****

Project Report on

**Emotion Analysis of Signature Images using SVM & RFC**

**Submitted by**

Dilli Krishna (CU23MCA0019A)

Submitted as part of minor project course Summative Assessment 2024-25

Under the guidance of

**Dr. Krishnan R**

**Course:** MCA

Minor Project

III Semester 2024-2025

School of Engineering

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **SI.NO** | **CONTENTS** | **PG.NO** |
| **1** | **Introduction** | **1** |
| **2** | **Methodology** | **3** |
| **3** | **Literature review** | **4** |
| **4** | **Experiment Design** | **8** |
| **5** | **Conclusion** | **13** |
| **6** | **Source Code** | **14** |
| **7** | **References** | **23** |

**ABSTRACT**

The project, *"**Emotion Analysis of Signature Images using SVM & RFC,"* focuses on predicting an individual's emotional state based on variations in their signature. The primary objective is to develop a robust model that analyses signature samples and classifies emotions such as Angry, Sad, Happy, and Normal.

This study leverages machine learning algorithms, specifically Support Vector Machines (SVM) and Random Forest, for emotion prediction. Cross-validation techniques were employed to evaluate model accuracy, with both algorithms achieving an accuracy of 80% across various test sizes. Additionally, the time complexity for predicting emotions was analysed, revealing that SVM has a lower prediction time compared to Random Forest, which is computationally heavier. To enhance usability, a user-friendly interface was developed using Flask.

The interface provides information about the project and allows users to upload signature images for emotion prediction. This project holds significant potential for applications in behavioural analysis, psychological studies, and forensic sciences, offering an innovative approach to understanding emotional states through signature dynamics. The best accuracy achieved by Random Forest is 82.1% and by SVM is 82.2%.

1. **INTRODUCTION**

The study of human emotions has always been a topic of interest in various fields such as psychology, behavioural sciences, and artificial intelligence. Emotions are an integral part of human behaviour and play a critical role in decision-making, communication, and social interactions. Recognizing emotions accurately can help in understanding individuals better and improving human-computer interactions. Among various ways to analyse emotions, handwriting and signature analysis have garnered attention due to their unique ability to capture subtle variations influenced by an individual’s mental state. This project, *"Emotion Analysis of Signature Images using SVM & RFC,"* focuses on predicting emotional states such as Angry, Sad, Happy, and Normal by analyzing signature variations. Signatures are a representation of a person’s identity and often exhibit subtle changes influenced by emotional states. For instance, an individual may write with more pressure or irregular patterns when angry, while a relaxed state may result in smoother and more consistent strokes. Leveraging these variations, this project aims to classify emotions using machine learning techniques. The dataset for this project is carefully organized, where each individual's signatures are stored in folders labelled by emotional states. This hierarchical structure facilitates effective feature extraction and model training.

Machine learning algorithms like Support Vector Machines (SVM) and Random Forest have been employed in this project to analyse and classify emotions based on the extracted features from the signatures. SVM, known for its robust performance in binary and multiclass classification tasks, works by finding an optimal hyperplane that separates the data points belonging to different classes. Random Forest, a powerful ensemble algorithm, constructs multiple decision trees during training and outputs the mode of their predictions for classification. Cross-validation techniques were used to evaluate the performance of these models, and both algorithms achieved an accuracy of 80% across different test sizes. Beyond accuracy, the project also emphasizes analysing time complexity to determine the efficiency of each algorithm. The results indicate that SVM is computationally more efficient, with faster prediction times, while Random Forest, though slightly slower, provides a comprehensive understanding of the feature space through its ensemble approach. To make the project accessible and interactive, a user interface was developed using Flask. The UI provides an overview of the project and allows users to upload images of signatures for real-time emotion prediction. This feature makes the system user-friendly and practical for real-world applications.

This project holds immense potential in areas such as psychological analysis, behavioural studies, and forensic sciences. By automating the process of emotion recognition from signatures, it offers a novel approach to understanding individuals' emotional states. This can be particularly useful in areas such as criminal investigations, where analysing the emotional state of suspects can provide crucial insights, or in educational settings, where understanding students' emotions can help improve teaching methods. Overall, this project bridges the gap between traditional signature analysis and modern machine learning techniques, providing a valuable tool for emotion recognition with practical implications in various domains.

**Dataset Design**

The dataset is designed to analyse and classify emotional states (e.g., Angry, Sad, Happy, Normal) based on variations in signatures. It is organized hierarchically to facilitate efficient analysis and model training. Each individual persons in the dataset has a unique folder labelled as "Person1," "Person2," and so on, representing the subjects whose emotional states are being studied. Inside each individual's folder, there are subfolders corresponding to different emotional states, namely Angry, Sad, Happy, and Normal. These subfolders contain multiple signature samples captured under the respective emotional state. For instance, the "Angry" folder for Person1 contains signature samples created when the individual was experiencing anger, while the "Sad" folder contains samples from moments of sadness, and so on. This structured organization ensures the dataset is well-suited for extracting features and training machine learning models. The variation in signatures across different emotional states and individuals enables the models to identify patterns and predict emotions effectively.

1. **METHODOLOGY**

A systematic approach to predict emotions based on signature variations, beginning with data collection, and progressing through feature extraction, model development, validation, and GUI integration. Data collection involved gathering 100 signature samples from five individuals, with each person providing five samples for four emotional states: Happy, Sad, Angry, and Normal. Emotional states were elicited using carefully crafted prompts—joyful memories for happiness, loss or disappointment for sadness, frustrating situations for anger, and neutral thoughts for normality—ensuring authentic emotional responses and reliable data for analysis. Key features such as pen pressure, slant, baseline deviation, signature size, and letter spacing were extracted to analyse the emotional variations. These features were selected for their strong correlation with emotional states, such as heavier pressure indicating anger and smaller signatures suggesting sadness.

These extracted features formed the foundation for training machine learning models. Support Vector Machines (SVM) and Random Forest Classifiers were employed for emotion classification.

SVM, known for its efficiency in handling high-dimensional data, was used to maximize the margin between emotional classes.

Random Forest leveraged multiple decision trees for robust predictions. Hyperparameter tuning using techniques like grid search optimized the performance of both models, enabling them to generalize effectively. Cross-validation was applied to evaluate the models, ensuring reliable performance across various data splits. Both SVM and Random Forest achieved 80% accuracy, with SVM demonstrating faster prediction times compared to the computationally intensive Random Forest. A user-friendly interface was developed using Flask to enhance accessibility. The interface allows users to upload or draw signature samples, which are processed in real-time to predict emotions.

The results are displayed with visual cues, ensuring an intuitive and engaging experience. This comprehensive methodology ensures robust emotion prediction and practical applicability in various domains.

1. **LITERATURE REVIEW**

This paper proposes a multi-stage approach for Arabic handwriting recognition using Hierarchical Agglomerative Clustering (HAC), a ranking algorithm with PHoG and Kullback-Leibler divergence, and DCNNs. HAC reduces complexity, while the ranking algorithm improves classification accuracy. The system achieves 95.6% recognition accuracy using only 11% of the database. We can expand this method to other scripts and enhance adaptability [1].

This study presents an ensemble-based system for emotion recognition in speech, using XGBoost and SVM classifiers combined with a weighted voting mechanism. The system aims to enhance accuracy by assigning more weight to the better-performing model. Feature extraction methods like Mel-frequency cepstral coefficients (MFCCs) and pitch analysis capture the emotional nuances in speech. The proposed system achieves an impressive 88.3% overall accuracy, outperforming individual classifiers. Future work could focus on real-time applications and refining feature extraction techniques for further improvements [2].

Developed an in-depth review of digital graphology, exploring handwriting as a behavioral biometric for signature verification. The study emphasizes the significance of handwriting in revealing identity and personality traits, influenced by motor and cognitive processes. Key aspects such as datasets, preprocessing methods, and feature extraction techniques, including stroke dynamics, pen pressure, and slant, are discussed. Algorithms like Support Vector Machines (SVMs) and Neural Networks are evaluated for their effectiveness in signature verification and behavioral analysis. The paper highlights the potential applications of digital graphology, from authentication to psychological assessments and the early detection of neurological disorders, while noting challenges in feature extraction and real-time verification systems [3].

Explores the ethics of emotion recognition in AI systems, emphasizing the need for a nuanced approach to understanding human emotions. It highlights the challenges of defining emotions and the impact of proxy data on AI design, deployment, and societal effects. The paper discusses concerns about bias, misrepresentation, and ethical misuse in applications like recruitment, surveillance, and mental health. It advocates for interdisciplinary collaboration among experts in psychology, ethics, computer science, and sociology to create more accurate and ethical emotion AI models. The study calls for a cautious, reflective approach to emotion AI development, prioritizing ethical and social considerations [4].

Represents a database designed to predict emotional states using handwriting and signature biometrics. Collected from 134 participants, the dataset links handwriting features like pressure, stroke dynamics, and slant to emotions such as happy, sad, and stressed. The study demonstrates the potential of these features in accurately predicting emotions through machine learning models, offering valuable insights for applications in mental health, user authentication, and emotion-aware systems [5].

This paper introduces a path signature-based approach for speech emotion recognition (SER), using a hierarchical tree structure to capture both short-term speech features and long-term temporal dependencies. It employs a Tree-Based Convolutional Neural Network (TBCNN) for efficient feature learning, preserving both local and global speech patterns. The method shows strong performance with reduced reliance on feature engineering, making it efficient for real-world applications. The approach offers potential advancements in affective computing, virtual assistants, and mental health diagnostics [6].

Research shows a method for predicting human behavior through the analysis of handwriting features, such as pen pressure, baseline consistency, and specific letter traits. The study extracts these features to uncover personality traits and behavioral patterns, using Support Vector Machines (SVM) for classification. The results demonstrate high accuracy in predicting behaviors, highlighting the potential of automated handwriting analysis for psychological assessments. The approach offers practical applications in areas like recruitment, counseling, and forensic investigations, combining graphology with advanced computational tools for behavioral prediction [7].

Deep learning approach for predicting stress levels through offline handwritten signature analysis, using architectures like AlexNet, ResNet, and DenseNet. The study analyzes signature features such as pen pressure and stroke dynamics, achieving an average accuracy of 77%. It emphasizes the use of soft-biometric features in emotion recognition and forensic applications, with potential uses in mental health monitoring and workplace wellness. DenseNet, in particular, shows promise due to its efficient feature reuse. The research opens new possibilities for applying deep learning to behavioral biometrics for stress prediction [8].

Investigates the neural basis of six basic emotions using multivariate pattern analysis (MVPA) in fMRI studies. The study identifies specific brain regions, such as the medial prefrontal cortex and posterior cingulate cortex, linked to emotions like happiness, sadness, anger, fear, disgust, and surprise. It demonstrates that these neural signatures are consistent across various emotion-induction methods and individuals. The research supports a distributed model of emotional processing, showing that emotions are represented by coordinated activity across multiple brain regions, with implications for developing emotion-based biomarkers and therapeutic strategies [9].

A novel approach to predicting psychological states by analyzing human signatures using time-series data derived from signature images. By transforming 2D signature images into 1D time series and applying the discrete cosine transform (DCT) for feature extraction, the study links variations in signature dynamics to emotional states like stress, happiness, and anxiety. The k-Nearest Neighbors (k-NN) algorithm is used for classification, achieving high accuracy in identifying psychological states. The research emphasizes the potential of signature analysis for psychological assessments and forensic applications, offering a new perspective on human behavior and emotional expression [10].

Explores the automatic prediction of emotional states through handwriting analysis, focusing on features such as stroke pressure, speed, slant, and baseline consistency. Using a dataset of 100 subjects, the study employs various classifiers to predict emotional states like happiness and stress, achieving up to 80% accuracy. The research highlights the potential of handwriting analysis in forensic investigations and psychological assessments, suggesting it can provide insights into the emotional state of individuals, such as suspects or witnesses, and enhance the accuracy of evaluations and interrogations [11].

This paper shows the use of online handwritten signatures for predicting personality traits, specifically based on the Big Five personality model. By analyzing features such as stroke pressure, slant, speed, and curvature, the study uses the k-Nearest Neighbour (kNN) algorithm to classify personality traits, achieving a classification accuracy of 87.5%. Validation against the Big Five Inventory (BFI) test shows a 90% match between signature-based predictions and test results. The study highlights the potential of signature analysis for non-invasive personality assessments, with applications in recruitment, counseling, and personal development [12].

Examines neural signatures related to emotion regulation using fMRI and multivariate pattern analysis (MVPA). The study compares cognitive reappraisal, where individuals regulate their emotional responses, with passive viewing of negative images. It identifies key brain regions like the prefrontal cortex and amygdala, showing distinct activation patterns for each task. The research achieves 82.5% accuracy in classifying emotional responses, suggesting that these neural patterns could help understand individual emotional processing, with potential applications in treating mental health conditions such as depression and anxiety [13].

Developed a neural signature of human affiliative emotions, focusing on brain regions involved in processing social bonds and attachment. Using fMRI, it identifies the septohypothalamic area as key in processing emotions linked to kinship and close social interactions. The study highlights distinct neural circuits for social bonding, separate from general emotional responses, and explores the roles of the prefrontal cortex and amygdala. These findings have implications for understanding social dysfunctions in conditions like autism, schizophrenia, and social anxiety, offering potential for future therapeutic interventions [14].

A review paper on the use of handwriting analysis (graphology) for predicting personality traits, focusing on handwriting features like pen pressure, letter slant, size, and spacing. It explores advanced techniques like machine learning (e.g., SVM and neural networks) to enhance accuracy and consistency in personality prediction, moving beyond subjective traditional methods. The paper discusses the significance of handwriting features in revealing emotional states and personality traits, with potential applications in psychology, personality profiling, and forensic investigations. Overall, it highlights handwriting analysis as a valuable tool for behavioral assessments in clinical and investigative settings [15].

A framework for detecting emotions in text translated from English into Chinese, French, German, and Spanish, focusing on how emotions are preserved during translation. Using computational intelligence techniques like TFIDF for feature extraction and PCA for dimensionality reduction, the study achieves high classification accuracy (99.04% for French). The research demonstrates that, while some nuances may be lost in translation, the core emotional content is largely preserved across languages. This framework has significant applications in global communication, sentiment analysis, and cross-cultural studies, advancing emotion recognition in multilingual contexts [16].

The study shows the neural signatures associated with human affiliative emotions, focusing on the brain regions involved in kinship-related social scenarios. Using fMRI, the study identifies the septohypothalamic area as a key region activated during affiliative experiences, distinct from general emotional reactions. The research highlights the role of this area in processing attachment, trust, and affection, which are central to human relationships. The findings have implications for understanding neuropsychiatric conditions like autism, schizophrenia, and social anxiety, offering insights into the neural mechanisms underlying social attachment and emotional connections [17].

Used a handwritten signatures to recognize emotions. Key features like pen pressure, signature length, and flow are linked to emotional states. Machine learning classifiers (k-NN and SVM) achieved 87.5% accuracy in predicting emotions. The study shows that pen pressure indicates emotional intensity, while signature length and flow reflect mood. This method offers a non-invasive way to detect emotions, with applications in psychology and forensics [18].

Advancements and challenges in emotion detection from text, focusing on its importance in fields like marketing, psychology, and customer service. A major challenge is the complexity of emotions, which are often subtle, multi-dimensional, and context-dependent. Emotions can be expressed implicitly through tone, word choice, and context, making detection difficult for traditional models. Current methods face limitations due to the reliance on supervised learning and lack of diverse, annotated datasets. The paper calls for improved models that consider implicit emotional expressions, context, and subtle variations in language. Future research should enhance dataset quality and explore deeper linguistic features to improve emotion detection for applications in marketing, mental health, and user experience [19].

Introduced the Deep Learning Assisted Semantic Text Analysis (DLSTA) model for emotion detection in text, combining semantic and syntactic features to achieve 97.22% accuracy in emotion detection and 98.02% in classification. The model utilizes word embeddings to capture contextual meanings of words, enhancing its ability to detect subtle emotions. It emphasizes the importance of emotion recognition in applications like human-computer interaction and sentiment analysis. Future work should focus on refining emotion models and expanding datasets to include diverse linguistic and cultural contexts [20].

Discussed the advancements and challenges in text-based emotion detection, highlighting its growing importance in fields like marketing, mental health, and customer service. The study emphasizes the complexity of emotions, which are subtle, multi-dimensional, and context-dependent, making them difficult to detect. Key challenges include the need for comprehensive datasets and the ability to capture both explicit and implicit emotions. The paper proposes future research directions, such as improving model accuracy, transparency, and incorporating multi-modal inputs, to enhance emotion detection systems for more empathetic interactions across industries [21].

The study shows an innovative, cost-effective emotion detection model combining questionnaire responses and text analysis. Using machine learning techniques like support vector machines (SVM) and artificial neural networks (ANN), alongside Dempster-Shafer Theory (DST) to handle uncertainty, the system classifies emotional states with improved accuracy. The model proves effective in detecting subtle emotional cues, benefiting applications like workplace stress, mental health support, and identifying suicidal tendencies in students. By integrating subjective and objective assessments, the approach aims to enhance mental health care and overall societal well-being [22].

The paper presents a deep learning model for emotion detection in text, combining semantic analysis and contextual understanding to improve accuracy. Unlike traditional methods that focus on keywords or basic sentiment analysis, the model captures subtle emotional cues and implicit expressions by considering both the meaning of words and their broader context. The model outperforms traditional techniques, achieving higher accuracy rates in emotion classification. The research highlights its potential in enhancing human-computer interaction (HCI) by providing more empathetic and personalized user experiences, with future applications in customer service, mental health, and sentiment analysis [23].

The research explains an attention-based transformer model for emotion detection from handwriting and drawing samples, achieving 92.64% accuracy on the EMOTHAW dataset. The model effectively captures long-range dependencies in dynamic handwriting and drawing movements, which traditional methods struggle to analyze. By integrating both handwriting and drawing, it captures a broader range of emotional states. The study suggests future improvements, focusing on expanding the dataset and enhancing the model’s robustness, with potential applications in mental health assessment and personalized user interactions [24].

A method combining handwriting analysis with the DASS (Depression, Anxiety, and Stress Scale) to detect emotional states, achieving 91.25% accuracy. Using a Convolutional Neural Network (CNN), the model analyzes handwriting features like stroke patterns, pressure, and speed, alongside psychological data from the DASS scale. This integration offers a comprehensive approach to emotion detection, particularly useful for identifying negative emotions such as depression and stress. The study highlights the potential of combining deep learning with psychological tools for improved emotional state assessment, especially in mental health applications [25].

Explores conversational emotion recognition using both text and audio, focusing on enhancing emotion detection in dynamic dialogues. The study employs a Bi-LSTM model for text, achieving 75% accuracy, and an LSTM model for audio, which performs at 47%. Using the MELD dataset from the TV show "Friends," the research highlights challenges in extracting emotional cues from spoken language in multi-party conversations. Text-based recognition outperforms audio-based methods, and the study suggests developing advanced multimodal models to better integrate both text and audio for improved emotion detection in complex dialogues [26].

A deep learning framework for emotion detection in multimedia text, specifically using TV show transcripts. The model employs a sequence-based CNN architecture and incorporates an attention mechanism to focus on relevant text parts, enhancing emotion classification. By analyzing both explicit and implicit emotional cues, the model outperforms traditional methods, demonstrating higher accuracy. The study highlights the potential of this framework in applications like brand management, customer feedback, and social media sentiment analysis, showcasing how deep learning can improve emotion recognition in dynamic textual data [27].

Semantic-Emotion Neural Network (SENN), a deep learning model designed for enhanced emotion recognition from text. It integrates semantic and emotional information by using a Bi-LSTM for semantic encoding and a CNN for extracting emotional features. SENN captures both contextual meaning and emotional cues, outperforming traditional models in accuracy. The study emphasizes the importance of combining semantic and emotional context for more precise emotion detection. Future work will focus on incorporating additional emotional word embeddings to further improve the model's performance across various contexts [28].

Used of machine learning for emotion detection in text to enhance human-machine interaction. It compares traditional models like Naïve Bayes and SVM with deep learning approaches, finding that neural networks perform the best in capturing emotional patterns. The study highlights the importance of emotion detection in applications like customer service and mental health monitoring, where empathetic responses are crucial. A web application demonstrates the model's real-time emotion analysis, showcasing its potential for practical use. Future work aims to address challenges like context and sarcasm to improve accuracy across diverse datasets [29].

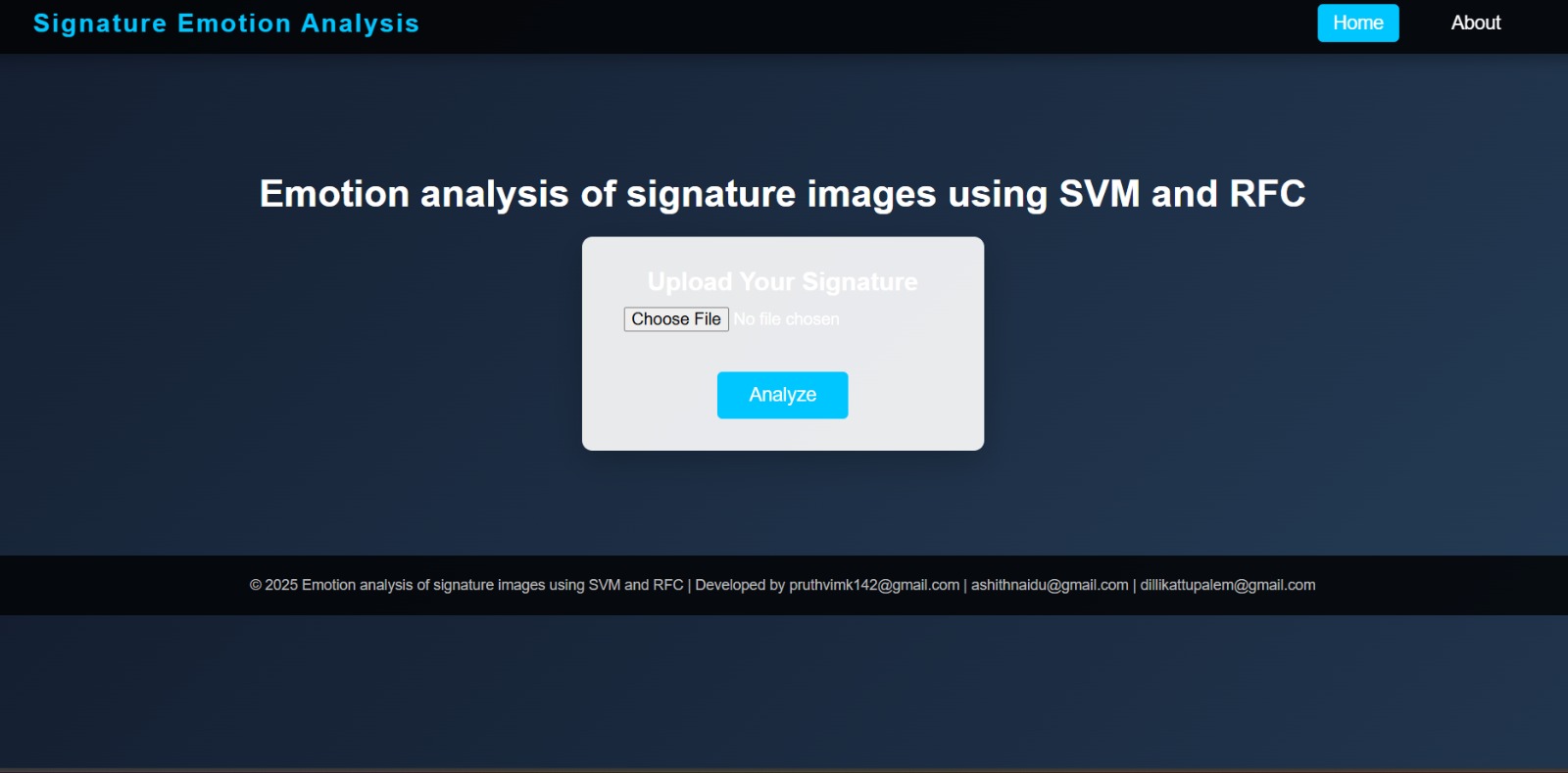
This paper outlines a multi-stage method for recognizing emotions through handwriting analysis, focusing on key features like baseline, slant, and pen pressure. The process involves digitizing handwritten samples, removing noise, and applying skew corrections, followed by classification using machine learning techniques. The study shows high accuracy in detecting emotions such as happiness, sadness, anger, and neutrality, highlighting its potential for psychological assessments, forensic analysis, and mental health evaluation. Future work will explore integrating deep learning models for improved real-time emotion tracking [30].

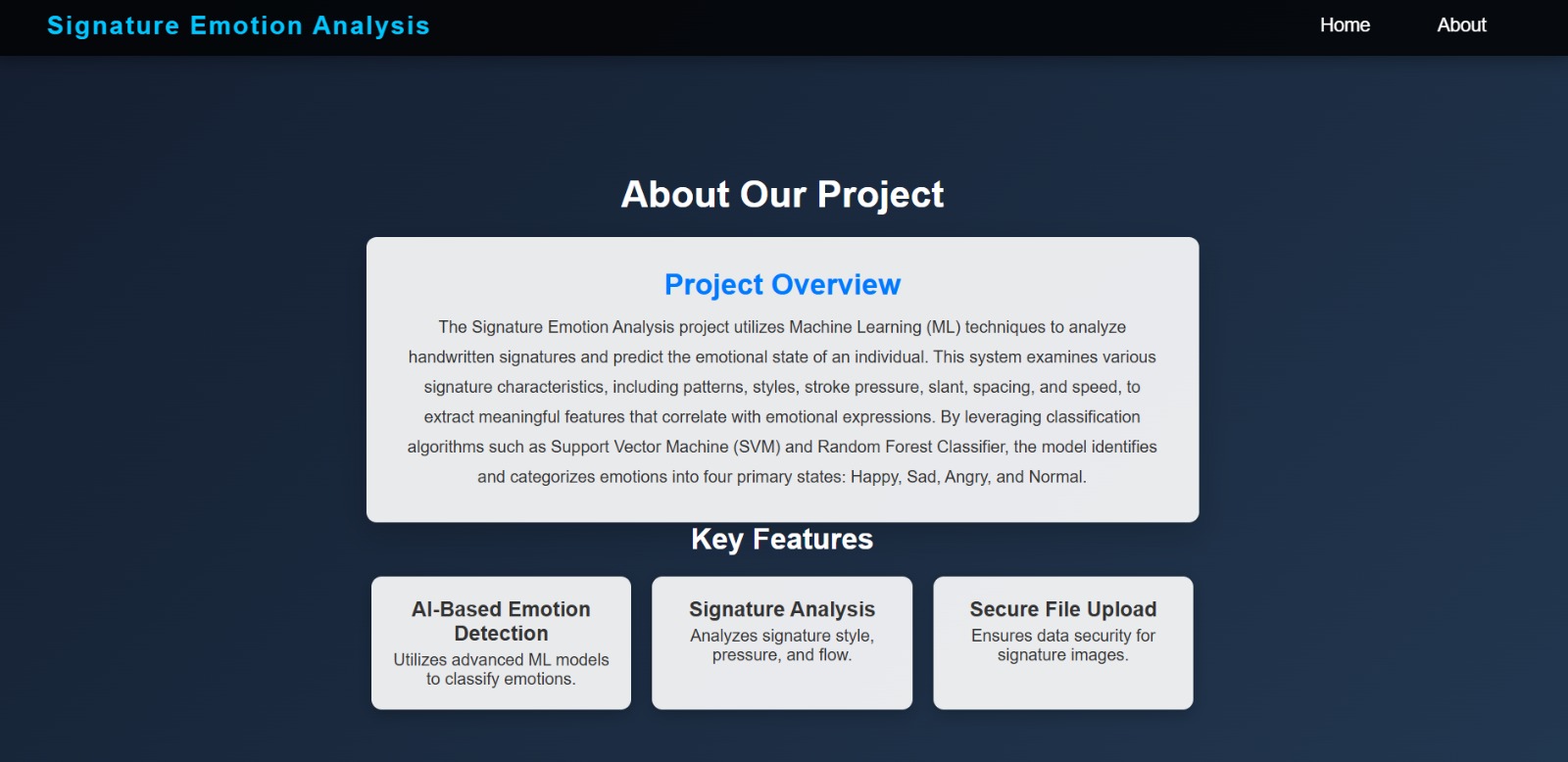
1. **EXPERIMENT DESIGN**

The goal of this experiment was to classify emotional states based on variations in handwriting signatures using two machine learning models: Support Vector Machine (SVM) with a linear kernel and Random Forest Classifier. Emotional states were induced through controlled prompts, and signatures were labelled accordingly. To predict these emotional states, six key features were extracted from each signature: average pixel intensity, symmetry features, thickness, gaps between contours, slant (using the Hough transform), and endline length. These features were selected to capture various aspects of handwriting that may reflect emotional variations, such as pressure, balance, and slant indicative of emotional intensity. The dataset was split into training and testing sets with proportions of 90%-10%, 80%-20%, and 70%-30%, and cross-validation was applied for model robustness. Both models underwent hyperparameter tuning via grid search, optimizing their performance. A Flask-based web interface was developed, allowing users to upload or draw signatures, with the system predicting the emotional state and displaying an emoji representing the detected emotion. The performance of both models was evaluated using accuracy, precision, recall, and F1-score, and the results were compared. Suggestions for improvement included adding more features, such as stroke speed or signature area, and exploring ensemble methods to combine the outputs of both models for enhanced accuracy.

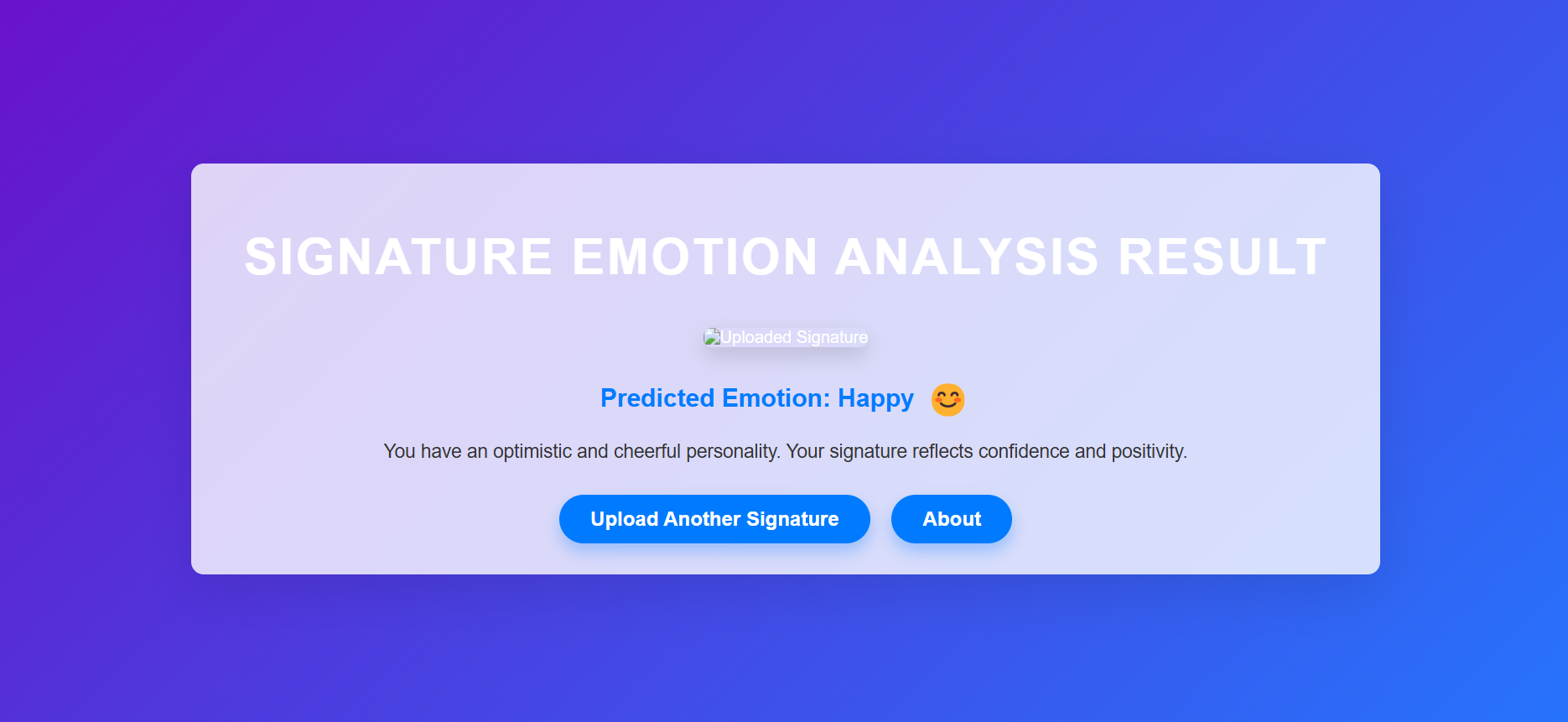
**RESULTS**

User Interface





After Predicting Emotion



1. **CONCLUSION**

This experiment demonstrated that handwriting-based emotional state classification using machine learning models. Both Support Vector Machine (SVM) and Random Forest classifiers achieved a peak accuracy of 80% for specific test sizes, demonstrating their capability to predict emotions. Key extracted features, including average pixel intensity, symmetry, thickness, contour gaps, slant, and endline length, played a critical role in capturing subtle emotional variations. The Random Forest model performed well at a 25% test size but showed sensitivity to data distribution with larger test sizes. Similarly, SVM exhibited competitive performance but showed fluctuations with varying test sizes, indicating opportunities for optimization. The implementation of a Flask-based web interface validated the system's practical application by enabling real-time emotion predictions with a user-friendly interface. The study underscores the importance of balanced train-test splits, cross-validation, and hyperparameter tuning for robust performance. Future work could focus on adding features like stroke speed and signature area and exploring ensemble methods to improve prediction accuracy and reliability. These advancements will further enhance handwriting analysis as a tool for emotion detection.

1. **SOURCE CODE**
   1. **Support Vector Machine**

import os

import cv2

import numpy as np

from sklearn.model\_selection import train\_test\_split

from sklearn.svm import SVC

from sklearn.preprocessing import LabelEncoder, StandardScaler

from sklearn.metrics import classification\_report, accuracy\_score

from sklearn.pipeline import Pipeline

import time  # Importing time module to measure execution time

# Feature Extraction Functions

def compute\_avg\_pixel\_intensity(image):

    """Compute the average pixel intensity of an image."""

    return np.mean(image)

def compute\_symmetry\_features(image):

    """Compute horizontal and vertical symmetry features."""

    vertical\_symmetry = np.mean(image[:, :image.shape[1] // 2] - image[:, image.shape[1] // 2:])

    horizontal\_symmetry = np.mean(image[:image.shape[0] // 2, :] - image[image.shape[0] // 2:, :])

    return vertical\_symmetry, horizontal\_symmetry

def compute\_thickness(image):

    """Estimate the stroke thickness based on edge detection."""

    edges = cv2.Canny(image, 100, 200)

    return np.sum(edges) / (image.shape[0] \* image.shape[1])

def compute\_gaps\_between\_contours(image):

    """Calculate the average gap between detected contours."""

    contours, \_ = cv2.findContours(image, cv2.RETR\_EXTERNAL, cv2.CHAIN\_APPROX\_SIMPLE)

    if len(contours) < 2:

        return 0

    centers = [cv2.boundingRect(c)[0] + cv2.boundingRect(c)[2] // 2 for c in contours]

    centers.sort()

    gaps = [centers[i + 1] - centers[i] for i in range(len(centers) - 1)]

    return np.mean(gaps) if gaps else 0

def compute\_slant\_using\_hough(image):

    """Detect slant using the Hough Transform."""

    edges = cv2.Canny(image, 50, 150, apertureSize=3)

    lines = cv2.HoughLines(edges, 1, np.pi / 180, 200)

    if lines is not None:

        angles = [np.rad2deg(np.arctan2(np.sin(theta), np.cos(theta))) for rho, theta in lines[:, 0, :]]

        return np.mean(angles)

    return 0

def compute\_endline\_length(image):

    """Calculate the length of the endline (bottom-most stroke)."""

    edges = cv2.Canny(image, 100, 200)

    bottom\_row = edges[-1, :]

    return np.sum(bottom\_row) / 255

# Combine all features

def extract\_features(image\_path):

    """Extract all features for a given image."""

    image = cv2.imread(image\_path, cv2.IMREAD\_GRAYSCALE)

    if image is None:

        raise ValueError(f"Image at {image\_path} could not be loaded.")

    # Preprocess: Resize image for consistency

    image = cv2.resize(image, (100, 100))  # Resize to 100x100 pixels

    # Extract features

    features = [

        compute\_avg\_pixel\_intensity(image),

        \*compute\_symmetry\_features(image),

        compute\_thickness(image),

        compute\_gaps\_between\_contours(image),

        compute\_slant\_using\_hough(image),

        compute\_endline\_length(image)

    ]

    return np.array(features)

# Load Dataset

def load\_dataset(dataset\_path):

    """Load dataset and extract features and labels."""

    X, y = [], []

    for emotion in os.listdir(dataset\_path):

        emotion\_folder = os.path.join(dataset\_path, emotion)

        if os.path.isdir(emotion\_folder):

            for image\_file in os.listdir(emotion\_folder):

                image\_path = os.path.join(emotion\_folder, image\_file)

                try:

                    features = extract\_features(image\_path)

                    X.append(features)

                    y.append(emotion)

                except Exception as e:

                    print(f"Error processing {image\_path}: {e}")

    return np.array(X), np.array(y)

def train\_and\_evaluate\_svm(X, y, test\_size):

    le = LabelEncoder()

    y\_encoded = le.fit\_transform(y)

    # Ensure that test\_size is large enough to have samples from all classes

    n\_classes = len(np.unique(y\_encoded))

    if test\_size < 1 / n\_classes:

        print("Increasing test size.")

        test\_size = 1 / n\_classes

    X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y\_encoded, test\_size=test\_size, stratify=y\_encoded)

    pipeline = Pipeline([

        ('scaler', StandardScaler()),

        ('svm', SVC(kernel='rbf', C=10, gamma='scale'))

    ])

    # Measure training time

    start\_time = time.time()

    pipeline.fit(X\_train, y\_train)

    training\_time = time.time() - start\_time

    # Measure prediction time

    start\_time = time.time()

    y\_pred = pipeline.predict(X\_test)

    prediction\_time = time.time() - start\_time

    accuracy = accuracy\_score(y\_test, y\_pred)

    # Store the accuracy with its corresponding test size and predicted emotions

    results = {

        'test\_size': test\_size,

        'accuracy': accuracy,

        'predicted\_emotions': le.inverse\_transform(y\_pred),

        'training\_time': training\_time,

        'prediction\_time': prediction\_time

    }

    print(f"Accuracy: {accuracy}")

    print(f"Predicted Emotions: {results['predicted\_emotions']}\n")

    print(f"Training Time: {training\_time:.4f} seconds")

    print(f"Prediction Time: {prediction\_time:.4f} seconds\n")

    return accuracy, pipeline, le, results

* 1. **Random Forest Classifier**

import os

import cv2

import numpy as np

import time

from sklearn.model\_selection import train\_test\_split

from sklearn.ensemble import RandomForestClassifier

from sklearn.preprocessing import LabelEncoder, StandardScaler

from sklearn.metrics import classification\_report, accuracy\_score

from sklearn.pipeline import Pipeline

# Feature Extraction Functions

def compute\_avg\_pixel\_intensity(image):

    """Compute the average pixel intensity of an image."""

    return np.mean(image)

def compute\_symmetry\_features(image):

    """Compute horizontal and vertical symmetry features."""

    vertical\_symmetry = np.mean(image[:, :image.shape[1] // 2] - image[:, image.shape[1] // 2:])

    horizontal\_symmetry = np.mean(image[:image.shape[0] // 2, :] - image[image.shape[0] // 2:, :])

    return vertical\_symmetry, horizontal\_symmetry

def compute\_thickness(image):

    """Estimate the stroke thickness based on edge detection."""

    edges = cv2.Canny(image, 100, 200)

    return np.sum(edges) / (image.shape[0] \* image.shape[1])

def compute\_gaps\_between\_contours(image):

    """Calculate the average gap between detected contours."""

    contours, \_ = cv2.findContours(image, cv2.RETR\_EXTERNAL, cv2.CHAIN\_APPROX\_SIMPLE)

    if len(contours) < 2:

        return 0

    centers = [cv2.boundingRect(c)[0] + cv2.boundingRect(c)[2] // 2 for c in contours]

    centers.sort()

    gaps = [centers[i + 1] - centers[i] for i in range(len(centers) - 1)]

    return np.mean(gaps) if gaps else 0

def compute\_slant\_using\_hough(image):

    """Detect slant using the Hough Transform."""

    edges = cv2.Canny(image, 50, 150, apertureSize=3)

    lines = cv2.HoughLines(edges, 1, np.pi / 180, 200)

    if lines is not None:

        angles = [np.rad2deg(np.arctan2(np.sin(theta), np.cos(theta))) for rho, theta in lines[:, 0, :]]

        return np.mean(angles)

    return 0

def compute\_endline\_length(image):

    """Calculate the length of the endline (bottom-most stroke)."""

    edges = cv2.Canny(image, 100, 200)

    bottom\_row = edges[-1, :]

    return np.sum(bottom\_row) / 255

# Combine all features

def extract\_features(image\_path):

    """Extract all features for a given image."""

    image = cv2.imread(image\_path, cv2.IMREAD\_GRAYSCALE)

    if image is None:

        raise ValueError(f"Image at {image\_path} could not be loaded.")

    # Preprocess: Resize image for consistency

    image = cv2.resize(image, (100, 100))  # Resize to 100x100 pixels

    # Extract features

    features = [

        compute\_avg\_pixel\_intensity(image),

        \*compute\_symmetry\_features(image),

        compute\_thickness(image),

        compute\_gaps\_between\_contours(image),

        compute\_slant\_using\_hough(image),

        compute\_endline\_length(image)

    ]

    return np.array(features)

# Load Dataset

def load\_dataset(dataset\_path):

    """Load dataset and extract features and labels."""

    X, y = [], []

    for emotion in os.listdir(dataset\_path):

        emotion\_folder = os.path.join(dataset\_path, emotion)

        if os.path.isdir(emotion\_folder):

            for image\_file in os.listdir(emotion\_folder):

                image\_path = os.path.join(emotion\_folder, image\_file)

                try:

                    features = extract\_features(image\_path)

                    X.append(features)

                    y.append(emotion)

                except Exception as e:

                    print(f"Error processing {image\_path}: {e}")

    return np.array(X), np.array(y)

def train\_and\_evaluate\_random\_forest(X, y, test\_size):

    le = LabelEncoder()

    y\_encoded = le.fit\_transform(y)

    # Ensure that test\_size is large enough to have samples from all classes

    n\_classes = len(np.unique(y\_encoded))

    if test\_size < 1 / n\_classes:

        print("Increasing test size.")

        test\_size = 1 / n\_classes

    X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y\_encoded, test\_size=test\_size, stratify=y\_encoded)

    # Create a pipeline with scaling and RandomForest

    pipeline = Pipeline([

        ('scaler', StandardScaler()),

        ('rf', RandomForestClassifier(n\_estimators=100))

    ])

    # Start time for training

    start\_time = time.time()

    pipeline.fit(X\_train, y\_train)

    # End time for training

    training\_time = time.time() - start\_time

    # Start time for prediction

    start\_prediction\_time = time.time()

    y\_pred = pipeline.predict(X\_test)

    # End time for prediction

    prediction\_time = time.time() - start\_prediction\_time

    accuracy = accuracy\_score(y\_test, y\_pred)

    # Store the accuracy with its corresponding test size and predicted emotions

    results = {

        'test\_size': test\_size,

        'accuracy': accuracy,

        'predicted\_emotions': le.inverse\_transform(y\_pred),

        'training\_time': training\_time,

        'prediction\_time': prediction\_time

    }

    print(f"Accuracy: {accuracy}")

    print(f"Predicted Emotions: {results['predicted\_emotions']}\n")

    print(f"Training Time: {training\_time} seconds")

    print(f"Prediction Time: {prediction\_time} seconds\n")

    return accuracy, pipeline, le, results

* 1. **Calling SVM & Random Forest for Results**

if \_\_name\_\_ == "\_\_main\_\_":

    dataset\_path = r"C:\ML\_DL\_Files\dataset\Person\_1"  # Replace with your dataset path

    print("------------------Loading dataset---------------")

    print()

    start\_loading\_time = time.time()

    X, y = load\_dataset(dataset\_path)

    loading\_time = time.time() - start\_loading\_time

    print(f"Dataset loaded in {loading\_time} seconds")

    best\_accuracy\_rf = 0

    best\_accuracy\_svm = 0

    best\_model\_rf, best\_model\_svm = None, None

    best\_encoder\_rf, best\_encoder\_svm = None, None

    all\_results\_rf = []  # Store all accuracy results for Random Forest

    all\_results\_svm = []  # Store all accuracy results for SVM

    # Evaluate Random Forest

    for test\_size in [0.1, 0.2, 0.3]:

        print(f"Evaluating Random Forest with test size {test\_size \* 100}%")

        accuracy\_rf, model\_rf, label\_encoder\_rf, results\_rf = train\_and\_evaluate\_random\_forest(X, y, test\_size)

        all\_results\_rf.append(results\_rf)  # Append the results for Random Forest

        if accuracy\_rf > best\_accuracy\_rf:

            best\_accuracy\_rf = accuracy\_rf

            best\_model\_rf, best\_encoder\_rf = model\_rf, label\_encoder\_rf

    # Evaluate SVM

    for test\_size in [0.1, 0.2, 0.3]:

        print(f"Evaluating SVM with test size {test\_size \* 100}%")

        accuracy\_svm, model\_svm, label\_encoder\_svm, results\_svm = train\_and\_evaluate\_svm(X, y, test\_size)

        all\_results\_svm.append(results\_svm)  # Append the results for SVM

        if accuracy\_svm > best\_accuracy\_svm:

            best\_accuracy\_svm = accuracy\_svm

            best\_model\_svm, best\_encoder\_svm = model\_svm, label\_encoder\_svm

    # Print all accuracy results for Random Forest and SVM

    print("\nAll Accuracy Results for Random Forest:")

    for result\_rf in all\_results\_rf:

        print(f"Test Size: {result\_rf['test\_size'] \* 100}% - Accuracy: {result\_rf['accuracy']}")

        print(f"Predicted Emotions: {result\_rf['predicted\_emotions']}\n")

    print("\nAll Accuracy Results for SVM:")

    for result\_svm in all\_results\_svm:

        print(f"Test Size: {result\_svm['test\_size'] \* 100}% - Accuracy: {result\_svm['accuracy']}")

        print(f"Predicted Emotions: {result\_svm['predicted\_emotions']}\n")

    print("Best accuracy achieved by Random Forest:", best\_accuracy\_rf)

    print("Best accuracy achieved by SVM:", best\_accuracy\_svm)

    print("\nModel Accuracy Comparison")

    print("------------------------------------------------")

    print("Models                 | ACCURACY ")

    print("------------------------------------------------")

    print(f"Random Forest         | {best\_accuracy\_rf}")

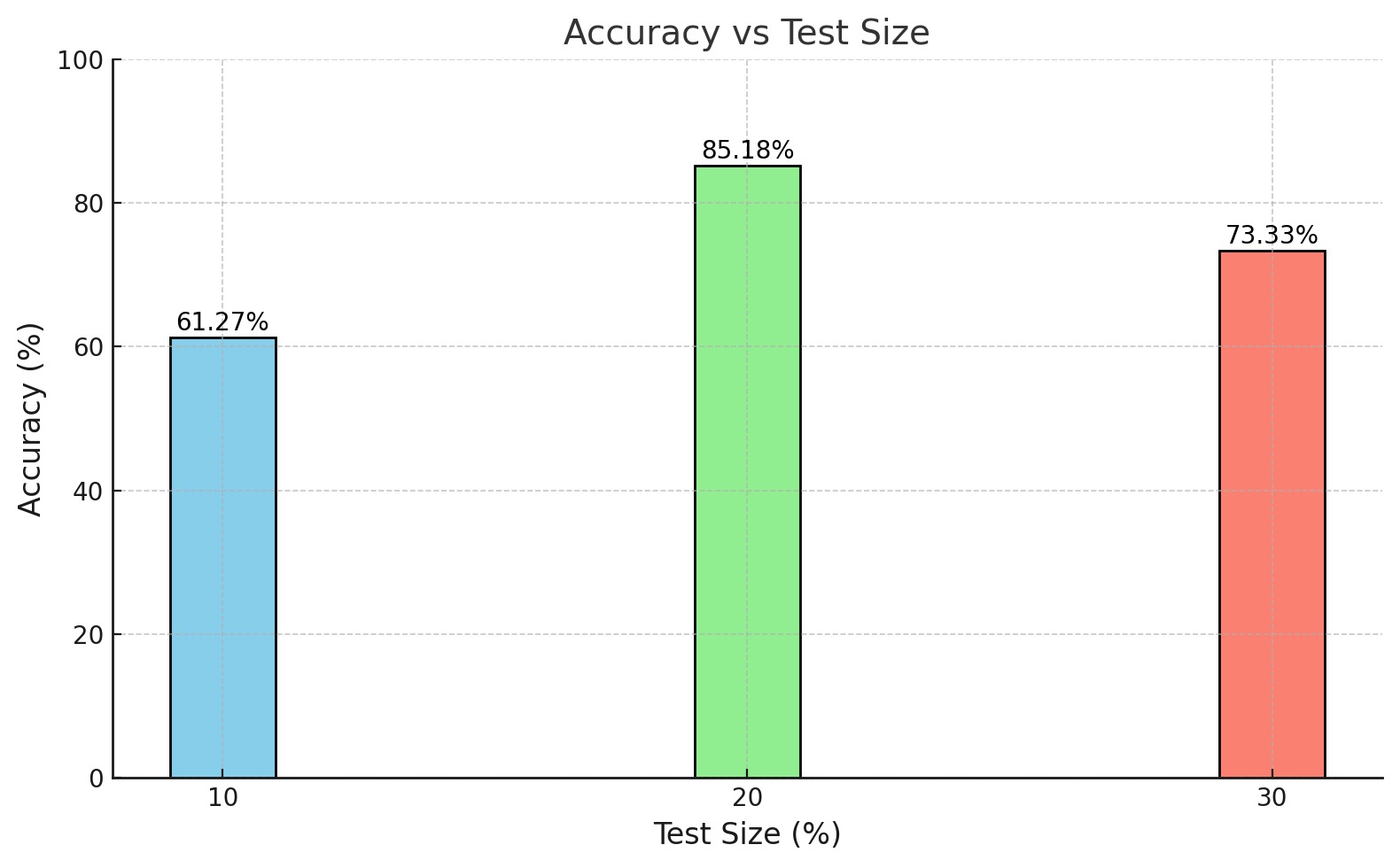
    print(f"SVM                   | {best\_accuracy\_svm}")

    # Include other models like ANN here if you have them

    print("------------------------------------------------")

* 1. **PREDICTIONS:**

***Support Vector Machine***

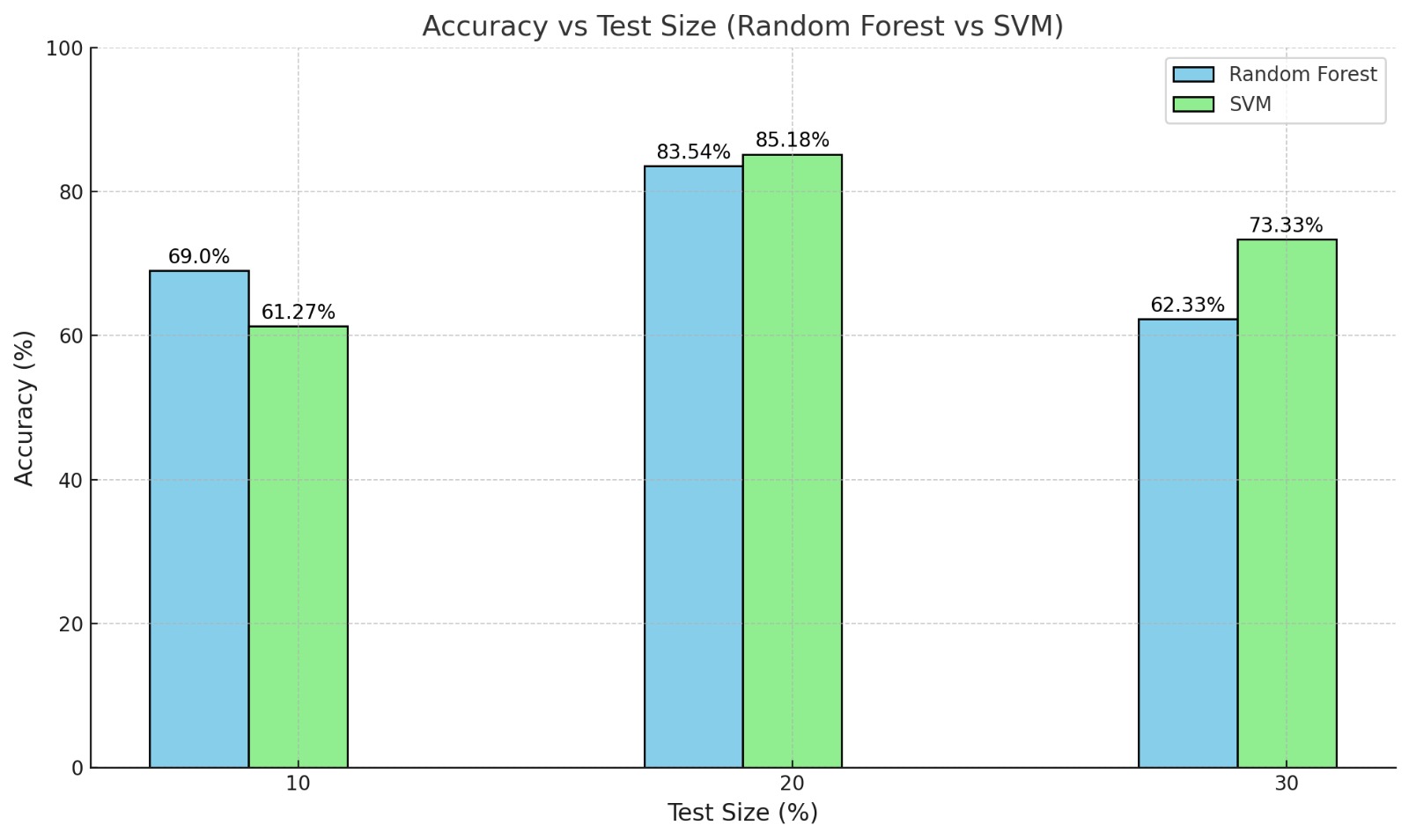


***Random Forest Classifier***

A graph showing a comparison of a test size

AI-generated content may be incorrect.

***Comparison Of Both Models***



Finally, by **Checking with Cross validation** For Both Support Vector Machine & Random Forest Classifier

Best accuracy achieved by Random Forest: 0.8

Best accuracy achieved by SVM: 0.8

Model Accuracy Comparison

------------------------------------------------

Models | ACCURACY

------------------------------------------------

Random Forest | 0.8

SVM | 0.8

------------------------------------------------

1. **REFERENCES**
2. *2019 International Seminar on Intelligent Technology and Its Applications (ISITIA). (2019). Institute of Electrical and Electronics Engineers.*
3. *2020 International Conference on Electronics and Sustainable Communication Systems (ICESC). (2020). Institute of Electrical and Electronics Engineers (IEEE).*
4. *Acheampong, F. A., Wenyu, C., & Nunoo-Mensah, H. (2020). Text-based emotion detection: Advances, challenges, and opportunities. In Engineering Reports (Vol. 2, Issue 7). John Wiley and Sons Inc. https://doi.org/10.1002/eng2.12189*
5. *Binali, H., Wu, C., & Potdar, V. (2010). Computational approaches for emotion detection in text. 4th IEEE International Conference on Digital Ecosystems and Technologies (IEEE DEST 2010).* [*https://doi.org/10.1109/DEST.2010.5610638*](https://doi.org/10.1109/DEST.2010.5610638)
6. *Kedar, S. V., Bormane, D. S., Dhadwal, A., Alone, S., & Agarwal, R. (2015). Automatic emotion recognition through handwriting analysis: A review. 2015 International Conference on Computing Communication Control and Automation, 811-816.* [*https://doi.org/10.1109/ICCUBEA.2015.162*](https://doi.org/10.1109/ICCUBEA.2015.162)
7. *Bay Ayzeren, Y., Erbilek, M., & Celebi, E. (2019c). Emotional State Prediction from Online Handwriting and Signature Biometrics. IEEE Access, 7, 164759–164774. https://doi.org/10.1109/ACCESS.2019.2952313*
8. *Desai, V. D. (n.d.). Cosmos Multidisciplinary Research E-Journal A Review on Prediction of Human Behavior Using Signature. II-Issue III. www.cmrj.in*
9. *Eldho, N. (n.d.). Emotion Detection Using Deep Learning Models on Speech and Text Data MSc Research Project MSc Data Analytics.*
10. *Emotional state detection. (n.d.).*
11. *Fairhurst, M., Erbilek, M., & Li, C. (2015a). Study of automatic prediction of emotion from handwriting samples. IET Biometrics, 4(2), 90–97. https://doi.org/10.1049/iet-bmt.2014.0097*
12. *Fairhurst, M., Erbilek, M., & Li, C. (2015b). Study of automatic prediction of emotion from handwriting samples. IET Biometrics, 4(2), 90–97. https://doi.org/10.1049/iet-bmt.2014.0097*
13. *Ghanim, T. M., Khalil, M. I., & Abbas, H. M. (2020). Comparative Study on Deep Convolution Neural Networks DCNN-Based Offline Arabic Handwriting Recognition. IEEE Access, 8, 95465–95482. https://doi.org/10.1109/ACCESS.2020.2994290*
14. *Ghosh, R., & Sinha, D. (2019). Human emotion detection based on questionnaire and text analysis. International Journal of Work Organisation and Emotion, 10(1), 66–89. https://doi.org/10.1504/IJWOE.2019.101753*
15. *Gu Su-Lim Tan Kai-Juan Wong, Y., & Ringo Ho Li Qu, M.-H. (n.d.). A Biometric Signature Based System For Improved Emotion Recognition Using Physiological Responses From Multiple Subjects.*
16. *Guo, J. (2022). Deep learning approach to text analysis for human emotion detection from big data. Journal of Intelligent Systems, 31(1), 113–126. https://doi.org/10.1515/jisys-2022-0001*
17. *Gupta, S., & Prasad, S. (2014). Human-signature psychology relation: A time-series analysis. ACM International Conference Proceeding Series, 11-16-November-2014. https://doi.org/10.1145/2677855.2677857*
18. *Kedar, S. v., & Rokade, S. (2019). Recognition of emotional state based on handwriting analysis and psychological assessment. International Journal of Engineering and Advanced Technology, 8(6), 4395–4402. https://doi.org/10.35940/ijeat.F8960.088619*
19. *Khan, Z. A., Xia, Y., Aurangzeb, K., Khaliq, F., Alam, M., Khan, J. A., & Anwar, M. S. (2024). Emotion detection from handwriting and drawing samples using an attention-based transformer model. PeerJ Computer Science, 10. https://doi.org/10.7717/peerj-cs.1887*
20. *Machová, K., Szabóova, M., Paralič, J., & Mičko, J. (2023). Detection of emotion by text analysis using machine learning. Frontiers in Psychology, 14. https://doi.org/10.3389/fpsyg.2023.1190326*
21. *Moll, J., Bado, P., de Oliveira-Souza, R., Bramati, I. E., Lima, D. O., Paiva, F. F., Sato, J. R., Tovar-Moll, F., & Zahn, R. (2012). A neural signature of affiliative emotion in the human septohypothalamic area. Journal of Neuroscience, 32(36), 12499–12505. https://doi.org/10.1523/JNEUROSCI.6508-11.2012*
22. *Punetha, N., Pal, A., & Sharma, V. (2020). BEHAVIOR PREDICTION OF HUMAN BY SIGNATURE AND HANDWRITING ANALYSIS USING MATLAB. International Research Journal of Engineering and Technology. www.irjet.net*
23. *Ravindra Kupekar, R. (n.d.). Conversational Emotion Recognition using Text and Audio Modalities MSc Research Project MSc. Data Analytics.*
24. *Rieck, J., Wrobel, J., Porras, A. R., McRae, K., & Gowin, J. L. (2024). Neural signatures of emotion regulation. Scientific Reports, 14(1). https://doi.org/10.1038/s41598-024-52203-3*
25. *Saarimäki, H., Gotsopoulos, A., Jääskeläinen, I. P., Lampinen, J., Vuilleumier, P., Hari, R., Sams, M., & Nummenmaa, L. (2016). Discrete Neural Signatures of Basic Emotions. Cerebral Cortex, 26(6), 2563–2573. https://doi.org/10.1093/cercor/bhv086*
26. *Seyeditabari, A., Tabari, N., & Zadrozny, W. (2018). Emotion Detection in Text: a Review. http://arxiv.org/abs/1806.00674*
27. *Shrivastava, K., Kumar, S., & Jain, D. K. (2019). An effective approach for emotion detection in multimedia text data using sequence based convolutional neural network. Multimedia Tools and Applications, 78(20), 29607–29639. https://doi.org/10.1007/s11042-019-07813-9*
28. *Stark, L., & Hoey, J. (2021). The ethics of emotion in artificial intelligence systems. FAccT 2021 - Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, 782–793. https://doi.org/10.1145/3442188.3445939*
29. *Tahir, M., Halim, Z., Waqas, M., & Tu, S. (2023). On the Effect of Emotion Identification from Limited Translated Text Samples Using Computational Intelligence. International Journal of Computational Intelligence Systems, 16(1). https://doi.org/10.1007/s44196-023-00234-5*
30. *Wang, B., Liakata, M., Ni, H., Lyons, T., Nevado-Holgado, A. J., & Saunders, K. (2019). A path signature approach for speech emotion recognition. Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH, 2019-September, 1661–1665. https://doi.org/10.21437/Interspeech.2019-2624*
31. *Yatbaz, H. Y., & Erbilek, M. (2020, April 1). Deep Learning Based Stress Prediction from Offline Signatures. 2020 8th International Workshop on Biometrics and Forensics, IWBF 2020 - Proceedings. https://doi.org/10.1109/IWBF49977.2020.9107942*