Principles of Data Science Project 3
Feature Encoding

Junhao Dai daijunhao@sjtu.edu.cn

May 10, 2024

1 Introduction

1.1 Experimental Configuration

The Python version used for this experiment is 3.11.5, with the following library versions:

- NumPy version: 1.26.2
- Pandas version: 2.0.3
- Scikit-learn version: 1.3.0
- pytorch version: 2.3.0+cu118
- cv2 version: 4.9.0

Additionally, part of the programs were run on a laptop with a CPU model AMD Ryzen 7 5800H with Radeon Graphics, operating at 3.20 GHz, and the other part were run on Siyuan-1 supercomputer with Intel Xeon ICX Platinum 8358 as cpu and NVIDIA HGX A100 as GPU.

1.2 My Work

In this project, we experimented with various feature extraction and feature encoding methods to accomplish an SVM-based image classification task. First, we used SIFT to extract local descriptors from images and then applied different feature encoding methods such as Bag of Words (BOW), VLAD, and Fisher Vector (FV). The resulting features were used to train an SVM model for image classification on the test set. Additionally, we used selective search to extract proposals of images and applied ResNet to extract local descriptors, repeating the same process. We evaluated the performance of each approach by measuring the final classification accuracy.

2 Experiments and Results

In this section we first process all 37,322 images using SIFT to extract to get the local descriptors, followed by random shuffling to disrupt the order and the resulting dataset will be used for feature coding. After this we preprocess the dataset by random sampling as well as feature dimensionality reduction, which can effectively shorten the training time and reduce the arithmetic demand, based on which different feature encoding methods are used. Finally we try to repeat the above process using selective search and ResNet instead of SIFT to extract local descriptors.

2.1 SIFT

Scale-invariant feature transform (SIFT) is a feature detection and description algorithm used in the fields of computer vision and image processing [3]. It was introduced by David Lowe in 1999 and further developed in 2004. The primary characteristics of the SIFT algorithm are its scale invariance and robustness to changes in illumination, noise, and minor changes in viewpoint. The SIFT algorithm consists of several steps:

Scale-Space Extreme Detection: SIFT first detects keypoints across different scales, which are implemented through the Difference of Gaussians (DoG). The DoG captures local features at various scales within the image.

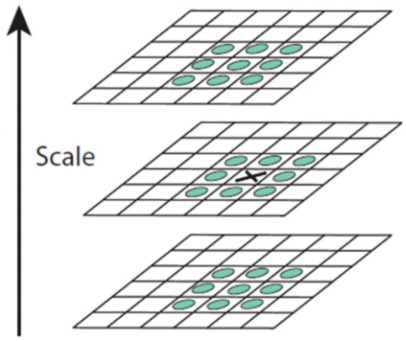


Fig 1: Extreme Detection.

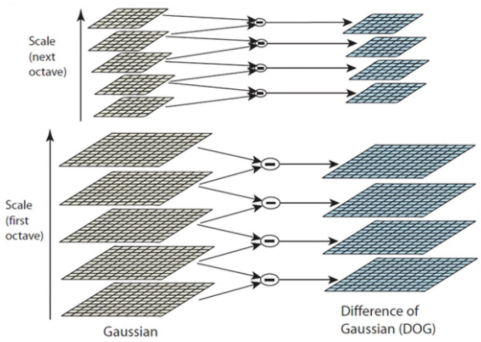


Fig 2: Difference of Gaussians.

Keypoint Localization: Once keypoints are detected, SIFT uses a Taylor series expansion to accurately locate these keypoints' positions and scales.

Orientation Assignment: To make the descriptor rotation-invariant, SIFT assigns one or more principal orientations to each keypoint, typically determined by the gradient directions in the local image surrounding the keypoint.

Keypoint Descriptor: SIFT generates a descriptor for each keypoint, which is a histogram of gradient orientations in the local region surrounding the keypoint. This descriptor is invariant to changes in scale, rotation, and illumination of the image.

Matching: Finally, these keypoint descriptors can be used to match features between different images, facilitating tasks such as image recognition, object tracking, or 3D reconstruction.

2.1.1 Baseline

The Bag of Words (BOW) model is a method used for image feature encoding[4]. It treats the features in an image as words, and the image as a document composed of these words. The image is described by counting the frequency of each word in the image.

For this experiment, the baseline model is the test set image classification accuracy of an SVM model obtained by directly using SIFT to extract local descriptors from 37,322 images and using a KMeans model to create a visual vocabulary for BOW encoding. We set the range of the BOW

model hyperparameters k to be [8, 16, 32, 64, 128, 256, 512, 1024], and the range of the SVM model hyperparameters C to be [0.001, 0.03, 0.1, 0.3, 1, 3, 5], with the results shown in Table 1.

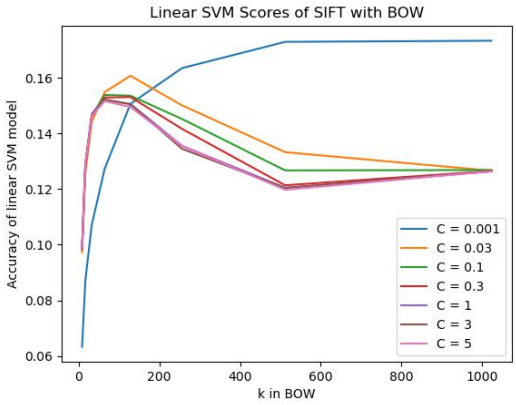


Fig 3: Results of Linear SVM with SIFT and BOW based on FULL set.

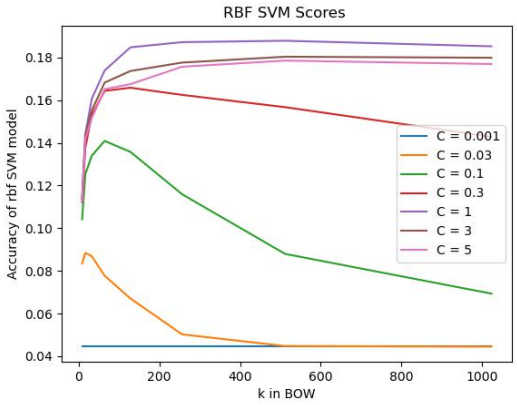


Fig 4: Results of RBF SVM with SIFT and BOW based on FULL set.

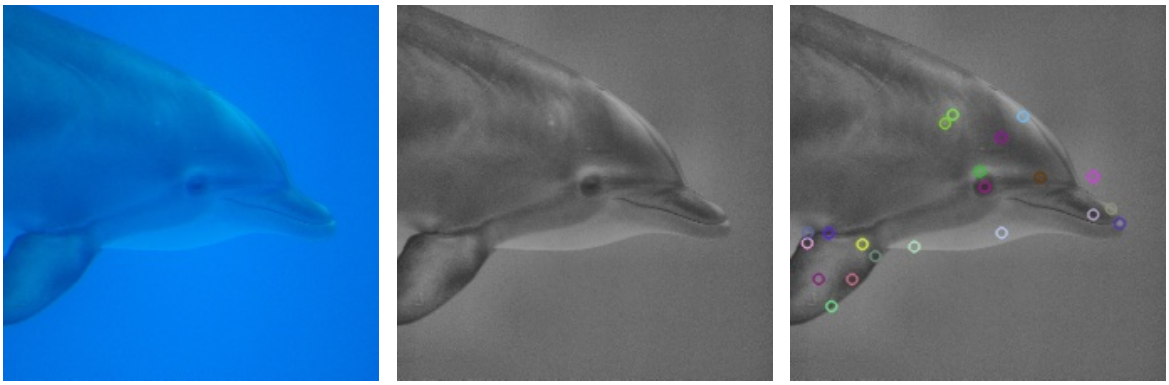
Table 1: Accuracy of SIFT feature based on BOW model with FULL size dataset

C	**k**	**linear**	**rbf**	**k**	**linear**	**rbf**	**k**	**linear**	**rbf**	**k**	**linear**	**rbf**
0.001		0.0634	0.0445		0.0870	0.0445		0.1072	0.0445		0.1273	0.0445
0.03		0.0973	0.0834		0.1255	0.0884		0.1442	0.0869		0.1549	0.0777
0.1		0.0998	0.1042		0.1289	0.1253		0.1465	0.1340		0.1539	0.1409
0.3	8	0.0992	0.1121	16	0.1291	0.1376	32	0.1461	0.1531	64	0.1529	0.1644
1		0.0983	0.1140		0.1283	0.1441		0.1472	0.1607		0.1521	0.1740
3		0.0983	0.1140		0.1282	0.1435		0.1464	0.1552		0.1522	0.1683
5		0.0987	0.1123		0.1286	0.1417		0.1462	0.1516		0.1517	0.1652
0.001		0.1506	0.0445		0.1635	0.0445		0.1730	0.0445		0.1734	0.0445
0.03		0.1608	0.0671		0.1501	0.0502		0.1333	0.0447		0.1266	0.0445
0.1		0.1536	0.1358		0.1452	0.1159		0.1267	0.0889		0.1269	0.0693
0.3	128	0.1531	0.1659	256	0.1417	0.1625	512	0.1214	0.1567	1024	0.1265	0.1429
1		0.1506	0.1848		0.1356	0.1872		0.1205	0.1879		0.1264	0.1853
3		0.1505	0.1737		0.1345	0.1776		0.1204	0.1804		0.1264	0.1799
5		0.1495	0.1675		0.1355	0.1757		0.1198	0.1786		0.1264	0.1770

The entire training process took approximately 17 hours to complete on the server. The data in the table shows the test set classification accuracy obtained using linear and rbf SVMs under different BOW k parameters and SVM model C parameters. For the linear model, the highest accuracy was achieved when k was 1024 and C was 0.001, at around 17.34%. For the rbf model, the highest accuracy was achieved when k was 512 and C was 1, at around 18.79%. Overall, the baseline accuracy is too low, and the results are depicted in Figure 3 and Figure 4.

During the calculation of the baseline, when we used SIFT to process the images, some images, due to their lack of clarity, could not extract valid features, such as ”dolphin_10180” and ”humpback+whale_10428”. For these images, we adopted an enhancement extraction method, converting

the images to grayscale and increasing the contrast, which allowed us to obtain valid results. The process demo is shown in Figure 5.



(a) origin image (b) enhanced image (c) demo image

Figure 5. The feature extraction results obtained for "dolphin_10180" after applying grayscale conversion and contrast enhancement.

2.1.2 Random Sampling BOW

To shorten the training time and attempt to improve the performance of the BOW, we employed a random sampling method. For all fifty categories, we selected 100 images per category, thus reducing the overall dataset size from 37,322 images to 5,000 images. It is important to note that the corresponding label files must also be selected accordingly. This way, we generated smaller local descriptor files for KMeans to create a visual vocabulary, resulting in the data shown in Table 2.

Table 2: Accuracy of SIFT feature based on BOW model with Random Sampled Dataset.

C	k	linear	rbf	k	linear	rbf	k	linear	rbf	k	linear	rbf
0.001		0.2145	0.2145		0.2145	0.2145		0.216	0.2145		0.227	0.2145
0.03		0.2275	0.2145		0.2375	0.2145		0.245	0.2145		0.244	0.2145
0.1		0.227	0.232		0.2265	0.232		0.2505	0.23		0.243	0.2275
0.3	8	0.228	0.2535	16	0.23	0.253	32	0.242	0.254	64	0.2385	0.25
1		0.226	0.263		0.232	0.2655		0.24	0.2705		0.2385	0.2725
3		0.227	0.25		0.23	0.25		0.238	0.2765		0.235	0.2655
5		0.2265	0.254		0.23	0.246		0.239	0.268		0.236	0.2605
0.001		0.237	0.2145		0.251	0.2145		0.2545	0.2145		0.2485	0.2145
0.03		0.2295	0.2145		0.2175	0.2145		0.211	0.2145		0.202	0.2145
0.1		0.227	0.2235		0.209	0.2145		0.2035	0.2145		0.2095	0.2145
0.3	128	0.2195	0.2435	256	0.213	0.2405	512	0.1915	0.237	1024	0.2095	0.2175
1		0.2255	0.268		0.215	0.274		0.1895	0.2745		0.209	0.2615
3		0.22	0.269		0.2135	0.266		0.1925	0.2545		0.209	0.2555
5		0.219	0.257		0.2145	0.263		0.1885	0.253		0.209	0.246

From the results table, we can observe that, compared to using the entire image dataset, the classification accuracy has significantly increased when using a randomly sampled smaller dataset,

and the training time has been greatly reduced. The speculated reason for this could be that after random sampling, the number of images for each category is more balanced, so the number of features for a particular category does not become overly prominent, which could lead to bias in the weights. The highest accuracy was achieved when using the rbf SVM model with k set to 8 and C set to 1. The overall results have been plotted in Figure 6.

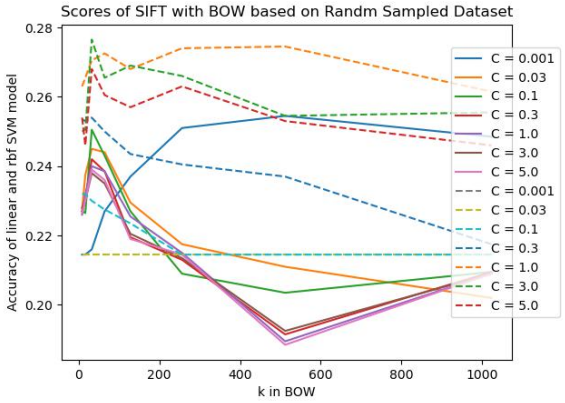


Fig 6: Results of SVM with SIFT and BOW based on Random Sampled Dataset.

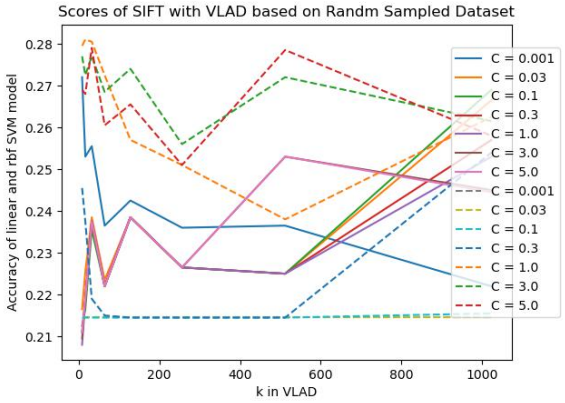


Fig 7: Results of SVM with SIFT and VLAD based on PCA LDs.

2.1.3 VLAD with PCA

VLAD is a method used for image feature encoding, primarily for image retrieval and classification tasks[2]. The main idea behind VLAD is to associate each local descriptor (such as SIFT or SURF) with its nearest visual vocabulary (obtained through clustering), and accumulate the residuals between each visual vocabulary and its associated descriptors. In this way, the VLAD encoding can capture more image information and has higher recognition accuracy than the BoW model.

During the experiment, we found that after completing the feature extraction, the process of training an SVM model to obtain test set classification accuracy was very slow, with each iteration taking approximately 1 to 3 hours. Therefore, we considered using PCA (Principal Component Analysis) to reduce the dimensionality of the extracted features, which could significantly shorten the overall computation time. The results obtained after using PCA to reduce the dimensionality of the features extracted by VLAD are shown in Table 3, and the corresponding results are also displayed in Figure 7.

When comparing the results of decomposing VLAD with those of BOW, we found that there was a significant improvement in accuracy for the linear SVM model, while the improvement for the rbf model was not substantial. Additionally, the performance of the linear model showed a noticeable improvement when k was larger. The highest accuracy achieved was 27.95%.

2.1.4 FV with PCA

FV (Fisher Vector) is a statistical model used for image feature representation in computer vision, based on the concept of Fisher information divergence. In image classification, after generating a vocabulary histogram, FV trains a statistical model for each visual word's histogram, typically a Gaussian Mixture Model (GMM). For each image, the statistical properties (such as mean and

variance) of its vocabularized histogram are calculated relative to the trained statistical model. The variation of these properties is the Fisher Vector.

Table 3: Accuracy of SIFT feature based on VLAD model with PCA.

C	**k**	**linear**	**rbf**	**k**	**linear**	**rbf**	**k**	**linear**	**rbf**	**k**	**linear**	**rbf**
0.001		0.272	0.2145		0.253	0.2145		0.2555	0.2145		0.2365	0.2145
0.03		0.2165	0.2145		0.2255	0.2145		0.2385	0.2145		0.2235	0.2145
0.1		0.2125	0.2145		0.2165	0.2145		0.2355	0.2145		0.222	0.2145
0.3	8	0.211	0.2455	16	0.219	0.2375	32	0.2375	0.219	64	0.222	0.215
1		0.208	0.2795		0.219	0.281		0.2375	0.2805		0.222	0.2725
3		0.2095	0.277		0.219	0.2725		0.2375	0.277		0.222	0.2685
5		0.211	0.269		0.219	0.268		0.2375	0.279		0.222	0.2605
0.001		0.2425	0.2145		0.236	0.2145		0.225	0.2145		0.222	0.2145
0.03		0.2385	0.2145		0.2265	0.2145		0.225	0.2145		0.2665	0.2145
0.1		0.2385	0.2145		0.2265	0.2145		0.225	0.2145		0.269	0.2155
0.3	128	0.2385	0.2145	256	0.2265	0.2145	512	0.225	0.238	1024	0.257	0.254
1		0.2385	0.257		0.2265	0.251		0.253	0.272		0.253	0.2625
3		0.2385	0.274		0.2265	0.256		0.253	0.2785		0.245	0.2615
5		0.2385	0.2655		0.2265	0.251		0.1198	0.1786		0.2445	0.258

However, the FV model is computationally more complex and takes longer to compute compared to the VLAD model. By observing the data from the first few iterations, we found that results with larger k values often had slow computation and stable trend changes. Therefore, in this section, we set the range of k to be [2, 4, 8, 16], with the range of C remaining the same as before, and obtained the final results as shown in Table 4.

Table 4: Accuracy of SIFT feature based on FV model with PCA.

C	k=2		k=4		k=8		k=16	
	linear	**rbf**	**linear**	**rbf**	**linear**	**rbf**	**linear**	**rbf**
0.001	0.215	0.2145	0.2145	0.2145	0.2155	0.2145	0.2145	0.2145
0.03	0.254	0.2145	0.2495	0.2145	0.2495	0.2145	0.2005	0.2145
0.1	0.255	0.215	0.2495	0.2145	0.248	0.2145	0.2005	0.2145
0.3	0.257	0.247	0.249	0.237	0.248	0.2445	0.201	0.2305
1	0.256	0.27	0.248	0.26	0.249	0.2425	0.202	0.2405
3	0.2555	0.2745	0.2505	0.2675	0.2485	0.246	0.2015	0.225
5	0.256	0.264	0.25	0.2555	0.2485	0.235	0.2015	0.2185

From the results, it can be seen that FV, despite having a larger computational load compared to VLAD, has a slight performance decrease. The highest accuracy was 27.45% for the rbf model when k was set to 2 and C was set to 3. The speculated reason for this could be that the effectiveness of FV is based on the assumption of Gaussian distribution of feature distribution. If the actual feature distribution deviates significantly from the Gaussian distribution, FV may not be as effective as VLAD. Moreover, FV encodes the variation of statistical properties, which may not be as effective

as VLAD's intuitive residual accumulation method in certain tasks. The results from the table have been plotted in Figure 9.

2.2 Selective Search and ResNet

Selective Search is a region proposal method used for object detection, proposed by Uijlings et al. in 2013[5]. Its main goal is to generate candidate regions that may contain objects, in order to reduce the computational load of object detection. When performing image classification tasks, there may be multiple objects in the image that need to be localized and classified separately. Therefore, before training the classifier, it is necessary to divide the image into smaller regions to improve efficiency. Selective Search has three main advantages: adaptability to different scales, diversity, and fast computation speed. The original pipeline of selective search algorithm is shown in Figure 8.

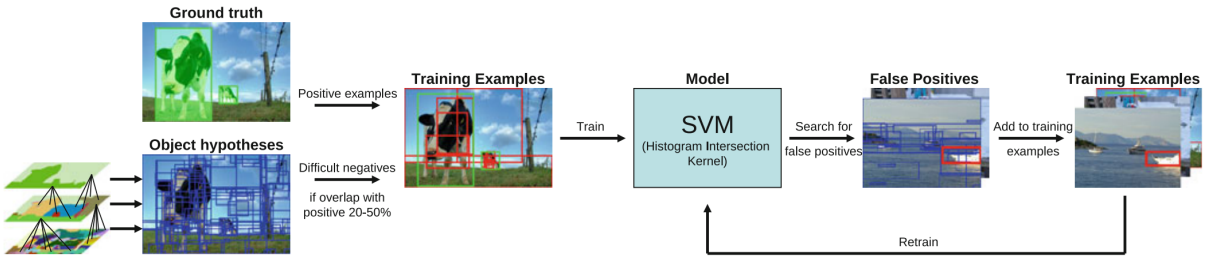


Figure 8: The training procedure of our object recognition pipeline. As positive learning examples we use the ground truth. As negatives we use examples that have a 20–50% overlap with the positive examples. We iteratively add hard negatives using a retraining phase

ResNet Residual Network is a type of deep learning model proposed by Kaiming He et al. from Microsoft Research in 2015 [1]. ResNet introduces so-called "skip connections" (also known as "shortcut connections" or "residual connections") to address the issues of vanishing gradients and representational bottlenecks in deep neural networks. In traditional neural networks, the output of each layer is computed based on the output of the previous layer. As the depth of the network increases, gradients may vanish during backpropagation, making it difficult for the network to learn.

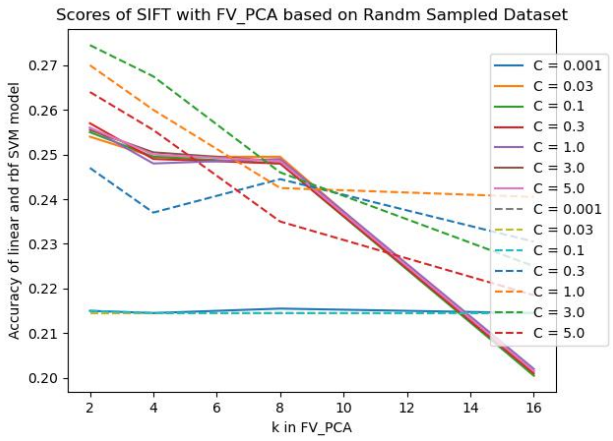


Fig 9: Results of SVM with SIFT and FV based on PCA.

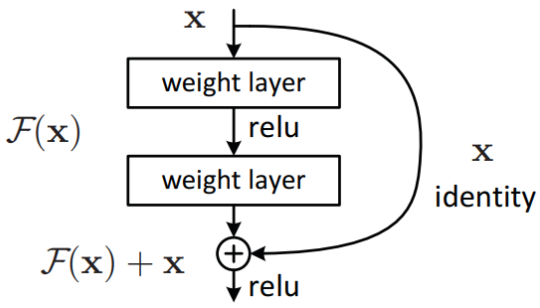


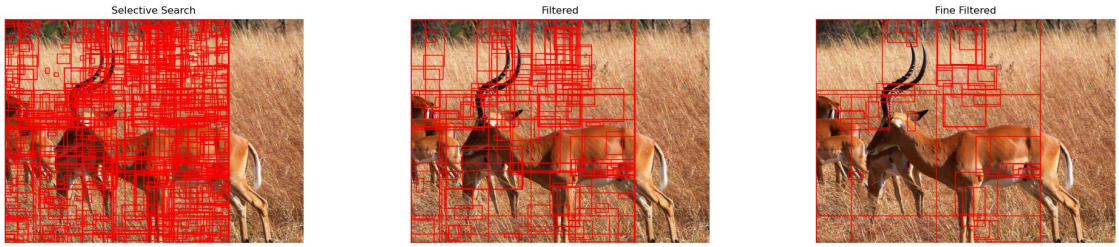
Fig 10: Residual Block of ResNet.

ResNet solves this problem by adding a skip connection in each convolutional block. This skip connection directly adds the input to the output of the convolutional block, forming a so-called ”residual” structure, as shown in Figure 10. This design allows gradients to propagate directly through the skip connections even in very deep networks, thereby alleviating the vanishing gradient problem.

It is important to note that the Selective Search algorithm returns a considerable number of proposals, so we need to apply some filtering to them. This can effectively reduce the computational load and shorten the computation time. A specific example of this is shown in Figure 11.

ResNet solves this problem by adding a skip connection in each convolutional block. This skip connection directly adds the input to the output of the convolutional block, forming a so-called ”residual” structure, as shown in Figure 10. This design allows gradients to propagate directly through the skip connections even in very deep networks, thereby alleviating the vanishing gradient problem.

It is important to note that the Selective Search algorithm returns a considerable number of proposals, so we need to apply some filtering to them. This can effectively reduce the computational load and shorten the computation time. A specific example of this is shown in Figure 11.



(a) original selective search image (b) filterd image (c) fine-filterd image

Figure 11. The optimization process for the Selective Search results.

2.2.1 BOW

In this experimental section, we use the BOW model to perform feature encoding on the local descriptors extracted using Selective Search and ResNet. We set k to the same range as before: [8, 16, 32, 64, 128, 256, 512, 1024]. Based on the results from the previous section, we set the C value for the linear model to 0.01 and for the rbf model to 5. This allows us to significantly reduce the computational load while still aiming to achieve the best possible results.

Table 5: Accuracy of SS feature based on BOW model.

c	0.001	5		0.001	5
k	**linear**	**rbf**	k	**linear**	**rbf**
8	0.0816	0.1079	128	0.2481	0.2424
16	0.1104	0.1385	256	0.2626	0.2574
32	0.1694	0.1846	512	0.2586	0.2468
64	0.2108	0.2117	1024	0.2548	0.2376

The results are presented in Table 5, where the highest accuracy achieved is 26.26%. This

represents a slight improvement over the results obtained with SIFT. The reason for this improvement could be that the ResNet network used in this section is a pre-trained network that is not domain-specific, and thus may not be as well-adapted to the experimental dataset used in this study.

2.2.2 VLAD

In this section, we use the VLAD model to encode the descriptors obtained from Selective Search and ResNet. Since the dimensionality of the features obtained from VLAD is high, resulting in large computational overhead, we also use PCA from the previous section to reduce the dimensionality of the features. Based on the data analysis from the previous section, we can further narrow down the range of k and C to reduce computational load while comparing with the results from the previous section. Therefore, we set the range of k to be [4, 8], the C for the linear kernel to be 0.001, and the C for the rbf kernel to be 5. The results obtained are shown in Table 6.

Table 6: Accuracy of SS feature based on VLAD model.

c	0.001	5		0.001	5
k	**linear**	**rbf**	k	**linear**	**rbf**
4	0.4805	0.0719	8	0.4281	0.1106

It can be observed that the results obtained with the linear kernel are significantly better than those with the rbf kernel, and they also surpass the results obtained with SIFT. The speculated reason for this could be that under the linear kernel, the SVM classifier is more concise, as it only utilizes the support vectors. This makes it more effective in handling high-dimensional but sparse data. Additionally, the method based on Selective Search and ResNet can preserve more complete local image information compared to SIFT.

2.2.3 FV

Taking into account computational resources and the trend of results from the previous section, in this section, for the FV model, we also narrow down the testing range. We set the range of k to be [2, 4], the C for the linear kernel to be 0.001, and the C for the rbf kernel to be 5. On the basis of using Selective Search and ResNet to obtain descriptors and encoding them with FV, we also use PCA to reduce the dimensionality of the features. The results obtained are shown in Table 7.

Table 7: Accuracy of SS feature based on FV model.

c	0.001	5		0.001	5
k	**linear**	**rbf**	k	**linear**	**rbf**
2	0.3146	0.1547	4	0.4637	0.0531

The results obtained by using FV for feature encoding are similar to those of VLAD, slightly inferior. The speculated reason for this should be the same as in SIFT: FV encoding is based on statistical properties, whereas VLAD uses a more direct first-order residual accumulation.

3 Conclusion

In this experiment, we tried different local descriptor extraction methods and different feature encoding methods, covering zero-order, first-order, and second-order feature encoding methods. For the baseline, using SIFT to extract local descriptors from the entire dataset and encoding features with BOW required a huge amount of computational resources and took a very long time, resulting in relatively poor final results. Considering the balance of various categories in the dataset and the computational load, we used a random sampling method to obtain a small dataset of 100 images for each animal. On this basis, the calculation speed was greatly improved, and models with higher classification accuracy were obtained.

Further, to address the performance bottleneck of the SVM model, after using methods like VLAD and FV to complete feature encoding, we applied PCA to reduce the dimensionality of the feature vectors, thereby improving efficiency. By comparing the results, VLAD was superior to the other two methods, but overall, the accuracy was relatively low.

To explore whether we can better obtain image descriptors, we used the deep learning method of Selective Search and ResNet to replace SIFT. We applied some filtering to the proposals obtained from Selective Search, and then used BOW, VLAD, and FV methods to encode features and reduce dimensionality before training an SVM classification model. After using the deep learning method, there was indeed a significant performance improvement compared to SIFT. The highest accuracy achieved in the final experiment was 48.05%, and under the deep learning method, VLAD still outperformed the other two methods.

References

[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[2] Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez. Aggregating local descriptors into a compact image representation. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3304–3311, 2010.

[3] D.G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pages 1150–1157 vol.2, 1999.

[4] Josef Sivic and Andrew Zisserman. Efficient visual search of videos cast as text retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(4):591–606, 2009.

[5] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International journal of computer vision*, 104:154–171, 2013.