Computer Vision

OCR: Handwritten Text Recognition

Aakash Saplya (B20BB052)
Tejas Patange (B20BB043)
Rahul Saha (B20CS081)
Yogesh Jangir (B21CS083)

Introduction

Handwritten Text Recognition (HTR) is a specialized branch of Optical Character Recognition (OCR) that focuses on recognizing handwritten text from images. Unlike printed text, handwritten text varies significantly in style, size, and shape, making it more challenging.

This project will focus on building an HTR system that can accurately recognize handwritten text from images. To build a robust and accurate HTR system, we will explore various techniques, including preprocessing, text detection, segmentation, feature extraction, and text recognition. Additionally, we will leverage machine learning and deep learning approaches to improve the accuracy and performance of our HTR model.

Problem Statement

The project aims to create an Optical Character Recognition (OCR) system specifically designed for recognizing English characters. The system will take a text written in pure English as input and produce recognizable English characters as output.

- The input image should exclusively consist of English alphabets. The limitation is imposed due to the reliance of the method on the geometrical, topological, and distributional aspects of various English letters.
- The characters should maintain a distance from each other, avoiding any contact. Segmenting touching characters has consistently posed a significant challenge in the field of text recognition.

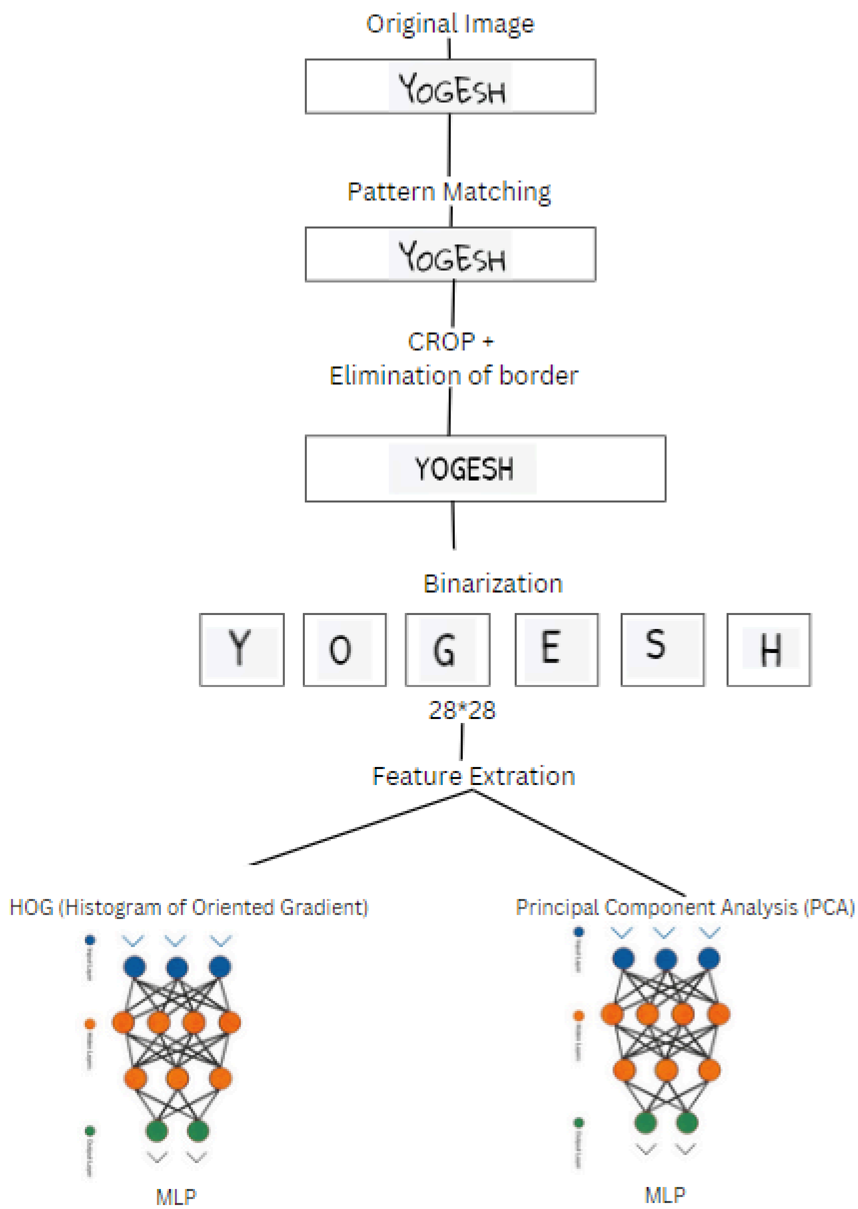
- The phrases entered into the OCR should not have significant inclinations. Their alignment should be almost straight.
- Another assumption regarding the input picture of English text is that it should be free of any noise. This assumption does not simplify the problem's complexity, as it is only a component of the preprocessing module. To eliminate noise from a picture, one can employ conventional functions and procedures.

Methodology

- Retrieve the data from the provided links:
  https://www.kaggle.com/datasets/crowdflower/handwritten-names/code

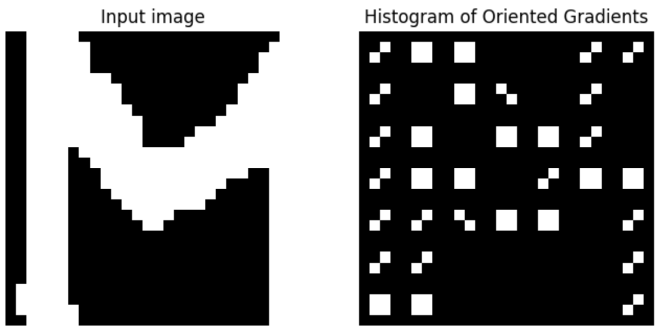
- This dataset contains links to images, but not the actual picture content. Therefore, we need to transform these links into the corresponding images.
- Convert the image into grayscale.
- Resize images to 28x28 pixels.
- Models are used:
  - **MLP with HOG features**
  - **MLP with PCA features**
- Histogram of Oriented Gradients (HOG):
  - Compute HOG features for each image using the hog() function from the scikit-image library.
  - Define parameters such as the number of orientations, pixels per cell, and cells per block.
- Principal Component Analysis (PCA):
  - Perform PCA on the HOG features to reduce dimensionality.
  - Choose the number of principal components that capture most of the variance in the data.
- Model Training
  - Split the dataset into training and testing sets.
  - Train the MLP classifier using the HOG and PCA features.
  - Optimize the parameters of the MLP classifier using techniques such as grid search and cross-validation.

- Model Evaluation
  - Evaluate the trained model using accuracy as the performance metric.
  - Visualize the performance of the model using confusion matrices.
- Results
  - Measure the performance of the MLP classifier with HOG and PCA features.
  - Compare the performance with other feature extraction techniques.



Results

- We calculated the Histogram of Oriented Gradients for the MLP_HOG classifier. Following is a hog-transformed image of M.



- We transform the data to extract Principal Component Analysis features for the MLP_PCA classifier.

- Testing individual character recognition scores for each classifier using the test batch.

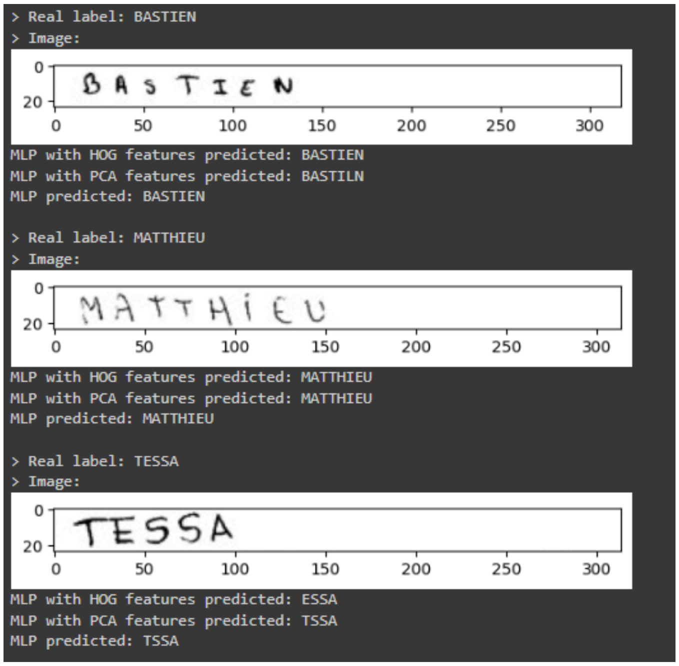
HOG + MLP classification:

	precision	recall	f1-score	support
-	0.53	0.33	0.41	24
A	0.93	0.91	0.92	1686
B	0.65	0.76	0.70	98
C	0.90	0.90	0.90	305
D	0.81	0.77	0.79	183
E	0.91	0.96	0.93	1471
F	0.68	0.41	0.51	61
G	0.79	0.86	0.83	110
H	0.83	0.79	0.81	327
I	0.89	0.91	0.90	979
J	0.89	0.78	0.83	109
K	0.84	0.75	0.79	48
L	0.95	0.93	0.94	1000
M	0.85	0.86	0.85	571
N	0.90	0.95	0.92	1060
O	0.87	0.91	0.89	598
P	0.68	0.85	0.75	93
Q	0.67	0.57	0.62	14
R	0.90	0.89	0.90	537
S	0.92	0.95	0.93	402
T	0.96	0.88	0.92	518
U	0.91	0.84	0.87	374
V	0.86	0.72	0.79	105
W	0.81	0.68	0.74	38
X	0.90	0.88	0.89	102
Y	0.91	0.89	0.90	163
Z	0.88	0.59	0.71	37
accuracy			0.90	11013
macro avg	0.84	0.80	0.81	11013
weighted avg	0.90	0.90	0.90	11013

PCA + MLP classification:

	precision	recall	f1-score	support
-	0.50	0.29	0.37	24
A	0.94	0.93	0.93	1686
B	0.89	0.71	0.79	98
C	0.89	0.92	0.90	305
D	0.78	0.76	0.77	183
E	0.94	0.94	0.94	1471
F	0.60	0.74	0.66	61
G	0.93	0.80	0.86	110
H	0.76	0.81	0.78	327
I	0.89	0.91	0.90	979
J	0.87	0.83	0.85	109
K	0.81	0.73	0.77	48
L	0.96	0.94	0.95	1000
M	0.87	0.87	0.87	571
N	0.94	0.93	0.93	1060
O	0.88	0.93	0.90	598
P	0.80	0.76	0.78	93
Q	0.69	0.64	0.67	14
R	0.88	0.90	0.89	537
S	0.92	0.95	0.93	402
T	0.95	0.94	0.94	518
U	0.85	0.93	0.89	374
V	0.94	0.75	0.84	105
W	0.72	0.74	0.73	38
X	0.95	0.82	0.88	102
Y	0.93	0.92	0.93	163
Z	0.82	0.76	0.79	37
accuracy			0.91	11013
macro avg	0.85	0.82	0.83	11013
weighted avg	0.91	0.91	0.91	11013

- some predictions are visualized here. Predictions of each classifier



Observations

1. Both models (HOG + MLP and PCA + MLP) have achieved relatively high accuracies, with the PCA + MLP model slightly outperforming the HOG + MLP model.

2. The precision, recall, and F1-score metrics vary across classes, indicating that the models perform differently for different classes.

3. Class '-' has the lowest precision, recall, and F1-score in both models, indicating that it is the most challenging class to classify.

4. Classes A, E, L, and N generally have high precision, recall, and F1 scores in both models, indicating that they are well-classified.

5. Classes F, G, H, P, Q, U, V, W, X, Y, and Z have varied performance across precision, recall, and F1 scores in both models, indicating that they might be more challenging to classify accurately.

6. Classes B, C, D, I, J, K, M, R, S, and T generally have good performance in terms of precision, recall, and F1 scores, but there are some variations across the metrics.

Conclusion

1. Both the HOG + MLP and PCA + MLP models demonstrate strong performance overall, with accuracies around 90%.

2. The PCA + MLP model slightly outperforms the HOG + MLP model, suggesting that the PCA dimensionality reduction technique might have captured more relevant information for classification compared to the HOG feature extraction method.

3. Classes with high precision, recall, and F1-scores are well-classified by both models, indicating robustness in recognizing these classes.

4. Classes with lower precision, recall, and F1-scores might require further investigation and possibly additional feature engineering or model tuning to improve classification performance.

5. Overall, both models show promise for classification tasks, but further analysis and refinement could potentially improve performance, especially for challenging classes.

Reference

https://pyimagesearch.com/2020/08/17/ocr-with-keras-tensorflow-and-deep-learning/

https://www.kaggle.com/datasets/crowdflower/handwritten-names/data