

LINK TO COLAB NOTEBOOK:

https://colab.research.google.com/drive/1fO53S07nfwghSae0btFHRY_f_WILr4Ef#scrollTo=0DuSTklov5NI

Here's the basic outline of what my approach was. This approach has maintained the assumptions and will accept the suggested improvements listed in "extra.pdf":

1. First, we form the template. This is done by finding the vertical and the horizontal contours in the image and merging them with equal weights to form the boxes which will highlight our regions of interest. Practically, this will be a one-time thing and may be skipped if a template is already formed and saved. This is the pageContour function in the notebook
2. Next, we upload the filled form. This can be in different circumstances. That's why I've decided to apply a perspective transform. A way around complicating things is by using an OMR scanner (see "extra.pdf"). That being said, a perspective transform won't hurt. The detailed processing step is typed in the colab notebook. The code is written in the transformPG function.
3. Now we can go on to the deep learning part. Again, a one-time step is training it. The model we've used has a ~99% accuracy on digits and ~90% accuracy on alphabets (but a far higher generalization error for the alphabets). The saved weights are stored in digiModel.h5 and alphaModel.h5.
4. Now we basically extract the regions of interest using the template we found and run an OCR on these using the models we trained. Again, this is processed according to the form and is the least generalizable step.

The results are as shown in "results.pdf". A more detailed, well-documented writeup is given in the colab notebook.