ASSUMPTIONS:
Here are a few assumptions I have considered as well as suggestions on violating these, had this been a real-life insurance form used by SBI:

1. **A decent resolution can be maintained.**
   Reason:
   I used several different examples with my approach. In most of them, I found that with the loss in resolution, the region of interest diminishes and it becomes difficult separating contours effectively, even for a human, let alone a computer algorithm.
   Proposed Solution:
   An OMR scanner. It will ensure proper resolution, similar to the example I used (see "result.pdf").
2. **The template is captured/created in the same circumstances as the forms.**
   Reason:
   This basically means that we have to keep in mind that the template will fit. The examples, font size, etc. should be the same as the original form.
   This is much more trivial to solve. The reason I decided to mention this one was that the original form goes out of the standard home printers' printing margin. When I used a printed form with the template, the boxes didn't quite align.
   Proposed Solution:
   Again, this need not be violated. However, we could use something similar to QR codes. A part of the form can be identified and based on the orientation, size, etc., it can be matched to the template using basic geometric transformations. For example, comparing the SBI Insurance logo on the template form and the filled form. Matching some key points and transforming the whole form with how the logo is oriented.

Note: If these assumptions are not to be held, the error can be *minimized* using a user interface. Manual adjustment of the original form inside the template can be helpful once scanned. This is similar to how mobile scanning apps allow you to adjust the detected edges in case the software detected it wrong.

SUGGESTED IMPROVEMENTS:
These are things I wanted to add in a real-life application but were not added due to lack of resources and/or time. More details below:

1. **Template forming and edge detection based:**
   These have already been covered in the assumptions part. The biggest addition would be a user interface that allows the user to position the form w.r.t the template that was learned by the algorithm. This was skipped due to time constraints.
2. **Deep learning-based:**
   This section can be in form of a research-page in itself. However, the basic additions could be:
   - **A richer dataset**. I would've preferred to have a better dataset with noise that's relevant to our environment (such as the block lines on the forms). This wasn't as bad on the MNIST dataset. However, on the EMNIST-alphabet dataset, this gets crucial. For example, one stroke can change 'g' to 'q'. I tried adding noise with

data augmentation, this increased my dataset to over 6 million examples (even after I left out a lot of variations). 6 million 28*28*3 images roughly translate to 52 GBs of data (considering we aren't storing this in floating-point). I doubt any home computer has this amount of RAM. The alternative was to flow this information from a hard disk. This took an extraordinary amount of time, which brings me to the next point.

- **A better system** would allow training of complex models rather than just basic neural networks, such as attention-based models that take into account what has already been processed and use the semantics in English to make better productions. This would also solve our RAM/Hard-disk problem to some extent, depending on the specifications.