Handwritten Text Recognition using TrOCR

Datasets

IAM Handwriting Dataset (Teklia/IAM-line):

- Sourced from Hugging Face, IAM dataset is majorly focused on Handwritten Text recognition . This includes a large set of handwritten text lines paired with transcriptions, these features made me to select for training and testing OCR models.

Imgur5K Dataset (vklinhhh/imgur5k_words):

- This one is also sourced from Hugging Face, this dataset complements IAM by adding more handwritten text images. The increased variety enhances the model's ability to generalize across different handwriting styles.

Model

TrOCR (microsoft/trocr-base-handwritten**)**:

- TrOCR is a transformer-based model optimized for OCR tasks, particularly handwritten text recognition. It combines a vision transformer encoder and a transformer decoder, making it adept at sequence-to-sequence tasks like converting images of text into readable strings.

Justification

Dataset: The combination of IAM and Imgur5K created a diverse and robust training set, critical for handling varied handwriting styles and improving model performance across different samples. Imgur5K adds more handwritten text images

Model: TrOCR's architecture is tailored for OCR, I had chosen base model opposing the large model but it still provides a strong balance of performance and computational efficiency, suitable for the task.

Preprocessing Steps and Fine-Tuning Strategy

Preprocessing Steps

1. **Image Processing**:
    - **Grayscale Conversion**: This is used to convert Images to grayscale to reduce complexity and focus on text features.
    - **Noise Reduction**: I had used fast non-local means denoising (h=10) and median blurring (kernel size 3) to remove noise while preserving text details.
    - **Adaptive Thresholding**: Used Gaussian adaptive thresholding to enhance text contrast against the background.
    - **Perspective Augmentation**: Random perspective transformations are applied with a probability of 0.5 to simulate variations in handwriting angles.
    - **Resizing**: Images are resized to 224x224 pixels (from 384x384) to standardize input for the model.
2. **Text Processing**:
    - Text labels are tokenized using the TrOCR processor, converting them into token IDs with a maximum length of 64, padded as needed.

Fine-Tuning Strategy

- **Training Configuration**:
    - **Batch Size**: Dynamically set based on GPU type (4 for P100, 8 for T4s), scaled by the number of GPUs, with gradient accumulation steps adjusted accordingly.
    - **Learning Rate**: Set to 5e-5 with a warmup ratio of 0.1 for stable convergence.
    - **Mixed Precision Training (fp16)**: Enabled to optimize memory usage and speed up training.
    - **Gradient Checkpointing**: Used to manage memory constraints during training.
    - **Early Stopping**: Implemented with a patience of 3 epochs to prevent overfitting, using CER as the key metric.

  - **Epochs**: Maximum of 10 epochs, with evaluation and checkpoint saving at the end of each epoch.
- **Evaluation**:
  - Metrics include Character Error Rate (CER) and Word Error Rate (WER), calculated on the validation set during training and the test set for final evaluation.

- ## Final CER and WER Scores on the Test Set

Final CER

Epoch	**Training Loss**	**Validation Loss**	**Cer**
1	No log	0.000000	1.000000
2	No log	0.000000	1.000000
3	No log	0.000000	1.000000
4	0.103700	0.000000	1.000000

Challenges Faced and Potential Improvements

Challenges

Dataset Variability:

1. The diverse handwriting styles and image qualities in IAM and Imgur5K posed challenges in achieving uniform performance.
2. While Importing Required Libraries like Hugging Face Hub.

Memory Management:

- Large images and batch sizes strained GPU memory, requiring gradient accumulation and checkpointing.

Potential Improvements

1. **Enhanced Data Augmentation**:
    - Introduce elastic distortions or synthetic noise to further improve robustness to handwriting variations.
    - Even I had done the data augmentation as much as possible but , introducing required noise helps the model to learn the non-linearity

Model Exploration:

- Testing with the larger TrOCR variant (trocr-large-handwritten) for potentially better accuracy as suggested in the assessment if it is computationally available

Post-Processing:

- Adding a spell-checking layer or language model to refine predictions and lower WER.