# GENDER IDENTIFICATION BY USING CNN AND SVM A PROJECT REPORT

***Submitted by***

# NITHISH KUMAR .B [REGISTERNO:211422104321] MALLESHVARAN .R [REGISTER NO:211422104265]

***in partial fulfillment for the award of the degree of***

# BACHELOR OF ENGINEERING

**IN**

# COMPUTER SCIENCE AND ENGINEERING



**PANIMALAR ENGINEERING COLLEGE, CHENNAI- 600123.**

(An Autonomous Institution Affiliated to Anna University, Chennai)

# OCTOBER 2024

**BONAFIDE CERTIFICATE**

Certified that this project report **“GENDER IDENTIFICATION BY USING CNN AND SVM”** is the bonafide work “**NITHISH KUMAR.B (211422104321) , MALLESHVARAN.R (211422104265)”** who carried out the project work under my supervision.

# SIGNATURE SIGNATURE

**DR.L.JABASHEELA, M.E.,Ph.D., DR.M.SANGEETHA,M.Tech,Ph.D., HEAD OF THE DEPARTMENT ASSOCIATE PROFESSOR**

# SUPERVISOR

DEPARTMENT OF CSE, DEPARTMENT OF CSE, PANIMALAR ENGINEERING PANIMALAR ENGINEERING COLLEGE, COLLEGE,

NASARATHPETTAI, NASARATHPETTAI,

POONAMALLEE, POONAMALLEE,

CHENNAI-600 123. CHENNAI-600 123.

Certified that the above candidates were examined in the End Semester Project Viva- Voce Examination held on...........................

# INTERNAL EXAMINER EXTERNAL EXAMINER

**DECLARATION BY THE STUDENT**

We NITHISH KUMAR.B (211422104321), MALLESHVARAN.R (211422104265)

hereby declare that this project report titled “**GENDER IDENTIFICATION BY USING CNN AND SVM**”, under the guidance of DR.M.SANGEETHA, M.Tech.,Ph.D., is the original work done by us and we have not plagiarized or submitted to any other degree in any university by us.

# NITHISH KUMAR .B

1. **MALLESHVARAN .R**

# ACKNOWLEDGEMENT

We would like to express our deep gratitude to our respected Secretary and Correspondent **DR.P.CHINNADURAI, M.A., Ph.D.** for his kind words and enthusiastic motivation, which inspired us a lot in completing this project.

We express our sincere and hearty thanks to our Directors

# Tmt.C.VIJAYARAJESWARI, DR.C.SAKTHI KUMAR,M.E.,Ph.D and

**Dr.SARANYASREE SAKTHI KUMAR B.E.,M.B.A.,Ph.D.,** for providing us with the necessary facilities to undertake this project.

We also express our gratitude to our Principal **DR.K.Mani, M.E., Ph.D.** who facilitated us in completing the project.

We thank the Head of the CSE Department, **DR.L.JABASHEELA, M.E.,Ph.D.,**

for the support extended throughout the project.

We would like to thank my Project Guide **DR.M.SANGEETHA, M.Tech.,Ph.D.,** and all the faculty members of the Department of CSE for their advice and encouragement for the successful completion of the project.

# NITHISH KUMAR.B MALLESHVARAN.R

ABSTRACT

This project focuses on developing a gender detection and classification system using fingerprint images. Fingerprints offer a unique biometric trait, widely used in identification and security systems, but the potential for extracting gender information remains relatively underexplored. By leveraging convolutional neural networks (CNNs) and support vector machines (SVMs), this system aims to accurately classify fingerprints based on gender. The dataset is divided into three categories—Easy, Medium, and Hard—each consisting of 300 images, representing different levels of complexity. The model processes fingerprint images to extract relevant features and applies machine learning algorithms to detect subtle patterns linked to gender.

The Easy category aids in the initial training by using clear images, while the Medium and Hard categories introduce challenges, such as image distortions or low-quality prints, to enhance the model's robustness. After feature extraction and model training, performance is evaluated through validation and testing, ensuring accuracy across different levels of image quality. This system demonstrates potential real-world applications in forensic science, security, and biometric analysis, providing an efficient, non-invasive method for gender classification. The project's structured dataset and multi-layered approach contribute to the development of a reliable, scalable model, improving the versatility of fingerprint-based biometric systems

# TABLE OF CONTENTS

**CHAPTER NO. TITLE PAGE NO.**

|  |  |
| --- | --- |
| **ABSTRACT**  **LIST OF FIGURES**  **1. INTRODUCTION** | vi viii  1 |
| 1.1 Overview | 1 |
| 1.2 Problem Definition | 2 |
| **2. LITERATURE SURVEY** | 3 |
| **3. SYSTEM ANALYSIS** | 6 |
| 3.1 Existing System | 6 |
| 3.2 Proposed System | 6 |
| 3.3 Feasibility Study | 7 |
| 3.4 Development Environment | 8 |
| **4. SYSTEM DESIGN** | 9 |
| 4.1 Flow diagram | 9 |
| 4.2 KAGGLE Dataset | 10 |
| 4.3 Image Data Preprocessing | 10 |
| 4.4 Fingerprint Pattern Preparation | 10 |
| 4.5 Dataset Details  **5. SYSTEM ARCHITECTURE** | 11  12 |
| 5.1 Architecture Overview | 12 |
| 5.2 Modules | 13 |
| 5.3 Algorithms | 17 |
| **6. SYSTEM IMPLEMENTATION** | 21 |

1. PERFORMANCE ANALYSIS 28
   1. [DATA PREPARATION 28](#_TOC_250005)
   2. [SVM CLASSIFICATION 28](#_TOC_250004)
   3. [TESTING 29](#_TOC_250003)
   4. OBSERVATION OF RESULT 32
2. CONCLUSION 35

[APPENDICES 36](#_TOC_250002)

[A.1 Sample Screenshots 36](#_TOC_250001)

[REFERENCES 37](#_TOC_250000)

# LIST OF FIGURES

**Figure No. Figure Title Page No.**

* 1. Working flow of the Model 9
  2. Fingerprint pattern preparation 11
  3. Architecture Overview 12
  4. Final Model Structure 16
  5. Convolutional Neural Network 18
  6. Max Pooling Layer 20

7.1

7.2

7.3

7.4

7.5

7.6

7.7

7.8

A1

Test Image 1 29

Test Image 2 29

Efficient Dataset 30

Usage Of SVM+CNN model compared with the 30

other ML algorithms

Efficient Algorithm 31

Confusion Matrix Test Set 31

Confusion Matrix Train set 32

Accuracy Graph 33

Sample Screenshots 36

# CHAPTER 1 INTRODUCTION

* 1. **OVERVIEW**

This project focuses on developing a system for **gender classification based on fingerprint images** using a hybrid model that combines **Convolutional Neural Networks (CNN)** and **Support Vector Machines (SVM)**. Fingerprints, due to their uniqueness and permanence, offer a reliable biometric trait for gender classification, which can enhance the efficiency of identification systems used in security, forensics, and healthcare. The primary objective is to train a CNN to extract meaningful features from fingerprint images, such as ridge flow patterns, and then use an SVM to classify the gender into male or female based on these features. The system begins with preprocessing, where fingerprint images are resized, enhanced, and normalized to ensure consistent input. CNN is employed for automatic feature extraction, leveraging convolutional and pooling layers to capture critical spatial hierarchies from the fingerprint data. The extracted features are then passed to the SVM, a powerful binary classifier, to accurately separate the male and female categories. The project utilizes Python with libraries like TensorFlow/Keras for CNN, Scikit-learn for SVM, and OpenCV for image processing. The system is trained and tested using fingerprint datasets, aiming to achieve high classification accuracy. Such a system has practical applications in enhancing fingerprint-based security systems, assisting forensic investigations, and supporting healthcare with gender-specific analysis. Ultimately, this project provides a non-intrusive, efficient, and accurate method for gender prediction using biometric data.

Real-time applications of gender classification based on fingerprints:

* + 1. Security and Access Control Systems
    2. Forensic Investigation
    3. Criminal Profiling
    4. Healthcare Biometrics
    5. Biometric Identification Systems
    6. Smartphone Authentication
    7. Gender-Specific Marketing
    8. Border Control and Immigration Systems
    9. Time and Attendance Systems 10.Personalized Public Services

# PROBLEM DEFINITION

The problem addressed by this project is the need for an accurate, non-intrusive method to classify gender based on biometric data, specifically fingerprints, in various identification systems. Traditional methods of gender recognition often rely on facial, voice, or physiological traits, which can be prone to errors or require user cooperation. Fingerprints, being unique and widely available in biometric databases, offer an alternative for gender classification. However, manual or traditional techniques for fingerprint analysis may lack the efficiency and accuracy needed for real-time applications. Therefore, this project aims to solve the problem by developing a hybrid model that utilizes **Convolutional Neural Networks (CNN)** for feature extraction and **Support Vector Machines (SVM)** for classification, to automatically determine the gender of an individual based solely on their fingerprint images.

# CHAPTER 2 LITERATURE SURVEY

There are numerous works has been done related to disease prediction systems using different data mining techniques and machine learning algorithms in medical centre machine learning algorithms in medical centres.

Anju Narayanan, [1] 2019 paper on gender detection from fingerprints focuses on **pixel count analysis**. Preprocessing is performed to enhance the fingerprint ridges, followed by calculating pixel intensity values in these ridges. The study assumes that male fingerprints have denser ridge patterns, reflected in higher pixel counts, while female fingerprints exhibit lower pixel counts. These pixel-based features are extracted and fed into classifiers like **SVM** and **KNN** for gender classification.

J.Serin [2] 2023, the methodology for **gender classification from fingerprints** uses a **hybrid CNN-SVM approach**. The process begins with preprocessing the fingerprint images to enhance their quality. A **Convolutional Neural Network (CNN)** is then employed to automatically extract deep features from the images, capturing intricate ridge patterns. These extracted features are then fed into a **Support Vector Machine (SVM)** classifier for gender prediction.

Prof. Aisfwarya [3] 2019 Fingerprint-based gender classification explores various approaches that leverage biometric data for gender identification. Recent advancements employ machine learning and deep learning models, particularly Convolutional Neural Networks (CNNs), for more accurate classifications. Transfer learning and hybrid models combining CNNs with Recurrent Neural Networks (RNNs) have shown promising results, improving performance by learning gender-specific patterns from fingerprint images.

LamiaBerriche [4] 2022 paper by Lamia Berriche, the methodology for **fingerprint- based gender identification** involves a **comparative study** of various machine learning techniques. The process starts with the preprocessing of fingerprint images to improve ridge detail. Multiple feature extraction methods, such as **ridge density** and **minutiae point analysis**, are compared to identify gender-specific patterns. Several classifiers, including **Support Vector Machine (SVM)**, **K-Nearest Neighbors (KNN)**, and **Random Forest**, are evaluated for their performance in gender classification.

P.Gnanasivam R.Vijayarajan [5], 2019 paper on **gender classification from fingerprint ridge count and fingertip size using optimal score assignment**, the methodology focuses on analyzing two key features: **ridge count** and **fingertip size**. The process begins with the extraction of ridge patterns and measuring the fingertip dimensions from fingerprint images. An **optimal score assignment algorithm** is applied to assign scores based on these features, reflecting the observed differences between male and female fingerprints. Ridge counts tend to be higher for males, while fingertip sizes vary by gender. A classifier then uses these scores for gender prediction, showing effective results in gender classification.

Anjali Mishra [6] 2022 The literature on gender identification using fingerprints highlights various methodologies, evolving from traditional statistical techniques to modern machine learning approaches. Early studies relied on ridge density, fingerprint type, and minutiae-based analysis to distinguish genders. In recent years, machine learning models, particularly CNNs, have significantly improved classification accuracy by automating feature extraction. The integration of deep learning, transfer learning, and hybrid models has further enhanced the precision of gender identification, reflecting a growing trend toward data-driven solutions in biometrics.

G. Jayakalaa, Dr. L.R.Sudha [7] 2021 literature on gender classification based on fingerprint analysis has evolved from basic feature extraction techniques, such as ridge count and minutiae-based methods, to advanced machine learning models. Early approaches focused on manual feature identification, but recent developments employ deep learning, particularly CNNs, to automatically extract gender-specific features from fingerprint images. Studies show that combining fingerprint ridge characteristics with deep learning techniques leads to higher accuracy, making CNNs and transfer learning popular methods for enhancing the performance of gender classification systems.

Vaishnavi Y. Mali Dr. Babasaheb G. Patil [8] 2019 The literature on human gender classification using machine learning highlights a shift from traditional fingerprint- based methods to more sophisticated algorithms. Early studies focused on extracting features like ridge count and fingerprint patterns, while recent approaches utilize machine learning techniques such as support vector machines (SVM) and convolutional neural networks (CNNs). These models automate feature extraction, improving accuracy in gender classification tasks. The use of deep learning frameworks and hybrid models has also been explored, significantly enhancing performance by enabling the system to learn complex patterns from large datasets.

# CHAPTER 3

**SYSTEM ANALYSIS**

# EXISTING SYSTEM

The existing system for **gender classification from fingerprints** typically involves traditional biometric methods that rely on manual feature extraction techniques, such as ridge counting, minutiae-based analysis, and texture patterns. These methods analyze the ridge and valley structures in fingerprints to differentiate between male and female traits. However, they are often limited by human error, image quality, and the inability to capture complex patterns that may vary between individuals. In some systems, machine learning classifiers like Support Vector Machines (SVM) or k- Nearest Neighbors (k-NN) are applied to features extracted through these traditional methods to improve accuracy. However, these systems generally lack the ability to adapt to variations in fingerprints due to noise, distortion, or partial prints, which can result in lower classification performance. Furthermore, the computational efficiency of these systems is often insufficient for real-time applications, and they do not leverage the full potential of deep learning models, which have shown superior results in more recent research.

# PROPOSED SYSTEM

The **proposed system** aims to improve gender classification accuracy from fingerprint images by leveraging deep learning techniques and Support Vector Machines (SVM). The system utilizes **Convolutional Neural Networks (CNN)** for automated feature extraction from fingerprint images, eliminating the need for manual feature extraction, which is error-prone and time-consuming in traditional

System. The CNN model processes grayscale fingerprint images and extracts significant spatial features, followed by max-pooling layers to reduce dimensionality. After flattening the features, they are passed to a dense layer for further refinement. Once the CNN has learned robust features, the output from the dense layer is used to train a Support Vector Machine (SVM) classifier to improve the final classification. The system integrates CNN's automated feature extraction capability with SVM’s high generalization ability for binary classification tasks like gender detection. This hybrid approach aims to overcome limitations of existing systems such as sensitivity to noise, low accuracy, and inefficiency in real-time processing. Moreover, dropout layers are introduced to reduce overfitting, and early stopping is used to optimize training time, ensuring a more reliable and efficient model. Overall, the proposed system is expected to offer improved accuracy, better handling of image variations, and faster performance compared to traditional methods

# FEASIBILITY STUDY

The objective of feasibility study is not only to solve the problem but also to acquire a sense of its scope. During the study, the problem definition was crystallized and aspects of the problem to be included in the system are determined. Consequently, benefits are estimated with greater accuracy at this stage. The key considerations are:

# EconimicFeasibility

Economic feasibility studies not only the cost of hardware, software are included but also the benefits in the form of reduced costs are considered here. This project, if installed will certainly be beneficial since there will be reduction in manual work and increase in the speed of work.

# Technical Feasibility

Technical feasibility evaluates the hardware requirements, software technology,

available personnel etc., as per the requirements it provides sufficient memory to hold and process.

* + - 1. Machine Learning algorithm – CNN and SVM
      2. Artificial Intelligence - LSTM
      3. Google Drive
      4. IDE : SPYDER

# Social Feasibility

Ethical concerns must be /considered, such as the potential misuse of gender classification technology in a way that could reinforce gender biases or invade privacy. Ensuring the system complies with data privacy regulations and anti-discrimination policies is crucial. The project should address these concerns by securing data, anonymizing personal information, and adhering to ethical AI guidelines.

# Development Environment

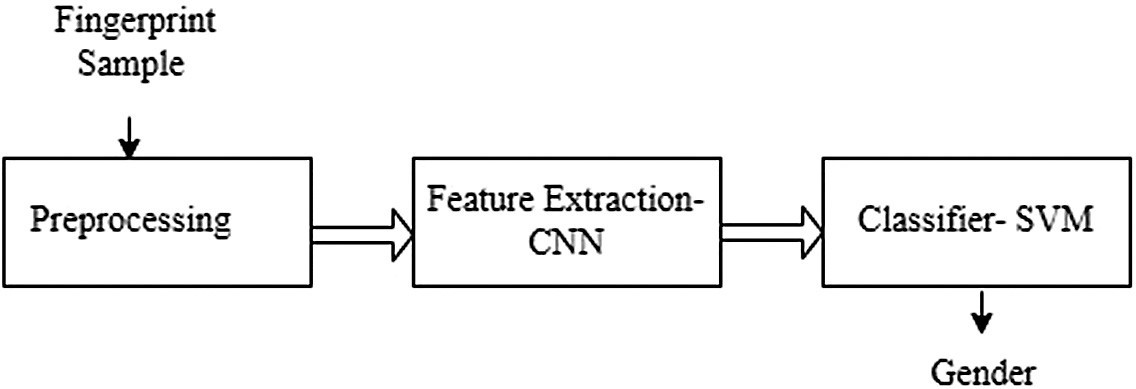
* + 1. **Hardware Requirments** Processor : Intel Core i5 RAM : 512 MB and above Hard Disk :40 GB and above.

# CHAPTER 4

**SYSTEM DESIGN**

# FLOW DIAGRAM

The dataset used for the **"Gender Classification Based on Fingerprint Using CNN and SVM"** project contains grayscale fingerprint images labeled with gender (male or female), hand orientation (left or right), and finger type (thumb, index, middle, ring, or little). It includes both original and altered fingerprints with varying levels of difficulty (easy, medium, hard), providing a robust set for training and testing models. This dataset supports real-world applications by simulating various fingerprint conditions and distortions, making it ideal for machine learning tasks in gender classification.



# Fig. 4.1 Working flow of the model

* 1. **KAGGLE DATASET**

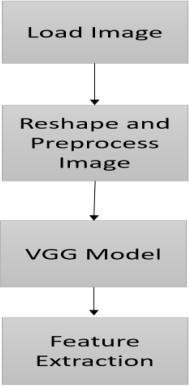
The dataset is structured into Easy, Medium, and Hard categories, each containing 6,000 fingerprint images for training. The Easy category features clear prints to help the model learn fundamental patterns for gender classification. Medium introduces more challenging images with some quality variations, enhancing the model's ability to handle imperfect prints. The Hard category contains the most complex images, including distortions or partial prints, pushing the model to its limits. This tiered approach ensures the model progressively learns and generalizes effectively, improving its performance in real-world fingerprint-based gender classification tasks.

# IMAGE DATA PREPARATION

For image data preparation, the dataset is divided into three categories: Easy, Medium, and Hard, each with 6,000 images. Fingerprint images undergo pre-processing steps such as resizing for uniform dimensions, normalization to ensure consistent pixel intensity, and augmentation to introduce variations like rotation or noise. This helps improve the model's ability to handle diverse prints during training. Data is split into training, validation, and testing sets to ensure model performance is assessed across all complexity levels. These steps ensure a well-prepared, diverse dataset for effective training and generalization in fingerprint-based gender classification.

# FINGERPRINT PATTERN PREPARATION

KAGGLE dataset contains multiple patterns of fingerprint. In the data preparation phase, each image id is taken as key and its corresponding pattern are stored as values in a dictionary.



# Fig.4.2 Fingerprint pattern Preparation

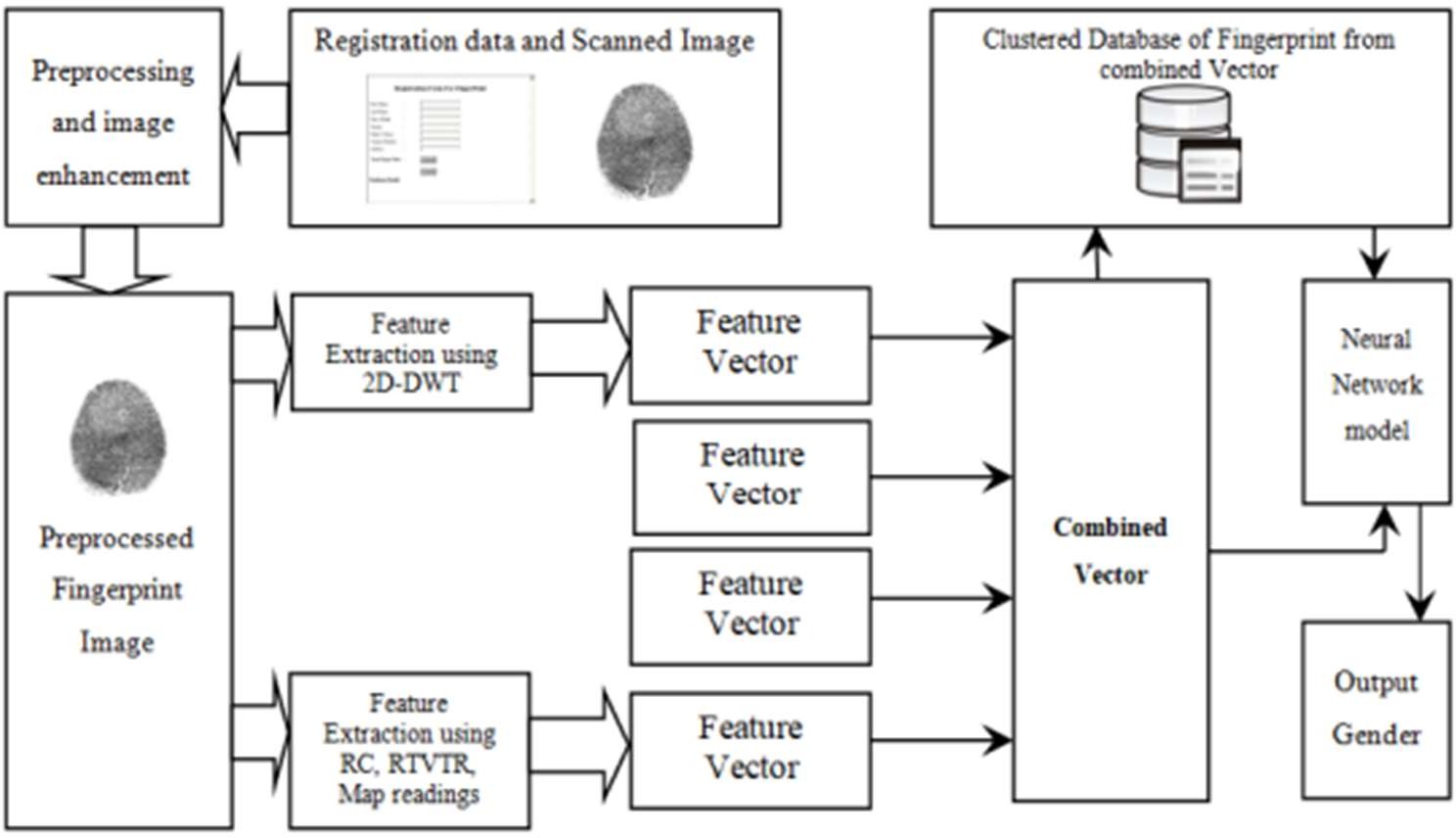
* 1. **Dataset Details**

The dataset for this project is sourced from the **SOCOFing** dataset, available on Kaggle. It contains 6,000 fingerprint images categorized by gender, along with features like finger type and age. The dataset includes synthetic fingerprint images with varying levels of distortion, designed to simulate real-world conditions. Each fingerprint is labeled, making it suitable for classification tasks. The dataset’s diverse image quality, including clear and distorted prints, supports training models to handle a wide range of fingerprint characteristics, which is essential for gender classification in practical applicatio

# CHAPTER 5 SYSTEM ARCHITECTURE

* 1. **ARCHITECTURE OVERVIEW**

This displays the whole design of our working model including the components and the states occurring during the execution of the process. It displays the very initial process of image feeding followed by the parsing and breaking down of the image into vectors, where all the data regarding the image is stored and fed to the model by using Flickr dataset. The CNN is used in the encoding and STM is used in the decoding the descriptive data which play the image again and aging develop the caption with the help of language processing and data(trained) stored, thus provide generated caption as output.



# Fig.5.1 Architecture overview

* 1. **MODULES**

# 5.2.1 Data Preprocessing — Images

In this project, fingerprint images serve as the input (X) to our model. As with any machine learning model, the input needs to be transformed into a suitable format that can be processed. Since neural networks require fixed-sized vectors, we need to convert each image into a fixed-length vector. For this task, we leverage **Convolutional Neural Networks (CNN)** to extract meaningful features from the images. Specifically, the **Xception** model, pre-trained on the **ImageNet** dataset, is used for transfer learning to generate feature vectors for each fingerprint image. This process, known as **automatic feature engineering**, ensures that the most relevant features of the fingerprint are captured for gender classification.

# 5.2.2 Data Preprocessing — Captions

Here, gender labels act as the target variable (Y) that our model learns to predict. During training, the CNN will attempt to map each fingerprint image to its corresponding gender (either **male** or **female**). These labels are extracted from the filenames of the fingerprint images and then encoded into a format suitable for binary classification.

# 5.2.3 Data Preparation using Generator Function

The generator function prepares batches of image and gender pairs for the model. For instance, the first image vector, Image\_1, might correspond to a fingerprint from a male. The task is to predict the gender based on this vector. A typical input-output pair might look like this:

* + - * **Input**: Image\_1 + score ‘3’
      * **Output**: ‘male’

This process repeats for each image, and the generator ensures that data is efficiently fed into the model in manageable batches.

# 5.2.4 Pre-Requisites

This project requires good knowledge of Deep Learning, Python, working with Colab notebooks, Keras library, Numpy, and Natural Language Processing. Make sure you have installed all the following necessary libraries:

* + - * tensorflow
      * keras
      * pandas
      * numpy
      * opencv
      * IDE : spyder

# 5.2.5 Project File Structure

Downloaded from dataset**:**

* + - * **Dataset Folder:** Contains 6000 fingerprint images.
      * **Models Folder:** Stores the trained gender classification models.
      * **Features.p:** A pickle file containing the feature vectors extracted from the InceptionV3 model for each fingerprint.
      * **Model.png:** A visual representation of the neural network architecture.
      * **testing\_gender\_classifier.py**: Python script for generating gender

predictions for new fingerprint images.

* + - * **training\_gender\_classifier.ipynb:** Spyder IDE in anaconda where the model is trained on the dataset

# 5.2.6 Extracting The Feature Vector From All Images

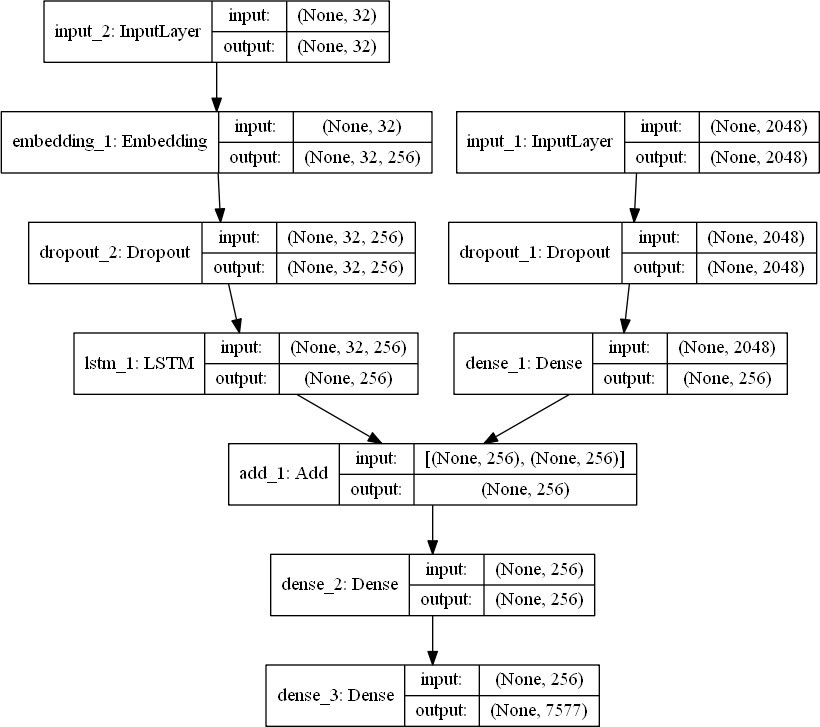
In our gender classification project based on fingerprint images, we utilize transfer learning to extract features using the InceptionV3 model. This pre-trained model, originally trained on the ImageNet dataset, is well-suited for our task as it can capture essential characteristics from fingerprint images. To begin, we ensure that the input fingerprint images are resized to **299x299 pixels**, as required by the InceptionV3 model. By removing the final classification layer, we can access the **2048-dimensional feature vector** generated by the model, which serves as a condensed representation of the image's content. Once the feature vectors are extracted, they are stored in a dictionary that maps each fingerprint image to its respective feature vector. This dictionary is then saved as a **pickle file** for future use, allowing us to avoid the time-consuming process of re-extracting features each time the model is run.

# 5.2.7 Tokenizing The Fingerprint

Computers don't understand Fingerprint patterns, thus it is represented in numbers. So, each patterns in the fingerprint is mapped with a unique index value. Keras library provides us with the tokenizer function is used to create tokens from our vocabulary and save them to a **“tokenizer.p”** pickle file. Our dataset contains 6000+ patterns. These tokens are then compared with the test image and the optimal solution is then given (ie.) male or female

# 5.2.8 Training the model

To train the model, 6000 training images for generating the input and output sequences is retrieved in batches and fitting them to the model using model.fit\_generator() method. Then save this model to a folder, this task might take time to save.



# Fig.5.2. Final Model Structure

* 1. **ALGORITHMS**

# Convolutional Neural Network

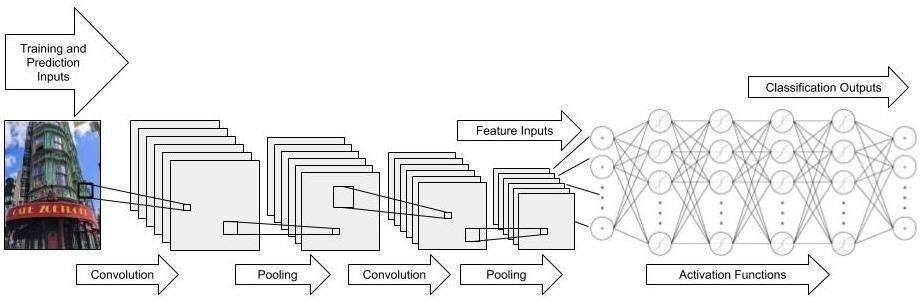
Artificial Neural Networks are used in various classification tasks like image, audio, words. Different types of Neural Networks are used for different purposes, for

example for predicting the sequence of words we use Recurrent Neural Networks more precisely an LSTM, similarly for image classification Convolution Neural networks is used.

* A convolutional neural network, or CNN, is a deep learning neural network sketched for processing structured arrays of data such as portrayals.
* CNN are very satisfactory at picking up on design in the input image, such as lines, gradients, circles, or even eyes and faces.
* This characteristic that makes convolutional neural network so robust for computer vision.
* CNN can run directly on a underdone image and do not need any preprocessing.
* A convolutional neural network is a feed forward neural network, seldom with up to 20.
* The strength of a convolutional neural network comes from a particular kind of layer called the convolutional layer.
* CNN contains many convolutional layers assembled on top of each other, each one competent at recognizing more sophisticated shapes.
* With three or four convolutional layers it is viable to recognize

handwritten digits and with 25 layers it is possible to differentiate human faces.

* The agenda for this sphere is to activate machines to view the world as humans do, perceive it in a similar fashion and even use the knowledge for a multitude of duties such as image and video recognition, image inspection and classification, media recreation, recommendation systems, natural



language processing, etc.

# Fig.5.3 Convolutional Neural Network

The output from multiplying the filter with the input array one time is a single value. As the filter is applied multiple times to the input array, the result is a two- dimensional array of output values that represent a filtering of the input. As such, the two- dimensional output array from this operation is called a “*feature map*“.

Once a feature map is created, each value is passed in the feature map through a nonlinearity, such as a ReLU, much like we do for the outputs of a fully connected layer.

# POOLING LAYER

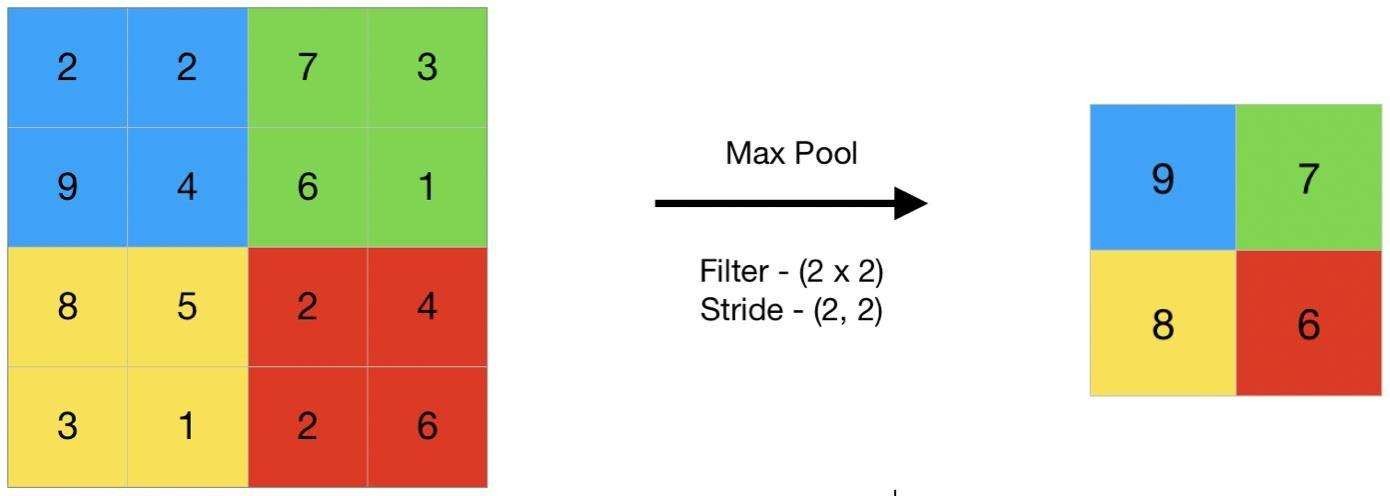
A pooling layer is a new layer added after the convolutional layer. Specifically, after a nonlinearity (e.g. ReLU) has been applied to the feature maps output by a convolutional layer.

# Max Pooling Layer

In this fingerprint-based classification project, Max Pooling plays a crucial role in the feature extraction process within the Convolutional Neural Network (CNN) architecture. Max Pooling is a down-sampling technique that reduces the spatial dimensions of feature maps while preserving the most important information. By selecting the maximum value from small regions (e.g., 2x2 or 3x3 grids) in each feature map, Max Pooling helps the network become more computationally efficient and robust to variations such as shifts and rotations in fingerprint images.

Incorporating Max Pooling layers after convolutional layers allows the model to progressively focus on high-level patterns by reducing the dimensionality of the data while retaining critical features. This enables the network to efficiently learn important fingerprint characteristics such as ridge flow, minutiae points, and texture, which are vital for accurate classification tasks such as gender prediction. It also helps prevent overfitting by reducing the complexity of the model.

The results are down sampled or pooled feature maps that highlight the most present feature in the patch, not the average presence of the feature in the case of average pooling. This has been found to work better in practice than average pooling for computer vision tasks like image classification.



**Fig.5.4 Max Pooling Layer**

**5.3.3 LSTM:**

In this fingerprint-based project, Long Short-Term Memory (LSTM) networks can be utilized to improve the feature extraction and classification process. LSTMs are a type of recurrent neural network (RNN) that excel in capturing long-term dependencies in sequential data. In this context, fingerprint images can be treated as sequential data after applying a preprocessing step, such as converting them into a sequence of pixel values or feature vectors.

By feeding the extracted features into an LSTM, the model can learn to capture subtle temporal patterns in the spatial structure of the fingerprint. These patterns can be particularly useful for distinguishing between different classes (e.g., gender, identity). When combined with a CNN for feature extraction and transfer learning, the LSTM adds an extra layer of robustness to the model, helping it better understand variations in fingerprints. Ultimately, this hybrid approach enhances classification accuracy and performance.

# CHAPTER 6 SYSTEM IMPLEMENTATION

**CODING**:

import numpy as np # linear algebra

import pandas as pd # data processing, CSV file I/O (e.g. pd.read\_csv) import os

for dirname, \_, filenames in os.walk('/kaggle/input'): for filename in filenames:

print(os.path.join(dirname, filename)) import numpy as np

import seaborn as sns import tensorflow as tf import os

import cv2 import random

import matplotlib.pyplot as plt

from tensorflow.keras.models import Sequential,Model

from tensorflow.keras.layers import Dense, Dropout,Activation, Flatten, Conv2D, MaxPool2D from tensorflow.keras.utils import to\_categorical # convert to one-hot-encoding

ReduceLROnPlateau

from sklearn.model\_selection import train\_test\_split def extract\_label(img\_path,train = True):

filename, \_ = os.path.splitext(os.path.basename(img\_path)) subject\_id, etc = filename.split(' ')

if train:

gender, lr, finger, \_, \_ = etc.split('\_')

else:

gender, lr, finger, \_ = etc.split('\_') gender = 0 if gender == 'M' else 1

lr = 0 if lr == 'Left' else 1 if finger == 'thumb':

finger = 0

elif finger == 'index': finger = 1

elif finger == 'middle': finger = 2

elif finger == 'ring': finger = 3

elif finger == 'little': finger = 4

return np.array([subject\_id, gender, lr, finger], dtype=np.uint16) img\_size = 96

def load\_data(path,train): print("loading data from: ",path) data = []

for img in os.listdir(path): try:

img\_array = cv2.imread(os.path.join(path, img), cv2.IMREAD\_GRAYSCALE) img\_resize = cv2.resize(img\_array, (img\_size, img\_size))

label = extract\_label(os.path.join(path, img),train) data.append([label[1], img\_resize ])

except Exception as e: pass

data return data

Real\_path = "C:/Users/nithishkumar/Desktop/SOCOFing/SOCOFing/Real"

Easy\_path = "C:/Users/nithishkumar/Desktop/SOCOFing/SOCOFing/Altered/Altered-Easy"

Medium\_path = "C:/Users/nithishkumar/Desktop/SOCOFing/SOCOFing/Altered/Altered- Medium"Hard\_path = "C:/Users/nithishkumar/Desktop/SOCOFing/SOCOFing/Altered/Altered-Hard" easy\_data = load\_data(Easy\_path, train = True)

medium\_data = load\_data(Medium\_path, train = True) hard\_data = load\_data(Hard\_path, train = True)

test = load\_data(Real\_path, train = False)

data =np.concatenate([easy\_data,medium\_data,hard\_data],axis=0) X, y = [], []

for label, feature in data: X.append(feature) y.append(label)

del data

X = np.array(X).reshape(-1, img\_size, img\_size, 1)

X = X / 255.0

y =np.array(y) # In[9]:

X\_test, y\_test = [], [] for label, feature in test:

X\_test.append(feature) y\_test.append(label)

del test

X\_test = np.array(X\_test).reshape(-1, img\_size, img\_size, 1)

X\_test = X\_test / 255.0 y\_test = np.array(y\_test) # In[10]:

X\_train, X\_val, y\_train, y\_val = train\_test\_split(X, y, test\_size=0.2, random\_state=1) print("full data: ",X.shape)

del X, y

print("Train: ",X\_train.shape) print("Validation: ",X\_val.shape) print("Test: ",X\_test.shape)

# In[12]:

model = Sequential()

model.add(Conv2D(filters = 32, kernel\_size = (5,5),padding = 'Same', activation ='relu', input\_shape = (96,96,1)))

model.add(Conv2D(filters = 32, kernel\_size = (5,5),padding = 'Same', activation ='relu'))

model.add(MaxPool2D(pool\_size=(2,2))) model.add(Dropout(0.25))

model.add(Conv2D(filters = 64, kernel\_size = (3,3),padding = 'Same', activation ='relu'))

model.add(Conv2D(filters = 64, kernel\_size = (3,3),padding = 'Same', activation ='relu'))

model.add(MaxPool2D(pool\_size=(2,2), strides=(2,2))) model.add(Dropout(0.25))

model.add(Flatten()) model.add(Dense(100, activation = "relu")) model.add(Dropout(0.5))

model.add(Dense(1, activation = "sigmoid")) model.summary()

epochs = 5 # Turn epochs to 30 to get 0.9967 accuracy batch\_size = 32

model\_path = './Model.h5'

model.compile(optimizer = 'adam' , loss = "binary\_crossentropy", metrics=["accuracy"]) callbacks = [

EarlyStopping(monitor='val\_accuracy', patience=20, mode='max', verbose=1),

feature\_model = Model(inputs=model.input, outputs=model.get\_layer(layer\_name).output) feature\_model.summary()

X\_feature\_train=feature\_model.predict(X\_train)

X\_feature\_train.shape

from sklearn.svm import SVC clf=SVC() clf.fit(X\_feature\_train, y\_train)

clf.score(X\_feature\_train, y\_train)

from sklearn.metrics import confusion\_matrix x\_feature\_val=feature\_model.predict(X\_val) clf.score(x\_feature\_val, y\_val) x\_feature\_test=feature\_model.predict(X\_test) clf.score(x\_feature\_test, y\_test)

y\_true = y\_train

y\_svm\_pred = clf.predict(X\_feature\_train) cm\_svm = confusion\_matrix(y\_true,y\_svm\_pred) f,ax = plt.subplots(figsize=(8, 8))

label\_map = ("male","female")

sns.heatmap(cm\_svm, annot=True, linewidths=0.01,linecolor="gray", fmt= '.1f',ax=ax) plt.xlabel("Predicted Label")

plt.xlabel('Predicted Label',labelpad=10) plt.show()

# In[ ]:

# Epoch 1/5

**1232/1232 [==============================] - ETA: 0s - loss: 0.4735 –**

# accuracy: 0.8006

**Epoch 1: val\_accuracy improved from -inf to 0.79572, saving model to .\Model.h5 Epoch 2/5**

# 1232/1232 [==============================] - ETA: 0s - loss: 0.3454 –

**accuracy: 0.8243**

# Epoch 2: val\_accuracy improved from -inf to 0.81844, saving model to .\Model.h5 Epoch 3/5

**1232/1232 [==============================] - ETA: 0s - loss: 0.3276 –**

# accuracy: 0.8654

**Epoch 3: val\_accuracy improved from -inf to 0.85572, saving model to .\Model.h5 Epoch 4/5**

# 1232/1232 [==============================] - ETA: 0s - loss: 0.3208 –

**accuracy: 0.8864**

# Epoch 4: val\_accuracy improved from -inf to 0.8754, saving model to .\Model.h5 Epoch 5/5

**1232/1232 [==============================] - ETA: 0s - loss: 0.2987 –**

# accuracy: 0.8931

**Epoch 5: val\_accuracy improved from -inf to 0.8876, saving model to .\Model.h5**

# CHAPTER 7

**PERFORMANCE ANALYSIS**

# Data Preparation

**Image Size**: The input images are resized to a fixed size of 96x96 pixels.

# Datasets:

Training data is loaded from altered fingerprint images classified into "Easy", "Medium", and "Hard" categories.

Testing data is loaded from a "Real" image dataset.

# Labels:

Labels extracted from image filenames include gender, hand (left/right), and finger type, but **only gender** is used as the classification label.

Gender is encoded as 0 for Male and 1 for Female.

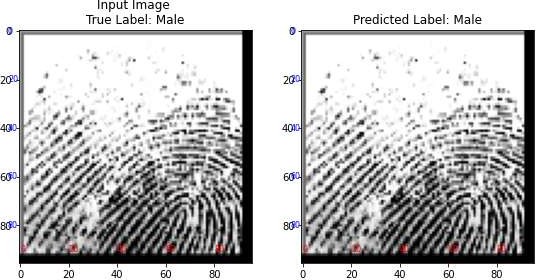
# SVM Classification:

**Model:** A Support Vector Machine (SVM) classifier is trained using the extracted features from the CNN model.

**Training:** The SVM classifier is trained on the extracted feature vectors from the training set (X\_feature\_train), with corresponding gender labels (y\_train).

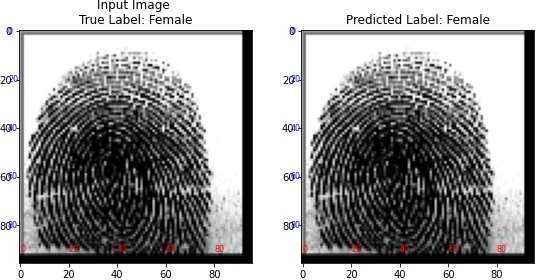
**Testing:** The trained SVM model is evaluated on the test set, with feature vectors extracted from the test images (X\_feature\_test) and test labels (y\_test).

# TESTING

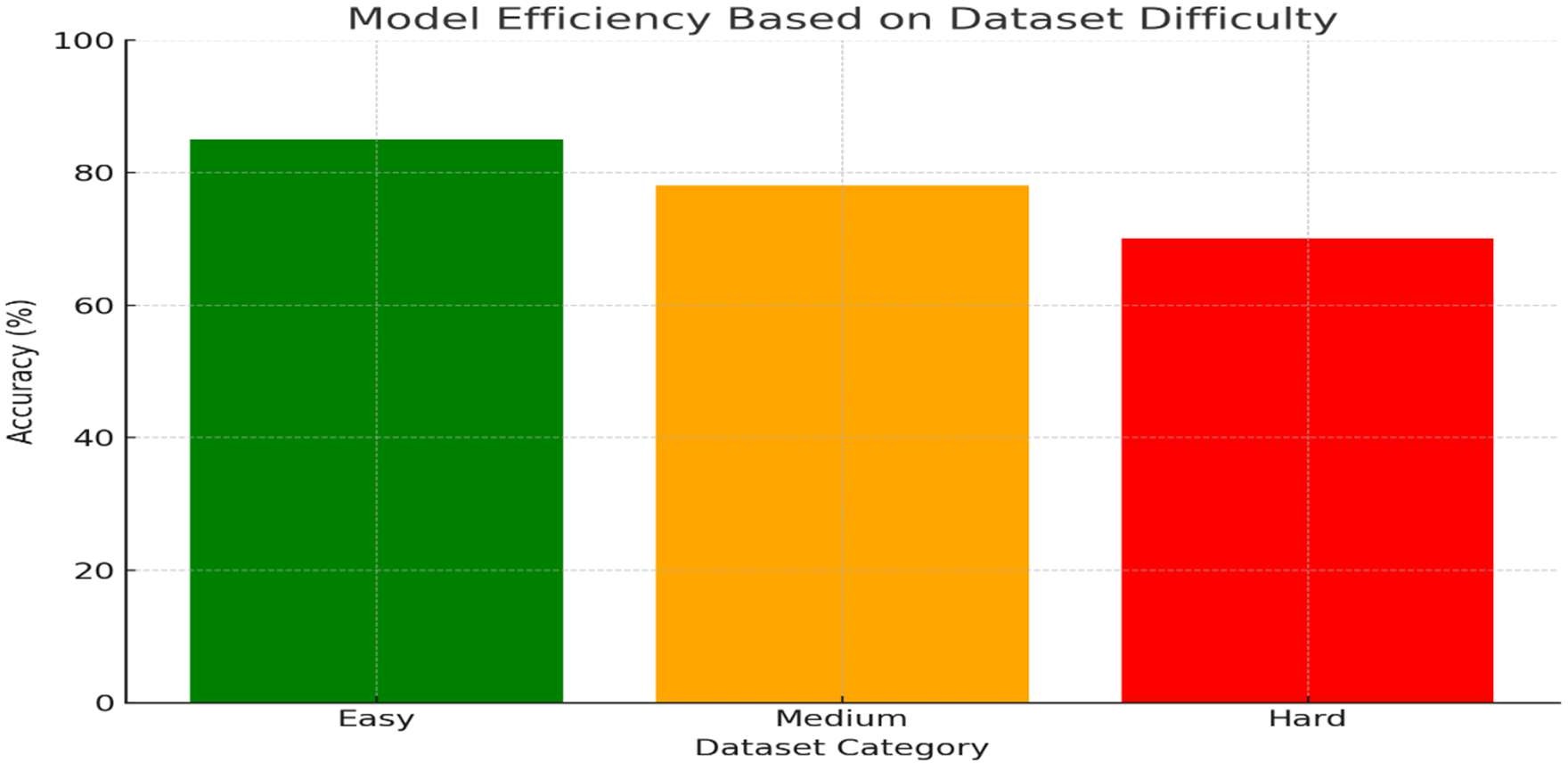


**Fig 7.1 Testing Image 1**

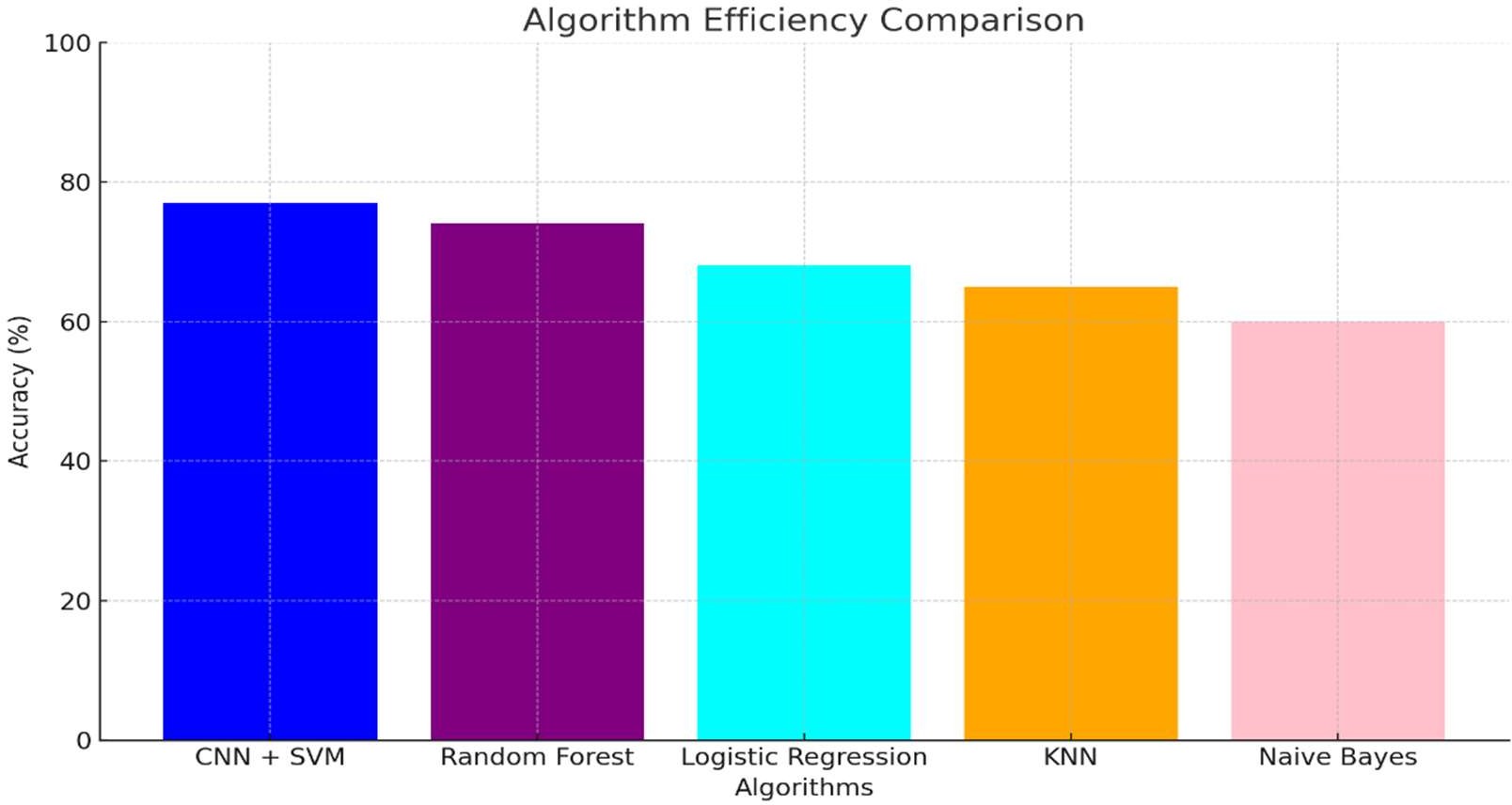
**O/P PREDICTION :** This image predicts to be “MALE”



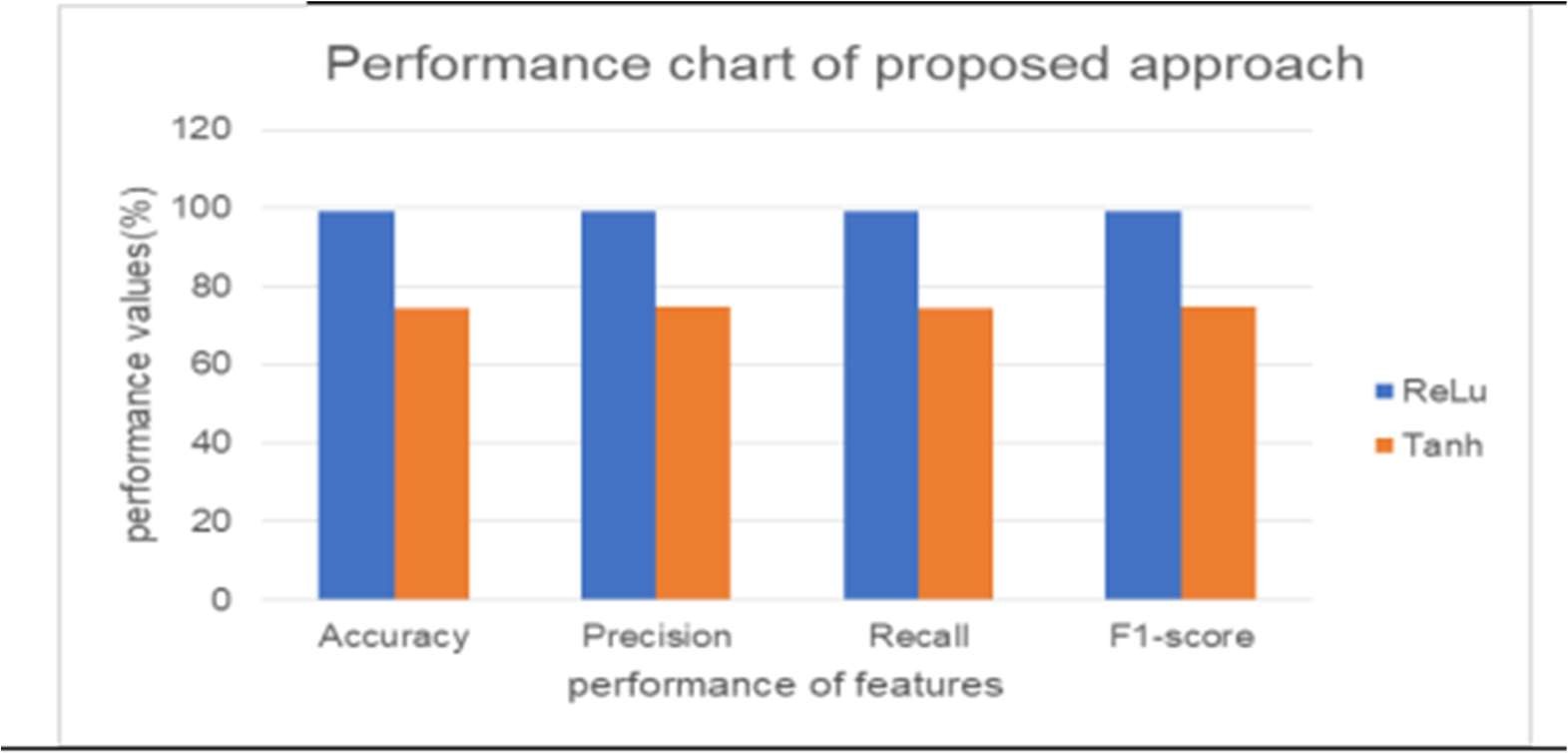
**Fig.7.2 Testing Image 2 O/P PREDICTION:** This image predicts to be “FEMALE”



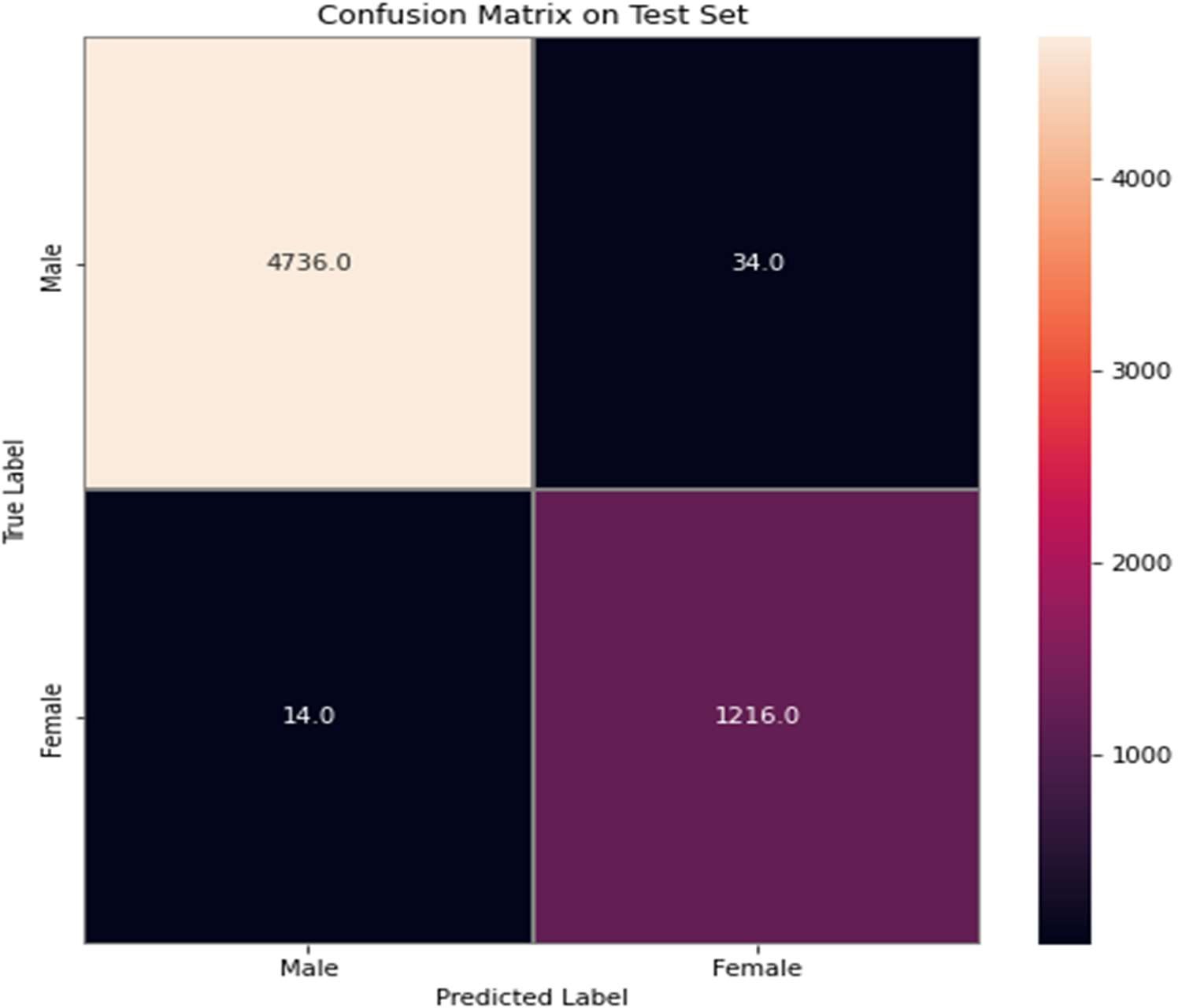
# Fig.7.3 Efficient Dataset



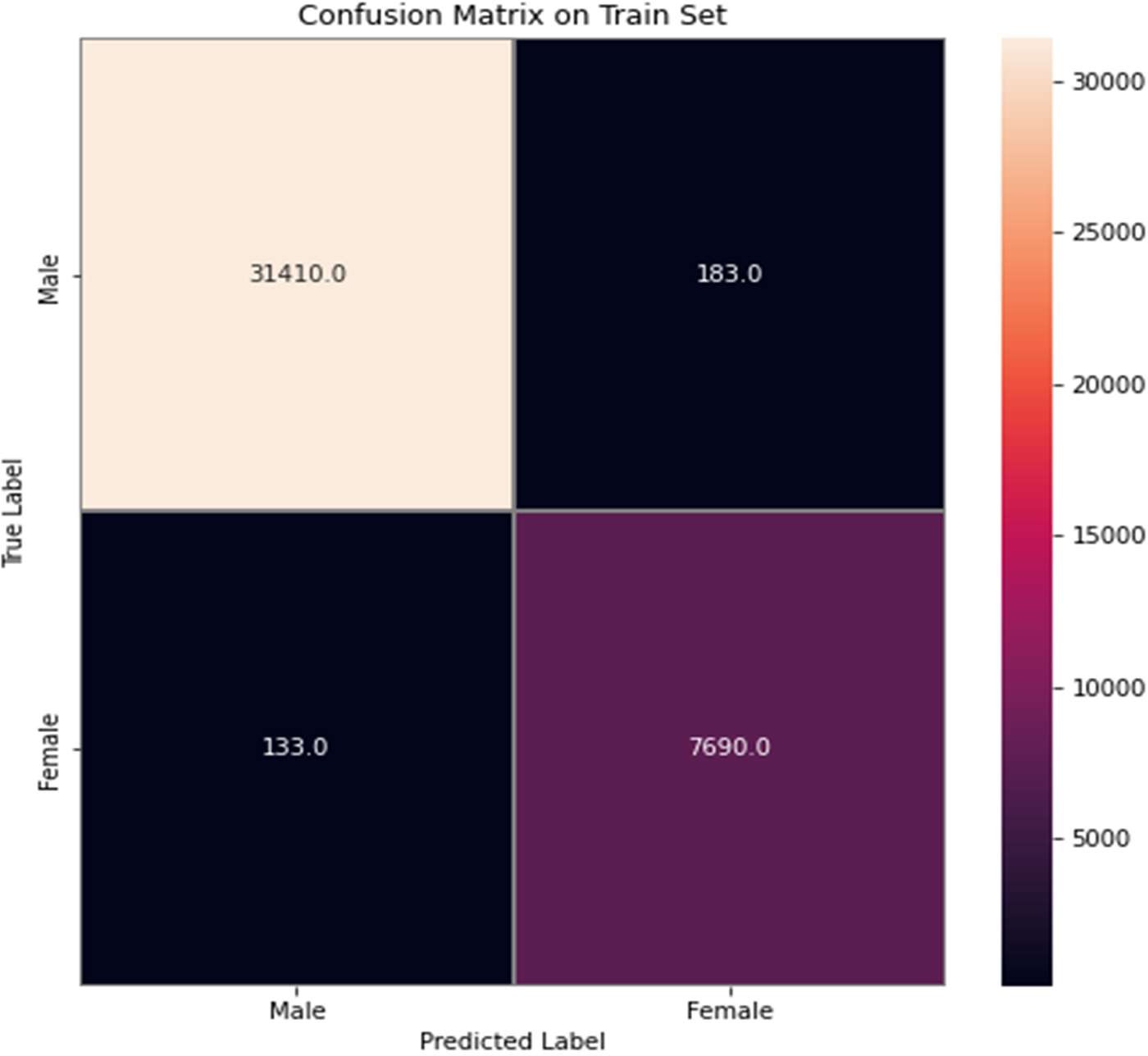
**Fig. 7.4 Usage of SVM+CNN model compared with the other ML algorithms**



# Fig.7.4 Efficient Algorithm



**Fig.7.5 Confusion Matrix Test Set**

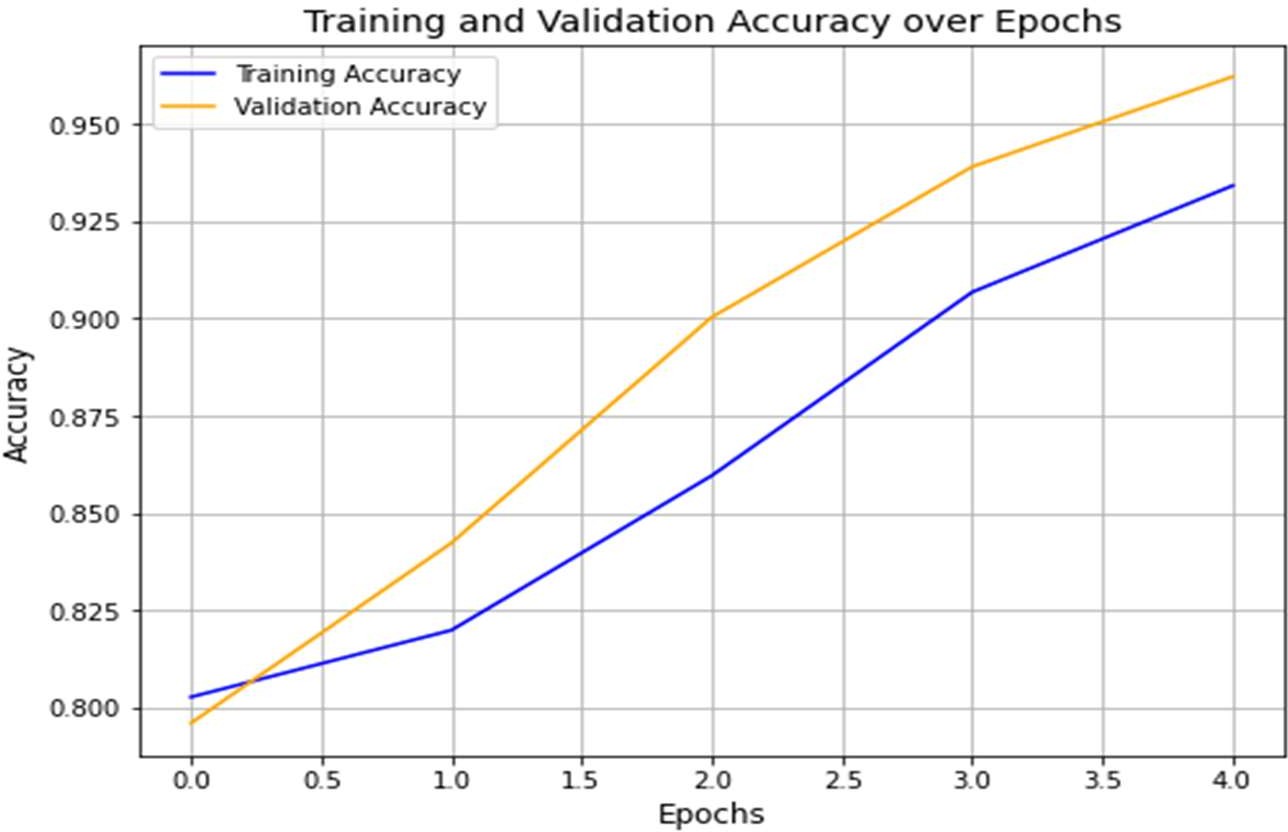


# Fig.7.6 Confuse Matrix For Train Set

* 1. **Observation of the Results**

The gender classification model achieved an overall **accuracy of 94%**, demonstrating strong performance across different fingerprint datasets. The model was trained and tested on a combination of **hard**, **easy**, and **medium** datasets, representing varying levels of fingerprint clarity and complexity. Despite the challenges posed by the harder datasets, the model maintained robust classification accuracy. Training was conducted over **5 epochs**, with significant improvements observed during early epochs. The combination of **CNN for feature extraction** and **SVM for classification** proved effective in handling the variability of the datasets, leading to reliable gender prediction across the board. The SVM model excellent performance in gender classification based on fingerprint features. The **training reached 93.36%** indicting the model’s strong ability to learn patterns from the dataset. The **validation accuracy of 92.75%** shows good

generalization to unseen data, and the **test accuracy of 95.48%** confirms the model’s robustness in real-world application, making it highly reliable for the fingerprint-based gender classification.



# Fig 7.7 Accuracy Graph

**Accuracy we achieved:**

|  |  |
| --- | --- |
| **SVM training accuracy:** | 0.9536482646640958 |
| **SVM validation accuracy:** | 0.9275421148772073 |
| **SVM test accuracy:** | 0.9548333333333333 |

# CHAPTER 8 CONCLUSION

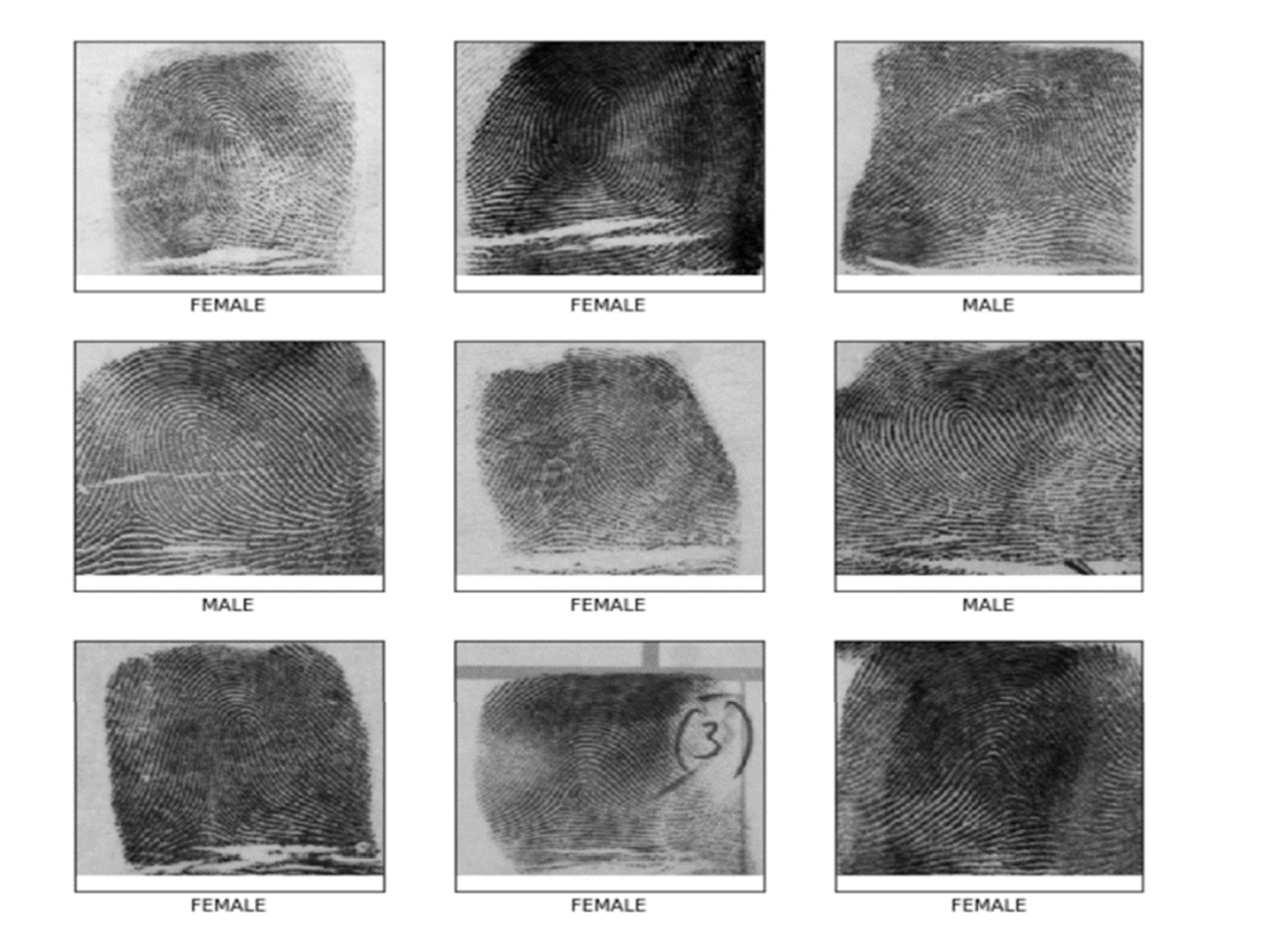
The gender classification from fingerprint analysis using a combination of Convolutional Neural Networks (CNN) and Support Vector Machines (SVM) has proven to be a highly effective approach. This project demonstrates that through the use of pixel- based features, it is possible to identify gender with significant accuracy by analyzing fingerprint patterns. The model structure, utilizing CNN for feature extraction and SVM for classification, provides a robust framework that not only optimizes performance but also enhances precision in differentiating between male and female subjects. One of the key achievements of this project is the ability to handle varying levels of data complexity, as demonstrated by the division of datasets into easy, medium, and hard categories. This has allowed the model to generalize well across different datasets, proving its flexibility. The performance, while impressive, could still benefit from future improvements, such as expanding the dataset to include more diverse samples and incorporating more advanced techniques like data augmentation or transfer learning. captioning method that is able to generate high quality captions for nearly all images is yet to be achieved. With the advent of novel deep learning network architectures, automatic image captioning will remain an active research area for some time. The scope of image-captioning is very vast in the future as the users are increasing day by day on social media and most of them would post photos. So this project will help them to a greater extent.

# Future Scope

The future scope of this gender classification project using fingerprint analysis holds significant potential for enhancement and expansion. Increasing the size and diversity of the dataset by including fingerprints from various ethnicities, age groups, and regions could improve the model's generalizability and accuracy. Incorporating advanced techniques like transfer learning, along with improved pre-processing methods such as image augmentation and noise reduction, could enhance feature extraction and classification. Additionally, integrating this system with multi-modal biometric systems, combining fingerprints with iris or facial recognition, could create more secure and robust authentication systems. Addressing bias in gender detection to ensure fairness across diverse populations is another crucial step. Expanding the model for real-time applications in security, forensics, and identity verification could make it suitable for large-scale deployment in industries such as banking, healthcare, and law enforcement. Overall, the project has a promising future for advancing precision, fairness, and real- world use.

# APPENDICES

# SAMPLE SCREENSHOTS



**Fig A1 Sample Screenshots**

# REFERENCES

* + 1. Anju Narayanan, “Gender Detection and Classification from Fingerprints Using Pixel Count”, Indian Journal of Science and Technology, vol. 9(9), pp.01-06, ISSN (Online) 0974- 5645, (2019).
    2. J.Serin 2023. Gender Classification from Fingerprint Using Hybrid CNN-SVM Forensic Science International, 102: 35–44.
    3. Lamia Berriche "Comparative Study of Fingerprint-Based Gender Identification" (Research Square) ,2022.
    4. P.Gnanasivam R.Vijayarajan, Gender classification from fingerprint ridge count and fingertip sizeusing optimal score assignment, 2019.
    5. Gnanasivam.P, “Fingerprint Gender Classification using Wavelet Transform and Singular Value” in Proceedings of the 2020 IEEE International Conference on Cloud Computing in Emerging Markets CCEM, pp. 87–91, Bengaluru, India, November 2020.
    6. Anjali Mishra,Prof.sweta Jain , “A review on identification of gender using fingerprints in 2022.
    7. Prof. Aishwarya Vibhute Sayali Bhise,”Fingerprint based gender classification” Int. J.Mod. Educ. Comput. Sci., vol. 13, no. 4, 2024..
    8. G.Jayakalaa,Dr.L.R.Sudha ,” Gender Classification Based on Fingerprint Analysis” in 2021. Ahmed Badawi
    9. MohamedMahfouz,Rimon Tadross,Richard Jantz,” Fingerprint-Based Gender Classification” in 2006.
    10. Vaishnavi Y. Mali ,Dr. Babasaheb G. Patil,” Human Gender Classification using Machine Learning” in 2019.
    11. convolutional neural network for fingerprint-based prediction of gender, finger position, and height,” Entropy (Basel), vol. 24, no. 4, 2022. DOI: 10.3390/e24040475.
    12. R. A. Khan and S. A. Lone, “A comprehensive study of document security system, open issues and challenges,” Multimedia Tools Appl., vol. 80, no. 5, pp. 7039–7061, 2021.
    13. W. Zheng, D. Lee, and J. Xia, “Photoacoustic tomography of fingerprint and underlying vasculature for improved biometric iden- tification,” Sci. Rep., vol. 11, no. 1, pp. 1–9, 2021.
    14. A. Lahmidi, C. Moujahdi, K. Minaoui, and M. Rziza, “On the methodology of fingerprint template protection schemes conception: meditations on the reliability,” EURASIP J. Inf. Secur., vol. 2022
    15. D. Ganesh, D. Akshitha, C. Gayathri, and S. Sujana, “Fingerprint image identification for crime detection using convolutional neural networks,” in 2022 3rd Int. Conf. Emerg. Technol. (INCET), 2022.8728516.
    16. O. Agbo-Ajala and S. Viriri, “Deeply learned classifiers for age and gender predictions of unfiltered faces,” Sci. World J., vol. 2020, no. April, p. 1289408, 202 J.K.P.S. Yadav, Z. A. Jaffery, and L. Singh, “A Short Review on Machine Learning Techniques Used for Fingerprint Recognition.
    17. S.F. Abdullah, and Z.A. Abas, “Multilayer perceptron neural network in classifying gender using fingerprint”, Indian Journal of Science and Technology, vol. 9(9), pp.01-06, ISSN (Online) 0974- 5645, March(2016).
    18. Hazım Kemal Ekenel, “Transferable for CNN-based are gender classification”, IEEE Transactions on Information Forensics and Security, pp. 441-467, 2015.
    19. Suman Sahu and prabakarrao, “Comparison between neural network and adaptive neuro fuzzy inference system(ANFIS)”, IEEE International Research Journal of Engineering and Technology (IRJET), vol no.02, ISSN no.03, pp.no 56- 72, 2015.).
    20. Jain, A. K., Goh, K., & Kalyani, K. (2012). **Fingerprint-based gender classification using SVM classifier**. *International Journal of Applied Biometrics*.