**Handwriting Classification System Documentation**

**By:**

**Muhammad Mehdy Hasnain (21L-1784)**

**Umair Bin Asim (21L-1847)**

**Hasnain Fatmi (21L-1773)**

**Section – 6F**

**Introduction**

The Handwriting Classification System is designed to classify handwritten images into three distinct categories: Umair, Hasnain, and Mehdy. Leveraging machine learning techniques, this system aims to accurately identify the author of a given handwritten note. By analyzing textural features extracted from handwritten images, the system offers applications in forensic analysis, document verification, and handwriting recognition technologies.

**Dataset Description and Features**

The dataset consists of handwritten images collected from three individuals, each contributing 100 images. The images are labeled according to the author's name and image number (e.g., "Umair\_img (1).jpg", "Hasnain\_img (1).jpg", "Mehdy\_img (1).jpg"). Features extracted from each image include contrast, dissimilarity, homogeneity, energy, and correlation, calculated using the Grey-Level Co-occurrence Matrix (GLCM) method.

**Imported Libraries**

The project utilizes several Python libraries for various tasks:

* os: Interacts with the operating system for file management.
* numpy: Performs numerical computations and array manipulation.
* PIL: Handles image processing tasks such as loading and converting images.
* matplotlib.pyplot: Creates visualizations like plots and charts.
* pandas: Manipulates structured data for analysis.
* seaborn: Provides statistical data visualization capabilities.
* skimage.feature: Extracts features from images using scikit-image.
* sklearn: Implements machine learning algorithms and tools.
* joblib: Saves and loads Python objects, useful for saving trained models.

**Paths Definition**

File paths are defined for storing original images and extracted features. This ensures organized data management and facilitates access to data and results throughout the project.

**Data Preprocessing and Feature Extraction**

Images are preprocessed by converting them to grayscale, simplifying subsequent analysis. Features are extracted from the grayscale images using the GLCM method, capturing textural information such as contrast, dissimilarity, homogeneity, energy, and correlation.

**Model Training**

The dataset is split into training and testing sets to evaluate the model's performance. Features are scaled using StandardScaler to improve convergence and performance. A K-Nearest Neighbors (K-NN) classifier is trained using GridSearchCV for hyperparameter tuning, optimizing parameters such as the number of neighbors and distance metric.

**Model Evaluation**

The Handwriting Classification System evaluates the model's performance using essential classification metrics, including accuracy, precision, recall, and F1-score. These metrics offer comprehensive insights into the model's ability to correctly classify handwritten images. The classification report provides a detailed breakdown of these metrics for each class, facilitating a nuanced understanding of the model's strengths and weaknesses. Additionally, the confusion matrix visually compares the model's predictions to the actual labels, offering further granularity in assessing its performance across different classes.

**Writer Prediction**

The trained model is applied to predict the author of single handwritten images, offering practical applications in document verification and handwriting recognition technologies.

**Outliers Detection**

Outliers in the feature space are detected using the Interquartile Range (IQR) method, ensuring data quality and reliability.

**Data Analysis and Visualization**

Various visualization techniques such as histograms, box plots, and pie charts are employed to explore the dataset visually, gaining insights into its characteristics and distributions.

**Visual Components**

**Home Page:**

The Home Page serves as the gateway to the Handwriting Classification System, featuring an intuitive and visually appealing design. Users are greeted with a well-designed interface that invites them to upload an image or handwritten text for processing. The layout is clean and user-friendly, guiding users through the process effortlessly. Clear instructions prompt users to upload their content, ensuring a seamless user experience.

**Transition Page:**

As users await the results of their handwriting processing task, they are directed to the Transition Page. This page acts as a bridge between the Home Page and the Result Page, providing users with visual feedback and engaging content during the processing period. A dynamic loading bar indicates the progress of the processing task, keeping users informed about the status. Additionally, the Transition Page offers interesting facts about handwriting and tips for improving handwriting. This content enriches the user experience and encourages interaction while users wait for the results.

**Result Page:**

Upon completion of the handwriting processing task, users are redirected to the Result Page, where they receive the outcome of the classification process. The Result Page features a clean and concise layout, prominently displaying the name of the individual whose handwriting has been identified. However, to ensure transparency and manage user expectations, a disclaimer is prominently featured on the Result Page. The disclaimer emphasizes that while the classification result is highly accurate, it may not be 100% precise. This disclaimer serves to instill confidence in the system's capabilities while acknowledging the inherent limitations of handwriting analysis.

**Saving the KNN Model**

The trained model and scaler are saved for future use, allowing for deployment in production environments or sharing with other team members. This model is integrated into an application or Webserver for further usage in prediction of writer.

**Methodology**

For our project, we have started by collecting a dataset comprising handwriting samples from three distinct categories: Umair, Hasnain, and Mehdy. These samples will encompass a variety of writing styles and variations to ensure comprehensive coverage. Subsequently, we will preprocess the data by normalizing or resizing the images, applying grayscale filtering to standardize the representation, and removing any noise or artifacts that may affect model performance. To ensure uniformity in input dimensions, we will scale them to be of equal dimensions. Following data preprocessing, we will extract features using grey-scale occurrence matrix filter which have proven effective in similar applications. Then, we will train a K-Nearest Neighbors (K-NN) classifier on the extracted features, tuning hyperparameters for optimal performance. Model evaluation will involve assessing performance metrics on a testing set to ensure robustness and generalization.

**Conclusion**

The Handwriting Classification System offers a robust solution for identifying the author of handwritten notes. By leveraging image processing and machine learning techniques, the system achieves accurate and reliable results, with applications in various domains including forensic analysis, document verification, and handwriting recognition technologies.