University of Amsterdam

Computer Vision 1

Lab Project - Part 1

October 21, 2023

Students:
Fien De Kok
12672289

Jakub Tomaszewski
15178331

Joan Velja
14950480

Group:
G4

1 Introduction

Image classification is a cornerstone task in the field of computer vision and has wide-ranging applications that span from medical imaging to autonomous navigation systems. The problem is typically formulated as a supervised learning task, where the aim is to learn a function $f : \mathcal{X} \to \mathcal{Y}$ from a set of labeled examples $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$. Here, $\mathcal{X}$ represents the input space, which in the context of image classification would be images represented as multi-dimensional arrays. $\mathcal{Y}$ represents the output label space, which is a set of predefined classes that the algorithm aims to categorize input images into.

Given the complexity and high dimensionality of the image data, standard linear models are generally insufficient for this task and fall short in detecting the underlying patterns in the data. An alternative which can circumvent the shortcoming of linear models is to take advantage of techniques known as feature extracton. These techniques aim to transform the input data into a more compact representation, which can then be used as input to a linear model. In the context of image classification, this can be achieved by extracting features from the raw image data, forming a feature vector, and then using this vector as input to a linear model. This can not only improve the performance of the model but also reduce the computational complexity of the learning algorithm.

In this report, we utilize a prominent approach known as Bag of Visual Words (BoVW), which represents each image as a set of features, also known as 'visual words', instead of standard pixel values. Such predefined features are stored in a dictionary of visual words and acts as a lookup table for the model. It then counts the number of occurrences of each visual word in the image and uses it as an input to a classifier whose task is to determine the image label. We employ the SIFT (Scale-Invariant Feature Transform) algorithm to detect and describe the local features in our input images as it is known to be a robust feature extraction method, invariant both to scale and rotation. Moreover, to validate the robustness of SIFT, we compare the obtained results to another feature descriptor, namely HOG - Histogram of Oriented Gradients. Subsequently, we form the visual dictionary by grouping the features using the K-Means clustering algorithm.

To evaluate our approach, we utilize the CIFAR-10 dataset, being a popular benchmark dataset for numerous computer vision models. Comprising 60,000 color images with a resolution of 32×32 pixels, CIFAR-10 is categorized into 10 classes, with each class having 6000 instances. Divided into 50000 training images and 10000 testing images per class, CIFAR-10 presents a challenging array of objects and scenes, thereby offering a comprehensive testbed for our model.

The remainder of this report is structured to offer an analysis of various parameter settings for the aforementioned Bag of Visual Words technique. Initially, we inspect the training dataset

and validate the performance of the SIFT algorithm. Subsequently, we produce a visual dictionary using the provided CIFAR-10 dataset and train an SVM classifier model using the extracted features. Eventually, we strive to find the best set of parameters by scrutinizing the performance of each model using evaluation metrics such as accuracy, F1-score, mean Average Precision (mAP). This is followed by a detailed discussion dissecting the observed results and highlighting possible improvements.

2 Dataset

CIFAR-10 is a labeled subset of 80 million tiny images dataset where CIFAR stands for Canadian Institute For Advanced Research. The images were collected by Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. The dataset consists of 60000 colored RGB images (50000 training and 10000 test) of 32 $\times$ 32 pixels. CIFAR-10 has 10 classes containing 6000 images each, where there are 50000 training images and 10000 testing images per class.

For the scope of our project, we are narrowing our focus to a subset of CIFAR-10, concentrating specifically on five classes namely 1: airplanes, 2: birds, 3: ships, 4: horses, 5: automobiles, each containing a 1000 images. The final training and test set have both the comprise 5000 images each. Furthermore, the training set is split into subsets of different rations [0.3, 0.4, 0.5], which are used to create the visual dictionary. The remaining images are then used to train the SVM classifier. Moreover, the test set is kept constant throughout the experiments.

Figure 1: CIFAR-10 dataset

3 Method

In this section, we describe the methodology used to form our visual dictionary and train an image classifier. We begin by utilizing the SIFT algorithm to extract key features from images before moving on to K-Means clustering to create a visual dictionary. Finally, to generate our classification model, we count the number of features in each image and train a Support Vector Machine (SVM) model as the final step in our approach. In the next sections, we elaborate on each step and present code snippets and obtained results.

3.1 Feature Extraction

Due to features being an essential component of the Bag of Visual Words approach, it is crucial to select a feature extraction algorithm that is robust to various image transformations. Scale-Invariant Feature Transform (SIFT) is our algorithm of choice for this task. The strength of SIFT comes from its ability to detect and describe local features in images that remain consistent in various conditions, such as different rotation angles, illumination changes, and resizing. The aforementioned features are thanks to SIFT obtaining the Difference of Gaussian (DoG) of an image, with various σ parameter values. Once this DoG are found, images are searched for local extrema (potential keypoints) and any outliers are removed. These keypoint descriptors are used to form the visual dictionary in the next chapters of the report. To better understand the SIFT algorithm, we illustrate the features detected in the sample images from the CIFAR-10 dataset. We notice, that the features are detected in various parts of an image, which could be potentially useful in categorizing the images into different classes.

In the utilized SIFT implementation from the OpenCV library the algorithm takes as input a grayscale image and returns a list of keypoint descriptors. Each keypoint descriptor is a 128-dimensional vector that describes the local features of the image. A code snippet of the SIFT algorithm is shown below.

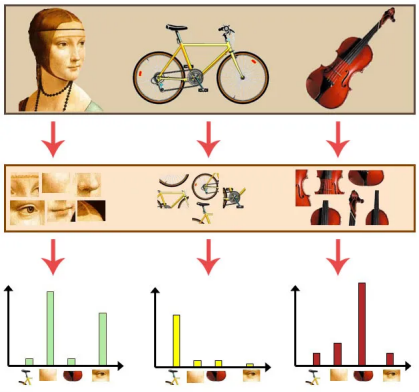
```
sift = cv2.SIFT_create()
sample_gray = cv2.cvtColor(sample, cv2.COLOR_BGR2GRAY)
_, descriptors = sift.detectAndCompute(sample_gray, None)
```

Figure 2: Detected SIFT features

3.2 Creating the Visual Dictionary

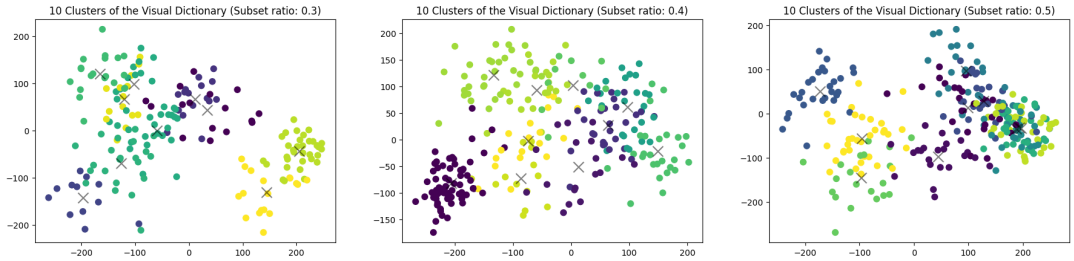
The popular Bag of Words model is an approach which has its roots in text analysis. Its main concept is to break down a documents into individual words or terms and create a set of these terms, which can be later used for classifying articles. Such an approach can be also adapted to photos, by representing each image into a set of features and combining them to form a large visual dictionary. The process of creating the visual dictionary can be split into three major steps, namely feature detection and extraction, feature clustering, and histogram formation. Two of these steps have been illustrated in Figure 3.

Figure 3: Bag of Visual Words [source]



After extracting the features from the images, we group them into clusters, representing our visual words. We accomplish this by using the K-Means clustering algorithm, which is a popular unsupervised learning algorithm that aims to partition the data into K clusters. The algorithm works by assigning each data point to the closest cluster center in a high dimensional space. For the purpose of this task, we employ the scikit-learn implementation of K-Means, and set the number of clusters to 1000. The code snippet below illustrates the process of creating our visual dictionary. Furthermore, we generate a separate visual dictionary for different dataset subset ratios, i.e. [0.3, 0.4, 0.5]. We randomly sampled 10 data clusters and illustrated them in Figure 4.

```
def build_visual_dictionary(data: np.ndarray, vocabulary_size=1000, random_state=42):
    """Creates a visual dictionary from the given data
    with K (vocabulary_size) visual words using KMeans algorithm.

    Args:
        data (np.ndarray): Data to build visual dictionary from.
        kvocabulary_size(int, optional): Num clusters. Defaults to 1000.
        random_state (int, optional): Random state. Defaults to 42.

    Returns:
        visual_dictionary: The visual dictionary.
    """
    visual_dictionary = KMeans(n_clusters=vocabulary_size, random_state=random_state).fit(data)
    return visual_dictionary
```



(a) Subset ratio: 0.3 (b) Subset ratio: 0.4 (c) Subset ratio: 0.5

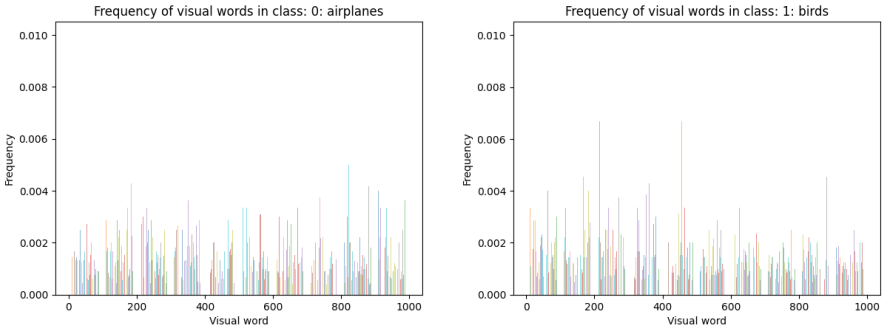
Figure 4: 10 Randomly chosen clusters and their centers, sampled from the visual dictionary created usind different subset ratios

3.3 Forming the Histograms

Once the visual dictionary is created using a subset of the training data, we use the remaining images to form the histograms, which represent the detected feature counts. To this end, we first extract the features from each image using SIFT and find the closest visual word in the dictionary for each feature. We then increment the corresponding bin in the image histogram and repeat this process for all features in the image. Eventually, we obtain a histogram of visual words for each image, which is later used as an input for our classifier. A function that performs this task is shown below.

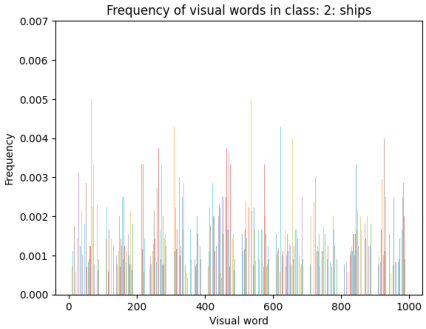
```python
def preprocess_data(data, labels, visual_dict):
    """Preprocesses the data by encoding the images using the visual dictionary and creating histograms.

    Args:
        data (np.ndarray): input data to be preprocessed
        labels (np.ndarray): input labels
        visual_dict (np.ndarray): visual dictionary

    Returns:
        data_histograms: the preprocessed data
        labels_encoded: corresponding labels
    """
    data_encoded, labels_encoded = encode_images(data, labels, visual_dict)

    labels_encoded = np.array(labels_encoded)

    assert len(data_encoded) == len(labels_encoded)

    data_histograms = []

    for sample in data_encoded:
        counts, _ = np.histogram(sample, bins=VOCABULARY_SIZE)
        data_histograms.append(counts)

    data_histograms = np.array(data_histograms)
    return data_histograms, labels_encoded
```

To better understand the distribution of visual words in each class, we plot the frequency of each visual word in the training dataset, which has is depicted in Figure 5. We notice, that the distribution is fairly uniform, with some visual words being more frequent than others.

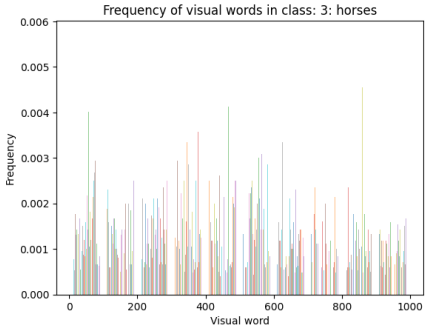


(a) Class 0: airplanes

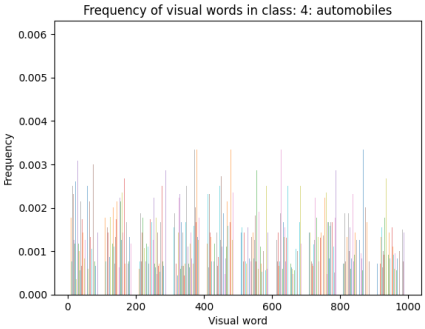
(b) Class 1: airplanes



(c) Class 2: ships



(d) Class 3: horses



(e) Class 4: automobiles

Figure 5: Visual word frequency histograms for each class in the train dataset

3.4 Training the Classifier

In the final step of our approach, we train 5 binary classifiers, one for each class, using the histograms obtained in the previous step. To do so, we utilize the scikit-learn implementation of the Support Vector Machines (SVM) classifier. Due to our dataset being highly imbalanced (positive:negative class ratio of 1:4), we employ class weights to the training process. We set these to be proportional to the number of samples in each class. This is done to ensure that the model does not overfit to the majority class and can still generalize well to the minority class. Additionally, we scrutinize multiple kernel types, namely linear, polynomial, and RBF, and eventually find the last one to be the most suitable for our task. We also experiment with different values for the regularization parameter C, which controls the reguralization of the model, and ultimately set its value to 1.0. In order to assess the quality of the trained models, we utilize the mean Average Precision (mAP) metric, which is commonly used among researchers to assess the quality of computer vision models. Moreover, we extend our analysis by computing the accuracy and F1-score of each model. The results of our experiments are presented in the next section.

4 Results

	Voc-Size: 1000 Sub-Ra: 30	**Voc-Size: 1000 Sub-Ra: 40**	**Voc-Size: 1000 Sub-Ra: 50**	**Voc-Size: 500 Sub-Ra: 40**	**Voc-Size: 1500 Sub-Ra: 40**
Airplanes	22.6	29.3	22.6	22.5	25.3
Birds	26.5	26.1	25.4	26.1	25.1
Ships	34.2	28.0	36.0	32.9	34.1
Horses	15.6	17.0	16.2	15.3	15.3
Automobiles	13.8	12.6	12.6	15.1	13.6
MAP Average	22.554	22.620	22.569	22.392	22.686

Table 1: SIFT MAPs, Vocabulary Size:, Subset Ratio:

	Voc-Size: 1000 Sub-Ra: 30	**Voc-Size: 1000 Sub-Ra: 40**	**Voc-Size: 1000 Sub-Ra: 50**	**Voc-Size: 500 Sub-Ra: 50**	**Voc-Size: 1500 Sub-Ra: 50**
Airplanes	19.3	17.5	19.3	19.1	19.6
Birds	17.7	20.5	18.8	19.9	18.0
Ships	23.3	22.5	19.7	22.3	20.9
Horses	19.5	21.7	24.0	19.0	21.2
Automobiles	21.0	18.5	19.1	20.3	20.9
MAP Average	20.155	20.140	20.171	20.128	20.121

Table 2: HoG MAPs, Vocabulary Size:, Subset Ratio:

The most effective model in our study is the SIFT algorithm, configured with a vocabulary size of 1500 words and a subset ratio of 40%. In Table 3, a detailed performance metrics, encompassing the accuracy, precision, recall and F1 scores of the individual classes and an overarching average are presented. For performing metrics of the other model configurations see the appendix. 6

	Airplanes	**Birds**	**Ships**	**Horses**	**Automobiles**	**Average**
Accuracy	68.5	65.6	68.6	74.5	74.2	70.3
Precision	29.2	24.4	29.9	31.9	27.7	28.6
Recall	41.5	34.4	42.1	23.7	17.6	31.9
F1 Score	34.3	28.6	35.0	27.2	21.5	21.5

Table 3: SIFT, Vocab Size: 1500, Subset Ratio: 40

Applying the most optimal model, the following figures 4 present a visualization of the top-5 and bottom-5 ranked test images for each class, based on the classifier's confidence for the target class.

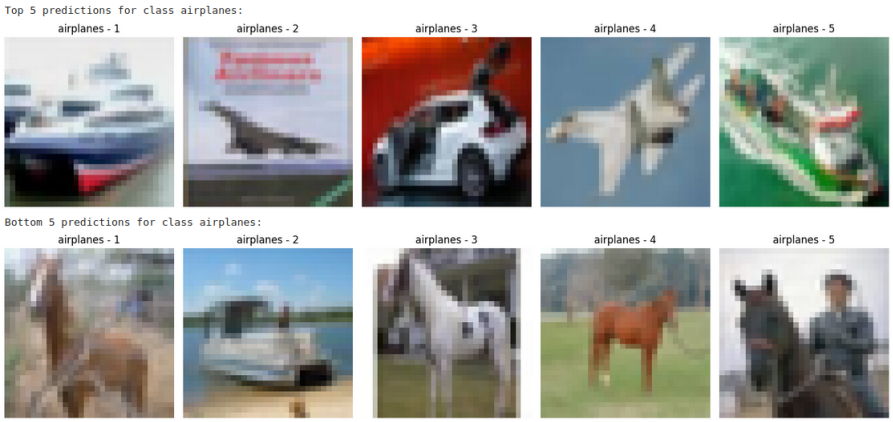


Figure 6: Top-5 and Bottom-5 Ranked Predicted Images for Class Airplanes

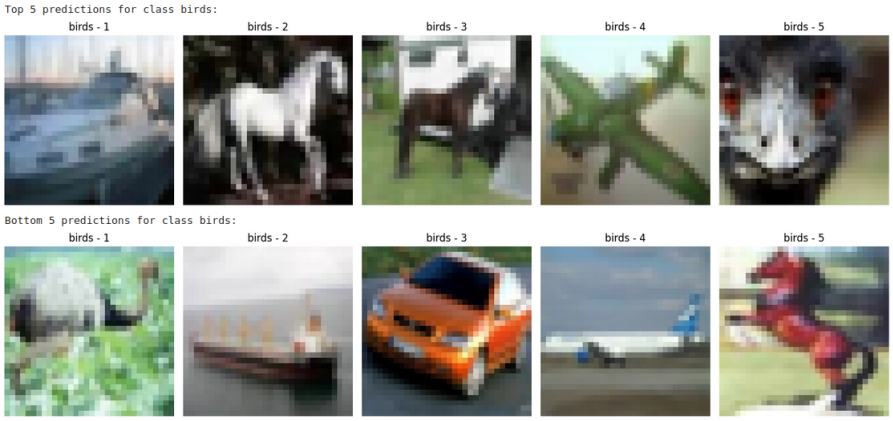


Figure 7: Top-5 and Bottom-5 Ranked Predicted Images for Class Birds

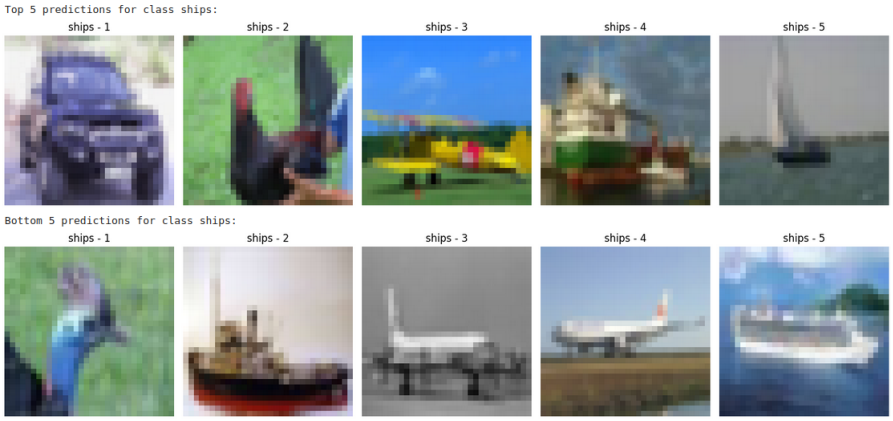


Figure 8: Top-5 and Bottom-5 Ranked Predicted Images for Class Ships

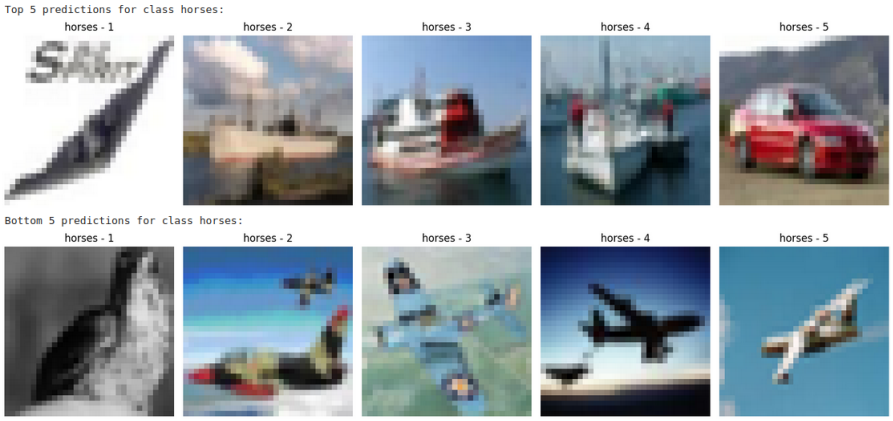


Figure 9: Top-5 and Bottom-5 Ranked Predicted Images for Class Horses

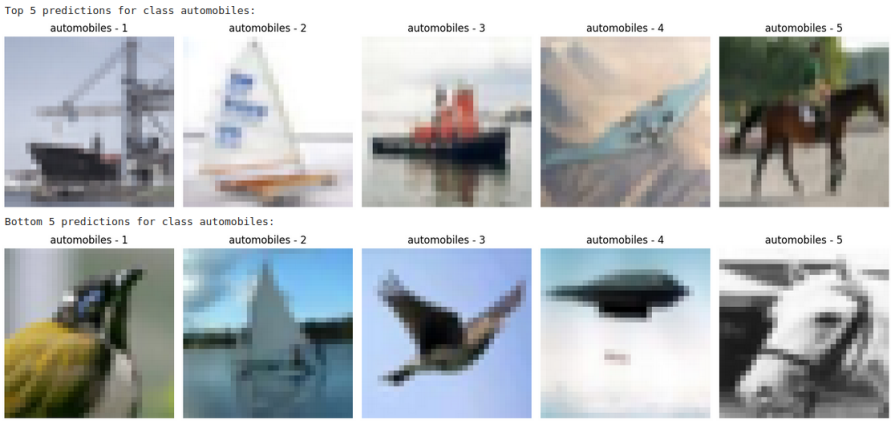


Figure 10: Top-5 and Bottom-5 Ranked Predicted for Class Automobiles

5 Discussion and Conclusion

In our exploration of the best parameters for both SIFT and HoG, a standard vocabulary size of 1000 words was employed to test subset ratios: 30%, 40%, and 50%. Based on the outcomes of these experiments, the most favorable subset ratio was identified and subsequently applied to evaluate two more vocabulary sizes: 500 and 1500. Unfortunately, none of these hyperparameter sets yielded satisfactory results. The best performing model was the one with the vocabulary size of 1500 and subset ratio of 40%, which achieved a mean Average Precision (mAP) score of 22.686%. The results of our experiments are presented in the following tables. Looking at the results, an intriguing observation arises from the performance of SIFT on the different classes. SIFT demonstrated significantly higher mAPs in the first three classes, but its performance sharply declined in the last two classes. This presents a puzzling challenge, particularly when considering that both SIFT and HoG methods are applied to the same dataset and HoG did not exhibit this Behaviour. Further investigation is needed to unravel the underlying factors contributing to this discrepancy. Then to explain the observed subpar results; As pointed out in the previous chapter, the utilized dataset is highly imbalanced which causes the model to overfit and leads to a rather poor performance. In order to mitigate this issue, we introduced a mechanism of class weights, which have been set to be proportional to the number of samples in each class. Looking ahead, there are numerous techniques for enhancing the models performance. Firstly, what has been noted before is that the dataset is highly imbalanced, which causes the model to overfit and leads to a rather poor performance. To alleviate this substantial issue, we could balance the dataset

by generating synthetic data. Not only can this be done by undersampling the majority class or oversampling the minority class, but also by performing data augmentation, which is a technique of applying various transformations to the existing data, such as rotation and translation, and producing new samples. Secondly, we could employ a different feature extractor algorithm, such as SURF or ORB, which could potentially yield better results. Lastly, we could utilize a more sophisticated classifier, such as a convolutional neural network or a transformer model, which are the current state-of-the-art choice for image classification tasks.

6 Appendix

	Airplanes	**Birds**	**Ships**	**Horses**	**Automobiles**	**Average**
Accuracy	67.6	64.2	66.1	73.0	71.9	68.6
Precision	27.8	23.7	28.1	30.2	26.3	27.2
Recall	39.7	35.4	44.3	26.4	22.3	33.6
F1 Score	32.7	28.3	34.4	28.2	24.1	29.5

Table 4: SIFT, Vocab Size: 1000, Subset Ratio: 30

	Airplanes	**Birds**	**Ships**	**Horses**	**Automobiles**	**Average**
Accuracy	66.8	64.4	67.7	72.5	72.5	68.8
Precision	27.0	24.6	29.1	31.0	26.8	27.7
Recall	39.6	37.8	42.9	30.0	21.4	34.3
F1 Score	32.1	29.8	34.7	30.5	23.8	30.4

Table 5: SIFT, Vocab Size: 1000, Subset Ratio: 40

	Airplanes	**Birds**	**Ships**	**Horses**	**Automobiles**	**Average**
Accuracy	66.8	64.0	67.6	73.3	73.4	69.0
Precision	27.2	23.1	29.2	31.7	28.7	28.0
Recall	40.2	34.1	43.6	28.6	22.0	33.7
F1 Score	32.4	27.5	35.0	30.1	24.9	30.2

Table 6: SIFT, Vocab Size: 1000, Subset Ratio: 50

	Airplanes	Birds	Ships	Horses	Automobiles	Average
Accuracy	67.2	63.6	66.9	70.8	70.6	67.8
Precision	28.2	24.3	29.3	29.5	25.1	25.1
Recall	42.1	38.6	46.2	32.7	23.4	36.4
F1 Score	33.8	29.8	35.9	31.0	24.2	30.9

Table 7: SIFT, Vocab Size: 500, Subset Ratio: 40

	Airplanes	Birds	Ships	Horses	Automobiles	Average
Accuracy	68.5	65.6	68.6	74.5	74.2	70.3
Precision	29.2	24.4	29.9	31.9	27.7	28.6
Recall	41.5	34.4	42.1	23.7	17.6	31.9
F1 Score	34.3	28.6	35.0	27.2	21.5	21.5

Table 8: SIFT, Vocab Size: 1500, Subset Ratio: 40

	Airplanes	Birds	Ships	Horses	Automobiles	Average
Accuracy	72.6	73.7	72.0	72.8	71.9	72.6
Precision	20.7	22.0	22.7	20.9	20.7	21.4
Recall	13.0	12.3	16.5	12.9	14.3	13.8
F1 Score	16.0	15.8	19.1	16.0	16.9	16.8

Table 9: HoG, Vocab Size: 1000, Subset Ratio: 30

	Airplanes	Birds	Ships	Horses	Automobiles	Average
Accuracy	73.7	72.4	72.5	72.1	72.4	72.6
Precision	23.2	20.3	23.0	19.9	18.7	21.0
Recall	13.7	12.9	15.9	13.0	11.4	13.4
F1 Score	27.2	15.8	18.8	15.7	14.2	18.3

Table 10: HoG, Vocab Size: 1000, Subset Ratio: 40

	Airplanes	Birds	Ships	Horses	Automobiles	Average
Accuracy	73.5	72.5	72.8	73.3	72.5	72.9
Precision	21.4	18.5	24.7	21.9	20.2	21.3
Recall	12.2	11.0	17.7	13.0	12.6	13.3
F1 Score	15.6	13.8	20.6	16.3	15.5	16.3

Table 11: HoG, Vocab Size: 1000, Subset Ratio: 50

	Airplanes	Birds	Ships	Horses	Automobiles	Average
Accuracy	70.5	70.9	70.1	70.4	70.6	70.5
Precision	21.5	22.9	21.9	24.7	20.3	22.3
Recall	18.0	19.1	19.2	23.6	16.1	19.2
F1 Score	19.6	20.8	20.5	24.2	18.0	20.6

Table 12: HoG, Vocab Size: 500, Subset Ratio: 50

	Airplanes	Birds	Ships	Horses	Automobiles	Average
Accuracy	74.3	75.3	74.3	74.6	74.1	74.5
Precision	19.1	22.8	21.8	18.8	17.8	20.1
Recall	8.8	9.9	11.1	8.1	8.2	9.2
F1 Score	12.1	13.8	14.7	11.3	11.2	12.6

Table 13: HoG, Vocab Size: 1500, Subset Ratio: 50