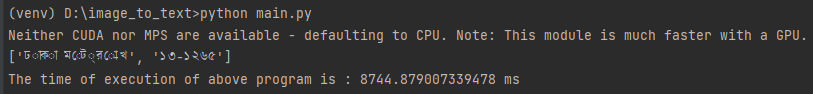
A close-up of a sign

Description automatically generated

**EasyOCR**

A screen shot of a computer program

Description automatically generated



**Explanation:** **EasyOCR** requires GPU to execute each pixels using parallel computing, if GPU is unavailable then it uses CPU which takes longer times for execution as CPU is limited in the concurrency of tasks that can be running

**Time:** The time of execution of **EasyOCR** program is 8444.87 ms which is more than 8 seconds

**TesseractOCR**

A screen shot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated

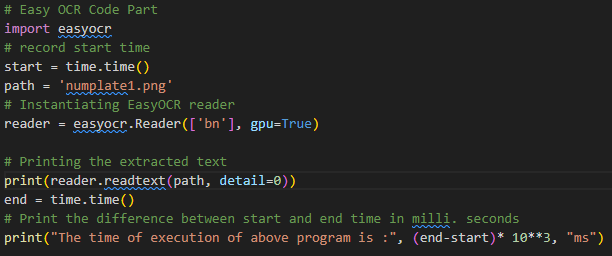


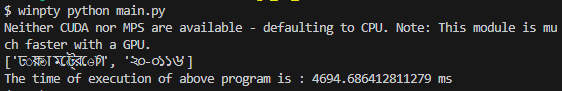
**Explanations**: TesseractOCR requires good quality image in order to get accurate result as it does not use GPU to read all the pixels using parallel computing. It uses multithreading to calculate lesser number matrices of the picture

**Result:** The time of execution of TesseractOCR program is **72.06** milli seconds which is 0.072 seconds.



**EasyOCR**



  
**Explanation:** **EasyOCR** requires GPU to execute each pixels using parallel computing, if GPU is unavailable then it uses CPU which takes longer times for execution as CPU is limited in the concurrency of tasks that can be running

**Result**: The time of execution of EasyOCR program is 4694.69 ms

**TesseractOCR**

A computer screen shot of text

Description automatically generated

A black screen with white text

Description automatically generated

**Explanations**: TesseractOCR requires good quality image in order to get accurate result as it does not use GPU to read all the pixels using parallel computing. It uses multithreading to calculate lesser number matrices of the picture

**Result:** The time of execution of TesseractOCR program is **103.10** ms which is 0.1 Seconds 1/10 of a second

Verdict on using the best suitable library to read number is Tesseract OCR. In python we used **“pytesseract”** library cause it does not occupy much resource from hardware and does not require gpu to run smoother and also execution time is very lower than using **“EasyOCR”.**

**General comparison between**

1. **Algorithm Optimization:** Tesseract may have implemented certain optimization techniques or algorithms that make it more efficient in terms of speed. The underlying OCR algorithms, language models, and image processing methods can greatly influence performance.
2. **Multithreading and Parallel Processing**: The ability to leverage multiple threads or parallel processing can significantly enhance the speed of OCR engines. Tesseract might be better optimized for parallelization, allowing it to process images faster, especially on multi-core systems.
3. **Community Contributions and Support**: Tesseract has a large and active open-source community, which often leads to regular updates, bug fixes, and performance improvements. If a community actively contributes to the development and optimization of Tesseract, it may result in better overall performance.
4. **Integration with External Libraries**: Tesseract is integrated with optimized external libraries for image processing, which can enhance its speed. The choice of libraries and their efficiency can impact the overall performance of an OCR engine.
5. **Focused Development**: Tesseract have a specific focus on optimizing speed and performance. Depending on the priorities of the development teams behind Tesseract and EasyOCR, one may prioritize speed optimizations more than the other.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **OCR** | **Most Dependent Machine** | **Time(average)** | **Accuracy** | **Ease of use** |
| TesseractOCR | CPU | 100 ms | Need more good quality picture in order to get more accurate picture | #Works better on Character level  #Complex on implementing  # Takes lesser time to get result |
| EasyOCR | CPU/GPU | 5000ms | Slicing the picture then compute the patterns using parallel computing | #Works better on word level  #Easier to implement  #Takes longer time to get result |