

Fast Discovery of Discriminative Mid-level Patches

Angran Lin, Xuhui Jia and Kowk Ping Chan

Department of Computer Science, The University of Hong Kong, Hong Kong

Keywords: Discriminative Mid-level Patches, Fast Discovery, Fast Exemplar Clustering.

Abstract: Learning discriminative mid-level patches has gained popularity in recent years since they can be applied to various computer vision topics and achieve better performance. However, state-of-the-art learning methods require a lot of training time, especially when the problem scale becomes much larger. In this paper we propose a simple but fast and effective way, the Fast Exemplar Clustering(FEC), to mine discriminative mid-level patches with only class labels provided. We verified our results on the task of scene classification and it took us only one day to train the model on the MIT Indoor 67 dataset using an Core i5 quad-core computer with Matlab. The results of our experiments revealed that the mid-level patches discovered by our method were semantically meaningful and achieved competitive accuracy compared to the state-of-the-art techniques. In addition, we created a new scene classification dataset named Outdoor Sight 20 which contains outdoor views of 20 famous tourist attractions to test our model.

1 INTRODUCTION

In the last few years, discovery of discriminative mid-level patches has become increasingly popular in computer vision. It has been successfully applied to problems like object detection (Li et al., 2013) (Rios-Cabrera and Tuytelaars, 2013), scene classification (Juneja et al., 2013) (Sun and Ponce, 2013), motion detection (Wang et al., 2013) and video classification (Jain et al., 2013) (Tang et al., 2013). Its popularity can be attributed to part classifiers' representativeness: types of patches that occur frequently in the visual world, and discriminativeness: types of patches that are different enough from other types. In particular, (Doersch et al., 2012) and (Felzenszwalb et al., 2010) have shown the benefits of desired patches acting as visual words. (Mittelman et al., 2013)

Recent works have focused on learning discriminative mid-level patches via a max-margin framework including variants of SVMs like exemplar SVM (Juneja et al., 2013) and miSVM (Li et al., 2013). To achieve broad coverage and better purity, thousands of training rounds are required. Moreover, in each round the classifiers/detectors are learned in an iterative manner. Thus, the computational complexity of using a standard procedure that involves hard negative mining for a huge amount of classifiers would be surprisingly high. It leaves us a major challenge: a simple, efficient and effective method is yet to be found.

In this paper, we proposed a fast algorithm

to discover discriminative mid-level patches. To achieve this, we developed the Fast Exemplar Clustering(FEC) based on the idea of k-means algorithm. FEC works extremely fast with dramatically reduced computation. As a comparison, the MIT 67 indoor scene classification problem in section 5 spent us only one day to train on an ordinary Core-i5 computer, while the commonly used methods today would take several days on a cluster (Singh et al., 2012).

The reason why FEC is so efficient is that, firstly it only requires spatial information of feature vectors and classifiers are trained using their distance measure rather than iteratively solving a time consuming optimization problem. Secondly our method uses only local information instead of global information. When the number of patches increases, the training time of SVM based methods for each round may increase sharply while for our method the time consumption will rise slowly in an $O(\log N)$ manner with the help of data structures like R-tree.

The biggest challenge of FEC is the risk of overfitting. However, we managed to solve it by using a properly designed evaluation function described in section 3.3 together with a large validation set. Our experiments showed that the patches discovered by FEC were both discriminative and representative. In summary, the contributions of this paper are:

1. A novel algorithm for efficiently and effectively detecting discriminative image parts is developed, which demonstrated promising performance in

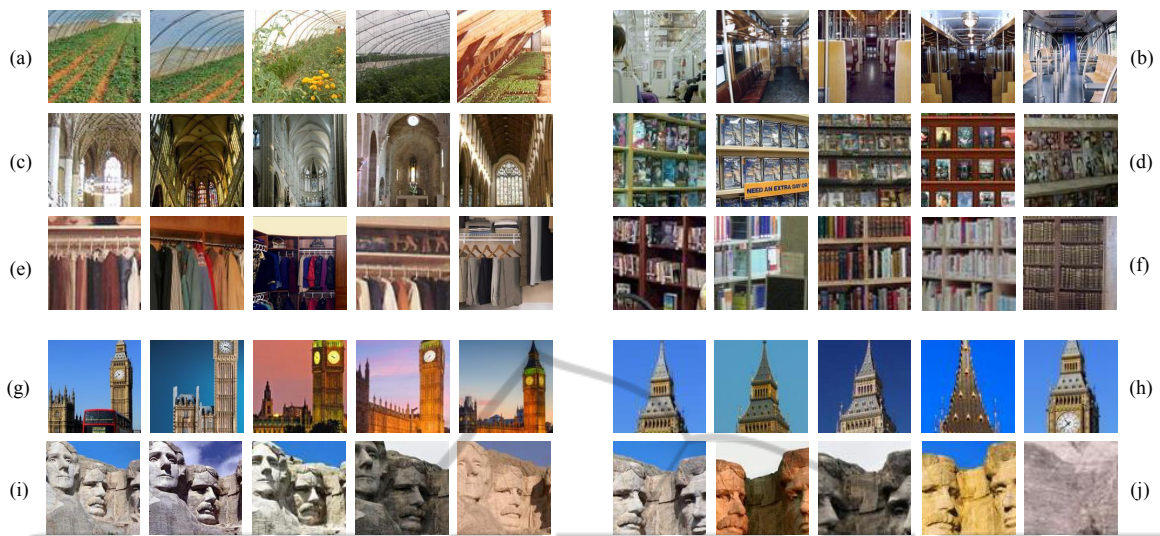


Figure 1: Visual elements extracted from classes (a) greenhouse (b) inside subway (c) church inside (d) video store (e) closet (f) library of MIT Indoor 67 dataset and (g) (h) Big Ben (i) (j) Mount Rushmore of our Outdoor Sight 20 dataset.

the task of part-based scene classification. Besides, our approach can be seamlessly integrated into bag of visual words models to improve the results of multiple computer vision problems.

2. A rich training dataset for outdoor scene detection and classification (Outdoor Sight 20) is built. To our best knowledge, this is the first dataset designed for discovering meaningful mid-level patches of outdoor scenes with good in-class consistency. Our dataset consists of images covering 20 famous tourist attractions around the world.

In the experiments, we evaluated our novel FEC method on the public benchmark: MIT Indoor 67 dataset, and the newly created Outdoor Sight 20 dataset, achieving extremely efficient performance (about 20x faster) while maintaining close to state-of-the-art accuracy.

Some of our results are shown in figure 1. (a)-(f) are discriminative visual elements extracted from MIT Indoor 67, while (g)-(j) come from our Outdoor Sight 20. As is shown in the figure, our method not only captures discriminative and representative visual elements from training data with only class labels provided, it also discovers and distinguishes different visual elements of the same concept, like (g)&(h), which is naturally capable of recognizing different scenes.

2 RELATED WORK

The practice of using parts to represent images has been adopted for quite a long time (Agarwal et al.,

2004). Since parts are considered more semantically meaningful compared to some low level features, the introduction of image descriptor generated by algorithms like ScSPM (Yang et al., 2009), LLC (Shabou and LeBorgne, 2012) and IFV (Perronnin et al., 2010) presented the promising future of parts. The idea of training classifiers discriminatively improved the performance of object detection (Felzenszwalb et al., 2010). However, the discovery of parts are still heavily relied on the training data. Some used the bounding box information on which several assumptions between the parts and the ground truth were based (Felzenszwalb et al., 2008), while others relied on partial correspondence (Maji and Shakhnarovich, 2013) to generate meaningful patches.

It was not until recent years that the issue of discovering discriminative mid-level patches automatically with little or no supervision was raised. Patch discovery using geometric information showed that such method has the ability to learn and extract semantically meaningful visual elements for image classification (Doersch et al., 2012) (Shen et al., 2013) (Lee et al., 2013). Unsupervised learning of patches which are frequent and discriminative in an iterative manner boosted the performance of object detection (Singh et al., 2012). (Juneja et al., 2013) summarized a simple and general framework to mine discriminative patches using exemplar SVM (Malisiewicz et al., 2011) and showed that this framework was efficient in scene classification in combination with the use of bag-of-parts and bag-of-visual-words models.

Recent works on discriminative mid-level patches can be categorized into two groups. One is to apply this method to other computer vision problems

like video representation (Jain et al., 2013), 2D-3D alignment (Aubry et al., 2014), movement prediction (Walker et al., 2014) (Lim et al., 2013) or learning image attributes (Sandeep et al., 2014) (Lee et al., 2013). The other is to collect Internet images to enrich the visual database of discriminative mid-level patches (Li et al., 2013) (Chen et al., 2013). In these works the most widely used types of classifiers are mainly variants of SVM. They can achieve satisfactory accuracy but the huge time consumption really becomes a factor that must be considered if we want to apply this technique to large scale computer vision problems. (Jia et al., 2013) (Jia et al.,)

3 DISCOVERING DISCRIMINATIVE PATCHES: DESIGNED FOR SPEED

Since our purpose is to speed up the training procedure of the model, we designed it to run very fast from sketch. We followed the idea that discriminative patches would be learned and discovered in a framework which had three stages: seeding, expansion and selection (Juneja et al., 2013). Generally speaking, to discover discriminative patches and the corresponding classifiers that are able to recognize them, we first need to get a bunch of seed patches from the given images. Since the number of patches is enormous, a selection procedure is carried out. They will then be used to train classifiers using our FEC method. Subsequently the classifiers will be ranked using an evaluation function to test whether they are discriminative and representative enough. Those who have top rankings will be kept and used to represent images in the way described in section 4.

3.1 Patch Selection and Feature Extraction

The commonly used ways in patch selection consist of two categories. One is to include all possible patches in an image or randomly select some (Singh et al., 2012), the other is to use some techniques like saliency detection (Li et al., 2013) or superpixels (Juneja et al., 2013) to reasonably remove the patches that are unlikely to contain meaningful information to reduce the problem scale and speed up the training procedure. Patch selection is an essential and indispensable part for a method which aims to run very fast as the training time can be reduced significantly with little impact on the results.

In our method, we introduced a very light-weight

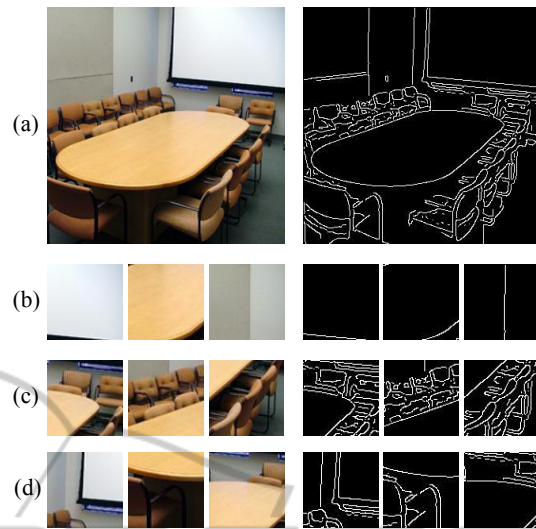


Figure 2: Images and their 'canny' edges, (a) original image (b) patches with few edges (c) patches with too many edges (d) patches with modest number of edges.

way by detecting the number of edges in a patch. The rationale is that we believe the most important feature that human uses to identify different objects and scenes is shape. Edge detection is able to discover the shapes of objects in patches while the number of edges inside a patch somehow suggests the importance of the patch. Intuitively thinking, a patch containing few edges may be a part of the background which lacks discriminativeness, while a patch containing a lot of edges may involve too much details which lacks representativeness. As a result, to ensure our patch selection procedure are able to choose patches that are meaningful, we shall select those with neither too many edges nor too few edges. Figure 2 shows how this works. (a) presents the initial training image from MIT Indoor 67 and its edges detected using Canny method (Canny, 1986). (b) and (c) are some patches with too few or too many edges. (d) shows the patches with modest number of edges, which contain only one or two objects and their spatial relationship. Even though edge detection is rather simple, it is very efficient and effective to find the patches that we need.

In our experiment, we selected patches with sizes of 80×80 , 120×120 , 180×180 , 270×270 . To avoid duplicates, very similar patches with close feature vectors (i.e. the city-block distance is smaller than a threshold $\delta = 0.01$) from the same image were removed. Then the percentage of the area covered by edges in each patch was calculated and 60 patches with medium number of edges among all the patches were kept for each image. We used the HOG feature (Dalal and Triggs, 2005) to represent each patch.

3.2 Classifier Training

The training procedure of the classifiers is the most time consuming section in discriminative part based techniques. Traditional approaches use SVM variants like exemplar SVM (Malisiewicz et al., 2011) and miSVM (Andrews et al., 2002). For example, Juneja introduced the exemplar SVM and the outcome was satisfactory in terms of classification results (Juneja et al., 2013). In each round, one patch from a certain class is a positive input, while all patches from other classes (Doersch et al., 2012) are used as negative inputs. After the SVM is trained, it is used to find the top best patches whose scores are highest among the current class. These patches are added into the positive input and the SVM is trained iteratively for several times. It is undeniable that these methods are able to mine discriminative part classifiers eventually. However, the total number of trained SVMs during the training procedure can reach millions and will take lots of time.

We managed to solve this problem by using a fast yet efficient type of classifier instead. We call it fast exemplar clustering(FEC). It follows the idea that each patch will be given a chance to see whether it is able to become a cluster (Juneja et al., 2013). Each cluster will then be tested to see if it is discriminative and representative among all the clusters.

The training procedure is shown in figure 3(b) and algorithm 1. Each exemplar cluster will be trained only twice. For a specific patch, it is treated to be a cluster center at first. Then the 10 closest patches whose class labels are the same as the initial patch will be added to the cluster. The cluster center is recalculated using the mean value of these points, followed by adding the next 10 closest and non-duplicate patches with the same class label into the cluster. Each cluster C_i is represented by a clustering center P_i and a radius r_i which is equal to the largest distance between the cluster center and the patches inside the cluster. A classifier can then be built from the resultant cluster. The center and the radius form an Euclidean ball which naturally divide the feature space into two parts, the inner part of the cluster and the outer part. The purpose of training is to transform the initial patch which is specific and particular to a visual concept which is generalized and meaningful.

Since we use the distance measure of feature vectors to form a cluster, the biggest challenge is the risk of over-fitting. The reason why we train only twice in clustering is that we want the clusters to be both **generalized** and **diverse** at the same time to help get rid of over-fitting. **Generalization** means that the cluster can represent not only the initial patch itself but

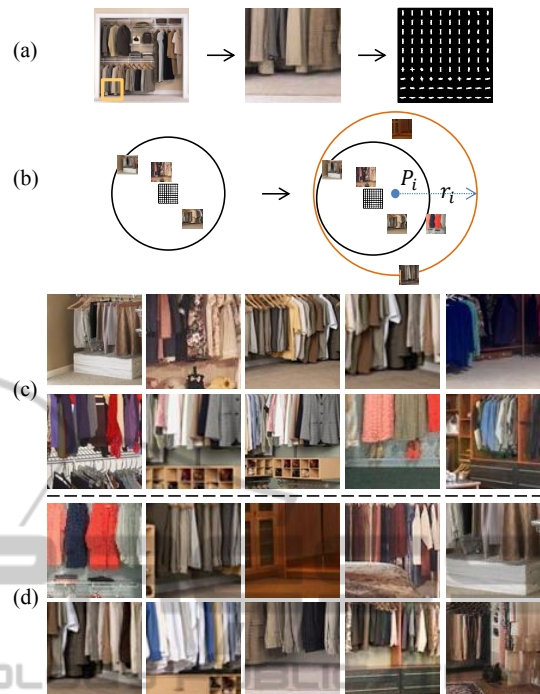


Figure 3: Illustration of training procedure: (a) initial patch and its HOG representation (b) illustration of cluster expansion using FEC (c) example of patches added in first round of training (d) example of patches added in second round of training.

also the patches that are visually similar to the initial patch. Generalization ensures that the patch chosen is identical and common. We want the clusters to be **diverse** since we still do not know which cluster can really represent a discriminative visual concept. If we are able to keep the diversity of the clusters, we will have more chances of obtaining the best classifier when ranking and filtering them in section 3.3.

We did several experiments to decide the optimal number of training rounds, in which two results are really revealing. In one experiment we clustered until the center converges, while in the other we simply did not cluster at all, i.e., we used the initial patch as the center directly with fixed radius for all clusters. It turned out both of them worked poorly. We looked into the results and found that the first way resulted in a lot of same classifiers which lack diversity, while the latter way resulted in serious over-fitting since one classifier is built merely on one data point. Good generalization and broad coverage are the key to find high quality classifiers.

3.3 Classifier Selection

Though we have obtained a bunch of classifiers $C = \{C_i\}$ centered at $P = \{P_i\}$ with radius of $r = \{r_i\}$ in

Algorithm 1: Build exemplar cluster from a patch.

```

function BUILDCLUSTER(patch)
  cluster ← []
  for  $i = 1 \rightarrow 2$  do
    euclidean(patch, patchesInSameClass)
    add 10 closest patches → cluster
    patch ← mean(cluster)
  end for
  center ← mean(cluster)
  radius ← max(euclidean(center, cluster))
  return < center, radius >
end function

```

the training procedure, the number of classifiers is still enormous and most of them are neither representative nor discriminative. To test whether a classifier C_i is good enough, we try to find all the patches inside the Euclidean ball centered at P_i with radius r_i , and compare the class labels of these patches with the class label of the classifier. Denote n_i to be the number of patches inside the ball and p_i to be the number of patches inside the ball with the same class label as the classifier's. Then the accuracy of each classifier is p_i/n_i .

However, if we use the accuracy as the only evaluation, it is very likely that the classifiers will only recognize features from very few images. It may lead to the absence of representativeness. To overcome this, we count the number of true positive patches that each image contributed and calculate the variance σ_i^2 of these numbers. A smaller σ_i^2 indicates that the true positive patches come from more training images, which suggests that the classifier is more representative than others with higher σ_i^2 values. The scoring function is then formulated as

$$F(C_i) = \frac{p_i}{n_i} \log\left(\frac{M}{\sigma_i^2 + N} + 1\right). \quad (1)$$

M, N are scaling constants to normalize the intensity of the two parts. The argument M, N are calculated by $\arg \min_{M, N} \sum_{j, C_j \in C} \left(\frac{(p_i)_j}{(n_i)_j} - \log\left(\frac{M}{(\sigma_i^2)_j + N} + 1\right)\right)^2$. Actually according to our experiment results, the actual value of M, N doesn't have much impact on the results as long as it roughly balance the two parts.

Figure 4 shows the best classifier selected using different evaluation. (a) shows the result of evaluating with accuracy rate only. The five nearest patches come from 3 different images. Even though they are visually consistent, they did not reveal the nature that really makes 'computer room' different from other classes. (b) shows the top classifier evaluated using our evaluation function. The five nearest patches come from 5 different images. The resultant classifier is more representative.

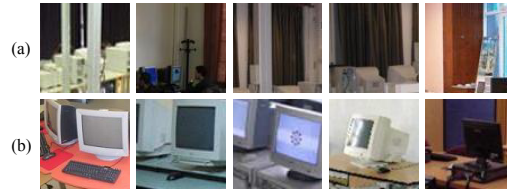


Figure 4: Evaluation comparison of classifier trained on class 'computer room': (a) evaluate using only accuracy rate (b) evaluate using function (1) in section 3.3.

In addition to evaluating the classifiers on the training set, we introduced a large validation set to be used in the same fashion described above. A number of classifiers with top rankings will be chosen as discriminative classifiers. Figure 1 shows the results.

4 IMAGE REPRESENTATION AND CLASSIFICATION

Since it is very hard to judge whether a patch classifier is good or not, we need to test our classifiers using a traditional computer vision task. In our experiment we introduced scene classification to compare our results with others to show that patch classifiers discovered in our method are meaningful and useful.

For the task of scene classification, we need to first represent each image as a vector. We followed the idea of 'bag-of-parts'(BoP) (Juneja et al., 2013) and used the discriminative classifiers learned in section 3 to generate the mid-level descriptor for each image in a spatial pyramid manner (Lazebnik et al., 2006) using 1×1 and 2×2 grids. In practice, patches are extracted using a sliding window and each patch together with its flipped mirror is evaluated using the part classifiers. As a result, each image is represented by a $n \times 5 \times m$ dimensional vector, in which m represents the number of classifiers kept for each class in section 3.3 and n is the total number of classes.

Scene classification accuracy could be further improved if BoP representation is used in combination with Bag of Words(BoW) models like Locality-constrained Linear Coding(LLC) BoW (Shabou and LeBorgne, 2012) or Improved Fisher Vectors (IFV) (Perronnin et al., 2010). However, to make sure our comparison is on an even base, we presented our results using only the BoP representation. We tested the union representations though in section 5 as a reference.

One-vs-rest classifiers are trained to classify the scenes. Linear SVM is used for BoP representation and linear encoding. For the IFV encoding, Hellinger kernel is used.

5 EXPERIMENTS AND RESULTS

The framework of FEC is simple and runs extremely fast. It is not surprising that people will question the effectiveness and correctness of these classifiers and the corresponding image descriptor generated in section 4. In order to test the classifiers we obtained, we focused on the task of scene classification using two datasets. One is the MIT Indoor 67 dataset (Quattoni and Torralba, 2009), the other is the Outdoor Sight 20 that we created.

MIT Indoor 67 consists of 5 main scene categories, including store, home, public places, leisure and working place. Each category contains several specific classes, making a total of 67 classes. This dataset is quite challenging thus widely used in scene classification problems.

Outdoor Sight 20 is a dataset we created which consists of outdoor views of 20 famous tourist attractions around the world such as Big Ben, The Eiffel Tower and The Great Wall of China. To test the ability of distinguishing different scenes, a 21st class which contains images of non-tourist attractions is introduced. Part of the sample images are shown in figure 5. We built this dataset since we wanted to test our models on both indoor and outdoor scenes. As a complementary of the MIT Indoor 67 dataset, it is specifically designed to include only outdoor images, most of which are photos taken from different angles with various lighting conditions while some are sketches or drawings. Among all the images, the majority have a good within-class consistency since they are portrayals of the same object while some are even difficult for human to classify due to a lot of shared characteristics, like (g) and (f) of figure 5.

In our experiment on MIT Indoor 67 dataset, we draw 100 random images from each class. They are partitioned into training set containing 80 images and test set from the remaining 20 images. The training set is further split equally into two parts to be used as training part and validation part, each with 40 images. 50 classifiers for each class are kept to recognize the visual words.

To test the discriminatively trained mid-level patches, we compared our results (FEC+BoP) with ROI (Quattoni and Torralba, 2009), MM-scene (Zhu et al., 2010), DPM (Pandey and Lazebnik, 2011), CENTRIST (Wu and Rehg, 2011), Object Bank (Li et al., 2010), RBoW (Parizi et al., 2012), Patches (Singh et al., 2012), Hybrid-Parts (Zheng et al., 2012), LPR (Sadeghi and Tappen, 2012), exemplar SVM + BoP (Juneja et al., 2013) and IVC (Li et al., 2013). The results are shown in table 1. Even though our method did not achieve highest accuracy rate, it

Table 1: Test results on MIT Indoor 67 dataset.

Method	Accuracy (%)
ROI	26.05
MM-scene	28.00
DPM	30.04
CENTRIST	36.90
Object Bank	37.60
RBoW	37.93
Patches	38.10
Hybrid-Parts	39.80
LPR	44.84
IVC(miSVM)	47.60
Exemplar SVM + BoP	46.10
FEC+BoP(Ours)	40.30

Table 2: Test results on Outdoor Sight 20 dataset. Comparison between accuracy rate and training time for part classifier is presented.

Method	Acc. (%)	Time(\approx)
Exemplar SVM + BoP	85.75	5 days
FEC+BoP(Ours)	79.25	7 hours

should be clarified that we did not mean to produce best scene classification result. We presented these numbers to show that the patches we obtained in the way described in section 3 are indeed meaningful and could be used as discriminative classifiers in various computer vision problems.

We compared the training time required to obtain discriminative mid-level patches with exemplar SVM (Malisiewicz et al., 2011) and ours. On an ordinary Quad-core i5-3570 computer with 16GB RAM installed using Matlab 2013b, the exemplar SVM took around 3 weeks to train while ours took only 1 day(20x faster). This is an impressive result as the accuracy rate did not show an enormous drop compared to the exemplar SVM + BoP method.

As is mentioned in section 4, the accuracy rate could be further improved if BoP representation is used in combination with BoW features. In our experiment, the FEC + BoP + LLC and FEC + BoP + IFV achieved the accuracy rate of 49.55% and 53.81% respectively using parameters introduced in (Chatfield et al., 2011).

As for our Outdoor Sight 20 dataset, we followed the exact same procedure as MIT Indoor 67 dataset on the same computers with the same number of images used in training, testing and validation for each class. We compared our results with exemplar SVM + BoP (Juneja et al., 2013) in table 2 to show that our FEC could train discriminative mid-level patches as well as the exemplar SVM with much less time.

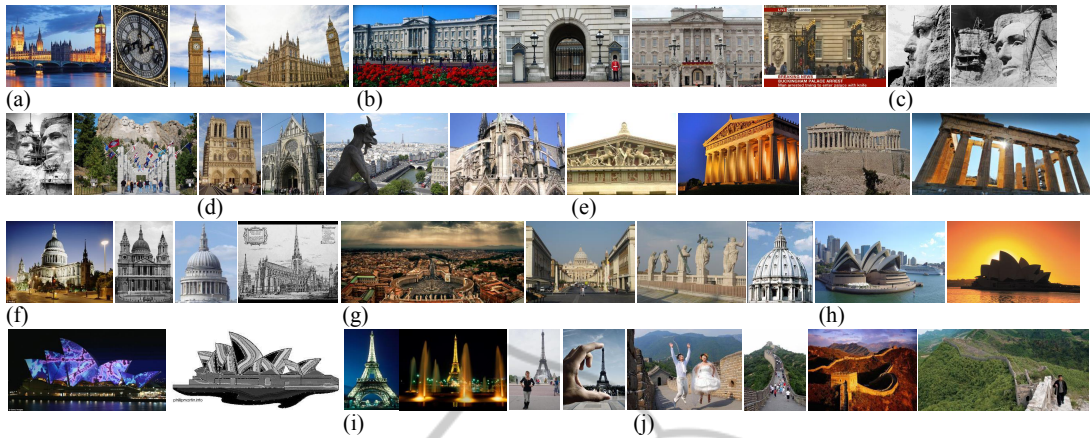


Figure 5: Sample images of Outdoor Sight 20 dataset of classes (a) Big Ben (b) Buckingham Palace (c) Mount Rushmore (d) Notre Dame (e) Parthenon (f) St. Paul's Cathedral (g) St. Peter's Basilica (h) Sydney Opera House (i) The Eiffel Tower (j) The Great Wall of China. The rest are: The Brandenburg Gate, The Colosseum, The Golden Gate Bridge, The Kremlin, The Leaning Tower of Pisa, The Pyramids of Giza, The Statue of Liberty, The Taj Mahal, The White House, Tower Bridge with an additional class of none attraction images.

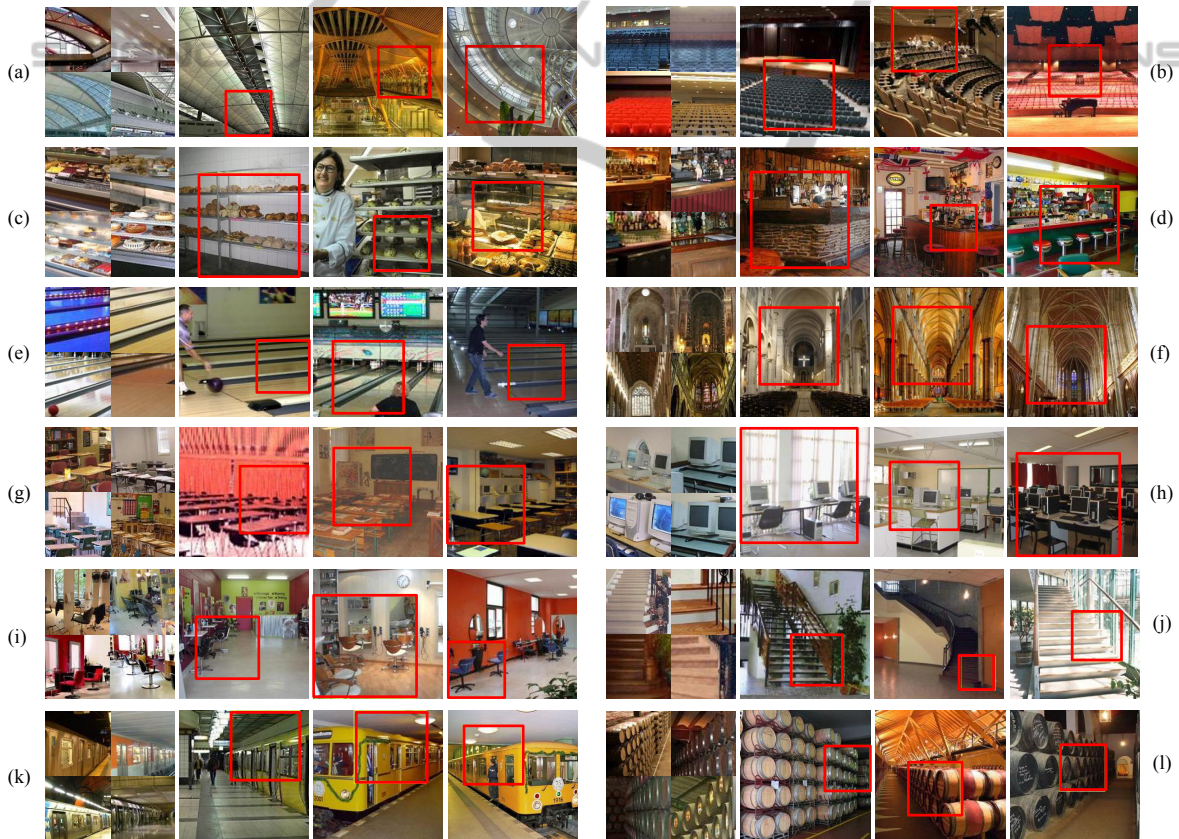


Figure 6: Classifiers trained on classes: (a) airport inside (b) auditorium (c) bakery (d) bar (e) bowling (f) church inside (g) classroom (h) computer room (i) hair salon (j) staircase (k) subway (j) wine cellar of MIT Indoor 67 dataset. The left four patches of each part show how this classifier is trained and the three images on the right show their detections on the testing image.

6 CONCLUSION

In this paper we proposed a novel approach to learn discriminative mid-level patches from training data with only class labels provided. The motivation lies in that current discriminative patch learning methods are too time-consuming and can hardly be applied to complicated computer vision problems with larger dataset. We proposed the FEC algorithm to train part classifiers. Under proper validation settings and appropriately designed evaluation function, we obtained classifiers whose accuracy could compete with state-of-the-art SVM based classifiers. We tested our classifiers on scene classification using MIT Indoor 67 and our Outdoor Sight 20. Both results revealed they were as good as classifiers generated by the contemporary methods. Our classifiers could be further applied to other computer vision problems like scene classification, video classification, object detection, 2D-3D matching.

REFERENCES

- Agarwal, S., Awan, A., and Roth, D. (2004). Learning to detect objects in images via a sparse, part-based representation. *Pattern Analysis and Machine Intelligence (PAMI), 2004 IEEE Transactions on*, 26(11):1475–1490.
- Andrews, S., Tsochantaridis, I., and Hofmann, T. (2002). Support vector machines for multiple-instance learning. In *NIPS*, pages 561–568.
- Aubry, M., Maturana, D., Efros, A. A., Russell, B. C., and Sivic, J. (2014). Seeing 3d chairs: exemplar part-based 2d-3d alignment using a large dataset of cad models. In *CVPR, 2014 IEEE Conference on*. IEEE.
- Canny, J. (1986). A computational approach to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (6):679–698.
- Chatfield, K., Lempitsky, V. S., Vedaldi, A., and Zisserman, A. (2011). The devil is in the details: an evaluation of recent feature encoding methods. pages 1–12.
- Chen, X., Shrivastava, A., and Gupta, A. (2013). Neil: Extracting visual knowledge from web data. In *ICCV, 2013 IEEE International Conference on*, pages 1409–1416. IEEE.
- Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *CVPR, 2005 IEEE Conference on*, volume 1, pages 886–893. IEEE.
- Doersch, C., Singh, S., Gupta, A., Sivic, J., and Efros, A. A. (2012). What makes paris look like paris? *ACM Transactions on Graphics (TOG)*, 31(4):101.
- Felzenszwalb, P., McAllester, D., and Ramanan, D. (2008). A discriminatively trained, multiscale, deformable part model. In *CVPR, 2008 IEEE Conference on*, pages 1–8. IEEE.
- Felzenszwalb, P. F., Girshick, R. B., McAllester, D., and Ramanan, D. (2010). Object detection with discriminatively trained part-based models. *Pattern Analysis and Machine Intelligence (PAMI), 2010 IEEE Transactions on*, 32(9):1627–1645.
- Jain, A., Gupta, A., Rodriguez, M., and Davis, L. S. (2013). Representing videos using mid-level discriminative patches. In *CVPR, 2013 IEEE Conference on*, pages 2571–2578. IEEE.
- Jia, X., Yang, H., Lin, A., Chan, K.-P., and Patras, I. Structured semi-supervised forest for facial landmarks localization with face mask reasoning. In *BMVC, 2014 IEEE International Conference on*. IEEE.
- Jia, X., Zhu, X., Lin, A., and Chan, K. P. (2013). Face alignment using structured random regressors combined with statistical shape model fitting. In *28th International Conference on Image and Vision Computing New Zealand, IVCNZ 2013, Wellington, New Zealand, November 27-29, 2013*, pages 424–429.
- Juneja, M., Vedaldi, A., Jawahar, C., and Zisserman, A. (2013). Blocks that shout: Distinctive parts for scene classification. In *CVPR, 2013 IEEE Conference on*, pages 923–930. IEEE.
- Lazebnik, S., Schmid, C., and Ponce, J. (2006). Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR, 2006 IEEE Conference on*, volume 2, pages 2169–2178. IEEE.
- Lee, Y. J., Efros, A. A., and Hebert, M. (2013). Style-aware mid-level representation for discovering visual connections in space and time. In *ICCV, 2013 IEEE International Conference on*, pages 1857–1864. IEEE.
- Li, L.-J., Su, H., Fei-Fei, L., and Xing, E. P. (2010). Object bank: A high-level image representation for scene classification & semantic feature sparsification. In *Advances in neural information processing systems*, pages 1378–1386.
- Li, Q., Wu, J., and Tu, Z. (2013). Harvesting mid-level visual concepts from large-scale internet images. In *CVPR, 2013 IEEE Conference on*, pages 851–858. IEEE.
- Lim, J. J., Zitnick, C. L., and Dollár, P. (2013). Sketch tokens: A learned mid-level representation for contour and object detection. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 3158–3165. IEEE.
- Maji, S. and Shakhnarovich, G. (2013). Part discovery from partial correspondence. In *CVPR, 2013 IEEE Conference on*, pages 931–938. IEEE.
- Malisiewicz, T., Gupta, A., and Efros, A. A. (2011). Ensemble of exemplar-svms for object detection and beyond. In *ECCV, 2011 IEEE International Conference on*, pages 89–96. IEEE.
- Mittelman, R., Lee, H., Kuipers, B., and Savarese, S. (2013). Weakly supervised learning of mid-level features with beta-bernoulli process restricted boltzmann machines. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 476–483.
- Pandey, M. and Lazebnik, S. (2011). Scene recognition and weakly supervised object localization with de-

- formable part-based models. In *ECCV, 2011 IEEE International Conference on*, pages 1307–1314. IEEE.
- Parizi, S. N., Oberlin, J. G., and Felzenszwalb, P. F. (2012). Reconfigurable models for scene recognition. In *CVPR, 2012 IEEE Conference on*, pages 2775–2782. IEEE.
- Perronnin, F., Liu, Y., Sánchez, J., and Poirier, H. (2010). Large-scale image retrieval with compressed fisher vectors. In *CVPR, 2010 IEEE Conference on*, pages 3384–3391. IEEE.
- Quattoni, A. and Torralba, A. (2009). Recognizing indoor scenes. In *CVPR, 2009 IEEE Conference on*. IEEE.
- Rios-Cabrera, R. and Tuytelaars, T. (2013). Discriminatively trained templates for 3d object detection: A real time scalable approach. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 2048–2055. IEEE.
- Sadeghi, F. and Tappen, M. F. (2012). Latent pyramidal regions for recognizing scenes. In *ECCV, 2012 IEEE Conference on*, pages 228–241. Springer.
- Sandeep, R. N., Verma, Y., and Jawahar, C. (2014). Relative parts: Distinctive parts for learning relative attributes. In *CVPR, 2014 IEEE Conference on*. IEEE.
- Shabou, A. and LeBorgne, H. (2012). Locality-constrained and spatially regularized coding for scene categorization. In *CVPR, 2012 IEEE Conference on*, pages 3618–3625. IEEE.
- Shen, L., Wang, S., Sun, G., Jiang, S., and Huang, Q. (2013). Multi-level discriminative dictionary learning towards hierarchical visual categorization. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 383–390. IEEE.
- Singh, S., Gupta, A., and Efros, A. A. (2012). Unsupervised discovery of mid-level discriminative patches. In *ECCV, 2012 IEEE Conference on*, pages 73–86. Springer.
- Sun, J. and Ponce, J. (2013). Learning discriminative part detectors for image classification and cosegmentation. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 3400–3407. IEEE.
- Tang, K., Sukthankar, R., Yagnik, J., and Fei-Fei, L. (2013). Discriminative segment annotation in weakly labeled video. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 2483–2490. IEEE.
- Walker, J., Gupta, A., and Hebert, M. (2014). Patch to the future: Unsupervised visual prediction. In *CVPR, 2014 IEEE Conference on*. IEEE.
- Wang, L., Qiao, Y., and Tang, X. (2013). Motionlets: Mid-level 3d parts for human motion recognition. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 2674–2681. IEEE.
- Wu, J. and Rehg, J. M. (2011). Centrist: A visual descriptor for scene categorization. *Pattern Analysis and Machine Intelligence (PAMI), 2011 IEEE Transactions on*, 33(8):1489–1501.
- Yang, J., Yu, K., Gong, Y., and Huang, T. (2009). Linear spatial pyramid matching using sparse coding for image classification. In *CVPR, 2009 IEEE Conference on*, pages 1794–1801. IEEE.
- Zheng, Y., Jiang, Y.-G., and Xue, X. (2012). Learning hybrid part filters for scene recognition. In *ECCV, 2012 IEEE Conference on*, pages 172–185. Springer.
- Zhu, J., Li, L.-J., Fei-Fei, L., and Xing, E. P. (2010). Large margin learning of upstream scene understanding models. In *Advances in Neural Information Processing Systems*, pages 2586–2594.