Student Name: Anumanchi Agastya Sai Ram Likhit
Student ID: 20048

Computer Vision
Assignment 3

1 Question 1

Use the CIFAR-10 dataset (https://www.cs.toronto.edu/ kriz/cifar.html) and perform the image classification using

1. SIFT Features + SVM (can use the library)
2. HOG Features + SVM (can use the library)

Answer 1a

Dataset

The CIFAR-10 dataset, which consists of 60,000 32x32 color images in 10 different classes, is loaded into the program. The dataset is divided into 50,000 training images and 10,000 test images. To simplify the feature extraction process, the color images are converted to grayscale by averaging the color channels. This conversion is expected to reduce the complexity of the data while retaining the essential features needed for classification.

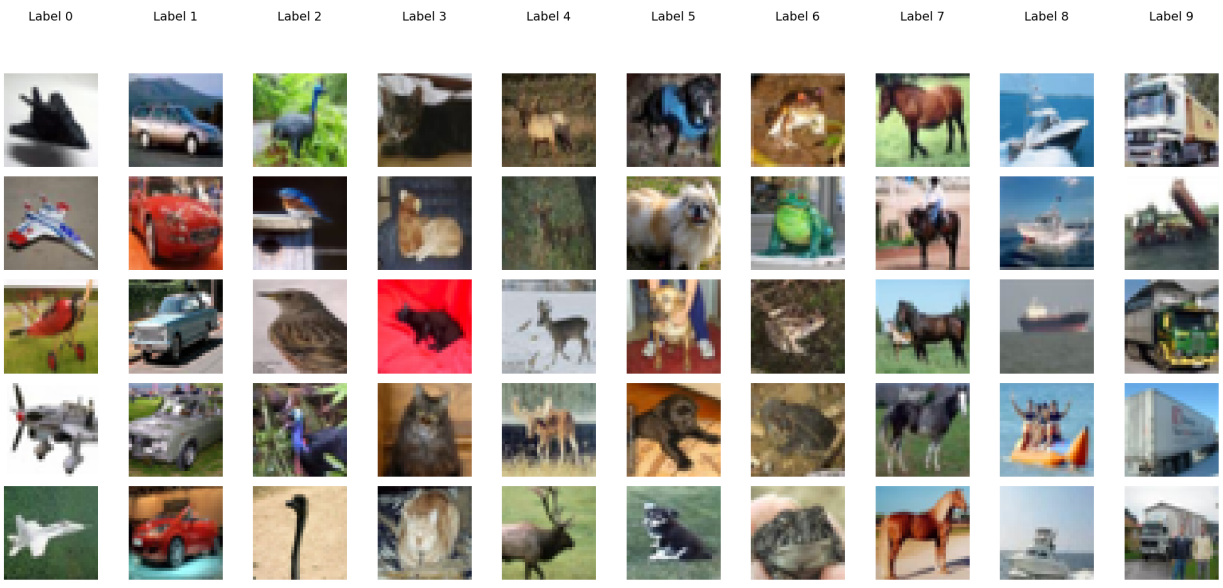

Figure 1: Data Set

SIFT

SIFT features are extracted from the grayscale images of the training set. SIFT is an algorithm used to detect and describe local features in images. The $extract\_sift\_features$ function iterates through each image, calculates the SIFT descriptors, and stores them in a list. These descriptors represent the local features of the images that will be used to train the classifier.

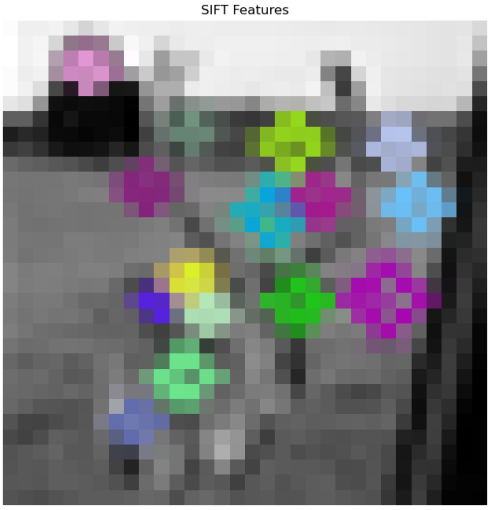

Figure 2: SIFT Local Features

K-Means Clustering to Create Bag of Words BoW (Images)

The $kmean_bow$ function performs K-Means clustering on all the SIFT descriptors extracted from the training set images. The number of clusters (set to 10 in the code) defines the "visual words" in the BoW model. Each cluster center represents a visual word, and the collection of all cluster centers forms the BoW dictionary.

Feature Encoding using sBoW

The $create\_feature\_bow$ function encodes the SIFT descriptors of each image into a fixed-size feature vector based on the BoW dictionary. For each image, the function calculates the Euclidean distance between its SIFT descriptors and the visual words, assigning each descriptor to the nearest visual word. This process effectively creates a histogram that counts the occurrences of each visual word in the image. The resulting histograms are used as feature vectors for training the SVM classifier.

Training the SVM Classifier

The feature vectors are split into training and testing subsets. An SVM classifier with a specified regularization parameter $C = 30$ is then trained on the training set. The trained model is saved to a file for later use.

Model Evaluation

The performance of the trained SVM classifier is evaluated on both the training and testing sets. The results are printed out to show how well the model is performing. The evaluation metric used here is the accuracy score, which measures the proportion of correct predictions.

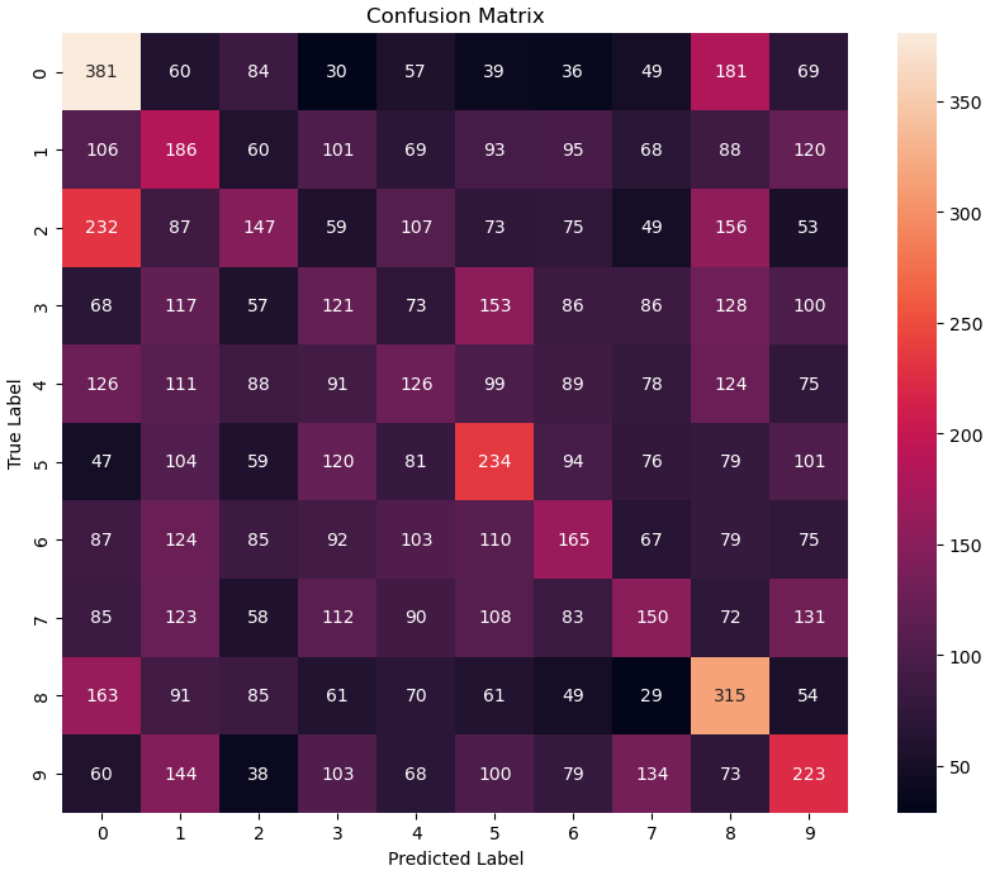

Figure 3: SIFT Local Features

Answer 1b

1.1 Feature Extraction with HOG

The HOG descriptor is computed for each grayscale image. HOG captures edge or gradient structure that is characteristic of local shape, which is done by dividing the image into small connected regions, called cells, and compiling a histogram of gradient directions or edge orientations for the pixels within each cell. These histograms are normalized across larger, spatially connected blocks for improved accuracy and to account for changes in illumination or shadowing.

Feature Standardization

The extracted HOG features are then standardized by removing the mean and scaling to unit variance. This standardization step is crucial as it ensures that each feature contributes equally to the distance computations in the SVM, which is sensitive to the scale of input data.

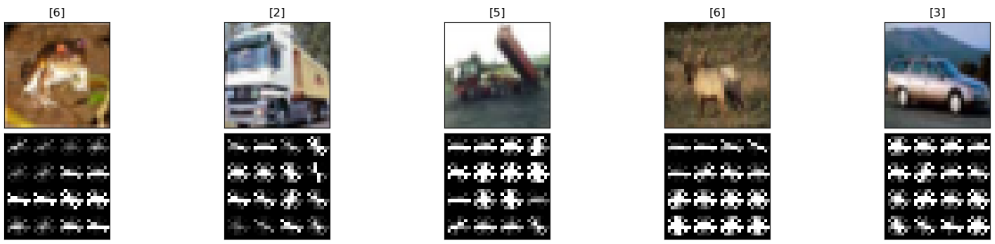

Figure 4: HoG Features

SVM Classification

An SVM classifier is trained on the HOG features. SVMs are particularly well-suited for classification tasks with complex feature spaces and limited training samples, as is the case with image recognition tasks.

Model Evaluation

The trained SVM model's performance is evaluated on a separate test set using accuracy as the metric. Accuracy measures the proportion of true results (both true positives and true negatives) among the total number of cases examined.

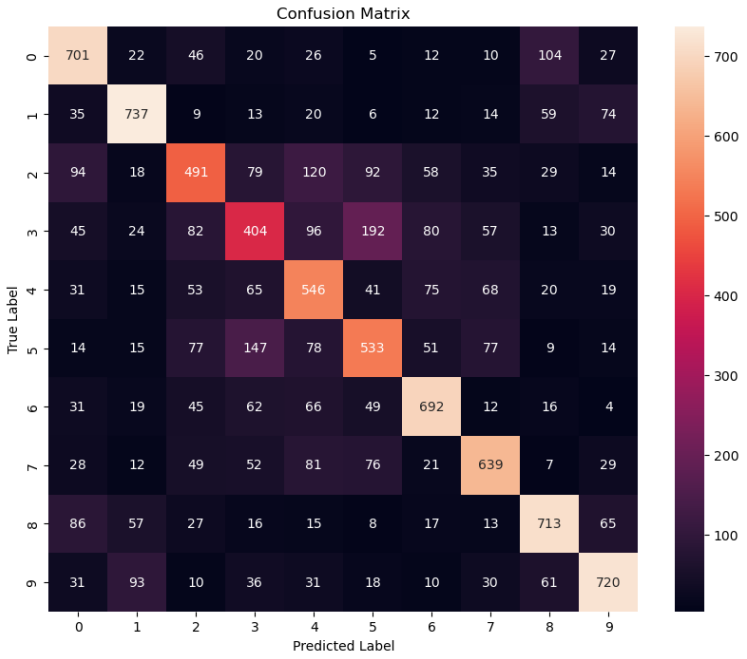

Figure 5: Confusion Matrix

2 Question 2

Compute the LBP images of the dataset and perform the above two steps. You can do one analysis by comparison when you apply the SIFT and HOG on raw RGB images and when they are applied to the LBP images.di

Answer 2

Now for the images we define a function to get the LBP image.LBP is a simple yet efficient texture operator that labels the pixels of an image by thresholding the neighborhood of each pixel and considers the result as a binary number. It is widely used for texture classification due to its discriminative power and computational simplicity.

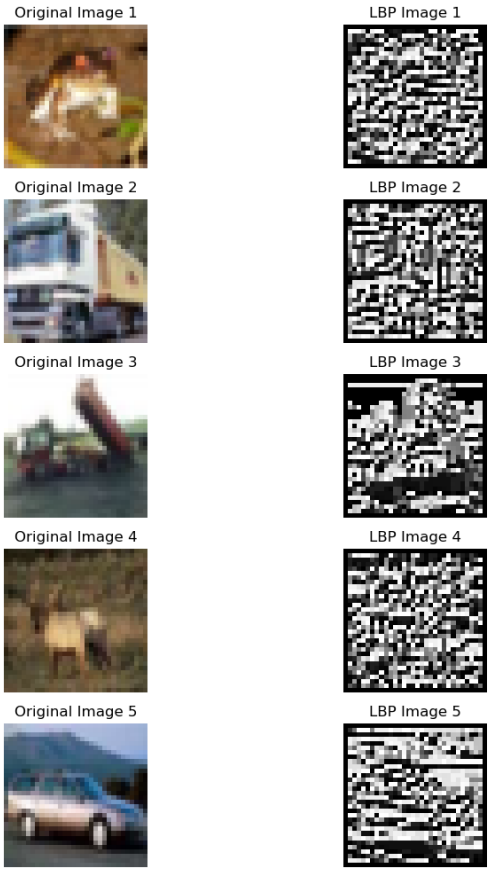

Figure 6: LBP Image

we utilize the CIFAR-10 dataset, a staple in the field of machine learning for image recognition tasks, comprising 60,000 images categorized into 10 distinct classes. The dataset is partitioned into a training set and a test set. To capture the textural characteristics of these images, we employ the Local Binary Pattern (LBP) method, which effectively transforms each image into a texture descriptor by examining the pixel and its immediate neighbors.

Once the LBP descriptors are obtained, we prepare our dataset by converting the images into grayscale and applying the LBP operation to each image. Subsequently, these descriptors are used to train a Support Vector Machine (SVM), a robust machine learning model, which learns

to distinguish between the various classes of images. Upon completion of the training phase, we evaluate the SVM's performance using the test set. The model's predictions are then juxtaposed with the actual labels to compute the accuracy, which serves as a metric to gauge the effectiveness of our texture-based classification approach.

The SVM classifier achieved an accuracy of 26.51, which suggests that the texture information captured by LBP alone may not be sufficient for high-accuracy classification in a complex dataset like CIFAR-10.

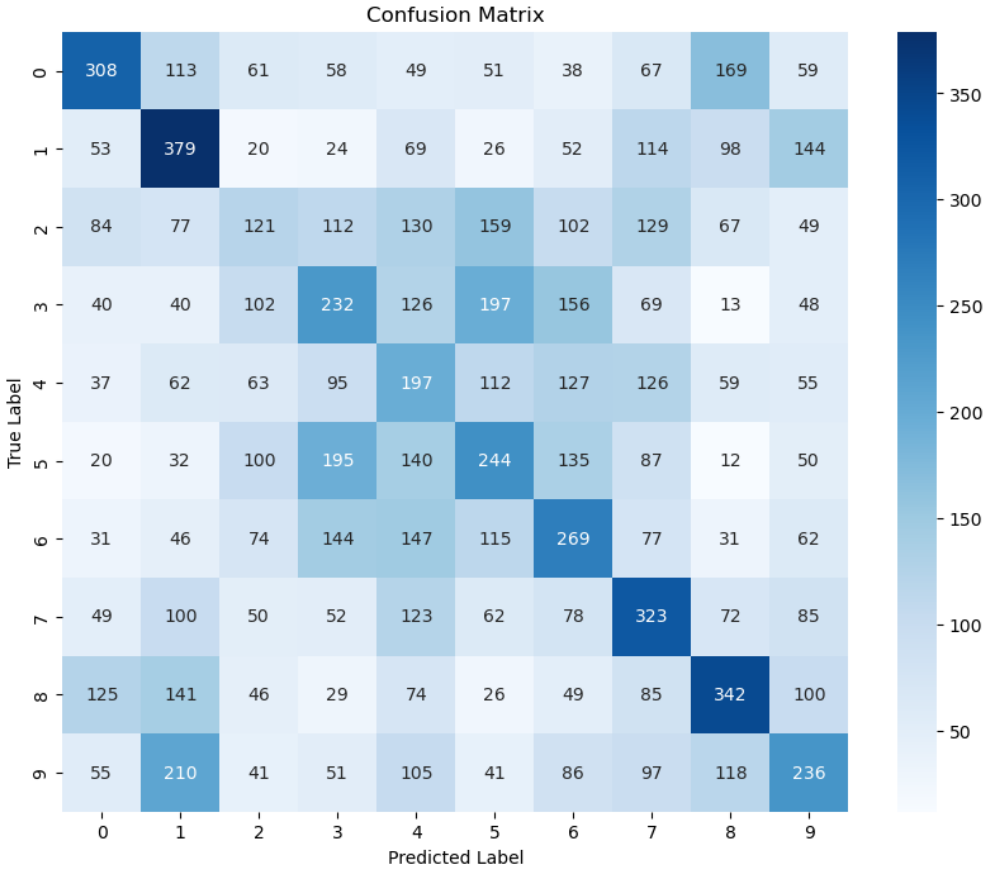

Figure 7: Confusion Matrix

3 Question 3

Take one video from any online source, compute the optical flow, and visualize the optical flow by mapping it back to the original video. (can use the library)

Answer 3

The task involves taking a video from an online source - a Moving Rocket in Space, computing the optical flow, and visualizing it by mapping it back onto the original video.

Optical flow refers to the pattern of apparent motion of objects, surfaces, and edges in a visual scene caused by the relative motion between an observer and the scene. The Lucas-Kanade method

implemented in OpenCV is used for sparse optical flow computation, which tracks the motion of selected feature points or corners between two image frames.

Initially, the video is loaded, and feature points are selected in the first frame based on their corner-like qualities. These points are tracked throughout the video using the Lucas-Kanade method for optical flow, which estimates the motion of these points across successive frames. Parameters for detecting features and calculating optical flow are meticulously set to capture the motion accurately. As the video progresses, each frame is processed to update the positions of these points, and the trajectory of each is visualized with lines and circles overlaying the video, resulting in a dynamic representation of the motion. The tracking continues until the video concludes or is interrupted by the user, providing a comprehensive visualization of the rocket's path as a pattern of motion against the backdrop of space.

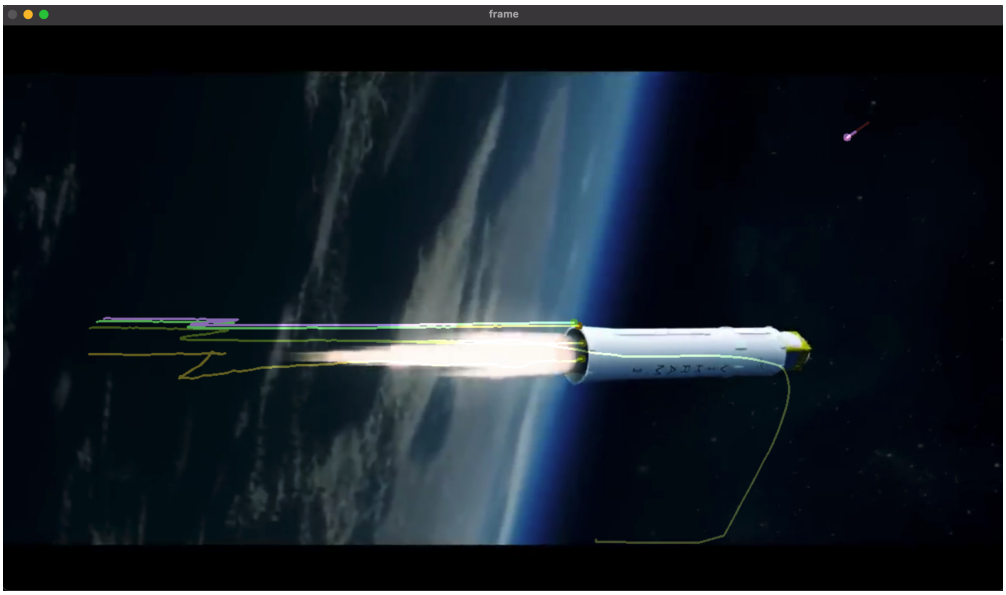

Figure 8: Optical Flow