Scene Recognition

Achyut Karnani (ak19g21) Samyuktha Babuganesh (sb9n22) Pooja Vijayakumar (pv1n21)

Tunchanok Ngamsaowaros (tn1u22)

Abstract

Scene recognition is a problem in computer vision where a label is assigned to an image based on features extracted from that image. This paper focuses on scene recognition using 3 different techniques. The accuracy of each classifier on the training and validation set will be evaluated and compared.

1. Introduction

Extracting features from an image can be done in many ways, namely cropping an image, concatenating it to a vector and passing it to a classifier. There are also feature encoding techniques such as SIFT to create vocabularies for image data. Moreover, Bag-of-visual-words can be used to extract features from fixed-size, densely-sampled pixel patches of images using K-Means to learn a vocabulary. Different classifiers can be used to determine the quality of information in the extracted features by calculating the accuracy on the training and validation sets. The training data used has 100 grey-scale images for each of the 15 classes. The test data consists of 2985 grey-scale images.

2. Run-1

In this run, the images were cropped from the centre and resized into tiny 16 x 16 images. The images were standardized to have zero mean and unit standard deviation. Then, the images were flattened into a single-dimensional vector.

The data of the vectors along with the labels will be split into the training and validation set, with 10% of the data in the validation set. Then, K-Nearest Neighbors classifier was trained on the data. The number of neighbours was set to 100. The algorithm will determine the class label of data by its nearest K value. The algorithm was evaluated on the validation set and the accuracy was recorded.

3. Run-2

In this run, a set of 15 linear classifiers were designed for each class respectively, which performed classification based on images represented using a Bag of Visual Words Model.

The Bag of Visual words is a technique to represent an image as a collection of different features. For this run, the bag of visual words was formed using the pixel intensities of patches of images. For each image, 8x8 pixel patches with a step size of 4 pixels were extracted to form the total feature set. 25% of this feature set was used to train the K-means algorithm with default parameters. Hence, groups with similar local features are formed. These groups are called code-book or visual vocabulary. The vocabulary was constructed by experimenting with cluster sizes of 250, 500, and 600 for this algorithm.

The images are then represented using a histogram of the visual words i.e. the frequency of occurrence of each visual word is counted. And this frequency distribution is the representation of a particular image. To identify each particular label, 15 logistic regression classifiers were trained on the frequency distribution data. To train these classifiers, 100 images of the particular class and 100 images of the mixture of non-class (the remaining 14 classes) were used. The classifier with the highest prediction probability for the class was chosen as the predicted label.

4. Run-3

In this run, the key points and feature descriptors were detected for the images. Using the Bag of Visual Words (BoVW), all the images were converted into feature vectors, which were then fed into various Machine Learning Algorithms for scene recognition.

4.1. Feature Detection

In computer vision, a feature is any piece of information that can distinguish one object from another. L. Fei-Fei and P. Perona [6], performed a comparative analysis for scene categorisation and showed that "Compared to the global features, local regions are more robust to occlusions and spatial variations" [6] and thus, local features were chosen for implementation. There are 2 major components of feature detection, namely: Key point detection and Key point

descriptor.

- **Key point detection**: Detection of key points is the most crucial part of feature detection. These points need to be well-defined and invariant to scale, orientation and illumination. Moreover, the quantity of key points is important for scene recognition, so dense key points were detected.
- **Feature Descriptors**: A Global descriptor is not very robust to changes made in smaller parts of an image, so local descriptors are used. The local feature descriptors provide information regarding the neighbourhood of the identified key points. The following feature descriptors were used: SIFT, BRIEF and ORB.

4.1.1 Dense Scale Invariant Feature Transform

Lowe, D.G [1], proposed a method to extract features that "are invariant to image scaling and rotation, and partially invariant to change in illumination and 3D camera viewpoint" [1]. These features are widely known as Scale Invariant Feature Transform (SIFT) and use the Difference of Gaussian (DoG) approach, using different magnitudes of blur and image scales, to identify key points that are scale and orientation invariant.

C. Liu, et al. [2] proposed a novel approach known as SIFT flow for scene alignment based on the concept of optical flow, where the SIFT descriptors are extracted for each pixel in an image. Choosing all the pixel values in an image will reduce the computational speed, thus, the key points were chosen using a step size of 5 along the rows and columns of the image. Once the key points have been identified, the descriptor for each point is computed. The area surrounding the key points is divided into 16x16 regions and further subdivided into 4x4 regions. For each of those sub-regions, the gradients are computed with respect to the orientation. Thus, a 128-dimensional (4x4x8) feature descriptor is generated for each key point in an image.

4.1.2 Dense Scale Invariant Feature Transform using Spatial Pooling

Lazebnik, S. et al. [4], proposed an approach that uses Global GIST and local SIFT feature descriptors. In this approach, each image is divided into smaller sub-regions at each increasing level based on the resolution. In this way, a Spatial Pyramid with 2 levels is constructed. At each level, Dense key points in the image and its sub-regions are detected, with a step size of 5.

The descriptors around each key point are computed using the 8-bin orientation histograms, as used in SIFT. At each level, the descriptors are stacked and the histograms are normalised by the weight of the features in the image. "The resulting 'spatial pyramid' is a simple and computationally efficient extension of an orderless bag-of-features image representation, and it shows significantly improved performance on challenging scene categorization tasks" [4].

4.1.3 Binary Robust Independent Elementary Features (BRIEF)

Calonder, et al. [3], proposed a novel approach to compute the feature descriptors, by representing them using binary values. "Not only is construction and matching for this descriptor much faster than for other state-of-the-art ones, but it also tends to yield higher recognition rates, as long as invariance to large in-plane rotations is not a requirement" [3]. We experimented with 2 different approaches to key point detection, namely: Dense key points and Dense Spatial Pooling.

The binary feature descriptors are computed for each patch in an image, based on a comparison of the intensity values between different pixels, located inside the patch. This test assigns a value of 1 if the intensity of the first pixel is less than the second pixel, 0 otherwise.

The key point feature descriptors generated by BRIEF can be of the dimension 128, 256 or 512 bits. In our approach, we have chosen the dimensionality of the feature descriptors to be 256 bits. Moreover, each patch in an image needs to be smoothed using Gaussian Smoothing, due to the high sensitivity to noise.

4.1.4 Oriented FAST and Rotated BRIEF (ORB)

E. Rublee, et al. [5], proposed an approach for feature extraction that uses FAST to detect key points and BRIEF to find descriptors. Both FAST and BRIEF perform poorly with orientation and rotation respectively with FAST having no orientation component. The authors used a method of measuring the corner of orientation by intensity centroid as FAST does not measure the cornerness of a patch. For this project, the ORB algorithm was first used to detect key points but the algorithm appears to not detect key points in some images. Hence, the Dense key points approach was used for key point detection.

The authors used rBRIEF method to maintain BRIEF's effect of producing large variance and mean close to 0.5 for each bit of feature. rBRIEF has enhanced results in variance and correlation over another method, steered BRIEF. For this project, Dense key points in the image and its sub-regions are detected, with a step size of 5. Feature descriptors for the pixel values have been computed using ORB with default parameters. The authors claimed the ORB feature encoding method is 2 orders of magnitude faster than SIFT but performs just as good in multiple scenarios [5].

4.2. Bag of Visual Words (BoVW)

Csurka, Gabriela & Dance, et al. [7], proposed an approach known as bag of key points, that allows images to be

represented compactly, by converting the feature descriptors into feature vectors. The main advantage is that each image is now converted into a histogram, with a count of the number of times a visual word appears in that image. These feature vectors are then fed into Machine Leaning models, to perform multi-class classification tasks. The main steps are as follows: Feature Descriptor, Visual Vocabulary, Bag of key points and Classification.

4.3. Feature Descriptor

The task of key point identification and description has been explained in detail in section 4.1.

4.4. Visual Vocabulary

The Visual Vocabulary for a set of images is computed using the mean feature descriptors of several similar descriptors. This allows similar feature descriptors to be grouped together and this is computed using K-Means clustering algorithm. Each visual word is obtained from the centroid of each of the clusters. We have experimented on cluster sizes of 50, 200 and 250 and the performance was recorded and reported in section 5.

4.5. Bag of key points

Each image in the dataset will now be mapped to a Visual Vocabulary, which is a histogram representing the number of times a visual word appears in an image. This reduces the dimensionality of the images significantly and we can now represent images as histograms rather than pixels.

4.6. Classification

We have experimented with 3 different classifiers, namely: Support Vector Machine, Naive Bayes and Random Forest.

Support Vector Machine (SVM) is a classifier that fits the hyperplane that maximises the margins between the data points. V. N. Mandhala, et al. [8], found that SVM with the Gaussian kernel resulted in the best test accuracy of 72.03%. Thus, SVM was chosen for this project with the Gaussian kernel as one of the classifiers.

Naive Bayes algorithm is based the on Bayes' theorem and it calculates the probability of classes given the input and selects the class with the highest probability. It makes predictions based on the assumption that one class is independent of another. Rafique A, et al. [9], achieved an accuracy of 85.09% while performing scene recognition for UIUC data-set using Naive Bayes. Thus, Naive Bayes is another classifier used in this project.

Random forest is an ensemble of decision trees, which aggregates the results of large number of decision trees and gives predictions. It is used for both regression and classification tasks. Each tree is made from a random subset of the dataset and a random feature set. Inspired by Kulkarni, et al. [11] who achieved an accuracy of 96% predicting landscapes, we also experimented with a random forest classifier.

5. Performance

Run-1 The Validation accuracy for Run 1 is 20%.
Run-2 The performance for Run 2, when experimented with different cluster sizes, is shown in the table 1.

Cluster Size	Validation Accuracy
250	48%
500	**50%**
600	44%

Table 1

Run-3 The performance of the 3 classifiers using different sets of features and cluster sizes was recorded. The Validation accuracy for cluster sizes of 250, 200 and 50 are shown in tables 2, 3 and 4 respectively.

Vocabulary size: 250, Step size: 5

Feature encoding	SVM	Naive Bayes	Random Forest
Dense SIFT	69.67%	61%	66.33%
Dense SIFT with spatial pooling	**72.67%**	63%	69.67%
Dense ORB	58%	53.33%	50.67%
Dense ORB with spatial pooling	59.33%	52%	51.33%
Dense BRIEF	54%	48%	49%
Dense BRIEF with spatial pooling	58.33%	49.33%	49.67%

Table 2

Vocabulary size: 200, Step size: 5

Feature encoding	SVM	Naive Bayes	Random Forest
Dense SIFT	71.33%	61%	68.67%
Dense SIFT with spatial pooling	**73.33**%	59.33%	68.67%
Dense ORB	55%	51.67%	53%
Dense ORB with spatial pooling	57.33%	51.33%	52.67%
Dense BRIEF	53%	47.33%	46.67%
Dense BRIEF with spatial pooling	58%	47%	50.67%

Table 3

Vocabulary size: 50, Step size: 5

Feature encoding	SVM	Naive Bayes	Random Forest
Dense SIFT	68%	63.33%	65.33%
Dense SIFT with spatial pooling	**71%**	64.33%	67.67%
Dense ORB	55.67%	47%	51.67%
Dense ORB with spatial pooling	54.67%	51%	50.67%
Dense BRIEF	50%	42.67%	47%
Dense BRIEF with spatial pooling	54%	42.67%	51.33%

Table 4

"The vocabulary used should be large enough to distinguish relevant changes in image parts, but not so large as to distinguish irrelevant variations such as noise" [7]. Therefore, a cluster size of 50 was experimented with (as shown in table 4) but not used to generate the results on the test set. Lazebnik, S. et al. [4], also showed in their results for scene recognition, that increasing the cluster size from 200 to 400 does not show any significant improvement in the results. Thus, a cluster size of 200 was chosen.

Dense SIFT and Dense SIFT with spatial pooling, along with Support Vector Machine provide the best overall results on the Validation set, as shown in table 3. BRIEF and ORB feature descriptors are very fast in terms of computational speed when compared to SIFT, and are mostly used for real-time applications. However, "BRIEF is not designed to be rotationally invariant, nevertheless, it tolerates small amounts of rotation" [3]. Tareen, S. A. K., et al. [10], performed a comparative analysis of the feature descriptors and also showed that ORB was the least scale invariant and "SIFT is concluded as the most accurate algorithm" [10]. This is reflected in the results shown in table 3 and thus, Dense SIFT with spatial pooling is reported to have the best overall accuracy in terms of the validation set.

6. Conclusion

In conclusion, incorporating different feature encoding techniques results in better classification performance when the features were input into machine learning classifiers. This is evident from the better results in accuracy produced in run 3 compared to run 1 and run 2, which do not make use of feature encoding techniques. Better feature extraction techniques, such as convolution neural networks (CNN), can be used to produce higher accuracy in scene recognition. F. Hu, et al. [12] used features extracted from CNN layers called Dense CNN features to produce higher classification accuracy compared to using SIFT features.

References

[1] Lowe, D.G. Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision 60, 91–110 (2004). https://doi.org/10.1023/B:VISI.0000029664.99615.94

[2] C. Liu, J. Yuen and A. Torralba, "SIFT Flow: Dense Correspondence across Scenes and Its Applications," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 33, no. 5, pp. 978-994, May 2011, doi: 10.1109/TPAMI.2010.147.

[3] Calonder, Michael & Lepetit, Vincent & Strecha, Christoph & Fua, Pascal. (2010). BRIEF: Binary Robust Independent Elementary Features. Eur. Conf. Comput. Vis.. 6314. 778-792. 10.1007/978-3-642-15561-1_56.

[4] Lazebnik, S. et al. (2006) "2006 Ieee Computer Society Conference on Computer Vision and Pattern Recognition (cvpr'06)," in Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories. IEEE, pp. 2169–2178. doi: 10.1109/CVPR.2006.68.

[5] E. Rublee, V. Rabaud, K. Konolige and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," 2011 International Conference on Computer Vision, Barcelona, Spain, 2011, pp. 2564-2571, doi: 10.1109/ICCV.2011.6126544.

[6] L. Fei-Fei and P. Perona, "A Bayesian hierarchical model for learning natural scene categories," 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 2005, pp. 524-531 vol. 2, doi: 10.1109/CVPR.2005.16.

[7] Csurka, Gabriela & Dance, Christopher & Fan, Lixin & Willamowski, Jutta & Bray, Cédric. (2004). Visual categorization with bags of keypoints. Work Stat Learn Comput Vision, ECCV. Vol. 1.

[8] V. N. Mandhala, V. Sujatha and B. R. Devi, "Scene classification using support vector machines," 2014 IEEE International Conference on Advanced Communications, Control and Computing Technologies, Ramanathapuram, India, 2014, pp. 1807-1810, doi: 10.1109/ICACCCT.2014.7019421.

[9] Rafique, Adnan & Jalal, Ahmad & Ahmed, Abrar. (2019). Scene Understanding and Recognition: Statistical Segmented Model using Geometrical Features and Gaussian Naïve Bayes. 10.1109/ICAEM.2019.8853721.

[10] S. A. K. Tareen and Z. Saleem, "A comparative analysis of SIFT, SURF, KAZE, AKAZE, ORB, and BRISK," 2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET), Sukkur, Pakistan, 2018, pp. 1-10, doi: 10.1109/ICOMET.2018.8346440.

[11] Kulkarni, Arun D. and Lowe, Barrett, "Random Forest Algorithm for Land Cover Classification" (2016). Computer Science Faculty Publications and Presentations. Paper 1.

[12] F. Hu, G.-S. Xia, J. Hu, and L. Zhang, "Transferring Deep Convolutional Neural Networks for the Scene Classification of High-Resolution Remote Sensing Imagery," Remote Sensing, vol. 7, no. 11, pp. 14680–14707, Nov. 2015, doi: 10.3390/rs71114680.