Training Tesseract 4.x (LSTM) For Bengali

# Introduction

Well, I guess [Tesseract](https://github.com/tesseract-ocr/tesseract) needs no introduction, suffice to say that, it is the best open source OCR out there, with a huge number of supported languages. Especially after the 4.x release, the Tesseract Engine has become more accurate than ever. However, it still has room for improvement in Bengali, especially when we feed in books with old print. I was trying to OCR a few scanned pages from [Mahabharat by Rajshekhar Basu](https://www.flipkart.com/mahabharat/p/itmezyv8p9d8mn4k). I was pretty disappointed when I saw the accuracy was around only 40%, even when I used the [tessdata\_best](https://github.com/tesseract-ocr/tessdata_best) for lstm. So, I decided to train and improve the existing model: easier said than done. Even though there is an excellent and exhaustive [tutorial](https://tesseract-ocr.github.io/tessdoc/TrainingTesseract-4.00.html), on this subject, I was kind of lost in the deluge of terms and details. Yes, the tutorial is too detailed. And it's very difficult to sift through the information to arrive at the magical commands that will get the job done. So, finally, after a lot of struggle, I have kind of figured out how to train the existing tessdata for Bengali with a scanned image from Mahabharat by Rajshekhar Basu, in **5 simple steps** as described below.

# Step 1: Extract recognition model

Well, like I said, I was planning to improve the existing model. So, we need to start with the existing tessdata. Since this is aimed at Tesseract 4.x with LSTM, it would only work with the tessdata\_best: <https://github.com/tesseract-ocr/tessdata_best>.

We would need to extract the recognition model. A recognition model can be extracted from an existing traineddata file. The output of this would be a lstm file.

|  |
| --- |
| combine\_tessdata -e /kaaj/installs/tesseract/tessdata\_best-4.0.0/ben.traineddata ./ben.lstm |

# Step 2: Creating lstmf file from a tiff/box pair

We would need to obtain a *lstmf* binary file from a tiff image and its corresponding box file. The below command will create a lstmf binary file, given a tiff and box file pair:

|  |
| --- |
| tesseract -l ben ben.rajshekhar\_mahabharat.exp0.jpg ben.rajshekhar\_mahabharat.exp0 --psm 6 lstm.train |

You can create multiple *lstmf* files from several tiff/box pairs.

# Step 3: Creating a list of lstmf files

Create a *ben.training\_files.txt* containing all the *lstmf* files that you have created in the previous step. The contents of this file will be the full path of each of the lstmf file:

|  |
| --- |
| /mydir/ben.rajshekhar\_mahabharat.exp0.lstmf /mydir/ben.rajshekhar\_mahabharat.exp1.lstmf /mydir/ben.rajshekhar\_mahabharat.exp2.lstmf |

# Step 4: Run the training command

Then run the below command to train:

|  |
| --- |
| lstmtraining --model\_output ./my\_output \ --continue\_from ./ben.lstm \ --traineddata /kaaj/installs/tesseract/tessdata\_best-4.0.0/ben.traineddata \ --train\_listfile ./ben.training\_files.txt \ --max\_iterations 400 |

The output from the above will be a *my\_output\_checkpoint* file.

# Step 5: Combining the outputs

In the final step, we have to combine the *my\_output\_checkpoint* file, which we obtain in the previous step, with the existing traineddata file:

|  |
| --- |
| lstmtraining --stop\_training \ --continue\_from ./my\_output\_checkpoint \ --traineddata /kaaj/installs/tesseract/tessdata\_best-4.0.0/ben.traineddata \ --model\_output /kaaj/source/tessdata\_best/ben.traineddata |

You can now use the new traineddata file by pointing the *TESSDATA\_PREFIX* environment variable to the new *tessdata\_best* directory.

# Source Files

The image and the box files can be found here:

<https://github.com/paawak/blog/tree/master/docs/tesseract/training-bengali>

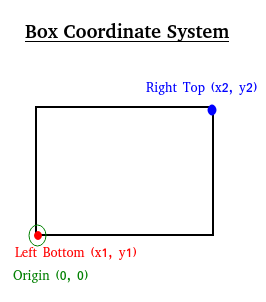
I have also checked in the [ocr-text-before.txt](https://github.com/paawak/blog/blob/master/docs/tesseract/training-bengali/ocr-text-before.txt) and [ocr-text-after.txt](https://github.com/paawak/blog/blob/master/docs/tesseract/training-bengali/ocr-text-after.txt), which are the OCR generated text *before* and *after* the training respectively.

# Addendum: Generating the Box File

For training Tesseract, creating *box files* is the first step. A *box file* is a *plain-text* file that is used to specify the text, or a character, at a given coordinate in the image. For the *lstm* system, the coordinates of an entire line is considered and **NOT** the individual coordinates of the character in the image. Note that this is [significantly different](https://github.com/tesseract-ocr/tesseract/issues/2357#issuecomment-477239316) from the earlier *Tesseract 3.x*, where the coordinates of a given character was needed. The format of the box file is:

|  |
| --- |
| <**character**> <**LeftBottom** x1> <**LeftBottom** y1> <**RightTop** x2> <**RightTop** y2> <**tiff** image page number> |

The coordinate system used in the box file has **(0,0)** at the **bottom-left** as shown below:



In a box file, every end of a word is marked with:

|  |
| --- |
| <**space**> <**LeftBottom** x1> <**LeftBottom** y1> <**RightTop** x2> <**RightTop** y2> <**tiff** image page number> |

Similarly, every end of a line is marked with:

|  |
| --- |
| <**tab**> <**LeftBottom** x1> <**LeftBottom** y1> <**RightTop** x2> <**RightTop** y2> <**tiff** image page number> |

The below command will generate a box file called my-box-file.box from an image:

|  |
| --- |
| tesseract -l ben bangla-mahabharat-1-page\_2.jpg my-box-file lstmbox |

Please note that you might have to manually edit the box file for the correctness of the text.

# Reference

1. <https://tesseract-ocr.github.io/tessdoc/TrainingTesseract-4.00.html>
2. <https://github.com/tesseract-ocr/tesseract/issues/2357#issuecomment-477239316>