Project3: Feature Encoding for Image Classification

Jieyu Li, Dongyue Li, Hongbin Chen, Haoxuan Wang
Course: CS245 - Data Science Principles

***Abstract*— This is a report on the experiments we have done for image classification using feature encoding. We extracted local descriptors for each image using different methods, including traditional methods such as SIFT and deep leanring methods. After obtaining the local descriptors, we use three different encoding methods: BOW, VLAD and Fisher Vector to encode the local descriptors into a feature vector. The feature vector is later put into SVM for classification. We further discuss the different extracting methods and encoding methods and analyze their reasons for such different performances.**

I. Introduction

A. Project Description

In this project, we are required to extract local descriptors from raw images and learn feature vectors from these descriptors. These features are further used for image classification. Based on the experiment results and observations, discussions are made to analyze why these methods performa in such different ways.

B. Dataset

The AwA2 (Animals with Attributes) dataset consists of 37322 images of 50 animal classes with pre-extracted deep learning features for each image. We split the images in each category into 60% for training and 40% for testing (which is the same as $Project1, 2$) and shuffle them.

1) Experiment Settings:

1) Python3.6
2) Sklearn, Pandas, Seaborn, Matplotlib, VLFeat
3) Matlab

II. Local Descriptors Extraction

In this section, we describe how we extracted local descriptors from raw images. The performance of classfication is highly dependent on this step as it determines whether or not important local information can be found. Traditional method and deep learning method are both used for comparison.

A. SIFT Descriptors

In this section, we use Matlab to extract SIFT features from the original images. For each image, the number of SIFT descriptors varies and are taken as follows:

$$N_{SIFT} = min(N_{SIFT}, 512) \tag{1}$$

In this way, we can get relatively reasonable number of SIFT descriptors and make them comparable.

B. Deep Learning Descriptors

We first need to extract proposals from the original images and selective search is used. After selective search, each image is able to produce from 10 to 300 proposals. Due to the huge variance of proposal numbers, we take at most 100 proposals for each image. To mention here, we only take proposals that have the largest size because we train the deep learning network with complete images. And this settings can help the network to extract more important features.

The deep neural network we chose is the pre-trained InceptionV3 network on ImageNet. After fixing the structure and parameters of all the convolutional layers in front, we added two fully connected layers of dimension 1024 and 128, and a softmax layer for classification at last. The model is fine-tuned for the first time using the original dataset, and then we fix the first 249 layers and fint-tune the network the second time. After fine-tuning, our accuracy can reach 91.9%. Our deep learning features uses the output of the dense layer (128 dimensions).

III. Feature Encoding And Image Classification

After obtaining the local descriptors, we try to encode them into feature vectors to stress important information and make them mathematically useful and computationally understandable. Three different feature encoding methods are adopted: BOW (bag-of-word), VLAD (vector of locally aggregated descriptors) and Fisher Vector, which extract different order information. The three encoding methods all uses k-means to group the descriptors into K clusters, and then uses different information of the clusters. After obtaining the feature vectors, we use SVM to do classification. Their performances with different types of local descriptors are stated below.

A. BOW

BOW (bag-of-word) encode $0-order$ information. The dimension after encoding is euqal to K.

1) SIFT Features: When using SIFT features, we obtain the results shown in Fig.1. From the graph, we can see that the absolute accuracy is low and the accuracy increases with the increase of K.

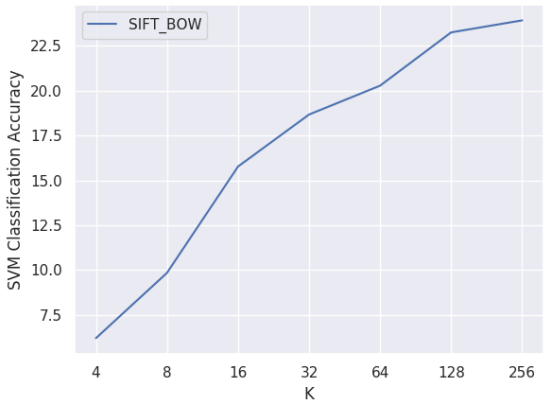

Fig. 1. Classification Acc with SIFT+BOW

After analyzing the theories and data, we induct that the reasons might be due to:

- The original image include around 4000 descriptors. After we encode them into only $4, 8, ..., 256$ dimensions, most of the information are lost. Thus, the features we have got is "underfitting" the original information. So the more dimensions, the more information is preserved.
- SIFT mostly deals with the features of rotation and scaling. But extracting deatiled features and learning an effective codebook might be too overwhelming.
- Also, BOW only possess $0-order$ information, and the local space image features are ignored.

2) Deep Learning Features: When using deep learning features, we obtain the results shown in Fig.2.

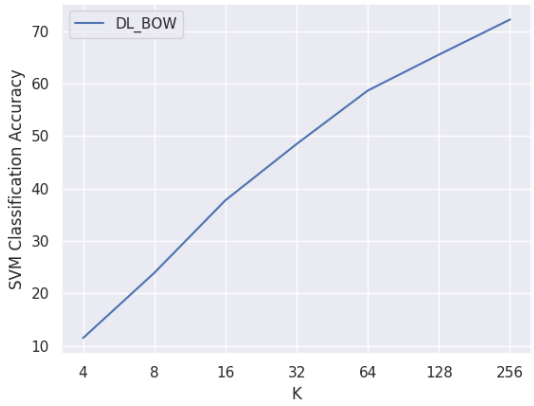

Fig. 2. Classification Acc with DL+BOW

From the graph, we can obtain the following observations:

- The curve is similar to a straight line. This is due to the property of BOW, as the information encoded by BOW positively correlated with K's value.
- The absolute values of accuracy are way better than SIFT+BOW. This is because deep learning features contain more useful information than SIFT. While SIFT only concentrates on points in graphs, deep learning features can make use of local features.

B. VLAD

VLAD encode $1-order$ information. It include the mean of the data samples. Within each cluster, VLAD encode more information into the feature vectors. Thus the dimensionality of the vectors are huge, so we reduce the dimension of each feature vector to 1024 dimensions whenever they

are larger than this limit. PCA is method used as the dimension reduction method for its robustness.

1) Using SIFT: The results for applying VLAD to SIFT vectors are shown in Fig.3. Just to mention that, the feature dimension of VLAD is Kd (d is the dimension of descriptors, which we set as 128), thus as $K > 8$, we have to do dimensionality reduction on the feature vectors to reduce them to a dimension of 1024.

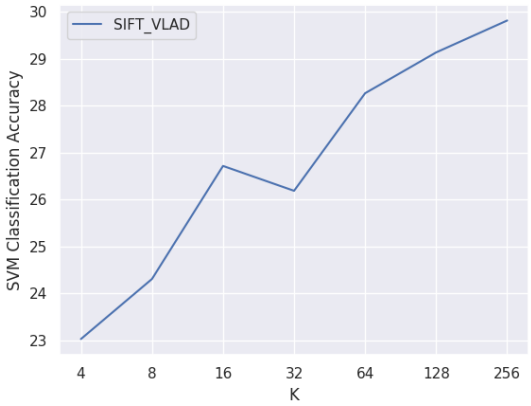

Fig. 3. Classification Acc with SIFT+VLAD

From the figure, we can get the following observations:

- The accuracy is mostly increasing with the increase with K. But the performance is limited to around 25%.
- Though the feature dimension is large, but in fact we only cluster them into at most 64 clusters. The induction is that much information are lost.
- VLAD performs better than BOW due to the fact that it can encode space features (such as the relative position of head and legs).

2) Using Deep Learning Features: The results for deep learning features with VLAD is shown in Fig.4.

We observe the following facts:

- The best performance is achieved when $K = 16$, which is also the point when feature dimension is 1024. When K is larger than 16, we begin to do dimensionality reduction on the feature vectors and the accuracy begin to drop. This is due to the loss of information when doing dimensionality reduction.

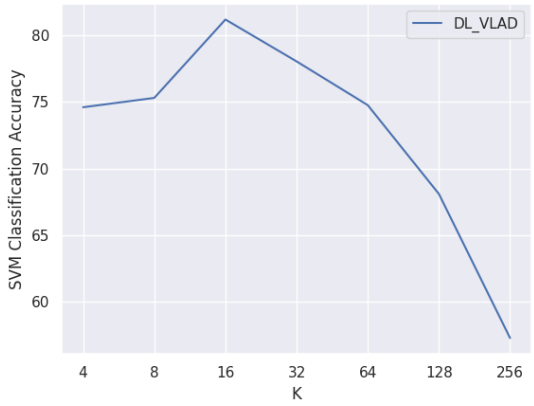

Fig. 4. Classification Acc with DL+VLAD

- This observation is different from what we have observed of SIFT features. We guess that this might be the reason of: While PCA can overcome the curse of high dimensionality, it can also lose information. SIFT descriptors may have benefited from PCA more while deep learning features may have suffered from PCA more. This might be due to the fact that SIFT contains more overlapping and useless information but deep learning features are mostly useful and important.

C. Fisher Vector

Fisher Vector encode $2 - order$ information. It include the mean of data samples as well as the covariance. Since Fisher Vector encode features into dimension of $2Kd$, thus we do dimensionality reduction when $K > 7$ and turn the vectors into 1024 dimensions.

1) Using SIFT: The results for using SIFT with Fisher Vector is shown in Fig.5.

From the figure, we reach the following observations:

- The performance of the Fisher Vector is slightly better than VLAD due to its encoding of more information. But the absolute value is still low due to SIFT's poor performance.
- The accuracy reaches the maximum when $K = 16$, which is also the case when feature dimension is 2048.
- As K gets bigger, the accuracy begin to decrease because: (1) We are doing dimen-

TABLE I

CLASSIFICATION ACCURACY

k	4	8	16	32	64	128	256
SIFT+BOW	6.20	9.84	15.77	18.67	20.28	23.25	23.92
SIFT+VLAD	23.03	24.31	26.72	26.19	28.27	29.14	29.82
SIFT+Fisher Vector	26.99	26.32	27.33	27.13	23.30	22.03	17.60
DL+BOW	11.47	23.90	37.76	48.49	58.69	65.55	72.25
DL+VLAD	74.59	75.29	81.17	78.03	74.75	68.09	57.28
DL+Fisher Vector	62.86	47.65	11.19	3.50	3.28	2.71	2.35

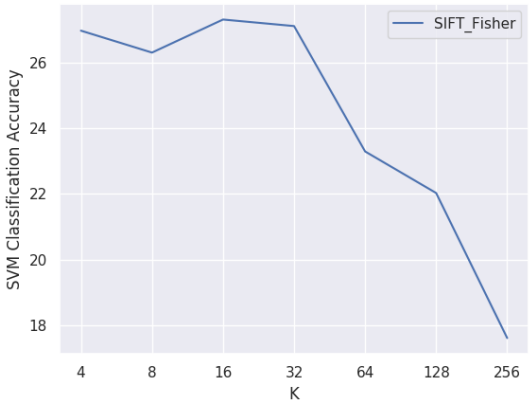

Fig. 5. Classification Acc with SIFT+Fisher Vector

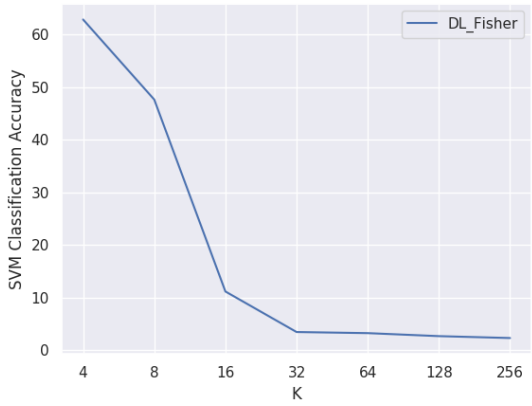

Fig. 6. Classification Acc with DL+Fisher Vector

sionality reduction on the features and information are lost. Though cluster number increases, the information in each cluster are reduced. (2) The multivariate Gaussian distribution might be performing as a drawback for the classification task. With the increase of K, the original distribution cannot be split into so many individual Gaussian distributions. Thus, much information have overlappings and some information might be lost.

2) Using Deep Learning Features: The results for using deep learning features with Fisher Vector is shown in Fig.6.

From the figure, we observe that:

- The classification accuracy decreases with the increase of K, which is quite surprising. The reason for this is inducted as: The output of deep learning models often follow Gaussian distribution, and Fisher Vector uses EM algorithm to convert the original distribution into a multivariate Gaussian distribution. Thus in our experiment, we are splitting a Gaussian distribution into multiple (K) Gaussian distributions, thus much information is lost. The decrease of accuracy is apparent.

D. Summary Of Observations

The detailed data for our experiments are shown in the Table I. And to make the curves more comparable, we plot the comparisons in the Fig.7, 8.

From the figures, we can observe that:

- When we are using SIFT descriptors and K is small, Fisher Vector should always be the best. However, the reduction of dimensions and the drawback of too-large-K multivariate Gaussian distribution are vital causes for the decrease of accuracy.
- As K gets larger, BOW and VLAD are both performing better and better. This might be due to the fact that SIFT descriptors contain too much unnecessary information: More dimensions makes BOW capture more important information, and dimensionality reduction does not reduce much useful information.

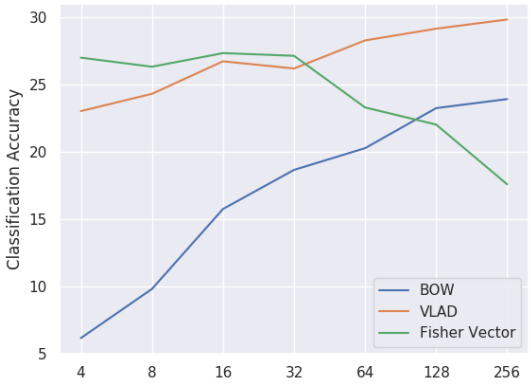

Fig. 7. Comparison Using SIFT Features

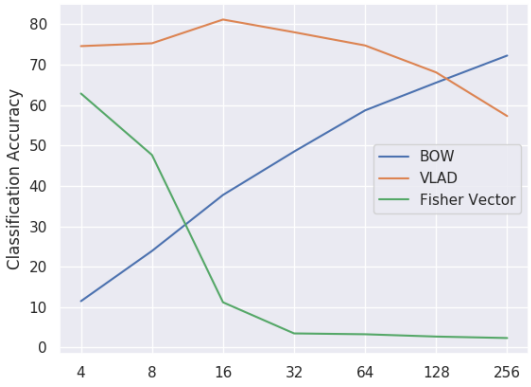

Fig. 8. Comparison Using Deep Learning Features

- When we are using deep learning features, VLAD and Fisher Vector all behave worse after a certain K value. This might be the reason that deep learning features are always useful and dimensionality reduction loses much information. Also, Fisher Vector's accuracy is always decreasing due to its use of GMM.

IV. Further Experiments And Discussion

A. Dimension Reduction Using LDA

Despite PCA for dimension reduction, we have also tried LDA. LDA has proved to be performing very well according to our conclusions in $Project1$. However, the dimension LDA can reduce is only limited to 50 dimensions and that results in huge loss of information. The best accuracy can only reach 40% according to our experiments.

B. Performance Of Different Encoding Methods

The performance of Fisher Vector is better than VLAD and better than BOW. Their different performances are due to the different order information that they are encoding. The amount and quality of encoded information both counts for the final performance of classification.

C. Comparison With Project1

Project1 also does SVM classification with features. We can compare our results with the ones in project1. Project1 uses already extracted 2048 dimesion deep learning features as the input and do dimensionality reduction, and at last do classification. We only compare features with the same dimensionality (1024). PCA achieves an accuracy of 92.92% and sparse encoding reaches an accuracy of 91.02%, which is much better than our results. We guess the following reasons: The features learned by the codebook is worse than the features output by the deep learning model due to the fact that we are always reducing dimensions and splitting Gaussian distributions, information are always lost and accuracy cannot improve.

V. Conclusion

In this project, we extract local descriptors from raw images, encode them into and feature vectors, and do classification based on these vectors. Our best classification result is achieved by using deep learning descriptors and VLAD method, getting an accuracy of 81.17%. We analyzed the different reasons of why our results behave in different ways and looked into the theories of different algorithms to explain their performance. We further compare our results with the ones in Project1, and point out the superiority of deep learing features.

References

[1] J.R.R. Uijlings and K.E.A. van de Sande and T. Gevers and A.W.M. Smeulders. Selective Search for Object Recognition. International Journal of Computer Vision, 2013.

[2] Van de Sande, K E A and Uijlings, J.R.R. and Gevers, T and Smeulders, A.W.M. Segmentation as Selective Search for Object Recognition. ICCV, 2011.