a.2 If we use the standard multiplicative representation of join probability, there are some key drawbacks. These drawbacks arise due to the limitations of computer memory for numerical representation. These distributions will likely have probability values on the lower side of the range 0-1, meaning that multiplications on such values could create very small numbers. After a certain point, the distinction between these small numbers may become unclear, detrimentally affecting our training process.

a.3 If we use natural logarithms, our computers' limitations will likely not be pushed as much and we will be able to see clear distinctions between probabilities, allowing for great precision. This is because the natural logarithm is a monotonic transformation, which has key unique properties. Given the product rule of logarithms, we can convert a logarithm of a product argument into the sum probability of that product's constituents. Additions when concerning small numbers eliminates the need to get quite small with our values without compromising on the accuracy of our training process. Moreover, usage of logarithms allows with the option to use Laplace smoothing on our data.

a.5

Training Level	**Correct Prediction Level**
10%	0.745
20%	0.779
30%	0.792
40%	0.793
50%	0.803
60%	0.806
70%	0.807
80%	0.819
90%	0.816
100%	0.817

a.6 k = 0.1 is used for the training process; 0.1 was chosen as an arbitrarily small value. Even in the case of only 10% training, the classifier performs quite well, and in the upper levels of training closer to 100%, the prediction level exceeds 80%, making it a pretty decent classifier.

b.1

Training Level	**Correct Prediction Level**
10%	0.087
20%	0.087
30%	0.087
40%	0.087
50%	0.087
60%	0.087
70%	0.087
80%	0.087
90%	0.087
100%	0.087

The prediction levels for the advanced set are both uniform and pretty poor. However here is the rationale for the two feature sets which were selected and used:

Feature set 1: the first feature set was computing adjacent consecutive blank space sequence lengths from either side of pixel (row-wise) at coordinate (row, column) or (i, j).

Feature set 2: the second feature set was computing adjacent consecutive blank space sequence lengths from up and down a pixel (column-wise) at coordinate (row, column) or (i, j).

These feature sets were created and used because I was of the belief that there should be significant variance in the lengths of consecutive blank space sequences for the different digits 0-9. Moreover, I thought there would be an interesting difference between row-wise iteration and column-wise iteration.

b.2

Training Level	**Correct Prediction Level**
10%	0.416
20%	0.433
30%	0.440
40%	0.440
50%	0.440
60%	0.447
70%	0.447
80%	0.453
90%	0.452
100%	0.452

The prediction levels when combing the two feature sets are worse than the basic set but better than the advanced set.