# A MAJOR -PROJECT

**ON**

**VIDEO OBJECT FORGERY DETECTION**

Submitted in partial fulfilment of the requirement for the

# BACHELOR OF TECHNOLOGY IN

**INFORMATION TECHNOLOGY**

**BY**

**MD.RAFE RAHEEL - 17R91A1228**

**R.SAI CHANDRA KIRAN - 17R91A1238**

**SAI PREETHAM GOUD.CH - 17R91A1242**

Under the esteemed guidance of

# DR. J. PRAVEEN KUMAR, Associate Professor

# Image result for tkr college of LOGO

**DEPARTMENT OF INFORMATIONTECHNOLOGY**

**TEEGALA KRISHNA REDDY ENGINEERING COLLEGE**

**(Affiliated to JNTUH, Hyderabad, Approved by AICTE, Accredited by NBA, NAAC with A Grade)**

**Medbowli, Meerpet, Saroornagar, Hyderabad – 500097.**

**(2017-2021)**

**TEEGALA KRISHNA REDDY ENGINEERING COLLEGE**

**DEPARTMENT OF INFORMATION TECHNOLOGY**

**Jawaharlal Nehru Technological University**

**Hyderabad – 500074**



**DECLARATION BY CANDIDATE**

We, **SAI PREETHAM GOUD CH, MD RAFE RAHEEL, SAI CHADRA KIRAN**, here by certify that the project we entitled “**VIDEO OBJECT FORGERY DETECTION**” is done under the guidance of **DR. J. PRAVEEN KUMAR**, Associate Professor, Department of Information Technology, Teegala Krishna Reddy Engineering College, JNTUH, is submitted in partial fulfillment of the requirements for the award of the degree of Bachelor of Information Technology.

This is a record of bonafide work carried out by us and the results embodied in this project has not been reproduced or copied from any source. The results embodied in this project have not been submitted to any other university or institute for the award of any other Degree or Diploma.

**MD.RAFE RAHEEL - 17R91A1228**

**SAI CHANDRA KIRAN.R - 17R91A1238**

**SAI PREETHAM GOUD .CH - 17R91A1242**

**TEEGALA KRISHNA REDDY ENGINEERING COLLEGE**

**DEPARTMENT OF INFORMATION TECHNOLOGY**



**CERTIFICATE**

This is to certify that the project entitled “**VIDEO OBJECT FORGERY DETECTION”** that is being submitted byin partial fulfillment of the requirement for the award of the degree Bachelor of Technology in Information Technology Engineering.

The result of investigation enclosed in this report have been verified and other satisfactory. The result of embodied in this thesis have not been submitted to any other University or Institute for the award of any Degree or Diploma.

# Internal guide: External

**Dr. J. Praveen Kumar Associate Professor**

**Head of the Department Principal**

**Dr. J. Praveen Kumar Dr. K. Venkata Murali Mohan**

**Associate Professor**

**ACKNOWLEDGEMENT**

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose encouragement and guidance have crowned our efforts with success.

We are indebted to our guide, DR. J. PRAVEEN KUMAR, Head of the Department, Department of Information Technology, Teegala Krishna Reddy Engineering College, Meerpet, for his constant support and guidance throughout my project.

We are indebted to Dr. J. Praveen Kumar, Professor and Head of the Department, Information Technology, Teegala Krishna Reddy Engineering College, Meerpet, for his support and guidance throughout our project.

We extend our deep sense of gratitude to Principal Dr. K. Venkata Murali Mohan, Teegala Krishna Reddy Engineering College, Meerpet, for permitting us to undertake this project.

Finally, we express thanks to one and all who have helped us in successfully completing this project report. Furthermore, we would like to thank our family and friends for their moral support and encouragement.

**Submitted by:**

**MD.RAFE RAHEEL - 17R91A1228**

**SAI CHANDRA KIRAN.R - 17R91A1238**

**SAI PREETHAM GOUD.CH - 17R91A1242**

**CONTENTS**

**Name of the Topic Page No.**

**ABSTRACT**

1. **INTRODUCTION 1**

1.1 Motivation

1.2 Problem Definition

1.3 Objective of the Project

1. **LITERATURE SURVEY 3**

2.1 Existing System

2.2 Disadvantages of Existing System

2.3 Proposed System

2.4 Software & Hardware Requirements

2.5 Framework Overview in VIDEO FORGERY DETECTION

1. **ANALYSIS 8**

3.1 Requirements

3.1.1 Functional Requirements

3.1.2 Non-Functional Requirements

3.2 UI Requirements

3.2.1 Administrative User Interface

3.2.2 The Operational or Generic User Interface

1. **DESIGN 11 `**

4.1 Pseudo Code

4.2 Data Flow Diagram

4.3 UML Diagrams

4.4 Input Design and Output Design

1. **IMPLEMENTATION AND TESTING 22**

5.1 Method of Implementation

5.2 Modules

5.2.1 Metadata based Features

5.2.2 Error Level Analysis

5.2.3 Machine learning

5.3 Software Requirements Specification

5.4 Life Cycle Models and Methodologies

5.5 Python Packages

5.6 System testing

1. **CONCLUSION 33**

6.1 Project Conclusion

**REFERENCES:**

**ABSTRACT**

In recent years, with the development of computer multimedia technology, video forgery has become more and more common. For the video object forgery can cover up some key evidence and it is hard to identify by the experts, the forgery detection technology for this class had always been a research hotspot. However, researchers mostly pay attention to traditional methods such as image processing and classifiers and rarely combine deep learning theory to the research. This paper proposes a video intra-frame forgery forensics algorithm based on the SSIM (Structural Similarity Index), which can automatically detect video forgery frames. The algorithm first decompresses the video into a series of frames, calculates the motion residual map of each frame, and extracts the steganographic features. Then four different steganographic feature sample sets are used to construct as the training set and the test set to train and test model. The best-performing feature was selected by the comparison experiment. Finally, the forged frame was marked from the forgery video successfully. A series of experiments show that the proposed algorithm can automatically identify original or forgery frames in forgery video.

**1. INTRODUCTION**

With the development of computer multimedia technology, digital video has become the main form of network with its intuitive, convenient and informative information content. It has also become critical evidence of news, politics, insurance claims, defense, legal trial, and many other important matters. However, due to the widespread use of powerful multimedia editing tools, some non-professionals can easily modify video content, while experts are difficult to distinguish some fake videos between true and fake. These have led to doubts about the credibility of digital video content. Therefore, there is an urgent need for effective forensic technology to verify the authenticity, originality, and integrity of video content. This technology is of great practical significance for reducing the damage caused to the public by malicious tampering video and maintaining social harmony and stability. Digital video forgery mainly divided into two classes: the whole frame forgery and object forgery, whole frame forgery refers to the modification of image frames as forgery units, and the whole frame forgery methods include frame deletion, frame insertion, and frame duplication. Object forgery refers to that takes part of the video frame as forgery object and modifies the time domain and space domain of video simultaneously. The main forgery methods are intra-frame copy-move forgery, objective remove, and video synthesis forgery.

In whole frame forgery, the methods to detect forgeries based on the following: scene dependency-based, flow methods based, compression artifacts exploitation based, and deep learning-based approaches. Scene dependency-based methods use the pixel value of each frame in a video to detect abnormal periodical artifacts in the video stream. Optical Flow methods in used motion and brightness gradients as the evidence of forgery. Compression artifacts-based methods depend on different types of compression artifacts produced during the video encoding and decoding processes to capture abnormal abrupt changes in video streams. Image processing methods operate by identifying and learning suitable features from the training samples of the video streams, automatically. This kind of forgery is relatively simple and the detection results of the above algorithms are satisfactory, but the effect of this kind of forgery is not so satisfactory for it is easy to identify by human’s eyes.

**1.1 Motivation**

This project provides two level analysis for the image. At first level, it checks the image metadata. Image metadata is not that much reliable since it can be altered using simple programs. But most of the images we come across will have non-altered metadata which helps to identify the alterations. For example, if an image is edited with Adobe Photoshop, the metadata will contain even the version of the Adobe Photoshop used.

In the second level, the image is converted into error level analyzed format and will be resized to 100px x 100px image. Then these 10,000 pixels with RGB values (30,000 inputs) is given in to the input layer of Multilayer perceptron network. Output layer contain two neurons. One for fake image and one for real image. Depending upon the value of these neuron outputs along with metadata analyzer output, we determine whether the image is fake or not and how much chance is there for the given image to be tampered.

**1.2 Problem definition**

In this project, we will be solving the problem being faced by the existing system which analyzes the images as a whole, it means that the whole image would be analyzed to check whether it is a true image or a fake one, but with this project, we make use of the structural similarity index and the neural network to create a software which identifies the area of alteration first and then it compares the whole image with the help of the structural similarity index algorithm.

## 1.3 Objective of the Project

## The objective of this project is to identify fake images (Fake images are the images that are digitally altered images). We approached the problem using machine learning and neural network to detect almost all kinds of tampering on images. Moreover, we aim to make it comfortable for everyone to have access to this idea by creating web applications and android applications, so that people can easily identify the difference between real images and fake images. This software also makes sure that there are no false propagandas spreading in the social media by applying filter to the images that are creating issues. This could also be helpful in the court hearings where the guilty often produce altered witnesses.

## Nowadays, it has become easy for anyone with a smartphone and little knowledge about editing or photoshop can alter the images by using their photoshop skills. This will create huge problems if not taken care of in the early phase itself. The fake images can create a havoc in the society where if a photoshopped image goes viral in the social media. Therefore, we create a software to distinguish real images and fake images with considerable accuracy.

# 2. LITERATURE SURVEY

The object forgery is more sophisticated to manipulate than whole-frame forgeries in general. Chen et al. adopted state-of-the-art image steganalytic techniques to detect the alteration of the inherent properties inside the motion residuals of the video frames, then use a forged video detector and a two-stage automatic algorithm locate the forged video segments in the forged video accurately. Bagiw et al. proposed a method to detect and locate the forgery objects by using the statistical correlation of the Hessian matrix, but this method could not detect video with changes in the background

**Reference 1:**

Video inter‐frame forgery identification based on the consistency of quotient of MSSIM

**Author Name:** [Zhaohong Li](https://onlinelibrary.wiley.com/action/doSearch?ContribAuthorStored=Li%2C+Zhaohong) ,[Zhenzhen Zhang](https://onlinelibrary.wiley.com/action/doSearch?ContribAuthorStored=Zhang%2C+Zhenzhen) ,[Sheng Guo](https://onlinelibrary.wiley.com/action/doSearch?ContribAuthorStored=Guo%2C+Sheng)

,[Jinwei Wang](https://onlinelibrary.wiley.com/action/doSearch?ContribAuthorStored=Wang%2C+Jinwei)

**Description:** Inter‐frame forgery is a common type of video forgery in digital videos. In this paper, a method based on the consistency of quotient of mean structural similarity (QoMSSIM) is proposed. For original videos, the QoMSSIM are consistent, but in forgeries the consistency will be destroyed. First, the mean structural similarity (MSSIM) between every two adjacent frames is extracted, and then the quotients between every two sequential MSSIM are calculated. Finally, the quotient of mean SSIM after post‐processing, normalization and quantization is used as distinguishing feature to identify inter‐frame forgeries.

**Reference2:** Exposing video inter-frame forgery by Zernike opponent chromaticity moments and coarseness analysis,

**Author Names:** [Yuqing Liu](https://link.springer.com/article/10.1007/s00530-015-0478-1#auth-Yuqing-Liu) & [Tianqiang Huang](https://link.springer.com/article/10.1007/s00530-015-0478-1" \l "auth-Tianqiang-Huang)

**Description:** Inter-frame forgery is the most common type of video forgery methods. However, few algorithms have been suggested for detecting this type of forgery, and the former detection methods cannot ensure the detection speed and accuracy at the same time. In this paper, we put forward a novel video forgery detection algorithm for detecting an inter-frame forgery based on Zernike opponent chromaticity moments and a coarseness feature analysis by matching from the coarse-to-fine models.

## 2.1 Existing System

The above object detection algorithms based on the traditional methods of image processing and classifier did not involve deep learning approach. The reason is that there too many objects within the video frame to apply the deep learning network directly on the characteristics of the study; therefore, the traditional methods are rarely combined with deep learning approach for the video object forgery detection.

**2.2 Disadvantages of Existing System**

* The existing system consumes more time by examining the whole image rather than identifying the forged part. This will affect the time complexity and it would highly impossible to quickly differentiate between real and fake images.
* The lesser available options also make it very inconvenient to examine all sorts of images, which could limit the way the project works.
* Therefore, we make necessary adjustments to this and then propose a new system with machine learning techniques along with the SSIM (Structural Similarity Index) algorithm.

**2.3 Proposed System**

In this paper, a model of video object forgery detection based on the SSIM. It is proposed to identify forged frames automatically. The algorithm deconstructs a video into a sequence frame firstly. Then the cryptic feature extraction algorithm is used to calculate the residual feature map of each frame. Four steganalytic feature sample sets after pre-processing (i.e., naturalization) are used for processing. Finally, the forged frames are marked out from the forgery video.

**2.4 SOFTWARE & HARDWARE REQUIREMENTS**

**Hardware Requirements:**

 Processor : Intel i3 and above

 RAM : 4GB and Higher

 Hard Disk : 500GB: Minimum

**Software Requirements:**

 Programming Language/Platform : Python

 IDE : PyCharm / Anaconda Navigator

**2.5 FRAMEWORK OVERVIEW IN VIDEO FORGERY DETECTION:**

Studies [39-46] were dedicated to digital image forensics but only a few have touched upon digital video forgery detections. One of the most popular tampering artifacts in video forgery is copy-move forgery. In this domain, it is challenging to detect regions or frames as the forged location may differ with regards to size and rate of compression. Video forgery detection methods are primarily utilized to determine the spatial domain and temporal domain of copy-move tampering. In Figure 2, the general detection method consisting of extract frames from the source video, feature extraction, overlapping block matching, and forgery decision are presented.

This method enables the application of many extraction techniques like the DCT, DWT, PCA, among others and allows the application of various matching methods [43] like K-SVD tree and radix sort. In editing a video sequence, the processing methods consists of three steps; first, the input sequence of frames are decoded; second, the actual frames sequence is edited and; third, the edited

In other studies, such as [39], Xiaoping brought forward a method that authenticates and detects tampered algorithm combined with semi-fragile watermark embedded into DCT coefficient with the help of Compressing Sensing Theory. He utilized MPEG-2 compression video as the research object, where content authentication of inner I-frames and tamper detection of P-frame can be carried out. The result showed that the algorithm Semi-fragile Watermarking algorithm obtained top effectiveness when it comes to ability and accuracy. In a related study Wang et al. [40] developed a method involving the use of the temporal and spatial correlation to determine frames duplication but the location of frame duplication is inaccurate in case of small forged regions. Similarly, [41] created a method according to two types of attacks; 1) spatial (pixel) copy-move attack detected via Histogram of Oriented Gradients (HOG), 2) temporal copy-move attack detected via exploitation of MPEG0-2 GOP structure. Also, Wang & Farid [42] proposed a video tampering detection method through the detection of duplicate frames. In such a method, a doubly compressed MPEG video frames sequence provides specific static and temporal statistical disarrangement whose existence can be used like an originally encoded MPEG compression method where frames are edited and re-saved as a doubly compressed MPEG video. ls to delete some moving frames objects in a video sequence and referred to it as one of the common methods of video forgery of frames. The differences of features between a video of frames were obtained with the help of Compressed Sensing, K-SVD (k-Singular Value Decomposition) and random projection was utilized to relay the features into the lower-dimensional subspace that is clustered by k-means. The detection results are eventually combined for each frame. Hsu et al. [44] brought forward video splicing method and demonstrated a technical method to detect forged frame regions in a video with the help of correlation of noise residue. The method primarily hinges on the notion that the tampered frames transform the correlation of noise residue on each frame and differentiates them from the nontampered parts. The results of the experiments reveal that the noise correlation is fairly dependable feature in case of fine-quality video although it is vulnerable to noise quantization. Added to this, the noise residue extraction is a complex process [44] – spatial (intra-frame) forgery and temporal (interframe) forgery. In the former, the tamper-free form, the same videos are utilized for clipping, and the inter-frame frames from the video are utilized for tampering. In a related study [46], a method according to the Tamura texture features and algorithm was proposed with the help of the vector matrix of the video through video frame extraction. The method calculates the differences between the Tamura texture feature vector and the adjacent vector matrix. In case the differences are lower than the threshold, their distance is contrasted for the serial number with the threshold and the pairs of the serial numbers bigger than the distance threshold is recorded to locate the copy-move sequences.

**RELATED WORK**

To the best of the researcher’s knowledge, passive approaches are the most important methods in the detection of digital video forgery [59-68]. Table 2 displays the summarized video forgery detection methods, under the headings of classifier, frame dimension with data set, and prior work, and remake of the evaluation performance.

The detection of blurring can be manipulated via the statistical characteristics of object-based forgery operations. In relation to this, Richao et al. [59] conducted an analysis of the concept of AWOB using statistical features as wavelet coefficients and the moment features that details the average gradient of every color channel were taken to include in the SVM. According to the experimental findings, the accuracy of detection is around 95% and the data set consisted of 20 videos from SULFA.

Moreover, Wang et al. [60] brought forward a method on the basis of the assumption that the correlation coefficients of gray values lying between the sequences of video following normalization and quantization to determine interframe forgeries involving small data set (five videos).

The accuracy was found to be 98.79%. Similarly, Su [61] proposed a method that detected tampering on the basis of compressive sensing with the help of feature clustering of the differences between frames obtained via K-SVD. The results showed an accuracy of 89.6%. Also, in [63] the method analyses impacts the attacks in temporal domain through machine learning methods. Meanwhile, Bestagini et al. [63] conducted analysis of the footprints in terms of video sequence through a detection algorithm that enables a forensic analyst to determine video forgeries and localize them in the domain of spatio-temporal They tested the analysis on 120 actual frame sequences with the resolution of 320X240 pixels comprising 20 videos.

The results showed an analysis accuracy of 87%. Moreover, Vazquez-Padin et al. [64] brought forward a technique that estimates the GOP size with a video sequence based on the assumption that VPF becomes evident in P-frames that are intracoded in the first double encoding. The experiment involved 14 video sequences allowing an accuracy of 95%. In relation to this, [65] provided a description of a method that determines video object contour on the basis of non-sub sampled contourlet transform and gradient information that employed feature vector combined with SVM.

Their dataset comprised 9 videos with the frame 320X240 and the accuracy was found to be 95%. However, this method is not very effective in detecting forgery areas in static scene videos. As a result, it is not appropriate for the detection of suspicious level areas in videos taken by a moving camera.

In relation to the above studies, Chetty et al. [66] suggested a method of video tampering detection based on transformation of feature from several intra-frame and inter-frame pixel sub-blocks in video sequences and their multi-modal combination.

The emulated copy-move tamper scene revealed that the quantization residue features performance for the entire experiments is similar to noise residue features. But the method is framelevel forgery focuses and thus it did not locate the issue of region-level tampering and localization. Furthermore, [67] brought forward a new approach to detect motion-compensated edge artifact to determine the changes of correlation among adjacent frames. Also, [68] proposed duplicate frames or frame parts to delete people or objects from the video-call in painting. Their method only worked in frame manipulation detection and not in localization of tampered object regions.

Libraries:

**SK IMAGE:**

skimage.feature.corner\_fast (image[, n, … ]) Extract FAST corners for a given image. skimage.feature.corner\_foerstner (image[, sigma]) Compute Foerstner corner measure response image.

**IMUTILS**

Imutils are a series of convenience functions to make basic image processing functions such as translation, rotation, resizing, skeletonization, and displaying Matplotlib images easier with OpenCV and both Python 2.7 and Python 3.

**ARGPATH**

The argparse module makes it easy to write user-friendly command-line interfaces. It parses the defined arguments from the sys. argv . The argparse module also automatically generates help and usage messages, and issues errors when users give the program invalid arguments.

**OPENCV:**

OpenCV-Python is a library of Python bindings designed to solve computer vision problems. ... All the OpenCV array structures are converted to and from Numpy arrays. This also makes it easier to integrate with other libraries that use Numpy such as SciPy and Matplotlib.

**3. ANALYSIS**

Although the scale of this project is relatively small, to produce a professional solution is it imperative that the current problem is understood accurately. However, this task has been made doubly difficult by the lack of support from the company. Thankfully, the Application manager has been kind enough to spare me some of his own time to discuss the problem with me further. Therefore, this chapter is concerning with analyzing the current situation and expectations of the user for this system.

**Requirements:**

The minimum requirements of the project are listed below:

* Examine the tools and methodologies required to gain an overview of the system requirements for the proposed database.
* Examine suitable database management systems that can be used to implement the proposed database.
* Evaluate appropriate website authoring and web graphic creation tools that can be used to develop web-based forms for the proposed database
* Produce and apply suitable criteria for evaluating the solution

**Requirement Analysis:**

Taking into account the comparative analysis stated in the previous section we could start specifying the requirements that our website should achieve. As a basis, an article on all the different requirements for software development was taken into account during this process. We divide the requirements in 2 types: functional and nonfunctional requirements.

**Functional requirements:**

Functional requirement should include function performed by a specific screen outline work-flows performed by the system and other business or compliance requirement the system must meet.

Functional requirements specify which output file should be produced from the given file they describe the relationship between the input and output of the system, for each functional requirement a detailed description of all data inputs and their source and the range of valid inputs must be specified.

The functional specification describes what the system must do, how the system does it is described in the design specification.

If a user requirement specification was written, all requirements outlined in the user requirements specifications should be addressed in the functional requirements.

* Original Video
* QoMssim Algorithm
* Post Processing Normalization and Quantization
* Calculate Histogram
* Video Forgery
* Data Set Preparation
* Data Transformation
* Data visualization
* Creating Training, Testing, Validation Set
* Generating Model
* Performance Evaluation
* Test Prediction
* Result Evaluation

**Nonfunctional requirement:**

Describe user-visible aspects of the system that are not directly related with the functional behavior of the system. Non-Functional requirements include quantitative constraints, such as response time (i.e. how fast the system reacts to user commands.) or accuracy (.e. how precise are the systems numerical answers.).

* Portability
* Reliability
* Usability
* Time Constraints
* Error messages
* Actions which cannot be undone should ask for confirmation
* Responsive design should be implemented
* Space Constraints
* Performance
* Standards
* Ethics
* Interoperability
* Security
* Privacy
* Scalability

**UI Requirements**

1. **Administrative user interface**

The ‘administrative user interface’ concentrates on the consistent information that is practically, part of the organizational activities and which needs proper authentication for the data collection. These interfaces help the administrators with all the transactional states like Data insertion, Data deletion and Date updating along with the extensive data search capabilities.

1. **The operational or generic user interface**

The ‘operational or generic user interface’ helps the end users of the system in transactions through the existing data and required services. The operational user interface also helps the ordinary users in managing their own information in a customized manner as per the included Flexibilities.

**4. Design**

**4.1 Pseudo Code**

**# USAGE**

**# python video\_diff.py 4o.avi 4f.avi**

**# import the necessary packages**

**from skimage.measure import compare\_ssim**

**import imutils**

**import cv2**

**from sys import argv**

**print("videos\\"+str(argv[1]))**

**print("videos\\"+str(argv[2]))**

**vs1 = cv2.VideoCapture("videos\\"+str(argv[1]))**

**vs2 = cv2.VideoCapture("videos\\"+str(argv[2]))**

**# loop over frames from the video file stream**

**while True:**

**# read the next frame from the file**

**(grabbed1, imageA) = vs1.read()**

**(grabbed2, imageB) = vs2.read()**

**# if the frame was not grabbed, then we have reached the end**

**# of the stream**

**if not grabbed1 or not grabbed2:**

**break**

**# convert the images to grayscale**

**grayA = cv2.cvtColor(imageA, cv2.COLOR\_BGR2GRAY)**

**grayB = cv2.cvtColor(imageB, cv2.COLOR\_BGR2GRAY)**

**# compute the Structural Similarity Index (SSIM) between the two**

**# images, ensuring that the difference image is returned**

**(score, diff) = compare\_ssim(grayA, grayB, full=True)**

**diff = (diff \* 255).astype("uint8")**

**print("SSIM: {}".format(score))**

**# threshold the difference image, followed by finding contours to**

**# obtain the regions of the two input images that differ**

**thresh = cv2.threshold(diff, 0, 255,**

**cv2.THRESH\_BINARY\_INV | cv2.THRESH\_OTSU)[1]**

**cnts = cv2.findContours(thresh.copy(), cv2.RETR\_EXTERNAL,**

**cv2.CHAIN\_APPROX\_SIMPLE)**

**cnts = imutils.grab\_contours(cnts)**

**# loop over the contours**

**for c in cnts:**

**# compute the bounding box of the contour and then draw the**

**# bounding box on both input images to represent where the two**

**# images differ**

**(x, y, w, h) = cv2.boundingRect(c)**

**cv2.rectangle(imageA, (x, y), (x + w, y + h), (0, 0, 255), 2)**

**cv2.rectangle(imageB, (x, y), (x + w, y + h), (0, 0, 255), 2)**

**# show the output images**

**cv2.imshow("Original", imageA)**

**cv2.imshow("Modified", imageB)**

**#cv2.imshow("Diff", diff)**

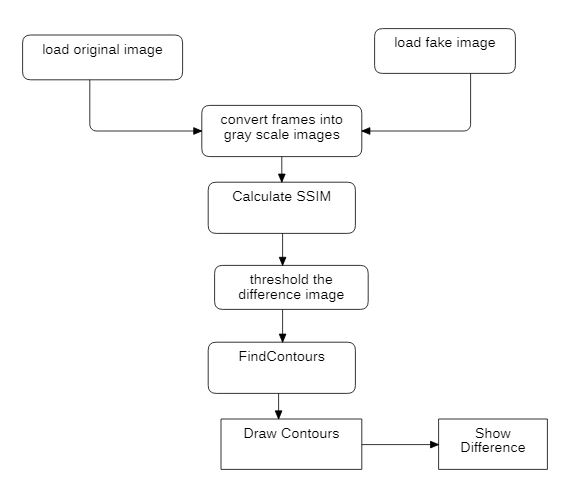
**#cv2.imshow("Thresh", thresh)**

**if cv2.waitKey(1) & 0xFF == ord('q'):**

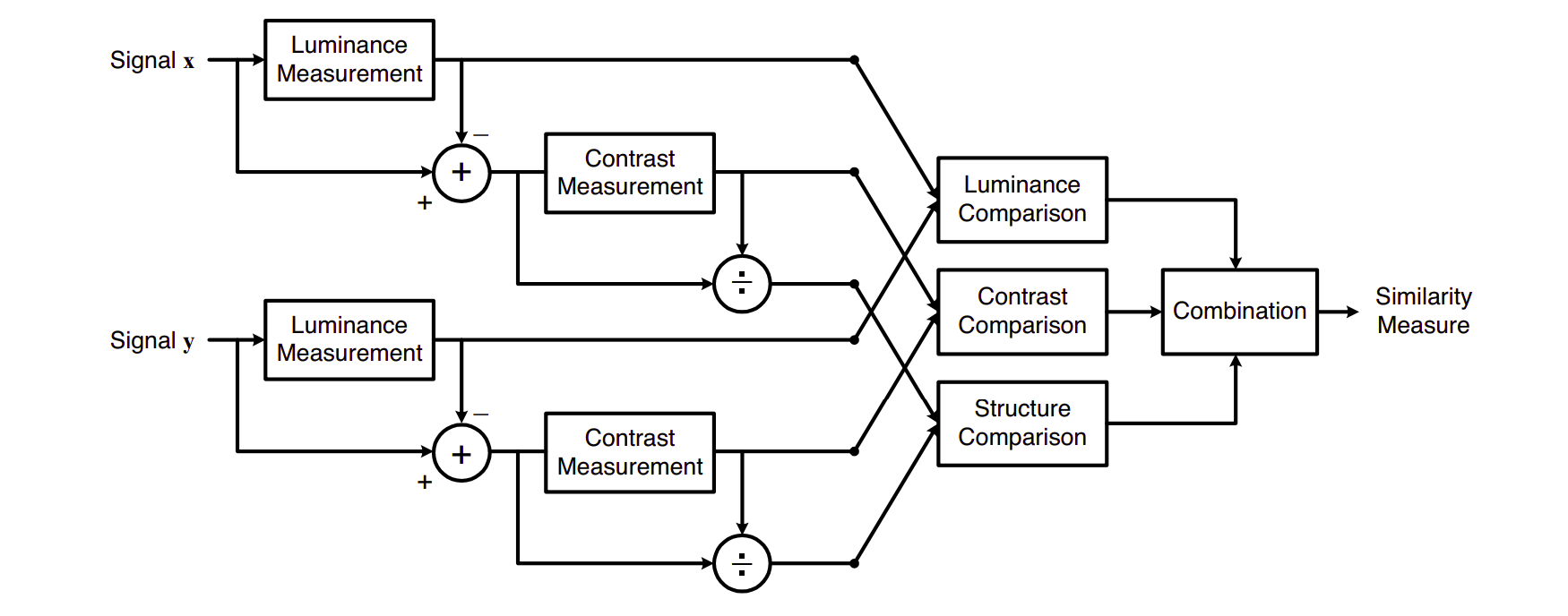
**break**

**4.2 Data Flow Diagram:**

Next step is to bring down whole knowledge of requirements and analysis on the desk and design the software product. The inputs from users and information gathered in requirement gathering phase are the inputs of this step. The output of this step comes in the form of two designs; logical design and physical design. Engineers produce meta-data and data dictionaries, logical diagrams, data-flow diagrams and in some cases pseudo codes.



**SSIM Architecture:**

****

**4.3 UML Design:**

Unified Modeling Language (UML) is a general-purpose modeling language. The main aim of UML is to define a standard way to visualize the way a system has been designed. It is quite similar to blueprints used in other fields of engineering.

UML is not a programming language; it is rather a visual language. We use UML diagrams to portray the behavior and structure of a system, UML helps software engineers, businessmen and system architects with modeling, design and analysis. The Object Management Group (OMG) adopted Unified Modeling Language as a standard in 1997. It’s been managed by OMG ever since. International Organization for Standardization (ISO) published UML as an approved standard in 2005. UML has been revised over the years and is reviewed periodically.

**Do we really need UML?**

* Complex applications need collaboration and planning from multiple teams and hence require a clear and concise way to communicate amongst them.
* Businessmen do not understand code. So, UML becomes essential to communicate with non-programmer’s essential requirements, functionalities and processes of the system.
* A lot of time is saved down the line when teams are able to visualize processes, user interactions and static structure of the system.
* UML is linked with object-oriented design and analysis. UML makes the use of elements and forms associations between them to form diagrams. Diagrams in UML can be broadly classified as:

**The Primary goals in the design of the UML are as follows:**

* Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
* Provide extendibility and specialization mechanisms to extend the core concepts.
* Be independent of particular programming languages and development process.
* Provide a formal basis for understanding the modeling language.
* Encourage the growth of OO tools market.
* Support higher level development concepts such as collaborations, frameworks, patterns and components.
* Integrate best practices.

**Types of UML Diagrams:**

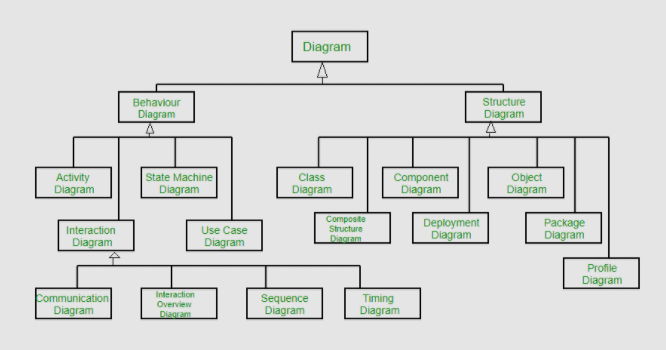
**Structural Diagrams:**

Capture static aspects or structure of a system. Structural Diagrams include: Component Diagrams, Object Diagrams, Class Diagrams and Deployment Diagrams.

**Behavior Diagrams:**

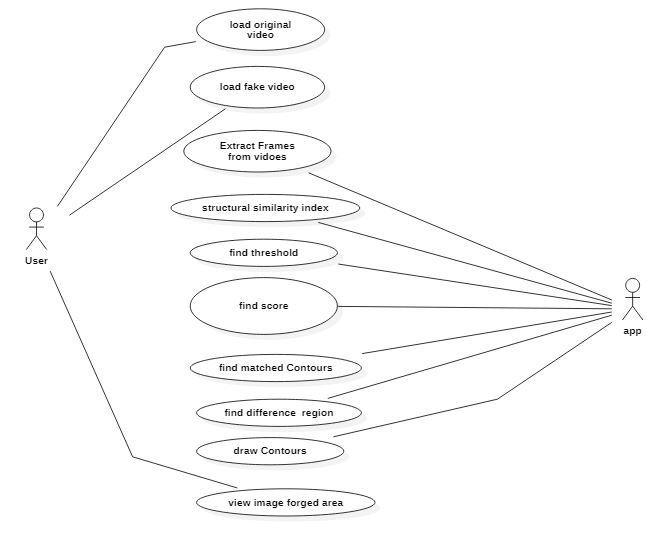
Capture dynamic aspects or behavior of the system. Behavior diagrams include: Use Case Diagrams, State Diagrams, Activity Diagrams and Interaction Diagrams.

**The image below shows the hierarchy of diagrams according to UML**

****

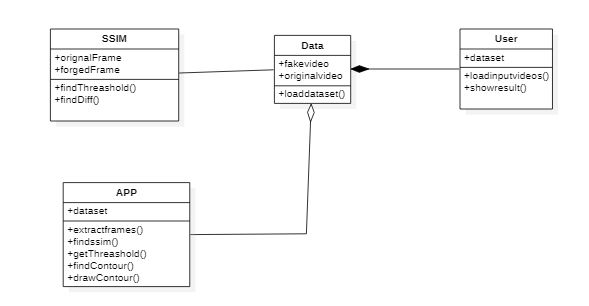
**USE CASE DIAGRAM:**

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.



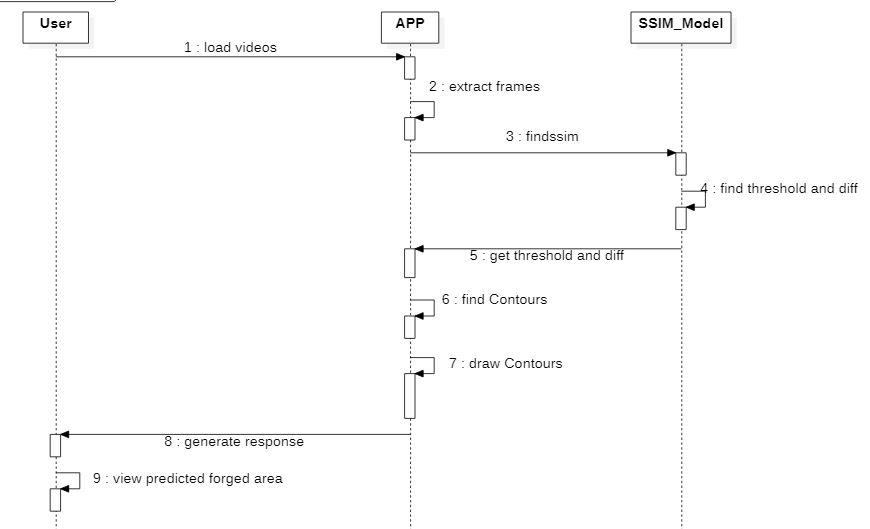
**CLASS DIAGRAM:**

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.



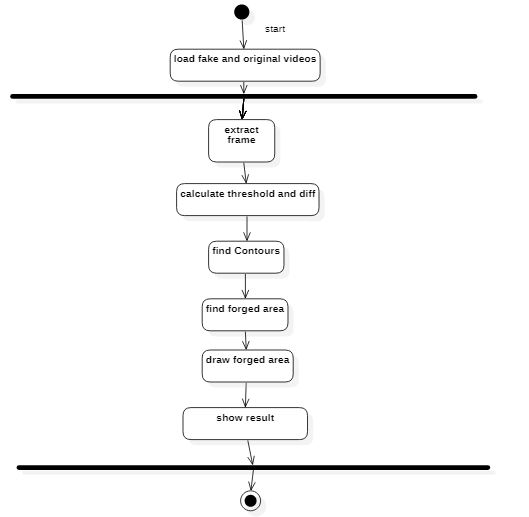
**SEQUENCE DIAGRAM:**

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.



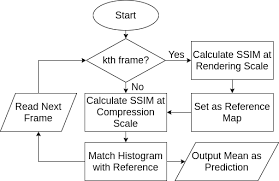
**ACTIVITY DIAGRAM:**

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency.



**Deployment diagram:**

There may be more steps involved, depending on what specific requirements you have, but below are some of the main steps:



**4.4 INPUT DESIGN AND OUTPUT DESIGN**

**INPUT DESIGN**

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

* What data should be given as input?
* How the data should be arranged or coded?
* The dialog to guide the operating personnel in providing input.
* Methods for preparing input validations and steps to follow when error occur.

**OBJECTIVES**

1. Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

2. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

3. When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow

**OUTPUT DESIGN**

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system’s relationship to help user decision-making.

1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.

2. Select methods for presenting information.

3. Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

* Convey information about past activities, current status or projections of the
* Future.
* Signal important events, opportunities, problems, or warnings.
* Trigger an action.
* Confirm an action.

**SYSTEM STUDY**

**FEASIBILITY STUDY**

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

* ECONOMICAL FEASIBILITY
* TECHNICAL FEASIBILITY
* SOCIAL FEASIBILITY

**ECONOMICAL FEASIBILITY**

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

**TECHNICAL FEASIBILITY**

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

**SOCIAL FEASIBILITY**

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

**5. Implementation & Testing**

**5.1 Method of Implementation**

**Implementation of Regression Algorithms**

Logistic Regression (LR): One algorithm used for paraphrase identification was Logistic Regression which uses a logistic function to aid in machine learning. This algorithm is most helpful for binary outputs and categorization. It is used to identify a clear comparison, or relationship, between two or more variables. Below given is a general cost function of logistic regression.

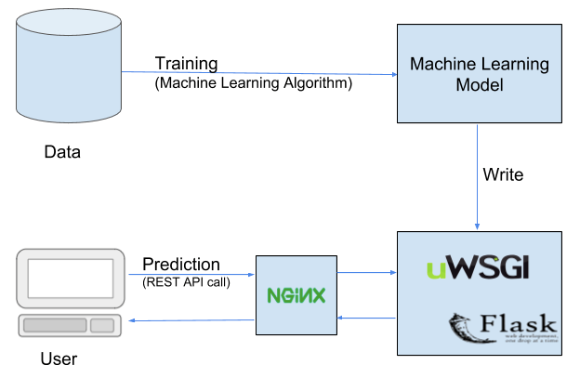
**Implementation Classification Algorithm**

Support Vector Machine (SVM): SVM is a non-probabilistic classifier of data. In other words, when a given set of data is fed through the algorithm, SVM outputs the optimum separation to categorize the information. It takes in the input of the question vector and creates a discrete classification model of the data that outputs the predicted value. In SVM, there are also multiple tuning parameters such as Kernel. Furthermore, there are various functions of Kernel, including linear, polynomial, and exponential.

**Neural Network Implementation**

Neural Networks (NN): Another algorithm that was used is Neural Networks (NN). Theoretically, Neural Networks are a universal function approximators. By the composition of various non-linear activations, they are able to represent any data distribution. Our neural network model comprises of a simple 3 layer deep model which takes the word2vec embedding as an input, a Siamese network taking two vectors representing two sentences as an input and a Long Short Term Memory (LSTM) network

Long Short-Term Memory (LSTM): Recurrent Neural Network are the class of neural networks which takes the feedback from previous time step into consideration while making the prediction for next time step. In sequence processing task like ours, RNN can be effective. Hence, we develop a model based on LSTM, which is a variant of recurrent neural network.



**5.2 Modules**

**5.2.1 Metadata based Features:**

Metadata provides information related to how the file was generated and handled. This information can be used to identify if the metadata appears to be from a digital camera, processed by a graphical program, or altered to convey misleading information. Common things to look for include:

* Image size
* Timestamps
* Types of metadata
* Descriptions
* Missing metadata
* Altered Metadata

**5.2.2 Error Level Analysis**

JPEG is a loss format, but the amount of error introduced by each resave is not line. Any modification to the picture will alter the image such that stable areas (no additional error) become unstable. The modified picture was based on the first 75%resave. Books on the shelf were duplicated and a toy dinosaur was added to the shelf. The 95% ELA identifies the changes since they are areas that are no longer at their minimal error level. Additional areas of the picture show slightly more volatility because Photoshop merged information from multiple layers, effectively modifying many of the pixels.

**5.2.3 Machine Learning**

The process of machine learning is similar to that of data mining. Both systems search through data to look for patterns. However, instead of extracting data for human comprehension as is the case in data mining applications machine learning uses that data to detect patterns in data and adjust program actions accordingly. Machine learning algorithms are often categorized as being supervised or unsupervised. Supervised algorithms can apply what has been learned in the past to new data. Unsupervised algorithms can draw inferences from datasets.

**5.3** **SOFTWARE REQUIREMENT SPECIFICATION**

Software Requirements Specification (SRS) is the starting point of the software developing activity. As system grew more complex it became evident that the goal of the entire system cannot be easily comprehended. Hence the need for the requirement phase arose. The software project is initiated by the client needs. The SRS is the means of translating the ideas of the minds of clients (the input) into a formal document (the output of the requirement phase).

The SRS phase consists of two basic activities:

**Problem/Requirement Analysis:**

The process is order and more nebulous of the two, deals with understand the problem, the goal and constraints.

**Requirement Specification:**

Here, the focus is on specifying what has been found giving analysis such as representation, Specification languages and tools, and checking the specifications are addressed during this activity.

The requirement phase terminates with the production of the validate SRS document. Producing the SRS document is the basic of this phase.

**Role of SRS:**

The purpose of the SRS is to reduce the communication gap between the clients and the developers. SRS is the medium though which the client and user needs are accurately specified.

It forms the basis of software development. A good SRS should satisfy all the parties involved in the system.

**Purpose:**

The purpose of this document is to describe all external requirements for the E-learning System. It also describes the interfaces for the system.

 **Scope:**

This document is the only one that describes the requirements of the system. It is meant for the use by the developers, and will also by the basis for validating the final deliver system. Any changes made to the requirements in the future will have to go through a formal change approval process. The developer is responsible for asking for clarifications, where necessary, and will not make any alternations without the permission of the client.

 **Overview:**

The SRS begins the translation process that converts the software Requirements into the language the developers will use. The SRS draws on the Use Cases from the user Requirement Document and analyses the situations from a number of perspectives to discover and eliminate inconsistencies, ambiguities and omissions before development progresses significantly under mistaken assumptions.

**Proposed System Architecture:**

The proposed system is built around conventional three-tier architecture. The three-tier architecture for web development allows programmers to separate various aspects of the solution design into modules and work on them separately. That is, a developer who is best at one part of development, say UI development need not worry about the implementation levels so much. It also allows for easy maintenance and future enhancements. The three-tiers of the solution include:

 **The Layout:**

This tier is at the uppermost layer and is closely bound to the user, i.e., the users of the system interact with it through this tier.

 **The business-tier:**

This tier is responsible for implementing all the business rules of the organization. It operates on the data provided by the users through the web-tier and the data stored in the underlying data-tier. So in a way this tier works on data from the web-tier and the data-tier in order to perform task for the users in agreement with the business rules of the organization.

 **The data-tier:**

This tier contains the persist able data that is required by the business tier to operate on. Data plays a very important role in the functioning of any organization. Thus, persisting of such data is very important. The data tier performs the job of persisting the data.

**5.4 Life Cycle Models and Methodologies:**

**Waterfall Model**

**Description:**

The Waterfall Model is a linear sequential flow. In which progress is seen as flowing steadily downwards (like a waterfall) through the phases of software implementation. This means that any phase in the development process begins only if the previous phase is complete. The waterfall approach does not define the process to go back to the previous phase to handle changes in requirement. The waterfall approach is the earliest approach and most widely known that was used for software development.

**The Usage:**

Projects which not focus on changing the requirements, for example, projects initiated from request for proposals (RFPs), the customer has a very clear documented requirement

**Advantages and Disadvantages:**

Advantages:

• Easy to explain to the users.

• Structure’s approach.

• Stages and activities are well defined.

• Helps to plan and schedule the project.

• Verification at each stage ensures early detection of errors/misunderstanding.

• Each phase has specific deliverables.

**Disadvantages:**

• Assumes that the requirements of a system can be frozen.

• Very difficult to go back to any stage after it finished.

• A little flexibility and adjusting scope is difficult and expensive.

• Costly and required more time, in addition to the detailed plan.

**Software Environment**

**Executive Summary**

Python is an interpreter, object-oriented, high-level programming language with dynamic semantics. Its high-level built-in data structures, combined with dynamic typing and dynamic binding; make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed. Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast.

Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.

**5.5 Python Packages**

Packages or additional libraries help in scientific computing and computational modelling. In Python, the packages are not the part of the Python standard library. Few major packages are –

• NumPy (Numeric Python): matrices and linear algebra

• SciPy (Scientific Python): many numerical routines

• matplotlib: (PLOTting LIBrary) creating plots of data

• SimPy (Symbolic Python): symbolic computation

• pytest (Python TESTing): a code testing framework

**What is Anaconda Python?**

Together with a list of Python packages, tools like editors, Python distributions include the Python interpreter. Anaconda is one of several Python distributions. Anaconda is a new distribution of the Python and R data science package. It was formerly known as Continuum Analytics. Anaconda has more than 100 new packages.

This work environment, Anaconda is used for scientific computing, data science, statistical analysis, and machine learning. The latest version of Anaconda 5.0.1 is released in October 2017.

The released version 5.0.1 addresses some minor bugs and adds useful features, such as updated R language support. All of these features weren’t available in the original 5.0.0 release.

This package manager is also an environment manager, a Python distribution, and a collection of open source packages and contains more than 1000 R and Python Data Science Packages.

**Why Anaconda for Python?**

There’s no big reason to switch to Anaconda if you are completely happy with you regular python. But some people like data scientists who are not full-time developers, find anaconda much useful as it simplifies a lot of common problems a beginner runs into.

Anaconda can help with –

• Installing Python on multiple platforms

• Separating out different environments

• Dealing with not having correct privileges and

• Getting up and running with specific packages and libraries

**How to Download Anaconda 5.0.1?**

The free version of Anaconda distribution community edition can be downloaded directly from Anaconda’s website. For the enterprise edition, one need professional support from Anaconda’s sales team.

Conda treats Python the same as any other package, so it is easy to manage and update multiple installations.

Anaconda supports Python 2.7, 3.4, 3.5 and 3.6. The default is Python 2.7 or 3.6, depending on which installer you used:

• For the installers “Anaconda” and “Miniconda,” the default is 2.7.

• For the installers “Anaconda 3” or “Miniconda 3,” the default is 3.6.

**How to Install Anaconda?**

Once downloaded the .exe file, run through the installer. Do accept the terms, and finish the installation. To check, close the browser and pull up the terminal.

Once the installation is complete, it should have automatically added that to the path. To test this, go ahead and type ‘python’.

The version of python i.e., 3 will show and also the anaconda distribution will be seen. If you install the 4 versions of Anaconda, then all the packages, which are there in the packages, can also be imported easily.

To check type import NumPy or import matplotlib and run that.

**5.6 SYSTEM TESTING**

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the

Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

**TYPES OF TESTS**

**Unit testing**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

**Integration testing**

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

**Functional test**

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

**System Test**

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

**White Box Testing**

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

**Black Box Testing**

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

**Unit Testing:**

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

• All field entries must work properly.

• Pages must be activated from the identified link.

• The entry screen, messages and responses must not be delayed.

Features to be tested

• Verify that the entries are of the correct format

• No duplicate entries should be allowed

• All links should take the user to the correct page.

Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

**DISCUSSION**

Among the fastest growing area of research in the field of video forgery detection is the passive blind methods and detection methods to verify the integrity and authenticity of digital video sequence. To this end, current studies dedicated to passive blind methods are not in need of prior knowledge of the video frames content or pre-embedded watermarks or signature. In this study, the issue of digital video manipulating detection is discussed with references to blind methods of video forgery detection. Various frames of video forgery detection methods are categorized and generalized in this paper and the rendering of some typical video forgery detection algorithms methods are compared. Some of the developed approaches for the detection and the determination of video manipulation are capable of localizing tampered object locations of frames sequence.

This study’s findings are expected to contribute to methods and ideas in the field of digital video forgery detection. At the onset, the drawback of existing methods is related to issues of automation like human interpretation of poor outputs. Another is the modification and extension to determine the accurate location of the video forgery that involves methods that insert/remove frames and objects to determine the region of inconsistencies. Accordingly, the first step identifies that the camera source identification is still confined to 9 to 15 cameras and in mobile camera identification and as such, the result can be negatively affected by the increasing number of cameras. It is not applicable for the detection of suspicious level regions in videos taken by a moving camera. Moreover, the camera source identification methods is noted to be dependent on intrinsic camera hardware features like lens and CCD sensor characteristics that generate valid outcomes compared to those based on the software parts of the camera (e.g. CFA interpolation algorithms). Further, the video double compression artifacts add difficulty to localization of the forgery especially when the video being analyzed is compressed by a low-quality factor in most methods.

Second, the image-splicing forgery detection in its accurateness is depleted after processing operations, which could lead to edge blurring, loss of compression and added noise although confined to the detection methods that can be expanded to image and audio. Comprehending the perception of visual semantics is significant in the identification of the extent of forgery. Lastly, copy-move forgery detection are computationally expensive and they bring about high false positives, and use high correlation between original and forged parts of the video frames in order to detect and determine copy paste forgery. However, high correlation between frames is commonplace in natural videos, and the method is not appropriate if copied regions are obtained from other views. On the other hand, copy-move forgery localization methods that are based on frames are appropriate with frame detection duplication and not the localization of forged region in case the video content is consistent and the prior modified region had lower quality frames than the current frame. In the context of pixel-based approaches, the manipulation of detection accuracy impacts postprocessing and compression and thus making the validation of performance measures (i.e. accuracy, robustness, security) becomes a major concern owing to the absence of established benchmarks and public testing dataset that evaluates the actual accuracy of digital video forgery approaches. Among the significant limitation of video forgery detection methods is their inability to distinguish between malicious manipulation and innocent retouching, like red-eye correction. Future studies are encouraged to determine a more robust statistical feature that are resistant to several postprocessing operations.

**6. Conclusion**

**6.1 Project Conclusion**

Neural network has been successfully trained using the error level analysis with 4000 fake and 4000 real images. The trained neural network was able to recognize the image as fake or real at a maximum success rate of 83%. The use of this application in mobile platforms will greatly reduce the spreading of fake images through social media. This project can also be used as a false proof technique in digital authentication, court evidence evaluation etc. By combining the results of metadata analysis (40%) and neural network output (60%) a reliable fake image detection program is developed and tested.

**REFERENCES**

[1] A picture’s worth, Digital Image Analysis and Forensics, N Krawetz - 2007 Ph D, Hacker Factor Solutions

[2] http://imagej.net/Welcome ImageJ is an open-source image processing program designed for scientific multidimensional images.J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[3] http://forensics.idealtest.org/ CASIA v2.0 CASIA V2.0 is with larger size and with more realistic and challenged fake images by using post-processing of tampered regions. It contains 7491 authentic and 5123 tampered color images.

[4] http://neuroph.sourceforge.net/ Neuroph Framework Neuroph is lightweight Java neural network framework to develop common neural network architectures. It contains well designed, open-source Java library with small number of basic classes which correspond to basic NN concepts.

[5] https://github.com/drewnoakes/metadata-extractor Metadata-extractor is a straightforward Java library for reading metadata from image files.

[6] https://www.github.com/afsalashyana/FakeImageDetection GitHub repositor for fake image detector desktop application written in java fx.