# TrOCR: Transformer-based Optical Character Recognition

## What is TrOCR?

TrOCR (by Microsoft) is a VisionEncoderDecoder model that combines:

- A Vision Transformer (ViT) as an image encoder

- A Transformer decoder (like BART or GPT2) for generating text.

It is pre-trained on printed and handwritten text images and fine-tuned for Optical Character Recognition (OCR) tasks.

## How TrOCR Works on Handwritten Documents & Images

The notebook demonstrates how TrOCR works step-by-step:

1. Load pretrained model using `TrOCRProcessor` and `VisionEncoderDecoderModel`.

2. Load and preprocess handwritten images with `PIL`.

3. Use the processor to convert image to tensor, and model to generate text.

4. Use `jiwer` for accuracy evaluation using CER and WER.

## Accuracy

TrOCR achieves State-of-the-Art (SOTA) results:

- IAM (handwritten): CER as low as 2.25%

- Uses `jiwer` for CER and WER metrics.

## Performance & Latency

Inference Speed:

- CPU: ~2–5 seconds/image

- GPU (T4/V100): < 1 second/image

Performance bottlenecks: large image sizes and model size.

## Parallel Processing Feasibility

TrOCR supports batch and parallel inference:

- Batch processing using `processor(images=[...])`

- Parallelism using multiprocessing or GPU threads.

## Costing

- Model Size: ~1.2 GB

- Deployment Examples:

- CPU (t3.medium): ~$0.0416/hr

- GPU (g4dn.xlarge): ~$0.526/hr

- Suitable for low-volume single image inference or high-throughput batch jobs.

## Infrastructure Requirements

|  |  |
| --- | --- |
| Component | Requirement |
| Python Version | >=3.7 |
| Memory | Minimum 4GB RAM, recommended 8GB+ |
| GPU | Recommended (T4/V100) |
| Storage | ~2–3GB for model and dependencies |
| Libraries | transformers, torch, Pillow, jiwer |

## Summary

|  |  |
| --- | --- |
| Feature | Details |
| Trained on | Handwritten and printed text datasets |
| Best Use Case | Scanned handwritten forms, notes, documents |
| Performance | SOTA on multiple OCR benchmarks |
| Parallelizable? | Yes, via batch or parallel jobs |
| Cost | Moderate (GPU recommended for speed) |
| Deployment | Python API, Docker, or REST |