Tegaki Kenshutsu

Introduction:
It would be easy if anyone in the world can write their own content in their favorite notebook and does not have to rewrite in computer. We will try to make it possible by creating software which can convert scanned handwritten pages into text. We will try to make universal which can be worked on any language.

Instead of depending on the language, our solution will be depending on the pattern of the individual writer. It will be achieved by below steps.

1. Writer will write pre-existing letters defined by our software in blank page, scan and upload.
2. Our software will link all individual characters to the actual font characters.
3. Once after reference received, writer can upload its writing scan copy.
4. Our software will try to match his writing with existing data and will give actual text.

Main advantage of this approach will be

1. knowing the individual writing style in advance which gives accuracy and speed
2. It can be language independent as it does not depend on the text but depend on the writer handwriting images.

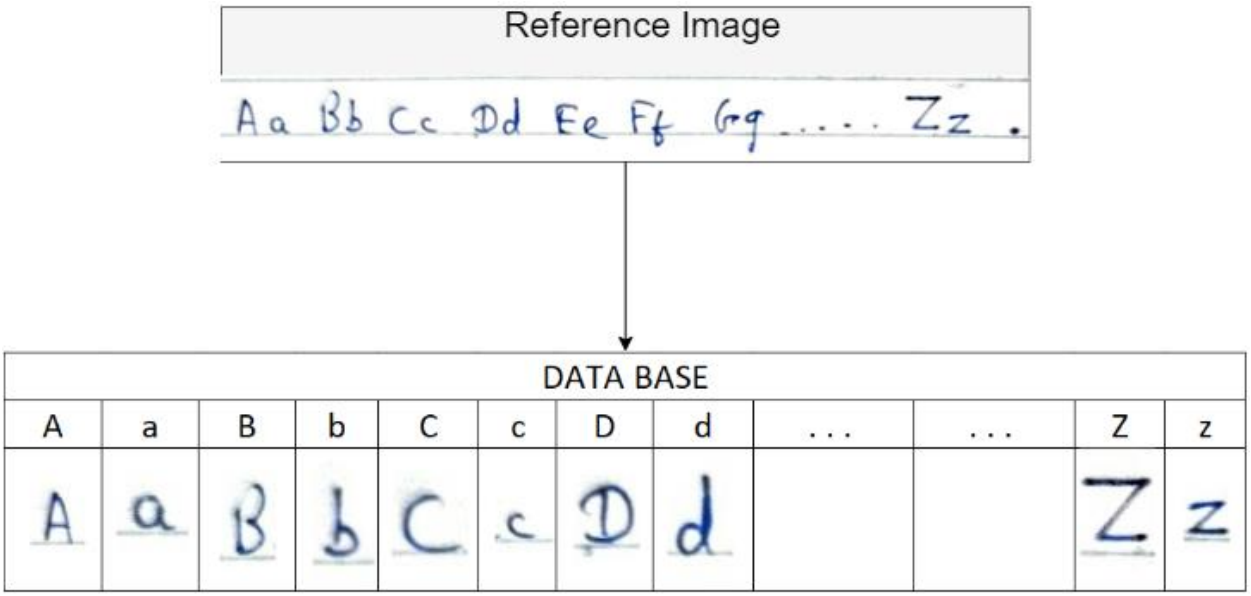
We will be doing project in two phases:
PHASE1: Overall project is split in two technical parts

1. Detecting individual characters from reference scan and create database
2. Detect actual writing scans from the database

Detecting reference scan

We will split reference scan image into small character level images and will link it with the

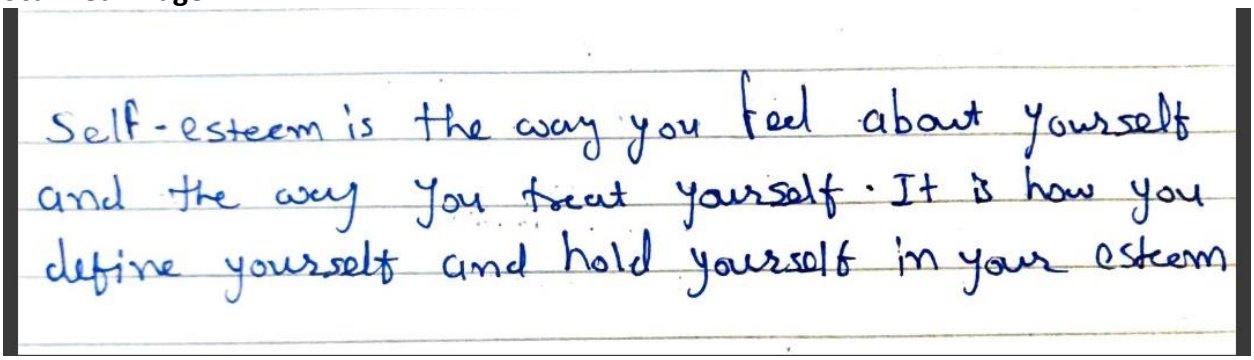
Ex:



Converting scan to actual text

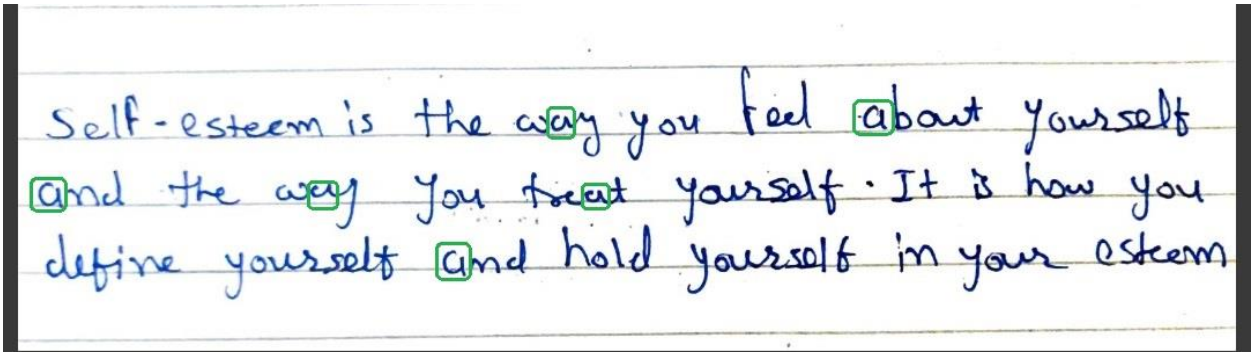
We will get actual scanned writing photo and will convert it to actual text using simple image compare mechanism in which logic falls as below.

Scanned Image:



First, we will scan all the ‘a’ in full image from above created database and plot its position in empty canvas.

We will repeat process until all the characters are scanned. Once scanning is completed, from location of all the characters we will form paragraph from database



While scanning we must write our model such a way that will train database itself hence in next written document error ratio will be reduces.

PHASE2:

- In detection phase check is actual expected character written or not.
- Auto correct or suggestion to nearest word.
- Asking user to check error character, based on user correction train our database.
- Support for famous language where similar character can be accurately detected based on word (Ex: i and l).