M.Tech (Computer Science & Engineering)

Piyush Priy

2023

IMPERSONATION DETECTION IN DATA CAPTURING SYSTEMS (DCS) WITH IMAGE DATA ANALYSIS USING MACHINE LEARNING

A DISSERTATION

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE AWARD OF THE DEGREE

OF

MASTER OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted By:

PIYUSH PRIY

2K19/CSE/501

Under the supervision of

DR. SHAILENDER KUMAR

(Professor)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

May, 2023

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

CANDIDATE'S DECLARATION

I, Piyush Priy, Roll No. 2K19/CSE/501 student of M. Tech (Computer Science and Engineering), hereby declare that the Project Dissertation titled "**Impersonation Detection in Data Capturing Systems (DCS) with Image Data Analysis using Machine Learning**." which is being submitted by me to the Department of Computer Science & Engineering, Delhi Technological University, Delhi, in partial fulfilment of requirements for the award of the degree of Master of Technology, is original and not copied from any source without proper citation. This work has not previously formed the basis for the award of any Degree, Diploma Associateship, Fellowship or other similar title or recognition.

Place: Delhi **PIYUSH PRIY**

Date: (2K19/CSE/501)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

CERTIFICATE

I hereby certify that the dissertation titled "**Impersonation Detection in Data Capturing Systems (DCS) with Image Data Analysis using Machine Learning.",** which is submitted by PIYUSH PRIY, Roll No. 2K19/CSE/501, Department of Computer Science & Engineering, Delhi Technological University, Delhi in partial fulfilment of the requirement for the award of the degree of Master of Technology, is a record of the project work carried out by the student under my supervision. To the best of my knowledge, this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

Place: Delhi

Date:

DR. SHAILENDER KUMAR

(Professor)

SUPERVISOR

ABSTRACT

Impersonation attacks pose a significant threat to the security and integrity of data-capturing systems (DCS). With the increasing reliance on image data in various domains, it becomes crucial to develop effective methods for detecting impersonation attacks using image data analysis. This research aims to address the problem of impersonation detection in DCS by leveraging machine learning techniques. Specifically, this thesis proposes a novel approach that combines image processing, feature extraction, and machine learning algorithms to identify and detect impersonation attacks in DCS.

Many of today's applications require cooperating with multiple database systems. The heterogeneous data representations of databases include schemas, actual data etc. The data inconsistencies in databases may occur for semantic inconsistencies stored in syntactical data; the data is distinguishable but semantically identical or for the same record representation.

Semantic heterogeneity in database systems has a traditional problem that always results in data duplication i.e., the same record is accumulated two or more times in multiple database systems. The problem of Data Duplication is that its highly pervasive in legacy software systems.

Data duplication refers to a data source having more than one record for the same identity, mostly with different syntaxes for the same item/identity. This problem has been known as highly important to many entities due to the size and complexity of today's database systems.

Several researchers tried to resolve the data duplicity problem by using different techniques such as the Sorted Neighbors Method (SNM), identity matrix, and many clustering methods. However, an Image classification linked approach has not been used for these studies. In recent studies, Performance comparison on image classification with the accuracy of the ML models, fuzzy measure, decision tree, artificial neural network, as well as other support vector machine methods oriented results are obtained from the literature survey.

ACKNOWLEDGEMENT

I am extremely grateful to my project guide, Dr Shailender Kumar, Professor, Department of Computer Science and Engineering, Delhi Technological University, Delhi, for furnishing inestimable guidance and being a constant source of alleviation throughout my research. I'll always be indebted to him for the expansive support and stimulant he provided. I am highly indebted to the panel faculties during all the progress evaluations for their guidance, constant supervision and for motivating me to complete my project work. They helped me throughout by giving me new ideas, providing necessary information and pushing me forward to complete the project work.

PIYUSH PRIY

(2K19/CSE/501)

CONTENTS

Contents

CANDIDATE'S DECLARATION i

CERTIFICATE ii

ABSTRACT iii

ACKNOWLEDGEMENT IV

CONTENTS v

LIST OF FIGURES vii

LIST OF SYMBOLS, ABBREVIATIONS AND NOMENCLATURE ix

CHAPTER 1 INTRODUCTION 1

1.1 OVERVIEW 1

1.2 PROBLEM STATEMENT 3

1.3 PROJECT OBJECTIVE 4

1.4 DATA CAPTURING AND PROCESSING 5

1.5 INFORMATION RETRIEVAL AND METHODS 6

1.6 IMAGE PROCESSING 9

1.7 APPLICATIONS OF IMAGE PROCESSING 12

1.8 ROLE OF MACHINE LEARNING 12

1.9 TYPES OF MODELS USED IN MACHINE LEARNING 15

1.10 PERFORMANCE METRICS 25

1.11 TOOLS USED 28

CHAPTER 2 LITERATURE SURVEY 31

CHAPTER 3 METHODOLOGY 35

3.1 STEPS FOLLOWED 35

3.2 DATA COLLECTION 36

3.3 TRAIN AND TEST SPLIT 36
3.4 PRE-PROCESSING INFORMATION.....38
3.4.1 DATA SIMILARITY AND HASH VALUE GENERATION.....38
3.5 MODEL BUILDING & TRAINING 39
3.5.1 FINE TUNING MODEL.....42
CHAPTER 4 RESULTS AND ANALYSIS.....43
4.1 RESULTS.....43
4.1.1 DATA IDENTIFICATION 43
4.1.2 IMAGE PROCESSING 44
CHAPTER 5 COMPARATIVE ANALYSIS OF THE PROPOSED APPROACH 46
5.1 TENSERFLOW ANALYSIS 46
CHAPTER 6 CONCLUSION AND FUTURE SCOPE.....47
6.1 CONCLUSION.....47
6.2 FUTURE SCOPE 47
REFERENCES 49

LIST OF FIGURES

FIG. NO.	FIGURE NAME
Fig. 1.1	Working of Data Capturing System
Fig. 1.2	Data Capturing System
Fig. 1.3	Flow of Data Capturing System
Fig. 1.4	Classification of Data Capturing System
Fig. 1.5	Image Resolution
Fig. 1.6	Face Detection
Fig. 1.7	Facial Recognition
Fig. 1.8	Valid image type
Fig. 1.9	Types of Machine Learning
Fig. 1.10	Decision Tree Classifier
Fig. 1.11	Random Forest Classifier
Fig. 1.12	Linear Regression Model
Fig. 1.13	K-Nearest Neighbour Classification
Fig. 1.14	Naive Bayes Classifier
Fig. 1.15	Gradient Booster Classifier
Fig. 1.16	CatBoost Classifier Model
Fig. 1.17	XGBoost Classifier model
Fig. 1.18	Multi-Layer Perceptron Classifier (MLP)
Fig. 3.1	Steps taken in this project
Fig. 3.2	Bar chart of the image dataset
Fig. 3.3	Training and Validation Set

Fig. 3.4 Histogram resembling all the labels in the dataset
Fig. 3.5 Correlation Matrix
Fig. 3.6 TensorFlow architecture
Fig. 3.7 Machine Learning Pipeline
Fig. 4.1 Applying Neumann distancing
Fig. 4.2 Performance Metrics for Data Relations
Fig. 4.3 Data De-Duplication
Fig. 4.4 Classification of class: Female
Fig. 4.5 Classification of class: Male
Fig. 4.6 Classification of class: Unclassified
Fig. 4.7 Confusion Matrix
Fig. 5.1 Hash value and Five Pattern Analysis
Fig. 5.2 Result on the final cluster of data

LIST OF SYMBOLS, ABBREVIATIONS AND NOMENCLATURE

SHORTCUT	ABBREVIATION
1. DCS	Data Capturing System
2. SVM	Support Vector Machine
3. ANN	Artificial Neural Network
4. ML	Machine Learning
5. IDR	Intelligent Document Recognition
6. NLP	Natural Language Processing
7. OCR	Optical Character Recognition
8. GPT	Generative Pre-trained Transformer
9. BERT	Bidirectional Encoder Representations from Transformers
10. WWW	World Wide Web
11. IDE	Integrated Development Environment
12. AI	Artificial Intelligence
13. QR	Quick Response
14. LR	Linear Regression

CHAPTER 1 INTRODUCTION

1.1 OVERVIEW

The data processing field in recent years has witnessed a remarkable revolution, primarily driven by the widespread adoption of online Data Capturing Systems by major organizations. These systems have become invaluable tools for gathering accurate and precise information about their targeted audience. However, despite the significant advantages offered by these systems, there are numerous challenges associated with reading, analyzing, and ensuring the correct identification of the captured data.

One of the primary benefits of data capture lies in its potential to enhance business processes. By employing automated techniques, data capture systems enable the extraction of relevant information from various types of documents, regardless of their structure or format, whether it be structured or unstructured, in the form of paper documents or electronic files. This process involves the sophisticated classification and validation of the extracted data, ensuring its accuracy and reliability. Subsequently, the transformed information is converted into a machine-readable digital format, facilitating faster and more effective information retrieval and subsequent analysis.

The advancements in technology, particularly Artificial Intelligence (AI), Big Data etc have played a pivotal role in propelling data capture to new heights. AI algorithms and techniques have greatly improved the accuracy and efficiency of data capture systems. Many algorithms in Machine learning such as deep learning, neural networks etc enable automated interpretation and understanding of various types of data, including text, images, and even audio. This has empowered data capture systems to handle a wide range of data sources and formats, further streamlining the entire process.

Additionally, AI-powered optical character recognition (OCR) technologies have significantly contributed to the success of data capture systems. OCR algorithms can accurately extract text from scanned documents or images, enabling the conversion of physical documents into editable and searchable digital formats. This capability

has revolutionized the way organizations handle large volumes of documents, leading to improved productivity and cost savings.

Furthermore, natural language processing (NLP) techniques have enabled data capture systems to comprehend and interpret unstructured textual data. This includes extracting key entities, relationships, and sentiments from textual sources, allowing organizations to derive valuable insights from unstructured information.

Overall, the convergence of online Data Capturing Systems and AI technologies has ushered in a new era of efficient and accurate data processing. The ability to capture, classify, and transform data from a wide range of sources has immensely benefited organizations, enabling them to take informed decisions and also improve operational efficiency to gain a competitive edge. As technology continues to advance, the potential for data capture systems to revolutionize the way organizations handle information is limitless, promising a future where data-driven insights drive innovation and success.

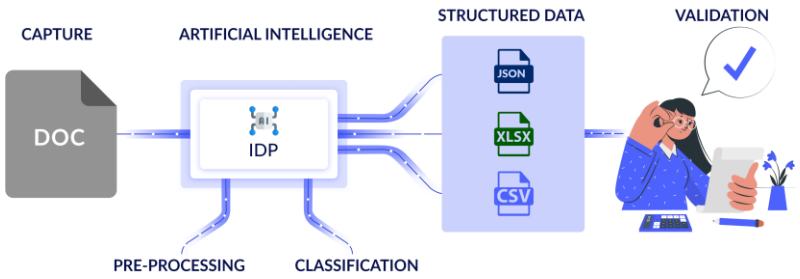

Fig 1.1 Working of Data Capturing System

1.2 PROBLEM STATEMENT

The presence of fake or dummy data in data-capturing systems (DCS) poses a significant challenge to the reliability and accuracy of the captured data. The problem at hand is to address the issue of fake or dummy data within DCS and develop effective methods to detect and mitigate its impact on data quality and analysis.

The specific challenges involved in this problem include:

a) Fake Data Generation: Identifying and understanding the various techniques used to generate fake or dummy data within DCS. This includes deliberate data manipulation, insertion of fabricated information, or the introduction of misleading data elements.

b) Data Validation and Integrity: Develop robust validation techniques to verify the authenticity and integrity of the captured data. This involves implementing data validation checks, anomaly detection algorithms, and verification mechanisms to identify suspicious or inconsistent data entries.

c) Source Verification: Establishing reliable methods to verify the source and origin of the captured data. This includes assessing the credibility and trustworthiness of the data sources to ensure the validity and accuracy of the information being captured.

d) Detection and Filtering: Designing algorithms and systems capable of detecting and filtering out fake or dummy data from the overall dataset. This requires the implementation of intelligent data analysis techniques, machine learning models, or pattern recognition algorithms to identify and separate fake entries from genuine ones.

e) Prevention and Security Measures: Implementing preventive measures to minimize the occurrence of fake or dummy data in the first place. This may involve strengthening data access controls, incorporating encryption techniques, or employing secure data transmission protocols to reduce the likelihood of data manipulation or unauthorized tampering.

By addressing these challenges, the goal is to develop robust mechanisms and methodologies to detect, validate, and mitigate the presence of fake or dummy data within DCS. The successful resolution of this problem will ensure the integrity, accuracy, and reliability of the captured data, enabling organizations to make informed decisions, improve data analytics, and maintain the trustworthiness of their data-capturing systems.

1.3 PROJECT OBJECTIVE

The primary objective of this project is to propose system methodologies for data processing and classification, addressing the needs of various sectors in the present world. As the volume of data in different domains continues to grow, there is a pressing requirement to process and classify this data efficiently at various stages of information processing.

In this project, we review recent machine learning (ML) automated methods that utilize deep learning techniques to determine and classify data. Deep learning has shown significant promise in accurately categorizing and classifying diverse types of data. Additionally, we explore the application of image classification, which focuses on automatically identifying object boundaries and other features. For understanding image content this plays a fundamental role for searching and mining in information processing domains. Manual classification or segmentation is often infeasible for large-scale systems and can overlook minute information fragments.

The development of impersonation detection methods using image data analysis is crucial to overcome the limitations of manual segmentation. The complexity of the information lies in the identification of intricate details and fragment ordering, which can be influenced by factors such as ambiguity, re-ordering, or the presence of partial data.

To address these challenges, our project aims to identify similarities in data through image analysis, along with other relevant parameters, in large-scale information systems. This analysis will facilitate the provision of consistent data across different levels and enable the deduplication of any discovered information.

By leveraging advanced ML techniques, particularly deep learning, and incorporating image analysis, this project seeks to enhance data processing and classification in various sectors. The proposed system methodologies will contribute to efficient and accurate data analysis, thereby enabling informed decision-making and optimizing the utilization of information resources.

1.4 DATA CAPTURING AND PROCESSING

The real-world data contain numerous differential forms of data that can be categorised into different sets. Since the data assembly systems keep the data in targeted directories and each record is identified based on the similarities and uniqueness of data.

In any data set, the information collection and its processing method depend on the nature of the data to be captured and the application area. During the initial phase, the system identifies the information fields to collect data followed by an appropriate method finalization. To digitalize unstructured data requires specialized data capture tools and processing the information before its use.

Following is a general structure of a Data Capturing and Processing shown in Fig 2.1

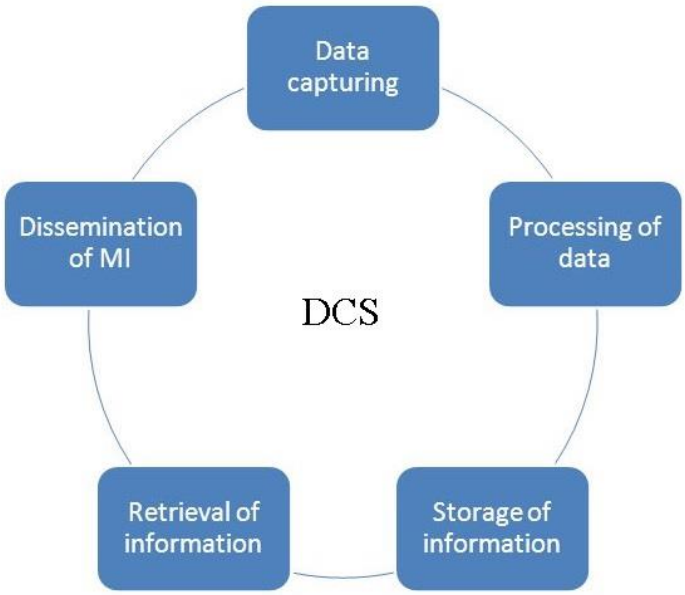

Fig. 1.2 Data Capturing System

The Data capturing methods have been used in various forms from last few decades. The data capturing methods also depends on the type of business and information requirement. Capturing data from written forms (hardcopy), PDF files, emails, etc. is also made possible using the technologies.

1.5 INFORMATION RETRIEVAL AND METHODS

Manual Data Capture: The elementary method to convert written forms (hardcopy) into digital data (softcopy). It is suitable for areas where the amount of data is very low and fewer variables.

Automated Data Capturing: Automated data capture systems not only manage data efficiently but also by reducing cost and labour inefficiency.

Optical Character Recognition (OCR): the OCR is a technology that identifies machine generated characters, typefaces and other form of data into digital form.

Intelligent Character Recognition (ICR): Intelligent Character Recognition is latest technology for OCR mechanism. It can also read handwritten characters.

Intelligent Document Recognition (IDR): IDR combines various AI technologies for e.g. natural language processing, Optical Character Recognition, Computer Vision, etc. for patterns recognition.

Barcodes and Quick Response (QR) codes: encrypted information are stored in Barcode technology as 1D barcodes which can be read using a barcode scanner. QR codes uses 2D barcode technology which is more complex.

Optical Mark Reading (OMR): The electronic data capture using OMR technology method identifies darkened fields. Its an ideal tool due to its high accuracy use in objective-type collections such as examinations.

Digital forms: this method also known as online forms facilitates the data capture through online mode using the web or a mobile application.

Digital Signatures: Equivalent to a signature (handwritten), they are used for approvals, authorizations and computerized access for messages and documents. DS are valid legally and provide high security to the original documents against impersonation.

Web Scraping: The use of various tools in the World Wide Web (WWW) such as bots, crawlers to search and collect data from different resources and for individual or collective use transfer it to different databases.

Magnetic Stripe Cards: Cards that contain encoded data on the magnetic stripes such as ATM, Credit Card etc.

Magnetic Ink Character Recognition (MICR): MICR recognizes encoded data in machine characters printed using magnetic ink.

Smart Cards: The information is store in encrypted for on a microprocessor chip for added protection for identification.

Voice Capture: Voice data capturing uses various technologies of speech recognition for processing data.

Video/ Image Capture: The digital print, for an Image or video can be captured using different AI technologies in order to identify and extract data regarding individuals with accuracy. It provides a machine-generated transcript as well as data related to the identified sections.

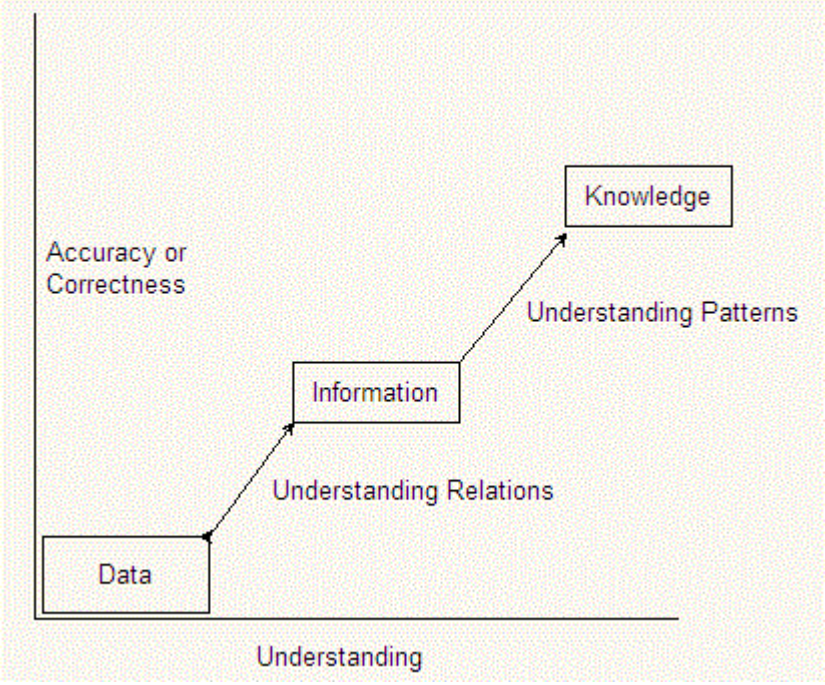

Fig. 1.3 Flow of Data Capturing System

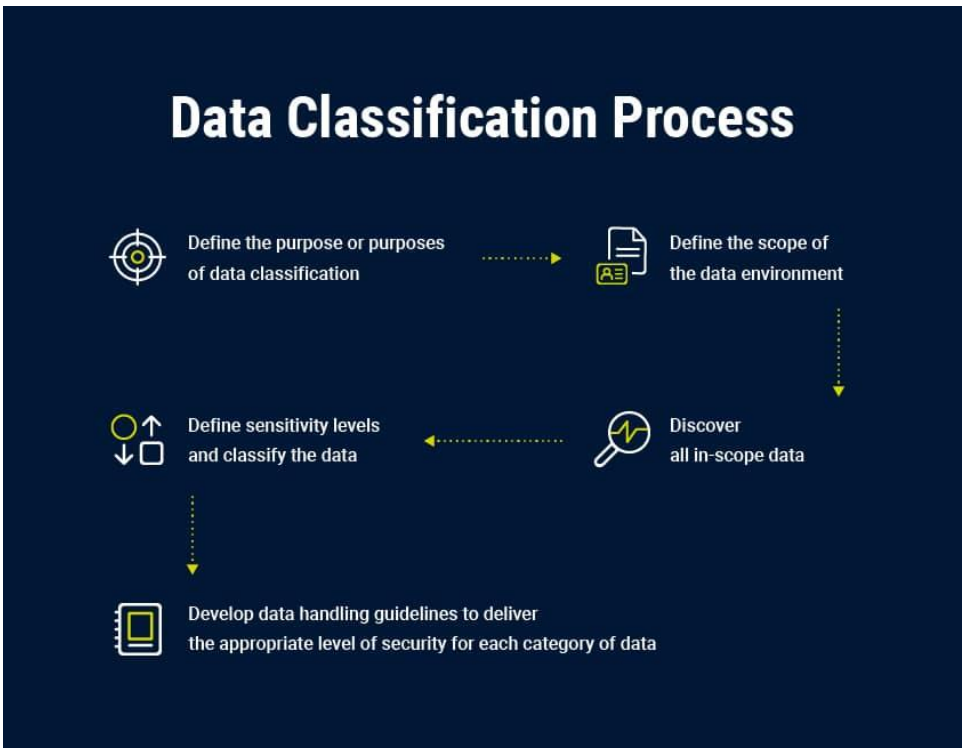

Fig. 1.4 Classification of Data Capturing System

1.6 IMAGE PROCESSING

The impact of digital images on data processing sectors is huge, and image processing is a critical element in Engineering as well as major science and technology sectors. The swift progress in automated image classification, and the related developments in analysis styles and various methods available to perform operations on an image with the goal of obtaining an enhanced version or extracting useful information from it., have made image analysis one of the most substantial subfields in this sector. Some of the extensively used operations involving the exercise of imaging in data processing are Image Resolution, Face Detection, Gender Classification, valid image classification etc.

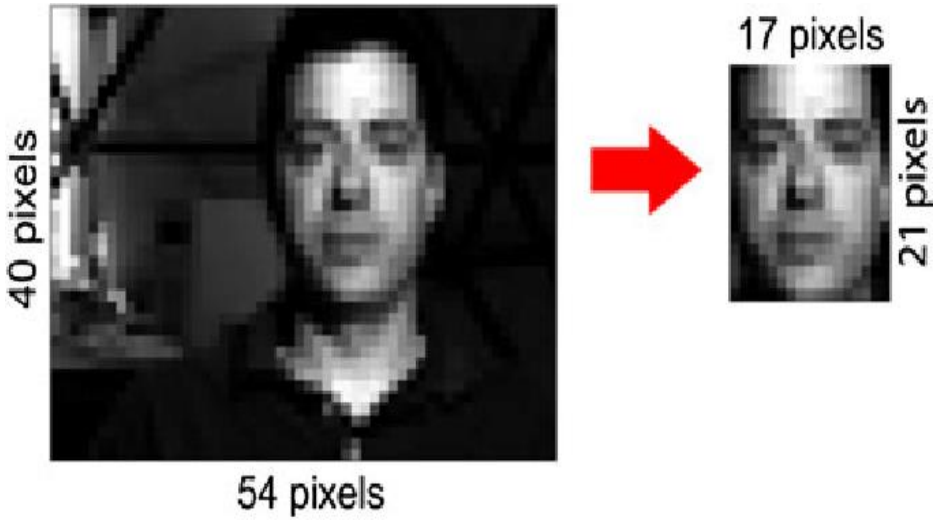

Fig. 1.5 Image Resolution

Fig. 1.6 Face Detection

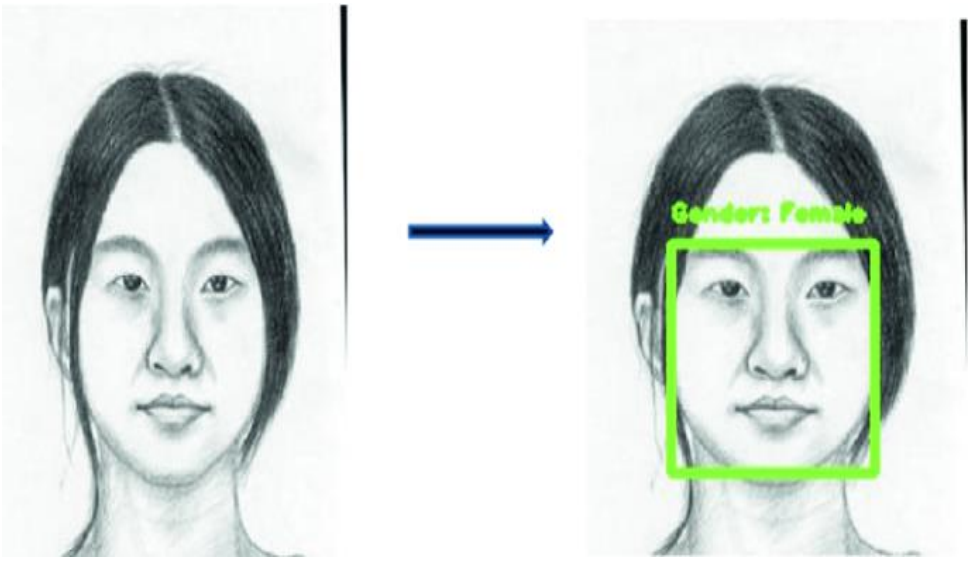

Fig. 1.7 Facial Recognition

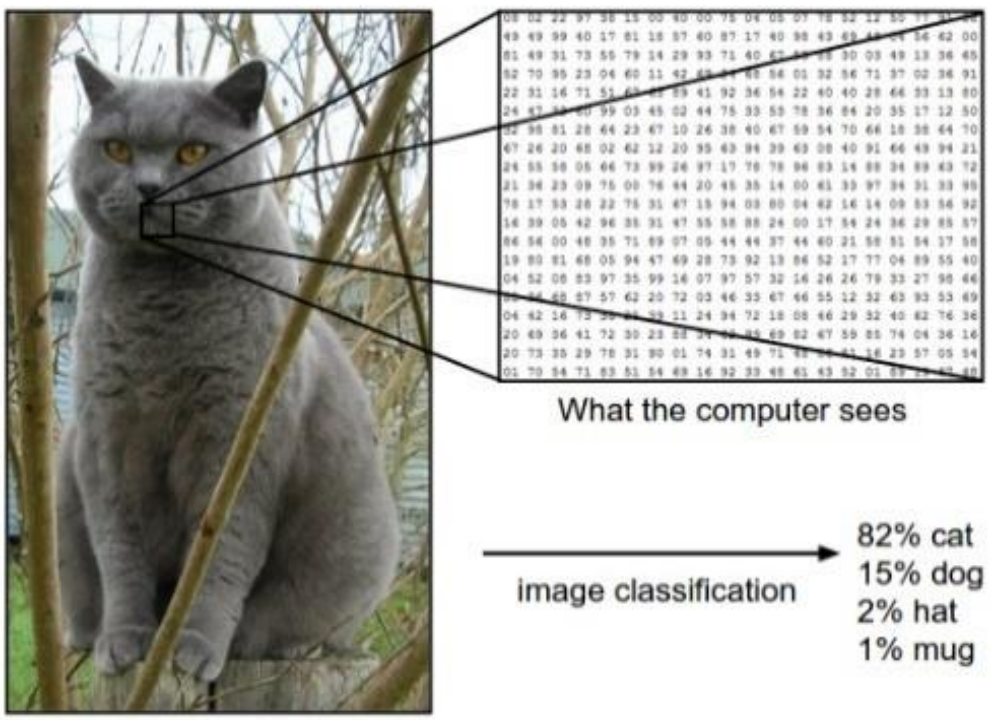

Fig. 1.8 Valid image type

1.7 APPLICATIONS OF IMAGE PROCESSING

Digital Image Processing has distinctive types of methodologies and techniques to manipulate or modify an image with the help of computers. There are so numerous types of operations carried out on the image.

Image is a 2-D array of values that are between 0- 255. 0 belongs to black while 255 belongs to white.

• Computerised Imaging (e.g., Photoshop)

• Deep Space Image processing (e.g., Interplanetary probe images)

• Fingerprint / face recognition / iris recognition

• Object / Traffic Image processing etc.

1.8 ROLE OF MACHINE LEARNING

Machine learning plays a crucial role in analyzing fake data or fake images by providing effective techniques to detect and identify instances of falsified or manipulated information. The following are key roles that machine learning techniques fulfil in analyzing fake data or fake images:

1. Anomaly Detection: Machine learning algorithms can also be trained for patterns recognition and identifying any anomalies in chunk of data. By learning from a large dataset of genuine data, these algorithms can detect deviations or inconsistencies that indicate the presence of fake data. Anomaly detection algorithms can flag suspicious entries or images that do not conform to the expected patterns, enabling further investigation.

2. Image Forgery Detection: Machine learning models can be trained to identify image forgery or manipulation. These models learn from a diverse range of authentic and manipulated images to detect specific artefacts or inconsistencies introduced during image tampering. Techniques such as image forensics, deep learning, and convolutional neural networks (CNNs) can be employed to analyze image features and identify signs of manipulation.

3. Natural Language Processing (NLP): Machine learning-based NLP techniques can be used to analyze textual data and identify indicators of fake information. NLP models can learn to distinguish between genuine and fake text by considering factors such as grammar, syntax, sentiment analysis, and semantic coherence. This enables the identification of misleading or fabricated textual content.

4. Data Validation and Verification: Machine learning algorithms can aid in data validation by learning from historical or reference datasets. These algorithms can assess the consistency and reliability of captured data, comparing it to trusted sources or known patterns. By verifying the authenticity and integrity of the data, machine learning can help identify and flag instances of fake data.

5. Ensemble Learning: Ensemble learning techniques, such as combining multiple machine learning models or classifiers, can improve the accuracy and robustness of fake data or fake image detection. Ensemble methods aggregate the predictions of multiple models, leveraging the strengths of individual models while minimizing the impact of false positives or false negatives.

6. Continuous Learning and Adaptation: Machine learning models can be continuously trained and updated with new data to adapt to evolving techniques used in generating fake data or fake images. By incorporating feedback loops and retraining the models with updated datasets, machine learning can stay ahead of emerging fake data or image manipulation methods.

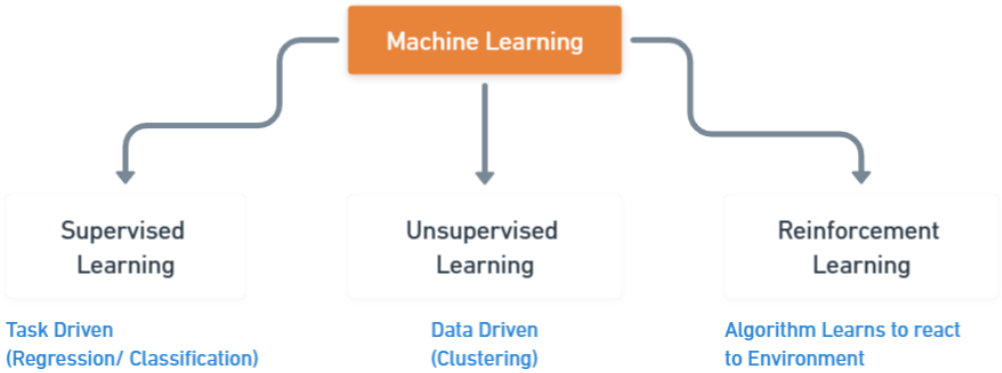

Fig. 1.9 Types of Machine Learning

The different machine learning algorithms under categories of supervised learning, unsupervised learning, and deep learning, can be used:

a) **Supervised learning:** The most prevalent kind of machine learning is supervised learning. the input data is accompanied by corresponding output or goal values, and the algorithm is trained on this set of labelled data. The algorithm learns to predict the output value by minimizing the disparity between its predicted output and the actual output, thereby improving its ability to anticipate the expected output based on a given input. Applications for supervised learning includes image recognition, natural language processing and speech recognition.

b) **Unsupervised learning**: this learning, involves learning from unlabelled data. The algorithm works towards finding various patterns and relationships in the data without any prior knowledge of what it should be looking for. learning of this type is commonly used for dimensionality and clustering reduction tasks where identifying hidden structures in data can provide valuable insights.

c) **Reinforcement learning:** this type of learning where an agent learns to decide (to make decisions) based on pulse/feedback received from its connecting

environment. The designated agent/learning interacts with the environment and receives rewards or penalties depending on the actions it takes. Over time, the agent learns to take actions that maximize its rewards. Applications for reinforcement learning include robotics, interactive games, and autonomous systems.

Machine learning algorithms provide valuable tools for the automated analysis and identification of fake data or fake images. By leveraging advanced techniques and models, organizations can enhance their ability to detect and combat the presence of falsified information, ensuring the integrity and reliability of their data analysis processes.

1.9 TYPES OF MODELS USED IN MACHINE LEARNING

1.9.1 Decision Tree: A popular ML technique called decision trees is used for regression and classification problems. The supervised learning method that is built on a structure that resembles a hierarchical tree. A choice is made based on a feature or attribute value at each level of the tree until one is made at the leaf node. Recursively dividing the data into smaller subsets depending on the feature that yields the greatest information gain is how the decision tree model operates. Till an conclusion for the requirement is satisfied, such as the bare minimum quantity of test data or till the depth (maximum) of the tree, this procedure is continued. One of the key benefits of decision trees is their interpretability, which makes them useful in applications where understanding the reasoning behind the predictions is important. Decision trees can also work well with both categorical and numerical data, and they can handle missing values. However, decision trees tend to overfit on training data, leading to poor performance on new data. This problem can be mitigated by techniques such as pruning, ensemble methods, or using random forests.

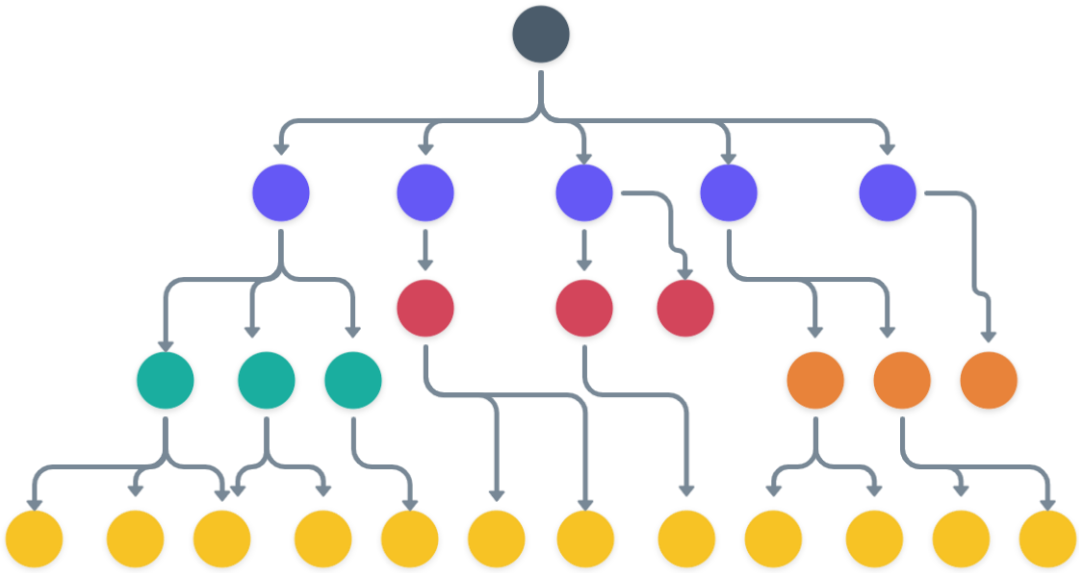

Fig. 1.10 Decision Tree Classifier

1.9.2 Random Forest: Random Forest is a popular machine learning algorithms which is used for regression and classification tasks. Different decision trees are combined in this ensemble learning method to increase accuracy and decrease overfitting. To ensure variation among the base models, the algorithm builds each decision tree by randomly selecting subsets of the input data and characteristics.

During prediction, the Random Forest aggregates the responses of all individual trees and outputs the most common result. One of the key benefits of Random Forest is its ability to handle high-dimensional datasets with many features, as well as missing or noisy data. It also provides insights into feature importance, allowing users to identify which variables are most influential in the model's predictions. Furthermore, the algorithm can be easily parallelized, making it efficient for large-scale datasets and distributed computing environments.

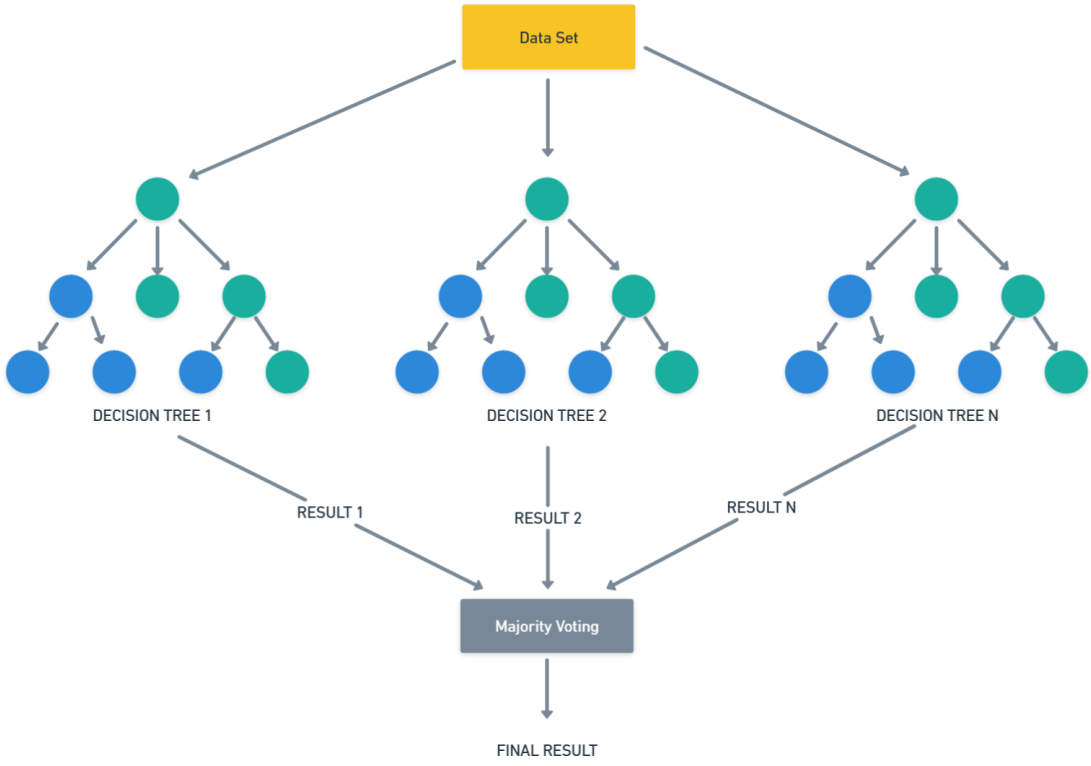

Fig 1.11 Random Forest Classifier

1.9.3 Linear Regression: linear regression is an statistical technique used to represent the relationship between two or more variables, where one is the dependent variable, and the other is the independent variable. Finding the best-fit line that best captures the connection between the variables is the basic goal of linear-regression, which may then be used for analysis or prediction. The process involves utilising the least squares methodology to fit a linear equation to the data points. The best fit is produced by this method, which minimises the squared sum of the discrepancies between the values predicted and the actual values. LR is widely used in various fields, such as economics, engineering, finance, and social sciences, for making predictions, forecasting trends, and analysing relationships between variables. It is a simple yet effective method that provides valuable insights into the underlying patterns in the data. Some common applications of Linear Regression include predicting stock prices, sales forecasts, weather predictions, and analysing the impact of marketing campaigns on product sales.

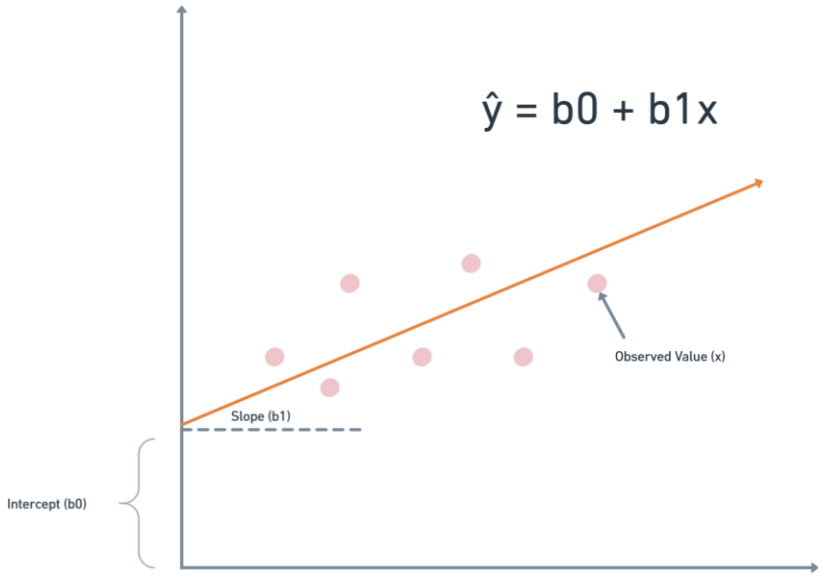

Fig. 1.12 Linear Regression Model

1.9.4 K-Nearest Neighbour: Popular machine learning algorithms for classification and regression issues include K-Nearest Neighbours (KNN). Because it is nonparametric, it doesn't make assumptions about the data distribution underlying. The class of a new data point's nearest neighbours determines its output in a KNN. The size of the dataset and the type of problem are taken into account when determining the value of 'K', which denotes the total number of nearest neighbors to take into account. To predict the class for a new data point, The algorithm (KNN) determines the separation of the new point from every other point in the training set data. The new data point is then given a class based on the most prevalent class among the K closest neighbours, which are chosen based on the estimated distance. KNN can be applied to a number of tasks, including predicting customer churn, identifying handwritten numbers, and detecting spam emails. One of the key advantages of KNN is that it does not require any training time, as it simply stores the entire training set in memory. However, it can be computationally expensive for large datasets.

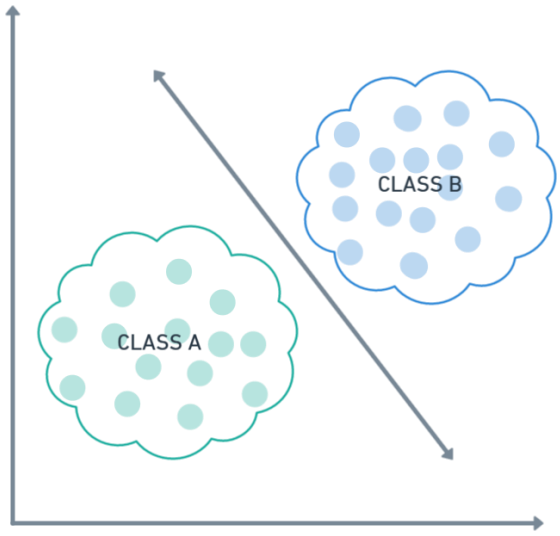

Fig. 1.13 K-Nearest Neighbour Classification

1.5.6 Naive Bayes Classifier [6]: A popular probabilistic machine learning approach for classification tasks is the Naive Bayes Classifier [6]. The term "naive" refers to its foundation in the Bayes theorem and the assumption that all features are independent of one another. This presumption makes calculating probabilities easier and more computationally efficient for huge datasets. The Naive Bayes Classifier model determines the likelihood that each class will exist given a set of input features. These probabilities are then multiplied together to get the posterior probability of the input belonging to a particular class. One of the key advantages of the Naive Bayes Classifier is its ability to work with high-dimensional data, making it suitable for text classification and spam filtering. It also performs well even with a small amount of training data. However, the Naive Bayes Classifier makes a strong assumption about independence between features, which can limit its accuracy in some cases. It also relies heavily on the quality of the input data and can be sensitive to outliers.

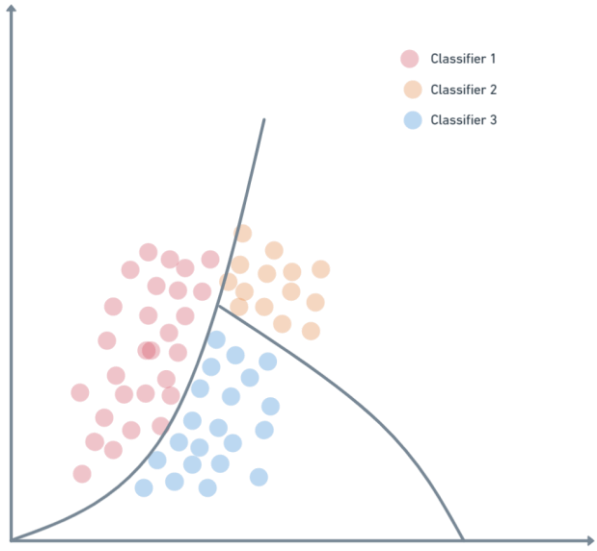

Fig. 1.14 Naive Bayes Classifier

1.5.7 Gradient Boosting Classifier: With the help of the well-known machine learning algorithm gradient boosting classifier, numerous weak learners can be combined to produce a powerful prediction model. It functions by iteratively adding new decision trees to the model, each of which tries to fix the flaws of the preceding trees. Based on the gradient descent optimisation process, which updates the parameters of model towards the direction of the loss function's, negative gradient, is the gradient boosting classifier model. This approach allows the model to learn from its mistakes and gradually improve its accuracy. One of the key benefits of gradient-boosting classifiers is their ability to handle complex, high-dimensional data sets with a large number of features. They also tend to perform well on imbalanced datasets, where the number of samples in each class varies widely. However, gradient-boosting classifiers can be prone to overfitting if not properly regularized, and they can be computationally expensive to train, especially when dealing with large datasets.

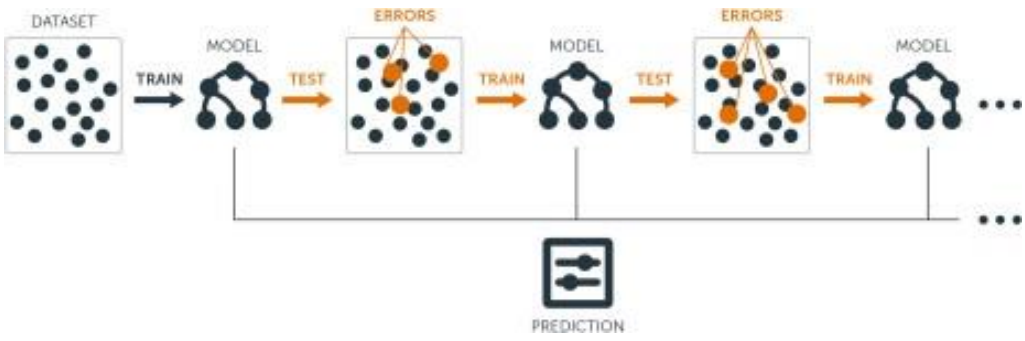

Fig. 1.15 Gradient Booster Classifier

1.5.8 CatBoost Classifier: One of CatBoost's main advantages is its ability to automatically handle missing values and categorical variables without the need for feature engineering or pre-processing. It was developed by Yandex and is known for its high performance on large datasets with mixed data types. The catBoost classifier model operates by constructing an ensemble of decision trees, following a similar approach as other gradient-boosting algorithms. However, it also includes a number of advanced techniques to handle categorical variables more effectively, such as target encoding, ordered boosting, and feature combinations. One of the key benefits of catBoost is its ability to automatically handle missing values and categorical variables without requiring any pre-processing or feature engineering. It is also designed for scalable toward higher end, making it suitable for large datasets with millions of rows and thousands of columns. However, like other gradient-boosting algorithms, catBoost can be computationally intensive and may require significant computational resources to train. It also requires attentive tuning of hyperparameters to get the optimal performance.

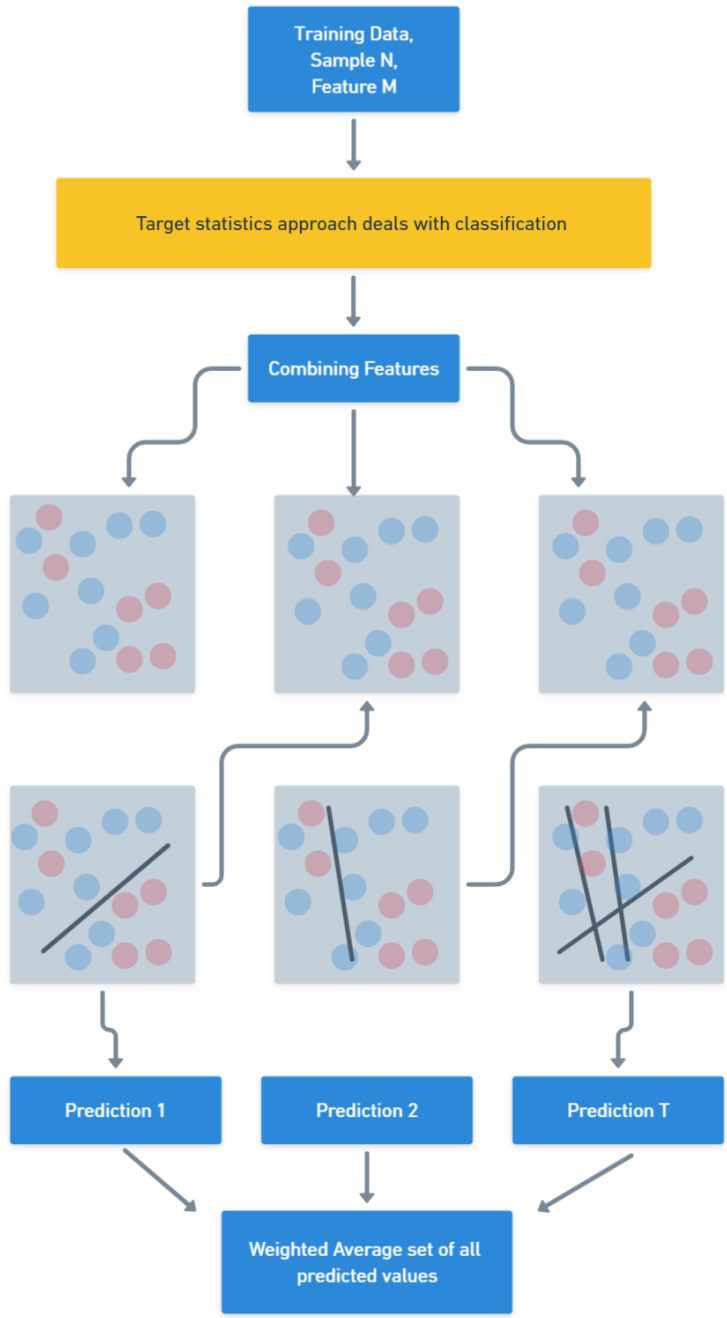

Fig. 1.16 CatBoost Classifier Model

1.5.9 XGBoost Classifier [20]: The gradient boosting algorithm is a well-liked ML approach for handling classification and regression issues, is optimised for use in the XGBoost Classifier. The name XGBoost, which stands for "Extreme Gradient Boosting," denotes the emphasis on performance optimisation through effective computation and feature engineering. The XGBoost classifier model works by creating an ensemble of multiple decision.

The catBoost classifier model utilizes a collection of trees, with each tree aiming to rectify the errors made by the preceding tree. Additionally, the model incorporates regularization techniques to counteract overfitting and enhance its ability to generalize well to unseen data. One of the key benefits of XGBoost is its speed and scalability, making it suitable for large datasets with many features. It can also handle missing values, making it robust to real-world data scenarios. However, like other gradient-boosting algorithms, XGBoost can be prone to overfitting if not properly regularized. It also requires careful tuning of hyperparameters to optimize its performance.

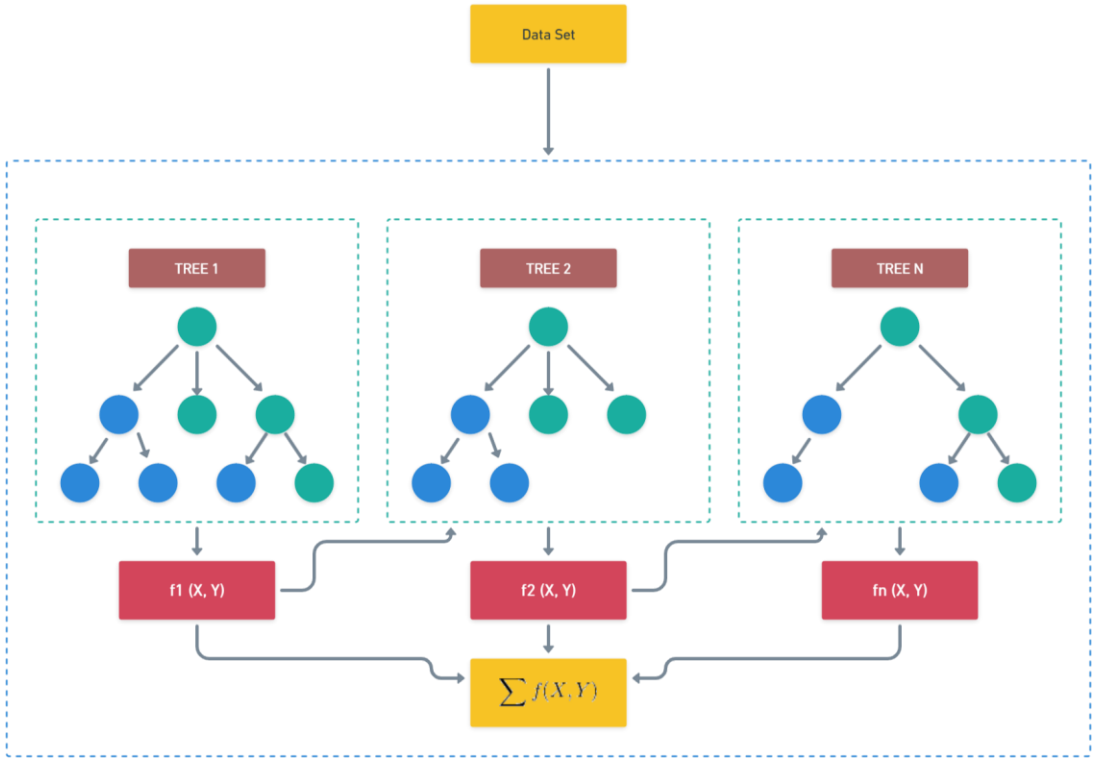

Fig. 1. 17 XGBoost Classifier model

1.5.10 Multi-layer Perceptron (MLP): MLP Classifier is a type of artificial neural network algorithm used for classification tasks. It consists of multiple layers of interconnected nodes, known as neurons, which are organized in a feedforward manner. The MLP classifier model works by learning a non-linear function that maps the input features to the output class labels.

Each node in the hidden layers applies a weighted sum of its inputs and passes the result through an activation function to produce an output. One of the primary benefits in MLP classifiers is learning of complex decision boundaries, making them suitable for a wide range of classification problems. They can also handle continuous and categorical data and can be trained using various optimization algorithms. Optimal performance of MLP classifiers necessitates careful tuning of hyperparameters, including the number of hidden layers, the number of neurons within each layer, and the choice of activation functions. Additionally, attention must be given to the risk of overfitting, particularly when working with limited datasets.

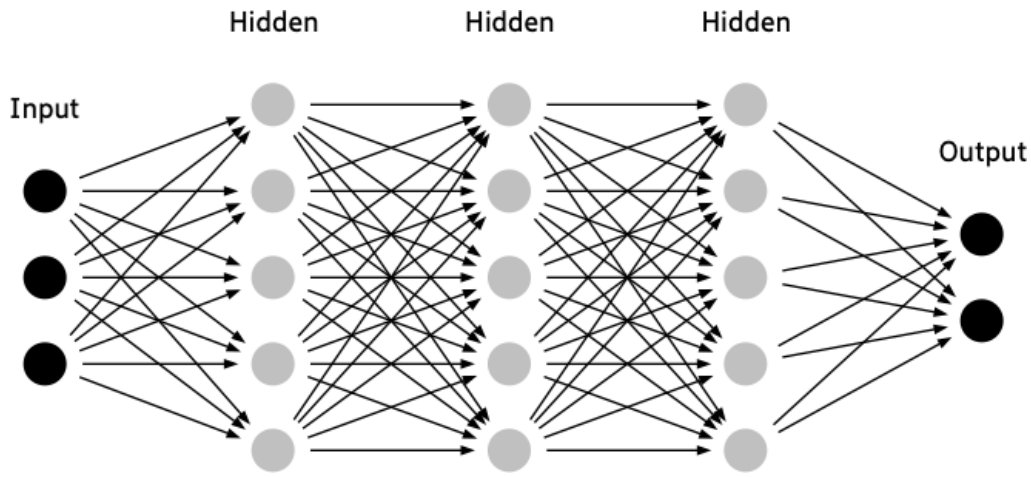

Fig. 1. 18 Multi-Layer Perceptron Classifier (MLP)

1.5.11 Fully Convolutional Network (FCN): Fully Convolutional Networks (FCNs) are a type of deep neural network that has been developed specifically for semantic segmentation in computer vision. Unlike other convolutional neural networks (CNNs), FCNs can take inputs of any size and produce outputs with the same spatial dimensionality as the input. This makes them particularly useful for tasks such as object detection, where it is important to identify the location of objects within an image. The flexibility of FCNs is achieved by replacing fully connected layers with convolutional layers. This allows the network to preserve spatial information throughout the layers rather than reducing it to a single vector for classification purposes. Additionally, FCNs

incorporate skip connections that enable the network to use information from earlier layers to refine the output. FCNs have proven highly effective for various applications in computer vision, including image segmentation, depth estimation, and image-to-image translation. They continue to be an area of active research, with ongoing efforts to enhance their performance and versatility.

1.10 PERFORMANCE METRICS

To compare different models used in this project, we have used the following performance metrics to evaluate models:

1. **Accuracy**: Accuracy is a commonly used performance metric in machine learning and data analytics that measures the proportion of correct predictions made by a model. It is a simple, straightforward measure that is easy to interpret, also making it a common choice for evaluating the effectiveness of different models. We divide the number of accurately predicted cases by the total number of instances in the data-set to determine accuracy. For example, if a model correctly predicts 85 out of 100 instances, its accuracy is 85%.

$$Accuracy = \frac{CorrectOutputs}{TotalDataItems} \tag{1.1}$$

While accuracy is a needed measure of overall performance, it can be misleading in certain situations. For example, if the dataset is imbalanced, with one class vastly outnumbering the others, a model that can easily predicts the majority class for every instance will have higher accuracy even though it is not actually making useful predictions. In addition, accuracy does not take into account the cost or benefit of different types of errors. In some cases, false negatives (predicting a negative outcome when it's actually positive) may be more costly than false positives (predicting a positive outcome when it's actually negative), or vice versa. Despite these limitations, accuracy remains a valuable performance metric in many applications, particularly when the dataset is balanced, and the cost of different types of

errors is roughly equal.

2. **Precision**: In the fields of ML and data science, precision is an extensively used performance parameter. Out of all occurrences that a model classified as positive, it calculated the percentage of instances that were correctly labelled as positive. More specifically, precision is defined as the ratio of true positives (tp) to the sum of true positives and false positives (fp).

$$Precision = \frac{TP}{TP + FP} \tag{1.2}$$

   In other words, precision can identify the total instances that were predicted as positive and were actually positive. High precision is desirable in situations where false positives are costly or detrimental. For example, in medical diagnosis, identifying a healthy patient as having a disease can lead to unnecessary and potentially harmful treatment. In this case, high precision is more important than high recall (the proportion of actual positives which were identified by the model). However, it's also important to note that precision should not be considered in isolation but rather in co-occurrence of performance metrics such as recall, accuracy, and F1 score. A model with high precision may have a low recall, meaning that it misses a large number of positive instances. Precision is a valuable performance metric that measures the accuracy of a model in identifying positive instances. However, it should be evaluated alongside other metrics to get a comprehensive understanding of a model's performance.
3. **Recall:** Recall is a performance metric used in machine learning to assess how well a categorization model is performing. It assesses the model's capacity to accurately identify every positive case in a given dataset. In other words, recall calculates the proportion of true positives (TP) out of all the actual positive cases (TP + false negatives). Mathematically, recall can be expressed as:

$$Recall = \frac{TP}{TP + FN} \tag{1.3}$$

A high recall value denotes that the defined model can correctly identify most of the positive cases in the dataset. On the other hand, a low recall value Suggests that the model is not performing well in identifying positive instances. The recall is particularly important in applications where it is crucial to identify all positive cases, even at the expense of some false positives. For example, in medical diagnosis, it is better to have a high recall rate so that all patients with a disease are identified, even if it means some healthy individuals are incorrectly diagnosed with the disease. Recall is an important performance metric in machine learning and helps to evaluate the classification models effectiveness in identifying positive cases.

4. **F-Score:** The F-score is a performance metric used in machine learning to assess how well a categorization model is performing. It creates a balance between these two criteria by combining precision and recall into a weighted harmonic mean. The F-score is calculated as follows:

$$F - Score = \frac{2 * (Precision * Recall)}{Precision + Recall} \tag{1.4}$$

where recall is the percentage of true positives among all occurrences of actual positive behaviour, and precision is the proportion of true positives among all positive predictions (TP / (TP + FP)). The F-score has a range of 0 to 1, with 1 denoting flawless recall and precision. In real-world applications, a high F-score means the model has successfully balanced precision and recall, ensuring that both false positives and false negatives are kept to a minimum. The F-score proves particularly valuable when working with imbalanced datasets, where one class is substantially more prevalent than the other. Since a model might obtain high accuracy by only predicting the majority class, accuracy may not be a desirable criterion in such circumstances. The F-score provides a better evaluation of the model's ability to correctly identify positive cases,

even in the presence of class imbalance. The F-score is an important performance metric in machine learning that helps to evaluate the effectiveness of classification models, especially in imbalanced datasets.

1.11 TOOLS USED

1. ML DOT NET: ML.NET is an open source framework for machine learning developed by Microsoft. It is specifically designed for .NET developers, allowing them to build machine learning models and easier integration of them into their .NET applications. ML.NET enables developers to leverage the power of machine learning algorithms and techniques without requiring expertise in data science or extensive knowledge of machine learning. ML.NET provides a wide range of machine learning tasks and algorithms that developers can utilize, including classification, regression, clustering, recommendation, anomaly detection, and more. It offers a high-level API and a variety of pre-built components, making it easier for developers to create and deploy machine learning models within their applications. ML.NET supports both training and inference scenarios, enabling the development of models that can scale to handle large datasets efficiently. It offers parallel processing capabilities, which can significantly accelerate training and prediction tasks. ML.NET provides features for model interpretability and explainability, enabling developers to understand and explain the decision-making process of their machine-learning models. This transparency helps build trust in the predictions made by the models.

2. **Visual Studio (VS):** Visual Studio is an integrated development environment (IDE) developed by Microsoft. It provides a comprehensive set of tools and features to aid developers in building software applications for various platforms. Visual Studio supports multiple standared programming languages, including C#, C++, Visual Basic, JavaScript, and Python, among others. Visual Studio offers a powerful and customizable feature rich code editor such as syntax highlighting, code completion, and code navigation. It provides a rich set of debugging tools, allowing developers to identify and fix

issues in their code efficiently. Visual Studio includes a robust debugger that enables developers to step through code, set breakpoints, and inspect variables during runtime. It supports both local and remote debugging, making it easier to diagnose and resolve errors. Visual Studio includes features for team collaboration, such as version control integration (e.g., Git) and built-in tools for code reviews and task management. It allows multiple developers to work on the same project concurrently and facilitates efficient collaboration and coordination. Visual Studio includes features for team collaboration, such as version control integration (e.g., Git) and built-in tools for code reviews and task management. It allows multiple developers to work on the same project concurrently and facilitates efficient collaboration and coordination.

3. **TensorFlow:** it is an open-source machine learning framework developed by Google. It is widely used for building and training machine learning models, particularly deep neural networks. TensorFlow provides a flexible and scalable platform for developing artificial intelligence (AI) applications, ranging from image and speech recognition to natural language processing and reinforcement learning. TensorFlow represents computations as a directed graph, where nodes represent mathematical operations and edges represent the data flow between operations. This graph-based approach enables efficient parallel processing and optimization of computations. TensorFlow provides a rich set of pre-built functions and tools for building various types of neural networks. It includes layers, activation functions, optimization algorithms, and regularization techniques that simplify the process of designing and training deep learning models. TensorFlow allows for distributed computing across multiple devices, machines, or clusters. It supports distributed training, inference, and data processing, enabling the scaling of computations to handle large datasets and complex models. TensorFlow allows for distributed computing across multiple devices, machines, or clusters. It supports distributed training, inference, and data processing, enabling the scaling of computations to handle large datasets and complex models.

4. **SQL Server:** SQL Server is a relational database management system (RDBMS) developed by Microsoft. It provides a robust and scalable platform for storing, managing, and retrieving structured data. SQL Server is widely used in various applications and industries, ranging from small businesses to enterprise-level organizations. SQL Server supports the relational model, allowing users to define tables, establish relationships between them, and enforce data integrity through constraints and rules. It provides a comprehensive set of SQL (Structured Query Language) functionalities for data manipulation, retrieval, and administration. SQL Server is designed to handle large-scale data and demanding workloads. It offers features like partitioning, parallel processing, and indexing techniques to optimize query performance and ensure efficient data retrieval. SQL Server also supports clustering and replication for high availability and scalability. SQL Server offers additional data services such as SQL Server Machine Learning Services, which allows users to execute R and Python scripts directly within the database engine. This enables the integration of machine learning and advanced analytics capabilities directly into SQL Server. SQL Server offers additional data services such as SQL Server Machine Learning Services, which allows users to execute R and Python scripts directly within the database engine. This enables the integration of machine learning and advanced analytics capabilities directly into SQL Server.

5. **Visual Studio Code:** Visual Studio Code is an advanced, free code editor developed by Microsoft that offers developers a lightweight cross-platform tool for editing and debugging code in various programming languages. It provides several powerful features such as IntelliSense code completion, syntax highlighting, Git integration, debugging, and extensions. The software's extensive set of extensions allows users to customize their development environment to suit different types of projects and workflows. Due to its user-friendly interface and comprehensive functionality, Visual Studio Code has become one of the most preferred tools for developers worldwide at all levels of experience, from novices to experts.

CHAPTER 2 LITERATURE SURVEY

2.1 RELATED WORK

There have been many research works proposed for this problem statement. Some briefs about their works are presented below for better understanding.

1. **"Deep Learning for Identity Theft Detection using Facial Images [1]":** focused on leveraging deep learning techniques for detecting identity theft using facial images. The paper was published in the International Conference on Pattern Recognition (ICPR) in 2016. The objective of the study was to address the growing concern of identity theft and develop an efficient method to detect such fraudulent activities using facial images. The authors proposed a deep learning-based approach that utilized convolutional neural networks (CNNs) to analyze and classify facial images. The research methodology involved collecting a dataset consisting of legitimate and forged identity images. The dataset was then preprocessed to enhance the quality and eliminate noise. Subsequently, a CNN model was designed and trained using the collected dataset. The model learned to extract relevant features from facial images and distinguish between legitimate and forged identities.
2. **"Impersonation Detection in Online Social Networks using Deep Learning [2]":** The objective of the study was to address the issue of impersonation, where malicious users create fake accounts or profiles to deceive others and engage in fraudulent activities. The authors proposed an approach based on deep learning, specifically using convolutional neural networks (CNNs), to automatically identify impersonators in online social networks. To conduct their research, the authors collected a large-scale dataset from a popular online social network platform, consisting of genuine and fake accounts. They performed preprocessing on the dataset to extract relevant features and prepare it for training the deep learning model. The CNN model was designed and trained to learn patterns and features indicative of impersonation in online social network profiles. The model was evaluated using various performance metrics, including accuracy, precision, recall, and F1-score, to assess its effectiveness in distinguishing between genuine and fake accounts. The experimental results presented in the paper demonstrated

the efficacy of deep learning techniques in impersonation detection. The proposed approach achieved high accuracy rates in identifying impersonators in online social networks, showcasing its potential in mitigating the risks associated with fraudulent activities. The study contributes to the field of cybersecurity and online social network analysis by highlighting the importance of deep learning methods in detecting impersonation. The findings have practical implications in combating online identity fraud, protecting user privacy, and ensuring a safer online social network environment. In conclusion, the paper "Impersonation Detection in Online Social Networks using Deep Learning" presents a valuable research contribution that showcases the effectiveness of deep learning, particularly CNNs, in detecting impersonation in online social networks. The work sheds light on the significance of leveraging advanced machine learning techniques to enhance the security and authenticity of online social interactions.

3. **" Fake Face Detection: A Survey [3]":** The objective of the study was to provide an overview of the existing methods and approaches for detecting fake or manipulated faces in images. With the rise of deepfake technology and the increasing threat of face forgery, it has become essential to develop robust and accurate techniques to identify fake faces. The authors conducted an extensive review of the literature, analyzing and categorizing the different fake face detection methods. They covered a wide range of techniques, including traditional methods based on handcrafted features, as well as more recent approaches utilizing deep learning and neural networks. The paper explores various aspects related to fake face detection, such as face manipulation techniques, image forensics, and benchmark datasets commonly used for evaluation. It also discusses the challenges and limitations in this field, including the emergence of sophisticated deepfake techniques that can produce highly convincing fake faces. The authors provide a detailed analysis of the strengths and weaknesses of different detection methods, discussing their performance, advantages, and potential applications. They also identify research directions and opportunities for future advancements in fake face detection.

4. **" Identity Theft Detection using Convolutional Neural Networks [4]":** The objective of the study was to address the growing concern of identity theft and develop an effective method to detect such fraudulent activities using machine learning techniques. The authors proposed an approach based on CNNs, which are a type of deep learning model known for their ability to extract relevant features from visual data. The researchers collected a dataset consisting of legitimate and fraudulent identity documents. The dataset was preprocessed to enhance the quality of the images and extract the necessary features. The CNN model was then designed and trained using the collected dataset to learn patterns and characteristics indicative of identity theft. The performance of the proposed method was evaluated using various metrics such as accuracy, precision, recall, and F1-score. The experimental results demonstrated the effectiveness of CNNs in identifying instances of identity theft, achieving high accuracy rates and demonstrating the potential for real-world application. The paper contributes to the field of identity theft detection by showcasing the efficacy of CNNs in automated identification and detection of fraudulent identity documents. The findings of this study have implications in various domains such as financial institutions, border control, and identity verification systems, where the detection and prevention of identity theft are crucial.

5. **Deep Learning Techniques for Detecting Facial Forgery [5]:** The objective of the study was to address the rising concern of manipulated and fake facial images, which can be used for various malicious purposes such as identity theft and misinformation. The authors proposed the use of deep learning models to analyze and detect the presence of facial forgery in images. The researchers developed a framework that utilizes convolutional neural networks (CNNs) to learn intricate features and patterns within facial images that can indicate signs of manipulation. The CNN model was trained on a large dataset of genuine and forged facial images, allowing it to learn and distinguish between real and manipulated faces. To evaluate the performance of their approach, the authors conducted experiments on

benchmark datasets and utilized metrics such as accuracy, precision, recall, and F1-score. The results demonstrated the effectiveness of the deep learning techniques in detecting facial forgery, with high accuracy rates achieved in identifying manipulated images. SThe paper contributes to the field of computer vision and image analysis by showcasing the potential of deep learning models, particularly CNNs, in addressing the challenge of detecting facial forgery. The findings have practical implications in domains such as forensics, authentication systems, and online security, where the identification of manipulated facial images is crucial.

CHAPTER 3 METHODOLOGY

3.1 STEPS FOLLOWED

The below diagram shows the steps taken in this project.

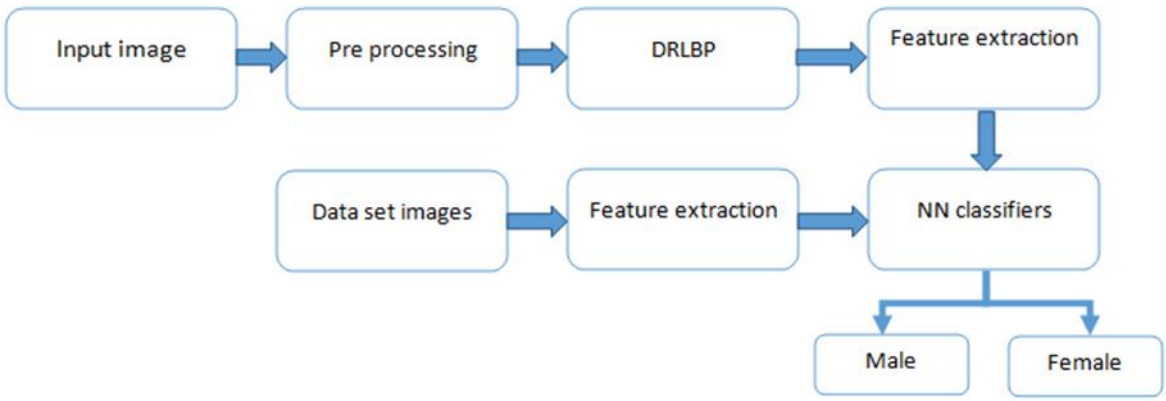

Fig. 3.1 Steps taken in this project

The following steps are taken in this project:

1. **Data Collection:** In this step, data is collected in the form of a CSV/Excel file. In this project, the DCS dataset is used.
2. **Data Analysis:** In this step, the information collection is being examined on different parameters.
3. **Data Pre-Processing:** In this step, the data is split into two categories, i.e., Training data and Testing data.
4. **Model Building and Training:** In this step, Machine Learning models are created and trained on the dataset processed in the above stage.
5. **Prediction:** The sample dataset is used to predict the outcomes of the trained model.
6. **Compare and Evaluate:** In this step, the performance of different models is evaluated and compared by different parameters like Precision, Recall, Accuracy, F-Score etc.

3.2 DATA COLLECTION

The Data, along with the image dataset, is collected from Examination Department conducting high-stakes examinations. It contains a total of 175551 records along with images and 60 testing images in the dataset. The following figure shows the chart and shows the availability of the dataset for five different classes.

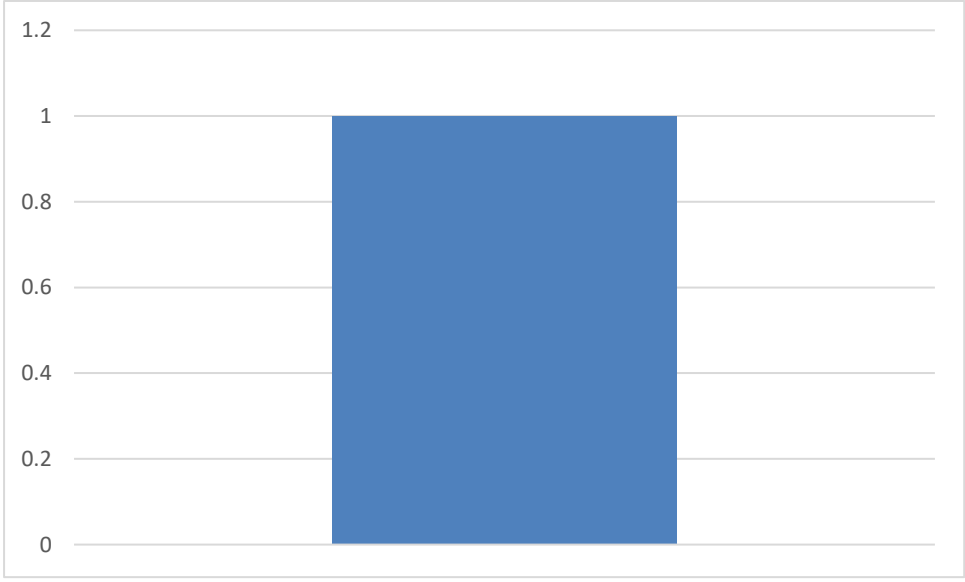

Fig. 3.2 Bar chart of image dataset

As we can see, the data contains a different group of information and need to define vector patterns to serialise and several certain factors to match the data along with the classifications.

3.3 TRAIN AND TEST SPLIT

As we have limited images in the dataset, we have to split the data into three forms, i.e., training, validation and testing.

We have used the train_test_split function to split the dataset. After performing splitting, we get the following results.

Training images = 1000

Validation Images = 600

Testing Images = 60

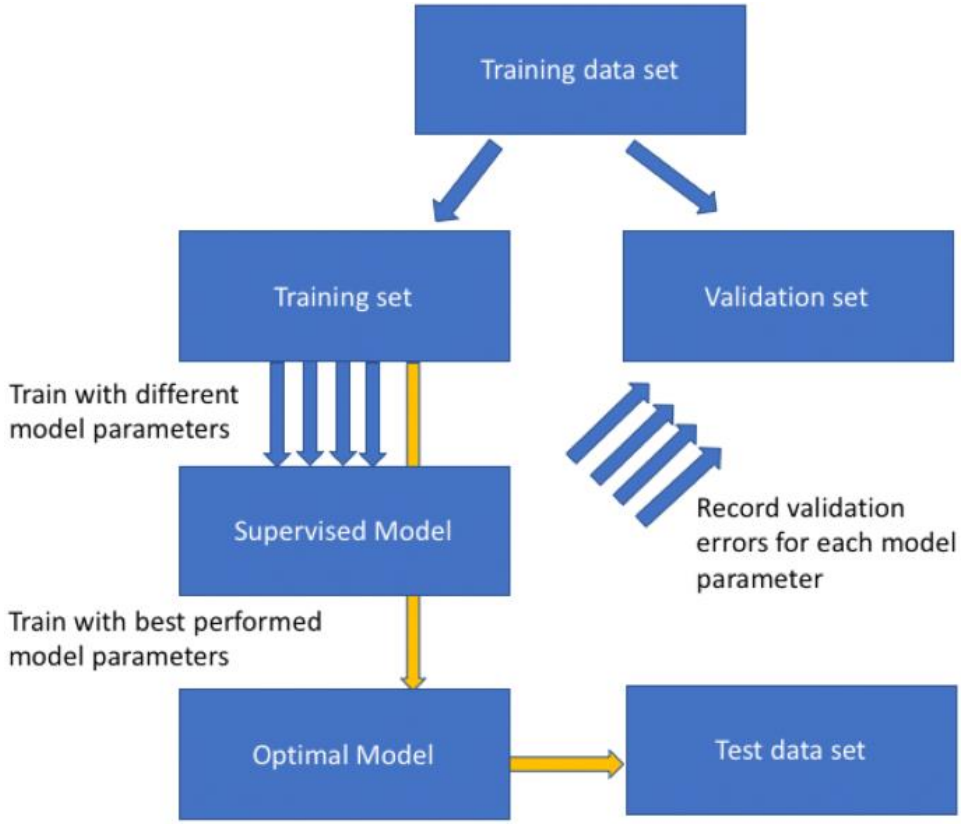

Fig. 3.3 Training and Validation Set

.

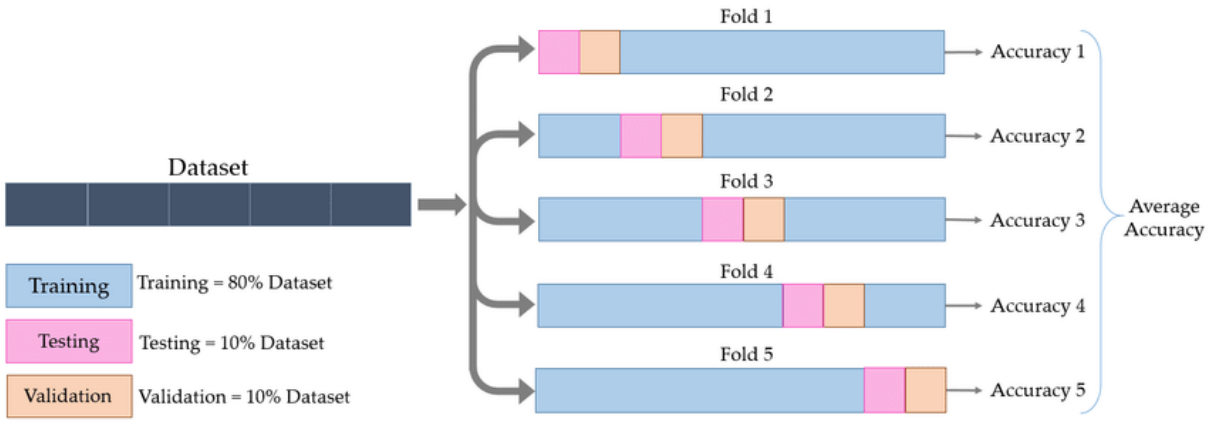

Fig. 3.4 Histogram resembling all the labels in the dataset

3.4 PRE-PROCESSING INFORMATION

In order to process the data set with comparative mapping along the other information and with the relatable classification of the image, a set of methods to process the Data and image to obtain a set of correlations between the data and parsing of same to identify any dissimilarities among them.

3.4.1DATA SIMILARITY AND HASH VALUE GENERATION

A comparative analysis of the given set of data can be made using selective parameters, which are major classifications of classes for the data. This information can classify each record along with their image data to be maintained and matched with their classifications.

The data set with the final parameters to be validated and the identity verification of each data set with the whole collective data to find any similarity which can correlate and identify the similarity and any possible impersonation or duplication of records (Intentionally or unintentionally).

As modern days applications are well-versed in the validation of data sets to avoid any possible duplication and inconsistent data entry into the system. The applications use a number of validation checks to ascertain the valid data records in the system.

Text similarity: As per Neumann[10], the possible similarity between two or more data sets can be identified by observing the similarity score of the following parameters

a. Edit measures

b. Token Based

c. Hybrid

d. Phonetic

e. Domain Dependent

Based on the proposed threshold value from the real data set, we make the decision for the possible duplication of data as to whether it is considered duplicate records or not.

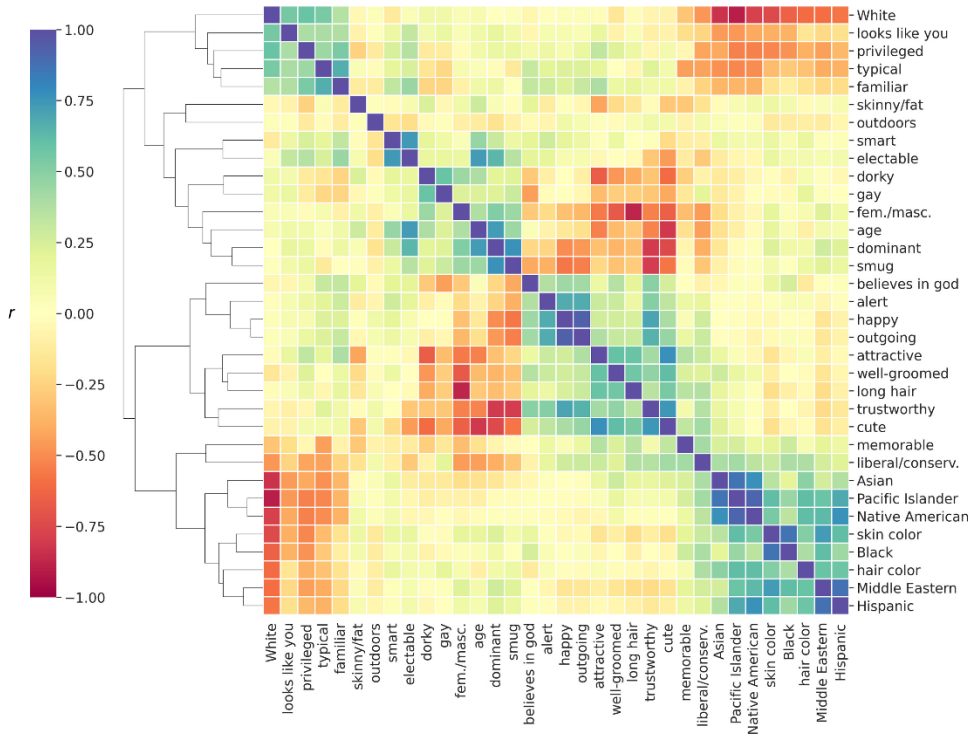

Fig. 3.5 Correlation Matrix

3.5 MODEL BUILDING & TRAINING

In this image classification model, we've used the TensorFlow inception deep learning model in the Learning approach. The TensorFlow model can recognize patterns in images for unsupervised Learning.

ML.NET model can ensure the use of part of it in its pipeline to convert raw images into features or inputs to train a classification model.

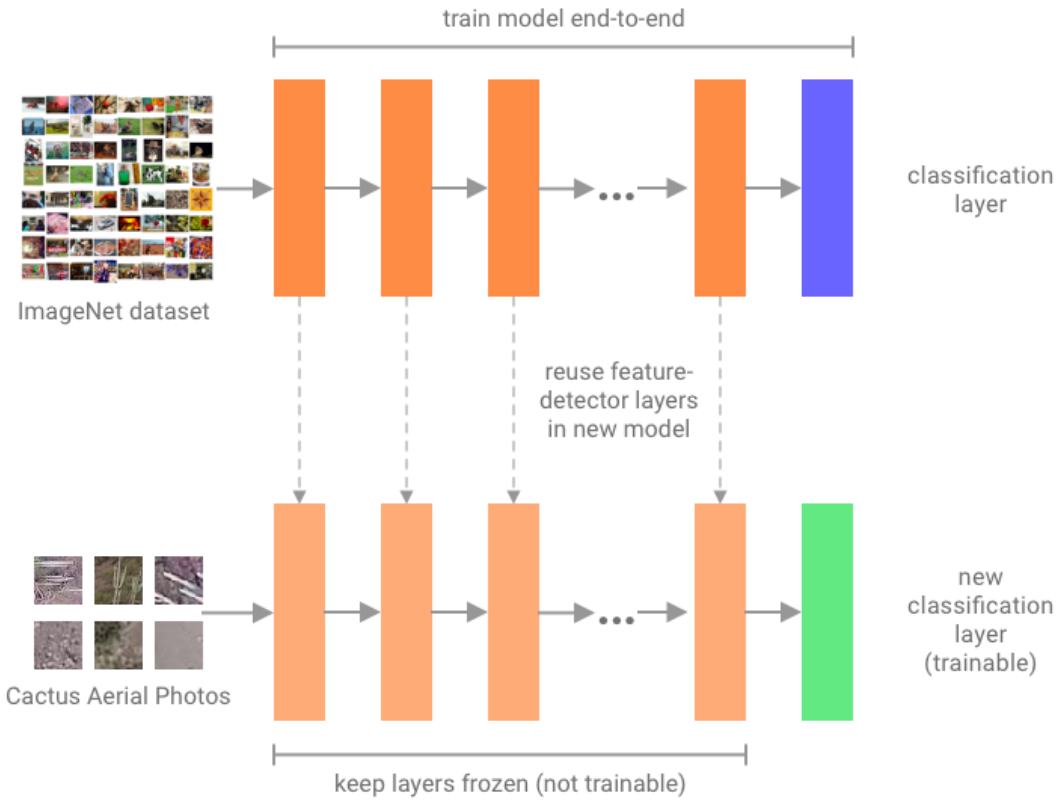

Fig. 3.6 TensorFlow architecture pipeline

The TensorFlow model classifies entire images into nearly thousand classes, although we are using its classification model of human face recognition to identify the images for three major classes only, i.e. a) Male, b) Female, c) invalid.

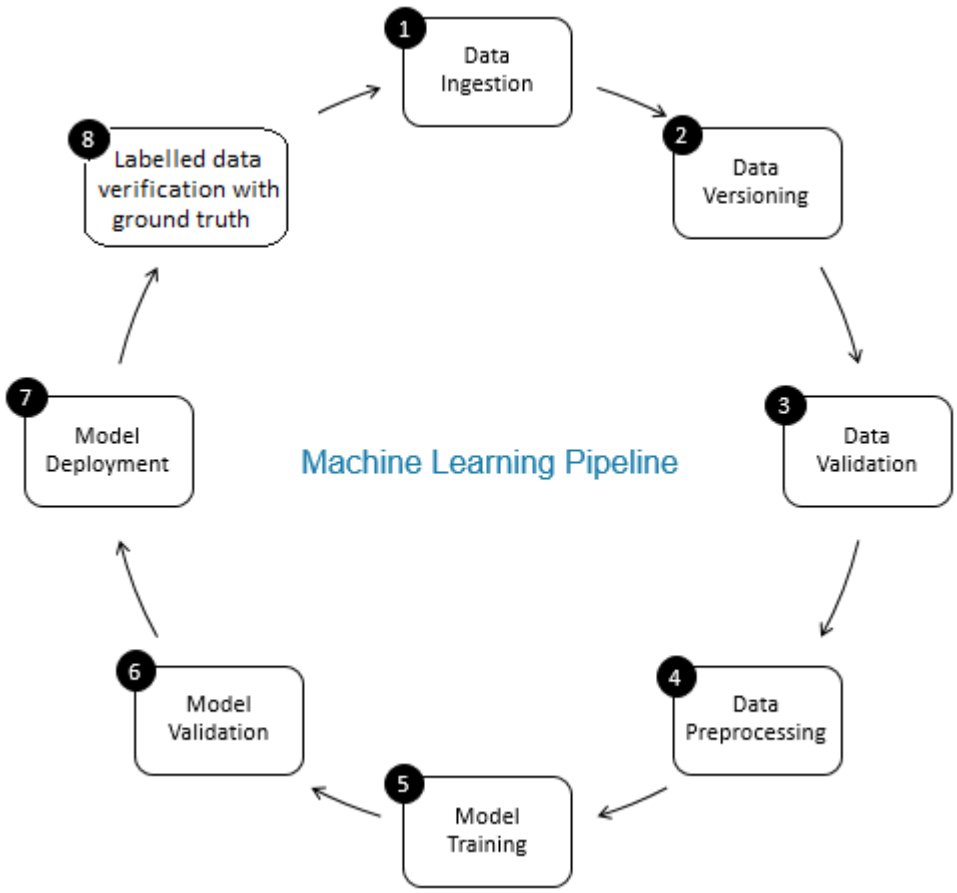

Fig. 3.7 Machine Learning Pipeline

The train function updates the facial parameter of the model to the given data set and similarities on the 80 nodal points and their classification for the given class. The weights and structure of the foremost convolutional layers were fixed, while those of the advanced layers, including the layers anteceding the classifier, were allowed to be fine-tuned.

After training the model on the given parameters, a node analysis was

Total nodes: 80 ,

Trainable nodes: 72 ,

Non-trainable nodes: 8

At this point in time, the accuracy of the model was 95.72 %, so it's balanced to identify the initial classification.

The F1 score is also known as the balanced F-score or F-measure. It's the harmonic mean of precision and recall.

3.5.1 FINE-TUNING MODEL

Metrics for the trained model on a given data set given with the individual Cross-Validation-Result.

The model was optimised using the sample data training", and ultimately correlated with the "Cross-validation" function.

The model can be fine tunes further using Hyperparameter tuning and algorithm to resemble the data set and image classification to identify more parameters for the optimised result.

The model was trained for three classes for 234 images, with an optimal learning rate aimed to minimize cross-entropy and loss.

CHAPTER 4 RESULTS AND ANALYSIS

4.1 RESULTS

Each machine Learning Model is trained and tested on the UNR-IDD dataset. The Accuracy, precision, Recall and F1-Score of these models are as such:

4.1.1 DATA IDENTIFICATION

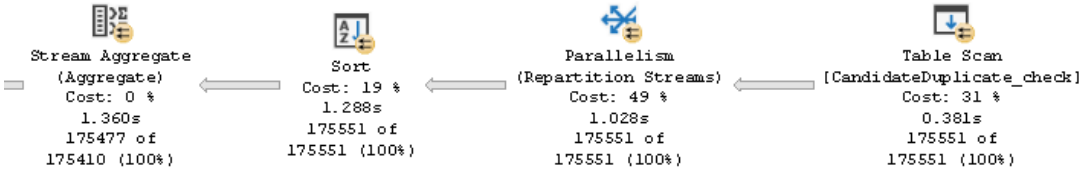

Fig. 4.1 Applying Neumann distancing

Rows	Executes	StmtText	StmtId	NodeId	Parent	LogicalOp	EstimateRows	EstimateIO	EstimateCPU	AvgRowSize	TotalSubtreeCost
53	1	select value4 from [dbo].[CandidateDuplicat	1	1	0		54182.52				18.69875
53	1	|--Parallelism(Gather Streams)	1	2	1	Gather Streams	54182.52	0	0.1417347	105	18.69875
53	8	|--Filter(WHERE:([Expr1003]>(1)))	1	3	2	Filter	54182.52	0	0.02104919	105	18.55702
0	0	|--Compute Scalar(DEFINE:([Expr1003]	1	4	3	Compute Scalar	175409.9	0	0	109	18.53597
175477	8	|--Stream Aggregate(GROUP BY:([SC	1	5	4	Aggregate	175409.9	0	0.04825889	109	18.53597
175551	8	|--Sort(ORDER BY:([SCI22].[dbo].[	1	6	5	Sort	175551	0.002815315	3.509501	105	18.48771
175551	8	|--Parallelism(Repartition Strea	1	7	6	Repartition Streams	175551	0	9.129821	105	14.97539
175551	8	|--Table Scan(OBJECT:([SCI22	1	8	7	Table Scan	175551	5.797277	0.04829615	105	5.845574

Fig. 4.2 Performance Metrics for Data Relations

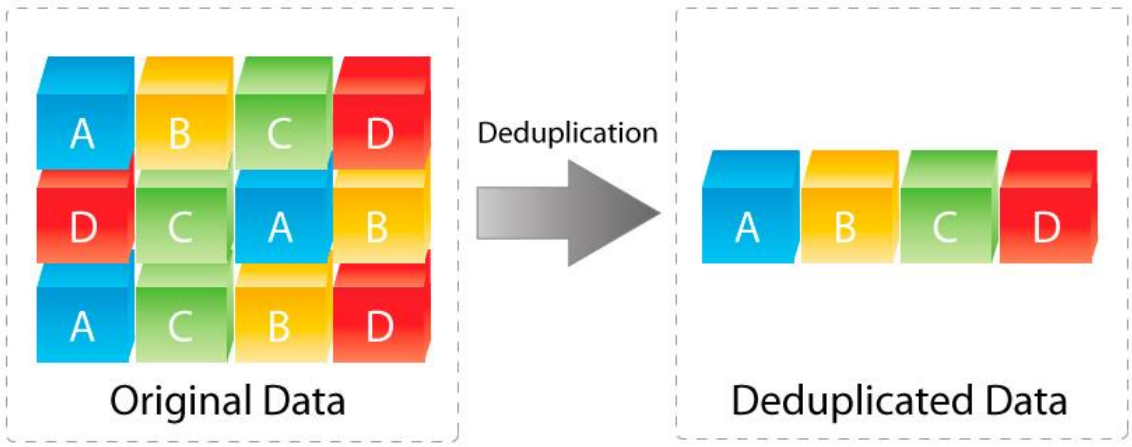

Fig. 4.3 Data De-Duplication

4.1.2 IMAGE PROCESSING

The Trained data set can be used to verify the data symmetry in the data set. Since the data has a cross-validation, it can identify the differences for any data set that can correlate with the possible identification of duplicates.

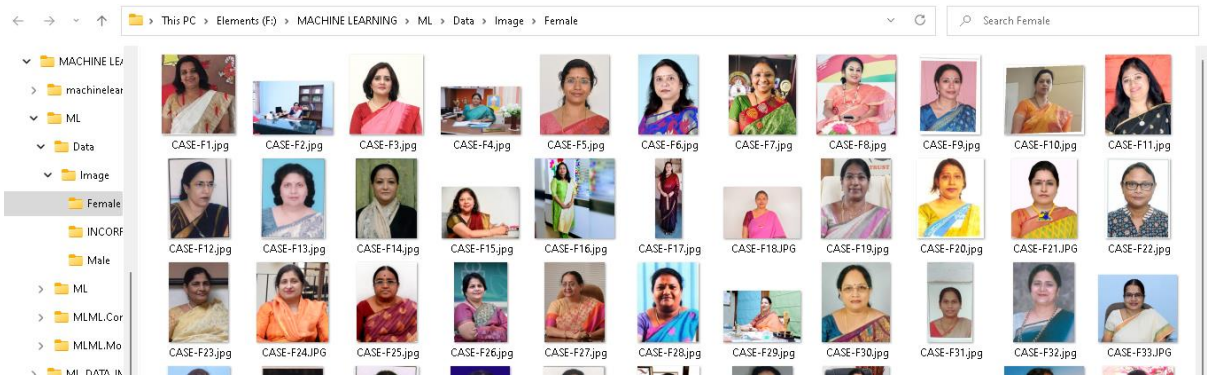

Fig. 4.4 Classification of class: Female

Fig. 4.5 Classification of class: Male

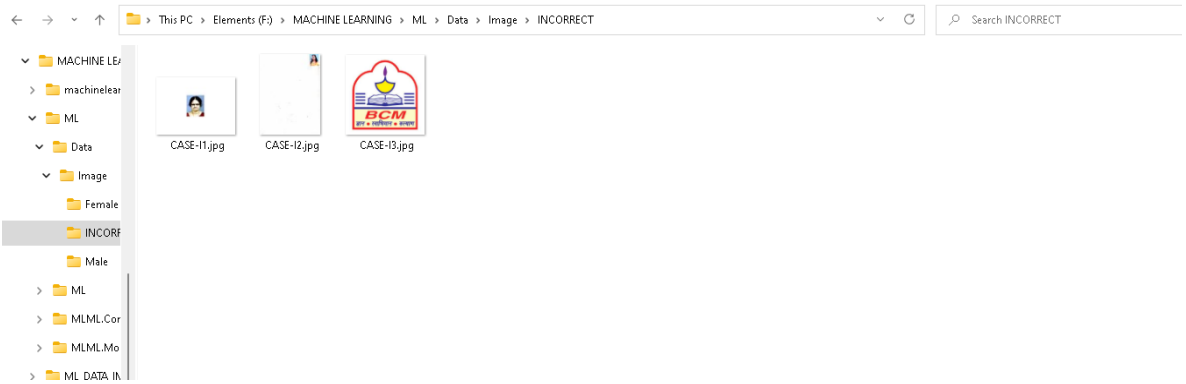

Fig. 4.6 Classification of class: Unclassified

Measure	Value	Derivations
Sensitivity	0.9789	TPR = TP / (TP + FN)
Specificity	0.3750	SPC = TN / (FP + TN)
Precision	0.9789	PPV = TP / (TP + FP)
Negative Predictive Value	0.3750	NPV = TN / (TN + FN)
False Positive Rate	0.6250	FPR = FP / (FP + TN)
False Discovery Rate	0.0211	FDR = FP / (FP + TP)
False Negative Rate	0.0211	FNR = FN / (FN + TP)
Accuracy	0.9592	ACC = (TP + TN) / (P + N)
F1 Score	0.9789	F1 = 2TP / (2TP + FP + FN)
Matthews Correlation Coefficient	0.3539	TP*TN - FP*FN / sqrt((TP+FP)*(TP+FN)*(TN+FP)*(TN+FN))

Fig. 4.7 Confusion Matrix

CHAPTER 5 COMPARATIVE ANALYSIS OF THE PROPOSED APPROACH

5.1TENSORFLOW ANALYSIS

value1	value2	value3	value4	value5	value6	FLG_RMK
YUKTISHARMASURESHKUMARSHARMALEENASHARMA	04041995	1995	AAAAAAAAEEEEHHHHIKKLMMMMNRRRRRSSSSSTUUUY	00144599	0404	D_ADDR
YUKTISHARMASURESHKUMARSHARMALEENASHARMA	25041996	1996	AAAAAAAAEEEEHHHHIKKLMMMMNRRRRRSSSSSTUUUY	01245699	2504	D_ADDR
MUKESHKUMARCHARANSINGHRAMESHWARIDEVI	26081992	1992	AAAAACDEEEGHHHHIIIKKMMMNNRRRRSSSUUVW	01226899	2608	D_ADDR
MUKESHKUMARCHARANSINGHRAMESHWARIDEVI	28081992	1992	AAAAACDEEEGHHHHIIIKKMMMNNRRRRSSSUUVW	01228899	2808	D_ADDR
KISHORMONDALNIRMALMONDALGOURRANIMONDAL	09061996	1996	AAAAADDDGHIIIKLLLLMMMMNNNNNOOOOORRRRSU	00166999	0906	D_ADDR
KISHORMONDALNIRMALMONDALGOURRANIMONDAL	09061995	1995	AAAAADDDGHIIIKLLLLMMMMNNNNNOOOOORRRRSU	00156999	0906	D_ADDR
PINKIGANGARAMKAMLESH	23101993	1993	AAAAEGGHIIKKLMMNNPRS	01123399	2310	D_ADDR
PINKIGANGARAMKAMLESH	29121993	1993	AAAAEGGHIIKKLMMNNPRS	11223999	2912	D_ADDR
AMANBHUSHANCHANDRABHUSHANPRASADABHAPRASAD	05031994	1994	AAAAAAAAAAAABBBCDDDHHHHHHMNNNNPPRRRSSSSUU	00134599	0503	D_PR
AMANBHUSHANCHANDRABHUSHANPRASADABHAPRASAD	05031994	1994	AAAAAAAAAAAABBBCDDDHHHHHHMNNNNPPRRRSSSSUU	00134599	0503	D_PR
RUCHIYADAVGULABSINGHYADAVPARVATIDEVI	30081999	1999	AAAAAAABCDDDEGGHHIIIILNPRRSTUUVVVVYY	00138999	3008	D_PR
RUCHIYADAVGULABSINGHYADAVPARVATIDEVI	30081999	1999	AAAAAAABCDDDEGGHHIIIILNPRRSTUUVVVVYY	00138999	3008	D_PR
RAHULRAMESHCHANDERNIRMALADEVI	20121993	1993	AAAAACDDEEEHHHIILLMMNNRRRRSUV	01122399	2012	D_PR
RAHULRAMESHCHANDERNIRMALADEVI	20121993	1993	AAAAACDDEEEHHHIILLMMNNRRRRSUV	01122399	2012	D_PR

Fig. 5.1 Hash value and Five Pattern Analysis

case	hash	Name	fatherName	motherName	dob	Gender	Image Classification	RMK
C01	195799	NARAYAN BHANJA	MAHESWAR BHANJA	DAMAYANTI BHANJA	07-06-1992	Female	FALSE	DUPLICATE
C01	196873	NARAYAN BHANJA	MAHESWAR BHANJA	DAMAYANTI BHANJA	07-06-1992	Male	TRUE	
C15	154387	JYOTI SHARMA	JAGDISH SHARMA	SUMAN SHARMA	24-10-1992	Male	FALSE	DUPLICATE
C15	165491	JYOTI SHARMA	JAGDISH SHARMA	SUMAN SHARMA	24-10-1992	Female	TRUE	
C17	222982	RUCHI YADAV	GULAB SINGH YADAV	PARVATI DEVI	30-08-1999	Male	FALSE	DUPLICATE
C17	236971	RUCHI YADAV	GULAB SINGH YADAV	PARVATI DEVI	30-08-1999	Female	TRUE	
C20	242601	B MEENAKSHI	K BALAMURUGAN	B LAKSHMI	05-09-1999	Female	TRUE	
C20	206448	B MEENAKSHI	BALAMURUGAN K	LAKSHMI B	05-09-1999	Female	TRUE	DUPLICATE
C27	172935	MADHURI SURENDRA OMKAR	SURENDRA	JYOTY	14-06-1994	Female	TRUE	DUPLICATE
C27	230131	OMKAR MADHURI SURENDRA	SURENDRA	JYOTY	14-06-1994	Female	TRUE	
C28	118911	PINKI	GANGARAM	KAMLESH	29-12-1993	Female	TRUE	
C28	111786	PINKI	GANGARAM	KAMLESH	23-10-1993	Female	TRUE	DUPLICATE
C30	241155	GARIMA	MUKESH	PARVESH	22-06-2000	Male	FALSE	DUPLICATE
C30	258240	GARIMA	MUKESH	PARVESH	22-06-2000	Female	TRUE	

Fig. 5.2 Result on final cluster of data

The overall accuracy for the TensorFlow model is around 90%.

The optimized model was saved and used in a Windows app developed using the ML.NET framework from C#.

An example of a possible identity match is shown in the below fig

CHAPTER 6 CONCLUSION AND FUTURE SCOPE

6.1 CONCLUSION

The Data classification on 05 parameters, along with Image classification methodologies, perform well compared to the methods used in practice. The result of the system is dependent on numerous factors, similar to syntactic feed and image content. In our strategy, a fast and effective system for.

The data set of 175551 records from a DCS was used to test the model.

A total of 107 records were identified as conflicting information from five parameter data analyses.

Further on processing with the image model, a total of 60 records were identified as conflicting and possibly duplicated with falsified information with an intent to duplicate the record, i.e., impersonation or possible human feed error.

The model identified 60 Cases of false candidates based on the parameters, and with verification of the 60 identified cases, the records were maintained only after performing de-duplication on the older entry.

This analysis shows a possible 0.0341781% of inconsistency in the dataset.

6.2 FUTURE SCOPE

Future work can be concentrated on the extraction of exudates by combining different machine learning and deep learning generalities to identify further parameters as:

Age analysis on image classification: Age analysis in image classification has various applications, including facial recognition systems, age estimation for age-restricted content, age-based marketing or personalized advertising, and demographic analysis. The goal is to infer age-related information from images without relying on explicit age annotations. To perform age analysis on image classification, machine learning models are trained using large

datasets of labelled images that contain age annotations or age ranges. Various techniques and algorithms can be applied, such as convolutional neural networks (CNNs) and deep learning, to extract meaningful features from facial images and make predictions about the age of individuals.

Image validation and possible falsified data: authenticity and integrity of an image to ensure that it has not been manipulated or falsified. In today's digital era, with the widespread availability of image editing tools and techniques, it has become increasingly important to validate the integrity of images, especially in applications where trust and accuracy are crucial, such as forensic analysis, journalism, and medical imaging.

Biometric analysis: study and evaluation of biological characteristics or behavioural patterns for the purpose of identification, authentication, and verification of individuals. Biometrics involves the measurement and analysis of unique physical or behavioural traits that can be used to establish a person's identity.

Aadhaar-based analysis: Aadhaar-based analysis involves leveraging the vast amount of data associated with Aadhaar to gain insights, perform statistical analyses, and make informed decisions in various domains. The Aadhaar system collects and stores biometric and demographic data, including fingerprints, iris scans, and personal information such as name, date of birth, and address.

The current model can be further enhanced by building up the dataset from which it learned. Since the current model used a limited size of images, that can be refined.

We can also employ recurrent neural networks (RNN) [5] to make the model learn from its guests. Secondly, rather than just using images, further related information to increase the parameter identification can be made.

REFERENCES

1. Fabien Scalzo; George Bebis; Mircea Nicolescu; Leandro Loss; Alireza Tavakkoli, Research on image classification model based on Feature Fusion Hierarchies for gender classification, 2011

2. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K. and Fei-Fei, L., 2009, June. Imagenet: A large-scale hierarchical image database. In Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference Ieee.

3. LeCun, Y., Bengio, Y. and Hinton, G., 2015. Deep learning. nature, 521(7553).

4. "ImageNet Classification with Deep Convolutional Neural Networks" by Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Published in the Journal of Neural Information Processing Systems (NIPS) 2012.

5. "Deep Residual Learning for Image Recognition" by Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Published in the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2016.

6. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks" by Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Published in the IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) 2017.

7. "Fully Convolutional Networks for Semantic Segmentation" by Jonathan Long, Evan Shelhamer, and Trevor Darrell. Published in the Conference on Computer Vision and Pattern Recognition (CVPR) 2015.

8. "Generative Adversarial Networks" by Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Published in the Advances in Neural Information Processing Systems (NIPS) 2014.

9. "Mask R-CNN" by Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Published in the IEEE International Conference on Computer Vision (ICCV) 2017.

10. "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning" by Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A. Alemi. Published in the AAAI Conference on Artificial Intelligence (AAAI) 2017.

11. "YOLOv3: An Incremental Improvement" by Joseph Redmon and Ali Farhadi. Published in the arXiv preprint arXiv:1804.02767, 2018.

12. "U-Net: Convolutional Networks for Biomedical Image Segmentation" by Olaf Ronneberger, Philipp Fischer, and Thomas Brox. Published in the International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI) 2015.

13. "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition" by Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Published in the European Conference on Computer Vision (ECCV) 2014.

14. "DeepFace: Closing the Gap to Human-Level Performance in Face Verification" by Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Published in the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2014.

15. "Fast R-CNN" by Ross Girshick. Published in the IEEE International Conference on Computer Vision (ICCV) 2015.

16. "DenseNet: Densely Connected Convolutional Networks" by Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Published in the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2017.

17. "Rethinking the Inception Architecture for Computer Vision" by Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Published in the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2016.

18. "VGGNet: Very Deep Convolutional Networks for Large-Scale Image Recognition" by Karen Simonyan and Andrew Zisserman. Published in the International Conference on Learning Representations (ICLR) 2015.

19. "Going Deeper with Convolutions" by Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Published in the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2015.

20. "DeepFace Recognition" by Taehoon Kim, Richard Peng, and Rajeev Ranjan. Published in the arXiv preprint arXiv:1604.02878, 2016.

21. "Deep Learning" by Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Published in the Nature Journal, Volume 521, Issue 7553, 2015.

22. "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks" by Mingxing Tan and Quoc V. Le. Published in the International Conference on Machine Learning (ICML) 2019.

23. "Spatial Transformer Networks" by Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Published in the Advances in Neural Information Processing Systems (NIPS) 2015.