Report

1. Implementation
   a) kNN: (See blocks 5-9) Based on the test, calculated the distance in the sample and find out the nearest K.
   b) decision tree: (See blocks 10-14) When a subtree's information content is lower than a given threshold, or if a subtree sample marked as a label, it can stop splitting. Otherwise, looking for the one with the most drop and perform the split.Since these pixels are numeric and decision tree is a discrete variable, the pixels must be quantised first. Quantisation is to take a value, less than it is 0, the rest is 1, this value is obtained by traversing the value of the pixel, specifically, for example, the value of all samples of the first pixel is 0,1,3,4. Then the quantified cutoff value is 0.5, 2, 3.5. (midpoint). Then take the largest drop in content as the final quantified result
   c) CNN: using keras built-in functions Conv2D(), MaxPooling2D(), Dropout(), etc, I implement LeNet to classify the images. See the other blocks.

2. Training and testing accuracy/runtime
   (note that the training time of KNN is the time of build a kdtree, so is the same, while retrieving k neighbors during testing varies a little.)

	Training acc	Testing acc	Training time	Testing time
K=1 NN	1.0	0.8	-	0.244
K=2 NN	0.905	0.785	-	0.247
K=3 NN	0.795	0.71	-	0.247
K=4 NN	0.73	0.645	-	0.248
K=5 NN	0.655	0.615	-	0.249
K=6 NN	0.62	0.58	-	0.249
K=7 NN	0.605	0.555	-	0.251
K=8 NN	0.585	0.555	-	0.252
K=9 NN	0.572	0.55	-	0.253
K=10 NN	0.57	0.54	-	0.250
DT (IC = 0 bits)	1.0	0.595	246.27	0.120
DT (IC = 0.1 bits)	1.0	0.595	245.74	0.100
DT (IC = 0.2 bits)	1.0	0.595	246.68	0.100
DT (IC = 0.3 bits)	1.0	0.595	243.56	0.115
DT (IC = 0.4 bits)	0.975	0.64	231.62	0.079
DT (IC = 0.5 bits)	0.975	0.64	227.52	0.078
CNN	1.0	0.985	20.596	0.022

3. answer to bonus question

The main reason why decision tree is slow is because of a large amount of calculating entropy related to each split on each dimension, so to pre-determine whether to use a dimension or not is important.

Here I statistics the **variance** of each pixel and sort them in decreasing form, the dimensions with a variance less than 0.05 will be cut off. See the last block.

With IC=0.1, the training accuracy of decision tree is 1.0, while the testing accuracy is 0.62. And the training runtime is 86.106, while that of testing is 0.143.

We can see that, with a save of runtime, the decision tree shows an improvement of accuracy, which indicates the advantages of feature selection.