COMP61342-Computer Vision Coursework

Introduction

This work implements two approaches for the object recognition task, bag of visual words (BoVW) and convolutional neural network (CNN). The baseline dataset for this work is CIFAR-10, 60,000 RGB images in 10 classes. One of the aims of this work is to explore different configurations of the two approaches and how they impact the performance. Another aim is to compare and contrast the two approaches with their performances on the baseline dataset.

Bag of Visual Words

Method

The bag of visual words method can be defined as a pipeline of operations, which are feature extraction, image representation, and classification. The workflow is as follows.

1. The input images are pre-processed to be acceptable for the feature extractor.
2. In the feature extraction step, the feature extractor takes the processed images as input and generates interest point descriptors. The descriptors are recorded in correspondence with images.
3. In the image representation step, the clustering algorithm is fit into all the descriptors and generates their labels. The key-value pairs of descriptors and labels form the visual dictionary. Then for each image, the dictionary is looked up according to the occurred descriptors that a list of labels could be found. The list of labels could be aggregated into a histogram according to their frequencies.
4. In the classification step, histograms of images are fit into a classification model that can predict classes of new images according to their histograms calculated as in the former step.

This pipeline is a process of discovering image features and encoding them into a form that is compatible with a classification algorithm. Step 2 is the process of algorithmically discovering image features. Step 3 is the process aiming at encoding features of images into histograms that are classifiable. Therefore, the performances of these intermediate steps contribute to the effectiveness of classification.

Comparison of Sub-methods

Feature Extraction

Two similar methods of feature extraction are implemented: SIFT and ORB [1]. They realize feature extraction in different ways of key point detecting and defining key point descriptors.

According to the study [1], in terms of efficiency, ORB is computationally faster than SIFT and could detect more features from images. As for accuracy, ORB is more invariant to affine changes but less robust to rotations. In general, SIFT points are more accurate compared to ORB points by the image matching experiment in the study.

The SIFT algorithm could be more adapted to this work, the reasons are as follows. As images of the baseline dataset are of low resolution (32*32), the flexibility on blurred images of SIFT can be demonstrated, while the ORB algorithm may require hyperparameter tuning to adjust to the resolution. Besides, as SIFT is more robust on rotation, its detected points may be more accurate, which means less noise can be generated for the following clustering algorithm.

Image Representation

K-Means and Gaussian Mixture (GM) [2] are applied as the clustering algorithms of image representation. K-Means tends to produce spherical clusters with similar sizes. It has the advantage that it is computationally effective on large data and can configure with a relatively larger number of clusters. However, its result is dependent on the initialization so more iterations should be run to achieve higher scores. Another disadvantage is that if the ideal clusters are not spherical or of similar size, the clustered result may not be as expected. Besides, it suffers from the curse of dimensionality that distances of points tend to be close if the input has a high dimension. As for Gaussian Mixture, several numbers of Gaussian distributions are fitted into the data that each point has probabilities of belonging to these Gaussian distributions. It has the advantage compared to K-Means that the clusters are shaped as a gaussian distribution rather than spherical, which could have more generalization. And it probabilistically assigns points to clusters rather than hard assignments.

In this work, there are two features of the clustering task. The first is that the number of points to be clustered is large and points are with high dimension. The second is that the number of clusters should be configured as a relatively large number compared to usual cases because in this way more information could be forwarded to the classification task. With such requirements, the K-Means algorithm could be computationally efficient but may produce unrealistic clusters and is not suitable for the high-dimensional data. The Gaussian Mixture algorithm may produce more realistic clusters, but it is more complex in computing with a large number of clusters. Both the two algorithms may produce biases due to the complexity of the task and the models are difficult to be interpreted due to the large dimensions of data.

Classification

Support Vector Machine (SVM) with Radial Basis Function (RBF) kernel [3], K-nearest Neighbors (KNN), and Random Forest (RF) are implemented as classifiers. For SVM, the selection of kernel could enable it to solve complicated problems. Compared to other algorithms, it is more effective for fitting high-dimensional data. And it is capable to generalize to risk less overfitting issues. However, this algorithm takes time to fit into the big dataset and it is difficult to interpret and adjust the generated model. As for KNN, it is a relatively trivial algorithm that can directly predict without the training period. However, as it is dependent on the calculation of distance, it is not robust on high-dimensional data and outliers.

In addition, it is not computationally efficient for big data and high-dimensional data. For RF, as it is an ensemble of many decision trees, the risk of overfitting could be reduced. Besides, it is robust on outliers as decision trees can exclude them automatically during learning. But the trade-off is that this algorithm may take a long time and cost considerable computational power to train a model.

The feature of the classification task is that the dataset is big and high-dimensional (50,000 histograms) where there exist outliers produced by the feature extractors. For such a dataset, the SVM algorithm could be competent for achieving a relatively high result because it is suitable for high-dimensional data. But fitting into the big data may be time-consuming. The KNN algorithm may not be ideal for solving this classification task because it is not robust on data with high dimensions. The RF algorithm could be suitable for this task because the decision trees can interpret high dimensions and it is robust on data with outliers.

Evaluation of Sub-methods

Feature Extraction	Image Representation	Classification	Accuracy	Weighted F1 Score
SIFT	K-Means	Random Forest	0.269	0.264
SIFT	K-Means	K-nearest Neighbors	0.229	0.216
SIFT	K-Means	RBF-SVM	0.292	0.288
SIFT	Gaussian Mixture	Random Forest	0.257	0.252
ORB	K-Means	Random Forest	0.198	0.193

Table 1 Accuracies and weighted F1 scores of BoVW models with different configures. The numbers of clusters in image representation are configured as 300.

Feature Extraction

The two feature extraction methods, SIFT and ORB, are evaluated that K-Means and Random Forest are implemented for clustering and classification. The parameters are tuned aiming at raising the final classification accuracy, tuned parameters are recorded in the Appendix.

Key points detected by the two algorithms of 3 pairs of sample images are shown in Figure 1, which demonstrates that ORB generally produces more key points than SIFT. According to Table 1, the model with ORB acquires an accuracy of 0.198 while the model with SIFT scores 0.269. This may indicate that both two methods have limitations. The possibilities of causes are manifold. From the perspective of the clustering algorithm, the feature extractor could extract false features that become noisy instances; it could extract too many features that result in overfitting; it may miss extracting some image features. Despite the limitations, the statistic demonstrates that SIFT could extract more image information for the BoVW model compared to ORB. It may indicate that despite ORB could generate more key points, more proportion of them may be noisy instances.

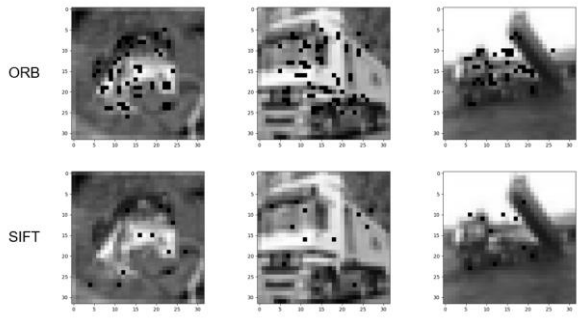

Figure 1 Key point positions of ORB and SIFT on 3 sample pairs of images, denoted as black pixels.

Image Representation

Two clustering methods, K-Means and Gaussian Mixture, are evaluated that feature extractor and classification are uniformly configured as SIFT and Random Forest, respectively. The number of clusters is an important hyper-parameter in this step. Tuning the number of clusters should consider the trade-off between the complexity of the clustering task and the information forwarded to the classification task. If the number of clusters is large, the clustering algorithm is more difficult to fit into the data, but more information could be retained and forwarded to the classification step. The final number of clusters is determined as 300 according to the criterion of raising the final accuracy score.

According to Table 1, the model with K-Means acquires an accuracy of 0.269 while the model with Gaussian Mixture acquires an accuracy of 0.257. Both the two methods may generate biases because the high-dimensional data results in a more centralized distribution of data point distances hence indistinguishable clusters. Besides, the large number of clusters increases the difficulty of finding the global optimal solution and computational complexity, especially for Gaussian Mixture.

Classification

Three classification methods, Random Forest (RF), K-nearest Neighbors (KNN), and SVM with RBF kernels, are evaluated that feature extractor and image representation are uniformly configured as SIFT and K-Means, respectively.

According to Table 1, the model with SVM acquires the highest accuracy among the three, 0.292, followed by RF, which is 0.269. The model with KNN comes last scoring 0.229. This result confirms the former analysis of the three methods. As for SVM, the compatibility with high-dimensional data and the kernel trick could contribute to its relatively high performance. The robustness of outliers and overfitting of RF could take effect. The incompatibility of big data and high-dimensional data could account for the relatively low score of KNN.

Convolutional Neural Network

Architectural Configurations

As for dataset pre-processing, the training set is altered that a proportion of images are horizontally flipped or slightly shifted, known as data augmentation. This operation could bring more challenges for the training but could allow the model to learn more general features from the images.

The process of designing the architecture of the CNN is separated into two steps. The first step is to build a model that is complex enough to ensure that it has high accuracy on the training set regardless of the overfitting problem. Then in the second step, various methods are applied for mitigating the overfitting problem to raise the accuracy of the test set for approaching the accuracy of the training set. Several CNN models are implemented for the exploration of hyper-parameter settings and model configurations.

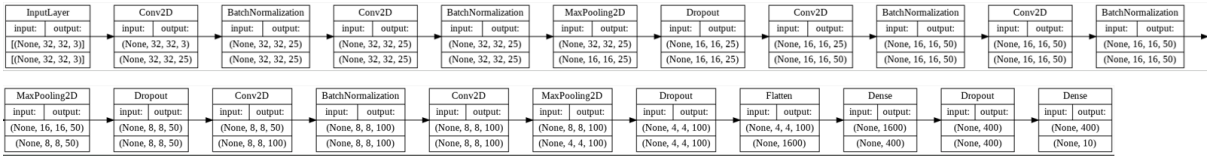

Figure 2 Network architecture of Net3.

For the first step, a baseline model is defined for fitting the training set. The model consists of 3 consecutive sets of two convolutional layers and one max pooling layer (hereafter referred to as convolutional component) with RELU activation connected by a dense layer with Hyperbolic tangent activation and an output dense layer with Softmax activation. CNN nodes are initialized with truncated normal distribution for faster convergence. Adam is used as the optimizer. The number of convolutional components is determined by selecting the one that acquires the highest score from models with 1, 2, and 3 convolutional components. As for the arrangement of CNN nodes in each layer, to retain information in deeper layers along with max pooling, the numbers of CNN nodes are doubled in succeeding CNN layers. The number of nodes in the first CNN layer is tuned that the model acquires a 0.912 accuracy score on the training set. Hence the goal of the first step is satisfied. The next step is to raise the accuracy on the test set.

In the second step, to mitigate the overfitting problem, several alterations to the baseline model are supplemented. In the second model, batch normalization and L2 regularization are supplemented. Batch normalizations are added between adjacent convolutional layers for them to learn more independently. In other words, the latter layers are less influenced by the former layers. L2 regularizations are added to each of the convolutional layers that can reduce the impact of smaller weights. This could improve the model as small weights may be redundant thus could be a factor for overfitting.

In the third model shown in Figure 2, dropout layers are added after each of the convolutional components for mitigating the overfitting problem. This alternation adds difficulty to the training but can effectively reduce the influence of former epochs of learning on the general learning. The goal of tuning the dropout rate is to raise the test score and try to sacrifice less accuracy on the training set.

In this model, the dropout rate is set as 0.2 for the first convolutional component and incremented with 0.05 with succeeding network components. This aggregative pattern can be effective as the overfitting problem may influence more in deeper layers.

Evaluation

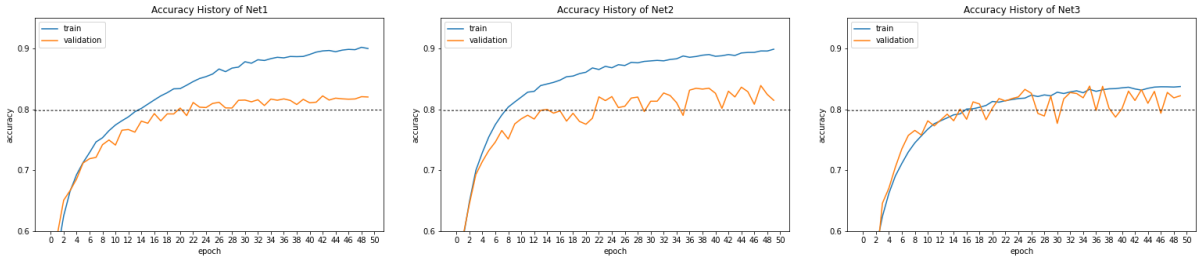

Figure 3 Training accuracy history of the baseline model (Net1), the model with batch normalization and L2 regularization (Net2), and the model with batch normalization, L2 regularization and dropout (Net3).

	Net1	Net2	Net3
Test Accuracy	0.821	0.829	0.834

Table 2 Test accuracy of Net1, Net2, and Net3.

Figure 3 and Table 2 demonstrate the training and test accuracy of the three models, respectively. In the training history of Net1, there is an obvious gap between the training accuracy and validation accuracy caused by overfitting. In the training history of Net2, the curve of validation accuracy is less stable. The reason may be that the model trained with reshaped data could perform differently on the validation data of its original form. It is noticeable that in a proportion of the epochs, this model performs better compared to Net1 with the validation set. The difference in test accuracies of Net1 and Net2 in Table 2 demonstrates that batch normalization and L2 regularization generally benefit the generalization of the model. As for Net3, the training curve noticeably scores less than the other two models due to the challenges brought by dropout. But the validation curve is closer to the training curve, which indicates that this model is to a lesser extent of overfitting compared to the other two. According to Table 2, it is slightly more efficient on the test set compared to Net2.

In conclusion, batch normalization and L2 regularization can increase the generalization of the deep network. Dropout can also increase the generalization with the sacrifice of converge speed of training or training accuracy. If the dropout rate is selected properly, the test accuracy could be increased. For future simulations, the model could be designed with more CNN nodes or CNN layers for increasing the model complexity. Higher dropout rates could be tested on longer epochs of runs or more approaches of mitigating overfitting could be adopted to exploit the limit of this network structure and achieve higher performance.

Comparison of the Two Approaches

In terms of architecture, BoVW is a pipeline of algorithmic approaches, while the CNN is a series of convolutional layers and dense layers. The BoVW method requires programmers to define and connect sub-approaches of extracting information from images and encoding the information into a classifiable

form. Due to the inconsistency of sub-approaches, there may exist information loss. For the CNN, the input is the pixel values of images so that there is no encoding operation required. Nodes in the entire network are connected so that no manual connection or transfer of data shape is required.

In terms of performance, the accuracy of BoVW is significantly lower than the CNN according to the results. The reasons may be manifold. First, the BoVW method must convert RGB images to grayscale as the requirement of feature extractors, which leads to information loss. Secondly, as there are different kinds of sub-approaches in BoVW, the data shape generated by the former approach may be relatively less acceptable for the succeeding one even though it is compatible. For example, the dimension of descriptors generated by the SIFT feature extractor is 128. It is challenging for most of the clustering algorithms to be well-fitted. This increases the possibility of generating biases in the clustering step. Thirdly, the feature extraction is more comprehensive in the CNN compared to BoVW. In BoVW, the feature extraction is executed algorithmically that may have some flaws. For example, false features are extracted or true features are not extracted. Whereas in the CNN, the feature extraction is executed by convolutional layers that can comprehensively and flexibly fit into the training images. Besides, the complexity of feature extraction can be more trivially configured by altering the number of convolutional layers or the number of nodes in these layers.

In terms of training speed, the CNN is more efficient in practice as training networks can utilize GPUs while the BoVW meets challenges in clustering visual words as stated above.

For application, CNN can be deployed on mobile devices locally or through cloud computing for real-time object recognition, image tagging, and visual searching [4]. A state-of-the-art approach [5] is proposed based on BoVW to interpret and classify images of athletic sports.

References

1. Tareen, S.A.K. and Z. Saleem, A comparative analysis of SIFT, SURF, KAZE, AKAZE, ORB, and BRISK. 2018.
2. Maugis, C., G. Celeux, and M.L. Martin‐Magniette, Variable selection for clustering with Gaussian mixture models. Biometrics, 2009. **65**(3): p. 701-709.
3. Han, S., C. Qubo, and H. Meng. Parameter selection in SVM with RBF kernel function. in World Automation Congress 2012. 2012. IEEE.
4. Hsu, C.-C., et al. Cooperative convolutional neural network deployment over mobile networks. in ICC 2020-2020 IEEE International Conference on Communications (ICC). 2020. IEEE.
5. Kesorn, K. and S. Poslad, An enhanced bag-of-visual word vector space model to represent visual content in athletics images. IEEE Transactions on Multimedia, 2011. **14**(1): p. 211-222.

Appendix

SIFT Parameters

The SIFT algorithm is implemented with opencv, unmentioned parameters are set as default. Parameters are as follows.

contrastThreshold	sigma	edgeThreshold	nOctaveLayers
0.05	1.6	10	4

Table 3 SIFT parameters.

ORB Parameters

The ORB algorithm is implemented with opencv, unmentioned parameters are set as default. Parameters are as follows.

edgeThreshold	patchSize	scaleFactor
5	5	1.1

Table 4 ORB parameters.

Confusion Matrix of BoVW Models

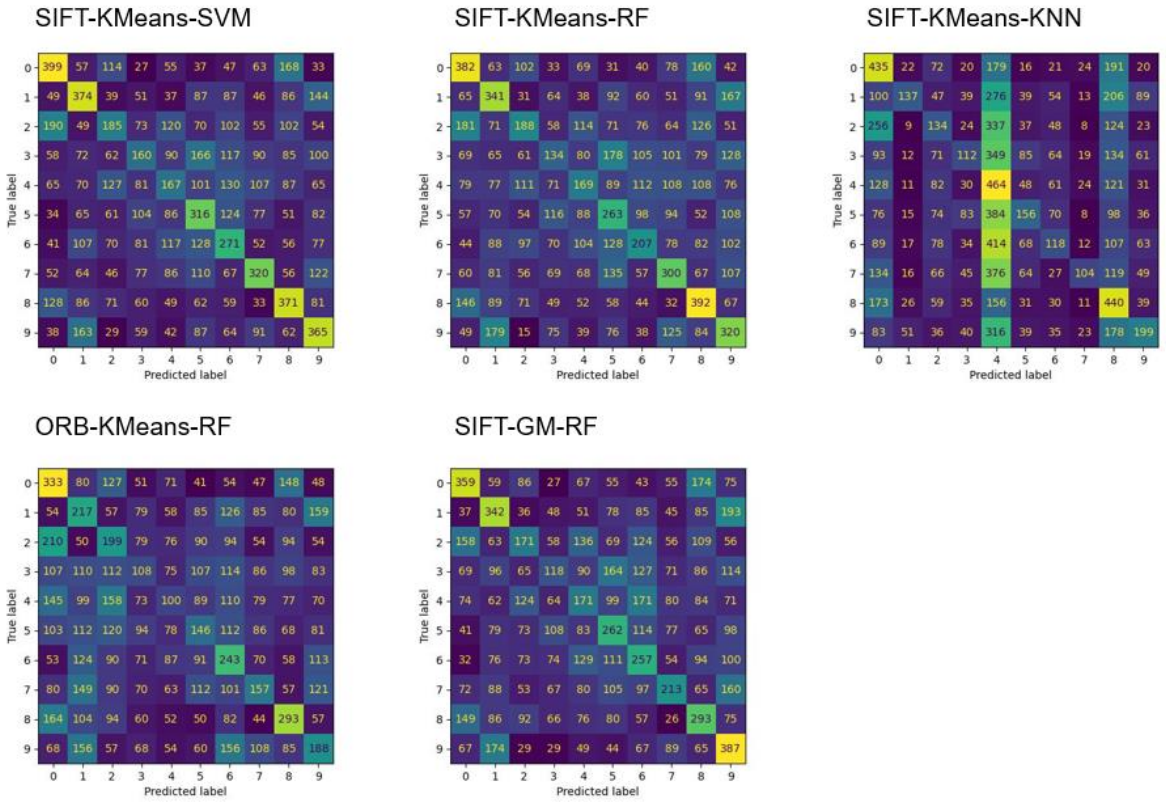

Figure 4 Confusion matrix of BoVW models of different selections of sub-methods.