

CHAPTER ONE

INTRODUCTION

1.1 BACKGROUND TO THE STUDY

The formal educational system in Nigeria has experienced several revolutions over the years and is named as such because it is characterized by a structural method of learning which is followed by an assessment at the end of a period of study. This assessment is done to test the cognitive ability of students over a particular course of study. A successful completion of all assessments during the course of study qualifies such student for certification.

In our world today, many studies have shown to us that the focus of the formal system of Education has shifted subtly from knowledge acquisition to certificate acquisition. Many earn the certificate without a proper knowledge of the subject involved, in the light of this, students now engage in so many activities such as, but not limited to, bribery or payment for certification, all forms of Examination malpractices, and most importantly, for the purpose of this study, impersonation in an attempt to get the certificate. Impersonation is a trending issue in the formal system of education where students who are meant to seat for an examination pay another person - '*machinery*', to write the examination for him or her. Study has shown that human invigilation may not achieve optimal result in the check of impersonation in an examination environment because an invigilator cannot be familiar with all the students in an examination hall.

Yaba College of Technology, like many other institutions in Nigeria, conducts paper and pencil test and uses human invigilators for her examinations. This process employs the use of human resources (personnel) in an examination to control the examinee, to monitor for exam misconduct and to discover any form of irregularities during exams. This process is not only time wasting, but also humanly demanding. In the sense that, it requires the invigilators to be active at all times and to also discover evidences that can be used to prosecute examinee caught in the act of misconduct. In many cases, when impersonation is discovered, the culprit outsmarts the invigilators and the security guards by absconding with every form of evidence, leaving the entire examination hall in a state of confusion. This state obviously would distract many students in the examination hall.

An automated form of the old system also found in many institutions today is the Computer Based Exam (CBT/CBE). This process allows students to write exam online using a computer system, but the level of monitoring and control of students during exam is not of a proper online exam standard.

A solution that had been provided is the Examination Surveillance System. The Examination Surveillance System is a system that is used to monitor students writing an exam using surveillance cameras. Amanze et al. (2016), defined surveillance as behavioral monitoring, activities, or other changing information, usually of people for the purpose of influencing, managing, directing, or protecting them. This can include

observation from a distance by means of electronic equipment (such as CCTV Cameras), or interception of electronically transmitted information. Surveillance cameras are video cameras used for the purpose of observing an area. They are often connected to a recording device or IP network, and may be watched by a security guard or law enforcement officer.

The Examination Surveillance System helped to monitor the examination environment but introduces a need to watch through the entire video clip in order to ascertain the integrity of the examination. During this process, there might be oversight on the part of whosoever is watching the video, the person can be stressed out or even may not be able to readily identify a case of impersonation.

As Artificial Intelligence and Expert system emerges, it had paved way for several uses of data and as applied to data mining today has helped in extracting hidden knowledge from data. A broad aspect of data mining is the Multimedia Data Mining which cuts across extracting knowledge from text (Text mining), video (Video data mining), audio (Audio data mining) and images (Image data mining). Video data mining can be used in analyzing videos for desired information using various data mining techniques like Clustering, Association, Classification etc. Video mining technique is used to check for irregularities in the shots, then the frames are grouped into categories. From the video technique(s) used, decision will be made based on the generated pattern. This makes video data more than just a sequence of frames but a repository for knowledge which can be harnessed.

In light of the above, this work therefore implements a video data mining system that helps discover impersonation and also monitors misconduct in an online examination video data. Video mining application to examination surveillance system is the use of video mining techniques to discover malpractice and misconduct in an examination environment. It employs various algorithm and techniques to find irregularities captured by a video surveillance system of an examinee in an examination.

1.2 STATEMENT OF THE PROBLEM

Impersonation among many other malpractices committed in the course of an examination is the major concern of this work. Students get more skillful in ensuring that they are not caught by invigilators who have a large number of other students to invigilate hence may not even pay attention to the fact that a student may be writing for another person. Invigilators in most cases cannot commit the image of every student to memory so extra efforts need to be made to tackle impersonation. In the case of impersonation, the problem encountered by the existing system includes time wasting in the process of truth establishment. The manual or existing system wastes time during the course of investigation to determine if a student is guilty or not.

Previous systems have used biometric features to uniquely identify students who are registered for the examination. This had been productive to the degree at which every single student has his finger print

collected before the examination. Research has proven that this method may fail during authentication as many factors like wet fingers or hardware failure may lead to this system not working as required. In ensuring the integrity of our examination, using only the manual method of invigilation and fingerprint authentication may prove abortive in the case of fatigue and or failure respectively.

Having seen the lapses of the aforementioned systems, an Examination Surveillance System has proven effective in the past, but this fails in the fact that human effort is needed eventually in ascertaining the integrity of the examination. The videos recorded are not in themselves intelligent enough to discover impersonation during the examination. Hence, a mechanism needs to be devised to automatically know from this video data whether or not impersonation took place in the examination. This mechanism is what this work provides using Video Data Mining Technique in uniquely identifying each student and also to identify an intruder in the examination hall.

1.3 AIM AND OBJECTIVES OF THE STUDY

The aim of this study is to detect impersonation in an online examination surveillance video. The specific objectives of this study are to:

- i. review existing study on online examination and video mining
- ii. design an online examination model based on the outcome of (i).
- iii. develop an online examination surveillance system using video data mining techniques
- iv. evaluate the performance of the new system in comparison with the existing online examination system

1.4 METHODOLOGY

To achieve the first objective, an extensive study is carried out on past literature relating to online examination and video mining techniques. Existing models are also reviewed and analysed.

An improved online examination surveillance model was designed, these achieved the second objective.

To achieve the third objective the online examination surveillance system was developed from the model. This system takes as its input Video data from an Examination Surveillance Camera, mines the video data and outputs frames containing unknown faces from the video data. In achieving this, a web-based user interface is designed using HTML, CSS and Javascript. Image processing is done using OpenCV (Short for Open Computer Vision - OpenCV is a python library used for computer vision). The video input is split into frames and each of the frames are converted to gray scale to reduce the amount of computation necessary. Face Recognition is done by training a Convolutional Neural Network to detect faces using a Classification Technique called K-Nearest Neighbour Classifier. This helps to separate faces trained from unknown faces.

In this study, it is important to know that an assumption is made that the surveillance camera is statically positioned, so the algorithm does not handle changes in the background as the camera moves. The output of the system are frames of name tagged students if student is valid and an *unknown* tag for impersonation prediction that decision makers can utilize in their decision making.

In achieving the last objective, the system is evaluated using test data gotten through random sampling of students whose face images are collected and trained then a video clip of them taking an examination is recorded for mining.

1.5 SCOPE OF THE STUDY

This work focuses on the design and implementation of a web-based surveillance system that discover impersonation in examination video data using video data mining techniques. The work uses the current EDP (Entrepreneurship Development Programme) Examinations taken in Yaba College of Technology as a case study since surveillance system is in place and cost overhead is reduced.

1.6 SIGNIFICANCE OF THE STUDY

The significance of this study is to ensure that examinations are independently monitored without the knowledge of the students, and also, to provide a more secured, faster and accurate detection of impersonation during an examination. Through this study it is believed that the credibility of exam will increased by making sure no form of impersonation is allowed during the exam, and it also minimizes the need for invigilators in the examination hall. Furthermore, it minimizes the time for proof finding against a suspected student.

1.7 DEFINITION OF TERMS

Impersonation: This is a criminal act in which a person pretends to be a different person.

Invigilator: Someone who watches examination candidates to prevent cheating.

Surveillance: Close observation of a person or group.

Online examinations: are examinations carried out on the internet or within an organization or institution using an intranet. Online exams are done via a network connection and they are sometimes referred to as e-examinations.

Misconduct: Activity that transgresses moral or civil law.

HTML: Hypertext Mark-up Language, a standardized system for tagging text files to achieve font, colour, graphic, and hyperlink effects on World Wide Web pages. HTML is the "hidden" code that helps us communicate with others on the World Wide Web (WWW). When writing HTML, you add "tags" to the text in order to create the structure.

Cascading style sheet(CSS): a style sheet language used for describing the presentation of a document written in a markup language like HTML.

Javascript: A high level, interpreted programming language, for web development.

Python: An interpreted high-level programming language for general purpose programming. Programming language for software development.

Video mining: Video mining is the process of extracting the data from the videos.

Database: a software program for storing, retrieving and manipulating a database.

Data mining: Data mining is the method of data analysis from the different collection of databases and summarizing it into some useful information.

CHAPTER TWO

LITERATURE REVIEW

2.1 INTRODUCTION

Formal Education is a form of structured education which takes place in an environment targeted at training students. Training takes place in classes and by qualified teachers of the courses Onyibe et al. (2015). Traditionally, in the African Educational System, the whole practice of teaching and learning was basically by observation. There was no issuance of certificates because education was not seen as an end in its self but a means to an end. The people were only interested in the skill acquired and its application. Ankaranga & Ongong, (2013). The western system of knowledge acquisition also known as formal education was measured on certificates as opposed to the African traditional system of Education even though certificates is not a full proof of knowledge retention. Nnam & Inah (2015). Certificates are issued after successfully completing the structured course outline given in a specific period of time. At the end of each of these periods, tests are usually conducted to ascertain the students' understanding of the course and to evaluate his intellectual competence. Emaikwu (2012)

This common belief that certificates provide the basis for evaluating one's qualification has led many students in Nigerian institutions into doing several illegal things to acquire the certificate. One of these things is Examination Malpractice, which we also call Examination Misconduct for the purpose of this study. Examination Misconduct according to Anzene, Uzoigwe, Onuka and Amoo (2014), was first reported in Nigeria in year 1914 where the students were said to have seen the questions before the scheduled date. Examination misconduct therefore has been a major concern in the educational system of the country. Various forms such as Collusion among candidates, and between them and examination officials, impersonation, 'giraffing', inscription on body parts, irregular activities inside and outside the examination halls, use of mobile devices, bribery, exchange of answer booklets and many other forms of misconducts usually occur before or during the course of an examination exercise. Examination misconducts therefore defeats the primary purpose of conducting an examination that is to ascertain the intellectual capability of the candidate before issuance of certificates.

2.2 ONLINE EXAMINATION

Technology, and its solutions, has been used to automate many of the manual processes we had around. Online Examination is described as conducting a test with an online computer connected to a network to measure the knowledge of the participants on a given topic. Online Examination just like the manual

method of paper and pen examination is used primarily to measure cognitive abilities, demonstrating learning progress within an instructional unit or chapter. The possibility of Online examination enhances running of courses remotely and taking international examinations online. Presently, large-scale examining bodies find the transition from traditional paper-based examination to fully electronic examination a long one. Issues such as hardware requirements to accommodate the large number of candidates and a more important issue of ensuring stringent level of security have been of practical consideration. Another feature of an online examination is Electronic marking where an examination can be marked electronically. This reduces overhead from marking by humans and also reduces the possibility of human errors. There are no restrictions to the types of tests that can be marked electronically, ranging from multiple choice to written. To make online examination worthwhile, academic dishonesty must be properly handled by the system else the purpose of the system is defeated.

2.2.1 ACADEMIC DISHONESTY

Academic dishonesty is an act that is rampant at all levels of educational institutions. Students in traditional classrooms cheat in various forms such as hidden paper notes, or reading from another candidate's paper during an examination. Online examination has been said to provide other possibilities for cheating such as hacking. Plagiarism is also noted as an act of Academic dishonesty which refers to the misrepresentation of someone else work; copying and pasting from the internet or retype from a source someone's idea either work for word or otherwise without properly citing the individual or body. Rovia et al. (2008). The focus of the work however will be on online examination misconducts.

2.2.2 INVIGILATION

Invigilation is an active means of preventing candidates from indulging in examination misconducts such as the ones listed earlier in this work. Originally, Human invigilators were employed to ensure compliance to examination codes. Research has shown that Human Invigilation could likely be nonobjective due to the fact that collusion may occur between candidates and invigilator. In addition to the vulnerability of human invigilators, most times they are not always efficient enough to effectively carry out invigilation as they ought to due to human weakness and inability to see everyone at the same time.

2.2.3 SURVEILLANCE

Surveillance refers to the monitoring of activities, behaviours, or other changing information, usually of people for the purpose of influencing, managing, directing, or protecting them. Amame et. al (2016). Video Surveillance involves observation from a distance using surveillance cameras (CCTV cameras) which are often connected to a recording device, IP network and/ or watched by a security guard. Amame et. al (2016).

Among many other types of surveillance, there is Data mining and Profiling surveillance which involves the search for hidden patterns in large amounts of data and processing or assembling information about a particular individual or group in order to generate a profile-that is, a picture of their pattern and behaviour.

2.3 ONLINE EXAMINATION SURVEILLANCE SYSTEM

Manual monitoring of students during an examination through invigilator could be successfully replaced by using video cameras. This solution however do not completely remove humans from the system as they would still have to go through the stress of having to watch through the entire video captured to detect suspicious activities. An online Examination Surveillance system makes use of various techniques in automatically detecting activities in a video and act as required. To achieve this, a major activity is to discover pattern from the videos recorded by the surveillance camera using video data mining techniques.

2.4 RELATED WORKS ON ONLINE EXAMINATION SURVEILLANCE SYSTEM

Nikhil, D., Komal, P., and Juiee R (2018), in their work ‘Online Surveillance for Exam’, proposed a face monitoring model which uses the Haar Cascade classifier algorithm in detecting faces before a web cam during examination. The system monitors that the face is not out of sight during the entire period of the examination, else, it reports a malpractice. Also, the System reports malpractice when the user attempts to open another window during the course of the examination.

Also Amanze B, Onpniwu C., Nwoke B.C (2016), in their work ‘Video Surveillance and Monitoring for examination Malpractice in tertiary institutions’, proposed a model which uses a Surveillance camera with thermal detectors. Their model was to detect body motions and temperatures which suggests fear or tension on the examinee. Videos of such scene is captured and stored for further evaluation. The result of the evaluation is whether or not the person is a culprit.

Finally, Sandhya, D.,Suvarna K., Chandini, S. (2017), aimed at developing a system that will analyze and identify students who indulge in malpractices/suspicious activities during the course of an academic offline examination. They performed impersonation test using Principal Component Analysis (PCA) based face recognition method (also called Eigen faces) which is done by cross checking a captured face against a database of images. Their system also detected for mouth motion by evaluating the height of the mouth relative to a certain threshold which is determined by the measurement of the size of the examinee’s mouth from the database of his images.

2.5 DATA MINING

Data Mining is an interdisciplinary subfield of computer science used for the computational process of finding patterns from large data sets which involves various areas of computer science like artificial

intelligence, machine learning, and database system. Data Mining requires the knowledge of statistics. The main aim of Data mining is to extract the useful patterns which are hidden in large data sets and transform the extracted information into a meaningful form (Jiawei & Micheline). Data mining is a major step in the knowledge discovery process (Manjunath et al., 2011). It is the analysis step of the "knowledge discovery in databases" process or KDD. The term is a misnomer because the goal is the extraction of patterns and knowledge from large amounts of data, not the extraction (mining) of data itself. It is frequently applied to any form of large-scale data or information processing (collection, extraction, warehousing, analysis, and statistics) as well as any application of computer decision support system, including artificial intelligence, machine learning, and business intelligence. The newly discovered patterns can further be used to make certain decisions for business development and forecast.

2.5.1 DATA MINING TYPES

1. **Predictive data mining:** It produces the model of the system described by the given data. It uses some variables or fields in the data set to predict unknown or future values of other variables of interest.
2. **Descriptive Data Mining:** It produces new, non-trivial information based on the available data set. It focuses on finding patterns describing the data that can be interpreted by humans.

2.5.2 DATA MINING ALGORITHMS AND TECHNIQUES

Different algorithms and techniques like Clustering, Regression, Classification, Artificial Intelligence, Neural Networks, Association Rules, Decision Trees, Genetic Algorithm, Nearest Neighbor method etc., are used for knowledge discovery from databases. Tamilselvi & Kalaiselvi (2013), in their paper where they did an overview on Data Mining Techniques and Applications, explained these techniques as follows:

A. CLASSIFICATION

Classification is the most commonly used data mining technique, which uses a set of pre-classified examples to develop a model that can classify the population of records at large. This is the task of generalizing known structure to apply to new data. For example, an e-mail program might attempt to classify an e-mail as "legitimate" or as "spam". Fraud detection and credit risk applications are particularly well suited to this type of analysis. This approach frequently employs decision tree or neural network-based classification algorithms. The data classification process involves learning and classification. Learning the training data are analyzed by the classification algorithm. In classification, test

data are used to estimate the accuracy of the classification rules. If the accuracy is acceptable the rules can be applied to the new data tuples. For a fraud detection application, this would include complete records of both fraudulent and valid activities determined on a record-by-record basis. The classifier-training algorithm uses these pre-classified examples to determine the set of parameters required for proper discrimination. The algorithm then encodes these parameters into a model called a classifier.

Types of classification models:

1. Classification by decision tree induction
2. Bayesian Classification
3. Neural Networks
4. Support Vector Machines (SVM)
5. Classification Based on Associations

B. CLUSTERING

A common descriptive task in which one seeks to identify a finite set of categories or cluster to describe the data. Clustering can be said as the identification of similar classes of objects. By using clustering techniques, we can further identify dense and sparse 16 regions in object space and can discover the overall distribution pattern and correlations among data attributes. This involves the task of discovering groups and structures in the data that are in some way or another "similar", without using known structures in the data.

Types of clustering methods

1. Partitioning Methods
2. Hierarchical Agglomerative (divisive) methods
3. Density-based methods
4. Grid-based methods
5. Model-based methods

C. PREDICTION

Regression technique can be adapted for prediction. Regression technique attempts to find a function which models the data with the least error that is, for estimating the relationships among data or data sets. It is used to model the relationship between one or more independent variables and dependent variables. In data mining, independent variables are attributes already known and response variables are what we want to predict. Unfortunately, many real-world problems are not simply a prediction. For instance, sales

volumes, stock prices, and product failure rates are all very difficult to predict because they may depend on complex interactions of multiple predictor variables. Therefore, more complex techniques (e.g., logistic regression, decision trees, or neural nets) may be necessary to forecast future values. The same model types can often be used for both regression and classification.

Types of Regression Methods:

1. Linear Regression
2. Multivariate Linear Regression
3. Nonlinear Regression
4. Multivariate Nonlinear Regression.

D. ASSOCIATION RULE (DEPENDENCY MODELLING)

Association and correlation are usually to find frequent item set findings among large data sets. It searches for relationships between variables. For example, a supermarket might gather data on customer purchasing habits. Using association rule learning, the supermarket can determine which products are frequently bought together and use this information for marketing purposes. This is sometimes referred to as market basket analysis. This type of finding helps businesses to make certain decisions, such as catalog design, and customer shopping behavior analysis. Association Rule algorithms need to be able to generate rules with confidence values less than one.

However, the number of possible Association Rules for a given data set is generally very large and a high proportion of the rules are usually of little (if any) value.

Types of Association Rule

1. Multilevel association rule
2. Multidimensional association rule
3. Quantitative association rule

E. NEURAL NETWORKS

This is a set of connected input/output units and each connection has a weight present with it. During the learning phase, the network learns by adjusting weights so as to be able to predict the correct class labels of the input tuples.

Neural networks have the remarkable ability to derive meaning from complicated or imprecise data and can be used to extract patterns and detect trends that are too complex to be noticed by either humans or other computer techniques. These are well suited for continuous-valued inputs and outputs.

For example, handwritten character reorganization, for training a computer to pronounce English text and many real-world business problems and have already been successfully applied in many industries. Neural networks are best at identifying patterns or trends in data and well suited for prediction or forecasting needs.

2.6 MULTIMEDIA DATA MINING

Today, an increase in multimedia creation and utilization, as well as advancement in their storage technology, have led to tremendous growth in very large and detailed multimedia databases; as a result of this growth, much more than a focus on the methods and tools for organizing, managing and querying such data, there is a shift of attention to methods and tools to discover hidden knowledge from such data. In developing these methods and tools, there is a challenge of lack of coincidence between information that one can extract from the multimedia data and the interpretation that can be given to it in a different context, this lack of coincidence is referred to as the Semantic Gap of Multimedia data (Zhang et al.,2004). This challenge has, in turn, led to the study of Multimedia Data Mining (Venkatadri, 2011).

Multimedia data mining can be defined as “the process of finding interesting patterns from media data such as audio, video, image and text that are not ordinarily accessible by basic queries and associated results” (Chitra, 2012). The term ‘multimedia’ already helps us to understand that it involves various media of data representation such as Text, Audio, Image, and Video. These various media form the various aspects of Multimedia Data Mining. The focus of this paper which we will later address will be streamlined to Video Data Mining as an aspect of MDM.

2.6.1 ASPECTS OF MULTIMEDIA DATA MINING

A. Text Mining: Text mining is the process of extracting meaningful information from text data for particular purposes. Due to the unstructured and amorphous structure of text data, it is difficult to deal with algorithmically as opposed to data stored in databases. None withstanding, since the text is the commonest medium for formal communication, the field of text mining usually deals with such text which is rich in factual information and opinions and extracts information from them automatically. This activity is compelling even if success is only partial (Witten, 2005)

B. Image Mining: In image mining, semantically meaningful information (knowledge) are automatically extracted from image data. The primary challenge in image mining is determining how low level, pixel

representation contained in a raw image or image sequence can be processed to identify high-level spatial objects and relationship (Ordenoz,1999).

C. Video Mining: The video is a hybrid of several kinds of multimedia data such as text, image, metadata, visual and audio. It is widely used in many major potential applications like security and surveillance, entertainment, medicine, education programs and sports. Video data mining has a primary objective of discovering and describing interesting patterns from a large amount of video data. This pattern discovery and description are considered as one of the core problem areas of the data-mining research community. Compared to the mining of the other types of data, video data mining is still in its early stage. (Vijayakumar & Nedunchezian 2012)

D. Audio Mining: Audio signals are transmitted in form of sound waves. In audio mining, the content of these audio signals is automatically analyzed and searched. An application area of audio mining is in speech recognition, where the analysis attempts to detect a speech within an audio (Manjunath & Balaji, 2014).

2.6.2 PROCESS OF MULTIMEDIA DATA MINING

The process of multimedia data mining is broken down into stages. These stages are applicable to the various aspects of MDM already discussed above. They are:

1. Domain understanding: This stage entails learning how the results of multimedia data mining will be used so as to gather all relevant prior knowledge before mining. For example, while mining sports video for a particular sport like football, it is important to have a good knowledge and understanding of the game to detect interesting moves by players. (Shrishrimal et al.,2008)

2. Data selection: The data selection stage requires the user to target a database or select a subset of fields or data records which will be used for the data mining. This stage requires a proper understanding of the domain in the identification of useful data. (Shrishrimal et al.,2008)

3. Data pre-processing, cleaning and transformation: During this stage, important features are discovered from raw data. The preprocessing stage includes data cleaning, normalization, transformation and feature selection. Cleaning removes the noise from data. Normalization is beneficial as there is often a large difference between maximum and minimum values of data. Constructing a new feature may be of higher semantic value to enable semantically more meaningful knowledge. Selecting a subset of features reduces the dimensionality and makes learning faster and more effective. (Shrishrimal et al.,2008)

4. Discovering patterns: The pattern discovery stage is the heart of the entire data mining process. In this stage, the hidden patterns, relationships, and trends in the data are uncovered. There are several approaches to the pattern discovery stage. These include association, classification, clustering, regression,

time-series analysis, and visualization. Each of these approaches can be implemented through one of several competing methodologies, such as statistical data analysis, machine learning, neural networks, fuzzy logic and pattern recognition. It is because of the use of methodologies from several disciplines that data mining is often viewed as a multidisciplinary field (Shrishrimal et al.,2008).

5. Interpretation: The interpretation stage is the stage when the quality and value of the discovery made are evaluated. The result here determines whether the previous stages should be revisited or not. A proper understanding of the domain helps to put a value on the discovered pattern (Shrishrimal et al.,2008).

6. Reporting: This is the final stage of the data mining process; it consists of reporting and putting to use the discovered knowledge to generate new actions or products and services or marketing strategies as the case may be. This stage is dependent on the application area of the discovered knowledge. (Shrishrimal et al.,2008).

Having understood the stages of MDM, the following section now focuses on Video Data Mining.

2.7 VIDEO DATA MINING

Video data mining is the process of finding correlations and patterns previously unknown from a large video database (JungHwan O., JeongKyu L. & Sae H., 2005). As discussed earlier, Video data mining has a primary objective of discovering and describing interesting patterns from a large amount of video data which is considered as one of the core problem areas of the data-mining research community

2.7.1 VIDEO DATA MINING APPROACHES

There has recently been proposed data-mining approaches employed to explore knowledge in a video database. These approaches have not been optimal as a result of the complexity and unstructured nature of video data. Consequently, many video mining approaches have been proposed (MatsuoY et al.,2002),(Oh&Bandi,2002),(Su et al. 2008).

These approaches are classified into five categories.

1. Video Content Structure Mining
2. Video Clustering and Classification
3. Video Association Mining
4. Video Motion Mining
5. Video Pattern Mining

A. VIDEO CONTENT STRUCTURE MINING

Due to the complexity of the task of getting an efficient access to video because of its unstructured nature, the main objective of the video structure mining is the identification of the content structure and patterns to carry out the fast random access of the video database. As video structure represents the syntactic level

composition of the video content, its basic structure is represented as a hierarchical structure constituted by the video program, scene, shot and key-frame. (Fu et al. 2005)

Video structure mining is defined as the process of discovering the fundamental logic structure from the preprocessed video program using the data-mining methods such as classification, clustering and association rule. In Video Structure mining, it is essential that the video content is analyzed semantically and multi-modality information are fused in order to bridge the gap between human semantic concepts and computer low-level features from both the video sequences and audio streams. Video structure mining gets not only the video content constructing patterns but also the semantic information among the constructive patterns (Dia, 2006).

Video structure mining is executed in the following steps

- (1) video shot detection,
- (2) scene detection,
- (3) scene clustering, and
- (4) event mining (Zhu et al. 2003).

Fu et al (2005), defined two kinds of structural knowledge, namely, video semantic and syntactic structure knowledge leading to the concepts of video semantic and syntactic structure mining. Syntactic structure mining is based on the video basic structure which adopts the methods of data mining according to the similar video units and video unit features. It acquires some syntactic rules in general, including dialogue, interview, news, talk show and so on. These video syntactic rules are structural knowledge triggering the process that mines constructional patterns in the video structure units and explores relations between video units and features. Semantic structure mining is a process where semantics and events in video basic structure units are discovered. The basic structure units explore the relations between video unit features and features such as color and texture pattern in the explosion scene, light and texture pattern in an indoor or outdoor scene, an audio pattern in highlight scene and so on. These relations are represented by association rules between video unit feature(s) and feature(s). The current researches on it focus on mining object semantic information and event detection. The video event represents the occurrences of certain semantic concepts. Chen et al. (2006,2007), presented a video event detection framework that is shot-based, following the three-level architecture and 28 proceeding the low-level descriptor extraction, mid-level descriptor extraction, and high-level analysis. Heuristic rules can be used to partially bridge the semantic gap between the low-level features and the high-level subjective concepts. The decision tree logic data classification model algorithm is then performed upon the combination of multimodal midlevel descriptors and the low-level feature descriptors for event detection. Zhao et al (2006), proposed the Hierarchical Markov Model Mediator mechanism to efficiently store, organize, and manage the low-level features, multimedia objects, and semantic events along with the high-level user perceptions such as user preferences in the multimedia database management system.

B. VIDEO CLUSTERING AND CLASSIFICATION

Video clustering and classification are employed in clustering and classifying video units into different categories. Therefore, clustering is a significant unsupervised learning technique for the discovery of certain knowledge from a data set. The problem of clustering video sequences to infer and extract activities from a single video stream is extremely important and so it is of significant potential in video indexing, surveillance, activity discovery and event recognition (Turaga et al,2007), (Weber et al. 2010). In the video surveillance systems, clustering analysis is used to find the patterns and groups of moving objects. These clusters help to eliminate redundancy and as a result, produces a more concise video content summary (Zhu et al, 2002). Clustering algorithms are categorized into partitioning methods, hierarchical methods, density-based methods, grid-based methods and model-based methods. Vailaya et al. (1996), proposed a method to cluster the video shots based on the key frames representing the shots. It is detecting by the shots from the video frame sequence that the keyframes are extracted. Next, a feature vector is computed so that the keyframes can be clustered based on the feature vector assigning the semantic interpretations to various clusters at the last stage. These semantic interpretations are employed in the retrieval system to index and browse the video database. At the clustering stage, it is desirable to cluster the shots into semantic categories such as presence/absence of buildings, texts, specific texture and so on, so that a higher-level abstract (semantic) label can be assigned to the shots (indoor shots, outdoor shots, city shots, beach shots, landscape shots). Video classification aims at grouping videos together with similar contents and to dis-join videos with non-similar contents and thus categorizing or assigning class labels to a pattern set under the supervision. It is the primary step for retrieval and the classification approaches are those techniques that split the video into predefined categories.

C. VIDEO ASSOCIATION MINING

Video association mining is the way toward finding an association in a given video. The video knowledge learning is investigated in two phases, the first being the video content processing in which the video clip is divided into certain investigation units separating their representative features and the second being the video association mining that extracts the knowledge from the feature descriptors. Mongy et al. (2005), introduced a system for video use mining to produce client profiles on a video internet searcher with regards to motion picture generation that analyses the client practices on an arrangement of video information to make reasonable apparatuses to help individuals in perusing and looking through a substantial measure of video information. In video association mining, the video handling and the existing data mining algorithms are flawlessly coordinated into mining video information.

Zhu et al. (2005), proposed a multilevel sequential association mining to investigate the affiliations between the sound and visual signals and grouped the relationship by allotting every one of them 30 with

a class name utilizing their appearances in the video to develop video indices. They coordinated the customary association measures (support and confidence) and the video temporal information to assess video associations.

Sivaselvan et al. (2007), introduced a video association mining comprising of two key stages. In the first place, the change stage changes over the first information video into a substitute value-based arrange, to be specific, a group succession. Second, the regular fleeting example mining stage that is worried about the age of the examples subject to the fleeting separation also, bolster limits.

Lin et al. (2007), built up a video semantic idea disclosure system that uses multimodal content investigation furthermore, affiliation lead mining procedure to find the semantic ideas from video information. The system utilized the apriori algorithm and association rule mining to discover the frequent item-sets in the element informational index and produced the classification rules to classify the video shots into various ideas (semantics).

Chen and Shyu (2007), proposed a various leveled temporal association mining approach that coordinates the association manages mining and the sequential patterns discovery to efficiently decide the temporal patterns for target events.

Goyani et al. (2011) proposed an A-priori algorithm to distinguish the semantic ideas from the cricket video. At first, a top-down event recognition and classification were performed utilizing the various leveled tree. At that point, the more elevated amount idea was recognized by applying the A-Priori algorithm.

Maheshkumar (2010), proposed a strategy that consequently extricates quiet occasions from the video and characterizes every occasion arrangement into a concept by sequential association mining. A various leveled system was utilized for soccer (football) video event sequence detection and classification. The association for the occasions of every excitement clip was processed utilizing an a priori mining algorithm utilizing a sequential association distance to classify the association of the excitement clip into semantic concepts.

D. VIDEO MOTION MINING

Motion is a key component that basically portrays the substance of the video, representing the temporal information of videos and more objective and consistent compared to other features such as color, texture et cetera. There have been a few ways to deal with extracting camera motion and motion activity in video sequences. While managing the issue of protest following, calculations are constantly proposed on the premise of known protest area in the edges thus the most difficult issue in the visual data recovery is the acknowledgment and location of the items in the moving recordings. The camera movement has a

fundamental part to play a portion of the key issues in video movement discoveries are, the camera set in a static area while the articles are moving (surveillance video, sports video); the camera is moving with moving objects (movie); multiple cameras are recording similar items. The camera movement itself contains an abundant learning identified with the activity of the entire match. The imperative sorts of camera movement are Pan (left and right), Zoom (in and out), Tilt (here and there), and Unknown (camera movements those are not Pan, Zoom, or Tilt are grouped to unknown)

Wu et al. (2002), proposed the extraction plan of the global movement and motion and object trajectory in a video shot for content-based video retrieval. For example, while perusing the video acquired by surveillance system or watching sports programs, the client dependably wants to discover the object moving some extraordinary way. Zang and Klette (2003), proposed an approach for extraction of (another) moving item from the background and tracking of a moving item. Mining patterns from the developments of moving objects are called motion mining. Initially, the features are extracted (physical, visual and aural, motion characteristics) utilizing objects location and tracking algorithms and afterward the meanings of the patterns, trends of moving object activities and patterns of events are mined by computing association relations and spatial-temporal relations among features.

E. VIDEO PATTERN MINING

Video Pattern mining detects special predefined patterns usually characterized as video interesting events. Such events could be dialogue or presentation events in a medical video for example. The existing work can be partitioned into two classifications which include mining similar motion patterns and mining similar objects (Anjulan & Canagarajah, 2007).

Sivic et al. (2004) depicted a technique for acquiring the key objects, characters, and scenes in a video by estimating the recurrence of the spatial configurations of the perspective invariant highlights features. This has three phases: The initial stage removes the areas happening in excess of a minimum number of keyframes considered for clustering, whereas the second stage matches the significant neighborhoods through a greedy progressive clustering algorithm, and in the final stage, the subsequent clusters are merged in light of spatial and temporal overlap. Burl et al. (2004), introduced an Algorithm to extricate data from raw, surveillance-style video of an outdoor scene containing a blend of individuals, bikes, also, mechanized vehicles. A feature extraction algorithm in view of the background estimation and subtraction preceding spatial clustering and multi-object tracking was utilized to process sequences of video frames into a track set, which encodes the positions, speeds, and the appearances of the various objects as the function of time are mined to reply the client created queries.

Lai et al. (2006), proposed a motion framework that provides a means of measuring the similarities among different animal movements in high precision. A clustering method can isolate the repeating movements

from the inconsistent irregular ones. Fleischman et al. (2006), exhibited an approach in which the temporal information is caught by representing events using a lexicon of hierarchical patterns of human movement that are mined from large corpora of un-annotated video data. These patterns are utilized as features for a discriminative model of occasion grouping that exploits tree kernels in a Support Vector Machine. The second classification frameworks aim at grouping frequently appearing objects in videos. Accordingly, it is valuable to have commonly occurring objects/characters/scenes for different applications Sivic & Zisserman (2004). There are various applications: First, they enable visual search in video databases. Second, they can be utilized in forming video synopses—the fundamental components of a synopsis frequently include the commonly occurring objects and these are then shown as a storyboard. The third application region is in recognizing item situations in a film where the frequently occurring logos or labels are conspicuous. Mining repeated short clips from video collections and streams are fundamental for video syntactic division, TV broadcast monitoring, business skipping, content synopsis and personalization, and in addition video redundancy identification and numerous different applications. Xie and Chang (2006) explored the pattern mining procedures in video streams. They connected distinctive pattern mining models (deterministic and statistic; static and temporal) also, formulated pattern blend procedures for creating a rich arrangement of pattern speculation. A portion of the factual bunching technique, for example, K-means, HMM, HHMM and Deterministic algorithms were considered for video clustering. Yang et al. (2007), proposed a technique to repeat the clip mining and the Knowledge recovering from the video information. The mining structure binds together to distinguish both the obscure video repeats and the known video clips of the subjective length by a similar element extraction and matching technique. The structure analysis technique is successful in finding and displaying the syntactic structure of the news videos and their fundamental goal is to identify the obscure video rehashes from the video stream without prior knowledge. Su et al. (2008), exhibited a method to 34 accomplish a powerful content-based video recovery by mining the temporal patterns in the video contents. It was the development of a unique index on video temporal patterns for an effective recovery (Fast-Pattern-Index tree) and a one of a unique search technique for effective recovery (Pattern-based Search).

2.8 RELATED WORKS

Adetoba, Awodele, & Kuyoro (2017), proposed a framework for observing the e-examination surroundings which involves the extraction of pictures/ or images and audio or voice data. The proposed framework had four major areas: knowledge pre-processing, mining, association and post processing. The pre-processing phases carries out the extraction and transformation of multimedia system knowledge options, the mining part will the classification and clustering of those options, the association for pattern matching whereas the post process carries out the data interpretation and reportage. The approach conferred during this study can afford economical and correct observation of e-examination surroundings which can facilitate, give adequate security and scale back unethical behaviour in e-examination surroundings. This work however only provides a theoretical framework for the system, but is not implemented hence the accuracy of the proposed system is not known.

Enrica et al. (2012) in their paper presents a solution for the digitally enhanced assessment by designing two different learning dashboards in order to represent the most interesting learning analytics aiming at providing for both the teachers and the learners with easy understandable view of learning data in virtual learning environments. The analytics provided on the dashboard reveals student's disparity of what was to be learned and what has been learned at a particular point in time. E-assessment was not considered in the work and multimedia data was neglected.

Chen et al,(2002) developed a framework for mining multimedia data information from traffic video sequences. The framework was to discover hidden knowledge such as vehicle identification, traffic flow, queue detection, incident detection, and the spatio-temporal relations of the vehicles at intersections which provide an economic approach for daily traffic monitoring operations. The proposed framework analyzes the traffic video sequences using background subtraction, image/video segmentation, vehicle tracking, and modeling with the multimedia augmented transition network (MATN) model and multimedia input strings, in the area of traffic monitoring over traffic intersections. The spatio-temporal relationships of the vehicle objects in each frame are discovered and accurately captured and modelled. The knowledge from the proposed multimedia data mining framework in terms of spatio-temporal tracking generates a capability for automation. This model applies better to object tracking rather than face recognition even though there is only a slight difference between both.

Jung et al, (2013) also proposed a general framework for real time video data mining to be applied to the raw videos; investigation was carried out as to whether the existing techniques would be applicable to this type of videos. They introduce new techniques which are essential to process them in real time. They group video data into frame relating to the video structure. Clustering method is applied in order to discover unknown and interesting patterns by extraction motion, object, colours etc from each frame(s)

with which motion is the primary focus. This framework looks promising as the grouping of video data into frames would be adapted in this paper work, hence neglecting the other parts.

Amigud, (2017) proposed an integrated approach to enhancing academic integrity of e-assessments based on behavioural biometrics and aided by machine-learning techniques. It provides continuous identity and authorship assurance throughout the learning activities within the existing learning space using Naive Bayes classifier and word n-gram based feature set in presenting a preliminary result. This is a modelling-based approach and not observation. It can give the analytics of the degree of learner collaboration with peers and interaction with the course content, concurrently verifying learner identity and validating authorship of the academic artefacts. This paper work is limited because the framework's performance wasn't recognized in the write up.

The works already reviewed reflects that a lot of work had already being done along the area of Data mining, especially video data mining. These works however have limited working implementations. This work focuses on developing a full system using computer vision for face recognition in an examination video data.

2.9 SUMMARY OF RELATED WORKS

S/N	Author	Methodology	Gap
1	Adetoba, Awodele, & Kuyoro (2017)	Monitoring of e-examination environment using video data mining techniques	No implementation hence validity of model is unknown. Specific method is not specified. Specific video mining field of application (eg Machine Learning, Data Science etc) was not discussed in the paper.
2	Enrica et al. (2012)	Proposed model for Students' analysis of course work for	This work handles student's evaluation of an e-

		revealing the disparity of student's towards a course	assessment rather than using mining techniques. Video mining is ignored to also enhance integrity of examination.
3	Chen et al. (2002)	Multimedia augmented transition network (MATN) for object tracking in an analysed traffic video	The work focuses on object tracking and not face recognition.
4	Amigud, (2017)	An integrated approach to enhancing academic integrity of e-assessments based on behavioural biometrics, aided by machine-learning techniques.	This paper work is limited because the framework's performance wasn't recognized in the write up

CHAPTER THREE

METHODOLOGY

3.1 INTRODUCTION

The importance of conducting a fair examination cannot be overemphasized in our educational system. What is prevalent today is video surveillance, where CCTV cameras are mounted at strategic places to record the activities. The focus of this work is to develop a system which would help in mining videos collected from online examinations for impersonation.

To achieve this aim, we have carried out a proper review of various techniques and algorithms used for video mining.

A study was done on Online Examination in order to properly understand the problem domain. In this study, the following were observed:

- Online examinations are prone to impersonation as a result of little or no supervision and even in the event of supervision, those involved in the supervision may not have the integrity required to carry out a trust-worthy examination.
- Students' activities in the examination hall tends to outsmart the supervisor's awareness due to the large number of students to be supervised.

In view of the above problems discovered from our review work, the proposed system applies video mining techniques in recognizing faces to ensure no unknown face is seen in the examination hall.

A mounted top view surveillance camera records the top view of entire hall during the course of the test while another front view camera is used to identify impersonation. The video collected from these cameras are then processed into frames and each frame are converted to grey scale for further analysis. The output of this stage is a collection of frames with students identified by name tags and unknown student is tagged unknown student. This information can be used to draw conclusions about the presence of impersonation in the examination hall.

3.2 DATA COLLECTION AND METHOD

The data used in this project was a video clip from an examination scene collected from Yaba College of Technology Central lab. Yaba College of Technology laboratory was chosen as a result of proximity and ease of access as we are students of the school and access was easily granted for us to get data from the laboratory. This data was collected using convenience non-probabilistic sampling of students as a result of their unavailability during the time of the implementation of this work.

3.2.1 METHOD OF COLLECTING DATA

Pictures of candidates are captured to train the system for face recognition.

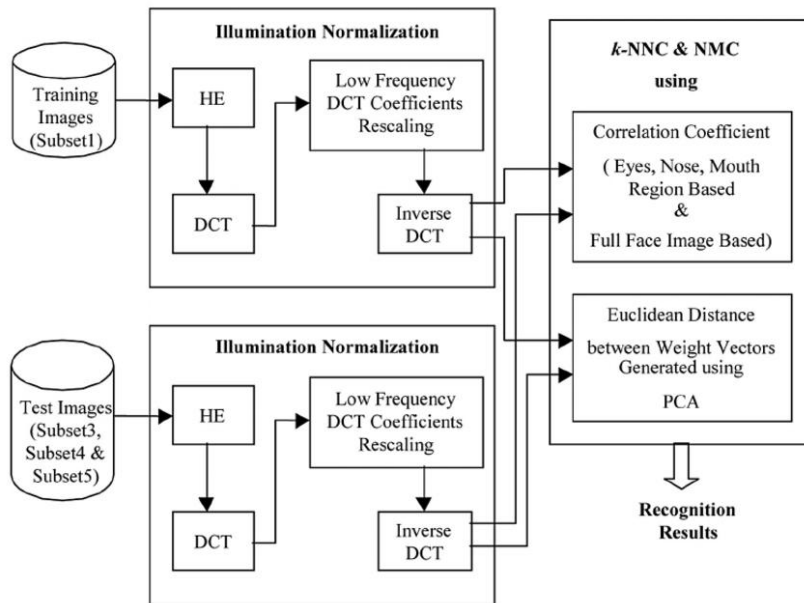
Video of the online examination is collected using the mounted CCTV camera. This video contained pre-registered students who wrote the examination and unregistered students to test the system's ability to detect an unknown candidate.

From the data collected, we categorize them as follows:

1. Train Set: The faces of each candidates constitute the train set for the face recognition module
2. Test Set: The videos containing normal and abnormal activities constitutes the test set. This is to test out system for efficiency.

3.3 SYSTEM ARCHITECTURE

Impersonation detection in a video data from an examination scenario is our input data which is processed and mined for unknown faces. The training of each faces in the video stream is first done before the examination using images of each candidates. The process of training is explained in subsequent sections of this write up. Below is a pictorial representation of the architecture of the system.



Demonstrating the architecture of the face recognition system. Source: Virendra P. V, (2018)

3.4 METHOD FOR DETECTION OF PRESENCE OF AN INTRUDER IN THE EXAMINATION HALL (IMPERSONATION)

Face recognition for impersonation detection

In this work, impersonation is detected by using face recognition. Face detection involves using computer vision to correctly recognize a face from a photo or a video stream. In the case of this work, we used image from an examination environment under surveillance by a mounted camera.

Face recognition is really a series of several related problems, I will enumerate these steps as follows;

First, look at a picture and find all the faces in it

Second, focus on each face and be able to understand that even if a face is turned in a weird direction or in bad lighting, it is still the same person.

Third, be able to pick out unique features of the face that you can use to tell it apart from other people—like how big the eyes are, how long the face is, etc.

Finally, compare the unique features of that face to all the people you already know to determine the person's name. (Adam Geitgey, 2016)

To achieve the above programmatically, we will build a pipeline, where each of the above steps of face recognition are solved separately and the result of the current step becomes the input for the next step till we have achieved our purpose. In short, we will be chaining together several different machine learning algorithms.

3.5 THE FACE RECOGNITION PIPELINE

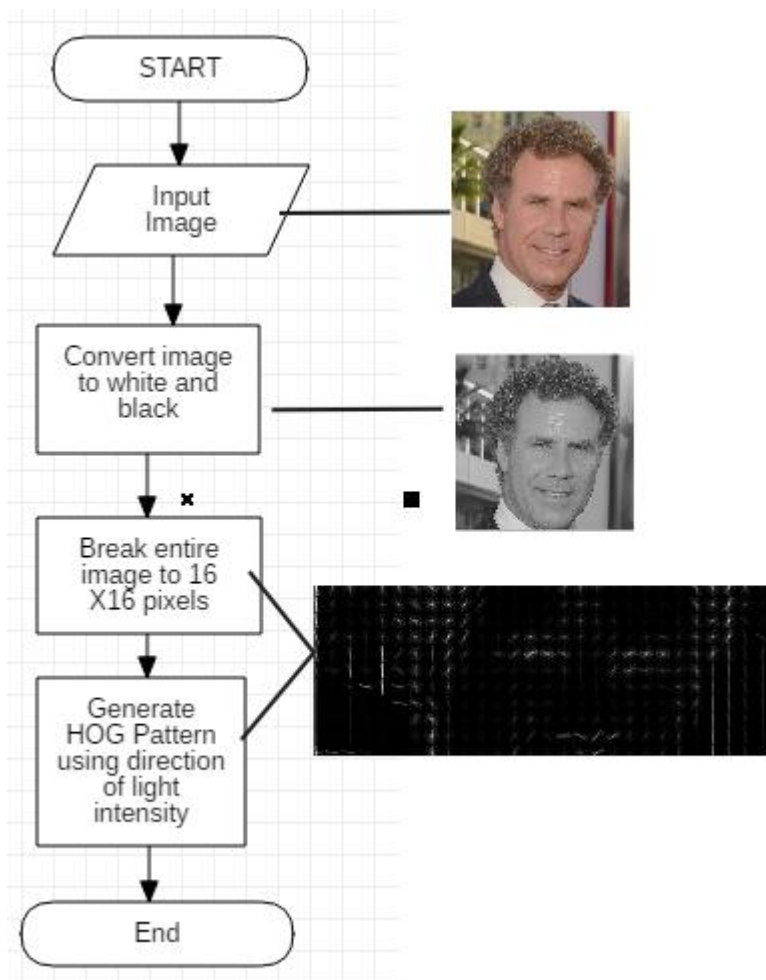
The process of detecting, encoding and recognizing faces summarizes the method implemented in this work. This process is referred to as the face recognition pipeline. The input of succeeding stages are outputs from preceding stages. A detailed explanation of the pipeline architecture is given as follows:

3.5.1 STEP ONE- FINDING ALL THE FACES

To find faces we implement a method called Histogram of Gradient (HOG). The HOG algorithm is given below:

- Image is converted to black and white since we do not need color data to find faces.
- Every single image is considered one at a time such that for every single pixel, the pixels directly surrounding it are examined. The goal here is to figure out how dark the current pixel is compared to the those surrounding it, then an arrow is drawn showing in which direction the image is getting darker. If this process is repeated for every single pixel in the image, what the output is that every pixel is replaced by an arrow. These arrows are called gradients and they show the flow from light to dark across the entire image. Saving gradient for each pixel however is way too much detail hence the entire is broken up into small squares of 16 X 16 pixels. In each of these squares, a count of how many gradients point in each major direction (Up, Up-right, Up-left, down, down-right, down-left) is taken, then that square in the image with the arrow is replaced by the direction that were the strongest. At the end the original image is turned into a very simple representation that entails the basic structure of a face.

-In finding faces, all to be done is to find the part of our image that looks most similar to a known HOG pattern that was extracted from a bunch of other training faces. The flow diagram below describes how this works:

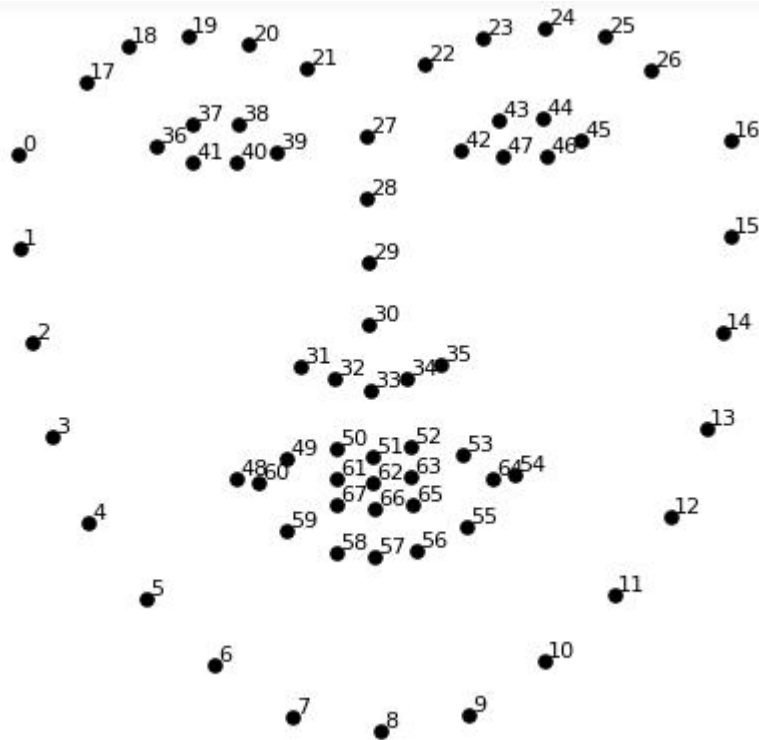


Process design for Step one of the face recognition pipeline

3.5.2 STEP TWO: POSING AND PROJECTING FACES

Even after finding faces in the image, situations where the faces turn to different directions need to be handled. This posture change appears different to the computer from a face. To account for this, each picture is warped so that the eyes and lips are always in the same place in the image. This will make it a lot easier to compare faces in the next steps. In achieving this, an algorithm known as Face Landmark Estimation is applied. The algorithm is stated below

- Identify 68 specific points known as Landmarks which exist on every face: the top of the chin, the outside edge of each eye, the inner edge of each eyebrow, etc.
- Train a machine learning algorithm to be able to find these 68 specific points on any face



Face 68 point Landmark. Source: Brandon A, 2016.

- Since the position of the eyes and mouth is now known, the image is simply rotated, scaled and sheared so that the eyes and mouth are centered as best as possible by preserving parallel lines during transformation. This is called Affline Transformation. After doing this, no matter how the faces in the image turned r the eyes and mouth are centered in roughly the same position in the image.

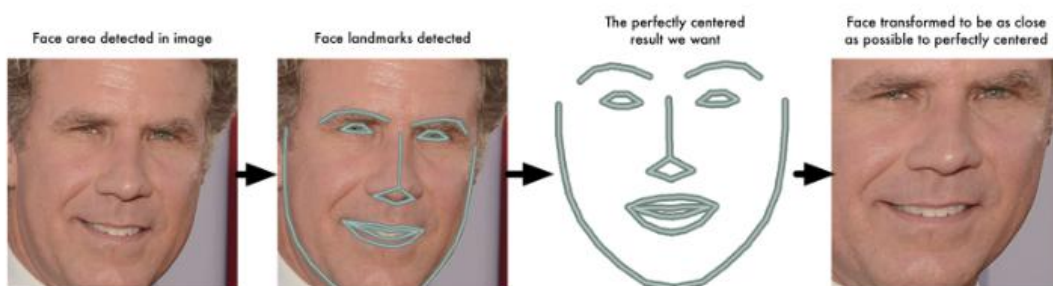


Image demonstrating posing face detection. Source Andrian R, 2018.

3.5.3 STEP THREE - ENCODING FACES.

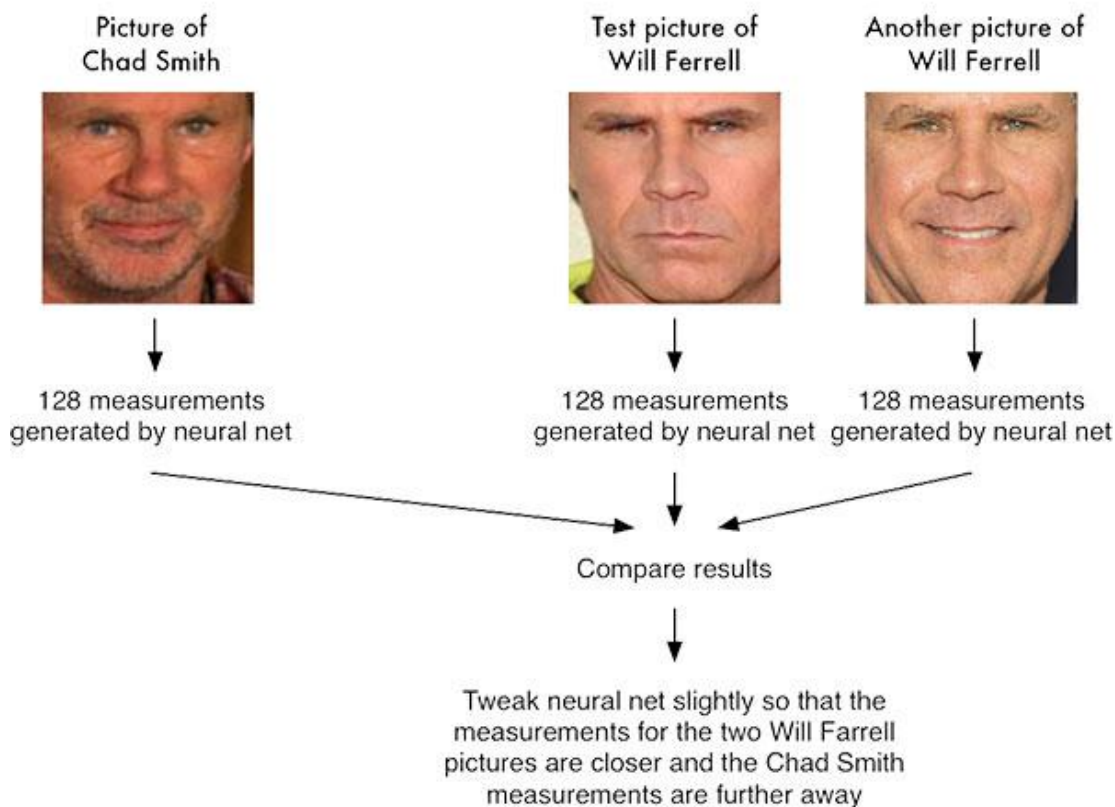
Faces are encoded by training a Deep Convolutional Neural Network to generate 128 measurements for each faces. Measurement in this case refer to dimensions of some face features like size of ear, mouth etc.

The feature to be measured is however determined by the computer itself. Training this network is done using Triplets.

3.5.3.1 THE TRIPLET TRAINING STEP

In the triplet training step, three images are provided to the network; two of the images are for the same person while the last one is any random face from the data set different from the first two images. This algorithm uses k-nearest neighbor classifier to classify two images as identical. This is done by embedding the face images in some vector space where similar images are closer to each other while that of different persons are far apart. Then in that vector space, it is easy to recognize similar faces by k-nearest neighbour classifier. As an example, looking at the image below,

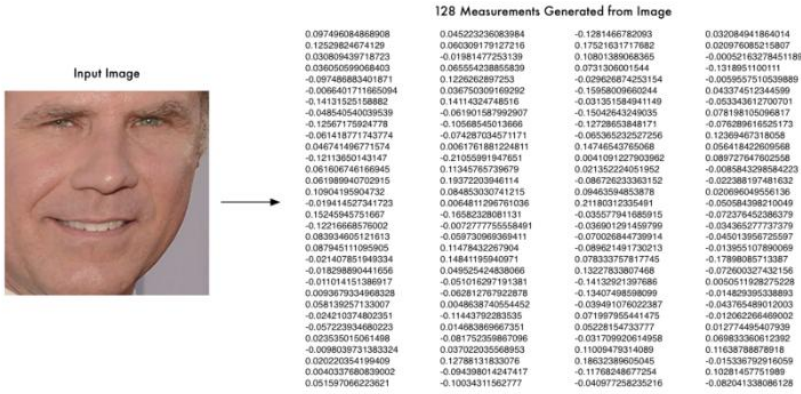
A single 'triplet' training step:



Images of Will Farrell and Chad Smith to be trained for recognition using K-NN .Source , Adam G., 2016

The Neural Network generates a 128-d feature

vector for each of the three images. That of the first two are tweaked such that they real values are close to one another and that of the different face is far apart. This is achieved using Distance Metric.

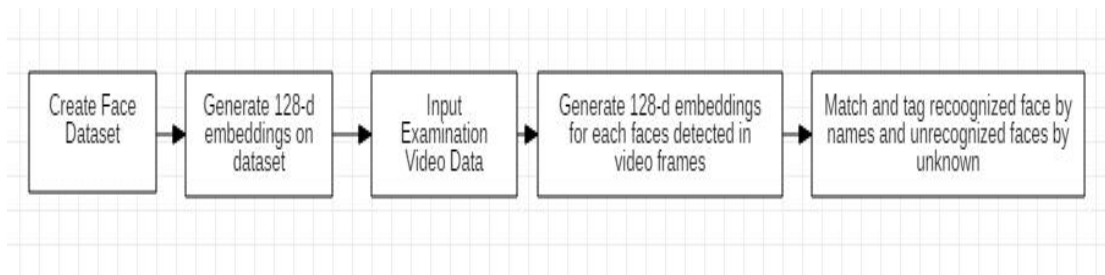


128-d Embedding generated for the input image on the left.

3.5.4 STEP FOUR: FINDING THE PERSON’S NAME FROM THE ENCODING

The last step of the face recognition pipeline is to find from the database of known faces who has the closest measurements to the test image. This is done using K-nearest neighbour classification algorithm.

3.6 PROCESS DESIGN FOR FACE RECOGNITION SYSTEM



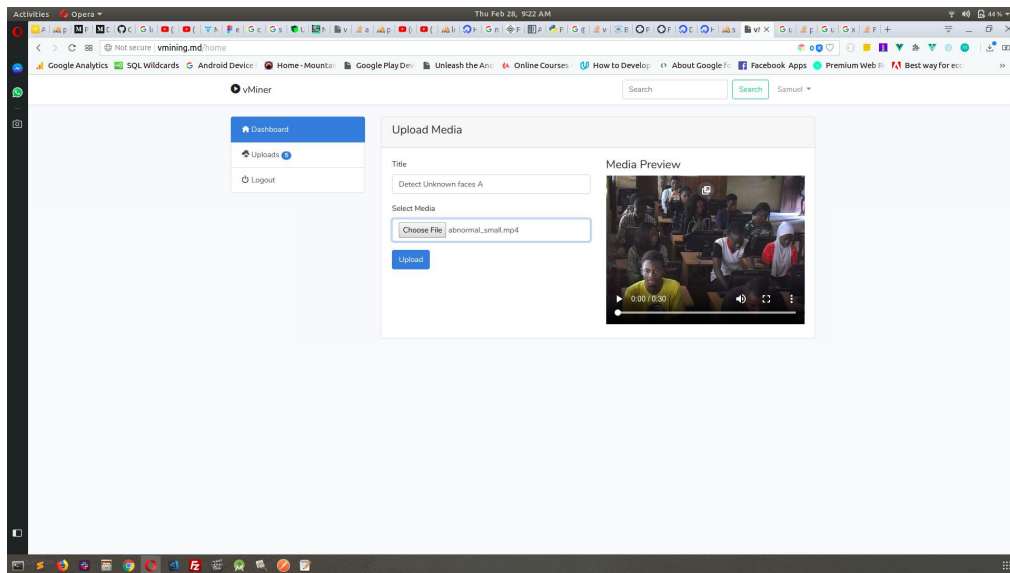
Process Design Flow Chart for Face Recognition System

3.7 SYSTEM DESIGN

Considering the methodology already discussed above to achieve face recognition. The system was implemented building a dedicated pipeline for detecting impersonation in an E-examination surveillance video, based on the above methodology. This system will be hitherto called Impersonation Detection System for the purpose of this writing. Video sample was collected and images of participating candidates are collected in order to use to train the Neural Network to recognize faces. The step by step procedure to this is provided below:

1. Video Upload

Videos from the surveillance camera are uploaded to be processed on using the dashboard below.

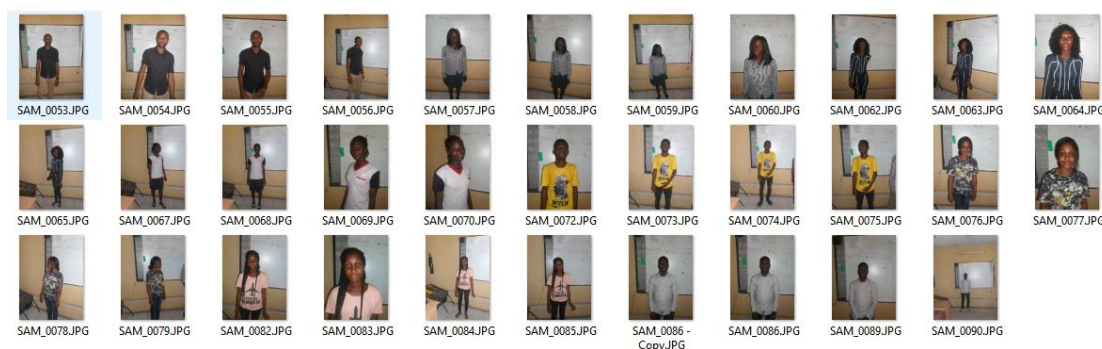


Face Detection

To perform face recognition on the Impersonation Detection System with python and OpenCv, two face recognition libraries namely, dlib and Face recognition were installed. Dlib implements the “deep metric learning” which is used to construct the face embeddings used for the actual recognition process while the face_recognition library, created by Adam Geitgey, wraps around dlib’s facial recognition functionality, making it easier to work with.

The Face Recognition data set

The data set used contains 4 pictures of eight candidates with name tags as appropriate. These pictures were trained for recognition using the Histogram of Gradient Methodology described in the previous section. The code for the encoding is saved in a file named encode_face.py in our project folder, the code snippet is available in the appendix session of this work.



Face

Recognition Data set

128-d embeddings for the above data set are generated by saved in a file called encodings.prickle. This content of this file is matched against features of the images in the Examination video to recognized the pre-trained images. Unknown faces are tagged with ‘unknown face’.

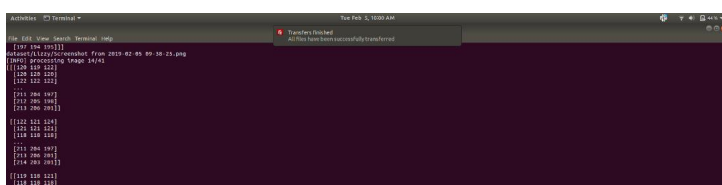


Fig 3.9 Image of 128-d feature vector generated on our data set.

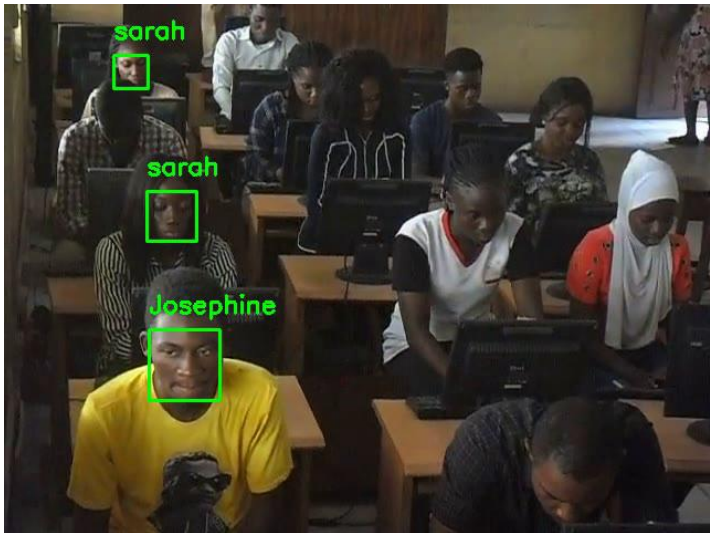
3.8 Face Recognition from Video Data

The video data from our examination scene is run through the face recognition library using OpenCV. The file for that is saved as recognize_faces_video.py in our folder structure. This python script includes the face recognition library and the pickle file generated after encoding. 128-d encoding is generated for the input frames from the video file and these encoding are matched for similar measurements from the already



Encoding the images on in a Neural Network.

This takes a lot of time depending on the length of the video input. The output at this stage is a video with name tags for each faces in the video based on the encoding from the previous pipeline.

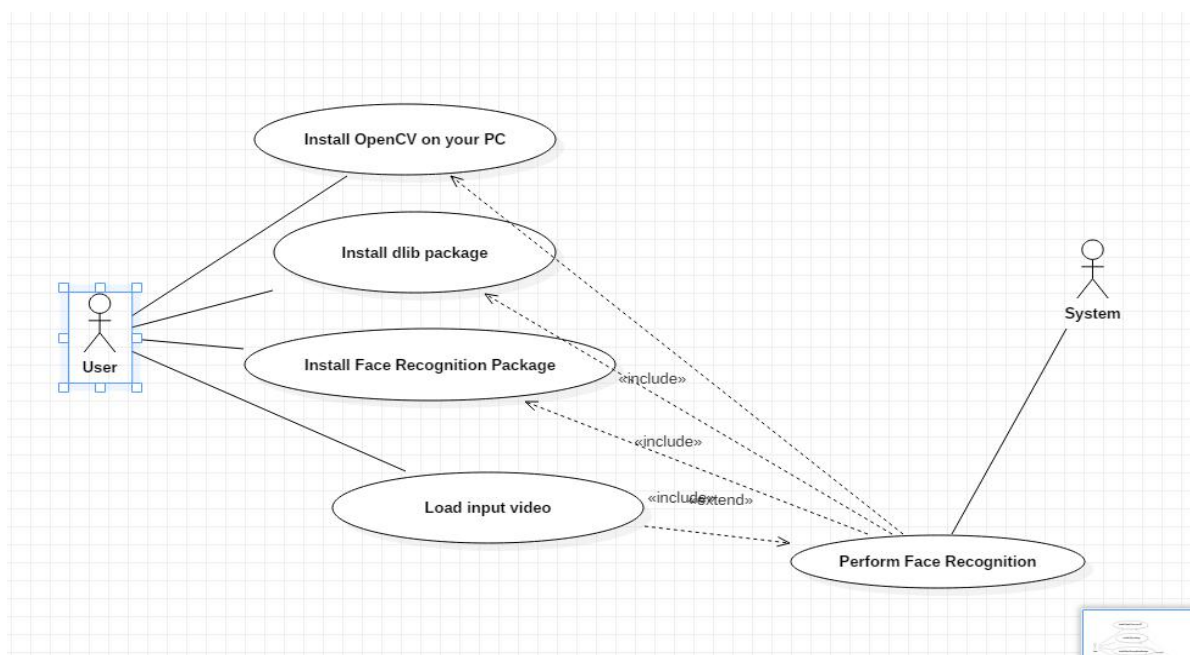


Example frames from Examination Video with recognized faces.



Example frame from Examination Video with unrecognized faces.

USE CASE DIAGRAM



Use case diagram for low level implementation of system

3.8 SYSTEM PROGRAMMING

3.8.1 The Choice of Programming Language

Python, as already stated is the programming language of our choice. Python programming language presently occupies the position of the most popular and widely used open source language for Artificial Intelligence and Machine learning with a non-complex easy-to-use syntax. More so, Python programming language (Simply put Python) is a general purpose language that can be used for web projects (as we have

in this case), desktop graphical user interface projects and scientific and mathematical computing (NumPy, Orange, SimPy). The version of python used for this work is version 3.6.

3.8.2 Software libraries Used

As already stated in the chapter three of this work, the libraries that were used in the implementation includes but not limited to OpenCV (Open Computer Vision). In this section the various uses of these libraries will be explored as it applies to our project.

A. Open Computer Vision (OpenCV)

OpenCV is a library of programming functions primarily for computer vision. It is a programming interface for Java, Python and C++. Its interface for python programming language was implemented in this work. OpenCV runs on Windows, Linux, macOS, FreeBSD, NetBSD and OpenBSD desktop operating systems and on Android, iOS, Maemo and BlackBerry mobile operating systems.

B. dlib

Dlib is an additional library that is installed in order to perform face recognition with python and OpenCV.

Dlib is a C++ toolkit which contains various algorithms for machine learning. This will be used to implement the deep metric learning algorithm as mentioned in Chapter three.

C. Face Recognition

Face recognition is a simple face recognition library built using dlib's state-of-art face recognition. It is available on Github to be included in this project. Face Recognition requires python version 3.3+ or 2.7 to run. It also runs most optimally on Mac or Apple operating systems. It however may also be run on a windows OS but this is not recommended.

3.8.3 Hardware Specification

To train the model that was used in the system, a Graphics Processing Unit (GPU) is required. The memory required was at least 4GB while the processing speed was at least 2.0GHz. A cloud system provided by Digital Ocean was used to perform the training and testing of the HOG models used because our system seemed to be overworked while trying to run the models on it which resulted in a system crash each time an attempt is made. Digital ocean provided with 4GB memory, 2vCPUs, 50GB SSD disk storage which are a good configuration for the model together with a availability of a CUDA compatible GPU for running our CNN model. The output video was available to be downloaded through the graphical user interface provided by the cloud service.

3.8.4 Software Specification

The system was implemented on Linux Ubuntu operating system. Therefore, it can only run on a Linux operating system. Also, Python 3.6 and all the software libraries discussed earlier (i.e. OpenCV, Dlib and Face Recognition) need to be installed on the operating system used.

3.9 DATABASE IMPLEMENTATION

The Database for this work was implemented as a pickle file containing encodings for all the faces in the data set. These encodings are later queried while performing face recognition on video data. The encodings are in form of a real valued feature vector representing measurements of certain computer-determined features of the data set. The diagram below shows an example of a pickle file.

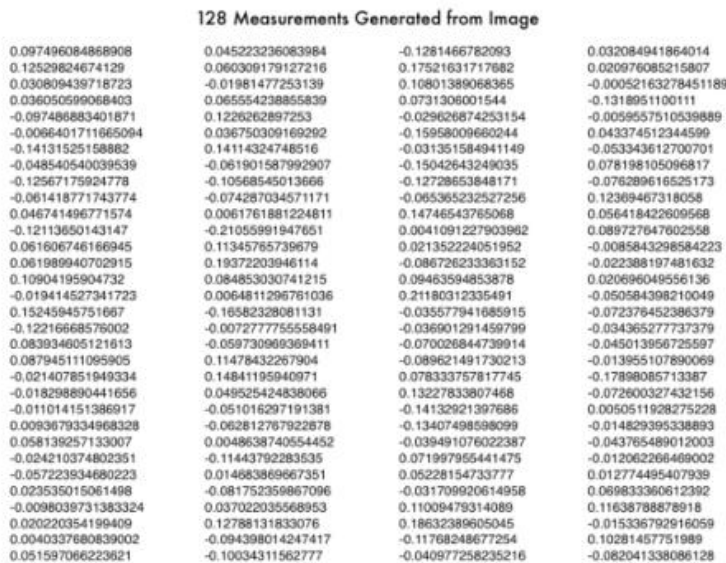


Fig 3. Showing an example of 128-d embedding from an image.

CHAPTER FOUR

RESULT AND DISCUSSION

4.1 INTRODUCTION

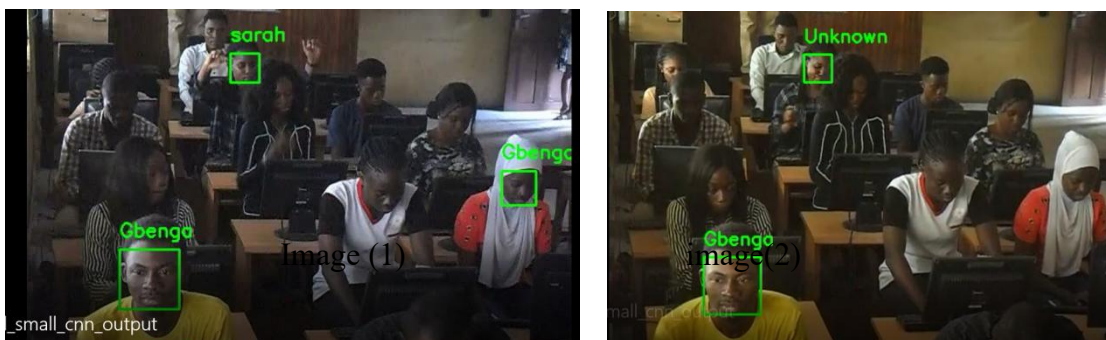
The face recognition system developed works for both video input from file and also from webcam hence the system can adapt to both conditions. Hardware requirement for this system is 4GB RAM or more, 2GHz or faster processing speed, GPU and a minimum of 500GB hard drive. Software requirements involves ability to run OpenCV on your PC, all the dependencies mentioned in chapter 3 must also be installed on the system. The operating system on which the system should be deployed is most preferably Linux. Any platform that can communicate with a python API can be used to implement the front end for the system.

4.2 RESULTS

In this section, findings made from the test set are discussed and are used to evaluate as well as make recommendations for the Impersonation Detection System.

4.2.1 RESULTS FROM STUDENTS DATA SET

The students data set as discussed in the chapter 3 of this work contained 4 images of 8 students. We used all 32 images to train the system from which 128-d encodings were generated using HOG algorithm mentioned in Chapter 3. The following are images from the output video.



4.3 DISCUSSION

From the two images above, we can deduce that the face most accurately detected was that closest to the camera as those farther away are either not recognized or tagged unknown. The reason for this is the little amount of images used for the train set. Increasing the number of images will obviously improve the performance of the system.

From the result above we can see that the confidence of the face recognition system increases as the number of images in our data set increases. This condition is typical of face recognition systems. Hence enough images must be used in the train set.

CHAPTER FIVE

SUMMARY OF FINDING, CONCLUSION AND RECOMMENDATION

5.1 INTRODUCTION

During the course of this project, the model was introduced, analyzed, designed, implemented and evaluated in order to achieve the aforementioned objectives. We used the face recognition pipeline to train, encode and recognize faces in video data.

5.2 SUMMARY OF FINDINGS

Computer Vision like all other mundane tasks carried out by the computer remains computationally complex compared to how easy it is to do the same by humans. In finding out whether there is a case of impersonation in an examination, humans may actually not be as prompt as computers will recognize an unknown face, hence the need for this system. The major factors that however affect the accuracy of the predictions made by the computer are Number of images used to train the Network and the computational capacity of the system used.

The low number of images in data set for student video is as a result of the difficulty of getting several images of the same student for training before an examination. This may be a problem which users of this system may run into as many students do not have a consistent name across their web presence hence, the Bing search API may not be able to help get a good number of the student across the web.

5.3 CONCLUSION

In order to get better result for the objective stated for this project the number of images in the data set must be large enough to guarantee accuracy of the predictions. This large number will require more computation hence we recommend that the system on which this will be deployed meets the following requirements 8GB RAM, 2GHz or more processing speed and dedicated GPU for high performance.

5.4 RECOMMENDATION

As we have seen, the Face Recognition System used to perform impersonation detection in this work require that the system is trained with faces of the candidates first. This requirement can be easily met in a scenario where a school wants to implement this system for her examinations. A database of the same students' images can be created from series of events in the school before the exams hence the train set is readily available before the examinations.

In the case of examinations organized by exam boards (eg. University Tertiary Matriculation Examination (UTME) in Nigeria), it become a more difficult task to get a collection of images of all the candidates within a short period.

We therefore recommend that this system is used for examinations conducted by private institution because it is easier for them to build a database of images for each students before the examination. Though public examination bodies may find it more challenging to build such database but they can also implement the system if the train set are available.

5.5 SUGGESTION FOR FURTHER STUDY

In this study, considerations were made for detecting only impersonation from video data, the system feature could get better and more effective if further study could be done on how to effectively implement video mining for other misconducts which could be present in an examination environment.

We also suggest that a means of building a database for public examination bodies is looked into for a wider use of the system.

REFERENCES

- Akaranga, S. I., & Ongong, J. J. (2013). The phenomenon of Examination Malpractice: An Example of Nairobi and Kenyatta Universities. *Journal of Education and Practice*, 4(18): 87 – 96
- Amos, B., B. Ludwiczuk, B., & M, Satyanarayanan.(2016), “Openface: a general-purpose face recognition library with mobile applications,” *Technical Report CMU-CS-16-118*, CMU School of Computer Science, Tech. Rep., 2016. Available: <http://reports-archive.adm.cs.cmu.edu/anon/anon/2016/CMU-CS-16-118.pdf>
- Anjulan, A. & Canagarajah, N. (2007). A novel video mining system. In: *Proceedings of 14th IEEE international conference on image processing*, San Antonio, Texas, pp 185–189
- Anzene, S. J. (2014). Trends in Examination Malpractice in Nigerian Educational System and its Effects on the Socio-Economic Development of Nigeria. *Asian Journal of Humanities and Social Sciences*, 2(3): 1 –8
- Arikan, O., Forsyth D.A., & O’Brien J.F. (2003), “Motion synthesis from annotations,” *ACM Transactions on Graphics*, vol. 22, no. 3, pp. 402–408.
- Benjamin, M., Osama, M. & Nikos, A. (2010). “Camera Surveillance of Crowded Traffic Scenes”, *IEEE Computer Society Press Vol. 22, No. 4*, pp.16-44.
- Burl, M.C. (2004) Mining Patterns of activity from video data. In: *Proceedings of the SIAM international conference on discrete mathematics*, pp 532–536
- Chen M, Chen S-C, & Shyu M-L (2007) Hierarchical temporal association mining for video event detection in video databases. In: *The second IEEE international workshop on multimedia databases and data management (MDDM’07), in conjunction with IEEE international conference on data engineering (ICDE2007)*, Istanbul, Turkey, pp 45-50
- Chen, F., Cooper, M. & Addock (2007) Video summarization preserving dynamic content. In: *Proceedings of the international workshop on TRECVID video summarization*, pp 40–44
- Chen, S.C, Chen, M., Zhang, C., Shyu, M. L. (2006) Exciting event detection using multi-level multimodal descriptors and data classification. In: *Proceedings of eighth IEEE international symposium on multimedia*, 193–200.
- Chitra, W. (2012), “Tools, Techniques And Models For Multimedia Database Mining”, *International Journal of Networking & Parallel Computing*, Volume 1, No 2, 1-5
- Dai, K., Zhang, J., Li, G. (2006) Video mining: concepts, approaches and applications. *Proc IEEE* 2006:477–481

- Dodge, Y. (2006) *The Oxford Dictionary of Statistical Terms*, Oxford University Press. ISBN 0-19-920613-9
- Emaikwu, S. O. (2012). Assessing the Impact of Examination Malpractices on the Measurement of Ability in Nigeria. *International Journal of Social Sciences & Education*, 2(4): 748 – 757
- Faloutsos, C., Hodgins, J., & Pollard, N. (2007). “Database techniques with motion capture,” in *SIGGRAPH '07: ACM SIGGRAPH 2007 courses 21. ACM*.
- Fleischman, M., Decamp, P. & Roy, D. (2006) Mining temporal patterns of movement for video content classification. In: *Proceedings of the 8th ACM international workshop on multimedia, information retrieval*, pp 183–192
- Fu, C.J., Li, G.H. & Dai, K.X. (2005) A framework for video structure mining. In: *Proceedings of the fourth international conference on machine learning and cybernetics*, vol 3, pp 1524–1528
- Gowsikhaa, D., Manjunath T. & Abiirami, S (2012), “Suspicious Human Activity Detection from surveillance Videos”, *International Journal on Internet and Distributed Computing Systems*. Vol: 2 No: 2, pp.141-149.
- Goyanil M, Dutta S, Gohil G & Naik S (2011) Wicket fall concept mining from cricket video using a-priori algorithm. *Proc Int J Multimedia Appl (IJMA)* 3:1 pp. 54-60
- Herley, C. (2005) “Accurate repeat finding and object skipping using fingerprints,” in *Proc. ACM Multimedia*, , pp. 656–665.
- Herley, C. (2006) “Argos: Automatically extracting repeating objects from multimedia streams,” *IEEE Trans. on Multimedia*, vol. 8, no. 1, pp. 115–129.
- Ian, H. (2005). Witten “Text mining”, Computer Science, University of Waikato, Hamilton, New Zealand.
- J. Meng, J. Yuan, M. Hans, & Y. Wu. (2008), “Mining motifs from human motion,” in *Proc. EUROGRAPHICS*, pp. 255–258.
- J. Yuan, J. Meng, Y. Wu, & J. Luo. (2008) “Mining recurring events through forest growing,” *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 18, no. 11, pp. 1597–1607.
- Hsu, J.L., Liu, C.C, & Chen, A. L (2001), “Discovering nontrivial repeating patterns in music data,” *IEEE Trans. on Multimedia*, vol. 3, no. 3, pp. 311–325.
- Jiawei, H. & Micheline, K. (2006) *Data Mining concepts and techniques*, 2nd edition, Morgan Kaufmann.
- Kovar, L., Gleicher, M., & Pighin F. (2002), “Motion graphs,” *ACM Transactions on Graphics*, vol.21, no. 3, pp. 473–482.

- Lai C, Rafa T & Nelson DE (2006) Mining motion patterns using color motion map clustering. *SIGKDD Explor*, 2,3–10
- Lin, L., Ravitz, G., Shyu, M.L. & Chen, S.C. (2007). Video semantic concept discovery using multimodal-based association classification. *ICME*, 07, pp. 45-50
- Manjunath T. N., Ravindra, S. H. & Ravikumar, G. K. (2010) “A Survey on Multimedia Data Mining and Its Relevance Today”, *IJCSNS International Journal of Computer Science and Network Security*, VOL.10 No.11, pp. 165-170.
- Manjunath, R., Balaji, S. (2008) Advances in Computer & Communication Engineering, Dept. of Computer Science & Engineering, City Engineering College, Bangalore-560061, India1 Centre for Emerging Technologies, Jain Global Campus, Jain University, Ramanagara Dist-562112, India *International Journal of Innovative Research in Computer and Communication Engineering* (ACCE - 2014) Vol.2, Special Issue 2
- Matsuo, Y., Amano, M. & Uehara, K. (2002) Mining video editing rules in video streams. In: *Proceedings of the tenth ACM international conference on multimedia*, pp 255–258
- Mongy, S., Bouali F & Djeraba, C. (2005) Analyzing user’s behavior on a video database. In: *Proceedings of the 6th ACM International workshop on multimedia data mining: mining integrated media and complex data (MDM/KDD)*, pp 95–100
- Moxley, E., Mei, T. & Manjunath, B.S. (2010) Video annotation through search and graph reinforcement mining. *IEEE Trans Multimedia* 12(3):184–194
- Muller, M., Roder, T., & Clausen, M. (2005), “Efficient content-based retrieval of motion capture data,” *ACM Transactions on Graphics*, vol. 24, no. 3, pp. 677–685.
- Nnam, M. U., & Inah, A. F. (2015). Empirical Investigation into the Causes, Forms and Consequences of Examination Malpractice in Nigerian Institutions of Higher Learning. *International Journal of Novel Research in Humanity and Social Sciences*, 2(1): 52 – 62
- Oh, J. & Bandi, B. (2002) Multimedia data mining framework for raw video sequences. In: *Proceedings of the third international workshop on multimedia data mining (MDM/KDD’2002) in conjunction with the eight ACM SIGKDD international conference on knowledge discovery & data mining*, pp 1–10.
- Rovai, A., Ponton, M. & Baker, J. (2008). Distance Learning in Higher Education: A programmatic Approach to Planning, Design, Instruction, Evaluation, and Accreditation. *Teachers College Press*. ISBN 978-0-8077-4878-7

- Shirahama K, Ideno K, & Uehara K (2005). Video data mining: mining semantic patterns with temporal constraints from movies. *In: Proceeding of seventh IEEE symposium on multimedia*, pp 598– 604
- Shrishrimal, O.P., Deshmukh, R. R. & Waghmare, V. B. (2015) “*Multimedia Data Mining: A Review*”, Dept of Computer Science & IT, Dr. Babasaheb Ambedkar Marathwada University, Aurangabad (MS) India.
- Siva, S. B & Gopalan, N.P (2007) Efficient algorithms for video association mining. *In: Proceedings of the 20th conference of the Canadian Society for computational studies of intelligence on advances in artificial intelligence*, vol 4509. Springer, Berlin, pp 250–260
- Sivic, J. & Zisserman, A. (2004) Video data mining using configurations of viewpoint invariant regions. *In: Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 488–495
- Su, J.H, Huang, Y.T, Tseng & V.S. (2008) Efficient content-based video retrieval by mining temporal patterns. *In: Proceedings of 9th international workshop on multimedia data mining associated with the ACM SIGKD*, pp 36–42
- Turaga, P.K, Veeraraghavan, A & Chellappa, R.(2007) From videos to verbs: mining videos for activities using a cascade of dynamical systems. *In: Proceedings of IEEE conference on computer vision and pattern recognition*, pp 1–8
- Vailaya, A, Jain, A. & Zhang, H (2011) Video clustering. *Tech Rep. No. MSU-CPS-96-64*, Michigan State University, USA
- Venkatadri, M. & Lokanatha, C.(2011), “A Review on Data mining from Past to the Future”, *International Journal of Computer Application (IJCA)*, Volume 15– No.7, pp. 19-22.
- Vijayakumar, V & Nedunchezian .N,(2012) “A study on video data mining”, *International Journal of Multimedia Information, Retrieval*, October 2012, *Volume 1, Issue 3*, Springer-Verlag, pp 153-172,
- Wang, P.,Liu Z.Q, & Yang, S.Q (2005), “A probabilistic template-based approach to discovering repetitive patterns in broadcast videos,” *in Proc. ACM Multimedia*, pp.407–410.
- Wang, Y., Xing, C. & Zhou, L. (2006) Video semantic models: survey and evaluation. *Int J Comput Sci Network Sec* 6(2A):10–21
- Weber, J., Lefever, S. & Gancarski, P. (2010) Video object mining: issues and perspectives. *In: Proceedings of IEEE fourth international conference on semantic computing*, pp 85–91
- Wu, C., He, Y., Zhao L.& Zhong Y (2002). Motion feature extraction scheme for content-based video retrieval, storage and retrieval for media databases. *Proc SPIE* 4676:296–305

- Xie, L. & Chang, S.F. (2006) Pattern mining in visual concept streams. *In: Proceedings of IEEE international conference on multimedia and expo (ICME06)*, Toronto, Canada
- Yang, X., Tian, Q., & Xue, P., “Efficient short video repeat identification with application on news video structure analysis,” *IEEE Trans. on Multimedia*, vol. 9, no. 3, pp. 600–609, 2007.
- Yang, X.F, Tian, Q. & Xue, P. (2007). Efficient short video repeat identification with application to news video structure analysis. *IEEE Trans Multimedia* 9(3):600–610
- Zang, Q. & Klette, R. (2003) Object classification and tracking in video surveillance. Communication and Information Technology Research Technical Report 128
- Zhang, X., Mamoulis N., Cheung, D.W., & Shou, Y. (2004) “Fast mining of spatial collocations,” *in Proc. ACM SIGKDD*, 2004.
- Zhao, N., Chen, S.C. & Shyu M.L. (2006) Video database modeling and temporal pattern retrieval using hierarchical Markov model mediator. *In: Proceedings of first IEEE international workshop on multimedia databases & data Management*, pp 10–19
- Zhu, X., Aref, W., Fan, J., Catlina, A. & Elmagarmid, A. (2003) Medical video mining for efficient database indexing, management and access. *In: Proceedings 19th international conference on data engineering, issue 5–8*, pp 569–580
- Zhu, X., Wu, X., Elmagarmid, A., Feng, Z. & Wu, L. (2005) Video data mining: semantic indexing and event detection from the association perspective. *IEEE Trans Knowl Data Eng* 17(5):1–14

APPENDIX A

FRONT END CODE

MAIN LAYOUTS

```
<!DOCTYPE html>

<html lang="{{ str_replace('_', '-', app()->getLocale()) }}">

<head>

    <meta charset="utf-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1">


    <!-- CSRF Token -->

    <meta name="csrf-token" content="{{ csrf_token() }}">


    <title>{{ config('app.name', 'Laravel') }}</title>


    <!-- Scripts -->

    <script src="{{ asset('js/app.js') }}" defer></script>


    <!-- Fonts -->

    <link rel="dns-prefetch" href="https://fonts.gstatic.com">

    <link href="https://fonts.googleapis.com/css?family=Nunito" rel="stylesheet" type="text/css">


    <!-- Styles -->

    <link href="{{ asset('css/app.css') }}" rel="stylesheet">

    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/open-iconic@1.1.1/font/css/open-iconic-bootstrap.min.css" />

    <!-- <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css"
    integrity="sha384-MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO"
    crossorigin="anonymous"> -->

    <style type="text/css">

        .icon {

            color: #fff;

        }

    </style>

</head>

<body>
```

```

<div id="app">

  <nav class="navbar navbar-expand-md navbar-light navbar-laravel">

    <div class="container">

      <a class="navbar-brand" href="{{ url('/') }}">

        <span class="oi oi-play-circle"></span>

        {{ config('app.name', 'Laravel') }}

      </a>

      <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-
label="{{ __('Toggle navigation') }}">

        <span class="navbar-toggler-icon"></span>

      </button>

      <div class="collapse navbar-collapse" id="navbarSupportedContent">

        <!-- Left Side Of Navbar -->

        <ul class="navbar-nav mr-auto">


        </ul>

        <!-- Right Side Of Navbar -->

        <ul class="navbar-nav ml-auto">

          <!-- Authentication Links -->

          <form class="form-inline my-2 my-lg-0 mr-2">

            <input class="form-control mr-sm-2" type="search" placeholder="Search">

            <button class="btn btn-outline-success my-2 my-sm-0" type="submit">Search</button>

          </form>

          @guest

            <li class="nav-item">

              <a class="nav-link" href="{{ route('login') }}">{{ __('Login') }}</a>

            </li>

            <li class="nav-item">

              <a class="nav-link" href="{{ route('register') }}">{{ __('Register') }}</a>

            </li>

          @else

            <li class="nav-item dropdown">

              <a id="navbarDropdown" class="nav-link dropdown-toggle" href="#" role="button" data-
toggle="dropdown" aria-haspopup="true" aria-expanded="false" v-pre>

                {{ Auth::user()->firstname }} <span class="caret"></span>

```


<div class="dropdown-menu dropdown-menu-right" aria-labelledby="navbarDropdown">

Change Password

<a class="dropdown-item" href="{{ route('logout') }}"

onclick="event.preventDefault();

document.getElementById('logout-form').submit();">

{{ __('Logout') }}

<form id="logout-form" action="{{ route('logout') }}" method="POST" style="display: none;">

@csrf

</form>

</div>

@endguest

</div>

</div>

</nav>

<style type="text/css">

.list-group-item.active {

z-index: 2;

color: #fff;

background-color: #37d469;

border-color: #37d469;

}

</style>

<main class="py-4">

@if(Auth::check())

<div class="container">

<div class="row justify-content-center">

```
<div class="col-md-3 col-lg-3">
```

```
<ul class="list-group flex-column">
```

```
<a class="list-group-item list-group-item-action {{ Route::is('home') ? 'active' : '' }}"
href="{{ route('home') }}">
```

```
<span class="oi oi-home"></span>
```

```
Dashboard
```

```
</a>
```

```
<a class="list-group-item list-group-item-action {{ Route::is('uploads') ? 'active' : '' }}"
href="{{ route('uploads') }}">
```

```
<span class="oi oi-cloud-upload"></span>
```

```
Uploads
```

```
<span class="badge badge-pill pull-right {{ Route::is('uploads') ? 'badge-light' : 'badge-
primary' }}">5</span>
```

```
</a>
```

```
<a class="list-group-item list-group-item-action" href="#">
```

```
<span class="oi oi-loop"></span>
```

```
Processed
```

```
<span class="badge badge-primary badge-pill pull-right">3</span>
```

```
</a>
```

```
<a class="list-group-item list-group-item-action" href="#">
```

```
<span class="oi oi-wrench"></span>
```

```
Settings
```

```
</a>
```

```
<a class="list-group-item list-group-item-action " href="{{ route('logout') }}"
```

```
onclick="event.preventDefault();
```

```
document.getElementById('logout-form').submit();">
```

```
<span class="oi oi-power-standby"></span> {{ __('Logout') }}
```

```
</a>
```

```
</ul>
```

```
</div>
```

```
<div class="col-md-9 col-lg-9">
```

```
@yield('content')
```

```
</div>
```

```
</div>
```

```
</div>
```

```
@else
```


@yield('content')

@endif

</main>

<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>

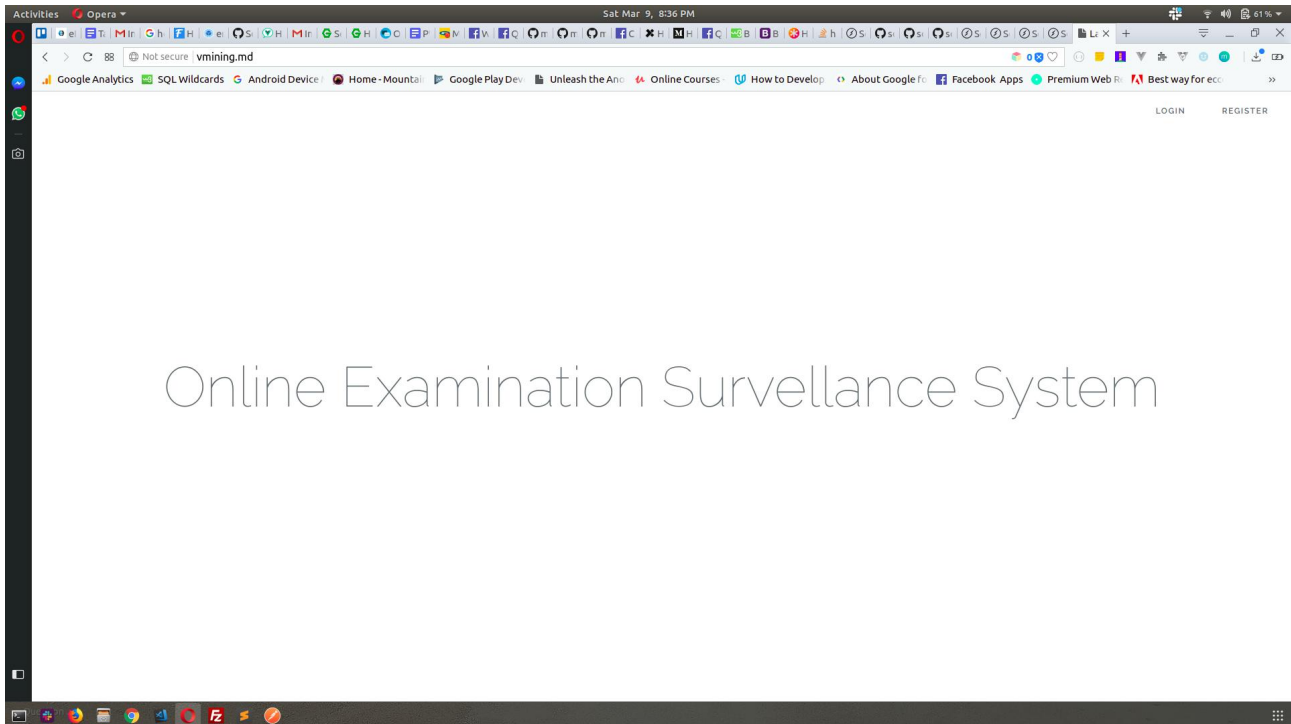
<!-- <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js" integrity="sha384-ZMP7rVo3mIykV+2+9J3UJ46jBk0WLaUAAdn689aCwoqbBJiSnjAK/l8WvCWPIpM49" crossorigin="anonymous"></script> -->

@stack("script")

</body>

</html>

LANDING PAGE



```
<!doctype html>
<html lang="{{ app()->getLocale() }}">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <title>Video Mining</title>

    <!-- Fonts -->
    <link href="https://fonts.googleapis.com/css?family=Raleway:100,600" rel="stylesheet" type="text/css">

    <!-- Styles -->
    <style>
      html, body {
        background-color: #fff;
        color: #636b6f;
        font-family: 'Raleway', sans-serif;
```

```
font-weight: 100;
height: 100vh;
margin: 0;
}
```

```
.full-height {
  height: 100vh;
}
```

```
.flex-center {
  align-items: center;
  display: flex;
  justify-content: center;
}
```

```
.position-ref {
  position: relative;
}
```

```
.top-right {
  position: absolute;
  right: 10px;
  top: 18px;
}
```

```
.content {
  text-align: center;
}
```

```
.title {
  font-size: 84px;
}
```

```
.links > a {
  color: #636b6f;
  padding: 0 25px;
```

```
font-size: 12px;

font-weight: 600;

letter-spacing: .1rem;

text-decoration: none;

text-transform: uppercase;

}
```

```
.m-b-md {

margin-bottom: 30px;

}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<div class="flex-center position-ref full-height">
```

```
@if (Route::has('login'))
```

```
<div class="top-right links">
```

```
@auth
```

```
<a href="{{ route('home') }}">Dashboard</a>
```

```
@else
```

```
<a href="{{ route('login') }}">Login</a>
```

```
<a href="{{ route('register') }}">Register</a>
```

```
@endauth
```

```
</div>
```

```
@endif
```

```
<div class="content">
```

```
<div class="title m-b-md">
```

```
Online Examination Surveillance System
```

```
</div>
```

```
<!-- <div class="links">
```

```
<a href="https://laravel.com/docs">Documentation</a>
```

```
<a href="https://laracasts.com">Laracasts</a>
```

```
<a href="https://laravel-news.com">News</a>
```

```
<a href="https://forge.laravel.com">Forge</a>
```

```
<a href="https://github.com/laravel/laravel">GitHub</a>
```

```

</div> -->

</div>

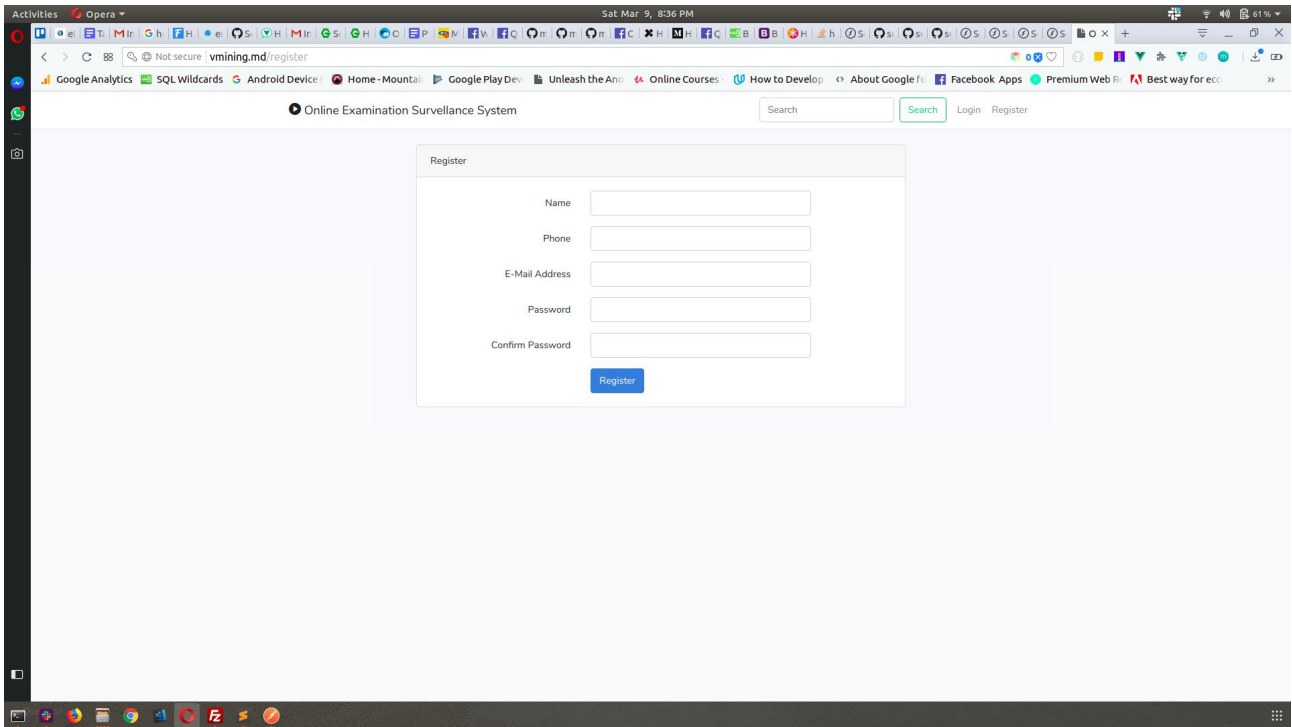
</div>

</body>

</html>

```

REGISTER



```
@extends('layouts.app')
```

```
@section('content')
```

```
<div class="container">
```

```
<div class="row justify-content-center">
```

```
<div class="col-md-8">
```

```
<div class="card">
```

```
<div class="card-header">{{ __('Register') }}</div>
```

```
<div class="card-body">
```

```
<form method="POST" action="{{ route('register') }}" aria-label="{{ __('Register') }}">
```

```
@csrf
```

```
<div class="form-group row">

  <label for="name" class="col-md-4 col-form-label text-md-right">{{ __('Name') }}</label>
```

```
  <div class="col-md-6">
```

```
    <input id="name" type="text" class="form-control{{ $errors->has('name') ? ' is-invalid' : '' }}"
name="name" value="{{ old('name') }}" required autofocus>
```

```
    @if ($errors->has('name'))
```

```
      <span class="invalid-feedback" role="alert">
```

```
        <strong>{{ $errors->first('name') }}</strong>
```

```
      </span>
```

```
    @endif
```

```
  </div>
```

```
</div>
```

```
<div class="form-group row">
```

```
  <label for="phone" class="col-md-4 col-form-label text-md-right">{{ __('Phone') }}</label>
```

```
  <div class="col-md-6">
```

```
    <input id="phone" type="tel" class="form-control{{ $errors->has('phone') ? ' is-invalid' : '' }}"
name="phone" value="{{ old('phone') }}" required>
```

```
    @if ($errors->has('phone'))
```

```
      <span class="invalid-feedback" role="alert">
```

```
        <strong>{{ $errors->first('phone') }}</strong>
```

```
      </span>
```

```
    @endif
```

```
  </div>
```

```
</div>
```

```
<div class="form-group row">
```

```
  <label for="email" class="col-md-4 col-form-label text-md-right">{{ __('E-Mail Address') }}</label>
```

```
  <div class="col-md-6">
```

```
<input id="email" type="email" class="form-control{{ $Errors->has('email') ? ' is-invalid' : " }}"
name="email" value="{{ old('email') }}" required>
```

```
@if ($Errors->has('email'))
```

```
<span class="invalid-feedback" role="alert">
```

```
<strong>{{ $Errors->first('email') }}</strong>
```

```
</span>
```

```
@endif
```

```
</div>
```

```
</div>
```

```
<div class="form-group row">
```

```
<label for="password" class="col-md-4 col-form-label text-md-right">{{ __('Password') }}</label>
```

```
<div class="col-md-6">
```

```
<input id="password" type="password" class="form-control{{ $Errors->has('password') ? ' is-
invalid' : " }}" name="password" required>
```

```
@if ($Errors->has('password'))
```

```
<span class="invalid-feedback" role="alert">
```

```
<strong>{{ $Errors->first('password') }}</strong>
```

```
</span>
```

```
@endif
```

```
</div>
```

```
</div>
```

```
<div class="form-group row">
```

```
<label for="password-confirm" class="col-md-4 col-form-label text-md-right">{{ __('Confirm
Password') }}</label>
```

```
<div class="col-md-6">
```

```
<input id="password-confirm" type="password" class="form-control"
name="password_confirmation" required>
```

```
</div>
```

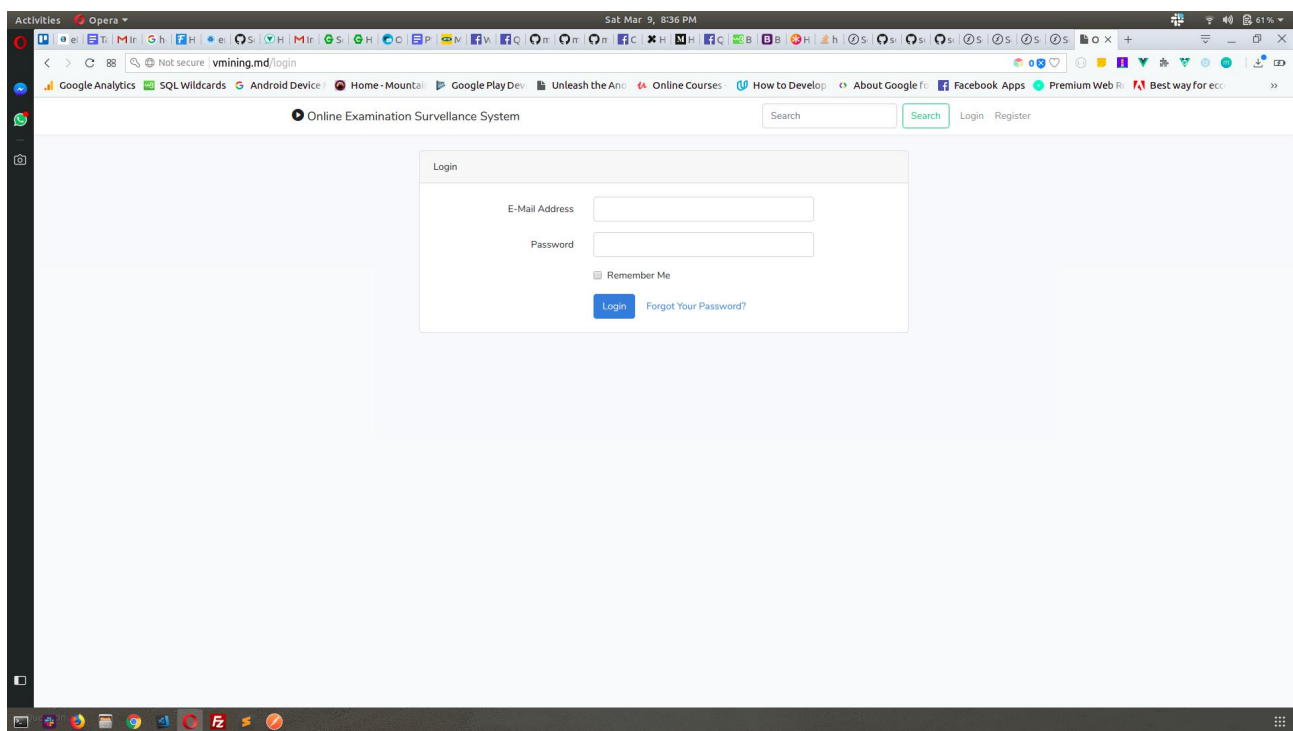
```
</div>
```

```

<div class="form-group row mb-0">
  <div class="col-md-6 offset-md-4">
    <button type="submit" class="btn btn-primary">
      {{ __('Register') }}
    </button>
  </div>
</div>
</div>
</form>
</div>
</div>
</div>
</div>
</div>
</div>
@endsection

```

LOGIN



```
@extends('layouts.app')
```

```
@section('content')
```

```
<div class="container">
```



```

<div class="row justify-content-center">
  <div class="col-md-8">
    <div class="card">
      <div class="card-header">{{ __( 'Login' ) }}</div>

      <div class="card-body">
        <form method="POST" action="{{ route('login') }}" aria-label="{{ __( 'Login' ) }}">
          @csrf

          <div class="form-group row">
            <label for="email" class="col-sm-4 col-form-label text-md-right">{{ __( 'E-Mail Address' ) }}</label>

            <div class="col-md-6">
              <input id="email" type="email" class="form-control{{ $Errors->has('email') ? ' is-invalid' : '' }}"
name="email" value="{{ old('email') }}" required autofocus>

              @if ($Errors->has('email'))
                <span class="invalid-feedback" role="alert">
                  <strong>{{ $Errors->first('email') }}</strong>
                </span>
              @endif
            </div>
          </div>

          <div class="form-group row">
            <label for="password" class="col-md-4 col-form-label text-md-right">{{ __( 'Password' ) }}</label>

            <div class="col-md-6">
              <input id="password" type="password" class="form-control{{ $Errors->has('password') ? ' is-
invalid' : '' }}" name="password" required>

              @if ($Errors->has('password'))
                <span class="invalid-feedback" role="alert">
                  <strong>{{ $Errors->first('password') }}</strong>
                </span>
              @endif
            </div>
          </div>
        </form>
      </div>
    </div>
  </div>
</div>

```

</div>

</div>

<div class="form-group row">

<div class="col-md-6 offset-md-4">

<div class="form-check">

<input class="form-check-input" type="checkbox" name="remember" id="remember"
{{ old('remember') ? 'checked' : '' }}>

<label class="form-check-label" for="remember">

{{ __('Remember Me') }}

</label>

</div>

</div>

</div>

<div class="form-group row mb-0">

<div class="col-md-8 offset-md-4">

<button type="submit" class="btn btn-primary">

{{ __('Login') }}

</button>

{{ __('Forgot Your Password?') }}

</div>

</div>

</form>

</div>

</div>

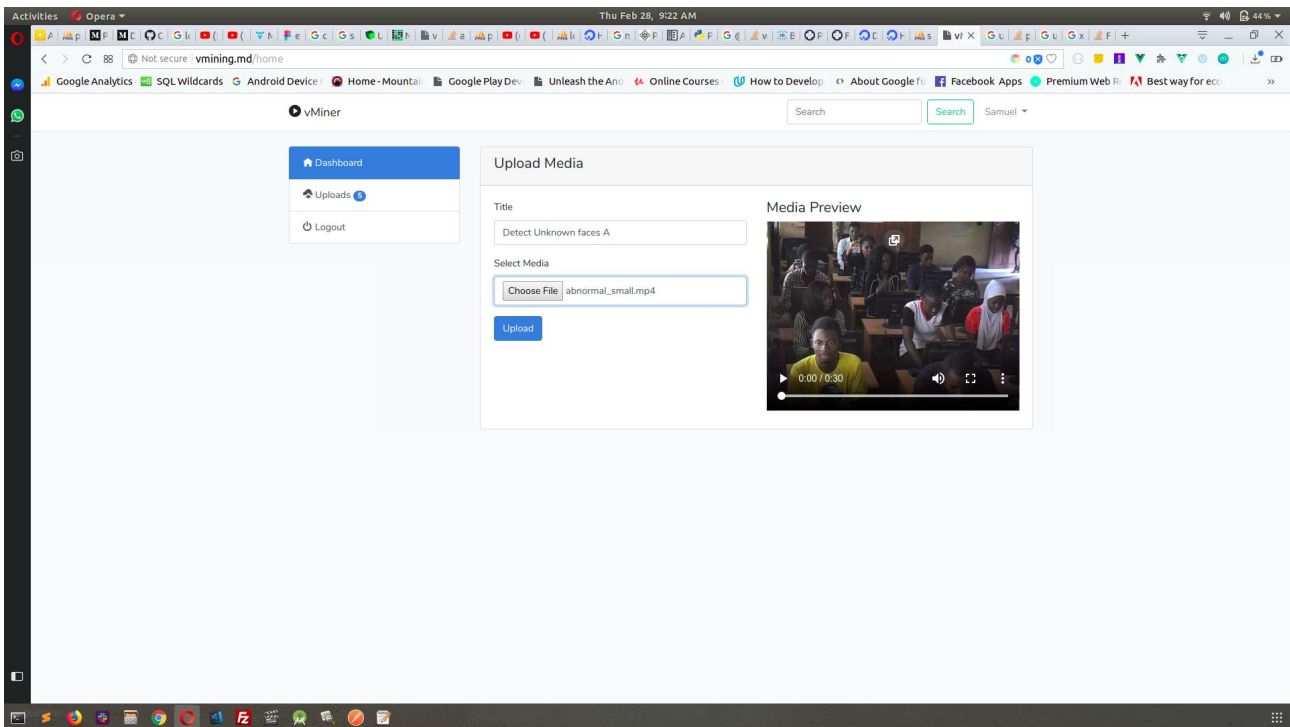
</div>

</div>

</div>

@endsection

DASHBOARD



```
@extends('layouts.app')
```

```
@section('content')
```

```
<div class="card">
```

```
<div class="card-header">
```

```
<h4>Upload Media</h4>
```

```
</div>
```

```
<div class="card-body">
```

```
@if (session('status'))
```

```
<div class="alert alert-success" role="alert">
```

```
{{ session('status') }}
```

```
</div>
```

```
@endif
```

```
<div class="row">
```

```
<div class="col-md-6">
```

```

    <form enctype='multipart/form-data' action="{{route('post_uploads')}}" method="POST"
enctype="multipart/form-data">
        {{ csrf_field() }}
        <div class="form-group">
            <label for="formGroupExampleInput">Title</label>
            <input type="text" class="form-control" id="title" placeholder="Title of Video" name="title"
required>
        </div>
        <div class="form-group">
            <label for="formGroupExampleInput2">Select Media</label>
            <input type="file" name="media" class="form-control" id="selectMedia">
        </div>

        <button type="submit" class="btn btn-primary">Upload</button>
    </form>
</div>
<div class="col-md-6" style="width: 100% width:auto;">
    <h4>Media Preview</h4>

    <video controls="controls" name="media" buffered="true" preload="true" style="width: inherit;"
id="videoTag" required>

        Your browser does not support the video tag.
    </video>
</div>
</div>
</div>
</div>

```

@endsection

@push("script")

```
<script type="text/javascript">
```

```

$("document").ready(function(){
    function readURL(input) {
        if (input.files && input.files[0]) {
            var reader = new FileReader();
            reader.onload = function(e) {

```

```
$("#videoTag").html('<source name="media" src="'+ e.target.result +' " type="video/mp4"></source>');
Your browser does not support the video tag.' );
```

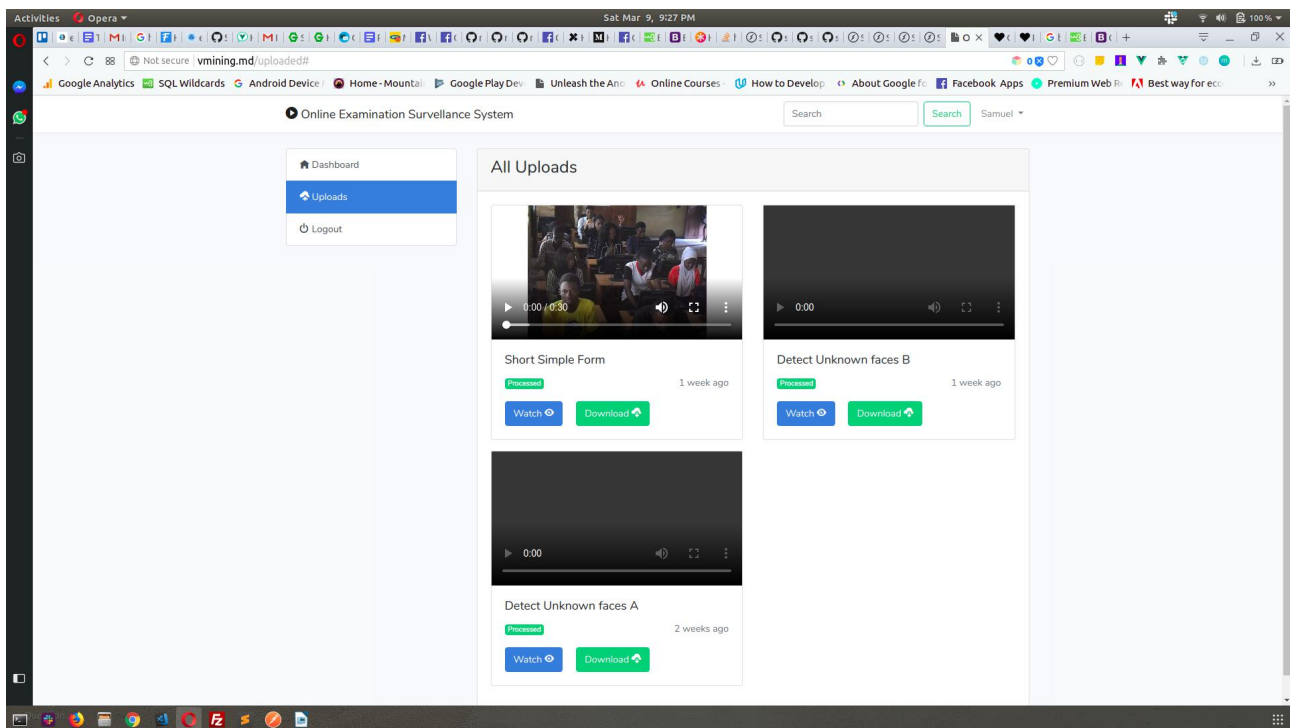
```
    }
    reader.readAsDataURL(input.files[0]);
  }
}
```

```
$("#selectMedia").change(function(){
    readURL(this)
    $("#videoTag").load();
});
});
```

```
</script>
```

```
@endpush
```

UPLOADED MEDIA WITH STATUS



```
@extends('layouts.app')
```

```
@section('Title','Uploads')
```

```
@section('content')
```

```
<div class="card">
```

```
<div class="card-header">
```

```
<h3>All Uploads</h3>
```

```
</div>
```

```
<div class="card-body">
```

```
<div class="row">
```

```
@foreach($result as $key => $res)
```

```
<div class="col-md-6 mb-3">
```

```
<div class="card">
```

```
<video controls="controls" height="200px" name="media" buffered="true" preload="true" style="width: inherit;" id="videoTag{{ $key }}" required>
```

```
<source name="media" src="{{ $res->path }}" type="video/mp4" ></source>
```

```
Your browser does not support the video tag.
```

```
</video>
```

```
<div class="card-body">
```

```
<h5 class="card-title">{{ $res->title }}</h5>
```

```
<p class="card-text text-muted">
```

```
@if($res->status)
```

```
<span class="badge badge-success">Processed</span>
```

```
@else
```

```
<span class="badge badge-warning">Pending</span>
```

```
@endif
```

```
<span class="pull-right" style="float: right;">{{ $res->created_at->diffForHumans() }}</span>
```

```
</p>
```

```

        <a href="{{ $res->path }}" class="btn btn-primary mr-3" target="_blank">Watch <span class="oi oi-eye
icon"></span> </a>

        <a href="{{ $res->path }}" class="btn btn-success">Download <span class="oi oi-cloud-download
icon"></span> </a>

    </div>

</div>

</div>

    @endforeach

</div>

</div>

</div>

@endsection

```

OTHERS FUNCTION

```
<?php
```

```
namespace App\Http\Controllers;
```

```
use Illuminate\Http\Request;
```

```
use App\Upload;
```

```
use Storage;
```

```
class HomeController extends Controller
```

```

{
    /**
     * Create a new controller instance.
     *
     * @return void
     */
    public function __construct()
    {
        $this->middleware('auth');
    }
}

```

```

/**
 * Show the application dashboard.
 *
 * @return \Illuminate\Http\Response
 */

public function index()
{
    return view('home');
}


public function uploads()
{
    $result = \App\Upload::orderBy('id', 'DESC')->get();
    return view('uploads', compact('result'));
}


public function processed()
{
    $result = \App\Upload::orderBy('id', 'DESC')->get();
    return view('processed', compact('result'));
}


public function doUploads(Request $request)
{
    $this->validate($request, [
        'title' => 'required',
        'media' => 'required',
    ]);

    $media = $request->file('media');

    $folder = 'media/video';

    $img_name = rand(1111111111111111,9999999999999999).time().'.'.$request->media-
    >getClientOriginalExtension();

```



```
$destination = env("APP_URL") .'/storage/'.$folder;
```

```
$storage = Storage::put($folder, $media);
```

```
$path = Storage::url($storage);
```

```
$data['storage'] = $storage;
```

```
$data['path'] = $path;
```

```
$data = array_except($data, ['medai']);
```

```
// Save to DB
```

```
$status = Upload::firstOrCreate([
```

```
    'title' => $request->title,
```

```
    'path' => $data['path'],
```

```
    'location' => $storage,
```

```
]);
```

```
if ($status) {
```

```
    return redirect()->route('processed');
```

```
}
```

```
}
```

```
}
```

REGISTER

```
<?php
```

```
namespace App\Http\Controllers\Auth;
```

```
use App\User;
```

```
use App\Http\Controllers\Controller;
```

```
use Illuminate\Support\Facades\Hash;
```

```
use Illuminate\Support\Facades\Validator;
```

```
use Illuminate\Foundation\Auth\RegistersUsers;
```

```
class RegisterController extends Controller
```

```

{
    /*
    |-----
    | Register Controller
    |-----
    |
    | This controller handles the registration of new users as well as their
    | validation and creation. By default this controller uses a trait to
    | provide this functionality without requiring any additional code.
    |
    */

    use RegistersUsers;

    /**
     * Where to redirect users after registration.
     *
     * @var string
     */
    protected $redirectTo = '/home';

    /**
     * Create a new controller instance.
     *
     * @return void
     */
    public function __construct()
    {
        $this->middleware('guest');
    }

    /**
     * Get a validator for an incoming registration request.
     *
     * @param array $data
     * @return \Illuminate\Contracts\Validation\Validator

```

```

*/

protected function validator(array $data)
{
    return Validator::make($data, [
        //'firstname' => 'required|string|max:255',
        //'lastname' => 'required|string|max:255',
        'name' => 'required|string|max:255',
        'email' => 'required|string|email|max:255|unique:users',
        'phone' => 'required|string|max:255|unique:users',
        'password' => 'required|string|min:6|confirmed',
    ]);
}

/**
 * Create a new user instance after a valid registration.
 *
 * @param array $data
 * @return \App\User
 */

protected function create(array $data)
{
    $name = explode(" ", $data['name']);
    return User::create([
        'firstname' => $name[0],
        'lastname' => $name[1],
        'phone' => $data['phone'],
        'email' => $data['email'],
        'ipaddress' => $_SERVER['REMOTE_ADDR'],
        'password' => Hash::make($data['password']),
        'verifiedPhone' => 1,
    ]);
}
}

```

LOGIN

```
<?php
```

```
namespace App\Http\Controllers\Auth;
```

```
use App\Http\Controllers\Controller;
```

```
use Illuminate\Foundation\Auth\AuthenticatesUsers;
```

```
class LoginController extends Controller
```

```
{
```

```
    /*
```

```
    |-----
```

```
    | Login Controller
```

```
    |-----
```

```
    |
```

```
    | This controller handles authenticating users for the application and
```

```
    | redirecting them to your home screen. The controller uses a trait
```

```
    | to conveniently provide its functionality to your applications.
```

```
    |
```

```
    */
```

```
use AuthenticatesUsers;
```

```
/**
```

```
 * Where to redirect users after login.
```

```
 *
```

```
 * @var string
```

```
 */
```

```
protected $redirectTo = '/home';
```

```
/**
```

```
 * Create a new controller instance.
```

```
 *
```

```
 * @return void
```

```
 */
```

```
public function __construct()
```

```
{  
    $this->middleware('guest')->except('logout');  
}  
}
```

APPENDIX B

PYTHON API CODE

CREATE HOG ENCODING

```
# USAGE

# python encode_faces.py --dataset dataset --encodings encodings.pickle
# python3 encode_faces.py --dataset dataset --encodings encodings.pickle


# import the necessary packages

from imutils import paths

import face_recognition

import argparse

import pickle

import cv2

import os


# construct the argument parser and parse the arguments

ap = argparse.ArgumentParser()
ap.add_argument("-i", "--dataset", required=True,
                help="path to input directory of faces + images")
ap.add_argument("-e", "--encodings", required=True,
                help="path to serialized db of facial encodings")
ap.add_argument("-d", "--detection-method", type=str, default="cnn",
                help="face detection model to use: either `hog` or `cnn`")
args = vars(ap.parse_args())


# grab the paths to the input images in our dataset

print("[INFO] quantifying faces...")

imagePaths = list(paths.list_images(args["dataset"]))


# initialize the list of known encodings and known names

knownEncodings = []

knownNames = []


# Function to crop faces out of image using face haarcascade

def facechop(image):
```

```

facedata = "haarcascade_frontalface_default.xml"
cascade = cv2.CascadeClassifier(facedata)

img = cv2.imread(image)

minisize = (img.shape[1],img.shape[0])
miniframe = cv2.resize(img, minisize)

faces = cascade.detectMultiScale(miniframe)

for f in faces:
    x, y, w, h = [ v for v in f ]
    cv2.rectangle(img, (x,y), (x+w,y+h), (255,255,255))

    sub_face = img[y:y+h, x:x+w]
    face_file_name = "faces/face_" + str(y) + ".jpg"
    cv2.imwrite(face_file_name, sub_face)

#cv2.imshow(image, img)

return face_file_name

```

```

# loop over the image paths
for (i, imagePath) in enumerate(imagePaths):
    # extract the person name from the image pathy
    print (imagePath)
    print("[INFO] processing image {}/{}".format(i + 1,
        len(imagePaths)))
    name = imagePath.split(os.path.sep)[-2]

    # load the input image and convert it from RGB (OpenCV ordering)
    # to dlib ordering (RGB)
    #imagePath_small = cv2.resize(imagePath, (0,0), fx=0.5, fy=0.5)
    #imagePath_small = facechop(imagePath)

```



```

image = cv2.imread(imagePath)

#image = imagePath_small

#print (image)

rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)


# detect the (x, y)-coordinates of the bounding boxes
# corresponding to each face in the input image
boxes = face_recognition.face_locations(rgb,
    model=args["detection_method"])


# compute the facial embedding for the face
encodings = face_recognition.face_encodings(rgb, boxes)


# loop over the encodings
for encoding in encodings:

    # add each encoding + name to our set of known names and
    # encodings
    knownEncodings.append(encoding)
    knownNames.append(name)


# dump the facial encodings + names to disk
print("[INFO] serializing encodings...")
data = {"encodings": knownEncodings, "names": knownNames}
f = open(args["encodings"], "wb")
f.write(pickle.dumps(data))
f.close()

```

DETECT FACES IN A VIDEO FILE AND LABEL IT

```

# USAGE

# python recognize_faces_video.py --encodings encodings.pickle

# python3 recognize_faces_video.py --encodings encodings.pickle --input videos/abnormal_scene.mp4
--output output/abnormal_scene_output.avi --display 0


# import the necessary packages
from imutils.video import VideoStream
from imutils.video import FileVideoStream

```

```

import face_recognition
import argparse
import imutils
import pickle
import time
import cv2

# construct the argument parser and parse the arguments
ap = argparse.ArgumentParser()
ap.add_argument("-e", "--encodings", required=True,
                help="path to serialized db of facial encodings")
ap.add_argument("-o", "--output", type=str,
                help="path to output video")
ap.add_argument("-y", "--display", type=int, default=1,
                help="whether or not to display output frame to screen")
ap.add_argument("-d", "--detection-method", type=str, default="cnn",
                help="face detection model to use: either `hog` or `cnn`")
args = vars(ap.parse_args())

# load the known faces and embeddings
print("[INFO] loading encodings...")
data = pickle.loads(open(args["encodings"], "rb").read())

# initialize the video stream and pointer to output video file, then
# allow the camera sensor to warm up
print("[INFO] starting video stream...")
#vs = VideoStream(src=0).start()
#vs = VideoStream(args["input"]).start()
#vs = FileVideoStream(args["input"]).start()
vs = cv2.VideoCapture(args["input"])
writer = None
time.sleep(2.0)
count = 0

# loop over frames from the video file stream
while True:

```

```

# grab the frame from the threaded video stream
frame = vs.read()

# convert the input frame from BGR to RGB then resize it to have
# a width of 750px (to speedup processing)
rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
rgb = imutils.resize(rgb, width=750)
r = frame.shape[1] / float(rgb.shape[1])

# detect the (x, y)-coordinates of the bounding boxes
# corresponding to each face in the input frame, then compute
# the facial embeddings for each face
boxes = face_recognition.face_locations(rgb,
    model=args["detection_method"])
encodings = face_recognition.face_encodings(rgb, boxes)
names = []

# loop over the facial embeddings
for encoding in encodings:
    # attempt to match each face in the input image to our known
    # encodings
    matches = face_recognition.compare_faces(data["encodings"],
        encoding)
    name = "Unknown"

    # check to see if we have found a match
    if True in matches:
        # find the indexes of all matched faces then initialize a
        # dictionary to count the total number of times each face
        # was matched
        matchedIdxs = [i for (i, b) in enumerate(matches) if b]
        counts = {}

        # loop over the matched indexes and maintain a count for
        # each recognized face face
        for i in matchedIdxs:

```

```

        name = data["names"][i]

        counts[name] = counts.get(name, 0) + 1

    # determine the recognized face with the largest number
    # of votes (note: in the event of an unlikely tie Python
    # will select first entry in the dictionary)
    name = max(counts, key=counts.get)

    # update the list of names
    names.append(name)

# loop over the recognized faces
for ((top, right, bottom, left), name) in zip(boxes, names):
    # rescale the face coordinates
    top = int(top * r)
    right = int(right * r)
    bottom = int(bottom * r)
    left = int(left * r)

    # draw the predicted face name on the image
    cv2.rectangle(frame, (left, top), (right, bottom),
                  (0, 255, 0), 2)

    y = top - 15 if top - 15 > 15 else top + 15
    cv2.putText(frame, name, (left, y), cv2.FONT_HERSHEY_SIMPLEX,
                0.75, (0, 255, 0), 2)

# if the video writer is None *AND* we are supposed to write
# the output video to disk initialize the writer
if writer is None and args["output"] is not None:
    fourcc = cv2.VideoWriter_fourcc(*"MJPG")
    writer = cv2.VideoWriter(args["output"], fourcc, 20,
                            (frame.shape[1], frame.shape[0]), True)

# if the writer is not None, write the frame with recognized
# faces to disk
if writer is not None:

```

```
writer.write(frame)

# check to see if we are supposed to display the output frame to
# the screen
if args["display"] > 0:
    cv2.imshow("Frame", frame)
    key = cv2.waitKey(1) & 0xFF

    # if the `q` key was pressed, break from the loop
    if key == ord("q"):
        break

# Increase Frame counted
count += 1

# do a bit of cleanup
cv2.destroyAllWindows()
vs.stop()

# check to see if the video writer point needs to be released
if writer is not None:
    writer.release()
```

SPLIT VIDEOS INTO FRAMES

```
import os

import errno

import cv2

rootPath = '/home/samalapsy/Desktop/vminer/frames/unknown'

# make Directory if it;s not existint
if not os.path.exists(rootPath):
    os.makedirs(rootPath)

# os.makedirs(os.path.dirname(rootPath), exist_ok=True)

print(cv2.__version__)

#vidcap = cv2.VideoCapture('sample.mp4')
vidcap = cv2.VideoCapture('abnormal_small_hog_output.avi')
success,image = vidcap.read()
count = 0
success = True
while success:

    cv2.imwrite(os.path.join(rootPath, "frame%d.jpg" % count), image)    # save frame as JPEG file
    success,image = vidcap.read()
    print ('Read a new frame: ', success)
    count += 1
```