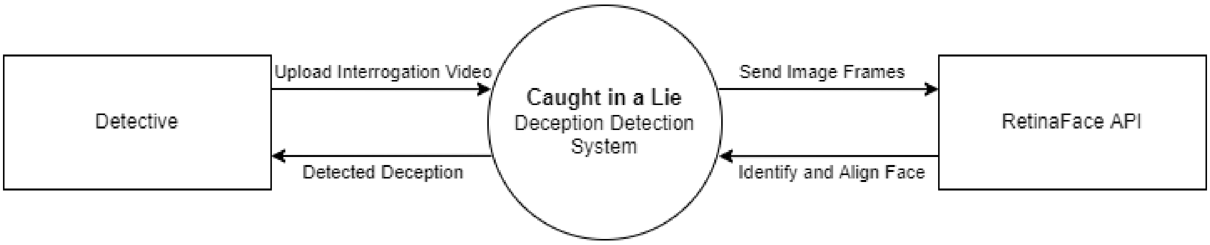

Figure 5 - Context Diagram (Self Composed)

4.7 Use Case Diagram

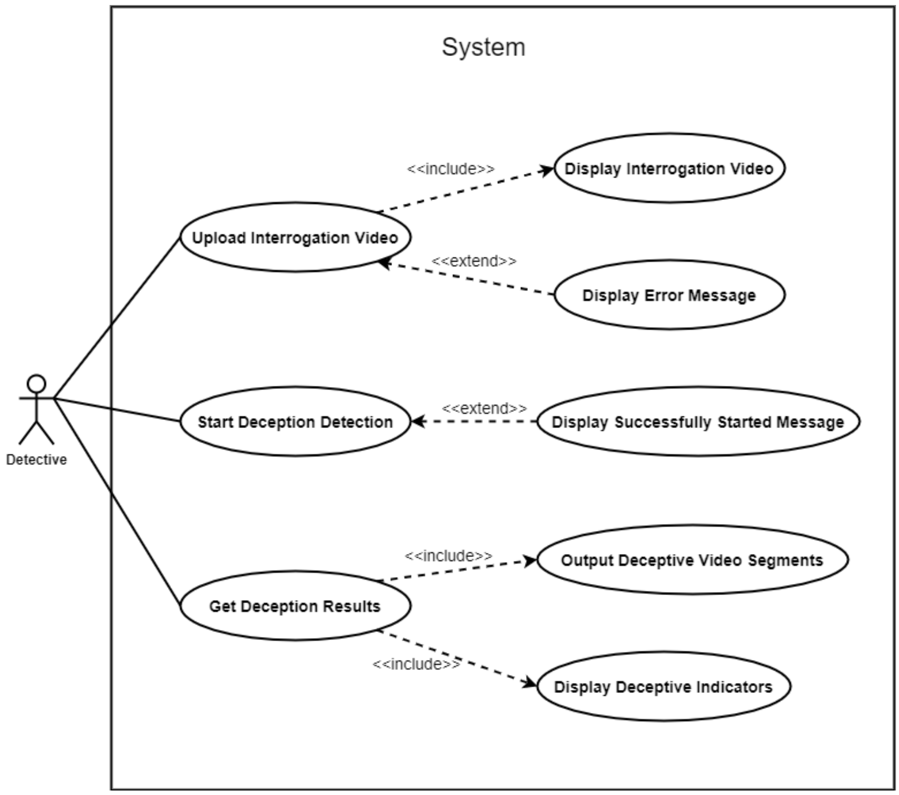

Figure 6 - Use Case Diagram (Self Composed)

The main actor of the system is the Detective. Detective has the ability to upload the interrogation video, start the deception detection process using a button click and get the deception results back as the output. System detects deception after the detective uploads the video and start detecting. Detailed description of the use cases is included in use case descriptions.

4.8 Use Case Description

The main identified use case descriptions of the systems are detailed below.

4.8.1 Upload Interrogation Video

Use Case Name	Upload Interrogation Video	
Use Case Description	Detective uploads the interrogation video to the system	
Priority Level	High	
Actor	Detective	
Pre-conditions	The system should be successfully accessed by the detective	
Trigger	Detective needs to upload the video	
Main flow	Actor	System
	1) Detective opens system 3) Detective uploads interrogation video	2) System displays upload video 4) System displays the interrogation video
Exception flow	Actor	System
	1) Detective uploads a different file type	2) System displays an error message asking to upload a video
Alternate flow		
Exclusions		
Post Conditions	Video starts uploading and displays uploaded video	

Table 43 - Upload Interrogation Video Use Case Description

4.8.2 Start Deception Detection

Use Case Name	Start Deception Detection
Use Case Description	Detective starts deception detection procedure by clicking start button
Priority Level	High
Actor	Detective

Pre-conditions	The detective should upload the interrogation video	
Trigger	Detective needs to click the start button	
Main flow	Actor	System
	1) Detective clicks on start button	2) System starts deception detection process
Exception flow	Actor	System
	1) Detective clicks on start button without uploading video	2) System displays an error message
Alternate flow		
Exclusions		
Post Conditions	System output deception results	

Table 44 - Start Deception Detection Use Case Description

4.8.3 Get Deception Results

Use Case Name	Get Deception Results	
Use Case Description	Detective gets deception results from the system as the output	
Priority Level	High	
Actor	Detective	
Pre-conditions	The detective should start the deception detection process	
Trigger	Detective needs to see the deception results	
Main flow	Actor	System
	3) Detective sees the displayed deception results from the system	1) System outputs deceptive video segments 2) System displays deceptive indicators
Exception flow	Actor	System
		1) System requires more time to output the results

	3) Detective sees the displayed waiting message	2) System displays a waiting message
Alternate flow		
Exclusions		
Post Conditions	System output deception results	

Table 45 - Get Deception Results Use Case Description

4.9 Requirement Specification

4.9.1 Functional Requirements

ID	Requirement	Priority Level
FR1	User should be able to login to the system	M
FR2	User should be able to upload the interrogation video	M
FR3	User should be able to proceed the deception detection by clicking start button	M
FR4	User should be able to see the deception results	M
FR5	System should be able to identify the uploaded file is in video file format and display error when different file types are uploaded	M
FR6	System should be able to identify human face from the interrogation video	M
FR7	System should be able to identify deceptive facial indicators including facial cues and chirality	M
FR8	System should be able to output the deceptive video segments	S
FR9	System should be able to display deceptive indicators and percentages	M

Table 46 - Functional Requirements Table

4.9.2 Non-Functional Requirements

ID	Requirement	Description
NFR1	Accuracy	Accuracy of the system should be to the point when detecting deception and system should not indicate normal behavior as deceptive.

NFR2	Performance	System should perform well without any issues or lags as the video analysis should happen without much delay.
NFR3	Usability	System should be user friendly and easy to use as the target users who are detectives are usually not from a technological background
NFR4	Reliability	System should be reliable for the detectives to detect deceptions of the suspects as it is a key process carried out in interrogations
NFR5	Maintainability	System should be easily supported, enhanced, and should have the ability to repair as the system will have further improvements

Table 47 - Non-Functional Requirements

4.10 Chapter Summary

The software requirement specification chapter included a descriptive stakeholder analysis supported with rich picture and onion models, a detailed description of the chosen requirement elicitation techniques and an analysis on the gathered requirements and findings. Use case diagrams along with use case descriptions elaborates the system process. Finally functional and non-functional requirements are specified in this chapter. Next chapter will be focusing on the design aspect of the proposed system **Caught in a Lie**.

CHAPTER 5: SOCIAL, LEGAL, ETHICAL AND PROFESSIONAL ISSUES

5.1 Chapter Overview

This chapter includes the social, legal, ethical, and professional issues occur throughout the project **Caught in a Lie**.

5.2 SLEP Issues and Mitigation

Legal or ethical issues were not faced in the design and development phases of the project. SLEP policy discussion is as follows,

Table 48 - SLEP Issues and Mitigation

Social	**Legal**
• Interviewees were informed prior to the interviews that information provided through interviews will be used for the research project. Information of interviewees were disclosed only if they agreed to disclose their information. • Dataset used for model training and testing purposes was a publicly available dataset with court trial videos. • Since non-verbal cues is used for deception detection, **Caught in a Lie** system does not own any cultural influences.	• Programming languages and tools used for prototype development were licensed under GPL license. • Implemented system will also be licensed under GPL license, • Requirement gathering interviews were only recorded with interviewees' consent. Information of the interviewee was only added with consent. • Evaluators evaluation feedback was added to the project with evaluators' consent.
Ethical	**Professional**
• Interviewees who were interviewed to gather requirements for the project were informed about the project and the contribution of the gathered information from interviewees.	• Software engineering best practices were followed throughout the design and development phases of the project. • User credentials used in the prototype are stored securely with encryption for passwords.

• Dataset used for the project is a publicly available real-life trial dataset which can be accessed online. • All materials received from outside which have included in the project were cited by giving credit to the author. None of the thesis data or results are plagiarized or fabricated.	• Machine which was used for development was constantly scanned for security vulnerabilities. • All the project results were truthful and none of the results were fabricated to deceive anyone associate with the project.

5.3 Chapter Summary

Social, legal, ethical, and professional issues were addressed in this chapter to verify that all issues in the project were mitigated and resolved. 3 social issues, 4 legal issues, 3 ethical issues and 4 professional issues were discussed in this chapter.

CHAPTER 6: SYSTEM ARCHITECTURE AND DESIGN

6.1 Chapter Overview

This chapter includes the design aspects of the entire system which have diagrammatically illustrated throughout this design chapter. The diagrams illustrated can be identified as high-level architecture diagram, class diagram, sequence diagram, activity diagrams and finally wireframes of the user interfaces.

6.2 Design Goals

Design goals of the system can be conveyed as follows,

Design Goal	Description
Accuracy	The intention of creating this automated deception detection system is to provide an accurate judgment for human deception detection in suspect interrogations. Since the suspect's truthfulness or deceptiveness lies on the system, this has to provide valid and accurate results. The system should outperform the accuracy of the manual deception detection systems.
Performance	As the manual deception detection approaches require an excessive amount of time and detective resources, creating an automated deception detection with better performance which does not require an excessive amount of time or detective resources. The system will provide accurate deception results without any errors with a lesser time and resources. Importance of better performance is that the system will not make the user worry about the process by providing results without complications.
Reusability	Since the system contributes to the domain of criminal investigation, the deception detection experts should be able to reuse the system for other developments. Therefore, the system must be built with reusability and flexibility where any developer can reuse the system components and experiment improvements in the system.
Scalability	The system should work well with larger datasets. The system currently uses 120 deceptive and truthful video datasets. In the future the system should have the ability to train with larger datasets which contain deceptive and truthful

	behaviors of humans belonging to different cultures and races. The users should also have the ability to access the system concurrently.

Table 49 - Design Goals Table

6.3 System Architecture Design

6.3.1 3-Tier Architecture Design

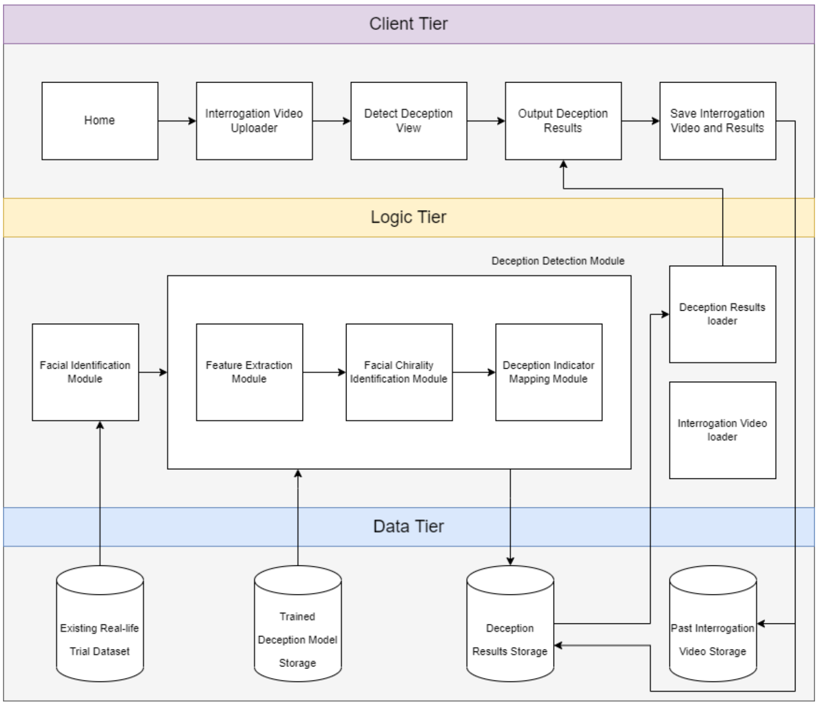

Figure 7 - 3-Tier Architecture Design (Self Composed)

Above figure illustrates the tied architecture of the system. This consists of three main tiers which can be identified as Data Tier, Logic Tier and Client Tier. Data Tier and the Client Tier are connected by the Logic Tier which is in between. Client tier can be identified as the presentation tier which includes the end-user interacting system. Data Tier contains the required data for the system. All three tiers are explained below.

6.3.1.1 Data Tier

- Existing Real-life Trial Dataset – This is the real-life trail dataset which includes 121 deceptive and truthful videos which has used to train the model and human facial identification. This dataset is accessible via facial identification module.
- Trained Deception Model Storage – This is the trained deception model storage which includes all the deep learning models that were trained in order to detect deception.
- Deception Results Storage – This is used to store the deception results which are deceptive and truthful percentages and the deceptive indicators.

- Past Interrogation Video Storage – This is used to store the past interrogation videos which have already been used to detect deception. This storage is used only when the detective needs to save the interrogation videos.

6.3.2 Logic Tier

- Facial Identification Module – This module the human facial identification module which will identify the human face and input that to the deception detection module for the deception detection tasks.
- Deception Detection Module – This module is the most important module which consists of feature extraction, facial chirality identification and deception indicator mapping modules which contribute to the deception detection task.
- Deception Results Loader – This module is used to load the deception results. Deception results will consist of two outputs which are deceptive and truthful percentages and the deception indicators.
- Interrogation Video Loader – This module is used to load the interrogation video for the detectives to see the video again.

6.3.3 Client Tier

- Home – This is the home page of the application where the name of the police department is displayed. The interrogation video upload button is available in the home page so that detectives can easily navigate to the uploader without any issue.
- Interrogation Video Uploader – This is the interrogation video uploader where the detective can upload a video with a compatible file format. (E.g., mp4)
- Detect Deception View – This will be the leading screen which is visible while detecting deception. So, the detective can see that the deception detection function is taking place.
- Output Deception results – This displays the deception results to the detective. This will include deceptive and truthful percentages of the video and the deceptive indicators identifies by the model.
- Save Interrogation Video and Results – This gives the option for the detective to save the interrogation video to the past interrogation video storage and store the results to the deception results storage. This gives the ability for the detectives to view the past videos whenever needed.

6.4 System Design

6.4.1 Class Diagram

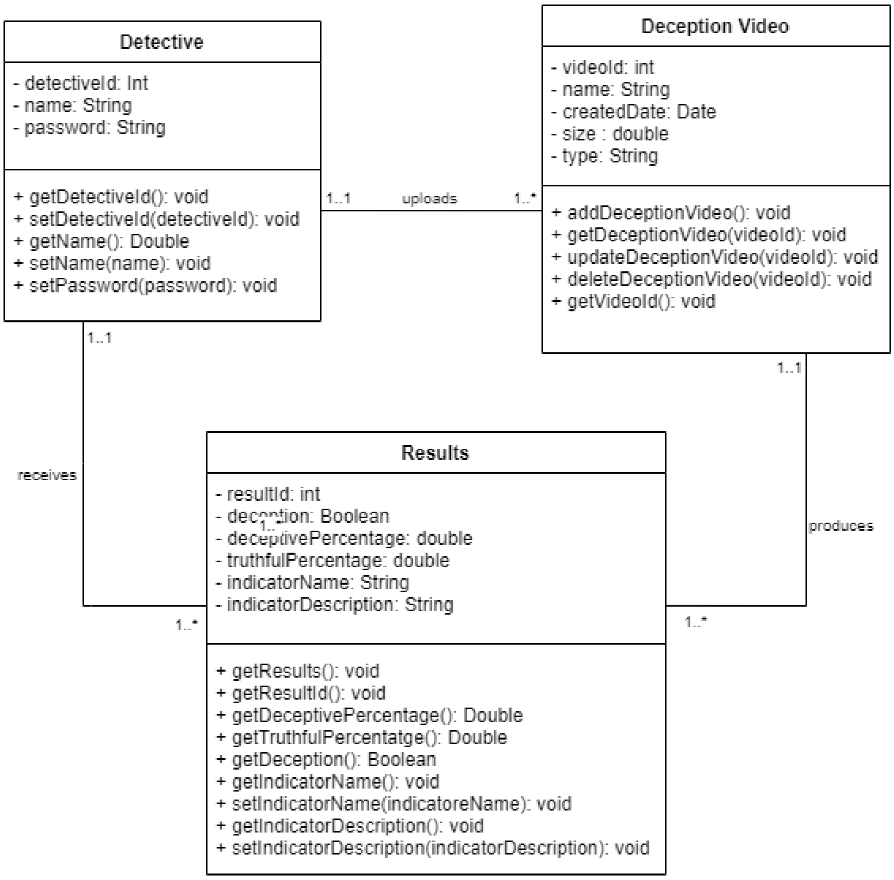

Figure 8 - Class Diagram (Self Composed)

Above figure depicts the class diagram of the system **Caught in a Lie**. Classes of the system have been identified and designed the class diagram by indicating the associations between classes. Detective, deception video and results can be identified as the three main classes. Deception results and indicators are both in results class. Detective has the ability to upload one or more deception videos, but a single deception video can be uploaded by only one detective. Deception video produces one or many results while a single result can be produced by only one deception video. Detective receives one or more results, and a single result can be received by only one detective. This class diagram has been designed considering these mentioned assumptions.

6.4.2 Sequence Diagram

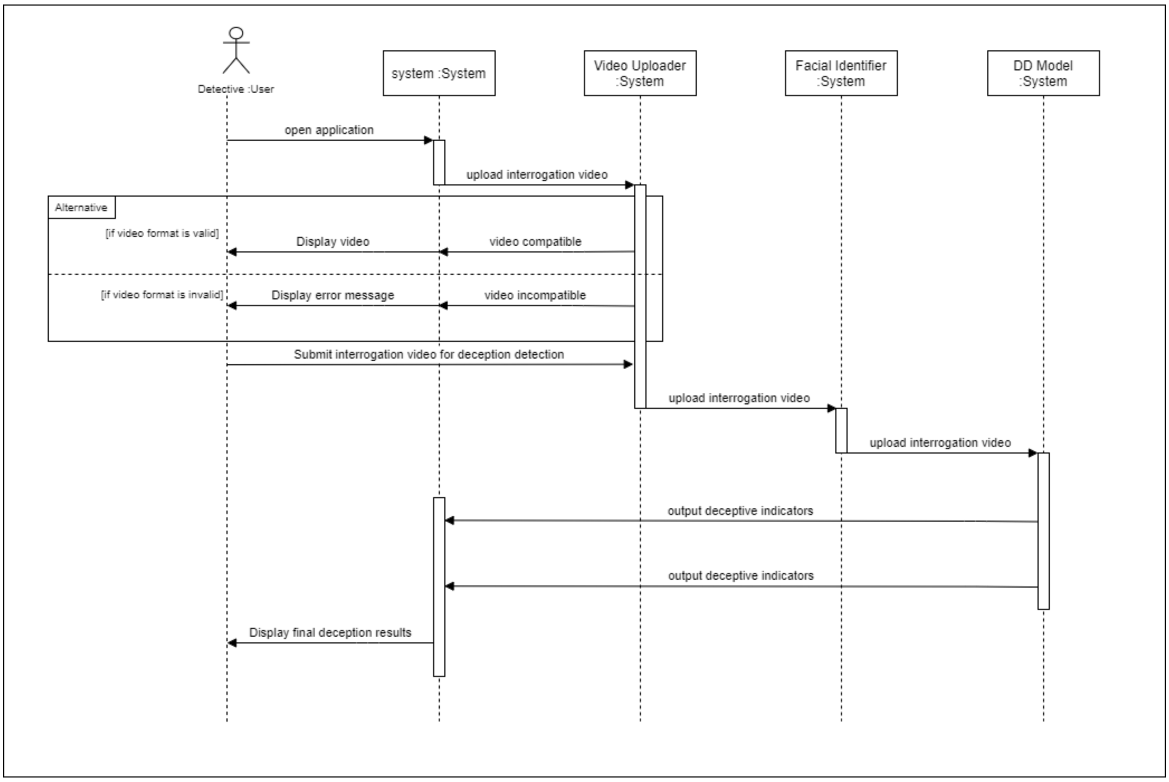

Figure 9 - Sequence Diagram (Self Composed)

Figure displays the sequence diagram of the **Caught in a Lie** system's main functionality which is upload video and detect deception. Detective can upload the interrogation video and after a compatible video is uploaded to the system, it will start identifying the human face. They the system will input to the deception model and detect deception. Finally, the system will display the deception results back to the detective in two forms which are deceptive/truthful percentages and deceptive indicators.

6.4.3 Activity Diagrams

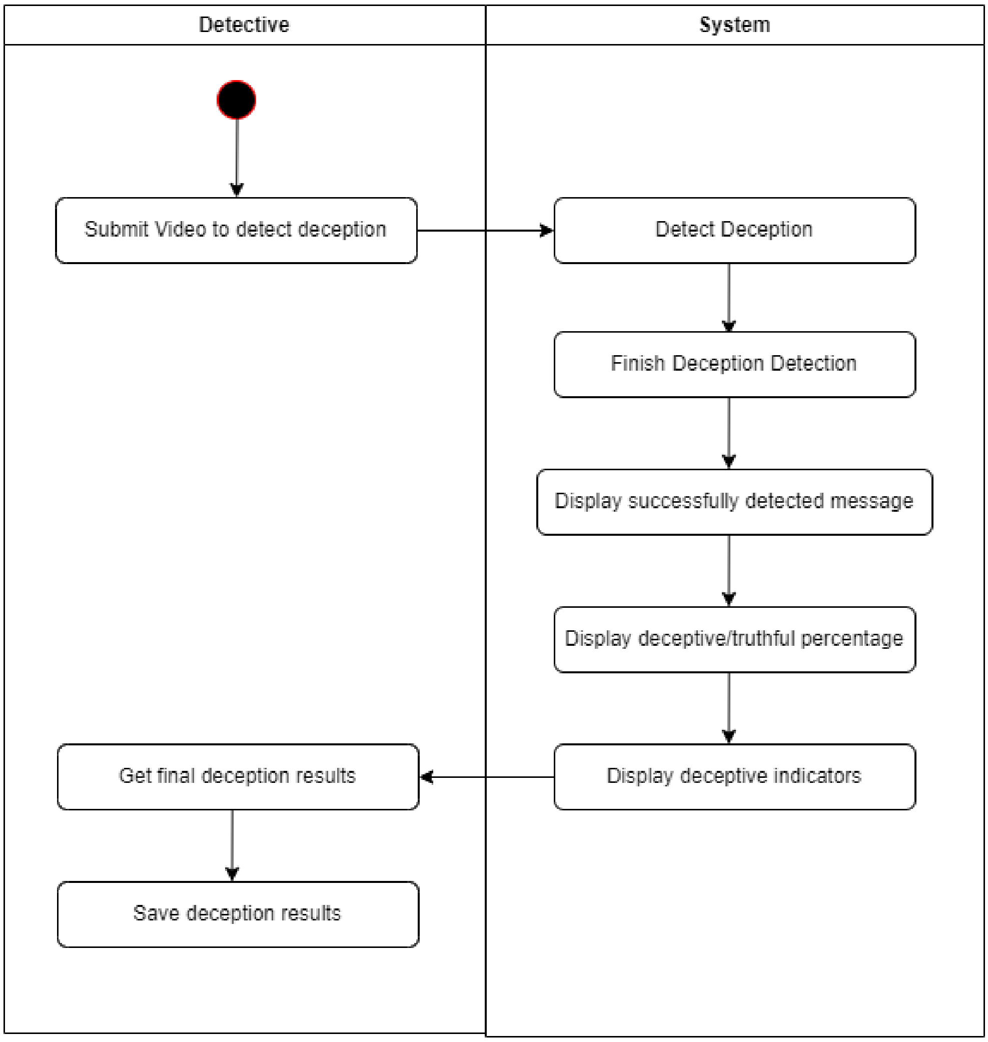

Figure 10 - Activity Diagram of Deception Detection (Self Composed)

Above figure illustrates the activity diagram of the deception detection functionality of the system **Caught in a Lie**. Activity diagram for the upload video functionality can be found under the **Appendix B** section.

6.4.4 Wireframes

The users of the **Caught in a Lie** system can be identified as detectives. Detectives may or may not have a technical background therefore the user interfaces of the system have been created to accommodate the factors which are simplicity and easy to use. System is a web application which contain three main pages where the detective can navigate through. Home page consists of minimal information about the police department and the video upload button which navigates to the deception detection page and provide the detective the facility to upload the interrogation video. The application has been created with great user experience where detectives can navigate easily within the application and get clear deception results after deception detection task. Detective also have the ability to save the previously uploaded interrogation videos in the system and use them later. Below figure depicts the wireframe for the home page of the **Caught in a Lie** system. The rest of the wireframes which are deception detection page and interrogation videos page can be found in the **Appendix C** section.

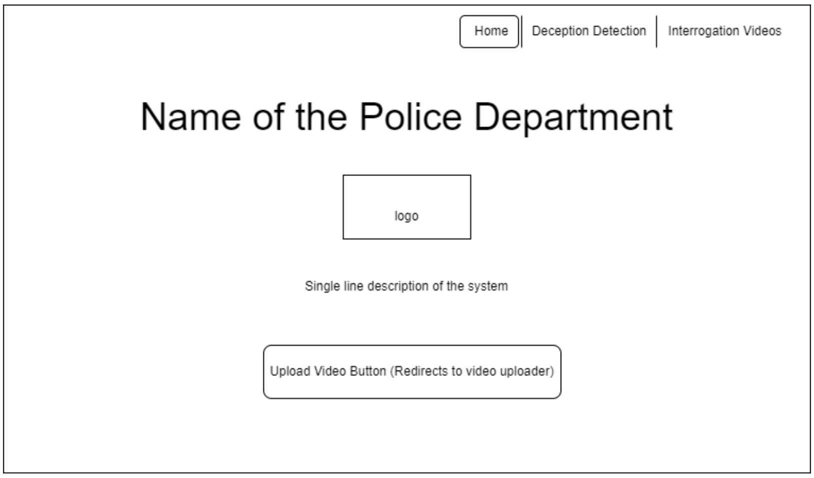

Figure 11 - Wireframe of the Home Page (Self Composed)

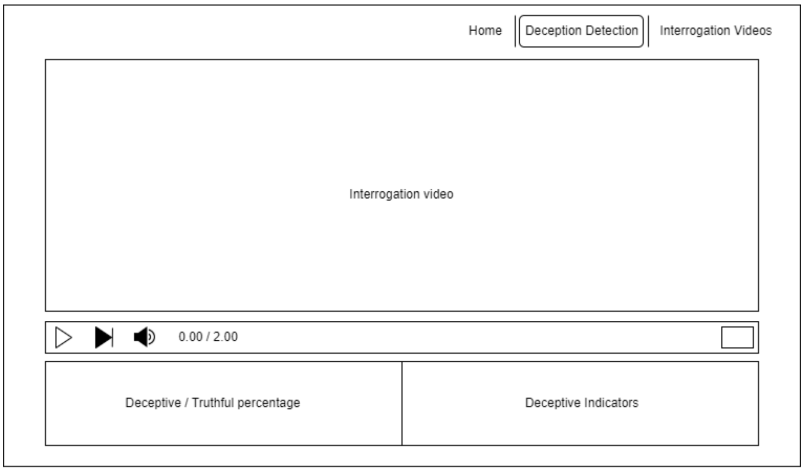

Figure 12 - Deception Detection Wireframe (Self Composed)

6.5 Chapter Summary

This chapter includes the entire design of the **Caught in a Lie** system including the design goals, high level architecture diagram, class diagram, activity diagrams, sequence diagram and the wireframes of the system. All the diagrammatic illustrations provide a better understanding and explanation of the system. The next chapter will include the initial implementation of the system where the author will be discussing about the technology selection, technology stack by providing the reasons for the selection.

CHAPTER 7: IMPLEMENTATION

7.1 Chapter Overview

This chapter includes the implementation of the **Caught in a Lie** system prototype. The chapter discusses the selection of technologies, development framework, programming languages, libraries, APIs, IDEs and datasets. Author wishes to explain the implementation of core functionalities by providing appropriate code snippets and screenshots to support the prototype implementation.

7.2 Technology Selection

7.2.1 Technology Stack

Technology stack includes the technologies chosen for the implementation of the system **Caught in a Lie**.

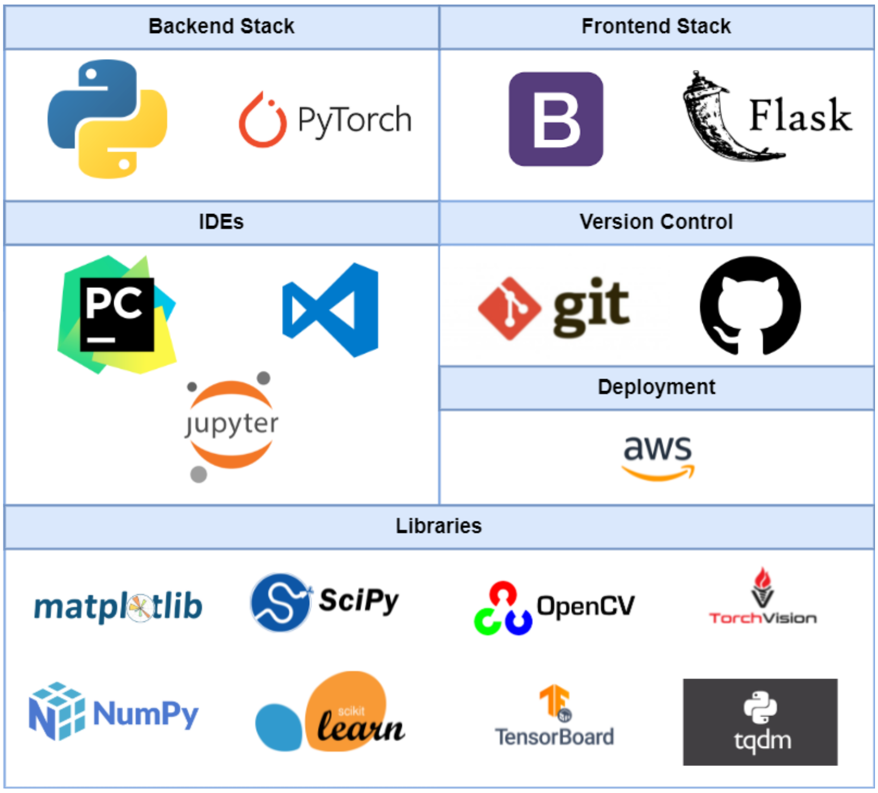

Figure 13 - Technology Stack (Self Composed)

As depicted in the above figure, backend stack was selected as python and python framework PyTorch. Frontend stack includes Flask framework with bootstrap to develop the web application. PyCharm, VSCode and Jupyter Notebook was chosen as the IDEs to implement the system. Version control was done using Git. Amazon Web Services cloud will be used to deploy the application.

7.2.2 Data Selection

Data selection is one of the main priorities in a data science project. Since the **Caught in a Lie** system detects deception in interrogation videos, the data can be identified as videos. On the subject of deception detection, system identifies the human facial cues and facial chirality to detect deception in interrogation videos. Therefore, a collection of deceptive and truthful videos was required for the training and testing purposes of the model.

In order to fulfill the data requirements of the project, author have identified a dataset which is known as Real-Life Trial dataset consists of video clips of public court trials. This real-life trial dataset consists of 121 videos which includes 61 deceptive video clips and 60 truthful video clips. The average length of the truthful video clips can be identified as 28.3 seconds and the average deceptive clip length can be identified as 27.7 seconds.

The dataset also includes the audio data, text data and micro expression data for the deceptive and truthful video clip sets. From these included data, author will be mainly focusing on micro expression data when implementing the system **Caught in a Lie**, as the system mainly focus on the non-verbal cues when detecting deception.

7.2.3 Selection of Development Framework

As for the deep learning framework. PyTorch framework was selected for the implementation of the **Caught in a Lie** system. PyTorch is a free and open-source framework which is based on the torch library (Learn the Basics — PyTorch Tutorials 1.10.1+cu102 documentation, 2022). Torchvision library will be used for deception video transformation, deception video classification, human face detection and human key point detection. PyTorch is known for its stability as it is built by Nvidia using C++. Therefore, PyTorch is a suitable framework for the system implementation.

Flask microframework was chosen to implement the web application. Flask is a lightweight web framework with higher flexibility. Flask was chosen considering the advantages which are higher compatibility with latest technologies, customization, higher scalability, ease of use, higher performance, and smaller code base. Flask uses bootstrap templates for the further designing in the system.

Bootstrap framework was chosen as the frontend framework of the system. The system **Caught in a Lie** requires a simple and user-friendly web application where detectives can use easily for the deception detection purpose. Reasons bootstrap was chosen as the frontend framework is because it is lightweight and customizable. Therefore, it does not affect the performance of the system

7.2.4 Programming Language

The development language which was chosen to implement the project is python. Python was selected as the main programming language for the system implementation because python is a powerful and flexible programming language which supports different deep learning and video analysis libraries and tools. Since python is open source and has a high community support, it is easier to overcome programming problems during implementation. Considering these reasons python was chosen as the main programming language of **Caught in a Lie** system.

7.2.5 Libraries Utilized

Plenty of libraries will be used for the system implementation such as Matplotlib, SciPy, tqdm, OpenCV, Pillow, TorchVision, NumPy, Scikit-image, TorchSummary and TensorBoard. Matplotlib will be used to save deceptive images and graphical plot outputs. Tqdm will be used to create progress meters and progress bars in the deception detection process. OpenCV is used for the facial recognition and detection. Pillow is also used to deal with deceptive and truthful images which were extracted from videos. NumPy will be used to deal with arrays and vectors of images. For image transformations, resize and rescale Scikit-image will be used. Tensorboard will be used to provide visualizations and measurements needed (Get started with TensorBoard | TensorFlow, 2022).

7.2.6 IDE's Utilized

PyCharm IDE was selected to develop the backend of the system and Visua l Studio Code IDE was selected to mainly develop the frontend of the system. PyCharm is one of the best IDEs for python language development. PyCharm supports packages which make the IDE flexible and easy to use (PyCharm: the Python IDE for Professional Developers by JetBrains, 2022). The frontend as well as the whole system will be developed using Visual Studio Code which is a high performing IDE (Visual Studio Code Frequently Asked Questions, 2022). Jupyter notebook was used for python script writing which will mainly be used for training and testing the deception detection model. PyCharm, Visual Studio Code will be used as main IDEs in the system development and Jupyter Notebook will be used for some data science aspects of the project.

7.2.7 Summary of Technology Selection

Component	Tool
Programming Language	Python
Deception Detection Model Development Framework	PyTorch
Final Product Development Framework	Flask
Frontend (UI) Framework	Bootstrap
Libraries	Matplotlib, SciPy, tqdm, OpenCV, Pillow, TorchVision, NumPy, Scikit-image, TorchSummary, TensorBoard
IDE	PyCharm VS Code Jupyter Notebook
Version Control	Git
Application Hosting	AWS Cloud

Table 50 - Summary of Technology Selection Table

7.3 Implementation of Core Functionalities

7.3.1 Image Extraction from Video

```python
import numpy as np
import imageio
import skimage
import cv2
from skimage import img_as_ubyte

def convertvideotoimage():
    filename = ['trial_lie_001.mp4']

    for order in range(len(filename)):
        video = imageio.get_reader(filename[order], 'ffmpeg')
        count = 0
        for im in video:
            image = skimage.img_as_float(im).astype(np.float32)
            frame = img_as_ubyte(image)
            cv_bgr_img = cv2.cvtColor(frame, cv2.COLOR_RGB2BGR)
            cv2.imwrite('./lie_images/lie_%d.jpg' % (count), cv_bgr_img)
            count += 1


if __name__ == "__main__":
    convertvideotoimage()
```

Figure 14 - Image Extraction from Video

This code snippet has been written to convert video file to image files and store those image files in a given folder names lie_images. Video file path is given in the filename. This code reads the video then convert the video into image frames and finally writes the images to the lie_images folder in jpg file format. Video trial_lie_001.mp4 is a 16 second video and 509 image frames were generated from that.

7.3.2 Facial Detection

7.3.2.1 Get Reference Facial Points

```
import cv2
import numpy as np
from skimage import transform as trans

# reference facial points, a list of coordinates (x,y)
REFERENCE_FACIAL_POINTS = [
    [30.29459953, 51.69630051],
    [65.53179932, 51.50139999],
    [48.02519989, 71.73660278],
    [33.54930115, 92.3655014],
    [62.72990036, 92.20410156]
]

DEFAULT_CROP_SIZE = (96, 112)

class FaceWarpException(Exception):
    def __str__(self):
        return 'In File {}:{}'.format(
            __file__, super.__str__(self))


def get_reference_facial_points(output_size=None,
                                inner_padding_factor=0.0,
                                outer_padding=(0, 0),
                                default_square=False):
    tmp_5pts = np.array(REFERENCE_FACIAL_POINTS)
    tmp_crop_size = np.array(DEFAULT_CROP_SIZE)

    # 0) make the inner region a square
    if default_square:
        size_diff = max(tmp_crop_size) - tmp_crop_size
        tmp_5pts += size_diff / 2
        tmp_crop_size += size_diff

    if (output_size and
            output_size[0] == tmp_crop_size[0] and
            output_size[1] == tmp_crop_size[1]):
        # print('output_size == DEFAULT_CROP_SIZE {}: return default reference points'.format(tmp_crop_size))
        return tmp_5pts

    if (inner_padding_factor == 0 and
            outer_padding == (0, 0)):
        if output_size is None:
            print('No paddings to do: return default reference points')
            return tmp_5pts
        else:
            raise FaceWarpException(
                'No paddings to do, output_size must be None or {}'.format(tmp_crop_size))


    # check output size
    if not (0 <= inner_padding_factor <= 1.0):
        raise FaceWarpException('Not (0 <= inner_padding_factor <= 1.0)')

    if ((inner_padding_factor > 0 or outer_padding[0] > 0 or outer_padding[1] > 0)
            and output_size is None):
        output_size = tmp_crop_size * \
                      (1 + inner_padding_factor * 2).astype(np.int32)
        output_size += np.array(outer_padding)
        print('              deduced from paddings, output_size = ', output_size)

    if not (outer_padding[0] < output_size[0]
            and outer_padding[1] < output_size[1]):
        raise FaceWarpException('Not (outer_padding[0] < output_size[0]'
                                'and outer_padding[1] < output_size[1])')
```

```
    if inner_padding_factor > 0:
        size_diff = tmp_crop_size * inner_padding_factor * 2
        tmp_5pts += size_diff / 2
        tmp_crop_size += np.round(size_diff).astype(np.int32)

    size_bf_outer_pad = np.array(output_size) - np.array(outer_padding) * 2

    if size_bf_outer_pad[0] * tmp_crop_size[1] != size_bf_outer_pad[1] * tmp_crop_size[0]:
        raise FaceWarpException('Must have (output_size - outer_padding)'
                                '= some_scale * (crop_size * (1.0 + inner_padding_factor)')

    scale_factor = size_bf_outer_pad[0].astype(np.float32) / tmp_crop_size[0]
    tmp_5pts = tmp_5pts * scale_factor
    tmp_crop_size = size_bf_outer_pad

    reference_5point = tmp_5pts + np.array(outer_padding)
    tmp_crop_size = output_size

    return reference_5point
```

Figure 15 - Get Reference Facial Points

7.3.2.2 Warp and Crop Face

```
def warp_and_crop_face(src_img,
                       facial_pts,
                       reference_pts=None,
                       crop_size=DEFAULT_CROP_SIZE,
                       align_type='smilarity'):
    if reference_pts is None:
        if crop_size[0] == DEFAULT_CROP_SIZE[0] and crop_size[1] == DEFAULT_CROP_SIZE[1]:
            reference_pts = REFERENCE_FACIAL_POINTS
        else:
            default_square = False
            inner_padding_factor = 0
            outer_padding = (0, 0)
            output_size = crop_size

            reference_pts = get_reference_facial_points(output_size,
                                                        inner_padding_factor,
                                                        outer_padding,
                                                        default_square)
    ref_pts = np.float32(reference_pts)
    ref_pts_shp = ref_pts.shape
    if max(ref_pts_shp) < 3 or min(ref_pts_shp) != 2:
        raise FaceWarpException(
            'reference_pts.shape must be (K,2) or (2,K) and K>2')

    if ref_pts_shp[0] == 2:
        ref_pts = ref_pts.T

    src_pts = np.float32(facial_pts)
    src_pts_shp = src_pts.shape #(5,2)
    if max(src_pts_shp) < 3 or min(src_pts_shp) != 2:
        raise FaceWarpException(
            'facial_pts.shape must be (K,2) or (2,K) and K>2')

    if src_pts_shp[0] == 2:
        src_pts = src_pts.T

    if src_pts.shape != ref_pts.shape:
        raise FaceWarpException(
            'facial_pts and reference_pts must have the same shape')

    if align_type == 'cv2_affine':
        tfm = cv2.getAffineTransform(src_pts[0:3], ref_pts[0:3])
    elif align_type == 'affine':
        tfm = get_affine_transform_matrix(src_pts, ref_pts)
    else:
        tform = trans.SimilarityTransform()
        tform.estimate(src_pts, ref_pts)
        tfm = tform.params[0:2, :]

    face_img = cv2.warpAffine(src_img, tfm, (crop_size[0], crop_size[1]))

    return face_img
```

Figure 16 - Warp and Crop Face

The FaceWarpException which includes two main functions get_reference_facial_points and warp_and_crop_face. These REFERECE_FACIAL_PPOINTS are the pixel coordinated for the features in face such as eyes, ears, nose, lips. The get_reference_facial_points functionality was used to retrieve the facial points from the images and the warp_and_crop_face was used to do any warp alignments of the face images and crop only the face from the image and saved.

7.3.3 Image Processing

```
import os
import argparse
import cv2
import numpy as np
from utils import align_face
from RetinaFace.RetinaFaceDetection import retina_face
import imageio
from skimage import img_as_ubyte
import skimage
parser = argparse.ArgumentParser()
parser.add_argument('-m', '--trained_model', default='./RetinaFace/weights/mobilenet0.25_Final.pth',
                    type=str, help='Trained state_dict file path to open')
parser.add_argument('--network', default='mobile0.25', help='Backbone network mobile0.25 or resnet50')
parser.add_argument('--cpu', action="store_true", default=False, help='Use cpu inference')
parser.add_argument('--confidence_threshold', default=0.02, type=float, help='confidence_threshold')
parser.add_argument('--top_k', default=5000, type=int, help='top_k')
parser.add_argument('--nms_threshold', default=0.4, type=float, help='nms_threshold')
parser.add_argument('--keep_top_k', default=750, type=int, help='keep_top_k')
parser.add_argument('-s', '--save_image', action="store_true", default=True, help='show detection results')
parser.add_argument('--vis_thres', default=0.6, type=float, help='visualization_threshold')
parser.add_argument('--gpu_num', default= "0", type=str, help='GPU number')
args = parser.parse_args()

face_detection = retina_face(crop_size = 224, args = args)
video_path = './d61.mp4'
vid = imageio.get_reader(video_path, 'ffmpeg')
for idx,im in enumerate(vid):
    image = skimage.img_as_float(im).astype(np.float32)
    frame = img_as_ubyte(image)
    img_raw, output_raw, output_det, output_det_draw, output_points, output_points_crop,face = face_detection.detect_face(frame)
    if(len(output_points)):
        for face in range(len(output_points)):
            out_raw = align_face(output_raw, output_points[face], crop_size_h = 128, crop_size_w = 128)
            out_raw = cv2.resize(out_raw,(224, 224))
            out_raw = cv2.cvtColor(out_raw,cv2.COLOR_RGB2BGR)

            num = ''.join([x for x in video_path.split('.')[0] if x.isdigit()])
            num = num.zfill(3)
            idx = str(idx).zfill(4)
            if not os.path.exists('./MSPL_YTD_images/lie_061'):
                os.makedirs('./MSPL_YTD_images/lie_061')
            cv2.imwrite('./MSPL_YTD_images/lie_061/image_%s.jpg'%(idx),out_raw)
    else:
        continue
```

Figure 17 - Image Processing

This is the image processing functionality of the facial identification core functionality of the system. Single video is taken into account in this process. RetinaFace deep learning based facial and facial landmark detector was used for the facial detection in this functionality. Facial alignment which was done in utils was used to do the facial alignment. Images are processed and facial detection and alignment was done in this code.

Rest of the code snippets of the utils.py can be found in the **Appendix D** section.

7.3.4 Chirality Action Process

```python
import os
import torch
from torch.utils.data.dataset import Dataset
from model.Chirality_Action.config import Config
from model.Chirality_Action.dataset import ImageList_lie
import tqdm
import csv

class Chirality_Action_Resnet:
    def __init__(self, args):
        self.model = torch.load(
            'C:/Users/Chamudi/Deception_Detection_Model/model/Chirality_Action/model/se_resnet50_MLSM/AU_0.4829_0'
            '.8912_116.pth')

    def _pred(self, img, args):
        test = ImageList_lie(image=img, transform=True)
        test_loader = torch.utils.data.DataLoader(test, batch_size=1)
        self.model.eval()
        with torch.no_grad():
            for idx, inputs in enumerate(test_loader):
                device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
                inputs = inputs.to(device)
                logps, emb = self.model(inputs, args)
        torch.cuda.empty_cache()
        return logps, emb

if __name__ == "__main__":
    args = Config()
    Chirality_Action_class = Chirality_Action_Resnet(args)
    if not os.path.exists(args.save_path):
        os.makedirs(args.save_path)
    data_folders = os.listdir(args.data_path)
    data_folders.sort()
    for folder in tqdm.tqdm(data_folders):
        if not os.path.exists(args.save_path + '/' + folder):
            os.makedirs(args.save_path + '/' + folder)

        csv_path_p = args.save_path + '/' + folder + '/' + 'Res_AU_pred.csv'
        csv_path_e = args.save_path + '/' + folder + '/' + 'Res_AU_embed.csv'

        with open(csv_path_p, 'w', newline='') as csvFile:
            writer = csv.writer(csvFile, dialect="excel")
            writer.writerow(
                ['AU01', 'AU02', 'AU04', 'AU05', 'AU06', 'AU09', 'AU12', 'AU15', 'AU17', 'AU20', 'AU25', 'AU26'])

        frames = os.listdir(args.data_path + '/' + folder)
        frames.sort()

        for frame in frames:
            if frame != '.ipynb_checkpoints':
                imgs_path = args.data_path + '/' + folder + '/' + frame
                logps, emb = Chirality_Action_class._pred(imgs_path)
                logps = logps.tolist()[0]
                for index, i in enumerate(logps):
                    if i >= 0.01:
                        logps[index] = 1
                    else:
                        logps[index] = 0
                with open(csv_path_e, 'a', newline='') as csvFile:
                    writer = csv.writer(csvFile, dialect="excel")
                    writer.writerow(emb.tolist()[0])
```

Figure 18 - Chirality Action Process

This Chirality_Action_Resnet Class was created to load the action chirality model and detect the chirality action using pretrained resnet50 model. 12 main action units were used to detect action chirality which are AU01 – inner brow racer, AU02 – outer brow racer, AU03 – brow lower, AU04 – upper lid racer, AU05 – cheek racer, AU06 – nose wrinkle, AU07 – lip corner puller, AU08 – lip conner depressor, AU09 – chin raiser, AU10 – lip stretcher, AU11 – lip part and AU12 – jaw drop. Deception was detected using chirality action units.

7.3.5 Emotion Cue Process

```python
from __future__ import print_function
import argparse
import torch
import torch.backends.cudnn as cudnn
import os
import torch.nn.parallel
import torch.optim
import torch.utils.data
import tqdm
from model.Emotion_Cues.Code import load_materials, Model
import csv
import cv2

class Emotion_Cues:
    def __init__(self, args):
        at_type = ['self-attention', 'self_relation-attention'][args.at_type]
        ''' Load model '''
        _structure = Model.resnet18_AT(at_type=at_type)
        _parameterDir = args.preTrain_path
        self.model = load_materials.LoadParameter(_structure, _parameterDir)
        cudnn.benchmark = True

    def validate(self, frame_list):
        val_loader = load_materials.LoadAFEW(arg_listeval=frame_list)  # list_eval
        self.model.eval()
        with torch.no_grad():
            for i, (input_var, index) in enumerate(val_loader):
                input_var = torch.autograd.Variable(torch.stack(input_var, dim=4))
                ''' model & full_model'''
                pred_score, self_embedding, relation_embedding = self.model(input_var, phrase='train')
        torch.cuda.empty_cache()
        return pred_score, self_embedding, relation_embedding

if __name__ == '__main__':
    parser = argparse.ArgumentParser(description='PyTorch CelebA Training')
    args = parser.parse_args()
    Emotion_class = Emotion_Cues(args=args)
    ''' Load and predict data '''
    save_path = '../svm/Real-life_dataprocess'
    if not os.path.exists(save_path):
        os.makedirs(save_path)
    data_path = '/model/Emotion_Cues/Real-life-trial-images'
    data_folders = os.listdir(data_path)
    for folder in tqdm.tqdm(data_folders):
        if not os.path.exists(save_path + '/' + folder):
            os.makedirs(save_path + '/' + folder)

        csv_path_e = save_path + '/' + folder + '/' + 'emotion_embedding_%d.csv' % (args.frames_embed)
        frames = os.listdir(data_path + '/' + folder)
        frames.sort()
        frame_list = []
        frame_embed_list = []
        for frame in frames:
            if frame != '.ipynb_checkpoints':
                img = cv2.imread(data_path + '/' + folder + '/' + frame)
                frame_list.append(img)
                frame_embed_list.append(img)
                if len(frame_list) == args.frames or frame == frames[-1]:
                    pred_score, _, _ = Emotion_class.validate(frame_list)
                    frame_list = []
                if len(frame_embed_list) == args.frames_embed or frame == frames[-1]:
                    pred_score, self_embedding, relation_embedding = Emotion_class.validate(frame_embed_list)
                    if args.at_type == 1:
                        embedding = relation_embedding
                    else:
                        embedding = self_embedding
                    frame_embed_list = []
                    with open(csv_path_e, 'a', newline='') as csvFile:
                        writer = csv.writer(csvFile, dialect="excel")
                        writer.writerow(embedding.cpu().numpy()[0])
```

Figure 19 - Emotion Cue Process

Facial cues were detected and used to detect deception. This class loads resnet model and predicts current emotion cue in the face of the image. Then save coordinates of the identified emotion cue to a csv file. 7 facial cues are used to train the model which are happy, angry, disgust, fear, sad, neutral, surprise. Model training and testing can be found in **Appendix D**.

7.4 User Interfaces

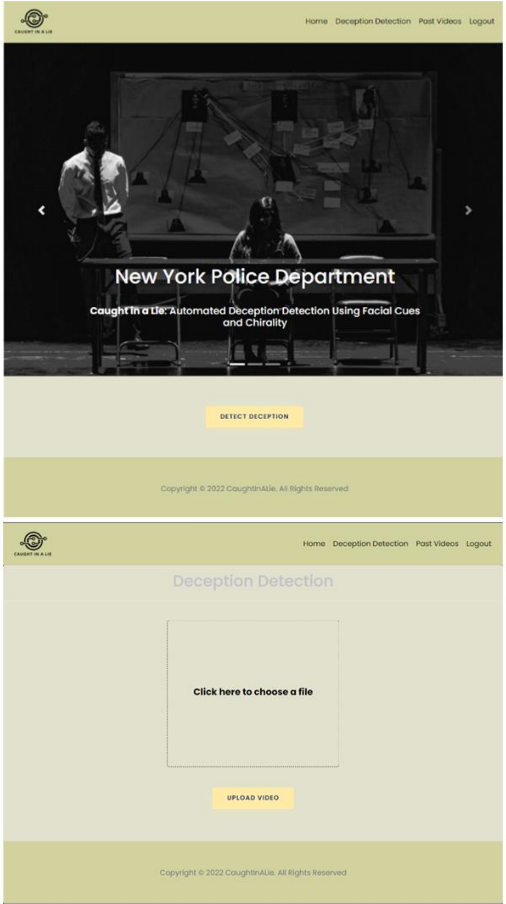

Figure 22 - Home & Deception Detection UIs

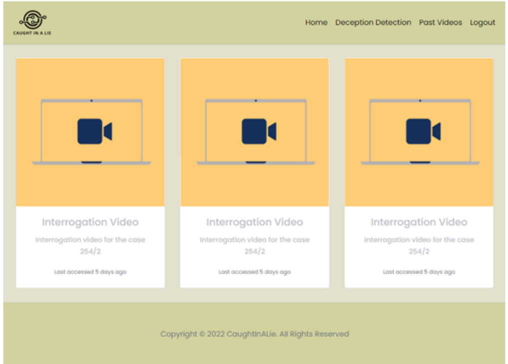

Figure 21 - Past Videos UI

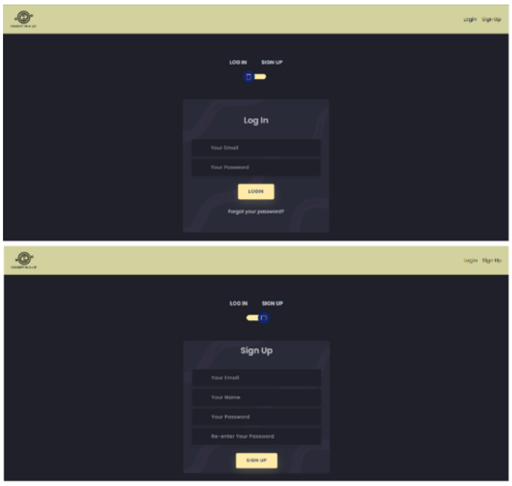

Figure 20 - Login & Sign-up UIs

7.5 Chapter Summary

This chapter includes a thorough explanation of the implementation of the system **Caught in a Lie**. The technology stack including development framework, programming languages, libraries, APIs, IDEs were identified and the reasons for choosing were explained well in the chapter. Core functionality implementation was described with the support of the screenshots of code snippets.

CHAPTER 8: TESTING

8.1 Chapter Overview

Previous chapter discussed about the implementation of **Caught in a Lie** system. This testing chapter will consist of objectives and goals of testing, testing criteria, model testing, benchmarking, functional and non-functional testing, module and integration testing and limitations of the testing process. This chapter will cover the testing phase of the system **Caught in a Lie** in detail.

8.2 Objectives and Goals of Testing

Once a system is developed, it should be tested to confirm whether the system works as expected by meeting all evaluation criterions. Testing objectives were followed in testing process, and they are as follows,

- Confirm **Caught in a Lie**'s functional requirements are satisfied and work as expected
- Confirm **Caught in a Lie**'s non-functional requirements are satisfied and up to end user satisfactory level
- Confirm **Caught in a Lie**'s models are properly implemented
- Validate code quality of the system by checking whether best practices were properly followed in implementation
- Identify **Caught in a Lie** system bugs and take actions in fixing the bugs before releasing to production

8.3 Testing Criteria

Testing criteria was followed to test the system covering the functional and non-functional requirement aspects of the system. The criterions are,

Functional Quality – This criterion is to validate if the system functions work as expected satisfying all functional requirements.

Structural Quality – This criterion is to validate if the non-functional requirements meet the expected results from the system.

8.4 Model Testing

Real-life trial dataset was used to train the newly created deception detection model Caught-in-a-Lie which specialized in identifying human facial cues and chirality actions using action units. Training the model with well refined data is important for the model to produce finest results.

Testing is conducted to test and verify the model has trained properly and produce expected results. Resnet50 convolutional neural network with 50 layers was used for human facial image classification to predict facial cues and chirality action unit. Resnet50 and SVM were together used for the deception detection model.

Accuracy and AUC of the model is depicted in below figure.

Methods	ACC(%)	AUC(%)
Our (all), frames = 120,320 features	88.65	92.40
Ours (all), frames = 130,290 features	88.75	91.30

Figure 23 - Accuracy and AUC Percentages

Accuracy percentage of the deception detection model is approximately 88% which is a good accuracy compared to currently available research. AUC also known as the area under the ROC curve is the probability that model ranks a random positive example. Performance measure across all possible classification entry is provided by AUC (Classification: ROC Curve and AUC | Machine Learning Crash Course | Google Developers, 2022). Value of AUC is ranges from 0-1. Deception detection model's AUC percentage is displayed in the above figure. AUC percentage is approximately 91% which is better AUC percentage.

Testing function was written to test the model and calculate precision, recall, f1 and accuracy values. The code of the testing function can be found in **Appendix D**. MSE which is the mean squared error and MAE which is the mean absolute error values are also calculated in through the testing function.

Config file which contains the model parameters, weights, dataset paths has been created to use and produce repeatable outcomes in testing and evaluation. Model parameters include 12 num_classes, 10 batch_size,120 epochs and BCE loss_function. Screenshot of the config file can be accessed in **Appendix D**. K-Fold Cross validation was used for model testing and evaluation of the deception detection model in **Caught in a Lie** system.

8.5 Benchmarking

Benchmarking was conducted to compare the **Caught in a Lie** system with other research conducted in the area of automated deception detection and manual deception detection.

8.5.1 Competitive Benchmarking

Automated deception detection systems are not commercially available. Therefore, competitive benchmarking was carried out on the research in the area of automated deception detection.

Since there are limited number of publicly available datasets for deception detection, most of the researches have been using the same dataset which is the ***Real-life trial dataset***. Below table compares accuracy of competitive research models with **Caught in a Lie** model.

Research	**Model Algorithms**	**Accuracy Percentage**
Automatic Deception Detection in RGB videos using Facial Action Units - Facial Action Units	Logistic Regression	54.86%
	Random Forest	58.42%
	SVM - Linear	65.56%
	SVM - Sigmoid	71.99%
	SVM - RBF	76.84
Deception Detection by 2D-to-3D Face Reconstruction from Videos	Time-CNN	69%
	RNN Model + CNN	73%
Facial Expression Based Imagination Index and a Transfer Learning Approach to Detect Deception	WFM	70%
	SVM	61%
Deception Detection in Videos using the Facial Action Coding System	FACs with LSTM	89.49%
	Decision Tree and Random Forest	75.2%
	SVM-RBF	76.84%
	ADABOOST	88%
LieToMe: An Ensemble Approach for Deception Detection from Facial Cues	Random Forest	83.61%
	L-SVM	91.87%
	Linear Regression	92.01%
	MLP	91.87%
	RBF-SVM	90.75%
Toward End-to-End Deception Detection in Videos	Vocal Model	74.16%
	Visual Model	75%
	Hybrid Model	84.16%

Table 51 - Competitive Benchmarking

Benchmarking results illustrates that **Caught in a Lie** system outperforms most of the researches carried out in the area of deception detection. Therefore, it concludes that the system satisfies the requirement of deception detection well. Please refer **Appendix E** to find screenshots of competitive benchmarking proof.

8.5.2 Functional Benchmarking

Functional benchmarking was conducted to compare **Caught in a Lie** model with the manual deception detection. Manual deception detection conducted by humans, only has an accuracy of 52% which is similar to the value achieved from a simple coin toss guess. Since the system model accuracy has a better value of 88% which outperforms the 52% of manual deception detection accuracy, it is proven that the **Caught in a Lie** model is a better system for the deception detection process.

8.6 Functional Testing

System's functional testing was carried out considering the functional requirements identified. Test cases are described in detail below.

Test case	FR ID	User Action	Expected Result	Actual Result	Result Status
1	FR1	User logs in to the system	Successfully login and get access to the system	Successfully login and get access to the system	Passed
2	FR2	User uploads the interrogation video	Video uploads and displays	Video uploads and displays	Passed
3	FR3	User proceeds the deception detection by clicking start button	Starts deception detection	Starts deception detection	Passed
4	FR4	User sees the deception results	Deception results are visible	Deception results are visible	Passed
5	FR5	User inputs invalid file format for videos	Display error message and video does not get uploaded	Display error message and video does not get uploaded	Passed
6	FR6	User starts deception detection process	System detects human face and align face image	System detects human face and align face image	Passed

7	FR7	User gets deceptive indicator results	Deceptive indicators are displayed	Deceptive indicators are displayed	Passed
8	FR8	User sees deceptive videos	System displays deceptive video	System displays deceptive video	Passed
9	FR9	User gets deceptive percentage results	Deceptive percentages are displayed	Deceptive percentages are displayed	Passed

Table 52 - Functional Testing

8.7 Module and Integration Testing

Integration testing was conducted to test each individual module of the system to verify they are working. Hence, the author can confirm they intent to work when they are integrated.

Module	**Input**	**Expected Output**	**Actual Output**	**Status**
Video input validation	Correct video format	Video gets uploaded	Video gets uploaded	Passed
	Incorrect video format	Display error message	Display error message	Passed
	Non-video file type	Display error message	Display error message	Passed
Video to Image conversion	Video	Sequence of image frames	Sequence of image frames	Passed
Face detection	Image	Detect, crop, and align face	Detect, crop and align face	Passed
Deception Detection	Face Image	Non-verbal deceptive indicators – AUs, Cues, Chirality	Non-verbal deceptive indicators – AUs, Cues, Chirality	Passed

Table 53 - Integration Testing

8.8 Non-Functional Testing

Project **Caught in a Lie** focused on a set of non-functional requirements to provide system with better user experience. Performance, accuracy, usability, reliability, and maintainability are the non-functional requirements satisfied in developing the system.

8.8.1 Performance

Performance of the system was tested considering the model aspect of the system as well as the product aspect.

8.8.1.1 Research Development Performance

Automated deception detection using human facial cues and facial chirality was the research carries out in **Caught in a Lie** system. Since the input is a video, author researched about video to image conversion. Facial identification, cue identification, action chirality identification was done as a part of the research. Therefore, this research required a better GPU, CPU and memory power which satisfies project requirements. NVIDIA GeForce RTX 3050 GPU was used for model training with better resource consumption. Below figure depicts the information of GPU usage.

```
Python 3.9.7 (default, Sep 16 2021, 16:59:28) [MSC v.1916 64 bit (AMD64)]
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2022 NVIDIA Corporation
Built on Thu_Feb_10_19:03:51_Pacific_Standard_Time_2022
Cuda compilation tools, release 11.6, V11.6.112
Build cuda_11.6.r11.6/compiler.30978841_0

In [6]: !nvidia-smi
Fri Apr 29 20:49:27 2022
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 511.69       Driver Version: 511.69       CUDA Version: 11.6     |
|-------------------------------+----------------------+----------------------+
| GPU  Name            TCC/WDDM | Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  NVIDIA GeForce ... WDDM  | 00000000:01:00.0 Off |                  N/A |
| N/A   59C    P5    11W /  N/A |   1782MiB /  4096MiB |     17%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|    0   N/A  N/A     26724      C   ...mudi\anaconda3\python.exe    N/A      |
+-----------------------------------------------------------------------------+
```

Figure 24 - GPU usage information

Using the provided GPU resources, **Caught in a Lie** system's model was trained properly to perform deception detection as expected.

As mentioned above, a good CPU and memory power was required for the training, testing of the model as well as the model usage in the application. The application was tested in the local environment to identify CPU and memory usage to depict the performance of the application. This was tested on a machine with AMD Ryzen 7 4800H with Radeon Graphics 2.90 GHz processor with a 16GB RAM.

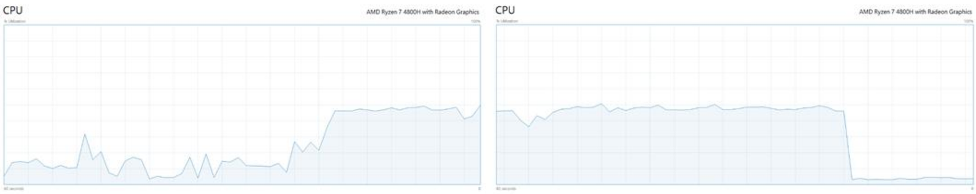

Figure 25 - CPU usage

Above figure depicts the CPU usage when the locally hosted application running on the background. An increase in CPU usage can be visible which was expected but there was no heavy usage once the system process is running. RAM usage is depicted in the below figure.

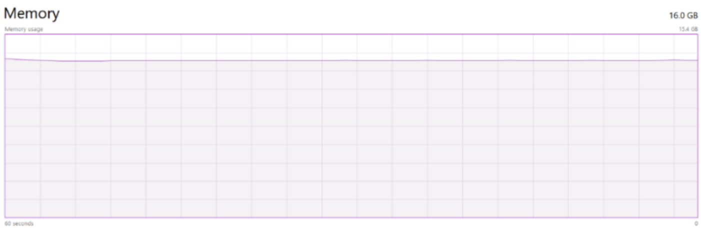

Figure 26 - RAM usage

Above mentioned testing results confirms that the **Caught in a Lie** system has better performance as expected.

8.8.1.2 Performance of Web Application

Caught in a Lie system's end user product is the web application which can be used to detect deception in interrogation videos. Therefore, performance of the web application was calculated with the performance feature available in Google Chrome. Web application performance is depicted in the below figure.

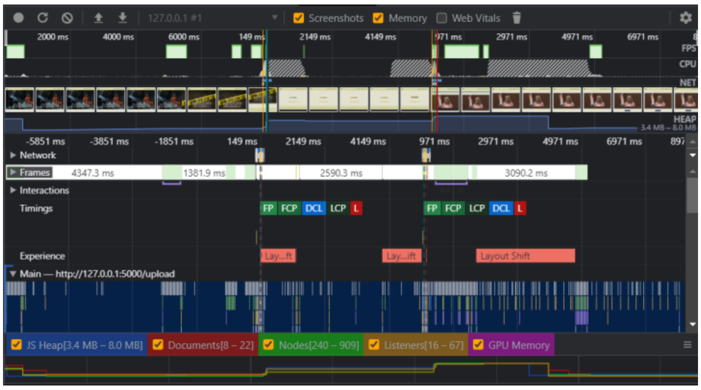

Figure 27 - Web Application Performance

It is proven that the application is performing well with a less memory allocation.

8.8.2 Reliability

Reliability of an application is built with security. Since this is a deception detection system which deals with interrogation videos of suspects which belongs to criminal investigation cases in different police stations or organizations, the data should be protected and secured without any data breaches or leakages. Security measures were taken to verify the security of the system.

- HTTPS which is the secured and encrypted protocol was used for the communication of the web application.
- User credentials were encrypted, and the system was designed for a target police station/criminal investigation organization with the capability of customizing according to the client. Therefore, data can be accessed only by the users within the organization.
- Software versions used for the application were verified to confirm they were up to date.4

These measures resulted in a secured and a well-protected web application for the system **Caught in a Lie**.

8.8.3 Usability

User interfaces of the system should be user friendly, and the system should be easy to ease target users' tasks. **Caught in a Lie** web application was created by following the UIUX principles on focusing on the minimalism. Usability testing was done by analyzing how well the user interfaces are created and the user experience are provided. Web application was made responsive which facilitated the usage from any device or browser. Selected color palette was used throughout the application for the user to easily identify web pages. GUI screenshots of the application can be accessed through the implementation chapter.

All the non-functional requirements of the system **Caught in a Lie** were satisfied, providing a better end user experience.

8.9 Limitations of the Testing Process

Publicly available deception detection datasets are limited. The dataset used for the model training was the Real-life trial dataset which included 61 Deceptive and 60 Truthful video clips. 10-fold cross validation was used to slip train and test data so 90% of data was used for training and only 10% of the data was used for testing the model. Therefore, approximately 20 clips were used for model testing. It would have been better if larger dataset was available for testing. Benchmarking was impossible to conduct for performance of the model as datasets used for similar models were not publicly available. Testing required high functional hardware resources and excessive time which made the testing phase difficult.

8.10 Chapter Summary

Testing phase of the project was discussed in detail in this testing chapter. Chapter started off with testing goals and objectives which were used to confirm whether the system works as expected. Testing criteria was defined for system testing covering functional and non-functional requirements. Afterwards, model testing, benchmarking, functional testing, module and integration testing, non-functional testing was discussed in the chapter. Finally, limitations of the testing process were mentioned. Testing chapter assisted the author to test and clarify that the application works as expected meeting project standards.

CHAPTER 9: EVALUATION

8.1 Chapter Overview

This chapter summarizes the evaluation aspect of the system. Critical evaluation was conducted by stakeholders which are domain experts and industrial experts. Self-evaluation of the author was also carried out and documented in this chapter. Evaluation results convey that the completion of the project according to the requirements has been done well.

8.2 Evaluation Methodology and Approach

Author took a qualitative approach in gathering requirements for the project as the target audience of the system are police officers and detectives who will be interrogating suspects. Therefore, author has decided to carry out the qualitative approach for the evaluation process of the system. Qualitative evaluation will be performed based on the system feedback received from evaluators, using thematic analysis. Detailed understanding of the system will be gained by conducting qualitative evaluation approach.

8.3 Evaluation Criteria

Qualitative approach was used to evaluate the system by following the evaluation criterions. Evaluation criterions and the purpose is explained below.

Criterion	Evaluation Purpose
Choice of topic, research gap	Evaluate and find out whether the project was up to the necessity of the domain and a fair contribution was done by filling the research gap identified. Comments and feedback on the system can be received from the target audience.
Project scope and depth	Project scope should be evaluated by criminal investigation experts to identify whether the selected scope is doable and worth investing time and hard work.
System design and architecture	To get an understanding whether the design and architecture of the system are done up to standards.
System development	Evaluate if the system implementation is up to industrial standards and properly completed.
System prototype	Prototype evaluation to prove that the proposed system works well and produce adequate results.

UIUX aspect of application	Evaluate the satisfaction of the target audience in using the system application in interrogation task.

Table 54 - Evaluation Criteria

8.4 Self-Evaluation

In self-evaluation, author will be evaluating work author has carried out in completing the system. Self-evaluation will be done to validate the identification of research topic and gap, scope and depth of the project, system design, architecture and development, prototype of the system and UIUX aspect of the system. Below evaluation criterions were used for the self-evaluation.

Criterion	Self-evaluation
Choice of topic, research gap	Deception detection is a key task carried out in interrogation process in order to solve crimes. Many researches have done in the area of deception detection, but validity of those systems are still in question. Therefore, usage in real-world interrogation process might be challenging. Author has introduced an automated deception detection system using nonverbal indicators to fill the identified gap. This research can be used in deception detection in other domains as well.
Project scope and depth	Project completion should be done with a short period of time which was less than 8 months. Author has identified the scope of the project considering the timeline. Therefore, author has broke down the automated deception detection by only using key nonverbal indicators which are facial cues and chirality. Since facial chirality model is involved in the project, author has a vast research area. Nonverbal indicators which are body posture, hand gestures will be taken into consideration in the future.
System design and architecture	Author created the system design and architecture fulfilling the identified requirements for the system. Tiered architecture was used also class diagrams, activity diagrams, sequence diagrams were used to depict the system design.
System development	The system was implemented according to the system architecture and design. System implementation was completed by attaining all functional and non-functional requirements. Author identified the best technologies for the implementation by deeply inspecting technologies.

System prototype	System prototype was successfully developed with all the identified necessary features. Detectives can use the prototype to test the deception detection in interrogation videos. System can undergo further improvements to give better results, but overall system prototype produces adequate results.
UIUX aspect of application	Graphical user interface of the system can be customized according to the police department or the organization. Detectives have the ability to easily navigate and use deception detection functionality using system's GUI. Author has created the application fulfilling the UIUX requirements.

Table 55 - Self Evaluation

8.5 Selection of evaluators

Target of the evaluation is to get feedback from wide variety of people associate with the system, System evaluation was done by 4 domain experts, 2 Doctoral researchers, 7 industrial experts. Three aspects were considered when selecting these evaluators. They are shown below.

1. Domain Experts - Evaluators with the knowledge of deception detection and interrogation process. Four policemen with high ranks were selected for this evaluation.
2. Industrial Experts - Evaluators with the knowledge of deep learning and technical knowledge. 7 industrial experts were chosen for this evaluation.
3. Researchers - Evaluators with the domain knowledge who are PhD holders further engaged in research.

Evaluators were chosen based on these three aspects and author received permission to add evaluators' details to the thesis. Please find the list of evaluators in Appendix F.

8.6 Evaluation Results and Expert Opinions

8.6.1 Qualitative Analysis

Thematic analysis was conducted with the qualitative data gathered from the system evaluators. Evaluator opinions for each evaluation criterion is mentioned below.

Criterion	**Evaluator Category**	**Evaluator opinion**
Choice of topic, research gap	Domain Experts	Currently deception detection is done manually in Sri Lanka therefore an automated deception detection system is very important to ease the

		task. Even though Criminal Investigation Department uses polygraph test for deception detection, the validity is still in question. Introducing an automated deception detection system using nonverbal indicators which are facial cues and chirality will be a great help for police officers.
	Industrial Experts	Topic and the research gap identified are sufficient and have a good impact on the domain. Automating deception detection is an interesting and important area of studies where the author can contribute to the domain as well as the technology.
	Researchers	Topic of the research and depth is appreciated because author have identified limitations of current systems to define a good research gap. Facial chirality is an area less research has been conducted. Facial chirality model for deception detection can be used for other domains in the future as well.
Project scope and depth	Domain Experts	Project scope is impressive because not only author has taken facial cues into consideration but also created a model for facial chirality which is a great help in indicating deception detection.
	Industrial Experts	Current scope is satisfactory, but a real time application would be more effective for the end user as they get the opportunity to detect deception then and there when the suspects are interrogating.
	Researchers	Project scope and depth is highly appreciable as project scope includes the significant non-verbal deceptive indicators and chirality in deception

		detection which is a great scope for a undergraduate project.
System design and architecture	Domain Experts	System design and architecture is well designed and established.
	Industrial Experts	Design aspect of the system is impressive as it covers all the requirements of the system.
	Researchers	Satisfied with the system design and architecture as it is well planned for the development.
System development	Domain Experts	System development was done well to perform the deception detection process.
	Industrial Experts	Since the input of the system is interrogation videos, system model is trained with video dataset where videos are converted to images and used. Model is trained with trial dataset using supervised learning approach. Backend and frontend development is also done perfectly.
	Researchers	Development of the system was completed according to the design and architecture of the system.
System prototype	Domain Experts	System prototype is well developed and easy to use for the detectives who do not have much technological knowledge. It is a great help for deception detection in the interrogation process.
	Industrial Experts	Prototype of the system works well and satisfies the users providing deception detection results for user uploaded interrogation videos.
	Researchers	According to the prototype demo, system is a perfectly developed with all the necessary functionalities.
UIUX aspect of application	Domain Experts	Simple and a perfect user interface was provided to the end user where users can easily upload a video and get deception results.

	Industrial Experts	UIUX aspect of the application is well thought and developed because users can easily navigate through the application and use the functionality.
	Researchers	Impressive user interface and user experience that users will be highly satisfied with.

Table 56 - Qualitative Analysis Evaluation

8.7 Limitations of Evaluation

Evaluators were asked about the limitations of the system they have identified. Feedback received on the system limitations are mentioned below.

Evaluator Category	Feedback for limitation
Domain Experts	Domain experts' evaluation results were positive about the system because currently Sri Lankan police or Crime Investigation Department do not use Automated Deception Detection systems. Only limitation identified by the domain experts were that currently system only focuses on facial indicators. The feedback was to analyze more non-verbal deceptive indicators. As future enhancements domain experts expected a real-time version of the system.
Industrial Experts	One of the feedback items received from industrial experts were to use an unsupervised learning approach so that author can use unstructured data for training purposes of the model. Also, received feedback that interrogation video upload time increases when a larger file is uploaded.
Researchers	Researchers were overly satisfied with the research carried out and the feedback received on limitations was to research more on other non-verbal deceptive indicators which plays a vital role in deception detection and include them in the future.

Table 57 - Limitations of Evaluation

8.8 Evaluation of Functional Requirements

ID	Requirement	Priority Level	Evaluation
FR1	User should be able to login to the system	M	**Implemented**

FR2	User should be able to upload the interrogation video	M	**Implemented**
FR3	User should be able to proceed the deception detection by clicking start button	M	**Implemented**
FR4	User should be able to see the deception results	M	**Implemented**
FR5	System should be able to identify the uploaded file is in video file format and display error when different file types are uploaded	M	**Implemented**
FR6	System should be able to identify human face from the interrogation video	M	**Implemented**
FR7	System should be able to identify deceptive facial indicators including facial cues and chirality	M	**Implemented**
FR8	System should be able to output the deceptive video segments	S	**Not Implemented**
FR9	System should be able to display deceptive indicators	M	**Implemented**

Table 58 - Evaluation of Functional Requirements

$$\text{Achieved Requirements} = \frac{\text{Number of requirements passed}}{\text{Number of total requirements}} = \frac{8}{9} = 88.8\%$$

Author was able to implement all the functional requirements with the highest priority level. A single functional requirement with the medium priority was not able to implement. Although there were many difficulties in implementing, author was able to implement the fully functional system which was proposed.

8.9 Evaluation of Non-Functional Requirements

ID	Requirement	Description	Evaluation
NFR1	Accuracy	Accuracy of the system should be to the point when detecting deception and system should not indicate normal behavior as deceptive.	**Implemented**
NFR2	Performance	System should perform well without any issues or lags as the video analysis should happen without much delay.	**Implemented**
NFR3	Usability	System should be user friendly and easy to use as the target users who are detectives are usually not from a technological background	**Implemented**
NFR4	Reliability	System should be reliable for the detectives to detect deceptions of the suspects as it is a key process carried out in interrogations	**Implemented**
NFR5	Maintainability	System should be easily supported, enhanced, and should have the ability to repair as the system will have further improvements	**Implemented**

Table 59 - Evaluation of Non-Functional Requirements

$$\text{Achieved Requirements} = \frac{\text{Number of requirements passed}}{\text{Number of total requirements}} = \frac{5}{5} = 100\%$$

All the non-functional requirements were implemented reassuring the user experience of the system. Accuracy, performance, usability, reliability and maintainability were taken into consideration when implementing the system.

8.10 Chapter Summary

The evaluation of the system which was conducted by selected evaluators and by the author herself was discussed in this chapter. Starting from evaluation criteria identification, author has explained the self-evaluation, evaluator selection, qualitative evaluation results and feedback received from selected evaluators, evaluation of limitations, evaluation of functional and non-functional requirements. Successful completion of the evaluation chapter will be a supporting the author in concluding the project in next conclusion chapter.

CHAPTER 10: CONCLUSION

10.1 Chapter Overview

Evaluation results from domain experts, industrial experts and author for **Caught in a Lie** system was discussed in the previous chapter, reaching the conclusion of the project. Achievement of research aims and objectives, research contribution, use of existing and new skills, achievement of learning outcomes, problems and challenged face, research limitations and future enhancements will be thoroughly explored in this chapter.

10.2 Achievement of Research Aims & Objectives

10.2.1 Aim of the Project

The aim of this research is to design, develop and evaluate an automated deception detection system which acts as a deception detection detective that analyses human facial cues and facial chirality to identify subject's truthful or deceptiveness in order to detect deception in the interrogation process.

The aim of the project was successfully achieved by conducting design, development and evaluation of the project. An automated deception detection system using human facial cues and facial chirality was successfully created by the author. System acts as a detective and identify truthfulness and deceptiveness of the suspect in interrogation. To the best of the author's knowledge this system will be the first to take in facial chirality analysis as a feature in order to detect deception.

10.2.2 Completion of Objectives of the Project

Research Objectives	Completion Status
Problem Identification	Completed
Literature Review	Completed
Data Gathering and Analysis	Completed
Research Design	Completed
Implementation	Completed
Testing and Evaluation	Completed

Table 60 - Completion of Project Objectives

10.3 Utilization of Knowledge from the Course

Module	Description

Programming Principles I & II	Programming principles and fundamental programming techniques were taught in this model which layered a basic infrastructure. Usage of programming languages assisted author to choose suitable languages for system developments.
Object Oriented Programming	Object oriented concepts were educated in this module which was used to achieve the class structure in the deception detection model.
Web Design and Development, Server-side Web Development	Basic web development and design skills were gained through these modules which assisted in developing the web application for the system. As flask framework was used, HTML, CSS, Bootstrap knowledge received from these modules were used to successfully implement the application.
Software Development Group Project	Software development group project layered a foundation for the final year project. SDGP report structure and the prototype implementation was indistinguishable to the final thesis and prototype. This module was a great learning practice for the final year project.
Algorithms, Advanced Analytics	Modules provided a better understanding about algorithms and data structures with time complexities for algorithms which was used in finding better algorithms for project development.
Database Systems	Database systems with data storing mechanisms were introduced in this module which later used in the project to store user details, interrogation videos and deceptive results.

Table 61 - Utilization of Knowledge from the Course

10.4 Use of Existing Skills

Many skills were required to successfully develop the system. Author has received and improved several programming skills throughout the degree program. Skills are as follows,

Python Machine Learning – Author gained basic python skills in the first year of studies in degree program and later improved machine learning skills in python through software development group project as development of machine learning section of the project was assigned to the author. Apart from that, python machine learning courses were followed using Coursera and Udemy to improve skills.

Web Development – Web application development skills were initially received through web design and development and server-side web development modules. Author was engaged in we application development of a project in her internship which improved skills in web application development.

10.5 Use of New Skills

Flask Framework – Flask framework was for the web application development in python. Author did not have knowledge in flask framework but with the existing knowledge of HTML, CSS, Bootstrap, and python author was able to learn flask framework and use it for the web application development of the project.

Deep Learning – Deep learning skills were a new skills author learnt in order to successfully develop the deception detection model. In order to become skilled in deep learning, courses were followed through edx, Udemy and Coursera.

Domain Knowledge – Author was able to receive knowledge on the domain specially in the area of criminal investigation and deception detection. Interviews with domain experts assisted in gaining a vivid understanding of the how interrogation process is carried out in interrogations and how manual deception detection is done which was a great help for problem identification, requirement gathering and evaluation of the project.

10.6 Achievement of Learning Outcomes

Learning Outcomes	**Achievement**
LO1	Appropriate problem selection with a good scope was done as the first step of the project. Then, appropriate methods, techniques which are ML, SVM were selected and applied for the project. Problem of deception detection was solved using them.
LO2	Project plan was developed, and tasks were divided within the timeframe of the project. Work breakdown structure was created to breakdown work accordingly and Gantt chart was created to illustrate the project schedule with timelines.
LO3	Requirement elicitation was conducted to gather requirements by following requirement elicitation methods which are interviews, prototyping, literature review, observations. Functional and non-functional requirements of the system were identified by analyzing gathered requirements.

LO4	Information on the project topic were collected by conducting research and analysis of existing work. This was helped to identify most suitable approaches, algorithms, and libraries to solve the identified problem.
LO5	Existing skills which are programming principles, OOP skills with knowledge of python programming language and machine learning algorithms and libraries were improved. New skills in flask web framework, deep learning algorithms were developed in order to produce deliverable of the project.
LO6	Identification of any legal, social, ethical, or professional issues related to the project were conducted and required steps were taken to avoid any issues
LO7	System prototype was implemented in order to solve the proposed, designed solution. Prototyping phase was an experimental study which implemented a prototype for the automated deception detection system.
LO8	Well-structured report was written covering all the chapters which are introduction, literature review, methodology, software requirement specification, social legal and ethical and professional issues, system architecture and design, implementation, testing, evaluation and conclusion. Report writing and structuring skills were improved through this phase.
LO9	A presentation was created for viva as for the defense of the project

Table 62 - Achievement of Learning Outcomes

10.7 Problems and Challenges Faced

Problems Faced	**Solution**
Extensive Project Scope	Deception detection has an extensive area of research because detectives consider many aspects such as verbal, non-verbal, non-linguistic indicators of a suspect in deception detection. Author broke down indicators into sub indicators and only considered non-verbal indicators in detecting deception.
Domain study was necessary	A detailed domain study was conducted in order to identify how detectives detects deception by analyzing suspects. Author had to understand current deception detection techniques and approaches hence a domain study was done by getting information from domain experts and watching documentaries on the domain studies.
Vast learning curve	Learning curve was vast because author had to learn deep learning, image processing in detail in order to successfully develop the project.

	Therefore, author spent large amount of time on learning and improving these skills and also developing the prototype.
Limited time for completion	Limited time was allocated for the completion of the project. Therefore, project scope was refined according to the timelines by only developing important requirements in the prototype.
Hardware requirements	Deep learning model training required high CPU and GPU requirements for the model to be trained and tested efficiently. Author used a machine with AMD Ryzen 7 CPU and Nvidia GeForce RTX3050 GPU which overcame issues.

Table 63 - Problems and Challenges Faced

10.8 Limitations of the Research

Identified limitations of the research are as follows:

- Project scope was only focused on non-verbal indicators of human face which are human facial cues and facial chirality. Project could be considering other non-verbal cues such as head posture, body posture and body movements in detecting deception.
- Currently system detects deception in interrogation videos nevertheless system could be implemented for real-time deception detection of suspects while interrogating.
- Model was only trained using real-life trial dataset which is a video dataset of 121 videos of court trials. System could be more effective if it was trained with interrogation videos in Sri Lanka but due to legal issues interrogation videos cannot be exposed to anybody therefore model was trained using publicly available dataset.

10.9 Future Enhancements

Since Sri Lankan Police Department currently does not use automated deception detection systems, **Caught in a Lie** will be the first system to be used. Therefore, author intends to overcome the limitations identified through evaluation, with enhancements in the future. Future enhancements are discussed below,

- Converting the system to a real-time deception detection system will be a major future enhancement.
- Training the model with interrogation videos in Sri Lanka will provide effective results as the input videos of the system will be similar to training data.

- Considering non-verbal indicators such as head posture, body posture and body movements which covers flight or fight, and perception management aspects of the suspect will be a necessary future enhancement for better deception results.
- It is better to have facial chirality model results being displayed in the application with the chiral detection of human face which will be clearer for detectives to identify deception and even learn from the system.

10.10 Research Contribution of Achievement

Gap identified in the research is within the domain of deception detection in interrogation processes carried out in criminal investigations. Currently, police officers are using manual deception detection approaches in Sri Lanka which are time consuming, require specialized detectives and less accurate due to human bias results. Researchers have come up with automated deception detection approaches to solve the issues in manual deception detection approach. However, currently available automated deception detection systems also lack validity and accuracy. Therefore, author has come up with an automated deception detection system using significant non-verbal cues which are human facial cues and facial chirality to detect deception. Since the system does not have any physical contact with the suspect, it has been more effective and accurate deception results have been produced. According to the knowledge of the author, **Caught in a Lie** is the first to use facial chirality model in automated deception detection. Research gap was successfully fulfilled by proposing and developing **Caught in a Lie** system. This research will be a new direction in the area of deception detection specially in Sri Lanka, and the facial chirality model will be great research even for different other domains. Future enhancements of the system will provide a more effective real-time automated deception detection system with better performance.

10.11 Concluding Statement

Caught in a Lie project was successfully completed by accomplishing research aim, objectives while fulfilling the research gap and contributing to the body of knowledge. Author was able to identify the importance of **Caught in a Lie** by conducting evaluations and getting feedback from domain experts. This automated deception detection system will be a huge impact on criminal investigation in Sri Lanka. It is evident that, system will be a great help for detectives to detect deception in interrogations with less time and effort. Author is well pleased about succeeding the project, **Caught in a Lie** since it contributes to improve one of the busiest domains in Sri Lanka, criminal investigation.

Appendices

Reference

Lawhandbook.sa.gov.au. 2021. *What is a Crime?*. [online] Available at: <https://lawhandbook.sa.gov.au/ch12s01.php> [Accessed 4 September 2021].

Encyclopedia Britannica. 2021. *Criminal investigation*. [online] Available at: <https://www.britannica.com/topic/criminal-investigation> [Accessed 6 September 2021]. Encyclopedia Britannica. 2021. *lie detector | Definition, Accuracy, & Facts*. [online] Available at: <https://www.britannica.com/technology/lie-detector> [Accessed 8 September 2021].

Osac.gov. 2021. *Working Together to Protect U.S. Organizations Overseas*. [online] Available at: <https://www.osac.gov/Country/SriLanka/Content/Detail/Report/02c5e26f-e7eb-44a8-b536-188dea9af39f> [Accessed 10 October 2021].

Youtube.com. 2021. *Former FBI Agent Explains Criminal Profiling | Tradecraft | WIRED*. [online] Available at: <https://www.youtube.com/watch?v=LHqbfJ3tq4Y&list=WL&index=13> [Accessed 27 October 2021].

Lakshan, I., Wickramasinghe, L., Disala, S., Chandrasegar, S. and Haddela, P., 2019. Real Time Deception Detection for Criminal Investigation. *2019 National Information Technology Conference (NITC)*,.

Chou, C., Huang, Y. and Ho, S., 2019. *Blink Detection Using Facial Landmark Blink Detector and Multi-Layer Perceptron*.

Zhang, J., Zhang, J., Ren, H., Liu, Q., Du, Z., Wu, L., Sai, L., Yuan, Z., Mo, S. and Lin, X., 2021. A Look Into the Power of fNIRS Signals by Using the Welch Power Spectral Estimate for Deception Detection. *Frontiers in Human Neuroscience*, 14.

Iacob, D. and Tapus, A., 2021. *First Attempts in Deception Detection in HRI by using Thermal and RGB-D cameras*. [online] Hal.archives-ouvertes.fr. Available at: <https://hal.archives-ouvertes.fr/hal-01840122> [Accessed 11 September 2021].

2019. *2019 8th International Conference on Affective Computing and Intelligent Interaction (ACII)*. Institute of Electrical and Electronics Engineers.

Khan, W., Crockett, K., O'Shea, J., Hussain, A. and Khan, B., 2021. Deception in the eyes of deceiver: A computer vision and machine learning based automated deception detection. *Expert Systems with Applications*, 169, p.114341.

Mathur, L. and Matarić, M., 2021. *Affect-Aware Deep Belief Network Representations for Multimodal Unsupervised Deception Detection*. [online] arXiv.org. Available at: <http://arxiv.org/abs/2108.07897> [Accessed 13 September 2021].

Belavadi, V., Zhou, Y., Bakdash, J., Kantarcioglu, M., Krawczyk, D., Nguyen, L., Rakic, J. and Thuriasingham, B., 2020. MultiModal Deception Detection: Accuracy, Applicability and Generalizability. *2020 Second IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA),*.

Research-Methodology. 2021. *Research Philosophy - Research Methodology*. [online] Available at: <https://research-methodology.net/research-philosophy/> [Accessed 13 September 2021].

Research-Methodology. 2021. *Positivism - Research Methodology*. [online] Available at: <https://research-methodology.net/research-philosophy/positivism/> [Accessed 14 September 2021].

Chetty, P., 2021. *Importance of research approach in a research*. [online] Project Guru. Available at: <https://www.projectguru.in/selecting-research-approach-business-studies/> [Accessed 14 September 2021].

Professionalqa.com. 2021. *Iterative Model: Advantages and Disadvantages |Professionalqa.com*. [online] Available at: <https://www.professionalqa.com/iterative-model> [Accessed 14 September 2021].

Plato.stanford.edu. 2021. *The Definition of Lying and Deception (Stanford Encyclopedia of Philosophy)*. [online] Available at: <https://plato.stanford.edu/entries/lying-definition/#:~:text=A%20lie%20is%20an%20untruthful,involves%20an%20intention%20to%20deceive.> [Accessed 27 October 2021].

Paul Ekman Group. 2021. *Deception Detection | How to Tell If Someone is Lying*. [online] Available at: <https://www.paulekman.com/deception/deception-detection/> [Accessed 27 October 2021].

Mayoclinic.org. 2021. *EEG (electroencephalogram) - Mayo Clinic*. [online] Available at: <https://www.mayoclinic.org/tests-procedures/eeg/about/pac-20393875> [Accessed 7 October 2021].

Encyclopedia Britannica. 2021. *Crime | law*. [online] Available at: <https://www.britannica.com/topic/crime-law> [Accessed 9 October 2021].

Jayathunga, N., 2011. A Sociological Study of the Homicide in Sri Lanka: A Case Study in Rathnapura Secretariat Division. *Sabaragamuwa University Journal*, 9(1), p.45.

Osac.gov. 2021. *Working Together to Protect U.S. Organizations Overseas*. [online] Available at: <https://www.osac.gov/Country/SriLanka/Content/Detail/Report/02c5e26f-e7eb-44a8-b536-188dea9af39f> [Accessed 10 October 2021].

Encyclopedia Britannica. 2021. *Criminal investigation*. [online] Available at: <https://www.britannica.com/topic/criminal-investigation> [Accessed 6 September 2021].

WIRED, 2021. *Former FBI Agent Explains How to Detect Lying & Deception | Tradecraft | WIRED*. [video] Available at: <https://www.youtube.com/watch?v=tpJcBozuF6A&list=WL&index=11&t=55s> [Accessed 10 October 2021].

Krishna, G., Sai Kumar N V, C., Tushal, B., Gopal, A. and Lövström, B., 2014. *Micro-Expression Extraction For Lie Detection Using Eulerian Video (Motion and Color) Magnification Submitted By*. [online] Karlskrona: BLEKINGE INSTITUTE OF TECHNOLOGY, pp.1-72. Available at: <http://www.bth.se/ing> [Accessed 11 October 2021].

Belavadi, V., Zhou, Y., Bakdash, J., Kantarcioglu, M., Krawczyk, D., Nguyen, L., Rakic, J. and Thuriasingham, B., 2020. MultiModal Deception Detection: Accuracy, Applicability and Generalizability. *2020 Second IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA),*.

Openface-api.readthedocs.io. 2021. *OpenFace API Documentation — OpenFace API Docs 0.1.1 documentation*. [online] Available at: <https://openface-api.readthedocs.io/en/latest/> [Accessed 19 October 2021].

Jaiswal, M., Tabibu, S. and Bajpai, R., 2016. The Truth and Nothing But the Truth: Multimodal Analysis for Deception Detection. *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW),*.

Krishna, G., Sai Kumar N V, C., Thushal, B., Gopal, A. and Lövström, B., 2014. *Micro-Expression Extraction For Lie Detection Using Eulerian Video (Motion and Color) Magnification Submitted By*. Master Thesis Electrical Engineering. [online] Karlskrona: Blekinge Institute of Technology, pp.1-72. Available at: <http://www.bth.se/ing> [Accessed 22 October 2021].

n.d. *2019 8th International Conference on Affective Computing and Intelligent Interaction (ACII)*. Institute of Electrical and Electronics Engineers.

Chou, C., Huang, Y. and Ho, S., 2019. *Blink Detection Using Facial Landmark Blink Detector and Multi-Layer Perceptron*. Taoyuan, Taiwan: Chung Cheng Institute of Technology, National Defense University.

Avola, D., Cinque, L., Foresti, G. and Pannone, D., 2019. Automatic Deception Detection in RGB videos using Facial Action Units. *Proceedings of the 13th International Conference on Distributed Smart Cameras*,.

Iacob, D. and Tapus, A., 2021. *First Attempts in Deception Detection in HRI by using Thermal and RGB-D cameras*. [online] Hal.archives-ouvertes.fr. Available at: <https://hal.archives-ouvertes.fr/hal-01840122> [Accessed 11 September 2021].

Ngô, M., Mandira, B., Yılmaz, S., Heij, W., Karaoglu, S., Bouma, H., Dibeklioglu, H. and Gevers, T., 2021. *Deception Detection by 2D-to-3D Face Reconstruction from Videos*. [online] arXiv.org. Available at: <http://arxiv.org/abs/1812.10558> [Accessed 25 October 2021].

Avola, D., Cascio, M., Cinque, L., Fagioli, A. and Foresti, G., 2020. LieToMe: An Ensemble Approach for Deception Detection from Facial Cues. *International Journal of Neural Systems*, 31(02), p.2050068.

scikit-learn. 2021. *1.4. Support Vector Machines*. [online] Available at: <https://scikit-learn.org/stable/modules/svm.html> [Accessed 31 October 2021].

Analytics Vidhya. 2021. *SVM | Support Vector Machine Algorithm in Machine Learning*. [online] Available at: <https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/> [Accessed 31 October 2021].

Medium. 2021. *Top 5 advantages and disadvantages of Decision Tree Algorithm*. [online] Available at: <https://dhirajkumarblog.medium.com/top-5-advantages-and-disadvantages-of-decision-tree-algorithm-428ebd199d9a> [Accessed 31 October 2021].

OpenGenus IQ: Computing Expertise & Legacy. 2021. *Convolutional Neural Network (CNN) questions*. [online] Available at: <https://iq.opengenus.org/cnn-questions/> [Accessed 31 October 2021].

OpenGenus IQ: Computing Expertise & Legacy. 2021. *Convolutional Neural Network (CNN) questions*. [online] Available at: <https://iq.opengenus.org/cnn-questions/> [Accessed 31 October 2021].

Techopedia.com. 2021. *What is a Multilayer Perceptron (MLP)? - Definition from Techopedia*. [online] Available at: <https://www.techopedia.com/definition/20879/multilayer-perceptron-mlp> [Accessed 31 October 2021].

scikit-learn. 2021. *1.17. Neural network models (supervised)*. [online] Available at: <https://scikit-learn.org/stable/modules/neural_networks_supervised.html> [Accessed 31 October 2021].

Akkaya, B. and Çolakoğlu, N., 2019. Comparison of Multi-class Classification Algorithms on Early Diagnosis of Heart Diseases.

Built In. 2021. *A Guide to RNN: Understanding Recurrent Neural Networks and LSTM Networks*. [online] Available at: <https://builtin.com/data-science/recurrent-neural-networks-and-lstm> [Accessed 31 October 2021].

(RNN), R., 2021. *Recurrent Neural Networks (RNN) | Working | Steps | Advantages*. [online] EDUCBA. Available at: <https://www.educba.com/recurrent-neural-networks-rnn/> [Accessed 31 October 2021].

Medium. 2021. *Deep Learning—Deep Belief Network (DBN)*. [online] Available at: <https://medium.datadriveninvestor.com/deep-learning-deep-belief-network-dbn-ab715b5b8afc> [Accessed 31 October 2021].

Ockenden, M., Tych, W., Beven, K., Collins, A., Evans, R., Falloon, P., Forber, K., Hiscock, K., Hollaway, M., Kahana, R., Macleod, C., Villamizar, M., Wearing, C., Withers, P., Zhou, J., Benskin, C., Burke, S., Cooper, R., Freer, J. and Haygarth, P., 2017. Prediction of storm transfers and annual loads with data-based mechanistic models using high-frequency data. *Hydrology and Earth System Sciences*, 21(12), pp.6425-6444.

Cmusatyalab.github.io. 2021. *OpenFace*. [online] Available at: <https://cmusatyalab.github.io/openface/> [Accessed 1 November 2021].

Algorithmia Blog. 2021. *Understanding Open-Source Facial Recognition Through OpenFace*. [online] Available at: <https://algorithmia.com/blog/understanding-facial-recognition-openface> [Accessed 1 November 2021].

Baltrusaitis, T., Robinson, P. and Morency, L., 2016. OpenFace: An open source facial behavior analysis toolkit. *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*,.

Personalpages.manchester.ac.uk. 2021. *Constrained Local Models*. [online] Available at: <https://personalpages.manchester.ac.uk/staff/timothy.f.cootes/Models/clm.html> [Accessed 1 November 2021].

Zadeh, A., Baltrušaitis, T. and Morency, L., 2021. *Convolutional Experts Constrained Local Model for Facial Landmark Detection*. [online] arXiv.org. Available at: <https://arxiv.org/abs/1611.08657> [Accessed 1 November 2021].

Azure.microsoft.com. 2021. *Transparency Note: Azure Cognitive Services Face API*. [online] Available at: <https://azure.microsoft.com/en-in/resources/transparency-note-azure-cognitive-services-face-api/> [Accessed 1 November 2021].

LII / Legal Information Institute. 2021. *Criminal Law*. [online] Available at: <https://www.law.cornell.edu/wex/criminal_law> [Accessed 5 November 2021].

Innocence Project. 2021. *Police Interviews versus Interrogation: 'Fairbanks Four' Case Illustrates Important Distinction - Innocence Project*. [online] Available at: <https://innocenceproject.org/police-interviews-versus-interrogation-fairbanks-four-case-illustrates-important-distinction/#:~:text=Interrogations%2C%20on%20the%20other%20hand,the%20suspect%20to%20the%20crime.&text=His%20suggestive%20and%20coercive%20methods,to%20do%20with%20the%20crime.> [Accessed 6 November 2021].

Paul Ekman Group. 2021. *Deception Detection | How to Tell If Someone is Lying*. [online] Available at: <https://www.paulekman.com/deception/deception-detection/> [Accessed 6 November 2021].

Burgoon, J., 2015. Deception Detection Accuracy. *The International Encyclopedia of Interpersonal Communication*, pp.1-6.

Software Testing Help. 2022. *Top 10 Most Common Requirements Elicitation Techniques*. [online] Available at: <https://www.softwaretestinghelp.com/requirements-elicitation-techniques/> [Accessed 6 December 2022].

Pytorch.org. 2022. *Learn the Basics — PyTorch Tutorials 1.10.1+cu102 documentation*. [online] Available at: <https://pytorch.org/tutorials/beginner/basics/intro.html> [Accessed 13 January 2022].

Code.visualstudio.com. 2022. *Visual Studio Code Frequently Asked Questions*. [online] Available at: <https://code.visualstudio.com/docs/supporting/FAQ> [Accessed 15 January 2022].

JetBrains. 2022. *PyCharm: the Python IDE for Professional Developers by JetBrains*. [online] Available at: <https://www.jetbrains.com/pycharm/> [Accessed 16 January 2022].

TensorFlow. 2022. *Get started with TensorBoard | TensorFlow*. [online] Available at: <https://www.tensorflow.org/tensorboard/get_started> [Accessed 2 March 2022].

Google Developers. 2022. *Classification: ROC Curve and AUC | Machine Learning Crash Course | Google Developers*. [online] Available at: <https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc#:~:text=AUC%20represents%20the%20probability%20that,has%20an%20AUC%20of%201.0.> [Accessed 24 February 2022].

Appendix A

Existing Work

Citation	(Lakshan et al., 2019)
Brief	Identify guilt of an accused criminal in interrogation videos using EEG signals.
Technology	MUSE 2 headband was used for emotion detection with EEG, KNN, CNN, Logistic Regression, Random Forest for classification
Improvement	• Real-time lie detection using electroencephalogram (EEG) • Emotion and attention detection using ML tools and techniques
Limitations	• Act differently for people with health issues • Participants should avoid movements and excessive facial muscle movements throughout the interrogation process

Table 64 - Existing Work Table 1

Citation	(Chou, Huang and Ho, 2019)
Brief	Facial landmark blink detector (FLBD) to detect blink behaviour using the facial landmarks
Technology	The multi-layer perceptron (MLP) with hidden layers, neurons, and activation functions used as training and testing phases classifier.
Improvement	• FLBD is fast and accurate • Can support real-time
Limitations	• Blink rates can be differed from people • Reliability issues as deception cannot be detected alone by blink rate

Table 65 - Existing Work Table 2

Citation	(Zhang et al., 2021)
Brief	Brain activation differences are examined to detect deceptive behavior using fNIRS signals
Technology	CW fNIRS system was used to perform fNIRS recordings PWELCH function in MATLAB was used to calculate the Welch power spectral density (PSD) classifier for training and testing phases
Improvement	Portable operation can be done Comfortable and quiet way with fewer body constraints

Limitations	• Act differently for people with health issues • Require sophisticated equipment setup, overt in nature and require trained operator to use

Table 66 - Existing Work Table 3

Citation	(Iacob and Tapus, 2021)
Brief	thermal and RGB-D imaging was used to detect deceptiveness and truthfulness of a person, based on an evaluation of their physiological state.
Technology	Developed in C++ using ROS and OpenCV, CLM Face Tracker
Improvement	• Estimated heart rate using facial green FFT • Estimated respiratory rate using nasal temperature FFT • Interocular and nasal relative temperature • Eye vertical relative position • Face distance with respect to the cameras
Limitations	• During the interrogation, perceived level of stress was above average • Lower accuracy of the non-invasive measurements • Measurement errors induced by the measurement techniques and the hardware limitations

Table 67 - Existing Work Table 4

Citation	(2019 8th International Conference on Affective Computing and Intelligent Interaction (ACII), 2019)
Brief	Framework to differentiate imagination vs remembering when interviewing using facial expression sequences
Technology	Baseline Knowledge Model to classify imagining vs. remembering given the facial expressions expressed while interviewing Weighted Fusion Model, SVM
Improvement	Transfer learning approach based on non-verbal features in deception
Limitations	Less accuracy (60%)

Table 68 - Existing Work Table 5

Citation	(Khan et al., 2021)

Brief	Identify features which provide significant clues for automated deception detection using fine-grained level eyes and facial micro-movements.
Technology	Artificial neural networks, random forests, and support vector machines, Haar Cascade algorithm
Improvement	• Features extraction using Silent Talker system • Multifaceted interactions between micro gestures to determine truthful or deceptive behaviour • Encode local appearance of objects using two dimensional Haar functions and Haar-like features
Limitations	Improvements in classification accuracy

Table 69 - Existing Work Table 6

Citation	(Mathur and Matarić, 2021)
Brief	Detect deception in videos using unsupervised approach without labels
Technology	Evaluate representations during human centred modelling tasks for speech and emotions using unsupervised GMMs
Improvement	Learn representations of deceptive and truthful behaviour effectively for unsupervised deception detection using DBN-based models
Limitations	Low performance

Table 70 - Existing Work Table 7

Citation	(Belavadi et al., 2020)
Brief	Feasibility inspection of AI/ML techniques application in lie detection in video using multiple datasets
Technology	CNN, OpenFace API, Random Forest, Multiple layer instance
Improvement	• Two video datasets - screening interviews, crime experiment • Deploy ML techniques to explore the reliability, generalizability, and applicability of identified algorithms in deception detection • Compare results to the state-of-the-art deception detection systems
Limitations	• Overfitting • Low accuracy and low performance

Table 71 - Existing Work Table 8

Gantt Chart

Task Name	Duration	Start	Finish
Project initiation	**91 days**	**Thu 7/1/21**	**Thu 11/4/21**
Selection of research problem	14 days	Thu 7/1/21	Tue 7/20/21
Mentor approval of the research	1 day	Wed 7/21/21	Wed 7/21/21
Finalizing the research domain	30 days	Thu 7/22/21	Wed 9/1/21
Identifying the research gap	15 days	Thu 9/2/21	Wed 9/22/21
Feasibility study on the identified gap	7 days	Thu 9/23/21	Fri 10/1/21
Identifying the existing/related work	14 days	Mon 10/4/21	Thu 10/21/21
Mentor and peer review of proposal	5 days	Fri 10/22/21	Thu 10/28/21
Self evaluation and finalizing proposal	4 days	Fri 10/29/21	Wed 11/3/21
Submission of Project Proposal	1 day	Thu 11/4/21	Thu 11/4/21
Literature Review	**213 days**	**Thu 7/1/21**	**Fri 4/22/22**
Identifying the components of LR	40 days	Thu 7/1/21	Wed 8/25/21
Gathering existing research articles	56 days	Thu 8/26/21	Thu 11/11/21
Reviewing related work	47 days	Fri 11/12/21	Mon 1/17/22
Preparing draft LR	42 days	Tue 1/18/22	Tue 3/15/22
Mentor and peer review	11 days	Wed 3/16/22	Wed 3/30/22
Self evaluation and finalizing LR	16 days	Thu 3/31/22	Thu 4/21/22
Submission of LR	1 day	Fri 4/22/22	Fri 4/22/22
Requirement Gathering	**52 days**	**Fri 11/5/21**	**Mon 1/17/22**
Analyzing current processes	6 days	Fri 11/5/21	Fri 11/12/21
Preparing questionnaires and distributing	2 days	Mon 11/15/21	Tue 11/16/21
Collecting feedback from the target audience	15 days	Wed 11/17/21	Tue 12/7/21
Analysing received feedback	7 days	Wed 12/8/21	Thu 12/16/21
Organizing data to be added to the SRS	10 days	Fri 12/17/21	Thu 12/30/21
Prepare SRS document	8 days	Fri 12/31/21	Tue 1/11/22
Design of the System Prototype	**26 days**	**Tue 1/18/22**	**Mon 2/21/22**
Identifying components of existing systems	14 days	Tue 1/18/22	Sat 1/29/22
Creating high level architecture	5 days	Sun 1/30/22	Thu 2/3/22
Mentor review of high level architecture	5 days	Thu 2/3/22	Wed 2/9/22
Preparing a draft of design specification	3 days	Wed 2/9/22	Mon 2/14/22
Mentor and peer review	8 days	Mon 2/14/22	Sun 2/20/22
Submission of design specification	1 day	Mon 2/21/22	Mon 2/21/22
Prototype Implementation	**91 days**	**Mon 2/21/22**	**Wed 4/20/22**
Implementing prototype	60 days	Mon 2/21/22	Fri 4/1/22
Reveiwing prototype features	30 days	Sat 4/2/22	Tue 4/19/22
Prototype demo to review	1 day	Wed 4/20/22	Wed 4/20/22
Testing and Evaluation	**30 days**	**Thu 3/17/22**	**Wed 4/27/22**
Identifying testing methodologies	2 days	Thu 3/17/22	Fri 3/18/22
Creating a test plan	2 days	Mon 3/21/22	Tue 3/22/22
Functional testing and evaluation	26 days	Thu 3/17/22	Thu 4/21/22

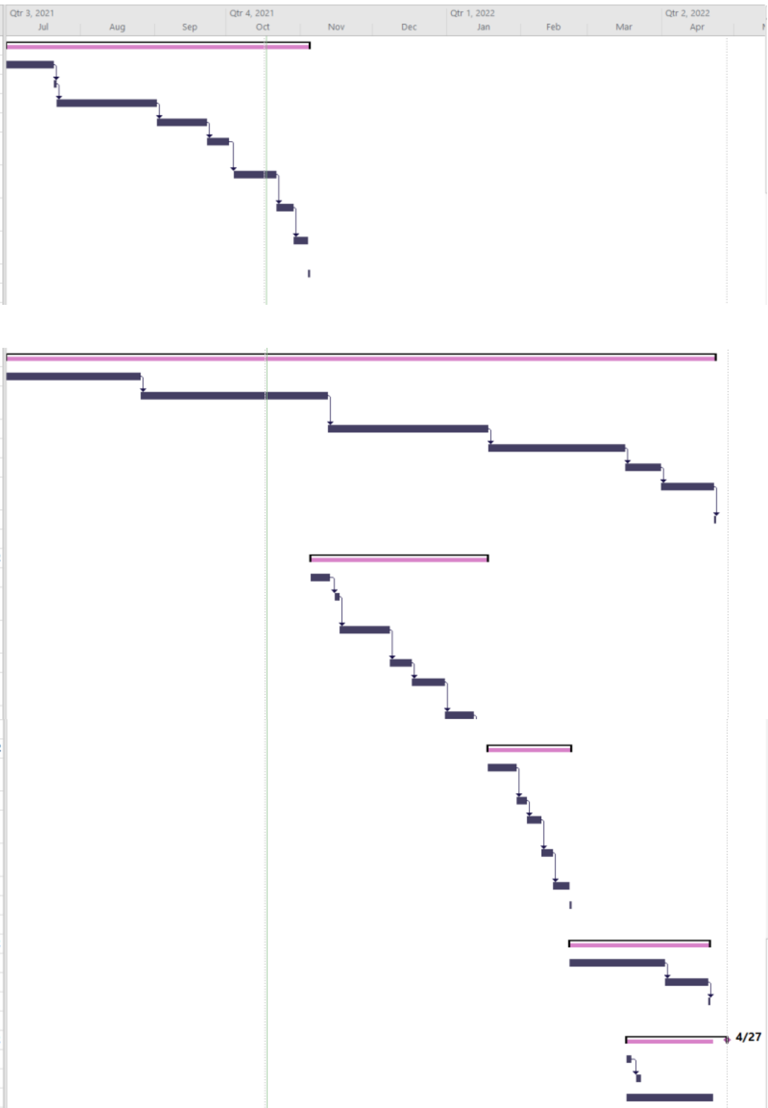

Figure 28 - Gantt Chart

Work Breakdown Structure

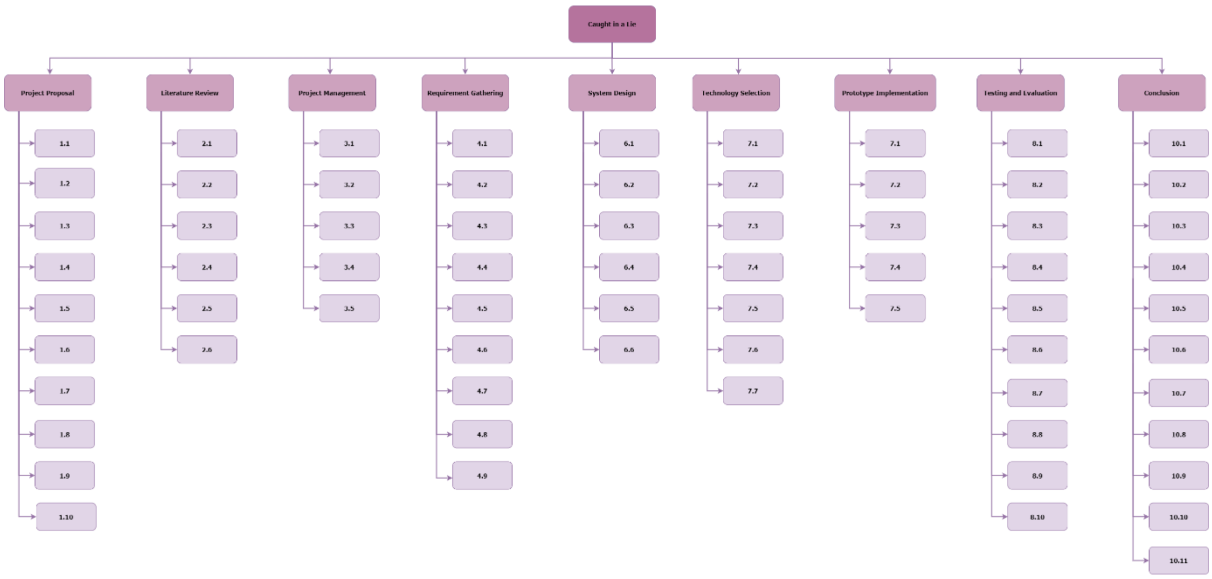

Figure 29 - Work Breakdown Structure

Appendix B

Activity Diagram

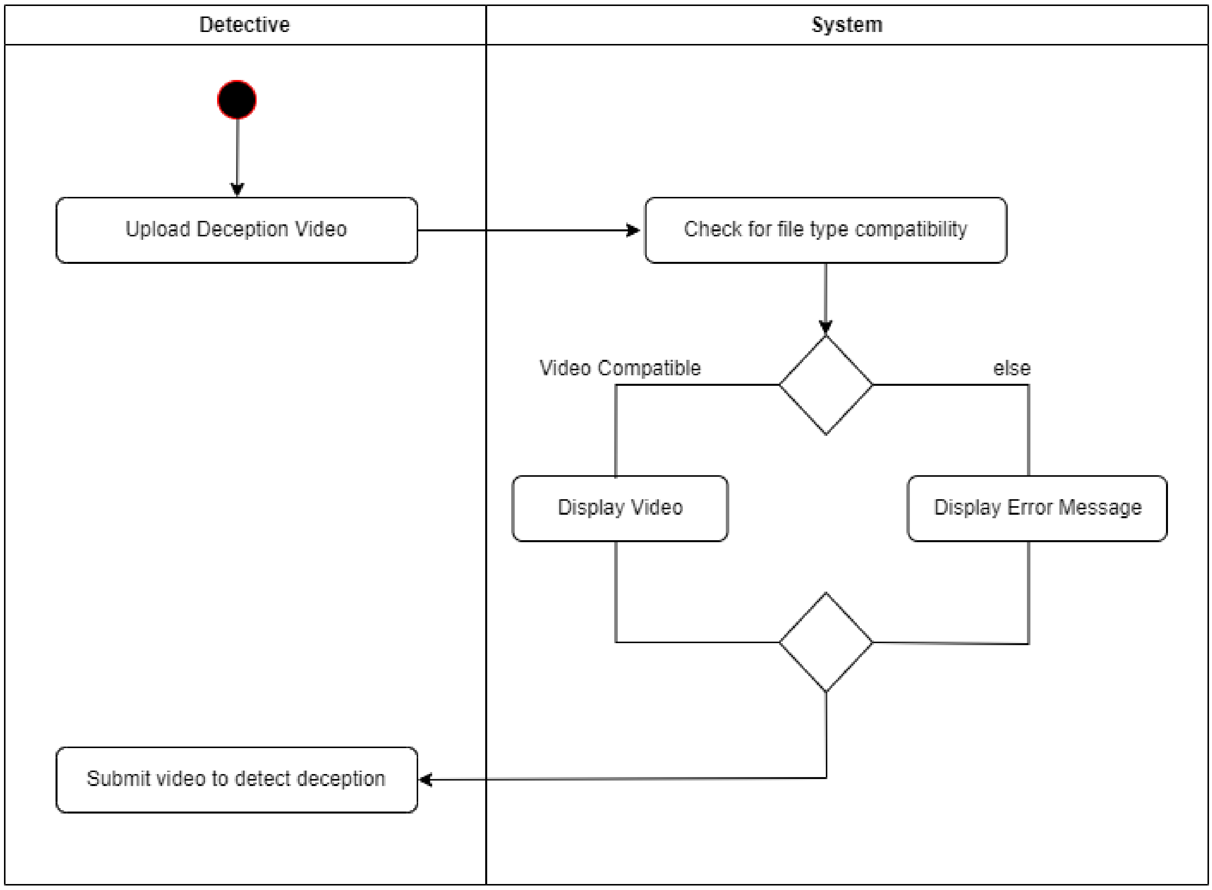

Figure 30 - Activity Diagram for Upload Video (Self Composed)

Appendix C

Wireframes

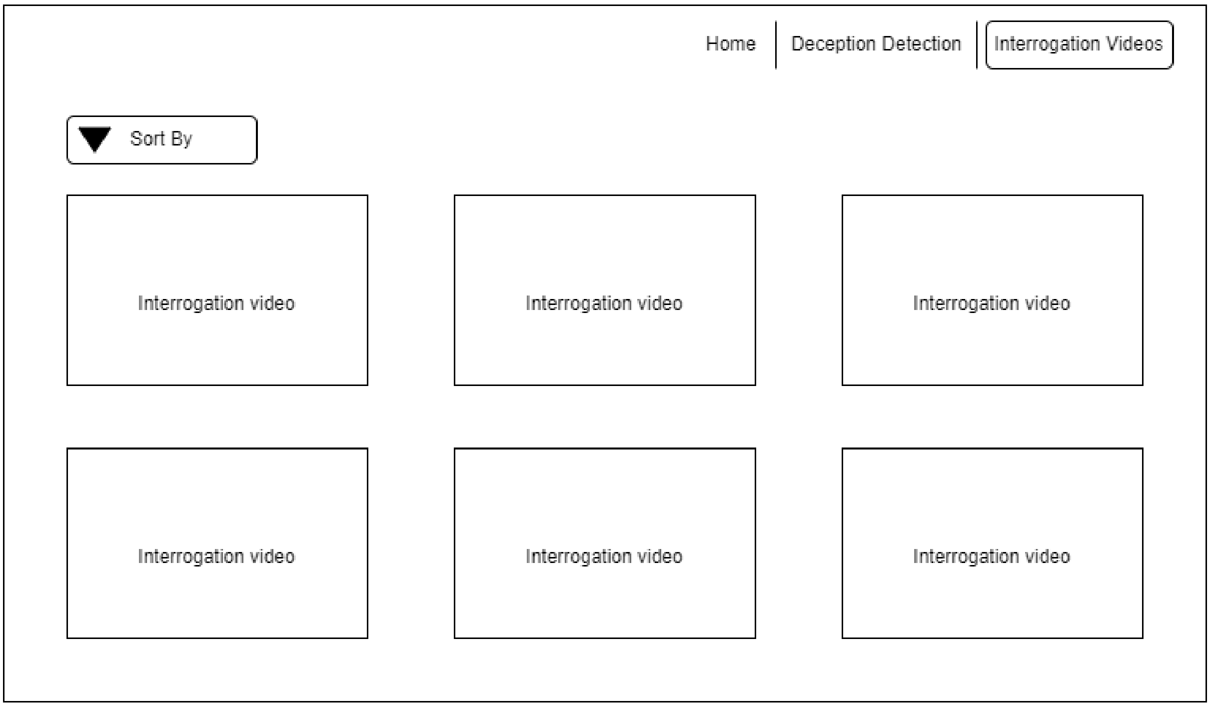

Figure 31 - Interrogation Videos Wireframe (Self Composed)

Appendix D

Utils.py

```python
import argparse
import math
import cv2 as cv
import numpy as np
import torch
from PIL import Image

from align_faces import get_reference_facial_points, warp_and_crop_face
from mtcnn.detector import detect_faces

def align_face(img_fn, facial5points,crop_size_h,crop_size_w):
    if type(img_fn) == str :
        raw = cv.imread(img_fn, True)
    else:
        raw = img_fn
    crop_size = (crop_size_h, crop_size_w)

    default_square = True
    inner_padding_factor = 0.1 #0.25
    outer_padding = (0, 0)
    output_size = (crop_size_h, crop_size_w)

    reference_5pts = get_reference_facial_points(
        output_size, inner_padding_factor, outer_padding, default_square)
    dst_img = warp_and_crop_face(raw, facial5points, reference_pts=reference_5pts, crop_size=crop_size)
    return dst_img

def get_face_attributes(full_path):
    try:
        img = Image.open(full_path).convert('RGB')
        bounding_boxes, landmarks = detect_faces(img)

        if len(landmarks) > 0:
            landmarks = [int(round(x)) for x in landmarks[0]]
            return True, landmarks

    except KeyboardInterrupt:
        raise
    except:
        pass
    return False, None

def select_central_face(im_size, bounding_boxes):
    width, height = im_size
    nearest_index = -1
    nearest_distance = 100000
    for i, b in enumerate(bounding_boxes):
        x_box_center = (b[0] + b[2]) / 2
        y_box_center = (b[0] + b[2]) / 2
        x_img = width / 2
        y_img = height / 2
        distance = math.sqrt((x_box_center - x_img) ** 2 + (y_box_center - y_img) ** 2)
        if distance < nearest_distance:
            nearest_distance = distance
            nearest_index = i

    return nearest_index
```

```python
def draw_bboxes(img, bounding_boxes, facial_landmarks=[]):
    for b in bounding_boxes:
        cv.rectangle(img, (int(b[0]), int(b[1])), (int(b[2]), int(b[3])), (255, 255, 255), 1)

    for p in facial_landmarks:
        for i in range(5):
            cv.circle(img, (int(p[i]), int(p[i + 5])), 1, (0, 255, 0), -1)

        break  # only first

    return img


def get_face_all_attributes(full_path):
    try:
        img = Image.open(full_path).convert('RGB')
        bounding_boxes, landmarks = detect_faces(img)

        if len(landmarks) > 0:
            i = select_central_face(img.size, bounding_boxes)
            return True, [bounding_boxes[i]], [landmarks[i]]

    except KeyboardInterrupt:
        raise
    except:
        pass
    return False, None, None
```

Figure 32 - Utils.py Code

Training

Train_test_split()

```python
import os
from random import shuffle

def train_test_split():
    Real_path = '../../dataset/video_chunks_image_cut_v2'
    data_list = []
    for i in os.listdir(Real_path):
        Real = os.path.join(Real_path, i)
        data_list.append(Real)
    if os.path.exists('../../dataset/train.txt'):
        os.remove('../../dataset/train.txt')
    if os.path.exists('../../dataset/test.txt'):
        os.remove('../../dataset/test.txt')
    shuffle(data_list)
    with open('../../dataset/train.txt', 'a') as train_data_txt:
        for i in range(int(len(data_list) * 0.8)):
            train_data_txt.write(data_list[i] +'\n')
    with open('../../dataset/test.txt', 'a') as test_data_txt:
        for i in range(int(len(data_list) * 0.2)):
            test_data_txt.write(data_list[int(len(data_list)*0.8) + i] +'\n')

def data_load():
    Real_path = '../../dataset/video_chunks_image_cut_v2'
    if os.path.exists('../../dataset/Real_life.txt'):
        os.remove('../../dataset/Real_life.txt')
    with open('../../dataset/Real_life.txt', 'a') as Real_txt:
        for i in os.listdir(Real_path):
            Real_txt.write(i +'\n')
data_load()
```

Figure 33 - Train_test_split Code

Train function

```python
def train(args, trainloader, vaildloader, model, optimizer, criterion):
    valid_f1_max = float('-inf')
    running_loss = 0
    DISFA_f1_max = float('-inf')
    mse_min = float('inf')
    mae_min = float('inf')
    metrics_func = metrics(args=args)
    for epoch in range(1, args.epochs + 1):
        model.train()
        learning_rate = adjust_learning_rate(args, optimizer, epoch)
        for idx, (inputs, labels) in enumerate(trainloader):
            device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
            inputs, labels = inputs.to(device), labels.to(device)
            if args.loss_func == 'MSE' or args.loss_func == 'L1':
                inputs, labels = inputs.type(torch.cuda.FloatTensor), labels.type(torch.cuda.FloatTensor)
            optimizer.zero_grad()
            logps = model(inputs, args)
            if args.loss_func == 'BCE':
                loss = criterion(input=logps, target=labels, weight=None)
            else:
                loss = criterion(logps, labels)
            loss.backward()
            optimizer.step()

            # Calculate F1 and accuracy
            p, r, F1, acc = metrics_func.metrics_equation(logps, labels,
                                                          intensity=(args.label_name == 'intensity_label'))
            if args.label_name == 'intensity_label':
                mse = metrics_func.MSE(logps, labels)
            else:
                mse = 'label is not intensity'
            if args.label_name == 'intensity_label':
                mae = metrics_func.MAE(logps, labels)
            else:
                mae = 'label is not intensity'

            running_loss += loss.item()
            if (idx % args.print_freq) == 0 or idx == (len(trainloader) - 1):
                print_str = str(datetime.datetime.today()) + '  |' + \
                            "Epoch: %d/%d   | " % (epoch, args.epochs) + \
                            "Batch: %s/%d   | " % (str(idx).zfill(len(str(len(trainloader)))), len(trainloader)) + \
                            "iter: %s/%d   | " % (
                            str((epoch - 1) * len(trainloader) + idx).zfill(len(str(args.epochs * len(trainloader)))),
                            args.epochs * len(trainloader)) + \
                            "lr: %0.6f   | " % (learning_rate) + \
                            "Train loss: %4.5f" % (running_loss)
                print(print_str)

        print_func(args, print_str, F1, acc, mse, mae)
        metrics_func.clear()
        running_loss = 0
```

Figure 34 - Train function

Config file

```python
class Config(object):
    # labels
    AU_labels_D1 = {0: 'au1', 1: 'au2', 2: 'au4', 3: 'au5', 4: 'au6', 5: 'au9', 6: 'au12', 7: 'au15', 8: 'au17',
                    9: 'au20', 10: 'au25', 11: 'au26',
                    'au1': 0, 'au2': 1, 'au4': 2, 'au5': 3, 'au6': 4, 'au9': 5, 'au12': 6, 'au15': 7, 'au17': 8,
                    'au20': 9, 'au25': 10, 'au26': 11
                    }

    AU_labels_D2 = {0: 'au1', 1: 'au2', 2: 'au3', 3: 'au4', 4: 'au5', 5: 'au6', 6: 'au7', 7: 'au8', 8: 'au9', 9: 'au10',
                    10: 'au11', 11: 'au12', 12: 'au13', 13: 'au14', 14: 'au15', 15: 'au16', 16: 'au17', 17: 'au18',
                    18: 'au19', 19: 'au20', 20: 'au21', 21: 'au22', 22: 'au23', 23: 'au24',
                    'au1': 0, 'au2': 1, 'au3': 2, 'au4': 3, 'au5': 4, 'au6': 5, 'au7': 6, 'au8': 7, 'au9': 8, 'au10': 9,
                    'au11': 10, 'au12': 11, 'au13': 12, 'au14': 13, 'au15': 14, 'au16': 15, 'au17': 16, 'au18': 17,
                    'au19': 18, 'au20': 19, 'au21': 20, 'au22': 21, 'au23': 22, 'au24': 23
                    }
    lie_labels = {0: 'truth', 1: 'lie', 'truth': 0, 'lie': 1}

    # GPU
    gpu_id = '0'

    # AU weights
    load_au_path = '../dataset/CK_LeftCamera_label'
    load_au_txt_path = '../dataset/train.txt'
    save_txt_path = '../dataset/CK_weight.txt'
    au_label_name = 'label'

    # Save dataset to text & Shuffle-
    isWriteTXT = False
    shuffle = True
    dataset_path = '../dataset/CK'
    seed = 20200702
    mode = 'cut'
    mode2 = 'folder'
    cut_ratio = 0.2
    cross = 3

    # Path of Dataset
    data_shuffle = True
    data_path = '/model/Chirality_Action/MSPL_YTD_images'

    # Model Parameter
    load_weight_path = '../dataset/CK_weight.txt'
    dataset_name = 'CK'
    isPretrain = True
    model_name = 'se_resnet50'
    pretrain_model = 'AU_0.4829_0.8912_116.pth'
    pretrained_weight = True
    num_classes = 12
    batch_size = 10
    epochs = 120
    loss_func = 'BCE'
    optimizer_name = 'adam'
    learning_rate = 1e-3
    weight_decay = 5e-4
    change_step = 20

    # Training Results
    print_freq = 500

    # Save Paths
    save_model = 'C:/Users/Chamudi/model-20220216T032754Z-001/model/Chirality_Action/model/%s_MLSM' % (model_name)
    save_path = '../../svm/Real-life_dataprocess'
```

Figure 35 - Config file Code

Testing

```
def test(args, model, test_loader):
    model.eval()
    metrics_func = metrics(args=args)
    with torch.no_grad():
        valid_loss = 0
        accuracy = 0
        for idx, (inputs, labels) in enumerate(test_loader):
            device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
            inputs, labels = inputs.to(device), labels.to(device)
            logps = model(inputs, args)
            # Calculate accuracy
            p, r, F1, acc = metrics_func.metrics_equation(logps, labels,
                                                          intensity=(args.label_name == 'intensity_label'))
            if args.label_name == 'intensity_label':
                mse = metrics_func.MSE(logps, labels)
            else:
                mse = 'label is not intensity'
            if args.label_name == 'intensity_label':
                mae = metrics_func.MAE(logps, labels)
            else:
                mae = 'label is not intensity'

        print_str = "TEST:"
        print_func(args, print_str, F1, acc, mse, mae)
```

Figure 36 - Testing Function Code

Appendix E

Competitive Benchmarking

Algorithms Used and Relevant Accuracy of Research

Automatic Deception Detection in RGB videos using Facial Action Units - Facial Action Units

Algorithm	Accuracy
Logistic Regression	54.86%
Random Forest	58,42%
SVM - Linear	65,56%
SVM - Sigmoid	71,99%
SVM - RBF	76,84%

Figure 37 - Competitive Benchmarking Proof 1

Deception Detection by 2D-to-3D Face Reconstruction from Videos

Model	Feature	Accuracy	Precision	Recall
Morales *et al.* [23] (DT)*	OpenFace features	0.55	0.54	0.50
Morales *et al.* [23] (RF)*	OpenFace features	0.50	0.45	0.25
Perez-Rosas *et al.* [25] (DT)*	Hand-labeled features	0.66	0.67	0.55
Perez-Rosas *et al.* [25] (RF)*	Hand-labeled features	0.67	**0.70**	0.55
Time-CNN [33]	FRC	0.69	0.65	0.77
Our RNN	FRC	**0.73**	0.68	**0.79**

Figure 38 - Competitive Benchmarking Proof 2

Facial Expression Based Imagination Index - Transfer Learning Approach to Detect Deception

Models	Accuracy	F1 Score
WFM (S)	0.68	0.68
WFM (P)	0.70	0.68
WFM	**0.70**	**0.70**
SVM (S)	0.52	0.55
SVM (P)	0.57	0.63
SVM	0.61	0.59
Human Performance	0.52	

Figure 39 - Competitive Benchmarking Proof 3

Deception Detection in Videos using the Facial Action Coding System

APPROACH	METHOD	CCR
CURRENT	**FACS WIH LSTM**	**89.49%**
[13]	DECISION TREES AND RANDOM FOREST	75.20%
[6]	SVM-RBF	76.84%
[12]	ADABOOST	88%

Figure 40 - Competitive Benchmarking Proof 4

LieToMe: An Ensemble Approach for Deception Detection from Facial Cues

	SoA results		Our results	
Method	Accuracy	AUC	Accuracy	AUC
RF[26]	76.03%	-	83.61%	89.54%
L-SVM[57]	-	57.40%	91.87%	93.37%
L-SVM[56]	-	77.31%	92.01%	93.57%
LR[56]	-	64.25%	92.01%	93.57%
MLP[58]	93.08%	95.96%	91.87%	93.37%
RBF-SVM[33]	76.84%	-	90.75%	93.12%

Figure 41 - Competitive Benchmarking Proof 5

Toward End-to-End Deception Detection in Videos

Source	Method	Accuracy (%)
–	Random	50
Vocal	IS09	71.5
	IS13	70.05
	DEV-vocal	**74.16**
Visual	AUs	68.5
	DEV-visual	**75.00**
Hybrid	Vocal: IS09 Visual: AUs	73.5
	Vocal: IS13 Visual: AUs	72.1
	DEV	**84.16**

Figure 42 - Competitive Benchmarking Proof 6

Appendix F

List of Evaluators

Evaluator Type	Role	Name of Evaluator	Organization
Domain Experts	Deputy Inspector General of Police - Former Director of CID	Mr. B. R. S. R. Nagahamulla	CID
	Senior Superintendent of Police - Former Crime OIC	Mr. Ashoka Mahinda Weerakkody	Crime Unit - Kalutara
	Minor Complaint Officer In Charge	Mr. Rohan Wishwanath	Maharagama Police Station
	Sub Inspector of Police	Mr. Prasad Samaraweera	Bandaragama Police Station
Industrial Experts	Senior Consultant – Technology (Data Science)	Mr. Chathusha Wijenayake	Virtusa Pvt Limited
	Data Analyst	Mr. Joseph Hungerman	Aeturnm Inc.
	Senior Software Engineer	Ms. Yashmika Kuruppu	Mitra Innovation
	Site Reliability Engineer	Mr. Chinthaka Hasakelum	Pearson Lanka Pvt Limited
	Technical Project Manager	Ms. Pumudi Vidanagama	Aeturnum Inc.
	Senior QA Lead	Mrs. Piyumi Jayasinghe	Persistent Systems
	Senior Software Engineer	Mr. Sarith Gunathilake	Noon Technologies
Researchers	Senior Staff Algorithm Engineer	Dr. Samith Abeywickrama	Huawei
	Project Support Officer	Dr. Sanduni Gunawardena	University of NSW, Australia

Table 72 - List of Evaluators