A Dissection and Evaluation of Scene Recognition Algorithms

Ahmed Gashgash [*1] **Yunxiang Zhang** [*2]

Abstract

A widely considered problem in Computer Vision is scene recognition. Its important in many applications since it defines a context for object recognition. However, with the abundance of image data sets and the rise of deep neural networks, scene recognition hasn't found as much success as other tasks such as object classification. In this paper we dissect and examine the pipeline of scene recognition algorithms. We treat this process as two separate stages: feature extraction and classification. We consider different feature descriptors, traditional and neural network based methods. We also explore different approaches to combine different feature vectors. We provide a comprehensive comparison of the performance of different descriptors, dimensionality reduction techniques and classifiers on the Places365 dataset.

1. Introduction

Scene recognition is one of the fundamental problems in the field of computer vision. Previous work in this domain tackled this problem through three main components, namely feature extraction(1; 2), classification (3; 4), and feature fusion or combination(5; 6). In this project, we provide a relatively extensive comparison between the performance of individual feature extractors and their combinations coupled with different classification models on the scene recognition task. In detail, we will:

1. Evaluate different feature descriptors including classical non-learning based descriptors and more modern deep learning based feature extractors, namely convolutional neural networks (CNN)
2. Compare the performance of different classifiers, namely SVMs and Multi Layer Perceptrons (MLP), on the previously extracted features
3. Analyze the change in performance of combining CNN feature extractor with other visual descriptors relative to the performance of each individual component

[*] Equal Contribution

1.1. Motivation and Related Work

Due to the success of CNNs on classification tasks (7), a lot of researchers tend to shift there focus from classical non-learning based approaches to neural networks. Therefore, one natural and essential question is to dissect the CNN to understand how the model gives rise to such performance increase. Therefore, we separate the CNN into the feature extraction stage and classification stage and compare them with the corresponding parts of the classic models individually. We believe that such an approach could provide an insight into the underlying reasons for the performance difference among various scene recognition algorithms. In addition, recent research shows that combining features extracted from different descriptors could provide performance gain in scene classification tasks. Thus, we conduct experiments to evaluate such gained performance by benchmarking different combinations of descriptors.

In previous work (8), the performance of SIFT and CNN is compared for scene recognition. However, these two models are evaluated as a whole, while we compare the feature extraction and classification separately. The work in (9) provides a full comparison of classic models with different types of visual descriptors and classifiers. However, in their work, they do not provide a detailed comparison between the classic models and CNN. (5; 6) propose new models that combine features extracted from both handcrafted features and CNN and compare those models with standard models. Those proposed models have distinctive structures, which might partially contribute to performance improvement. For example, in (6), the proposed model uses improved HDCT based saliency method in the feature extraction stage of the SIFT descriptor. However, in our experiments, to exclude such influence, we do not have any special modification in the feature extraction and classification stages.

2. Methods and Models

2.1. Visual Descriptors

In this project, we consider the following four descriptors:

2.1.1. Histogram of Oriented Gradients(HOG)

HOG (10) is a visual descriptor of the image data, which computes the number of occurrences of gradient orientation in localized portions or regions, i.e., dense grids of uniformly spaced cells, of an image. The HOG would generate a histogram of oriented gradients for each region separately, normalizes the result using a block-wise pattern, and returns a descriptor for each region. We will use the generated descriptors of the regions as the image feature vector extracted by the HOG.

2.1.2. Scale-Invariant Feature Transform (SIFT)

SIFT is a method for detecting and extracting distinctive invariant features or keypoints from images. The extracted features are invariant to image scale and rotation. Here, we first apply the SIFT method to extract the keypoints of each image and then we use a bag of visual words model to classify the images. In detail, after extracting the keypoints of images, we use k-means clustering algorithm to create k cluster centers(centroids) as the dictionary of visual words. Then we create a histogram of k values for each image by assigning the keypoints to their nearest centroids and counting the corresponding frequencies.

2.1.3. GIST

GIST (11) descriptor is designed specifically for the task of scene recognition. It forms the representation of the image by modeling the shape of the scene. The shape of the scene is computed or estimated by a set of perceptual dimensions(naturalness, openness, roughness, expansion, ruggedness) specifically dedicated to describing the spatial properties of the scene. We apply the GIST to extract features from each image and use the feature vector as the input for the classifier.

2.1.4. Convolutional Neural Network (CNN)

CNN is a class of deep neural networks that can learn and extract relevant structural features from an image at different levels, which mimics the functionality of a human vision. Here, we use the ResNet-50 without the last fully connected layer as the feature extractor. The resulting representation of each image is a 2048-dimensional feature vector.

The reason for considering HOG, SIFT, and GIST descriptors is that these three descriptors are the most fundamental visual descriptors, which can well represent the traditional class of the image feature extractors. A lot of the other descriptors are the follow-up work based on these three, such as dense sift (12). And the reason why we choose ResNet-50 as our CNN model is as follows. In our project, we would like to make a fair comparison among all the descriptors. Following this guideline, we want to make the dimensions of the feature vectors of all the visual descriptors the same so that the model complexity of the classification algorithms will stay fixed. And the dimension of the feature vector by ResNet-50 is a more balanced choice in our case compared to those by other CNN models. For example, although, given the existing benchmark, VGG16 has the best performance on the dataset we experiment on, the dimension of its feature vector is 7x7x512, which is not suitable for other descriptors.

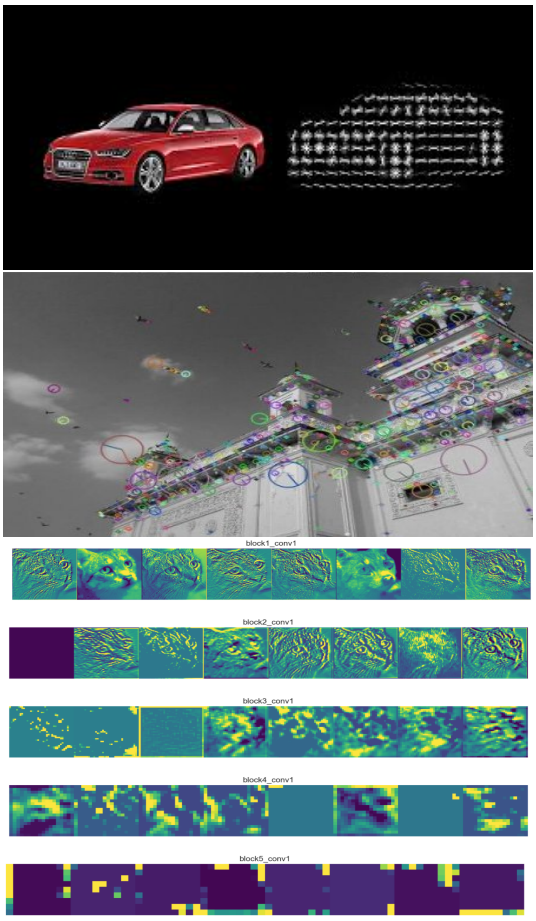

Figure 1. Visualizations of descriptors. From top to bottom: HOG descriptors of a car image, SIFT descriptors of an architect image, and CNN features of a cat image.

2.2. Feature Combination

To combine the extracted CNN features with the visual descriptor (HOG, SIFT, GIST, respectively), we first concatenate the two feature vectors and then perform dimensionality reduction to compress the concatenated vector. By doing so, we can make sure the combined feature vector has the same dimensionality as the individual feature vector. The reason for conducting dimensionality reduction is to

avoid the increase in the classifier's model complexity due to the dimensionality increase of the combined feature vector and for fair comparison. Here, we specifically consider the following two methods.

2.2.1. Principal component analysis (PCA)

Principal component analysis (PCA) simplifies the complexity in high-dimensional data while retaining trends and patterns. It uses an orthogonal transformation to convert a set of possibly correlated variables into a set of linearly uncorrelated variables called principal components.

2.2.2. Autoencoder

An autoencoder is a type of artificial neural network used to learn efficient data encodings in a supervised manner. The aim of an autoencoder is to learn a compressed knowledge representation of the original input for a set of data by imposing a bottleneck in the neural network. In addition to the encoder network, a decoder is also learned, that aims to reconstruct the initial input from the latent bottleneck layer.

2.3. Classification Models

2.3.1. Multiclass SVM

SVM is one of the most wildly used supervised learning models to perform image or scene classification. The algorithm tries to find the optimal hyperplane to categorize the data points. Here, we use the multiclass SVM with the one-versus-the-rest strategy. The strategy fits one classifier per class and for each classifier, the class is fitted against all the other classes. This is different from the one-vs-one strategy, which constructs one classifier per pair of classes. The reason we choose the one-versus-the-rest is due to the computational efficiency compared to the other strategy. Also, we use the linear kernel for the multiclass SVM.

2.3.2. MLP

MLP is a class of feedforward neural networks that consist of multiple fully-connected layers. In this project, we will use a two-layer MLP, which has one input layer of dimension 2048, one hidden layer of dimension 1024 and one softmax layer as the output layer with dimension 365(number of categories).

3. Experiments and Results

3.1. Dataset

The Places365 dataset is the core set of Places2(7) dataset, which is one of the most recent and wildly used datasets for the scene recognition task. The Places365 has 365 semantic categories of places such as bedroom, streets and etc. In our experiments, we used a subset of the Places365 dataset, which has the same number of categories as the original dataset. For each category, we have 1200 images as training examples and 100 images as testing examples. Also, for each category, we used 1000 training examples as the training data and 200 training examples as the validation data. And each image has a dimension of 256x256x3. We used Python 3.7.3 throughout the project. Our neural network models were implemented using PyTorch library 1.3. All models were trained on the Ryzen Threadripper 2990WX CPU and two NVIDIA GeForce RTX 2080 Ti GPUs.

Figure 2. Sample Images from various categories of the Places2 dataset (two samples per category).

3.2. Evaluation Metric

In the experiments, we will use three measures to evaluate the performance of each model: precision, recall, and F1 score. Note here, these three evaluation metrics are measured across all the classes and we formally define them as follows. Let M be the confusion matrix of all classes, the precision and recall for class i are defined as:

$$\text{Precision}_i = \frac{M_{ii}}{\sum_j M_{ji}}$$

$$\text{Recall}_i = \frac{M_{ii}}{\sum_j M_{ij}}$$

We take the mean of precision and recall across the classes. The F1 score is defined as:

$$\text{F}_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

which can be viewed as a combined metric using precision and recall.

3.3. Experiments and Results

3.3.1. Comparing individual visual descriptors

	HOG	SIFT	GIST	CNN
SVM	0.2732	0.3052	0.1522	0.4678
MLP	0.3377	<0.1	0.1589	0.4770

Figure 3. Comparisons of HOG, SIFT, GIST, and CNN under multiclass SVM and MLP in terms of F1 score.

In this experiment, we compare the performance of the scene recognition algorithms using different visual descriptors. We apply the HOG, SIFT, GIST, and CNN to extract the features of the image data separately. Then we pass the extracted features of the image data to perform the classification using both classifiers independently (multiclass SVM or MLP). In order to exclude the influence of the model complexity of the classifier, we force that the length of the feature vectors extracted by each descriptor is the same. For example, the feature vector from the HOG has the same dimension as the feature vector from the SIFT. Here, we choose the dimension to be 2048. The reasons we choose this particular dimension are as follows. First, 2048 is the dimension of the feature vector extracted by the convolutional layers of ResNet-50, i.e., the result right before the last fully connected layer. Therefore, to keep the feature vector dimension of each descriptor the same. We use this particular value. More importantly, as we mentioned before, the main reason we choose ResNet-50 as our CNN model to make comparisons is that the dimensionality of the feature vector of ResNet-50 is also a reasonable choice for other descriptors.

As we can see in Figure 3, the CNN feature extractor outperforms the traditional visual descriptors, namely HOG, SIFT, and GIST, within both classification models. This is not surprising since, through training and learning, the convolutional layers are able to effectively extract the spatial structure and information of an image at a relatively high level. If we take a detailed look at the results, we can also see that the CNN feature extractor is significantly better than the other three visual descriptors. And we believe this is due to the advantage of the depth of our neural network model. We have not conducted experiments on other CNN structures, but we expect to see that in general, the feature extractors of other CNN models should have better performance than the three visual descriptors. And we will explore other structures in our future work.

3.3.2. Combining CNN features with other descriptors

	CNN+HOG+Autoencoder	CNN+HOG+PCA	CNN	HOG
SVM	0.3647	0.3325	0.4678	0.2732
MLP	0.3772	0.3593	0.4770	0.3377

Figure 4. Comparisons of CNN+HOG, CNN, and HOG under multiclass SVM and MLP in terms of F1 score.

In this experiment, we combine the CNN features with the features extracted from other descriptors, namely HOG, SIFT, and GIST. First, we combine the CNN with other descriptors by concatenating the two feature vectors. Since each feature vector has 2048 dimension, the combined feature vector would have 4096 dimensions. Then we apply the PCA or the autoencoder separately to reduce the dimension of the combined feature vector from 4096 to 2048. Note here that the autoencoder is trained separately for each combination of the feature vectors. Followed by that, we pass the combined feature vectors as the inputs into the classifier, MLP and multiclass SVM separately. Note here, when comparing the different combined feature vectors, we need to ensure the comparisons are made under the same dimensionality reduction technique and the same classification model.

Figure 4 shows the result of combining CNN with HOG feature vectors and its performance relative to the individual components. As we can see in the table, for either multiclass SVM or MLP model, although HOG+CNN outperforms the HOG, the performance of the combined feature is much worse than the CNN feature extractor. And the underlying reason might be that when applying the dimensionality reduction technique, i.e., the autoencoder or PCA, we are fusing the feature vectors we obtained from the CNN and HOG instead of just putting them together through concatenation. Therefore, the quality of the feature vector of CNN might be severely impacted by the feature vector of HOG during this process, which results in the decrease of the performance for the combined feature vector. According to Figure 6, we can see that similar situations also happen in CNN+SIFT and CNN+GIST. To further confirm our reasoning, we conducted one more experiment for CNN+HOG, which is to concatenate the CNN feature vector and the HOG feature vector without any dimensionality reduction. Due to the time constraint, we only use the multiclass SVM as our classification model. And the result is shown in Figure 5.

Image Descriptor	Dimensionality Reduction/Classifier	Precision	Recall	F1 Score
HOG	SVM	0.2659	0.281	0.2732
	MLP	0.3298	0.346	0.3377
SIFT	SVM	0.3031	0.3074	0.3052
	MLP	<0.1	<0.1	<0.1
GIST	SVM	0.1593	0.1457	0.1522
	MLP	0.1576	0.1603	0.1589
CNN	SVM	0.4518	0.4849	0.4678
	MLP	0.4772	0.4768	0.4770
CNN+HOG	Autoencoder + SVM	0.3427	0.3898	0.3647
	Autoencoder + MLP	0.3859	0.3689	0.3772
	PCA + SVM	0.3218	0.3439	0.3325
	PCA + MLP	0.3545	0.3642	0.3593
CNN+SIFT	Autoencoder + SVM	0.3252	0.276	0.2986
	Autoencoder + MLP	0.1322	0.1239	0.1279
	PCA + SVM	0.2961	0.2574	0.2754
	PCA + MLP	0.1041	0.1163	0.1099
CNN+GIST	Autoencoder + SVM	0.2545	0.2687	0.2614
	Autoencoder + MLP	0.2733	0.2926	0.2826
	PCA + SVM	0.2606	0.2642	0.2624
	PCA + MLP	0.2724	0.2836	0.2779

Figure 6. Comparisons of all descriptors, dimensionality reduction techniques, and classification algorithms in terms of precision, recall, and F1 score.

	CNN+HOG+Autoencoder	CNN+HOG+PCA	CNN + HOG + Concatenation	CNN
SVM	0.3647	0.3325	0.4853	0.4678

Figure 5. Comparisons of CNN+HOG with and without dimensionality reduction technique in terms of F1 score. The classification model is multiclass SVM.

According to the result, the CNN+HOG with concatenation has the best performance among all the descriptors shown in the table, which provides evidence for our theory of performance degradation due to the dimensionality reduction. In future work, we will conduct a more comprehensive experiment to support our theory.

3.3.3. Comparing different classification algorithms

As we can see in Figure 6, the MLP has a generally better performance than the multiclass SVM and we believe that this is due to the higher complexity and the expressiveness of the MLP model compared to the multiclass SVM. However, there is one case that particularly draws our attention. That is applying MLP to the SIFT descriptor or the CNN+SIFT descriptor. Figure 3 shows that applying MLP to SIFT features perform poorly in our experiment. The F1 score is less than 0.1. From our perspective, the underlying reason for this phenomenon is that in SIFT, we use the visual-bag-of-words technique. Since the number of SIFT keypoints for some images is very low such as 100 or 200, the feature vectors for those images are very sparse. However, the MLP model we define is a relatively large model. Therefore, the MLP model we used here might not be suitable for the SIFT descriptor in this case. Furthermore, this observation also shows the difficulty of establishing the correct and fair benchmarks of different algorithms. In our project, to ensure fairness, we use the same classification algorithms under the same model complexity for all visual descriptors. However, this strategy might also be considered unfair in the sense that the classification algorithms might not work well or fail naturally with some visual descriptors such as SIFT in our case.

4. Conclusion

In this project, we dissect and examine the pipeline of scene recognition algorithms. In detail, we separate the algorithms into two parts: the feature extraction stage and the classification stage. Also, we explored different ways to combine the feature vectors extracted from different descriptors. From our results, we have found that the CNN feature extractor has the best performance across the descriptors we consider in our project. Also, the coupling of the CNN feature vector with the feature vector of other descriptors through autoencoder and PCA will severely affect the quality or the performance of the CNN feature extractor. As for the classification algorithms, the MLP model outperforms the multiclass SVM in general. However, in our experiment, we also demonstrate that in some cases, certain classification algorithms might fail naturally when applied to some descriptors.

5. Future Work

As mentioned before, considering the performance representation and the dimensionality issue, we particularly choose ResNet-50 as our CNN model. However, according to our experiments, we can see that the feature vector dimension we choose might not be suitable for all the descriptors. Although it is hard to choose a CNN model whose feature vector dimension will work universally well with other descriptors, it is still worth trying other CNN models such as ResNet-18 or VGG16 with different feature vector dimensions. In addition, in the experiments, we have shown that the feature vector combination with simple concatenation outperforms the ones with dimensionality reduction. In future, we would like to conduct experiments on other descriptors to further check the result we have seen in the CNN+HOG case. At last, to form a deeper understanding of the different models, we will also look at the class activation maps to analyze their behaviors in future work.

References

[1] 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.

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

[3] Bernhard Scholkopf and Alexander J Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2001.

[4] Stephan Dreiseitl and Lucila Ohno-Machado. Logistic regression and artificial neural network classification models: a methodology review. *Journal of biomedical informatics*, 35(5-6):352–359, 2002.

[5] Ke Yan, Yaowei Wang, Dawei Liang, Tiejun Huang, and Yonghong Tian. Cnn vs. sift for image retrieval: Alternative or complementary? In *Proceedings of the 24th ACM International Conference on Multimedia*, MM '16, pages 407–411, New York, NY, USA, 2016. ACM.

[6] Muhammad Rashid, Muhammad Attique Khan, Muhammad Sharif, Mudassar Raza, Muhammad Masood Sarfraz, and Farhat Afza. Object detection and classification: A joint selection and fusion strategy of deep convolutional neural network and sift point features. *Multimedia Tools Appl.*, 78(12):15751–15777, June 2019.

[7] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1452–1464, 2017.

[8] Varsha Devi Sachdeva, Junaid Baber, Maheen Bakhtyar, Ihsan Ullah, Waheed Noor, and Abdul Basit. Performance evaluation of sift and convolutional neural network for image retrieval. 2017.

[9] Xue Wei, Son Lam Phung, and Abdesselam Bouzerdoum. Visual descriptors for scene categorization: Experimental evaluation. *Artif. Intell. Rev.*, 45(3):333–368, March 2016.

[10] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1 - Volume 01*, CVPR '05, pages 886–893, Washington, DC, USA, 2005. IEEE Computer Society.

[11] Aude Oliva and Antonio Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42(3):145–175, May 2001.

[12] L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 524–531 vol. 2, June 2005.