Assignment 2 Report

a) What features did you select and implement? Why did you select these features and how do you think they will be informative for this classification problem?

We choose HOG (Histogram of oriented gradients) to capture the features of brick patterns. The feature vector is statistical results of gradients of the picture, and each element represents the weight of gradient of a cell on one specific direction, which we are going to explain in detail in the following implementation.

Implementation:

First of all, we cut images into many small cells (the typical size of our cell is 8×8 pixels) and we group a square of cells into a block (typical size of a block is 3×3 cells). Second, calculate the gradient of pixels in every cell and we can divide these gradients into N bins as well as N directions. For example, we can divide 0-180° into 9 parts, each part contains 20°. All cells within the block will be normalized, which is used to reduce the influence of illumination and shadow. After catenating all the oriented gradients vectors of blocks, we can get the feature vector of an image. The size of vectors is $N \times p^2 \times m_r \times m_c$, where N is the number of bins we use to partition the direction, p is the cells of each block in raw/column, m_r is the # of blocks we have in a raw and m_c is the # of blocks we have in a column.

Since the gradient of the pixels around brick joints will be perpendicular to the joints and in each block, different brick patterns will influence the gradients of each pixel so that generate different histograms of gradients directions, HOG is able to capture the features of different brick patterns.

b) Given your data set and features, were there any outliers in your data set that needed to be removed or handled specially during pre-processing/feature extraction? If so, what was causing these images to be outliers? If not, do you think your data set is a good representation for the problem or are you missing imagery of a certain type in your data set that could impact your overall performance?

We do not need to remove any outliers in our data set. We think our data set is a good representation for this problem since we shot the brick pattern in different angles and different light condition. And our data set has all the 5 classes, each class has 60 pictures.

c) How effective is your trained system (preprocessing + feature extraction + classifier)? Do you think you already have a system that would be competitive/effective for the project overall? If so, why? If not, what is needed to improve your approach?

The choice of N will influence the efficiency of model training and the relationship of time cost and N is given in figure 1. In this figure we can find they have a linear relation. Besides, the best value of N varies with different training data, and we find that the typical time of our trained system range from 8 to 12 secs in total (Ryzen 5 3600) with $p = 8$ and 3×3 cells per block.

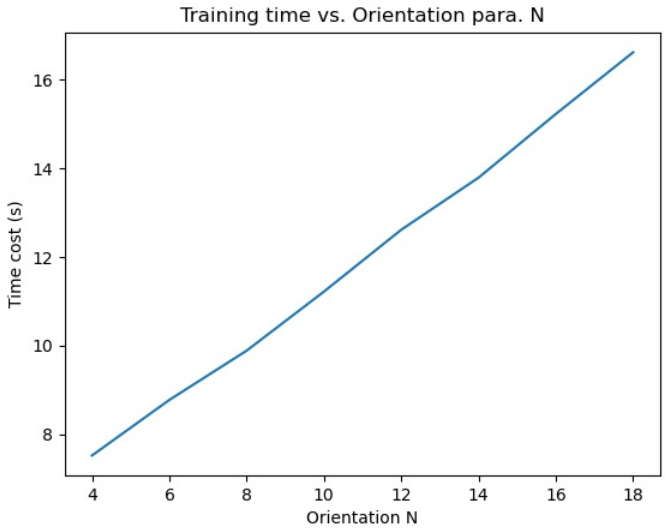

Figure 1. Linear relation of training time and orientation N

To achieve 80+% precision, here we apply K-fold cross-validation (typically, K = 10) to optimize the parameter of orientation value N for different training data from a series of choices: N = 4, 6, 9, 12, 18, corresponding to 45°, 30°, 20°, 15° and 10°, and it costs less than 1 minutes. By repeating model training with random selected training data set, we find that in most cases, the gradient direction interval of 15° and 10° has better performance than other choices, so we suggest to set the N = 12 or 18.

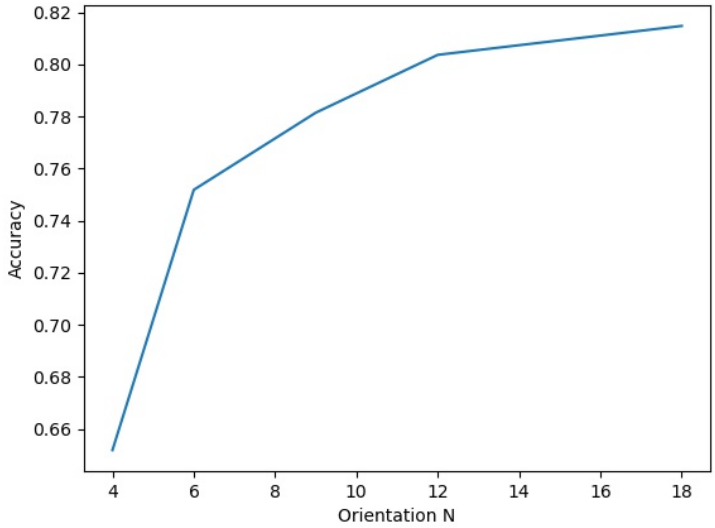

Figure 2. The accuracy of cross validation result with different orientation m

I do not think this system is effective enough for the project overall. HOG is useful for almost all the image classification problems. However, for classification of simple and specific patterns, such as brick pattern, the size of feature vector of each picture extracted by HOG is too big. If we set N = 18, for a 200×200×3 RGB picture each vector includes 42849 elements (flatten from a 23×23×3×3×9 feature matrix). We think that there should be some simpler and more effective feature extraction methods, which can reduce the dimension of the feature vector to reduce the computation time.

About our trained system

Preprocessing: No (Optional method: Grey processing or Gamma correction)
Feature extraction: HOG (Histogram of oriented gradients)
Parameter: Orientation channel N – optimized by K-fold cross-validation, 3×3 cells per block, 8×8 pixels per cell
Generative model: Gaussian Naïve Bayes model