Towards Scene Recognition: From Hand Crafted Features to Deep Learning Based Classifier

Yuzhe Yang*, Yuqian Dai†, Lyu Zheng‡ and Qi Zhang§
School of Electronics and Computer Science, University of Southampton
Member ID: *yy1a19, †yd6u19, ‡ zl9y19, §qz7y19

***Abstract*—Scene recognition has been a challenging and classic problem in the field of artificial intelligence and computer vision for a long time. Generally, scene recognition works extract different scene features from the image and collaborate object information to make a scene prediction. Traditional approaches such as K-Nearest-Neighbor (KNN) and Support Vector Machine (SVM) have obtained a great success in scene recognition. In recent years, the deep learning technique has greatly developed, which could extract sufficient scene features such as object information, salient region information and etc from the training set. However, the deep learning approaches usually need a large number of labelled images, which is difficult to obtain and high cost. In this assignment, based on the aforementioned ideas, we develop three different classifiers, which include a KNN classifier, a linear classifier using bag-of-visual-words, and the third one is our proposed model Saliency-Aware Scene Recognition (SASR). We split 10% of the training set and use it as our validation set. The Top-1 accuracy on validation set is 0.86 and the Top-5 accuracy is 1.00, which indicates that our proposed SASR can outperform the other approaches and solve the scene recognition problems efficiently.**

I. Introduction

Scene recognition is a challenging problem and it has been widely used in object detection, image understanding and etc. In this assignment, we develop three classifiers based on the provided scene recognition data sets. The sampled images from the training set is shown in Fig 1, which contains 15 classes and each class folder includes 100 images. In the first section, we implement a K-Nearest-Neighbor (KNN) classifier and give an introduction about the implementation details together with parameter setting. Then, we develop a linear classifier using Bag-of-Visual-Words. Finally, we proposed our Saliency-Aware Scene Recognition (SASR) model and discuss the model structure and the model performance.

A. Team Work Declaration

In this assignment, we work as a team, whose name is "U6C". For reference only, in this work, each member takes an individual part of the whole assignment. Yuqian Dai is responsible for the run1 and he develops the KNN classifier. Zheng Lyu is responsible for the run2 and he takes the responsibility to the linear classifier using a bag-of-visual-words feature based on fixed size densely-sampled pixel patches. Yuzhe Yang and Qi Zhang are responsible for the run3 and they propose the SASR model together, so their contributions to the run3 are equal. The report paper is written by our group members together.

Fig. 1. Sample images and labels from the provided training data

II. K-Nearest-Neighbor classifier

A. Implementation

The calculation process of the KNN classifier is as follows. The training data and its label are feed in, and this data is directly saved in the classifier. When the test set is passed in, the distances from all samples in the test set to all training set samples are calculated in order. For the distance calculated for each group, the first k bits are taken, and the label of the corresponding training set with the most types of the k bits is the label of the sample. In essence, it divides the training sample set and uses it as a model, and then makes predictions based on the majority vote. The average value of classification accuracy is around 0.12.

B. Training

Training a KNN classifier is not a process of fitting a function, it meremly saves all the data into the classifier, which are used to calculate the distance between the test set and the training set. The training set is divided into two parts, 10% of training data is used as the validation set. It is expensive to compare the test set with all the training sets. Since the dimensions of the image are generally high, KNN is generally not used because the cost of calculating the distance is high.

C. Patameters

KNN has only one parameter, k or n neighbor. This value of it is used to view the labels of the nearest k training set samples after calculating the distance between the training set and the test set. The inapproprate K choosing may lead to reducing accuracy and bias. Therefore, accuracy is not

positively related to k, it has an extreme value at some point. Generally speaking, when using the nearest neighbor classifier, K will always be assigned a relatively large value. This will make the decision boundary smoother and get better results. However, a relatively large K will also cause overfitting problems. Therefore, we need to observe the validation set to confirm the model performance.

D. Pre-processing feature extraction

The pre-processing phase is carried out in accordance with the recommended way of operation. First, cut out the middle piece of the picture, and then compress the piece to the specified size. Secondly, divide all values of the picture by 255 to normalize it to the range of 0 to 1. Finally, subtract the average value of the sample for each sample, so that the average value becomes 0. When returning, the data are connected line by line to become one-dimension data.

III. Linear Classifier using Bag-of-visual-Words Features(BOW)

The idea of Bag of word is original from texture recognition [1]. The basic mechanism of basic Bag of words is that we first compute the decriptors of image. Then,we scale those descriptors to histogram and find those clusters and frequence. Finally,we use different classifiers to make a classification. Generally speaking,there are three steps for implying Bag of words model. The first step is to extract image features and compute descriptors around each interest point. The second step is quantization, which means we use K-means algorithm to generate the visual vocabulary and build frequency histogram. Finally, we can use different classifiers to classify the BOW features.

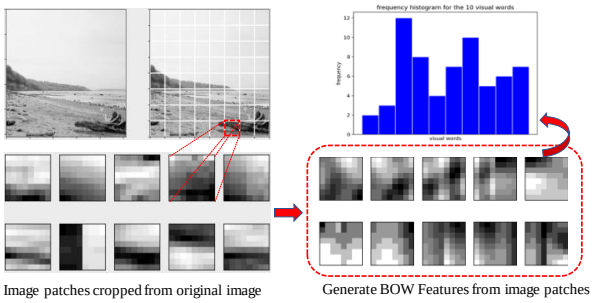

Fig. 2. Split the sample image into patches and extract BOW features

A. Feature extraction

To generate interesting point, optional choices contains SIFT, Harris, Dense etc. Here we use densely-sampled pixel patches. We split the image into 8x8 patches and each patch has 64 pixels. Then do mean-centring and L2 normalising for each patch. Finally, we sampled every 4 pixels in the x and y directions and flatten those vectors to implement feature extraction.

B. Quantization

After sampling all the training image into those feature, K-means is used to cluster features vectors to find the bag of visual words. Then we build frequency histogram for those visual words by vector quantisation. Ten bag of visual words and frequency histogram are displayed in Fig. 2.

C. Classification

The codebook for the visual words has been built, then we select the one-versus-rest Linear Support Vector Classifier(LVC) to classify the feature. We fit the histogram feature and its label into LVC and the classification confusion matrix is shown in Fig. 3.

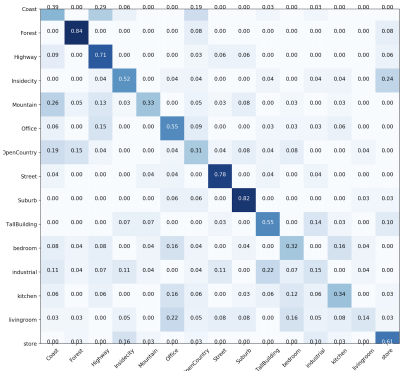

Fig. 3. Confusion matrix for 500 cluster

We test this model using different numbers of clusters. It turns out that the average prediction accuracy of this model is around 50%.

IV. Scene recognition using saliency information

The aforementioned approaches are using hand-crafted features, which usually cannot express the images sufficiently. In recent years, with the development of deep learning, a large amount of deep model are used for scene recognition. Current works extract object information and other features from images and map them to the scene domain and make a prediction to different classes. However, according to the characteristics of human visual system (HVS), the salient regions in the image can also influence the scene classification results greatly. In this work, we use one fully trained model to provide saliency information and one to provide image scene features based on the provided dataset.

A. Network Structure Designing

The deep learning based approach usually need a large amount of training data to fit the model weight parameters. In this work, the amount of images in training set is 1500, which is much too small to train a deep neural network fully. Towards this problems, we introduce one classical solution which is fine tuning network to tackle the limited size of the provided training data.

Fine-tuning network has been widely regarded as an efficient approach in transfer learning especially to deal with small

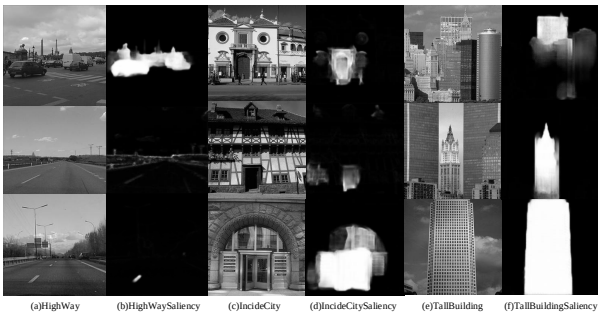

Fig. 4. Sample images and saliency maps from the provided training data

size data. The fundamental process of Fine-tuning is that at the very first, we train a deep model on a large scale of dataset, which contains a lot of training samples. Then, we treat the pretrained model as a prior knowledge and feature extractor. When we need to deal with a new visual task in the same domain, we can use the related pretrained model to extract sufficient, efficient and correct image features and fine tuning the network model on our new vision mission and solve the new problems. Since we do not need a large amount of training data to help the model to adjust the image features, we can save both money and time without building up a new large scale dataset. Furthermore, the fact proves that training a fine tuning neural network is much easier than training a new model from a random initialized status. To enhance the model performance, we take the saliency information into consideration and use it as an auxiliary input.

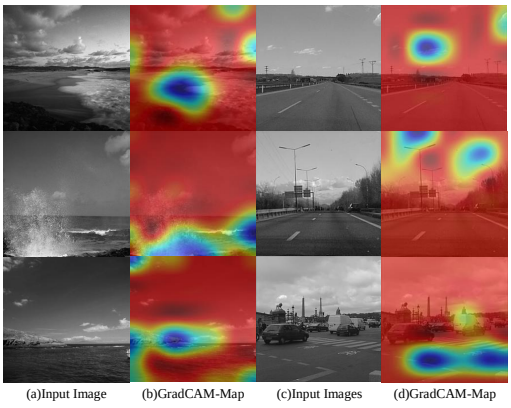

Fig. 5. Attention map generated through Grad-CAM from pretrained model[2]

1) Pretrained saliency map generation model selection: To extract saliency information and adopt it into our model, we choose to use salient object detection model which is proposed by Wang et al. [3] and trained on the Saliency in Context (SALICON) dataset [4]. We generate the saliency maps based on the provided training data and merge the features which are extracted from both the generated maps and original images and feed them into our proposed classification model to predict the image class. In Fig. 4, we illustrate a group of sample images and their related saliency maps, which are obtained from the saliency model.

2) Pretrained scene recognition model selection: The whole network contains two parts, generate saliency map and scene classification. To choose the pretrained model, we have tried the traditional InceptionV3 [5] and VGG16 [6] model pretrained on ImageNet [7] and fine tune the pretrained model on training data. However, the model exists overfitting problem and the performance on the validation data is poor. To solve the problems, we use the scene classification model pretrained on place365 [8], which is a large dataset contains 1.8 million images from 365 scene categories and this model can extract scene-aware image features.

To observe the model performance on our training data, we use the Grad-CAM to generate the attention map of the input image. The attention map is shown in Fig. 5. It can be observed that the attention map shows the high interested region which are related to the prediction results, and we think the attention region are human-like, which means the model can be applied to our data. The performance of different pretrained models on place365 is shown in TABLE I, which is provided by Zhou et al in [8].

TABLE I
DIFFERENT MODEL PERFORMANCE ON PLACE365 [8]

Models On Place365	Validation Set on Place365		Test Set on Place365	
	Top-1 acc.	Top-5 acc.	Top-1 acc.	Top-5 acc.
Place365-AlexNet	53.17%	82.89%	53.31%	82.75%
Place365-GoogLeNet	53.63%	83.88%	53.59%	84.01%
Place365-VGG	**55.24%**	84.91%	**55.19%**	85.01%
Place365-ResNet	54.74%	**85.08%**	54.65%	**85.07%**

From TABLE I, it can be observed that both Top-1 accuracy of VGG16 model can outperform the others. Therefore, we choose the Place365-VGG16 model as our pretrained model[2] in the scene classification part.

B. Training Pipeline

The proposed model tructure is shown in Fig. 6, which contains two inputs and two parts. The first part is to generate image saliency map and the second part is to make a classificaiton. The first input is original image input, the second one is saliency map input. Therefore, to train the proposed model on our dataset, we need to prepare the image saliency map. First, we use the salient object detection model proposed by Wang et al.[3] to predict the image salient region and save the saliency information. Then, we load both saliency map and original image and feed the data into our proposed model.

In this work, the batchsize is set to 32 and the learning rate is rather low which is 0.0001, since we do not want the model learning performance change rapidly and cause model oscillation, which will make the training process unstable and changable. In addition, we reduce the learning rate during the training process to help the model reach the optimal solution smoothly. Finally, we use the categorical crossentropy

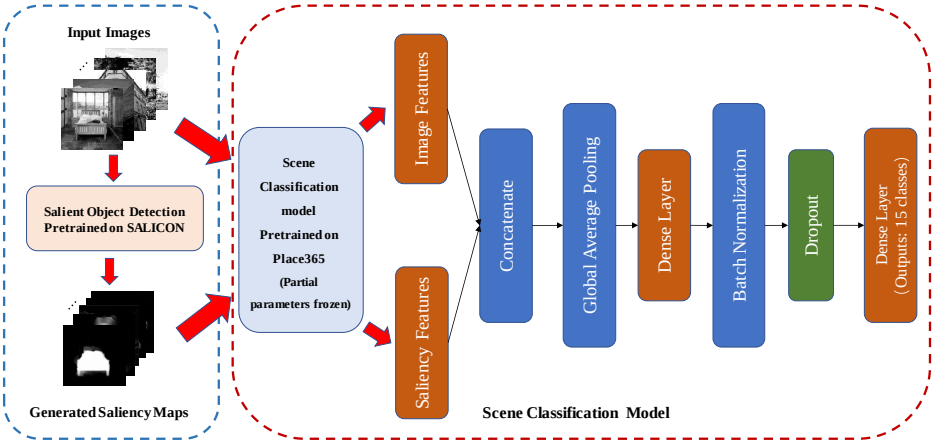

Fig. 6. Our proposed Saliency-Aware Scene Recognition (SASR) model

as our loss function since it is very efficient in multi-class classification problems.

C. Environments

For reference only, the proposed model is developed based on Keras 2.0+ and the backend is Tensorflow. The model is trained on work station, whose CPU is Intel Core i7-9750H, and the RAM is 16G. The type of GPU is NVIDIA GeForce RTX2060, with CUDA 10.1 with driver version 419.72.

D. Experiment results

Furthermore, since the training data has 15 classes and each class contains 100 images, we choose 10 images from each class folder and use them as our test data. Therefore, there is no inter-class bias when training the network. We use the Top-1 Accuracy and Top-5 Accuracy as our metric during the training process. The loss and accuracy are shown in Fig. 7.

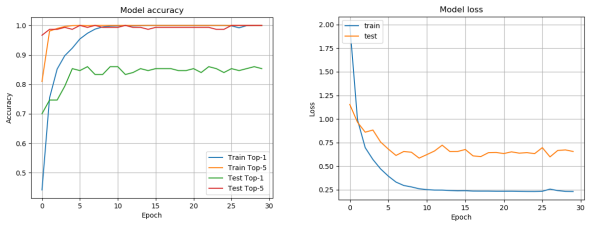

Fig. 7. Model categorical loss and accuracy during training process

From Fig. 7, it can be observed that the model exist an approate overfitting on our training data. However, the performance on our split test data is good as well. Therefore, the model has been trained successfully and the final Top-1 test accuracy on our test data is 86% and Top-5 accuracy is 100%, which can outperform the other classifiers aforementioned in this report and solve the scene classification problems effectively.

TABLE II
DIFFERENT APPROACHES ACCURACY ON TEST DATA

Classifier On Test data	Test Set	
	Top-1 acc.	Top-5 acc.
KNN	12.5%	-
LVCBOW	50.4%	-
Ours SASR	**86.1%**	**100%**

V. CONCLUSION

In this assignment, we implement three scene classifier, which include KNN, LVCBOW, and our proposed SASR model. The experiment result indiacate that our proposed model can outperform the others and give an acceptable prediction. According to TABLE. II, the Top-1 accuracy is 86.1% and Top-5 accuracy is 100%.

REFERENCES

[1] Y. Yang and S. Newsam, "Bag-of-visual-words and spatial extensions for land-use classification," in *Proceedings of the 18th SIGSPATIAL international conference on advances in geographic information systems*. ACM, 2010, pp. 270–279.

[2] G. Kalliatakis, "Keras-vgg16-places365," 2017.

[3] W. Wang, J. Shen, X. Dong, and A. Borji, "Salient object detection driven by fixation prediction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1711–1720.

[4] M. Jiang, S. Huang, J. Duan, and Q. Zhao, "Salicon: Saliency in context," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[5] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[6] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. IEEE, 2009, pp. 248–255.

[8] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 million image database for scene recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 40, no. 6, pp. 1452–1464, 2017.