UNIVERSITY OF AMSTERDAM

FINAL LAB PART 1

Keypoint Detection, Bag of Visual Words and Image Classification

October 18, 2024

Students:
Lisanne Wallaard
15735664

Jose L Garcia
15388867

Julio Smidi
11307943

Group:
Group 9

Lecturer:
Martin Oswald and Arnoud Visser

Course:
Computer Vision 1

1 Introduction

In computer vision, one of the most fundamental challenges is accurately classifying objects in images. Solving this issue opens doors to many different types of applications, such as autonomous driving, disease detection (e.g., cancer and tumours), and monitoring a diverse range of objects in different environments. However, building a robust image classifier is particularly hard, especially when working with diverse datasets containing thousands of objects across vastly different categories.

This report addresses this challenge by implementing an image classification system that can identity objects from five different classes (Frog, Automobile, Bird, Cat and Deer) using the publicly available CIFAR-10 dataset. The solution implements the Bag of Visual Words (BoVW) method, which constructs a visual dictionary from different local features extracted from the images. These features are then transformed into histograms, which are used to train a classifier to predict the class of an image based on its visual content.

The project consisted of several key steps: First, detecting and extracting key points in the images using two different techniques, SIFT and ORB. Then, creating the visual dictionaries by clustering the features using the K-Means method. Next, these features were encoded into histograms of visual words, which were used in the training of a One-vs-Rest (OvR) Support Vector Machine (SVM) classifier. Finally, the classifier's performance was evaluated quantitatively using the mean Average Precision (mAP) metric.

Through the selected approach, the project aims to provide a deeper understanding of the pipelines that must be followed for image classification and explore the accuracy and potential of a method such as BoVW to do so for different object categories.

2 Methods

2.1 CIFAR-10 Dataset

Our first task was to use an appropriate dataset for image classification. The chosen dataset, the CIFAR-10 dataset, consists of 60.000 mutually exclusive 32x32 colour images divided across 10 labelled classes, with 6.000 images per class. The dataset is pre-divided into 50.000 training images, with 5.000 randomly selected images from each class, and 10.000 testing images, with 1.000 images

per class. The 10 classes included in the CIFAR-10 dataset are airplanes, automobiles, birds, cars, deers, dogs, frogs, horses, ships and trucks 1.

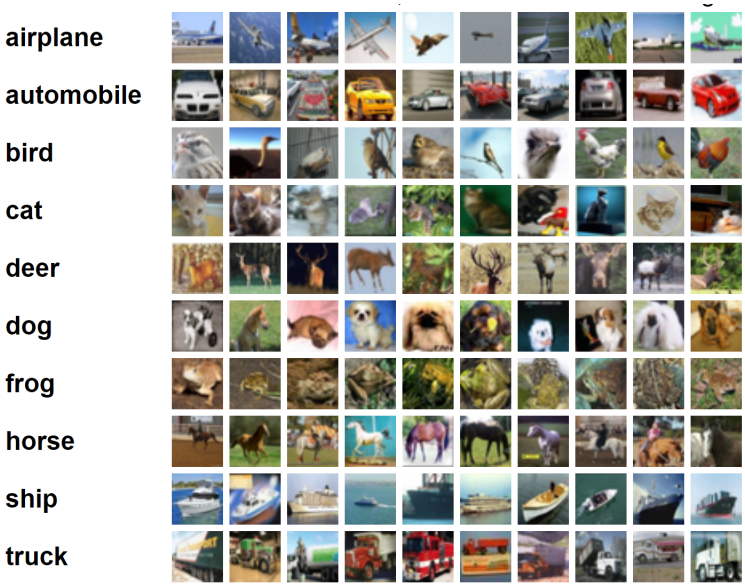


Figure 1: CIFAR-10 dataset classes

However, for this project, we were tasked to focus on a subset of five of these categories: automobiles, birds, cats and deers and frogs.

Having downloaded the dataset, we proceeded to perform a series of tasks to implement the Bag of Visual Words (BoVW) method.

2.2 Keypoint Detection and Feature Extraction

First, we needed to extract the key visual features or points from the selected images. Key points represent different points in the image where significant changes in intensity happen, such as edges or corners.

For this step, we employed two different feature extraction techniques included in the OpenCV Python library: Scale-Invariant Feature Transform (SIFT) and Oriented FAST and Rotated BRIEF (ORB). Both algorithms are used for feature detection and description of an image, but they differ on some key elements:

- SIFT is an accurate and robust algorithm that is invariant to changes in scale and rotation despite noise and illumination irregularities, making it effective for most cases. It uses a Difference of Gaussians (DoG) function to identify the key points, and then, after assigning an orientation to each key point based on its surrounding pixels, creates detailed descriptors based on their gradient orientations.
- ORB, on the other hand, combines two methods:
  - the Features from Accelerated Segment Test (FAST) key point detector, which identifies corners by examining the pixel intensities around a point.
  - the Rotated Binary Robust Independent Elementary Features (BRIEF) descriptor, with an added orientation handling component to make it rotation-invariant.

Both algorithms are widely used for detecting key points in images, with SIFT being known for its robustness against scaling and rotation, and ORB for being a computationally efficient and faster alternative.

Using the mentioned techniques, we detected the key points in each image of the dataset. After detecting key points, we extracted feature descriptors that describe the local appearance around

each key point. For visualization, we plotted key points on sample images from each class to observe how the two extraction techniques behaved across different categories, as seen in figure 2.

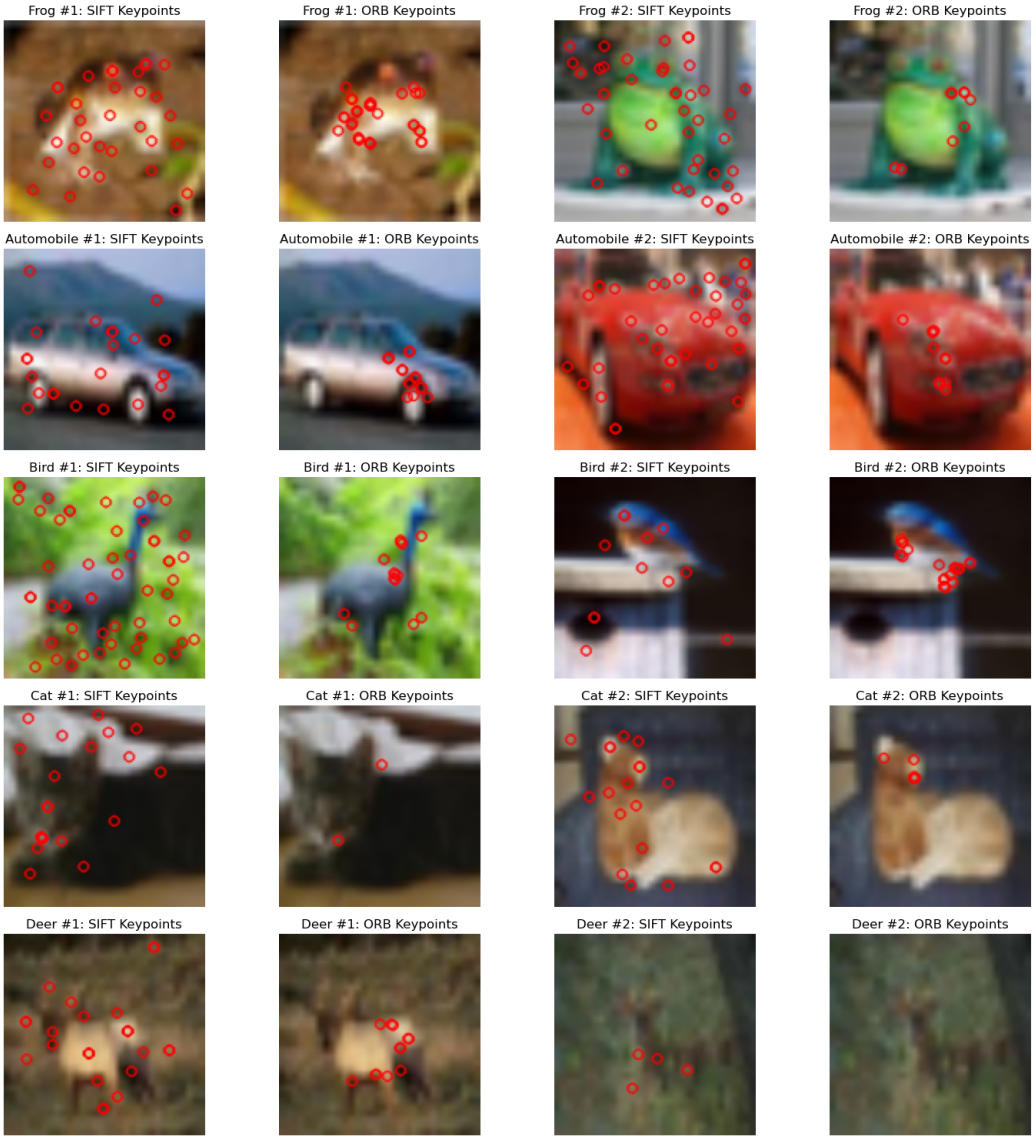


Figure 2: Extracted key points from sample images of each class using the SIFT and ORB techniques.

These feature descriptors were used as the basis for creating a visual vocabulary in the next step.

2.3 Building the Visual Vocabulary

Next, we constructed the visual vocabulary by applying K-Means clustering to the extracted feature descriptors, grouping them into a fixed number of clusters. These clusters represented the "visual words", where each cluster centre represents a particular visual pattern.

For this project, we used 1.000 clusters to build the visual vocabulary and we experimented with three different subset sizes of the training images (30%, 40%, and 50%) to understand how the amount of training data influences the quality of the clusters. To speed up the clustering process, we utilized the 'faiss' library, which is optimized for large-scale K-Means clustering.

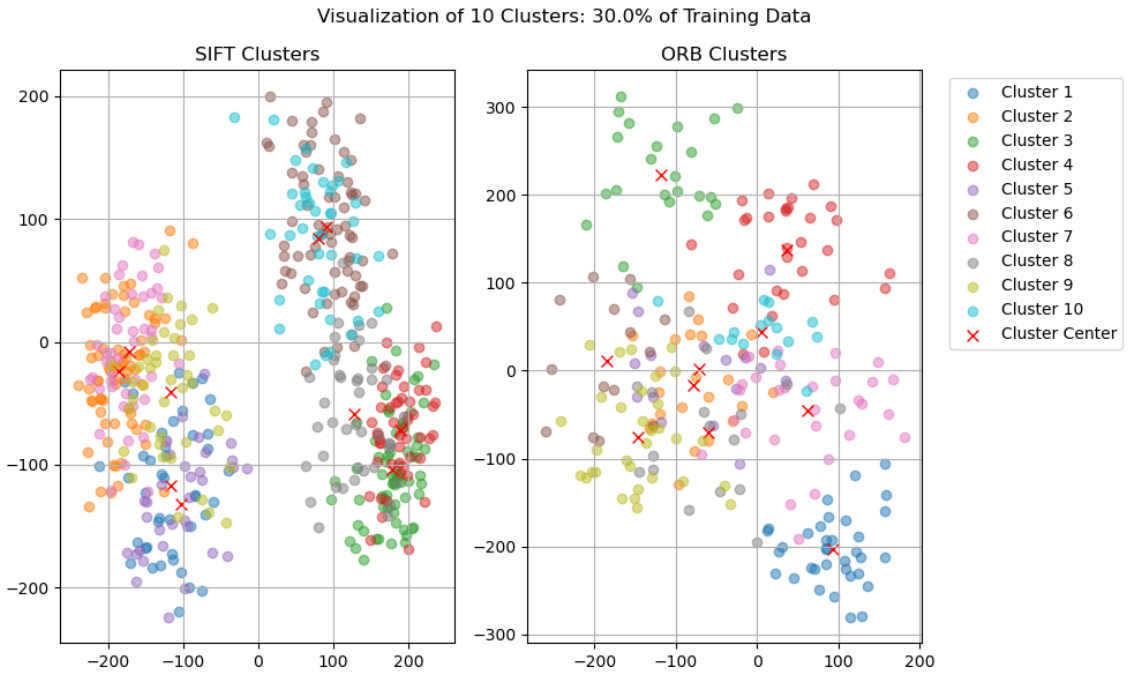


Figure 3: Visualization of 10 Clusters: 30% of Training Data.

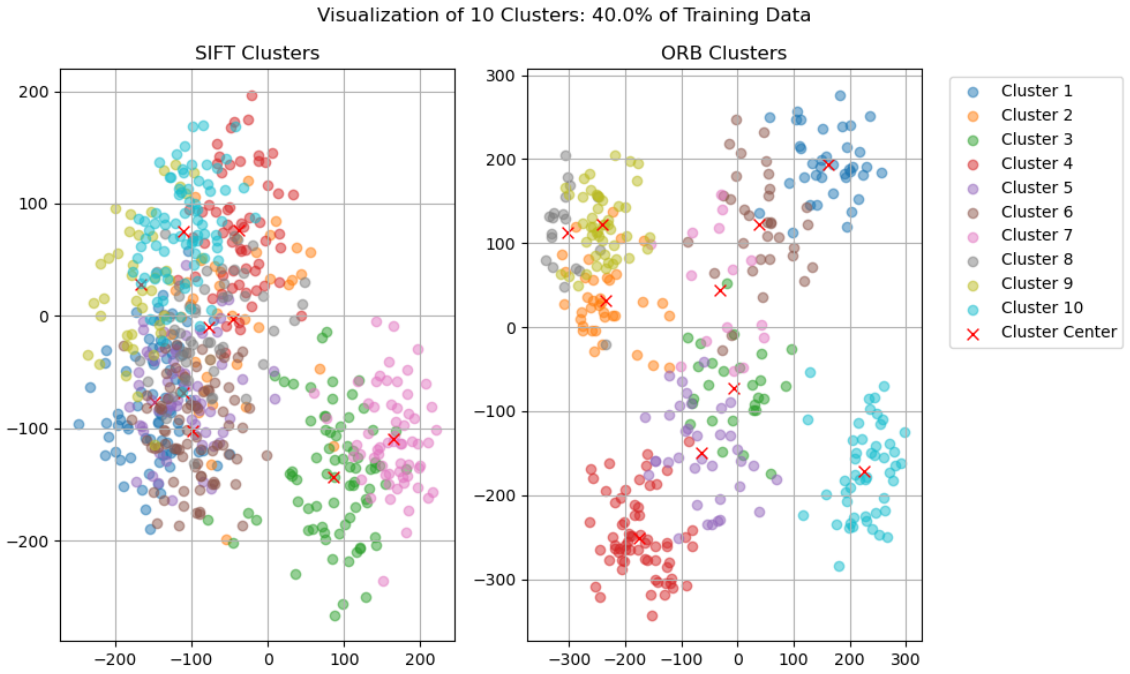


Figure 4: Visualization of 10 Clusters: 40% of Training Data.

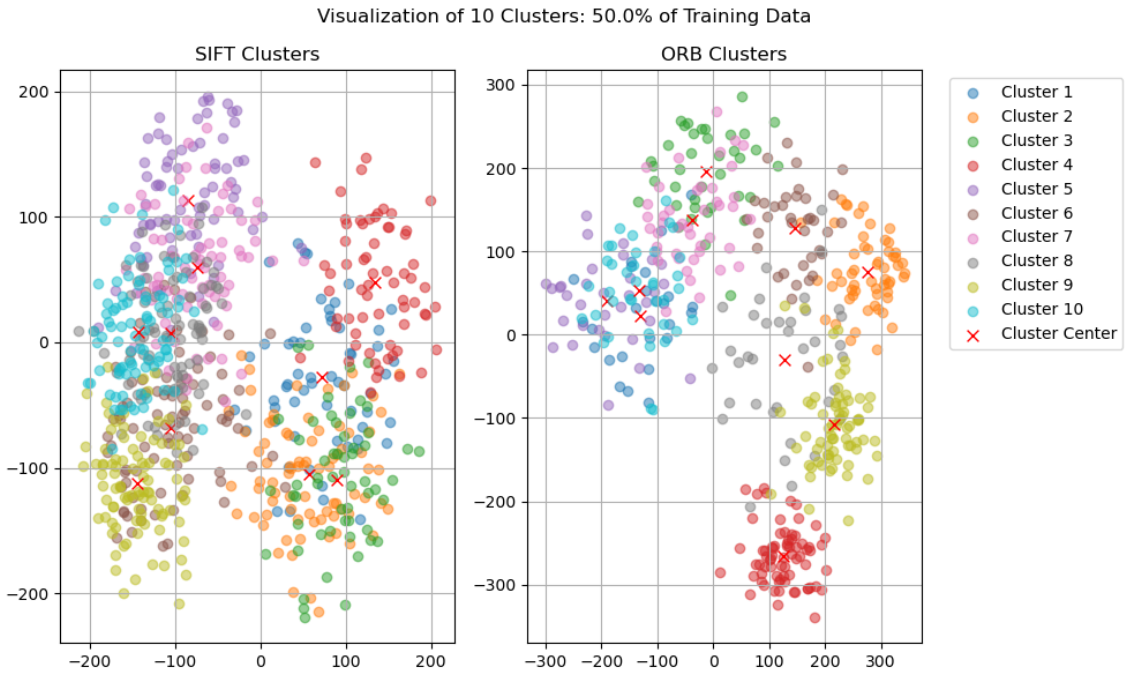


Figure 5: Visualization of 10 Clusters: 50% of Training Data.

After creating the visual vocabulary, we visualized the first 10 clusters for each feature extraction technique using Principal Component Analysis (PCA) to reduce the dimensionality of the descriptors and show them in 2D scatter plots, as seen in Figures 3 4 5.

2.4 Encoding the Image Features and Visualizing the Bag of Visual Words

Subsequently, we encoded the entire image training and test data set into histograms using the visual dictionary trained on 50% of the training dataset. A histogram of a certain image shows the frequency of each visual world (from the visual library) in that image. In total, the visual dictionary has a thousand visual words.

First, we extracted feature descriptors from each image by using both the SIFT as the ORB technique. Each descriptor of an image (for both extraction techniques) is matched to the closest cluster center (visual word) in the visual library using the `faiss` library due to its efficiency. The histograms are being normalized to compare histograms of images with a different number of keypoints.

We visualized the mean histogram of the images in the entire training dataset for each of the 5 classes in figure 6 & 7. In figure 6, the mean encoded histograms using the SIFT technique are shown. The mean histograms show different patterns for each class, which is a good thing as it enables us to make a distinction between the classes through the histogram. Figure 7 shows the mean encoded histogram using the ORB technique. Also, these mean histograms illustrate different shapes for different classes. Notably, the plots are quite different from the ones using the SIFT technique for the same class. This is probably because both techniques extract different keypoints and therefore different descriptors, which results in matching with different cluster centers (visual words).

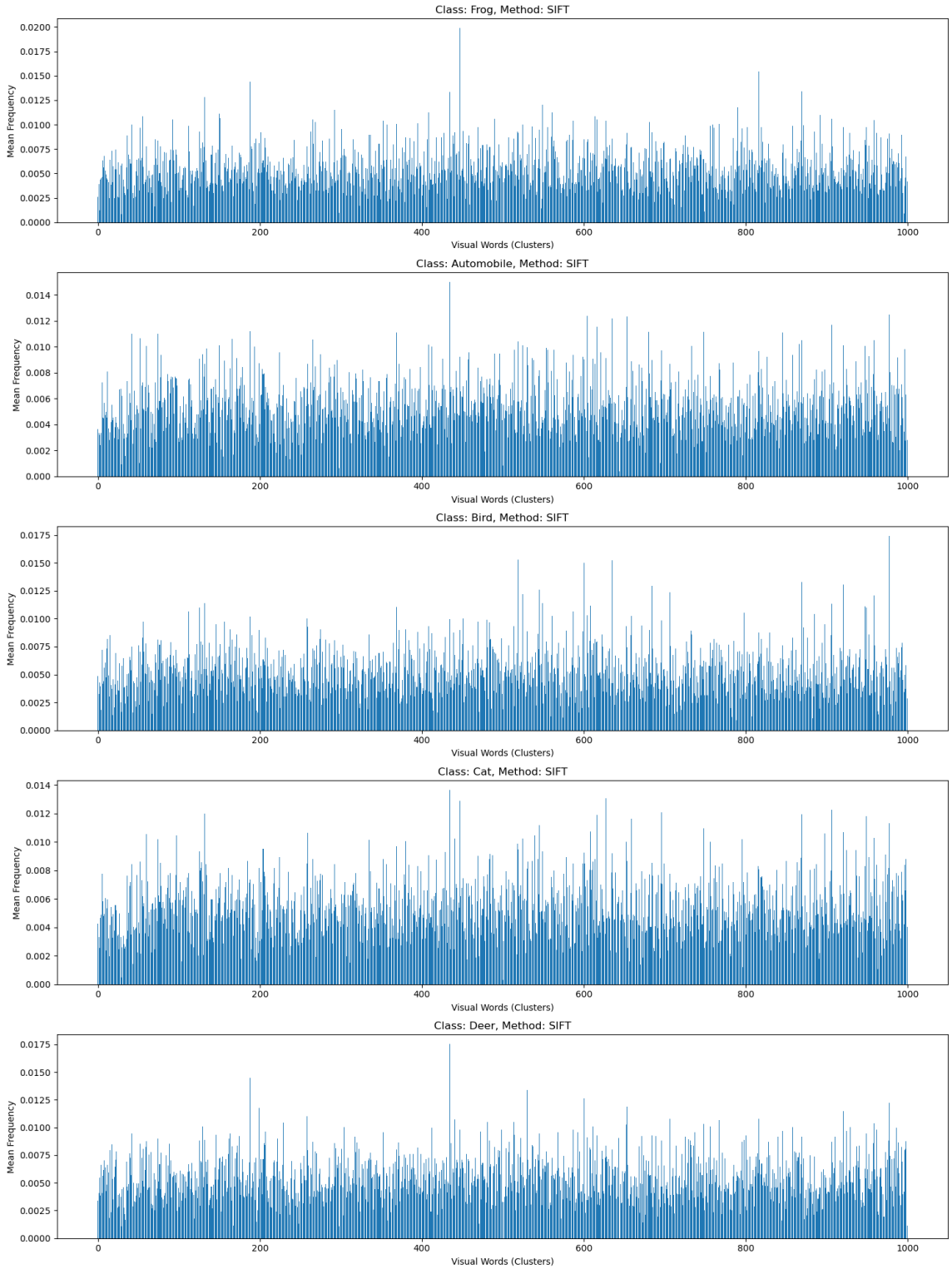


Figure 6: Mean histogram of training dataset using the SIFT feature extraction technique.

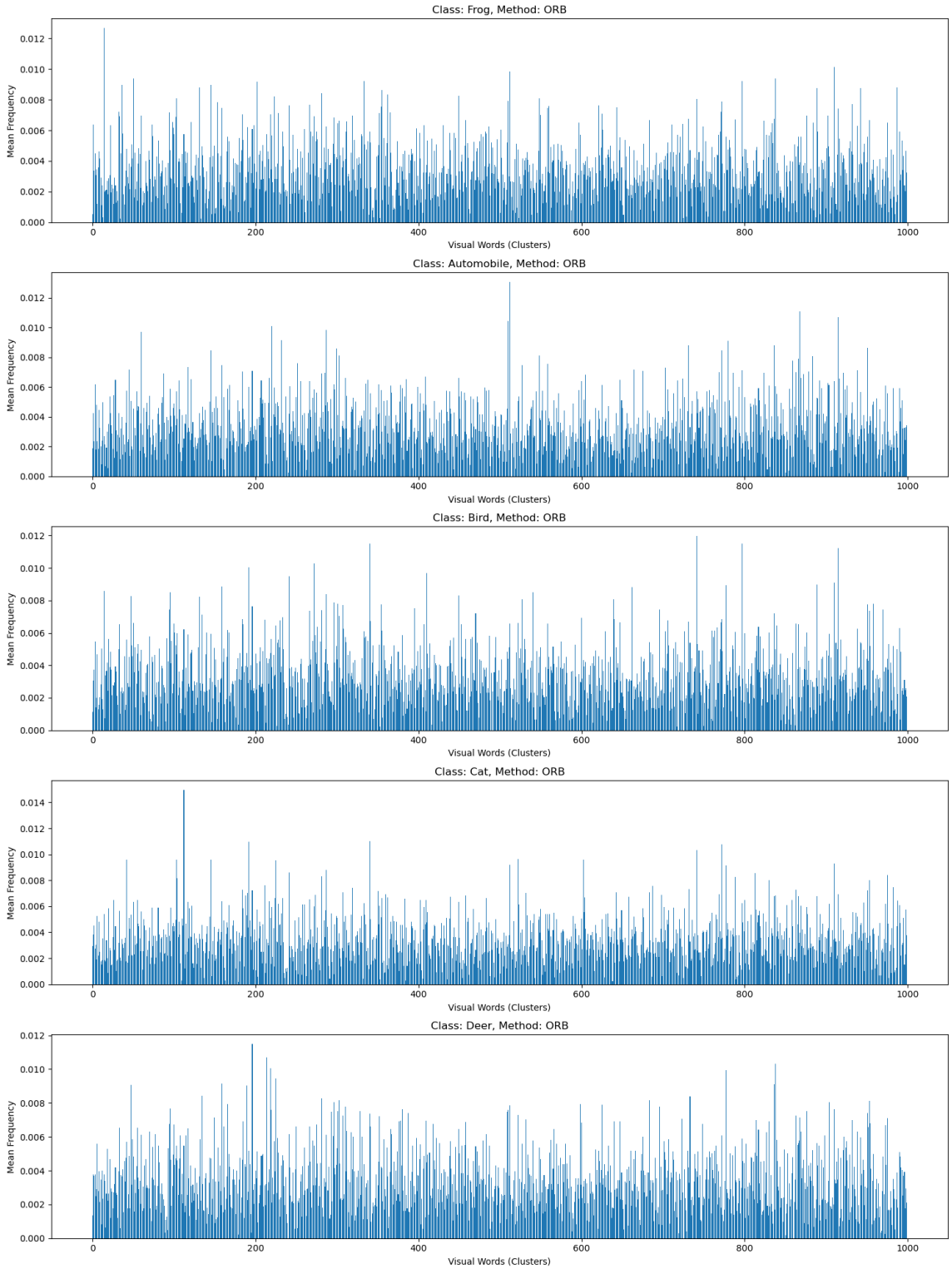


Figure 7: Mean histogram of training dataset using the ORB feature extraction technique.

2.5 Training the Classifiers

Now, we are able to train two One-vs-Rest SVM classifiers (one for each feature extraction technique), using half of the encoded training images that were not used for training the visual dic-

tionary. By utilizing the other half of the training set, we get a more reliable evaluation of the performance of the visual library.

In order to evaluate the performance of these SVM classifiers, we ranked the classified test images based on their confidence score and calculated the mean Average Precision (mAP_c) for each class:

$$\text{mAP}_c = \frac{1}{N_c} \sum_{i=1}^{N} \frac{f_c(x_i)}{i}$$

- N: total number of test images (1000)
- N_c: number of test images in class c (200)
- x_i: the i^{th} test image in list $X = \{x_1, x_2, \ldots, x_N\}$ ranked by the confidence scores for class c
- $f_c = \begin{cases} \text{the number of test images of class } c \text{ in the first } i \text{ test images,} & \text{if } x_i \text{ belongs to class } c \\ 0, & \text{otherwise} \end{cases}$

Then, we calculated the mAP across all classes by averaging the mAP_c over the five classes.

2.6 Tuning the Classifier Hyperparameters

To search for the optimal set of hyperparameters, we performed a grid search to identify the parameter settings that maximized the mAP. The tuned hyperparameters and their associated grid search values are listed in Table 1.

Parameter	List of values
Number of visual words	[500, 1000, 1500]
Training set size	[30, 40, 50]
Kernel type	[Linear, Radial Basis Function (RBF)]
C (regularization parameter)	[0.1, 1, 10]
Gamma (kernel coefficient for RBF)	[0.01, 0.001, 'scale']

Table 1: Hyperparameter variables and values used in grid search.

3 Results

3.1 Classifier Evaluation

We evaluated the performance of the SIFT- and ORB-based SVM classifiers on the CIFAR-10 test set, consisting of a 1000 test images. The average mAP for the SIFT classifier across all classes is approximately 0.398, while the ORB classifier class average mAP is approximately 0.229. Over all classes, we can see that the SIFT classifier performs significantly better than the ORB classifier on the CIFAR-10 images.

For a more qualitative investigation of the classifier performance, we look at class-specific predictions for the SIFT and ORB classifiers. In Figure 8, the top and bottom five ranked images for a certain class, based on the confidence score of the classifier, are shown for the "automobile" class. Other classes showed similar results, and can be seen in Figure 10 in the appendix. Three out of the five top-ranked images are in fact automobiles, while two are not. For the bottom five ranked images none of the images are automobiles. Here we can see the classifier accuracy is definitely not perfect, as is reflected by the mAP score. However, from this qualitative investigation, we can see that the confidence score of the classifier is quite indicative of its accuracy.

For the top and bottom five ranked images for the ORB classifier (Figure 9, we see that there are only two automobiles out of the top five ranked images, and there is one automobile in the bottom five ranked images. For this last image, the classifier is very unsure of whether it is an automobile. We can therefore cautiously say that the confidence score for the ORB classifier is a little less indicative of its performance compared to the SIFT classifier, and also here the difference in mAP between the feature extraction techniques can be seen.

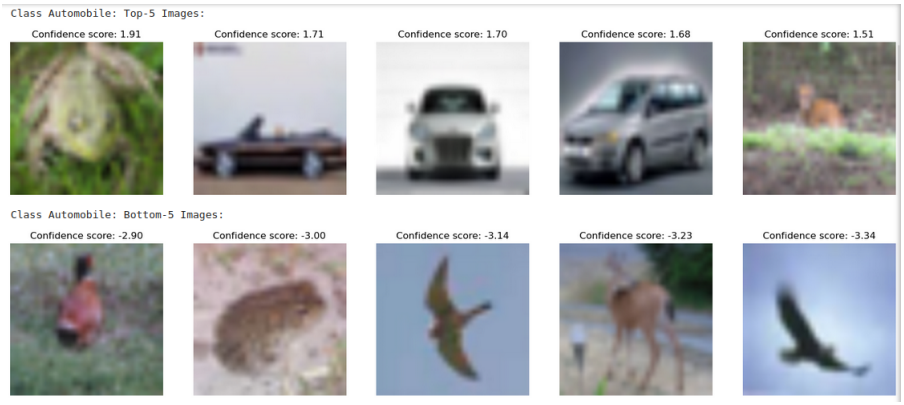


Figure 8: SIFT top and bottom five ranked images for the "automobile" class.

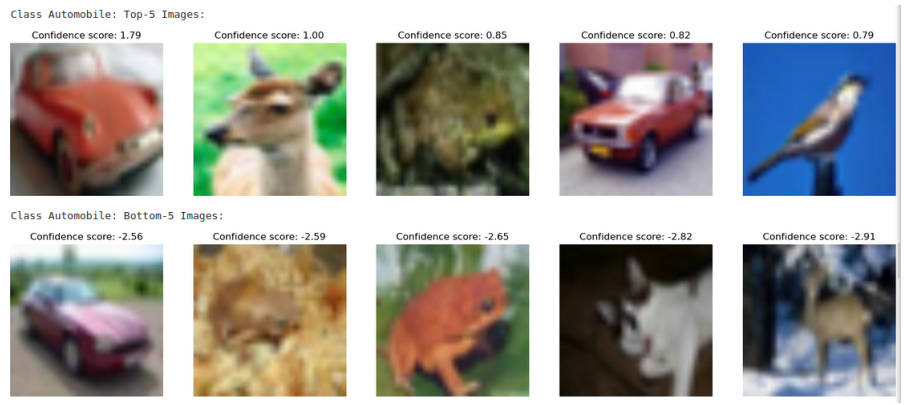


Figure 9: ORB top and bottom five ranked images for the "automobile" class.

3.2 Hyperparameter Tuning

The results of the grid search for the optimal hyperparameters for the SIFT models have been summarized in Table 2, for the ORB models they are shown in 3. As the total set of results is too large to add to this section, a selection of the combination of options has been shown here, including the tunings resulting in the lowest and highest mAP.

The biggest improving trend that can be seen from this data, is that in general, using radial basis function (RBF) kernels give higher mAP scores that using linear kernels in the SVM. There also seems to be a slight improvement when the size of the training set used for clustering is smaller, which could be due to the classifier then using a larger part of the train set. Also, it seems that keeping the number of visual words used on the lower side, as results are higher overall for 500 words, than for 1000 or 1500 words.

4 Discussion

Overall, we found that the SIFT-based classifiers did significantly better than the ORB-based classifiers, both performing above chance level ($\sim$ 20% for 5 classes), but there the performance of the models was limited. Changes in other parameters (e.g. training set size, number of visual words used) led to some improvements of the models, but those were relatively small.

Multiple explanations can be given for the performance of the models. One limitation is that BoVW models suffer from a fixed vocabulary, where too small a vocabulary can lead to low discriminative power, and one too big can lead to overfitting issues. We attempted to counter this problem by including this variable in the parameter grid search.

Another potential downside of BoVW models, is that they often cannot encode information

Parameters	mAP
C: 10; Gamma: scale; kernel: linear; visual words: 1500; train size: 30%	**0.336**
C: 10; Gamma: scale; kernel: RBF; visual words: 1500; train size: 50%	0.411
C: 10; Gamma: 0.01; kernel: linear; visual words: 500; train size: 50%	0.384
C: 0.1; Gamma: 0.01; kernel: linear; visual words: 1000; train size: 40%	0.416
C: 0.1; Gamma: 0.001; kernel: RBF; visual words: 500; train size: 30%	0.425
C: 10; Gamma: 0.001; kernel: RBF; visual words: 1500; train size: 30%	0.410
C: 10; Gamma: scale; kernel: linear; visual words: 1500; train size: 40%	0.347
C: 0.1; Gamma: scale; kernel: RBF; visual words: 1000; train size: 30%	**0.450**

Table 2: Summarized results of the hyperparameter tuning grid search for SIFT (lowest and highest values in bold)

Parameters	mAP
C: 0.1; Gamma: 0.001; kernel: RBF; visual words: 500; train size: 40%	**0.232**
C: 0.1; Gamma: scale; kernel: linear; visual words: 1500; train size: 30%	0.243
C: 1; Gamma: scale; kernel: linear; visual words: 1500; train size: 50%	0.258
C: 10; Gamma: 0.001; kernel: RBF; visual words: 1000; train size: 50%	0.255
C: 1; Gamma: 0.01; kernel: RBF; visual words: 1000; train size: 40%	0.254
C: 1; Gamma: 0.01; kernel: RBF; visual words: 500; train size: 30%	0.269
C: 1; Gamma: scale; kernel: RBF; visual words: 500; train size: 30%	**0.276**

Table 3: Summarized results of the hyperparameter tuning grid search for ORB (lowest and highest values in bold)

about spatial relations, although this issue can be partially improved by using keypoint detection algorithms such as SIFT (Koniusz, Yan, and Mikolajczyk 2013). There is also still uncertainty whether BoVW models are viewpoint and scale invariant, which might not aid the classification.

We do see that the SIFT models consistently outperform the ORB models based on the mAP scores. This is supported by findings from Karami, Prasad, and Shehata (2017), who found SIFT performs better overall for most transformations and distortions in images.

5 Conclusion

In this project we have investigated two keypoint detection algorithms, SIFT and ORB, and have used those to create visual bag-of-word dictionaries of the features using K-means clustering. A histogram encoding of these features was then used to train and evaluate different OvR SVM classifiers. Additionally, we performed hyperparameter tuning to find the optimal parameters for the classifiers.

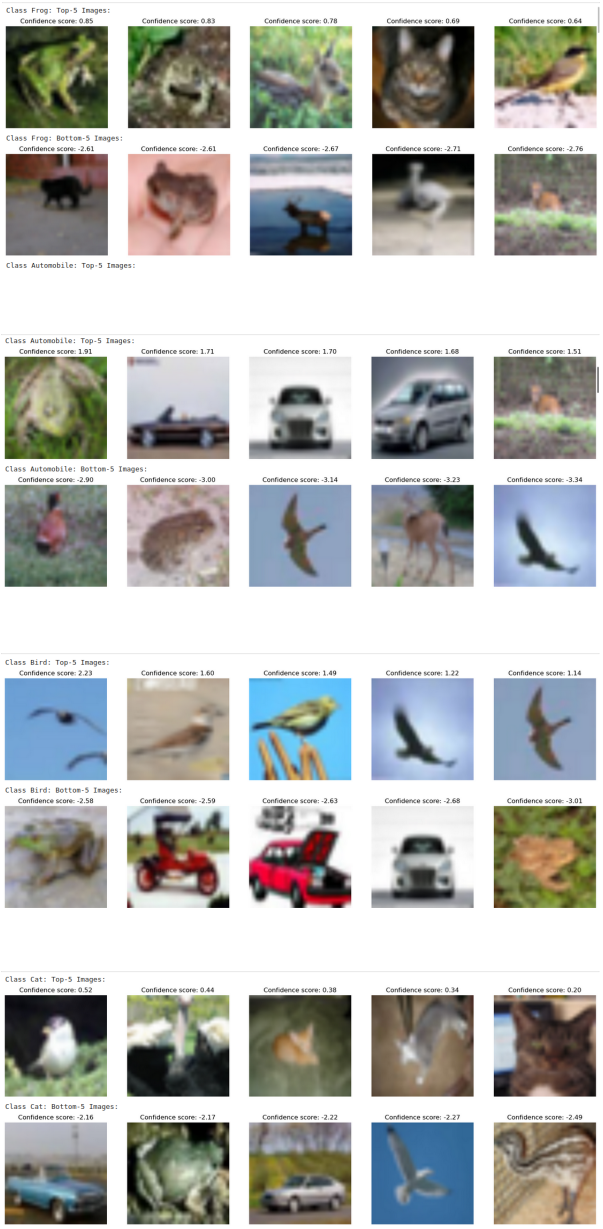
References

Karami, Ebrahim, Siva Prasad, and Mohamed Shehata (2017). "Image matching using SIFT, SURF, BRIEF and ORB: performance comparison for distorted images". In: *arXiv preprint arXiv:1710.02726*.

Koniusz, Piotr, Fei Yan, and Krystian Mikolajczyk (2013). "Comparison of mid-level feature coding approaches and pooling strategies in visual concept detection". In: *Computer vision and image understanding* 117.5, pp. 479–492.

A Appendix LaTeX code

Code and References.



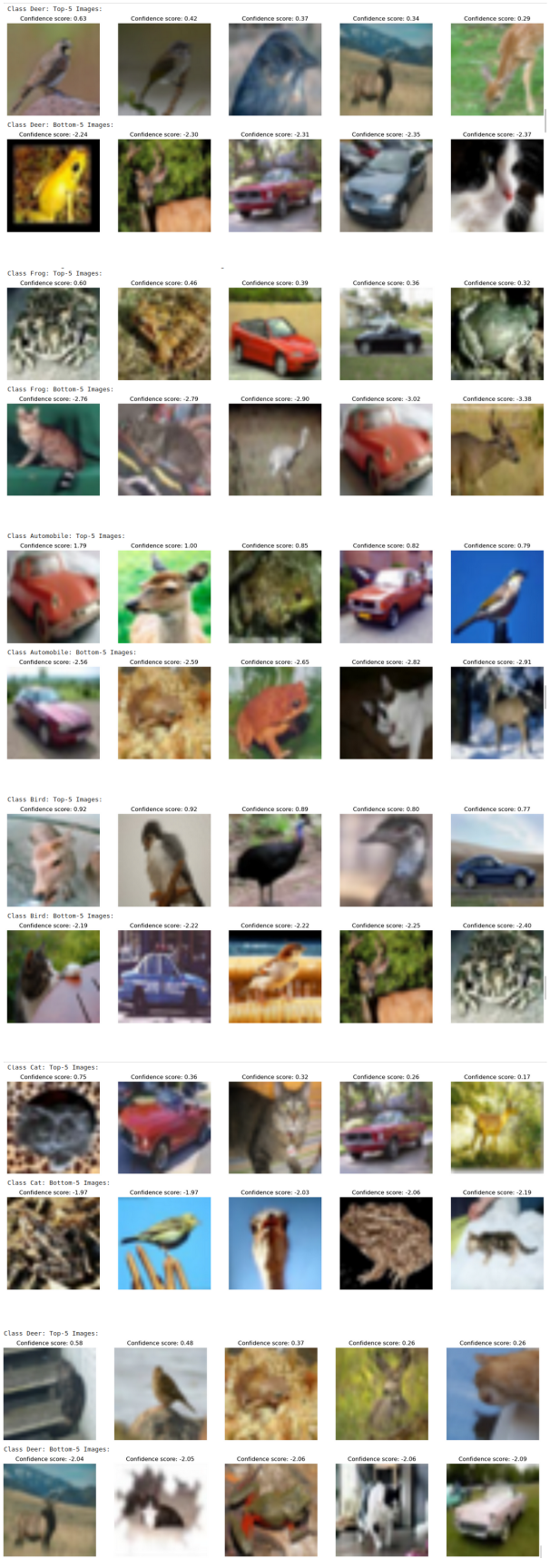

Class Deer: Top-5 Images:
Confidence score: 0.63
Confidence score: 0.42
Confidence score: 0.37
Confidence score: 0.34
Confidence score: 0.29
Class Deer: Bottom-5 Images:
Confidence score: -2.24
Confidence score: -2.30
Confidence score: -2.31
Confidence score: -2.35
Confidence score: -2.37
Class Frog: Top-5 Images:
Confidence score: 0.60
Confidence score: 0.46
Confidence score: 0.39
Confidence score: 0.36
Confidence score: 0.32
Class Frog: Bottom-5 Images:
Confidence score: -2.76
Confidence score: -2.79
Confidence score: -2.90
Confidence score: -3.02
Confidence score: -3.38
Class Automobile: Top-5 Images:
Confidence score: 1.79
Confidence score: 1.00
Confidence score: 0.85
Confidence score: 0.82
Confidence score: 0.79
Class Automobile: Bottom-5 Images:
Confidence score: -2.56
Confidence score: -2.59
Confidence score: -2.65
Confidence score: -2.82
Confidence score: -2.91
Class Bird: Top-5 Images:
Confidence score: 0.92
Confidence score: 0.92
Confidence score: 0.89
Confidence score: 0.80
Confidence score: 0.77
Class Bird: Bottom-5 Images:
Confidence score: -2.19
Confidence score: -2.22
Confidence score: -2.22
Confidence score: -2.25
Confidence score: -2.40
Class Cat: Top-5 Images:
Confidence score: 0.75
Confidence score: 0.36
Confidence score: 0.32
Confidence score: 0.26
Confidence score: 0.17
Class Cat: Bottom-5 Images:
Confidence score: -1.97
Confidence score: -1.97
Confidence score: -2.03
Confidence score: -2.06
Confidence score: -2.19
Class Deer: Top-5 Images:
Confidence score: 0.58
Confidence score: 0.48
Confidence score: 0.37
Confidence score: 0.26
Confidence score: 0.26
Class Deer: Bottom-5 Images:
Confidence score: -2.04
Confidence score: -2.05
Confidence score: -2.06
Confidence score: -2.06
Confidence score: -2.09

Figure 10: