Requirements:

For sift features, the method used to extract features was to first convert the cifar-10 dataset into grayscale images with rgb2gray from skimage.color library, then using functions from skimage.feature features are extracted from the grayscale images and stored into NumPy arrays (we only saved the descriptors of the images, not the key points).

For hog features, a similar method was used to extract features. First with converting the cifar-10 dataset into grayscale images with rgb2gray, then using hog from skimage.feature library to extract the images features into NumPy arrays.

The number of sift features extracted was 15986, with a classification accuracy of 27% and 870 images correctly classified.

The number of hog features extracted was 16000, with a classification accuracy of 39% and 1257 images correctly classified.

Questions:

1) Since HOG features are typically generated for the entire image, we can perform key points matching with HOG features by dividing the image up into smaller, overlapping patches. Within each patch, a local descriptor is computed and used for matching with a brute-force method or any other method.
2) In terms of performance or speed, HOG was able to extract features and go through classification much quicker than SIFT. This is probably because HOG extracts global features for the entire image, while SIFT extracts features based on interest points. In terms of feature quality, HOG can produce a gradient across an entire image while SIFT only produces a gradient across interest points. This could explain the difference in classification accuracy between HOG and SIFT, since producing a gradient over interest points only means that SIFT struggles with smaller, low-resolution images since it is harder for it to discern interest points making the classification accuracy lower than what HOG got.