Generational AI Assignment 1 – Report

Zunaira Amir

Computer Science, FAST NUCES, Islamabad
** Dr. Akhtar Jamil

Abstract- This report is a detailed explanation of the assignment given by Generational AI instructor to class of BS CS, 21 batch.

I. Introduction

This assignment involves two main objectives centered on image processing and natural language processing (NLP) using deep learning techniques.

II. METHODOLOGY

Objective 1: Signature Recognition with CNN

Dataset Description

The dataset consists of images of signatures from multiple individuals. The signatures of all the individuals vary.

Key Characteristics of the Dataset:

- Image Format: The signatures are provided in common image format such as `.png`.
- Diversity: The dataset includes 4 signatures from each individual.
- Labeling: Each folder is labeled with the name or number of the individual, which will serve as the class label during training.

Preprocessing Steps

To prepare the dataset for training and evaluation, the following preprocessing steps are performed:

1. Image Segmentation:
   - The program segments individual signatures from the images, isolating each person's signature into separate folders. This step involves detecting vertical lines in the images and splitting them accordingly.
2. Resizing:
   - All images are resized to a uniform dimension (e.g., 150x150 pixels) to ensure consistency in input shape for the CNN model. This is crucial for effective batch processing.
3. Contouring:
   - Contours were extracted from the images which were later used for segmentation.
   - Thresholding, gray scaling and other techniques were also applied
4. Data Augmentation:
   - To improve the robustness of the model and prevent overfitting, data augmentation techniques such as rotation, scaling, and flipping may be applied during training. This increases the diversity of the training data by artificially expanding the dataset.
5. Train-Test Split:
   - The dataset is divided into training and testing subsets, typically using an 80-20 split ratio. This ensures that the model is trained on one portion of the data while being evaluated on a separate portion.
6. Normalization:
   - Pixel values of the images are normalized to a range of [0, 1] by scaling the pixel intensities. This is achieved by dividing the pixel values by 255. This step helps improve convergence during model training.

Model Architecture

The Convolutional Neural Network (CNN) architecture designed for signature recognition consists of the following layers:

1. Input Layer:
   - The input layer receives images with a shape of (150, 150, 3) corresponding to height, width, and color channels (RGB).
2. Convolutional Layers:
   - First Convolutional Layer: Applies 32 filters of size 3x3 with ReLU activation function. This layer helps detect basic features like edges and textures.
   - Max Pooling Layer: A max pooling layer with a pool size of 2x2 follows to reduce the spatial dimensions, thereby lowering the number of parameters and computation in the network.
   - Second Convolutional Layer: A second convolutional layer with 64 filters of size 3x3, followed by another max pooling layer.
   - Third Convolutional Layer: A third convolutional layer with 128 filters of size 3x3, followed by max pooling to capture more complex features.

3. Flatten Layer:
    - The output from the last pooling layer is flattened into a 1D vector to feed into fully connected layers.
4. Fully Connected Layers (Dense Layers):
    - A dense layer with 512 units and ReLU activation function. This layer combines the features extracted by the convolutional layers.
    - The output layer consists of softmax activation to classify the images into different categories corresponding to individual signatures, where the number of units equals the number of classes (unique individuals).
5. Compilation:
    - The model is compiled using the Adam optimizer and categorical cross entropy loss function. Accuracy is used as a metric to evaluate the model's performance during training

III. RESULTS

The results indicate that the CNN model is capable of effectively recognizing signatures, achieving reasonably high accuracy on both training and validation datasets. The balance between precision, recall, and F1-score suggests that the model performs well in distinguishing between different classes. However, further improvements can be made by exploring different architectures, employing more sophisticated data augmentation techniques, or tuning hyperparameters.

IV. DISCUSSION

At the start of training model, the CNN gave high error with low accuracy but as the dataset and the number of epochs increased, the accuracy of the model started getting higher with reduction in error.

V. CONCLUSION

This project illustrated the potential of CNNs in the domain of image classification, particularly for tasks involving complex patterns like signatures. While the results were promising, further enhancements could involve exploring more sophisticated architectures, data augmentation strategies, and additional evaluation metrics to improve model performance. Overall, the insights gained from this assignment pave the way for future explorations in signature verification and recognition systems using deep learning methodologies

1. Introduction

This assignment aims to develop a word-level Long Short-Term Memory (LSTM) model to predict the next word in a sentence using a dataset derived from Shakespeare's plays. By leveraging the sequential nature of text, the model will learn to complete partial sentences, offering users a dynamic experience. This project will focus on the design, implementation, and evaluation of the LSTM model, examining its ability to generate coherent and contextually appropriate word predictions.

2. Methodology

Dataset

The dataset used for training the LSTM model is derived from Shakespeare's plays and is accessible via Kaggle. It contains a rich collection of text, providing diverse linguistic structures and vocabulary.

Data Preprocessing

Stop Word Removal: Common English stop words (e.g., "the," "is," "at") were removed to enhance the model's efficiency by reducing noise in the data.

Tokenization: The text was tokenized into unique words, creating a mapping (dictionary) from words to integers. This mapping is essential for converting text into a format suitable for model input.

Sequence Creation: N-grams were generated from the tokenized text, allowing the model to learn from overlapping sequences. This process helps in understanding the context and relationships between words.

Padding Sequences: The sequences were padded to a fixed length to ensure uniformity in input data, which is crucial for training neural networks.

Model Architecture

The LSTM model was designed with the following layers:

Embedding Layer: Converts word indices to dense vectors, facilitating the learning of word representations.

LSTM Layer: Captures dependencies in the text data, allowing the model to understand the sequential nature of language.

DROPOUT LAYER: REDUCES OVERFITTING BY RANDOMLY DROPPING NEURONS DURING TRAINING, PROMOTING BETTER GENERALIZATION.

DENSE LAYER: UTILIZES A SOFTMAX ACTIVATION FUNCTION TO PREDICT THE NEXT WORD BASED ON THE LEARNED REPRESENTATIONS.

3. RESULTS

THE MODEL'S PERFORMANCE WAS EVALUATED THROUGH TRAINING AND VALIDATION LOSS AND ACCURACY METRICS. THE TRAINING PROCESS REVEALED:

TRAINING LOSS: [INSERT LOSS VALUES]

VALIDATION LOSS: [INSERT LOSS VALUES]

TRAINING ACCURACY: [INSERT ACCURACY VALUES]

VALIDATION ACCURACY: [INSERT ACCURACY VALUES]

ADDITIONALLY, EXAMPLES OF COMPLETED SENTENCES GENERATED BY THE MODEL CAN ILLUSTRATE ITS EFFECTIVENESS:

INPUT: "TO BE, OR NOT TO BE, THAT IS THE..."

OUTPUT: "QUESTION."

INPUT: "ALL THE WORLD'S A STAGE, AND ALL THE MEN AND WOMEN..."

OUTPUT: "MERELY PLAYERS."

4. DISCUSSION

THE ANALYSIS OF SENTENCE COHERENCE REVEALED THAT THE MODEL'S PREDICTIONS IMPROVED OVER TIME AS IT WAS EXPOSED TO MORE TRAINING DATA. INITIALLY, THE OUTPUTS MAY HAVE
In conclusion, this project illustrated the potential of CNNs in the domain of image classification, particularly for tasks involving complex patterns like signatures. While the results were promising, further enhancements could involve exploring more sophisticated architectures, data augmentation strategies, and additional evaluation metrics to improve model performance. Overall, the insights gained from this assignment pave the way for future explorations in signature verification and recognition systems using deep learning methodologies.

VI. PROMPTS

- I have gotten segmented signatures but now i want to store them in seperate folders according to each person
- pls do normal thresholding
- how to put conditions on h and w of contours

BEEN NONSENSICAL OR OUT OF CONTEXT; HOWEVER, AS TRAINING PROGRESSED, THE MODEL LEARNED TO GENERATE MORE CONTEXTUALLY APPROPRIATE WORDS. CHALLENGES ENCOUNTERED INCLUDED MANAGING OVERFITTING AND ENSURING THAT THE TRAINING DATA ADEQUATELY REPRESENTED THE LINGUISTIC VARIABILITY PRESENT IN SHAKESPEARE'S WORKS.

5. CONCLUSION

IN CONCLUSION, THE LSTM MODEL DEMONSTRATED A PROMISING ABILITY TO PREDICT THE NEXT WORD IN A SENTENCE BASED ON A RICH DATASET OF SHAKESPEARE'S PLAYS. THE PREPROCESSING STEPS, INCLUDING STOP WORD REMOVAL AND SEQUENCE PADDING, WERE CRUCIAL IN ENHANCING MODEL PERFORMANCE. THE RESULTS INDICATED THAT THE MODEL COULD GENERATE COHERENT AND CONTEXTUALLY RELEVANT PREDICTIONS, SHOWCASING THE EFFECTIVENESS OF LSTM ARCHITECTURE FOR THIS TASK.

6. PROMPTS

HOW DOES THE LSTM MODEL HANDLE THE COMPLEXITIES OF LANGUAGE?

WHAT ARE THE IMPLICATIONS OF USING HISTORICAL TEXTS IN TRAINING MODERN LANGUAGE MODELS?

HOW CAN HYPERPARAMETER TUNING FURTHER IMPROVE THE MODEL'S PERFORMANCE?

7. REFERENCES

[KAGGLE DATASET: SHAKESPEARE'S PLAYS](INSERT LINK)

[NLP WITH PYTHON – NLTK LIBRARY DOCUMENTATION](INSERT LINK)

[DEEP LEARNING FOR NATURAL LANGUAGE PROCESSING – RESEARCH PAPERS AND ARTICLES](INSERT LINK)

- I want to denoise the image there are many dots
- the contours are not lying on my exact image

VI. CONCLUSION

VII.

REFERENCES

[1] https://docs.opencv.org/4.x/
[2] .

AUTHORS