Homework report

Problem1

Part A
The average accuracy of 10 times I got is 0.7457516.

Part B
The average accuracy of 10 times I got is 0.7320261.

Part C
I used 10 fold cross validation and grid search, The final parameter for the model is fL = 1, usekernel = TRUE and adjust = 1.
Accuracy is 0.7255.

Part D
The average accuracy of 10 times I got is 0.7437908.

Problem2

Part A

Accuracy	Gaussian	Bernoulli
Untouched images	0.55579999999999996	0.84130000000000005
Stretched bounding box	0.81179999999999997	0.82999999999999996

With untouched images, Bernoulli model is better. But some feature engineering, in this case crop and stretch, improved Gaussian model performance. My understanding is that when we train the model, the edge pixels are regarded as features even though they are actually noises, so to cut the edges off can better separate the distribution model among different labels.

Part B
Untouched images

Accuracy	Depth=4	Depth=8	Depth=16
#trees=10	0.71679999999999999	0.69830000000000003	0.71430000000000005
#trees=20	0.79700000000000004	0.78779999999999994	0.79530000000000001
#trees=30	0.81759999999999999	0.81510000000000005	0.8256

Stretched bounding box

Accuracy	Depth=4	Depth=8	Depth=16
#trees=10	0.67410000000000003	0.67700000000000005	0.66290000000000004
#trees=20	0.76039999999999996	0.75180000000000002	0.7621
#trees=30	0.77290000000000003	0.8004	0.79659999999999997

The random forest seems doing slightly better job for Untouched images than Stretched images.

Resource

1. http://luthuli.cs.uiuc.edu/~daf/courses/AML-18/aml-home.html (problem 1 is adapted from instructor code)
2. https://piazza.com/class/jchzguhsowz6n9?cid=160 (referred this piazza post code for the implementation of NB in problem 1A)
3. http://blog.csdn.net/qq_32166627/article/details/62218072 (problem 2 used code from this blog for extracting raw data into numpy array)
4. https://codereview.stackexchange.com/questions/132914/crop-black-border-of-image-using-numpy (referred this stack overflow answer for problem 2 image crop solution in Python)
5. http://scikit-learn.org/stable/modules/naive_bayes.html (problem 2A referred example code for Gaussian Naive Bayes and Bernoulli Naive Bayes)
6. http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html (problem 2B referred example code for RandomForestClassifier)