# A Comprehensive Report on My Summer Internship Journey and Experience

## Introduction

To start with, I am Bereket Siraw, a rising sophomore pursuing a degree in Computer Science at New York University. The opportunity to contribute to the future advancement of the Ethiopian Artificial Intelligence institution in the field of AI has been incredibly fulfilling. I would like to extend my gratitude to Dr. Taye Girma for accepting my internship request, to Sisay for his unwavering kindness and support throughout the internship duration, to Dr. Rosa for her collaborative efforts and provision of essential resources for the model, and last but certainly not least, I am thankful to all my dedicated coworkers who have been invaluable collaborators on this journey.

## Objectives and Goals

- Dataset Collection and Preparation
- Model Training
- Evaluating Model Performance
- Developing an Accurate Tesseract-Based OCR Model for Typewriter Amharic Corpus

## Internship Activities

### Week-1 / Week-4

- In an effort to enhance the ongoing model, I undertake on a thorough analysis to identify the factors hindering its optimal performance. This entailed a comprehensive examination of the model's capabilities and a study of the setbacks that contributed to its underperformance.
- To address these challenges, I consolidate available resources to create a more robust and efficient model. During this phase, I was presented with a single multi-page TIF (Tagged Image Format) file alongside its corresponding box file. It's important to mention that the TIF file contained a limited dataset, comprising only 46 pages and approximately 11,000 characters. This quantity is insufficient for constructing a robust model.
- Nevertheless, I proceeded to build the initial model using the provided TIF and box files. As anticipated, this initial attempt yielded a discouraging validation character error rate of approximately 67 percent. Drawing from past experiences, wherein I encountered similar

obstacles, I opted to use another algorithm centered around the utilization of segmented characters for model training.

- This strategic approach proved to be effective, considering the challenges associated with manual box file preparation and the complexities of sourcing public domain typed scripts. By using segmented characters, I was able to generate infinity training set. And this devise a tailored solution that not only addressed the existing limitations but also paved the way for a more accurate and reliable model.
- Subsequently, I successfully generated multiple models, each demonstrating a linearly decreasing CER (Character Error Rate). Finally, I was able to reduce the validation Character Error Rate to **17** on simulated dataset and **12** percent on the wild test file.

## Technical Aspects of Model Training For Multipage TIF Files

### Step 0: Initial Data Preparation

Prior to proceeding, ensure that Tesseract is installed on your local machine. Subsequently, initiate the process by cleaning the training data. Essential langdata components, which encompass punctuation, wordlists, and numerical data, are accessible through the provided link: langdata download.

### Step 1: Download Source Repositories

Clone essential repositories from the source:

```
git clone <https://github.com/tesseract-ocr/tesseract.git>
git clone <https://github.com/tesseract-ocr/tesstrain.git>
```

### Step 2: Data and Directory Setup

**Note**: **Precise directory organization is crucial:**

- Within the `tesstrain` directory, create a new folder named `dataset`.
- Inside the `dataset` directory, create a sub-folder named `ground_truth`.
- Further, within the `dataset` directory, create another folder named `your_model_name`.
- Transfer all files from the previously acquired langdata into the newly created `your_model_name` folder.
- Move the cleaned training data to the `ground_truth` folder.

### Step 3: Model Training with Custom Parameters

Navigate to the `tesstrain` directory to start the model training, tailoring parameters as needed:

```
cd tesstrain
make training MODEL_NAME=your_model_name DATA_DIR=/dataset GROUND_TRUTH_DIR=/dataset/ground_trut
h MAX_ITERATIONS=desired_value LEARNING_RATE=0.0001
```

### Step 4: Model Evaluation

Within the `tesstrain` directory, evaluate the model's performance through the `lstmeval` tool:

```
cd tesstrain
lstmeval --model dataset/your_model_name.traineddata --eval_listfile dataset/your_model_name/all
-lstmf
```

### Step 5: Model Testing

Concluding the process, proceed to test the model on an image file:

```
cd tesstrain
tesseract image_file_name --tessdata-dir path_to_your_model -l your_model_name
```

## Technical Aspects of Model Training For Single TIF Files

### Step-0 Initial Data Preparation and Model Training

To initiate the process, ensure your training data is meticulously organized. Begin by downloading essential langdata components, encompassing punctuation, wordlists, and numerical data, from the provided link: langdata download. Subsequently, move these langdata files to your designated `your_workspace` directory.

```
cd your_workspace;
tesseract your_model_name.my.exp0.tif  your_model_name.my.exp0 box.train.stderr;
unicharset_extractor your_model_name.my.exp0.tif;
mftraining -F font_properties -U unicharset -O your_model_name.unicharset train.my.exp0.tr;
cntraining your_model_name.my.exp0.tr;
wordlist2dawg frequent_words_list your_model_name.freq-dawg your_model_name.unicharset;
wordlist2dawg words_list your_model_name.word-dawg your_model_name.unicharset;
set_unicharset_properties -U unicharset -O output_unicharset --script_dir=./ ;
mv inttemp your_model_name.inttemp;
mv pffmtable your_model_name.pffmtable;
mv shapetable your_model_name.shapetable;
mv normproto your_model_name.normproto;
combine_tessdata your_model_name.;
```

### Step-1: Model Evaluation

Advancing to evaluation, open the `tesstrain` directory. Then run the `lstmeval` tool to assess your model's efficacy, leveraging the trained data.

```
cd tesstrain
lstmeval --model dataset/your_model_name.traineddata --eval_listfile dataset/your_model_name/all
-lstmf
```

## Step-2: Model Testing

Utilize Tesseract to extract meaningful insights from image files. Execute the command below, ensuring the specified paths and language model designation are accurately aligned.

```
tesseract image_file_name --tessdata-dir path_to_your_model -l your_model_name
```