Finding the other columns of visual bits

1. Randomly generate matrix of weights

   500,000 weights

   128 dimensions

   ...

2. Get both positive and negative SIFT descriptors (also create label vector, 1 being positive, -1 being negative)

   ~5000 SIFT descriptors

   128 dimensions

   ~5000 labels
Finding the other columns of visual bits

- Use AdaBoost for first visual bit system

3. Initialize distribution:

\[ D_1(i) = \frac{1}{m}, i = 1, \ldots, m. \]

where \( m \) is \( \sim 5000 \)

4. Find the best weights and threshold that minimize the error with respect to the distribution:

\[ \epsilon_j = \sum_{i=1}^{m} D_t(i)[y_i \neq h_j(x_i)] \]
Finding the other columns of visual bits

5. Update the distribution according to:

\[ D_{t+1}(i) = \frac{D_t(i) e^{-\alpha_t y_i h_t(x_i)}}{Z_t} \]

where: \[ \alpha_t = \frac{1}{2} \ln \frac{1 - \epsilon_t}{\epsilon_t} \]

\( \epsilon_t \) is the error rate

\( Z_t \) is a normalization factor

This is basically saying that if we incorrectly classified one of the descriptors, then we add some value to the corresponding element in the distribution in order to make it more important for future rounds

6. Find new matrix of random weights and repeat the process for ~200 rounds
Progress

- Studied boosting in general
- Implemented AdaBoost
Plan

- Talk to Dr. Sukthankar today
- Implement other classes and finish the training part of the visual bits system
- Start on the testing part of visual bits system