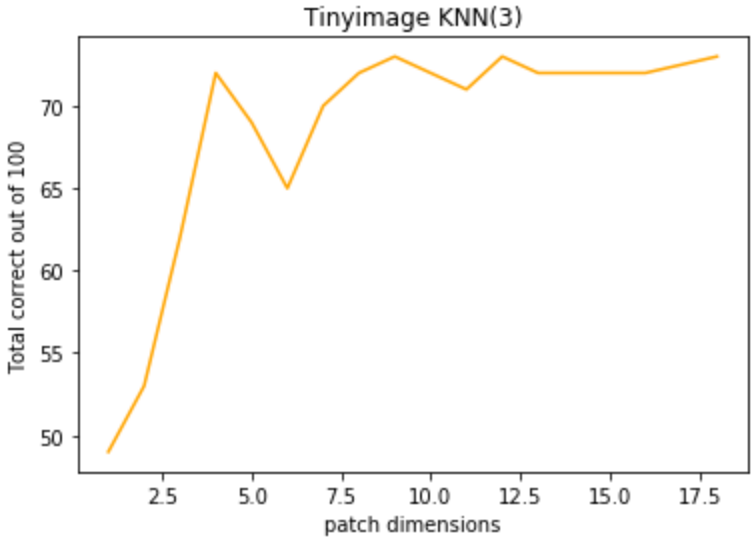
Implement param.feature = 'tinyimage':

- What is the size of the patch that achieves the best classification rate on the validation set?

The following are the results when knn is used as classifier with k=3.

(patchDim, accuracy): (1, 49), (2, 53), (3, 62), (4, 72), (5, 69), (6, 65), (7,70), (8, 72), (9, 73), (10, 72), (11, 71), (12, 73), (13, 72), (14, 72), (16, 72), (18, 73), (24, 70), (32, 71), (64,70), (128, 69), (256, 68)



The best value seen is 73 which occurs for patch size 12. But patch size of 4 would be sufficient to get good results and save from more memory and computation time.

- What classifier works better for these features?

The following are the results for svm classifier when lambda is set to 4.

(patchDim, accuracy): (1, 60), (2, 54), (3, 71), (4, 62), (5, 67), (10, 66)

When compared with above knn results we can say that svm works better when patch size is small, but as patch size increases knn starts to outperform svm. The above conclusion is based on k =3 and lambda = 4 for svm. Results may vary for other hyperparameters.

- What value of k works best for kNN classification?

The following are the results obtained for patch dimension of 4.

(k, accuracy) : (1, 62), (2, 63), (3, 72), (5, 63), (7, 69), (11, 70), (17, 70).

Even though the results are good for higher values of k, 3 is good enough.

Implement param.feature = "bow-patches":

- What value of k works best for kNN classification?

Parameters used: patch radius- 8, dictionarysize- 128, patch-stride- 20, classifier- knn

(k, accuracy): (3, 63), (4, 65), (6, 66), (8, 64)

- What values of r work best?

Following are the parameters used (stride = 20, k = 5, dictionary size = 128)

(r, accuray): (2, 61), (4, 65), (6, 64), (8, 59)

- What values of stride works best?

Parameters used: r = 3, k = 7, dictionary size = 128

(stride, accuracy): (8, 61), (12, 57), (20, 61), (30, 58), (40, 60), (60, 54), (100, 54)

- What values of dictionary (K value in k-means) size works best for bow patches?

These values are obtained for the following parameters (r = 5, stride = 13, k = 5)

(Dictionary Size, accuracy): (32, 60), (64, 62), (128, 70), (256, 61), (512, 68)

- Which classifiers did work best?

From my observations it seems like SVM works better for smaller values of r. For larger values both of them work equally well. Also K value greater than 5 yields good results.

Implement param.feature = "bow-sift":

Parameters used: (sift_dictionary = 128), (sift-stride = 20), (KNN k = 5).
Accuracy obtained: 61% when sift-radius = 4
68% when sift-radius = 6

BHANU PHANINDRA GORANTLA (QJ82419) **C.V HW3**

Note: It takes more than an hour for run one round result. So it is not possible for me get results for different parameters.

Visualization of the learned dictionary using patches:

The above is the visualization of dictionary build by bag of words using patches. Each of image is also included in zip along with helper python code used to generate.

Show the most confused images on the test set:

Most confident correct classification: yorkshire_terrier_152.jpg, confidence: 0.78

Most confident wrong classification: japanese_chin_114.jpg, confidence: 0.62

Most wrong classified cat: Egyptian_Mau_153.jpg, confidence: 0.6