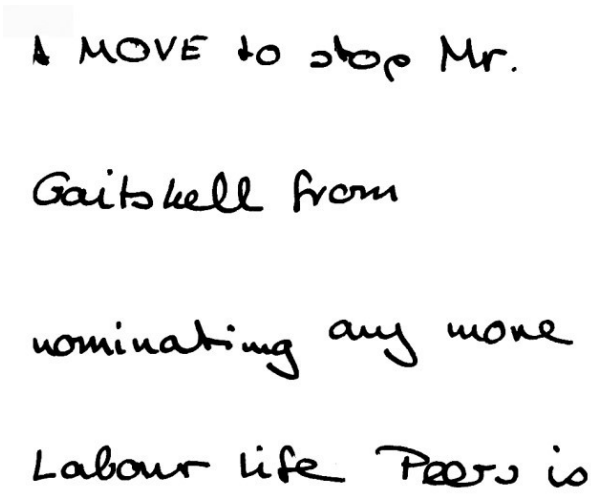
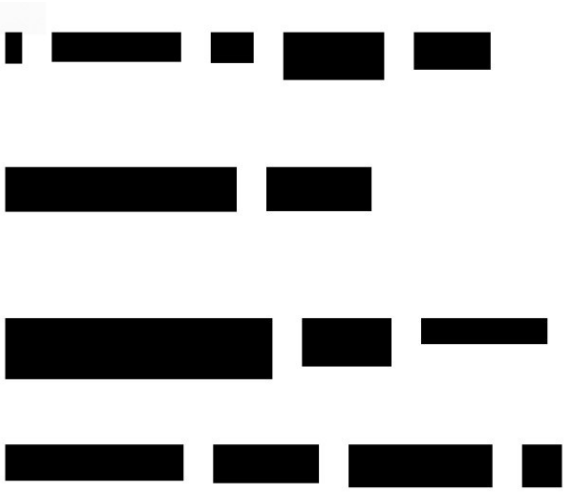
# IV. Solution Approach

## Current Solution Approach

Before constructing our neural network model we tried to generate a good dataset. To achieve such a goal we asked several questions to our supervisor and later decided to use images with multiple words on them. After this decision we searched for several libraries to fasten this process however we could not be able to find a proper one that fits to our needs and wrote down the code for the sample input and map generation by ourself. You can see our base input image and base output image in the figures 4.1 and 4.2.

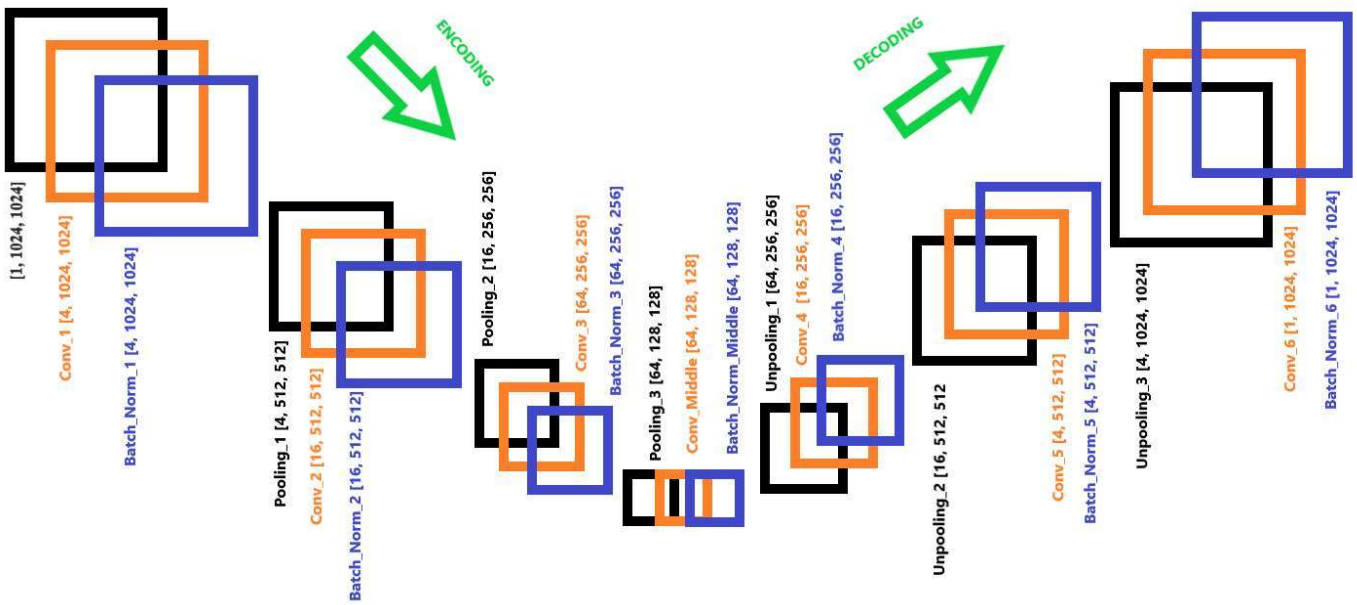


**(Figure 4.1 shows an input)**



**(Figure 4.2 shows the map of the input image shown in left)**

Our current solution approach is to use Auto-Encoder Convolutional Neural Network to encode the text regions and decode back to map of text regions. We chose this type of CNN's because we were able generate inputs and outputs to an encoder decoder system by simply putting our words into single image files. Since this CNN type fits our data the best, we tried to understand how to do encoding and decoding while preserving the important features of each layers. To preserve the previous layer's contribution to the backpropagation we tried to add activations of encoding layers to decoding layers while going to higher resolutions from the middle convolution layer but did not receive any concrete results. Thus we decided to use the Auto-Encoder model illustrated in figure 4.3.




(Figure 4.3 illustrates our Auto-Encoder CNN model)

After each convolution layer we applied the ReLU non-linearity to do activation operation. Over time it enabled our system to learn more complex features. Because if we think R as ReLU function:

- At the first step the input of Batch_Norm_1 layer was R(Conv_1(image)).
- But after this step the input of Batch_Norm_2 layer was, R(Conv_2(Pooling_1(Batch_Norm_1(R(Conv_1(image)))))).

Thus the input of the Batch_Norm_2 became a more complex non-linear function. (not linear means it is not in the form of y = mx + n)

We did not go deeper than 128 by 128 in resolution because we thought that segmentation task in our case was not that complicated since we were able to classify each pixel as 1 or 0. To train well our model, instead of having deeper networks we used the approach of weighted pixels which will be explained in later sections in detail.