Implmentation & Evaluation of A. Bosh's Hybird Scene Classification Approach using the Places2 dataset

Geoffrey Harper
Faculty of Engineering Univerisy of Vicoria
University of Victoria
Victoria, Canada

Andreas Anglin
Faculty of Engineering Univeristy of Victoria
Univeristy of Victoria
Victoria, Canda

Abstract

An implementation of the scene classification approach designed by Bosch is explored [6]. This implementation was developed using Python in conjunction with numpy, cv2, and various other libraries. The dataset chosen was a subset of the MIT Places dataset [7]. Hyperparameter tuning was required due to time constraints. Small deviations from the proposed implementation were taken such as: the choice of descriptor and the use of Non-negative Matrix Refactoring instead of a Probabilistic Latent Semantic Analysis model. Classification performance, measured in average precision, was found to be 30% lesser on average than the results reported in the proposed implementation. Project code, which produced the results in this report, is available on Github [9].

I. Introduction

This implementation of Bosch's algorithm is an attempt at measuring its performance on a novel dataset. Image classification is an important area of research, especially in the area of remote sensing and image database organization. As these areas have increased in popularity due to the power and number of cameras, so has the number of images available for processing making computer vision algorithms – which needed large amounts of to be successful – more feasible [1].

However, one of the major issues with classification is dealing with properly identifying and labelling images given the numerous possible scenes in a real world setting. This is mainly due to the enormous amount of different images and scenes that are possible; as well as infinite range of illumination and scale conditions that could apply. Another difficulty is being able to distinguish between intra-class variations such as, the difference between a swimming pool indoors or a swimming pool outdoors. Asides from these two issues there is also the problem of image ambiguity. When images have parallelism, symmetry, or common regions within a set of images there can be an ambiguity when classifying. For example being able to distinguish a hilly countryside could be confused with mountains given certain perspectives.

Bosh's algorithm specifically attempts to tackle this issue by making the classification more generalizable by using a probabilistic layer to calculate the likelihood of the topic given the set of test images. These topic probabilities are then used to train a discriminative classifier (KNN or SVM). Classification of an unseen image occurs by computing an array describing the relationship between the image and a set of latent variables, which is fed to the discriminative classifier [6].

II. Literary Review

There are a few papers that Bosh took inspiration from that attempted to tackle this problem. Notable ones are: Fei-Fei's "A Bayesian Hierarchical Model for Learning Natural Scene Categories" [2], A. Oliva's "Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope" [3]; two papers which attempted to use intermediate representation of an image before classifying – the strategy that Bosh's paper uses — however; the number of categories they used was only up to 15. As well, Bosh's algorithm is an expanded version of A. Bosch's "Scene Classification via pLSA" [4]. Lastly, a notable CONVNet paper that attempts to tackle this problem is B Zhou's "Learning Deep Features For Scene Recognition using Places Database" [5] —, which at this time (April 22nd, 2020) has 2277 citations making it an even more prominent article than Bosch's paper which only has 1014 citations at this timestamp.

Fei-Fie's bayesian hierarchical approach attempts to solve the problem categorizing the image "theme" without the use of hand-annotated images for training. Thus, they would be able to classify and recognize an image without having to first recognize objects that are present. The issue that arose from their system is that it struggled to perform for "the indoor scenes" which seemed to suggest it needed a "richer set of features" [2]. However, Bosh's algorithm does not struggle with this specific problem. This seems to be due to dense feature patch extractions methods that were applied [6].

A. Oliva's proposed scene recognition model attempts to bypass the segmentation and processing of individual "objects or regions" by creating a "low dimensional Spatial envelope" which describes the images

based on the degree of 8 predefined properties [3]. Although the paper at first appears to be successful it is restricted to a predefined set of annotated categories (which consists of naturalness, roughness, ect.). It was shown in Bosh's paper that just the pLSA model they implemented produced better results than a spatial technique and did not require image annotation thus making the solution more generalizable than Oliva's [6].

Bosch's former paper on the comparisons of different types of scene classification methods also found that the Bag-of-words pLSA model was the most accurate modeling type in comparison with Low-level image representation, low-level block representation and image segmentation by classifying "present objects" [4]. However, the accuracy of the pLSA model was worse than the classifying accuracy of Bosh's new model in [6]. The increase in accuracy is likely due to two different factors. One, the dense feature extractions in more thorough and dense in [6]; and two, Bosh found, through experimentation, that a K-means classifier was more effective than the SVM classifier used in [4].

Finally there is a popular convnet approach to this problem proposed by B. Zhou. The premise around Zhou's paper is that CNNs need massive amounts of data to be effective, thus by using the MIT Places database with predefined MIT Places-CNN architecture one would be able to have an effective and powerful scene classifier. However, one drawback that this approach has asides from the enormous amount of data that is it is not efficient, nor lightweight. As stated in [5], just properly training took 6 days to complete. As well, this approach to the problem does not appear to be as flexible — an important aspect of having a generalizable algorithm that can be used in a variety of ways. For example, in [6] Bosh's discusses the success of classifying individual film frames into scenes for a movie. In the example application they trained only on the movie *"Pretty Women"*. This is something Zhou's approach would fail, as there would be not enough data to properly tune the algorithm.

The following report will cover the proposed implementation of the algorithm, the evaluation of the dataset used, and finally how the hyper parameters were tuned (including their results).

III. Proposed Approach

Bosch's proposed algorithm is complex; the definition in Figure 1 can help aid understanding [6]. It starts by normalizing image intensities for zero mean and unit standard deviation. Unfortunately, the CV2 library containing the SIFT functionality for our implementation did not accept float intensity values. Therefore, this image normalization could not be done. Next, both dense and sparse descriptors are collected from images in the dataset. Descriptors include: colour patches of different sizes, Gary Harris Affine sparse, dense colour SIFT with 4 concentric circles, and dense gray SIFT with n concentric circles. Our implementation utilizes dense gray SIFT descriptors due to the high level of performance shown in the results of the proposed algorithm [6]. A library housing SIFT with concentric circles could not be located and therefore our descriptors are merely gray SIFT descriptors. Dense gray SIFT descriptors are collected with the following spacings: 5 pixels, 10 pixels, and 15 pixels. The optimal descriptor spacing, proposed by Bosch, was found to be 10 and this is the spacing utilized in our implementation.

The training set of descriptors is then clustered via Lloyd's algorithm into visual words. The optimal number of centroids found in the proposed algorithm was 1500. This number was too resource intensive to implement and therefore parameter tuning was done to find an optimal value in a lesser range. Visual word frequency histograms are created for each image after their corresponding descriptors are vectorized via clustering.

Visual word frequency histograms are then normalized for combatting variation in image size and thus variation in the number of visual words associated with each image. Visual word frequency histograms, from the training set, are passed to a probabilistic Latent Semantic Analysis (pLSA) model for the creation of a document (image) / topic (latent variable) relationship matrix. Our implementation harbours a Non-negative Matrix Factoring (NMF) model to create the document (image) / topic (latent variable) relationship matrix.

This NMF model utilizes Kullback-Leibler divergence as its beta loss function resulting in a matrix refactoring similar if not identical to pLSA [8]. Finally, this training document/topic relationship matrix and the labels corresponding to each array in this matrix are passed to the discriminative classifier for training. The proposed algorithm utilizes both Support Vector Machines (SVM) and K-Nearest Neighbour (KNN) models. Results from the proposed algorithm show the performance difference between these two classifiers is negligible.

Therefore, our implementation uses a KNN model as the discriminative classifier. Test images undergo a similar process. However, the different models employed by this algorithm, namely Lloyd's algorithm, NMF, and SVM/KNN, are fitted from the data generated from the training set. No further training is required and therefore the data generated by the test set can merely be run through each model in turn for each example's eventual classification. All data normalizations and steps are present in both the training and test steps. The deviations from the proposed algorithm in our implementation are present in both testing and training stages.

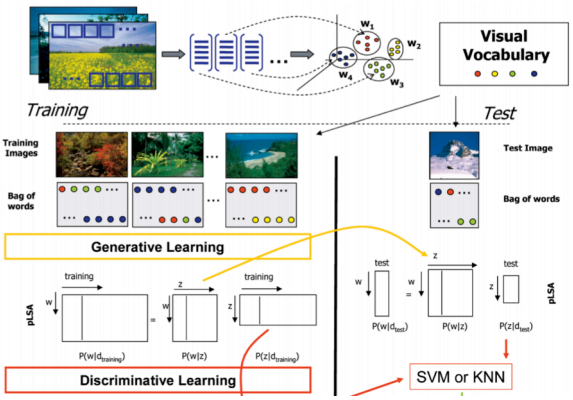

Fig. 1. Depiction of Bosch's proposed Scene Classification Algorithm.

IV. Evaluation & Datasets

A. Datasets

Our dataset is a subset of MIT's Places365 dataset [7]. The subset was chosen to mimic the dataset highlighted in the results of Bosch's proposed implementation [3], [6]. This subset contains 8 image categories. 4 of them are 'manmade' scenes, namely, 'hostel', 'stadium', 'catacomb', and 'alley'. The other 4 are 'natural' scenes, namely, 'wheatfield', 'volcano', 'swamp', and 'valley'. Similarly to the results in the proposed dataset, we compare the results of the entire set against the subsets of 'manmade' and 'natural' scenes. Each category contains 350 images of size of 256x256.

B. Parameter Tuning

Due to a lack of computational resources, the optimal parameters from the proposed algorithm cannot be used. Therefore, parameter tuning is required. Parameter tuning is done on a tuning subset of the entire dataset. This tuning subset contains 100 images for each of the 8 categories; as opposed to the dataset's entirety, 350 images in each category.

This subset is used to improve the efficiency of the parameter tuning process. Three parameters require tuning: number of visual words (V), number of topics (Z), and the number of neighbours (K) for the KNN classifier. Initially, the number of visual words will vary from 40-120; the number of topics will vary from 2-30. During this stage of tuning, K (the number of neighbours) will be set to the optimal value found in the proposed implementation, 11.

The V and Z pairing with the optimal average precision will be brought forward to tune the number of nearest neighbours, ranging from 1-20. These ranges are defined by our resource limitations and the optimal parameters found in the proposed implementation. Performance criteria are measured in average precision for consistency to the proposed implementation.

Figure 2 shows the results of the initial iteration. Please note that although the Z label is 'Accuracy' it is plotting the average precision. V=80, Z=26 was found to be the optimal pairing. Figure 3 shows the results of the second iteration. V=84, Z=26 was found to be the optimal pairing.

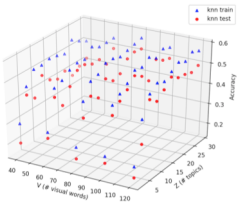

Fig. 2.

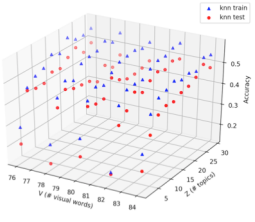

Fig 3.

Figure 4 shows the results of the third iteration. V=83, Z=29 was found to be the final optimal pairing. Figure 5 shows the results of K tuning for V=83, Z=29. K=26 was found to be the optimal value

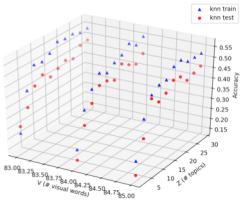

Fig. 4.

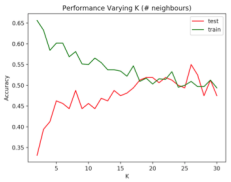

Fig. 5.

Figure 6 compares the average precision of our implementation on the three datasets. Figure 7 shows results of the proposed implementation when comparing results from their three datasets [6].

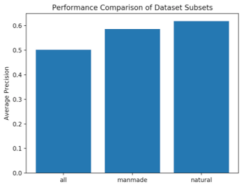

Fig. 6.

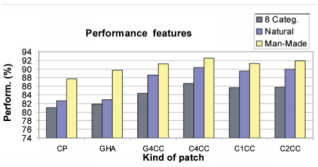

Fig. 7.

Our implementation's descriptors, dense gray SIFT, are most similar to 'G4CC' of the proposed implementation, dense gray SIFT descriptors with 4 concentric circles. Unlike the results from the proposed implementation, the natural data subset performed best. However, all the results from the proposed implementation are superior by an average margin of 30.5%.

The discrepancy in performance between our and the proposed implementation is most likely due to the drastic difference in the number of visual words. This is the largest deviation between the two implementations. The optimal number of visual words was found to be 1500 in the proposed implementation and our optimal number of visual words was only 83.

Intra-class variation is the most complex part of scene classification. Images within a class of scenes have extreme variation to the point of ambiguity. The ambiguity is illustrated in Figure 8 and Figure 9 [7]. These images were taken from the classes' volcano and valley, respectively.

Both images could possibly reside in either classes; an unfortunate side effect of defining images by scenes. Analysis of the confusion matrix resulting from the 'natural' dataset shows that the average precision of 'valley' and 'volcano' was 0.52. The majority of incorrect predictions for examples of these classes were the result of incorrect guesses of 'valley' and 'volcano'. For comparison, the other two natural classes, 'wheatfield', and 'swamp', had an average precision of 0.715.

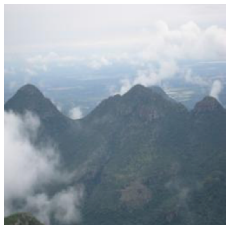

Fig. 8.

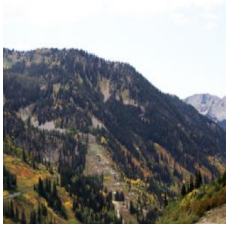

Fig. 9.

The clustering of image descriptors is the backbone of this algorithm. Without a larger visual vocabulary the visual word assignments created by Lloyd's algorithm are too ambiguous for a problem as complex as scene classification. Figure 10 shows the results of the proposed implementation as the visual vocabulary decreases [6]. Performance of the proposed implementation maintains itself well as the visual vocabulary decreases. Unfortunately, data within the range of our parameter tuning of V (40-120) is not included.

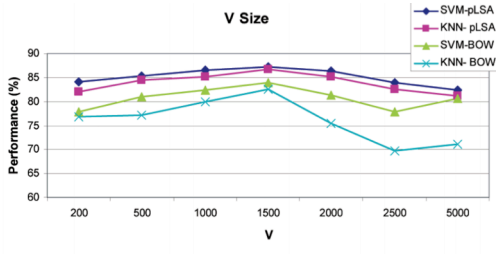

Fig. 10.

Our other deviations could also contribute to the lesser performance. The inability to perform zero mean, unit standard deviation image pre-processing on our dataset was unfortunate. Additionally, the difference between our descriptors (dense gray SIFT) and those of the proposed implementation (dense gray SIFT with 4 concentric circle supports) was also a major deviation, which cannot be ignored. It is difficult to speculate how these differences affected performance on an algorithm as complex as this.

V. Conclusion

The problem domain of scene classification presents a unique set of challenges such as intra-class variation and scene ambiguity. Scene classification is an important and continually growing field particularly in areas of remote sensing and image database organization [1]. The approach to this problem to domain was to implement a variation of the algorithm proposed by Bosch [6].

The architecture of the algorithm was very similar to Bosch's; however, dense gray SIFT descriptors were used instead of dense gray SIFT descriptors with concentric circles. In addition, a NMF model utilizing Kullback-Leibler divergence is used in place of pLSA. Due to processing power limitations and time constraints the optimal number of visual words was concluded to be 83 instead of the proposed 1500. Our implementation was trained on the Places365 dataset using 'hostel', 'stadium', 'catacomb', 'alley', 'wheatfield', 'volcano', 'swamp', and 'valley' as the scenes for testing and training.

It was interesting that average precision of the algorithm was approximately 0.5 (which seems to be decent considering the algorithm was not running on the most optimal parameters due to processing power limitations) with the natural scenes having the highest precision of 0.6 and the man made scenes of approximately 0.57.

Our implementation presented similar issues to Bosch's as it struggled to accurately identify man-made scenes such as stadiums and hostels. Our results – similar to Bosh's – show struggles with ambiguous scenes. The proposed implementation describes scene ambiguity issues with "open country" and "coast" [6].

Similarly, our results depict struggles with 'volcano' and 'valley' — scoring an average precision of 0.52. These issues are consistent across both the proposed implementation and our own. This suggests that the problem domain of scene classification has fundamental flaws stemming from human errors via the determination of scenes and the assignment of images to these scenes.

References

1] G.Cheng, *"Remote Sensing Image Scene Classification: benchmark and State of the Art"*,

[2] Fie-Fie, *"A Bayesian Hierarchical Model for Learning Natural Scene Categories," Proc. IEEE CS Conf. Computer Vision Pattern Recognition, pp. 523-531, 2005*

[3] A. Olivia, *"Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope," Int'l J. Computer vision, vol 42, no. 3, pp. 145-175,2001.*

[4] A. Bosch, *"A Review: Which Is the Best Way to Organize/Classify Images by Content," Image and Vision Computing, vol 25, no. 6, pp. 778-791, June 2007.*

[5] B. Zhou, *"Learning Deep Features for Scene Recognition using Places Database," Advances in Neural Information Processing Systems 27 (NIPS), 2014*

[6] A. Bosch, *"Scene Classification Using a Hybrid Generative/Discriminative Approach," Proc. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 30, No. 4, April 2008*

[7] A. Khosla, *"A Large-Scale Database for Scene Understanding,"* Places2. [Online]. Available: http://places2.csail.mit.edu/download.html. [Accessed: 27-Apr-2020].

[8] E. Gaussier, *"Relation between PLSA and NMF and implications,"* [Online]. Available: https://www.researchgate.net/publication/221301249_Relation_between_PLSA_and_NMF_and_implications

[9] andre3racks, "andre3racks/scene-classification," GitHub, 27-Apr-2020. [Online]. Available: https://github.com/andre3racks/scene-classification. [Accessed: 29-Apr-2020].