Computer Vision 1 - Final Lab Report 1

Student1 ID: 13213334
Student1 Name: Jesse Wonnink

Student2 ID: 15816397
Student2 Name: Ruben Figge

Student3 ID: 12856320
Student3 Name: Chileshe Lukwesa

October 18, 2024

1 Abstract

Bag of Visual Words (BoVW) is one of the simplest methods of representing image features for the task of image classification. This paper investigates the performance of one-versus-the-rest image classifiers using a sample of images belonging to five classes drawn from the CIFAR-10 dataset. In doing so, we also investigate the effect on classifier performance of extracting image features using two different techniques – Scale-Invariant Feature Transform (SIFT) and Oriented FAST and Rotated BRIEF (ORB). For each feature extraction method, we train one-versus-the-rest binary SVM classifiers and evaluate performance using the mean average precision (mAP) achieved across all five image classes. Although the performance of the classifiers is higher when using SIFT-generated image features, we find that, overall, a BoVW encoding of image features results in poor classification performance, with the highest achieved mAP being only about 0.47.

2 Introduction

Bag-of-words models are a frequently used method in several machine learning fields due to their lax assumptions and ease of implementation [1]. This is also the case in the field of Computer Vision, where using images as an input for machine learning models presents several challenges due to factors that complicate the retrieval and processing of image data, including issues relating to perspective, scale differences, occlusion and contextual understanding. Bag-of-visual-words models (BoVW), where extracted image features are encoded

based on features found in all images in the dataset, became popular in the 2000's, providing an efficient and quick approach to represent useful features from an image.

However, because BoVW representations ignore the spatial relations among image features, advances in deep learning methods that can maintain such relations, such as Convolutional Neural Networks (CNNs), have resulted in models based on BoVW techniques losing ground: The release of the AlexNet CNN architecture in 2012 [2] marked a turning point for the use of deep learning for computer vision tasks, as it showed that adaptively learning image features is a more effective approach to image analysis which also takes into account the images' local context. Although advances in Computer Vision and deep learning, in particular, have led to BoVW as a technique falling out of favor, we find that it is nevertheless worth investigating, as it allows the use of simpler classification methods such as Support Vector Machines (SVMs) to classify images.

Before image features can be represented using BoVW, it is necessary to first extract meaningful features from the images. Although there have long been methods available to represent images without extracting features, such as global descriptors, including color distribution histograms [3] or edge detectors [4], these methods usually only form the initial part of an image analysis pipeline, and for tasks like image classification, which we investigate in this paper, feature extraction represents a crucial step. This is why, in addition to investigating BoVW as a feature representation method for image classification, we also investigate different techniques for extracting useful features from images.

This report focuses on the performance of image classification using BoVW as a technique for representing image features. We investigate two methods for extracting image features: Scale-Invariant Feature Transform (SIFT) [5], introduced in its current form in 2004, and Oriented FAST and Rotated BRIEF (ORB), introduced in 2011 as a more computationally efficient feature extraction alternative to earlier methods such as SIFT [6]. We then use these extracted features to create BoVW image representations. In order to compare the two feature extraction techniques, we perform a hyperparameter search for two sets of binary SVM classifiers trained on image features generated using each method then compare the mean average precision (mAP) achieved for each technique (using the tuned hyperparameters) across all five image classes used in our research.

3 Dataset

The research presented in this paper uses a subset of the CIFAR-10 dataset [7]. This dataset is a staple in image recognition, and although it may be easy for deep learning approaches to classify, we use it in our research as we are relying on more traditional SVM classifiers. Due to time and computational constraints associated with conducting hyperparameter tuning for the many different sets of possible hyperparameters used in our research, we limit the size of the dataset

to a total of 2500 training set images and 1000 test set images. We use images corresponding to 5 classes from the CIFAR-10 dataset: 'frog', 'automobile', 'bird, 'cat' and 'deer', with 500 training images and 200 test set images per class. Each image in the dataset has a 32x32 pixel RGB format.

4 Methods

4.1 Image Feature Extraction

Before encoding the images in our dataset using a BoVW representation, we first extract features from each image using two different techniques, SIFT and ORB.

4.1.1 SIFT

SIFT in its current form is used to detect image features that are invariant to scale and rotation. The SIFT algorithm follows a few steps: First, potential keypoints are located within the image then filtered to retain only the most stable keypoints. The remaining keypoints are evaluated by investigating surrounding pixels in order to determine whether the keypoint is actually relevant, i.e. if the keypoint has an intensity that meets a set threshold and if it does not lie on an edge. As the next step, any retained keypoints are made rotation invariant by giving them a 'direction'. Finally each keypoint is given a corresponding feature, by checking the brightness of the area around it. The process results in a rich and varied feature set.

4.1.2 ORB

ORB as a feature extraction method is a composition of two earlier methods: Features from Accelerated Segment Test (FAST) [8] and Binary Robust Independent Elementary Features (BRIEF) [9]. FAST is an effective and quick feature detector, which uses simple changes in surrounding pixel intensities to detect corners or keypoints, simply by thresholding a certain number of pixels around the candidate pixel. BRIEF, on the other hand, similar to the last step in 4.1.1, encodes the located keypoint in an effective manner. To make the resulting keypoints rotation invariant, ORB adds a direction to each keypoint, again similar to the method in 4.1.1. The end result is a high-valued feature set, with more features that are close in proximity to one another than 4.1.1 generates.

4.2 BoVW

BoVW is centered around the idea of Naive Bayes [10]. Naive Bayes models assume conditional independence between data points, in this case features, to make modeling easier. This is often not the case, especially for an image, but the results can be surprisingly good. In our case with the BoVW model,

this idea is implemented by using the features we extract naively. It works as follows: Firstly, we extract features with the aforementioned methods 4.1.1 and 4.1.2. These features are then used to create a visual vocabulary. This visual vocabulary is created by clustering the features extracted from a subset of images in the training set. This creates common feature categories, instead of treating the features in each image as being independent from those in other images. With these feature categories, we can describe any image, which allows us to classify new images if their description falls close to the center of any of the generated clusters. In our research, the clustering of the features is done using Kmeans, a simple algorithm which excels at performing clustering in a computationally cheap manner.

In our preliminary testing, we attempted the clustering with three different proportions of the training data: 30%, 40% and 50% in order to determine whether using a smaller amount of the training data for clustering and reserving the rest for training the classifiers would still result in an effective visual vocabulary. We decided to experiment with different training data proportions, as it may not necessarily be the case that more data creates better clustering, i.e. it is possible more data may create more overlap between the clusters, which could make it difficult to distinguish these clusters from one another, thereby leading to less clear-cut demarcations. Before tuning our classifiers, we defaulted to using 50% of the training data to create 1000 clusters, as this proportion gave us the best initial clustering results as judged from plots of the 10 largest clusters. Finally, the trained Kmeans algorithm was used to predict the clusters that all other images fall in. We then generated histograms for the remaining proportion of the training set not used for clustering as well as for the test set images, with each histogram encoding the frequency in each image's feature set of each of the 1000 'words' in our visual vocabulary. These image histograms correspond to the BoVW representation of the images.

The image histograms of the training set were then used to train simple one-versus-the-rest binary classifiers for the five classes using a support vector machine (SVM) approach. For each of the two feature extraction methods, SIFT and ORB, a separate set of binary classifiers was trained on the same images, with the initial hyperparameters being kept the same for the two sets of classifiers.

4.3 Support Vector Machine (SVM)

As the CIFAR-10 dataset [7] is a multi-class single-label dataset, we need to choose our classification model appropriately. The model of choice for our research is an SVM, which in its most basic form is a linear classification model. To account for the possibility that the images in our training set may not be linearly separable, we also make use of non-linear kernels, which can map the data to a new (possibly infinite-dimensional) space where the data becomes linearly separable due to the non-linearity in the kernel functions.

For our multi-class classification task we use a one-vs-the-rest strategy to assign images to the five classes. This involves training a class-specific model

for each of the classes in the dataset: images that are class members get a positive label while images belonging to all other classes get a negative label. In this way, the classifiers effectively function as five separate binary classifiers, one for each of our five classes. To obtain a single label prediction for an image, the predictions of the class-specific models are aggregated. For an optimal aggregation we want to know the probability associated with each predicted label. However, this is not something that SVMs support natively, as they are non-probabilistic classifiers. Many implementations of the SVM overcome this limitation by estimating the probabilities. This can be done by fitting a logistic regression model to the SVM output scores, and these are then calibrated using Platt scaling [11]. To aggregate the class-specific probabilities, the softmax function can be used to get a normalized probability for each class label such that the probabilities across all classes sum to one.

4.4 Evaluation

In order to be able to compare the performance of our image classifiers when using SIFT versus when using ORB feature extraction, we use both quantitative and qualitative classifier performance metrics, as outlined below.

4.4.1 Mean Average Precision (mAP)

To quantitatively evaluate the performance of the classifiers used in our research, we measure the mean average precision (mAP) achieved by each classifier on the test set. Because we use a one-versus-the-rest approach for our multi-class classification task, effectively creating a separate binary classifier for each class, we first measure a mAP score for each of the five classes based on whether the training image features were extracted with SIFT or ORB. Then, for each feature extraction method, we take an average of the mAP scores across all five classes.

4.4.2 Qualitative Evaluation

In addition to evaluating the performance of our classifiers using mAP scores, we also take a qualitative approach by creating a ranking of all 1000 test set images based on the class probabilities predicted by the different binary classifiers: an image with a high predicted probability of belonging to a certain class is ranked highly for that class, and images with the lowest predicted probability of belonging to a class end up at the bottom of the ranking for that class. We then visualize the five images ranked highest for each class as well as the five images ranked lowest. That is, for a good classification, we would expect the top five images for each class to be mostly images whose ground-truth label corresponds to that class and we would expect that very few or none of the lowest ranked images have a ground-truth label corresponding to that class.

4.5 Hyperparameter Tuning

For our research, we tune the following hyperparameters in order to find the set of hyperparameters that results in the highest mAP for each feature extraction technique:

- The number of features/keypoints to extract from each image: For our baseline model, we extract 100 keypoints from each image. During our hyperparameter search, we also consider values of 50, 200, and 500.
- The proportion of the training data to use for clustering (to create the visual vocabulary) versus for training the classifiers: the baseline model uses a split of 0.5/0.5 (i.e. 50% for clustering and 50% for training). During the hyperparameter search we also test clustering/training splits of 0.4/0.6, 0.3/0.7, and 0.2/0.8.
- The size of the visual vocabulary: the baseline model encodes image features in histograms using a vocabulary consisting of 1000 visual words (i.e. from 1000 clusters). For our hyperparameter tuning, we also consider visual vocabulary sizes of 250, 500, and 2000.
- SVM kernel type: our baseline model uses the sklearn SVM implementation with a linear kernel. For our hyperparameter tuning, we also test the non-linear 'rbf', polynomial and sigmoid kernels.
- SVM regularization parameter C: for our baseline model, we use the default sklearn value of 1.0 with a linear kernel. For our hyperparameter tuning, we test values of 0.1, 1.0, and 10 in combination with non-linear kernels.
- SVM kernel coefficient 'gamma' for non-linear kernels: During our hyperparameter search, we use values of 'scale', 0.001, and 0.01 for the gamma parameter of the sklearn SVM implementation.

Due to computational and time constraints, we did not make use of every possible combination of the above hyperparameters – methods such as grid search proved impractical for our research due to the sheer number of all the possible combinations of the aforementioned hyperparamaeters. Rather, we tuned a limited set of hyperparameters in two stages:

1. First, we used an SVM with a linear kernel and regularization parameter 1.0 for each of the two classifier types (SIFT versus ORB features). Without changing the SVM settings, we then searched the hypeparameter space of number of keypoints, visual vocabulary size, and clustering/training data split using random search. We generated 50 random samples from all the possible combinations of the three hyperparameters in this space. We then trained the SVMs for each feature extraction technique using these 50 hyperparameter samples, evaluating the classifiers' performance (mAP score) on the test set for each sample. Although this sampling of

the hyperparameter space may not find the optimal hyperparameters, we decided on this approach as it allows us to search for hyperparameters in a more efficient manner and can theoretically find a set of hyperparameters that results in good performance, even if this is not the optimal. For SIFT features, the hyperparameter set that was found to yield the highest mAP was 200 keypoints, a clustering/training data split of 0.3/0.7, and a visual vocabulary size of 2000. For ORB features, this was 100 keypoints, 0.3/0.7 data split, and vocabulary size of 500.

2. Next, we took the best set of values found in the first stage of tuning for the number of keypoints, visual vocabulary size, and clustering/training data split for each feature extraction method and used these to tune the SVM hyperparameters. Because we used a linear kernel in the first stage of tuning, in the interest of efficiency, we considered only the three non-linear kernels for our second hyperparameter search, together with the regularization and gamma parameters. We again used random search to sample the SVM hyperparameter space, using the same set of 25 samples for each of the best hyperparameter sets found for SIFT and ORB features respectively from the first stage of tuning. Again, the mAP was measured, and the hyperparameters yielding the best mAP score for each feature extraction method were taken to be the best for that method.

To compare classifier performance when using features generated with SIFT versus ORB, for each methods we took the hyperparameters that were found to yield the best mAP from either of the two hyperparameter tuning stages. For SIFT features, the highest mAP was obtained using 200 keypoints, a clustering/training data split of 0.3/0.7, and a visual vocabulary size of 2000 together with an SVM using an 'rbf' kernel, a 'C' value of 0.1, and a 'gamma' value of 'scale'. For ORB features, the best hyperparameters were: 100 keypoints, 0.3/0.7 data split, and vocabulary size of 500, together with an SVM using a 'linear' kernel, and a 'C' value of 1.0. The mAP scores obtained on the test set with these hyperparameters are presented in Table 1.

5 Results

5.1 Quantitative Results

Model	**mAP Score**
Baseline SVM using SIFT features	0.45
Baseline SVM using ORB features	0.32
Tuned SVM using SIFT features	0.47
Tuned SVM using ORB features	0.36

Table 1: Model performance before and after hyperparameter tuning

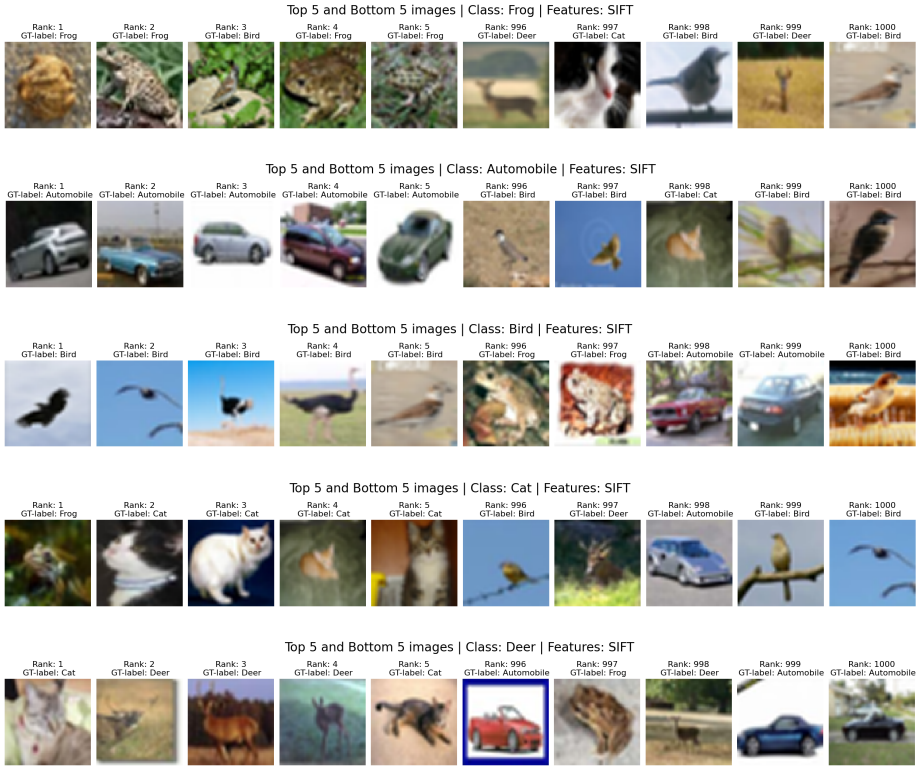

Figure 1: The Top-5 and Bottom-5 ranked test set images (using SIFT features) for each class after hyperparameter tuning

The baseline classifiers trained on the BoVW representations of SIFT features outperforms those using ORB features, with a difference in mAP scores of 0.13.

After tuning the hyperparameters for each feature extraction method, we observe an increase in the mAP scores for both methods: when using SIFT features, a tuned model achieves a 0.47 mAP compared to a baseline model, and for ORB, the tuned model achieves a 0.36 mAP compared to 0.32 for the baseline. There is a difference of 0.11 in the mAP obtained when using SIFT features (0.47) compared to when using ORB features (0.36). Although SIFT features still outperform ORB features for the tuned models, the 0.04 increase in mAP for ORB relative to the baseline model represents a 12.5% improvement in performance versus a much smaller 4.4% increase for SIFT.

5.2 Qualitative Results

Inspecting the top five and bottom five ranked images for each class, for both SIFT (Figure 1) and ORB (Figure 2) features, we observe that when using

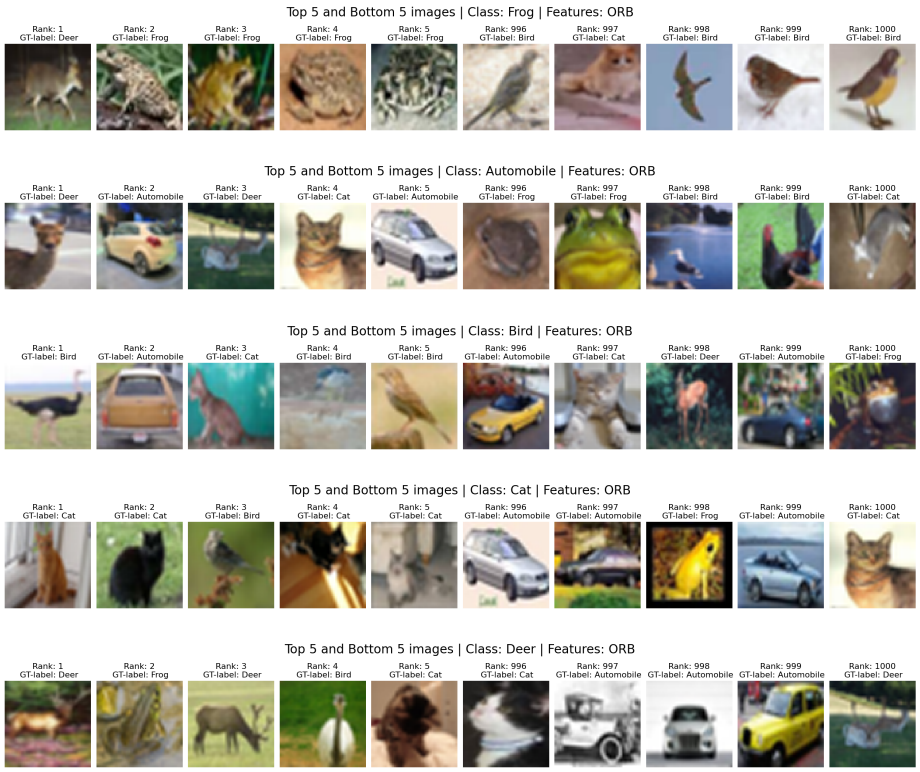

Figure 2: The Top-5 and Bottom-5 ranked test set images (using ORB features) for each class after hyperparameter tuning

SIFT, the rankings for the top five images of each class are better than when using ORB. For every class, using SIFT results in at least three images with a ground-truth (GT) label corresponding to that class appearing in the class's top five ranked images. For two of the classes, 'bird' and 'automobile', all top five images are ones that actually belong to that class. On the other hand, when using ORB features, two of the classes ('automobile' and 'deer') have only two correctly classified images in each of their top five rankings. These results suggest that the classifiers are better at identifying true positives for each class when image features are extracted using SIFT.

However, looking at the bottom five rankings for each class, both feature extraction methods result in an incorrect ranking for two classes: when using SIFT, the 'bird' class has an image of a bird in its bottom five ranking, and the 'deer' class has an image of a deer in its bottom five ranking. For ORB, the 'cat' and 'deer' classes each have an image belonging to that class incorrectly appearing in the bottom five ranking. Therefore, although using SIFT features may be better for identifying true positives, there seems to be no difference for the two feature extraction methods in the rate of false negatives.

6 Discussion

Although we found that our image classifiers performed very poorly, with mean average precision (mAP) scores of less than 0.5, this should not come as a surprise, given that we are using a Bag of Visual Words (BoVW) representation of image features, which effectively treats the features in an image as being unconnected to one another. However, an object in an image, such as the 'cats', 'birds', 'deer', etc. that we tried to classify with our research is not made up of independent features in the image but rather of connected components. Thus, by treating features as independent, we are ignoring contextual/spatial relationships that are important for determining what object is depicted in an image, and thereby losing some information that is necessary for accurately classifying the image. Our research suggests that BoVW is not an effective technique for representing images, or at least not for classification purposes.

Moreover, the low mAP scores we found probably also overstimate the performance of the classifiers we used: Because we used the test set instead of a separate validation set for tuning hyperparameters, we effectively tuned our classifiers to achieve good performance on the test set. Therefore, the final evaluation of these classifiers, conducted on the same test set, probably overestimates their performance, and we would likely find performance on never-encountered images (i.e a 'true' test set) to be much lower than what we observed. This would mean that a BoVW representation of image features results in even poorer image classification than what is indicated by our research.

Comparing the two feature extraction techniques treated in this paper, ORB, although developed as a more efficient alternative to methods such as SIFT, seems to result in poorer image classification: The 0.11 difference in mAP scores between the two methods is large, suggesting that using SIFT feature extraction results in better image classification for a BoVW image representation than using ORB. However, this could also be because we did not find good SVM settings for ORB features, i.e. even though we found that a linear kernel yielded the best mAP for ORB features, we did not tune the regularization parameter for this kernel. It is possible that tuning this may result in a mAP score for ORB features that is comparable to or even better than that of SIFT features.

Also, our limited search of the hyperparameter space may have missed the optimal parameters or even several other sets of hyperparameters that could have resulted in better classifier performance than the ones we found to be the best. In future research, we would like to remedy this by conducting an exhaustive grid search of all possible hyperparameter combinations in order to find the ones that are optimal.

Another limitation that likely negatively influenced our research results is the use of a very small dataset with only 500 training images per class. Because machine learning models tend to perform better with more training data, it is possible that with a much larger image dataset, we would find that, contrary to what our research indicates, BoVW is actually an effective method for representing image features such that we can achieve acceptable image classification performance even when using simple models such as SVMs.

References

[1] Chih-Fong Tsai. Bag-of-words representation in image annotation: A review. *ISRN Artificial Intelligence*, 2012, 11 2012.

[2] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[3] Michael J Swain and Dana H Ballard. Color indexing. *International journal of computer vision*, 7(1):11–32, 1991.

[4] Nikolaos Kanopoulos, Niranjan Vasanthavada, and Robert L Baker. Design of an edge detection filter using the sobel operator. *IEEE Journal of Solid-State Circuits*, 23(2):358–367, 1988.

[5] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[6] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *2011 International Conference on Computer Vision*, pages 2564–2571. IEEE, 2011.

[7] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, Technical report, University of Toronto, 2009.

[8] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *European conference on computer vision*, pages 430–443. Springer, 2006.

[9] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. Brief: Binary robust independent elementary features. In *European conference on computer vision*, pages 778–792. Springer, 2010.

[10] Pedro Domingos and Michael Pazzani. On the optimality of the simple bayesian classifier under zero-one loss. In *Machine Learning*, pages 103–130, 1997.

[11] John Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in large margin classifiers*, pages 61–74. MIT Press, 1999.