What's that Style?
A CNN-based Approach for Classification and Retrieval of Building Images

Rachel D. Meltser
Lake Forest College
555 North Sheridan Road
Lake Forest, IL 60045
Email: meltserrd@lakeforest.edu

Sugata Banerji
Lake Forest College
555 North Sheridan Road
Lake Forest, IL 60045
Email: banerji@lakeforest.edu

Atreyee Sinha
Edgewood College
1000 Edgewood College Drive
Madison, WI 53711
Email: asinha@edgewood.edu

***Abstract*—Image classification and content-based image retrieval (CBIR) are important problems in the field of computer vision. In recent years, convolutional neural networks (CNNs) have become the tool of choice for building state-of-the-art image classification systems. In this paper, we propose novel mid-level representations involving the use of a pre-trained CNN for feature extraction and use them to solve both the classification and the retrieval problems on a dataset of building images with different architectural styles. We experimentally establish our intuitive understanding of the CNN features from different layers, and also combine the proposed representations with several different pre-processing and classification techniques to form a novel architectural image classification and retrieval system.**

1. Introduction

Content-based Image Retrieval (CBIR) is a very important Computer Vision problem being addressed by researchers around the world today. CBIR is the task of searching for similar images in a large image database using a user-provided image as a query. The concept of similarity is context-dependent. For example, in the context of a building images dataset, similar images may mean buildings having the same purpose as the query, or it may mean images of the same architectural style. Hence the retrieval problem is non-trivial and rather challenging for several domains. Also, CBIR may be used for classification if the images in the dataset are already labelled with different predefined category labels. These category labels may be based on some low-level features such as color, texture or shape, but more often, they are based on more high-level features such as semantic description, activity, or as in our case, architectural style. In the past few years, convolutional neural networks (CNNs) have been among the best-performing tools used by vision researchers for a variety of classification tasks. The initial use of CNNs became popular by the availability of large labelled image datasets such as ImageNet [1] and Places [2] and the large improvement in object and scene classification results obtained thereafter [3]. Later, researchers have adapted the network of [3] for different tasks by modifying the architecture or tweaking the network parameters. Convolutional neural networks typically contain multiple convolution and pooling layers followed by a few fully connected layers and a soft-max classifier. It has been demonstrated in [4], [5] and [6] that using the output from the last fully connected layer of pre-trained CNNs such as [7] with linear classifiers such as support vector machines (SVMs), yields better classification performance.

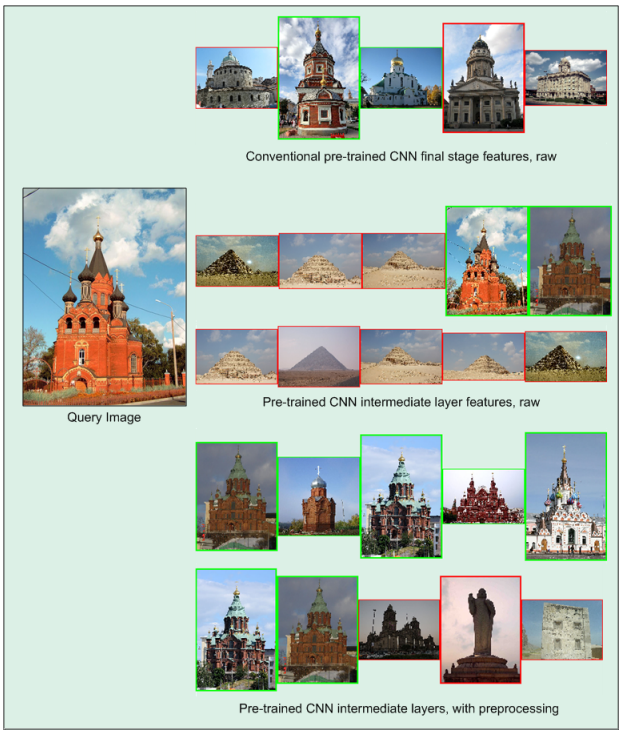

Figure 1. Images retrieved using the query on the left and raw CNN features from the last and intermediate layers of a pre-trained CNN. Note that the pre-processing vastly improves the retrieval results in the intermediate layers. Green borders indicate images from the same style category as the query and red borders indicate those from a different category.

Architectural style classification and retrieval is an emerging research area in computer vision, which has gained

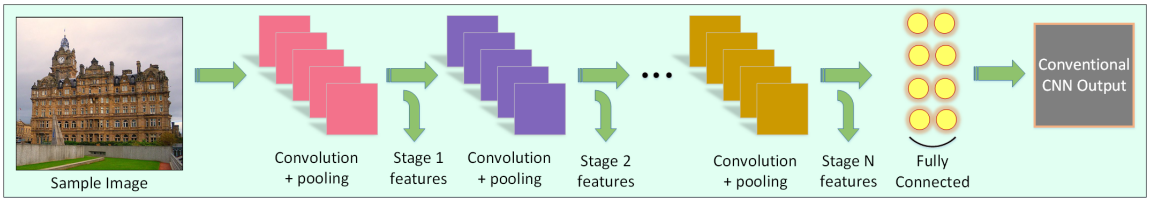

Figure 2. The proposed image representation uses features from intermediate layers of a pre-trained CNN. These features are used for retrieval and classification. The process is described in detail in Section 2

attention in the recent years [8]. It has many potential applications in the tourism industry, historical analysis of building styles, cinema and theater set design, architecture education, geo-localization, etc. Each architectural style possesses a set of unique and distinguishing features [9] some of which, like the facade and its decorations, enable automatic classification using computer vision methods. Architectural styles are not independently and identically distributed, and the styles evolve as a gradual process over time which may lead to complicated relationships between different architectural styles. A comparative evaluation of different conventional classification techniques by [8] for architectural style classification clearly suggests the need for more powerful visual features for architectural style classification and retrieval tasks. This is our primary motivation in selecting this problem for the current work.

In the presented work we first propose a novel scene representation and associated similarity measure, which exploits evidence about the presence of different visual patterns in the various architectural styles by using the outputs of the intermediate layer features of a pre-trained CNN for classification and retrieval of architecture images from the large Architectural Style Dataset [8]. Using a CNN pre-trained on ImageNet [3] we consider the response maps computed at several different layers to compare their performance. We demonstrate that these features are more effective for the retrieval task and also for the architectural style classification task. Next, we use a novel pre-processing technique to parse the image into sky and non-sky regions to minimize the effect of sky-region features in the retrieval and classification results and endow the proposed representation only by visual cues from the buildings in the scene.

Finally, we also provide an in-depth visualization and discussion on the suitability and effectiveness of the different layer features for an architecture dataset. The intuition behind the proposed approach is that in initial layers of the CNN, the encoded information is more low-level and spatially localized, and as we move up the layers, the information becomes more and more semantic. In the fully connected layers the information is fully semantic and free from stylistic details or spatial fluctuations. Hence, the lower-layer features may be more ideal for recognizing attributes of buildings rather than the class of "buildings" as a whole. Figure 1 shows the different nearest neighbors to a query image for features extracted from different layers of the pre-trained CNN.

2. Proposed Method

The proposed method uses a pre-trained CNN for extracting features at various stages and compares their performance for both architectural style classification and retrieval problems. We use features from several different layers for image representation and compare the classification results from three different classifiers. We also create a novel pre-processing step to remove the sky from the images. These steps are discussed in detail in subsections 2.1, 2.2 and 2.3.

2.1. Feature Extraction

We use the OverFeat image features extractor [7] for feature extraction. OverFeat is based on a convolutional network similar to [3] trained on the 1000-category ImageNet dataset [1]. We do not use the classifier included with OverFeat as it classifies into one of the ImageNet categories. We use the 'fast' network of OverFeat which uses input images of size 231×231 and has 21 layers divided into 8 stages before the final softmax output stage. The first six of these stages consist of convolution and pooling layers and the last two stages are fully connected layers. OverFeat can be used to extract the features from any of these layers and use them for representation. In the proposed method we extract features following each of the first six stages as well as the layers in between the stages, and use them with our own classifiers. This is shown in Figure 2.

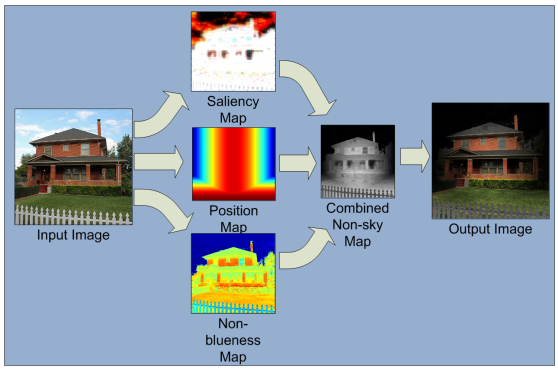

Figure 3. The pre-processing of input images for removing the sky pixels to make the representation more robust. The process is explained in Section 2.2

2.2. Pre-processing

One of the problems that we observed in our early retrieval experiments was that for nearest-neighbor comparisons based on the raw CNN features from most layers, the retrieved images were mostly images of pyramids. This can be observed in Figure 1. In the classification experiments using the KNN classifier, this caused all queries to be classified into the Ancient Egyptian Architecture class. On back-tracking through the network and visualizing what part of image contributed to each feature, we found that the sky formed a significant portion of most images in the database. Since images of the Ancient Egyptian Architecture category are mostly pyramids with bright blue sky, they were being retrieved as close matches.

To overcome this problem we designed a sky detector that tries to predict the pixels representing the sky and blocks them out before feature extraction. We combine three cues for detecting the sky, namely saliency, position and color.

For generating the saliency cue, we use a difference of Gaussian (DoG) operation at multiple scales to detect interest points within the image. Wherever we find an interest point at some scale, we turn that pixel white (the value 1) and we turn other pixels black (the value 0). This creates a response map with the interest points marked. We create such response maps for eight scales and eventually add them. The sum image is normalized to [0,1] and each pixel now has a score between 0 and 1 based on the presence of interest points at different scales at that pixel. The area with very few interest points is potentially the sky.

For the position cue, we use a subset of the images in the dataset to learn the position of the buildings in several categories. The assumption made here is that, these training images are representative of the position that buildings occupy in most images of the dataset. We manually mark the position of the buildings in these images and then, using the position information from these representative images from different categories, we fit two 1-dimensional Gaussian kernels over the image, one for each axis. The horizontal Gaussian kernel is centered at the horizontal mid-point of the image while the vertical one is centered on the bottom edge. We take the maximum of the values of the two kernels at any pixel to get a measure of the probability of that pixel being part of the building.

For the color cue, the average of the red and the green values subtracted from the blue value is used as a 'blueness' measure. Once normalized, this cue predicts the location of sky pixels with some accuracy, although this cue is not completely accurate for glass buildings that reflect the sky. The complement of this cue is taken as a 'non-sky' score for each pixel. Next, we multiply the three scores from the three cues and normalize the product in the range [0,1]. Finally, we multiply the original image with this normalized product to get the skyless image ready for feature extraction. This process is demonstrated in Figure 3.

It should be noted that the sky removal step is only important for the unsupervised retrieval and classification steps (KNN classification). When we train classifiers using training samples from each class (SVM and EFM-KNN classification), the classifiers can offset the effect of the sky being present in all classes and they perform better without the pre-processing step.

Figure 4. Sample images from the 25 categories of the Architectural Style dataset.

2.3. Classification

2.3.1. The K-Nearest Neighbor Classifier. The simplest classifier that we use is the K-nearest neighbor (KNN) classifier. This is an unsupervised classification technique. All the images are ranked by their distance from the query image, and the closest k matches are used to determine the class label for the query. For this classifier, a training step is not needed as the neighbors are taken from a set of images whose class is known. For all of the results shown in this paper, the value of K is 5.

2.3.2. The EFM-KNN Classifier. Principal component analysis, or PCA, which is the optimal feature extraction method in the sense of the mean-square-error, derives the most expressive features for signal and image representation [10]. However, they are not the optimum features for classification. Fisher's Linear Discriminant (FLD), a popular method in pattern recognition, first applies PCA for dimensionality reduction and then discriminant analysis for feature extraction.

The FLD method, however, often leads to overfitting when implemented in an inappropriate PCA space. To improve the generalization performance of the FLD method, a proper balance between two criteria should be maintained: the energy criterion for adequate image representation and the magnitude criterion for eliminating the small-valued trailing eigenvalues of the within-class scatter matrix. The Enhanced Fisher Model (EFM) improves the generalization capability of the FLD method by decomposing the FLD procedure into a simultaneous diagonalization of the within-class and between-class scatter matrices [11]. The simultaneous diagonalization demonstrates that during whitening the eigenvalues of the within-class scatter matrix appear in the denominator. As shown by [11], the small eigenvalues tend to encode noise, and they cause the whitening step to

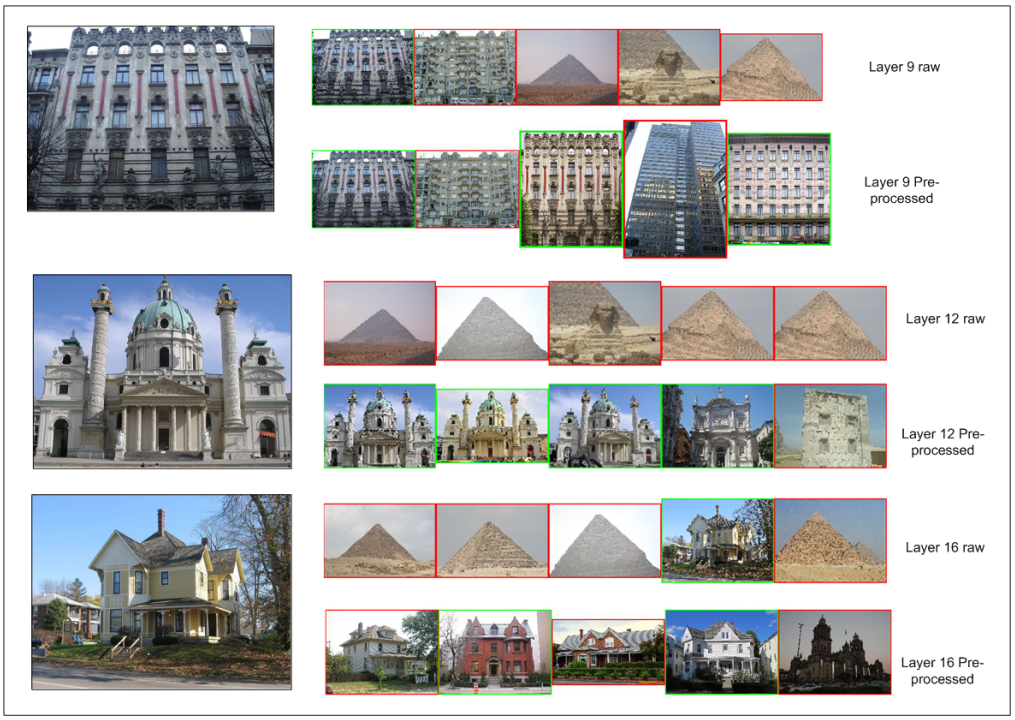

Figure 5. Some results from the retrieval task. For each query, the top row shows 5 nearest neighbors retrieved by the raw-CNN representation. The lower row shows the retrieval set obtained after pre-processing.

fit for misleading variations, leading to poor generalization performance. To enhance performance, the EFM method preserves a proper balance between the need that the selected eigenvalues account for most of the spectral energy of the raw data (for representational adequacy), and the requirement that the eigenvalues of the within-class scatter matrix (in the reduced PCA space) are not too small (for better generalization performance).

After dimensionality reduction and feature extraction by EFM, we use the KNN classifier on the reduced feature vector for the final classification. The EFM feature extraction process followed by nearest neighbor classification has been shown to perform well with a large number of classes [12], [13].

2.3.3. The Linear SVM Classifier. The Support Vector Machine (SVM) minimizes the risk functional in terms of both the empirical risk and the confidence interval [14]. SVM is very popular and has been applied extensively for pattern classification, regression, and density estimation since it displays a good generalization performance. We use the one-vs-all method to train an SVM for each class.

The SVM implementation used for our experiments is the one that is distributed with the VlFeat package [15]. The parameters of the support vector machine are tuned empirically using only the training data, and the parameters that yield the best average precision on the training data are used for classification of the test data.

TABLE 1. CLASS NAMES AND NUMBER OF IMAGES IN EACH CLASS

Class Name	Image Count
Achaemenid architecture	69
American craftsman style	195
American Foursquare architecture	59
Ancient Egyptian architecture	256
Art Deco architecture	366
Art Nouveau architecture	450
Baroque architecture	239
Bauhaus architecture	92
Beaux-Arts architecture	191
Byzantine architecture	111
Chicago School architecture	153
Colonial architecture	177
Deconstructivism	213
Edwardian architecture	79
Georgian architecture	154
Gothic architecture	109
Greek Revival architecture	327
International style	207
Novelty architecture	212
Palladian architecture	113
Postmodern architecture	163
Queen Anne architecture	425
Romanesque architecture	107
Russian Revival architecture	165
Tudor Revival architecture	162

3. Experiments

We run three sets of experiments on our dataset, one with the KNN classifier, one with the EFM-KNN classifier

and the third with the linear SVM classifier. We test all three classifiers to see their effectiveness in the architectural style recognition problem. It should be noted that we use the sky removal pre-processing for the image retrieval experiments and the KNN classification experiments. The dataset used for our work is the Architectural Style Dataset which is described in the following subsection.

3.1. Dataset

We evaluate our representation and classification techniques on the challenging Architectural Style Dataset created by [8]. This dataset consists of color images of buildings from 25 different architectural styles, containing 4794 photographs. These images have been downloaded from the Wikimedia collection and feature an extensive selection from different eras. The names of the categories and the number of images in each are shown in Table 1. Some sample images from the dataset are shown in Figure 4.

For our experiments using supervised classification methods, we use 30 images from each class for training the classifiers, and the rest of the images for testing. These are the same numbers as used by [8]. The training and test splits used for these experiments are randomly generated and we do five-fold cross-validation and report the mean scores. In retrieval tasks, all images other than the query image itself are used to generate the retrieval set.

3.2. Results

The results of our classification experiments are shown in Table 2. Our best results are comapred to other results reported in [8] in Table 3. Our experiments show that the CNN features from the intermediate layers (layers 9-16) outperform the features from both the lower and higher layers. In particular, the highest classification accuracy that we get for the architctural style classification task is 63.9% which is over 17% improvement over the MLLR+SP method

TABLE 2. COMPARISON OF CLASSIFICATION PERFORMANCE (%) BETWEEN THE BEST-PERFORMING CNN LAYERS USING DIFFERENT CLASSIFIERS ON THE ARCHITECTURAL STYLE DATASET

CNN Features Used	KNN	SVM	EFM-KNN
Layer 9 raw	13.2	43.0	59.7
Layer 12 raw	10.0	45.5	62.0
Layer 16 raw	9.1	47.9	**63.9**
Layer 21 raw	41.9	33.7	55.2
Layer 21 skyless	35.9	28.2	50.3
Layer 9 skyless	25.4	39.1	56.3
Layer 12 skyless	23.4	41.5	58.3
Layer 16 skyless	22.7	42.4	59.2

TABLE 3. COMPARISON OF CLASSIFICATION PERFORMANCE (%) BETWEEN OUR BEST-PERFORMING CNN LAYER AND OTHER METHODS AS REPORTED BY [8] ON THE ARCHITECTURAL STYLE DATASET

Method Used	25-Class Classification Rate(%)
GIST	17.39
SP	44.52
OB-Partless	42.50
OB-Part	45.41
DPM-LSVM	37.69
DPM-MLLR	42.55
MLLR+SP [8]	46.21
Layer16 Raw+EFM	**63.90**

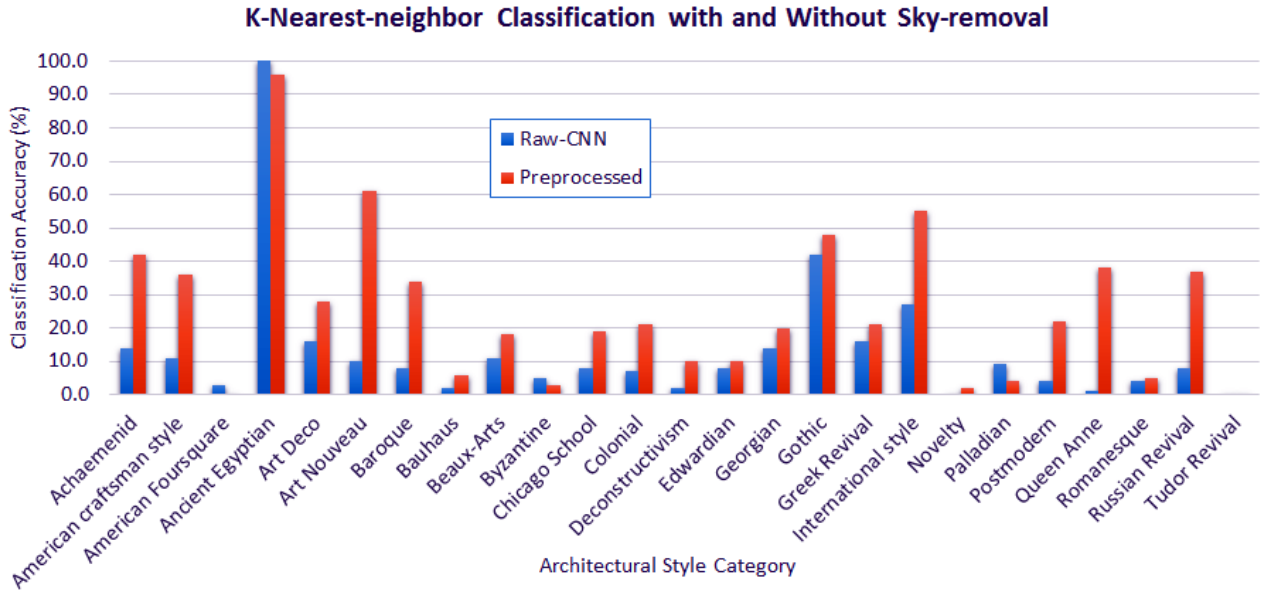

Figure 6. A Comparison of the class-wise classification performance between the layer 9 raw-CNN features and the pre-processed (sky-removed) CNN features. Both the features use a KNN classifier.

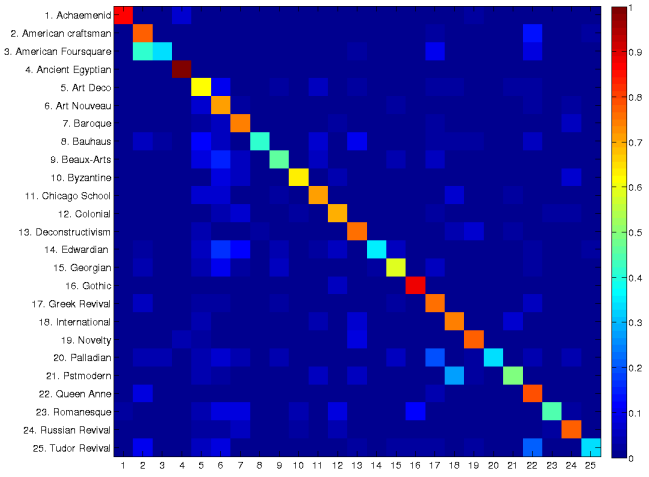

Figure 7. The confusion matrix for architectural style classification using Layer 16 CNN features and EFM-KNN classifier. The rows show the real style categories and the columns show the assigned style categories.

proposed by [8]. We get this result with the layer 16 raw CNN features and the EFM-KNN classifier. The highest success rate yielded by the KNN classifier is 53.8% and for the SVM classifier it is 47.9%. In general, the EFM-KNN classifier performs more consistently well as compared to the SVM classifier. All these features perform better than the stage 8 (final CNN output) layer that is obtained after the fully connected layers. The confusion matrix for the highest result is shown in Figure 7. It can be seen from this figure that the most certain classification is for Ancient Egyptian architecture while the most confusion is between American Craftsman and American Foursquare styles.

The class-wise classification results obtained by using a simple KNN classifier with the raw CNN and the pre-processed representation are compared in Figure 6. It can be seen from the figure that the pre-processing improves the result in almost all the categories.

Figure 1 shows the type of images retrieved by different CNN layers for the same query image, before and after pre-processing. Figure 5 shows three examples of retrieval using features from the layers 9, 12 and 16 of the CNN respectively. The green and red borders around retrieval results indicate correct and incorrect style class labels, respectively.

It should also be noted that the dataset has some problems which affect the classification performance. For instance, there is overlap between the Romanesque and Gothic architecture classes and the same building is present in both categories.

4. Conclusions

We have proposed an image representation based on features extracted from intermediate layers of a pre-trained CNN, and combined this representation with three different classifiers to perform building image classification and retrieval tasks on a large architectural image dataset. Our proposed representation performs better than traditional final-stage CNN features at both retrieval and classification tasks.

In future, we would like to extend this work by fusing the information encoded by the different layers, either at feature level or at decision level, to obtain better classification and retrieval results than those obtained by single layers.

Acknowledgment

The authors would like to thank the Richter Scholars Program at Lake Forest College for partially supporting the research presented in this paper.

References

[1] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR09*, 2009.

[2] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, "Learning deep features for scene recognition using places database," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 487–495.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *NIPS*, 2012, pp. 1106–1114.

[4] Y. Gong, L. Wang, R. Guo, and S. Lazebnik, *Multi-scale Orderless Pooling of Deep Convolutional Activation Features*. Cham: Springer International Publishing, 2014, pp. 392–407. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-10584-0_26

[5] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN features off-the-shelf: an astounding baseline for recognition," *CoRR*, vol. abs/1403.6382, 2014. [Online]. Available: http://arxiv.org/abs/1403.6382

[6] S. Banerji and A. Sinha, "Painting classification using a pre-trained convolutional neural network," in *2nd Workshop on Computer Vision Applications (WCVA) at the Tenth Indian Conference on Vision, Graphics and Image Processing (ICVGIP)*, 2016.

[7] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," *CoRR*, vol. abs/1312.6229, 2013. [Online]. Available: http://arxiv.org/abs/1312.6229

[8] Z. Xu, D. Tao, Y. Zhang, J. Wu, and A. C. Tsoi, "Architectural style classification using multinomial latent logistic regression," in *ECCV*, 2014.

[9] C. Dunlop, *Architectural Styles*. Dearborn Real Estate, 2003.

[10] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, 2nd ed. Academic Press, 1990.

[11] C. Liu and H. Wechsler, "Robust coding schemes for indexing and retrieval from large face databases," *IEEE Transactions on Image Processing*, vol. 9, no. 1, pp. 132–137, 2000.

[12] S. Banerji, A. Sinha, and C. Liu, "New image descriptors based on color, texture, shape, and wavelets for object and scene image classification," *Neurocomputing*, vol. 117, no. 0, pp. 173 – 185, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0925231213001987

[13] A. Sinha, S. Banerji, and C. Liu, "Novel color gabor-lbp-phog (glp) descriptors for object and scene image classification," in *ICVGIP*, 2012, p. 58.

[14] Y. Vapnik, *The Nature of Statistical Learning Theory*. Springer-Verlag, 1995.

[15] A. Vedaldi and B. Fulkerson, "Vlfeat: An open and portable library of computer vision algorithms," 2008.