Dimension Reduction for Scene Recognition

Jonathan H. Lu
Princeton University
Department of Computer Science
jhlu@princeton.edu

Ethan R. Cohen
Princeton University
Department of Computer Science
ethanrc@princeton.edu

Matthew C. Rosen
Princeton University
Department of Computer Science
mcr3@princeton.edu

Jeffrey Gleason
Princeton University
Department of Computer Science
jgleason@princeton.edu

Abstract

Dimensionality reduction is an important problem in machine learning, especially for finding interpretable, visualizable representations of high-dimensional data. We aim to test the effectiveness of a variety of dimensionality reduction techniques to images from the 13Scenes dataset and evaluate how well they represent scenes via classification. We also test several feature representations of the images and find Histogram of Oriented Gradients (HOG) to be optimal. We find that PCA and t-SNE achieve the best classification and also run the fastest among dimensionality reduction techniques. We discuss our classification results, quality of embeddings, and the underlying components found by PCA and NMF.

1 Introduction

Dimensionality reduction techniques are an active field of research in machine learning, as larger and larger datasets must be analyzed. Furthermore, unlike the default convolutional neural networks, these techniques 1) don't requires millions of training examples and 2) have the advantage of being interpretable and visualizable. These are especially important for data scientists who wish to make decisions based off their analysis. We chose to apply a variety of these techniques for scene recognition, due to the complexity of feature types in this field. We wished to find the technique that best represented scenes.

We first represented our scenes as a variety of features, including HOG and SIFT. We then tested a variety of dimensionality reduction techniques on these, evaluating their performance by running them through a variety of classifiers. We find that HOG features performed the best using PCA reduced dimensions. We also find that PCA and t-SNE perform well compared to the other methods in both classification rate and time. Finally, we find that forest is well classified and discuss feature importances.

1.1 Dataset

We performed our analysis on the 13 Natural Scenes dataset compiled by Li and Perona [7]. Considered "the most complete scene category dataset used in literature" at the time of its creation, its size (3849 images) and number of scene classes (13) are commensurate to the scale of our analysis and our resources. Though other, newer scene category datasets exist (namely MIT's Indoor-67 and Places2), we found them too large to use effectively. Places2, for example, includes 2.5 million images, almost 115 GB even when compressed. Classifying images in 13 Scenes proved more manageable.

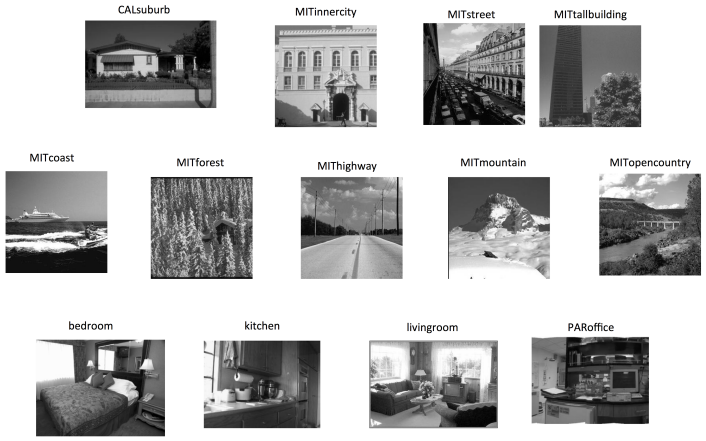

Figure 1: 13Scenes Dataset. Examples from each of the classes.

2 Related Work

Several methods have been developed for representing images in scene recognition. Oliva et al. propose holistic features, e.g. naturalness and ruggedness, to capture a scene's spatial envelope. Training a KNN classifier, they achieve accuracy of 85% on a database of $256x256$ images [11]. More recently, Li et al. use objects as features. They first detect objects in fixed regions over the dataset, and then perform Latent Dirichlet Allocation over detected objects; this achieves 65% accuracy on the 13Scenes dataset [7]. Lazebnik et al. use a kernel method, spatial pyramidal matching, to provide a more hierarchical similarity measure between pictures than the bag of pixels method [14]. Using an SVM classifier, they achieve 81% accuracy on the 15Scenes dataset. They also find that latent factor analysis techniques in [7] hurt performance. Finally, Razavian et al. use OpenFeat, a convolutional neural network (CNN) trained for object recognition, to generate features for scene recognition. They achieve superior performance of 69% on the MIT-67 indoor scenes dataset [13].

3 Methods

3.1 Feature Generation

1. **Naive approaches:** As a performance baseline for the high-dimensional, high-level features mentioned below, we used two naive approaches: flattening an image's greyscale pixels and using corners detected in the image. After using the Harris method [9] to find corners in each scaled image, we clustered corners by their location and formed a bag-of-words representation for each image using these clusters. We have included reduction and classification results for the former below; the latter did not perform significantly differently.
2. **HOG:** Pioneered by Navneet Dalal and Bill Triggs in 2005, the method of using grids of histogram oriented gradient (HOG) descriptors in human identification in images represented a significant advance. These grids, which are locally normalized, are generated by dividing the image window into small spatial regions [and] for eachaccumulating a local 1-D histogram of gradient directions or edge orientations over the pixels of the cell [6]. Each images feature represents the flattening of its gradient grid into one dimension. Because of HOG features' utility in identifying objects [8] and human figures, past analyses [17] have used them in addressing the scene classification problem. Our analysis also seeks to capture this intuition.
3. **SIFT:** SIFT, or Scale-Invariant Feature Transform, identifies objects (described by descriptors) and their locations (or their features) within an image. Introduced in 1999 by David Lowe (formerly of the University of British Columbia, currently of Google), SIFT finds features invariant to image scaling, translation, and rotation [10]. SIFT involves five pri-

mary steps: 1) the detection of scale-space extrema; 2) the localization of keypoints; 3) the assignments of orientations; 4) the creation of descriptors for each keypoint; 5) the matching of keypoints between images. Each individual image's SIFT features are then combined using Fisher Vectors, which use Gaussian Mixture Models to represent the distribution of the detected objects across the whole dataset [3] The Fisher Vector encoding then compares an image's SIFT features to the Gaussian distribution of features with respect to mean and variance [15] We also sought to represent SIFT in a more compact representation: Vector of Linearly Aggregated Descriptors (VLAD). VLAD uses K-means clustering to cluster the training set and then encodes local feature descriptors as a vector of the residuals of the descriptors with respect to the cluster mean. VLAD is thus an ultra-compact image descriptors that is often used for large scale data bases. [2]

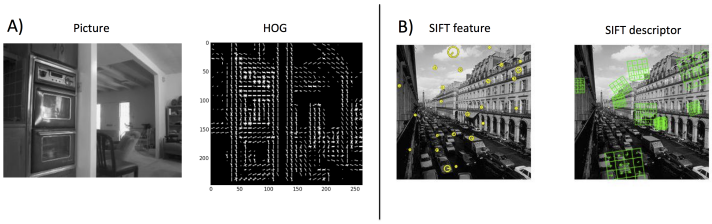

Figure 2: HOG and SIFT features. A) HOG features represent an image by counting occurrences of gradient orientation in localized, connected cells. B) SIFT identifies scale and orientation-invariant keypoints represented by 128-long descriptors, right. The SIFT features represent the scale and orientation of the descriptors, left.

3.2 Dimensionality Reduction Techniques

We first note that the performance of these techniques depend on the type of feature that we feed in. As we will later see, no matter how effective the technique, feeding in poor or insufficiently separated features such as greyscale features can negatively impact the performance. This is because these methods rely on the features' similarity measure between images. If an image of a car shifted left by 1 bit causes a huge change in the greyscale similarity, then the image will also be far away from the original in the reduced dimensions. On the other hand, without using the original image pixels, one cannot interpret the embeddings found by the methods in the original image space (e.g. an embedding that's a linear combination of Fisher Vector indices can't be easily mapped back).

Below we discuss our usage and motivation for the dimensionality reduction techniques. All algorithms used the sklearn implementations. [12] See figure 3 for illustration.

1. PCA. Map onto an orthogonal basis with the directions of highest variance. Advantages: Interpretable; one can analyze the principal components and directions of highest variance; efficient algorithms exist for finding the low-rank approximation. Caveats: it considers the whole data matrix and tries to preserve large pairwise distances on top of small ones, whereas we may care less how different a forest is from a kitchen in our reduced space. In Practice: We use the components that explained at least 90% of the variance. The top 6 explained the top 50% of the variance.
2. t-SNE. Using t-distributions to quantify the similarity between points in the original space, minimize the KL-divergence between the original and reduced probability distributions. Advantages: t-SNE has worked on many types of data (including SIFT on Caltech-101), is fast, focuses on preserving local distances (most relevant for classification), and doesn't penalize outliers as much due to t-distribution. Because of the t-distribution, it also doesn't have the collapsing variance problem that uses outliers to explain the co-variance [16]
3. Isomap. Infer the underlying manifold of the points via nearest-neighbor graph construction. Advantages: interpretable, and consistent, not stochastic. Disadvantages: graph instruction takes $O(D\log(k)N\log(n)$ and shortest path search across the weighted graph

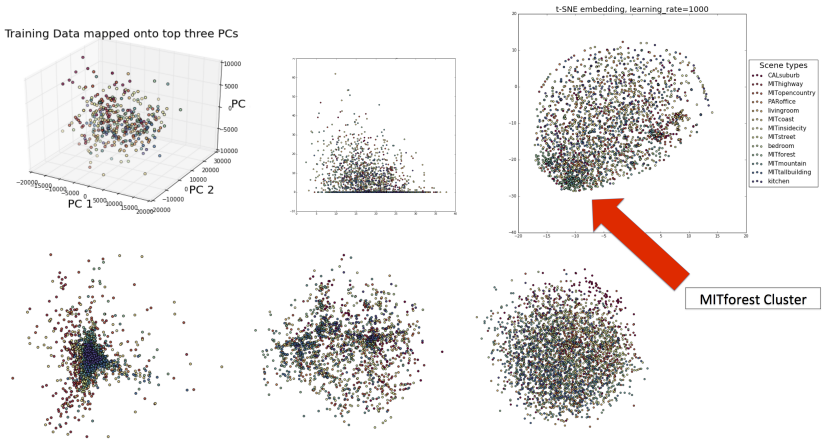

Figure 3: **Embeddings from Dimensionality Reduction Techniques.** Clockwise, from top-left: PCA, Nonnegative Matrix Factorization, t-SNE, MDS, Isomap, Spectral Clustering. For PCA, we see high variance but no clear separation of data. For NMF, many datapoints are set close to 0. t-SNE (on Fisher vectors) finds an intuitive spherical representational of the data, where the forest clusters at one side. This allows it to classify the forest well. MDS also finds a spherical mapping, but it focuses on preserving all distances, not primarily local ones. Isomap finds a more concentrated embedding because it tries to find the distances on the underlying manifold. Spectral clustering seems to suffer from the collapsing variance problem, in which it uses outliers to satisfy the variance constraint on the embedding.[16]

takes $O(N^2 \log n)$, which is not scalable; this is on top of the matrix factorization which takes $O(N^2)$.

4. Nonnegative Matrix Factorization. Advantages: fast and interpretable: data are nonnegative combination of components. Disadvantages: data as nonnegative combination may not apply for greyscale. Also, found factors and components may not be unique. [1] In practice: we set a stopping threshold of gradient norm at 20 instead of 0.0001 due to the $66000+$ dimensions of our data, which necessarily has a higher gradient norm.
5. Spectral Embedding. Performs spectral clustering using the eigenvalues of the similarity matrix.[5] Advantage: also finds a local-distance preserving mapping. Disadvantages: weighted graph constructed in $O(D \log(k) N \log(n)$–not scalable. In practice: we find that it seems to suffer from the collapsing variance problem, in which all the points are clustered together with a few outliers explaining most of the variance. See figure 3.
6. Multidimensional Scaling: a generic term for methods for finding embeddings that minimize ”stress”: dissimilarity of the distances in loss and gain function. Disadvantages: unlike Isomap, MDS doesn't attempt to learn the underlying manifold, of the data but rather preserves the straight line distances of the data. In practice:

3.3 Classification Techniques

Since our goal was to evaluate the reduction technique which allowed best separation of the classes, we didn't want to our analysis to be limited by the classifier we chose. Thus, we tested a variety of classifiers, including K-nearest Neighbors, Decision Trees, RandomForests, AdaBoost, SVM with RBF/linear kernel, and Logistic Regression. We tested a wide range of hyperparameters for each.

4 Results

4.1 Classification Results

HOG features proved to be the best way of representing images for classification, achieving a F1 score of 0.5 and an AUC score of 0.74 when combined with a 10-component PCA reduction and a

Support Vector Machine classifier. The fisher encoding of SIFT features also demonstrated relatively good classification results (a F1 score of 0.31 and an AUC score of 0.63) when combined with a 500-component PCA reduction and a Decision Tree classifier. Finally, greyscale and the VLAD encoding of the SIFT achieved F1 scores of only 0.19 and 0.04 respectively.

Greyscale							
Reduction Type	Time	Components	Best Classifier	Precision (Best)	Recall (Best)	F1 (Best)	AUC (Best)
PCA	10m	500	AdaBoost	0.34	0.17	0.19	0.57
t-SNE	8m	2	DT (depth = 50)	0.19	0.19	0.19	0.56
NMF	3m	10	DT (depth = 50)	0.12	0.12	0.12	0.52
Isomap	30m	2	DT (depth = 50)	0.31	0.13	0.13	0.53
MDS	18m	3	DT (depth = 100)	0.19	0.19	0.19	0.56
S. Embedding	33m	2	DT (depth = 50)	0.15	0.15	0.15	0.54

HOG Features							
Reduction Type	Time	Components	Best Classifier	Precision (Best)	Recall (Best)	F1 (Best)	AUC (Best)
PCA	n/a	10	SVCRBF (C = 18, gamma = 0.19)	0.68	0.49	0.5	0.74
Random Projection	6m	500	SVCRBF (C = 8, gamma = .0005)	0.61	0.43	0.46	0.714

Fisher Vector Encoding of SIFT Features							
Reduction Type	Time	Components	Best Classifier	Precision (Best)	Recall (Best)	F1 (Best)	AUC (Best)
PCA	4m	500	DT (depth = 50)	0.60	0.32	0.31	0.63
t-SNE	44s	2	DT (depth = 100)	0.38	0.17	0.18	0.55
MDS	16m	3	DT (depth = 50)	0.15	0.11	0.11	0.52
Isomap	16m	50	DT (depth = 50)	0.26	0.14	0.14	0.53

VLAD Encoding of SIFT Features							
Reduction Type	Time	Components	Best Classifier	Precision (Best)	Recall (Best)	F1 (Best)	AUC (Best)
PCA	1m	500	RF(depth = 50, estimators = 100)	0.01	0.06	0.02	0.5
t-SNE	1m	2	DT (depth = 100)	0.05	0.06	0.04	0.49

However, both of these more poorly performing methods did have their best classification results when combined with Decision Trees. In fact, Decision Trees were the best classifiers of 10 of the 14 reduced image representations. One possible explanation was that the most pertinent information about an image was whether it included a particular high-level feature. For example, if we learn that an image doesn't include a cloud, we don't learn much about which scene category it belongs too. However, if we learn that an image does include a cloud, it makes it almost certain that we are dealing with one of the outdoor scene categories.

Furthermore, it was surprising that the fisher vector encoding and VLAD encoding of SIFT features achieved such different classification results, despite combining SIFT features with similar comparison algorithms. VLAD's hard assignment have less information than the Fisher Vector's soft-assignments which expressed the data in terms of our 256 Gaussians. Another explanation is the covariance unique to each Gaussian in the GMM are a better representation than Euclidean residuals.Additionally, most of the examples using VLAD for image classification involve extremely large scale datasets, "e.g. 1 billion images," and while our dataset was fairly large, it was not nearly that large.[2] Thus, it is possible that without an enormous training set, the slight differences in residuals due k-means clusters is not sufficient for classification Finally, 100 clusters for VLAD may have been too little to adequately represent the whole data.

The Confusion Matrices in Figure 4 demonstrate how scenes were mis-classified with each of the image representations (paired with their best classifier). Again, it's clear that the HOG features combined with a Support Vector Machine classifier attributed images to the correct scene category

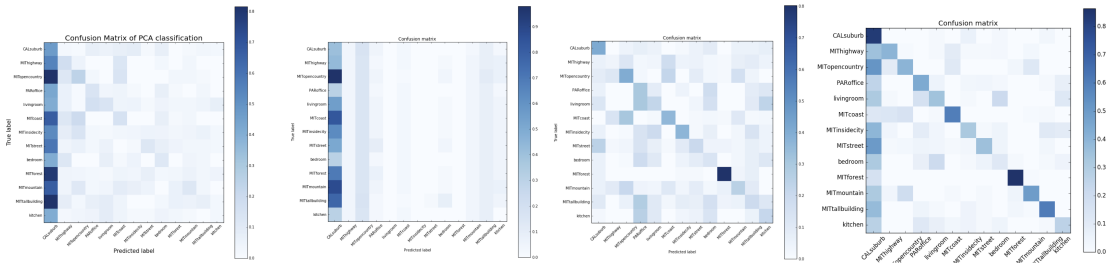

Figure 4: Confusion Matrices. From left to right: greyscale, VLAD, Fisher Vector, and HOG features. Dark blue means a high rate. The dark leftmost columns were default guesses by the classifiers. The darkest square in the two right columns was forest. The dark square in the top-left of the two was suburb. The dark sixth diagonal square for HOG is coast, while the third-to-last and second-to-last squares are mountain and tall building.

more consistently than other image representations. Additionally, we also see that the HOG features, along with the greyscale representation and VLAD encodings, classified a large proportion of images as CALsuburb. Although we originally thought this might have been due to variability within the CALsuburb category, we discovered that this was just an artefact of how the classifiers defaulted. Finally, the two most successful image representations, the HOG features and the fisher vector encodings, were best at classifying images from the forest category. Forest images were accurately classified over 80 percent of the time, and other scene categories were rarely mistaken as forest. We believe this success was due to the fact that scenes from the forest category were the most distinctive. For example, their illumination and viewpoint, the two attributes that define SIFT features, seemed to be the most consistently unique across all of the scene categories.

We note that Li and Perona's theme-based method [7] classified the forest, suburb, mountain, and coasts the best. It is interesting to see that our HOG method performed similarly well on these classes. Similar to use, they use the SIFT features, instead treating them as a bag of words and learning a generative model where scenes optionally contain different classes. Lazebnik et al did the best on suburb, forest, street, and tall building. [14], similar to us. We note that Li and Lazebnik achieved better results, at classification accuracy of $66\% and 81\ \% respectively with hand-tuned features.$

It is surprising that HOG performed better than SIFT because both compute histograms of image gradients. We think the marginal improvement of HOG over SIFT is because SIFT is focused on detecting objects within images, because it finds keypoints of interest via Difference of Gaussian detector (i.e. convolves with two Gaussian kernels and finds extrema) [4]. HoG, on the other hand, uses sliding windows and has been used more for classification than object recognition. This suggests that there were less consistent "objects" for SIFT to find across all the scenes, as may be expected with nature scenes with broad images like forests and lakes, or scenes like tall buildings with large patches occupying the image . In fact, HOG performs better than Fisher Vector in classifying coasts, mountains, and tall buildings.

Finally, we also tried running the full fisher vector encodings through the classifiers to confirm that dimension reduction improved classification accuracy. Although the length of these encodings (65,536) was prohibitively large to run over the entire script of classifiers, we did run the full encodings through the best overall classifier, the Decision Tree. While the reduced dimension fisher vectors achieved an accuracy of over 80 percent in the scenes it classified best, the full fisher vectors only achieved an accuracy of around 50 percent in these same scenes. This demonstrates that the dimension reduction techniques did in fact remove noise from the image representations and isolate a handful of high-level objects as signal.

4.2 Dimensionality Reduction Results

The top results were achieved with PCA and t-SNE for each feature method. t-SNE is particularly impressive for using only 2 dimensions to represent the 4000 data-points. In Figure 3, we see that t-SNE applied to the Fisher Vectors achieves a great separation of the forest class. This may be due to t-SNE's emphasis on preserving local similarities– suggesting the forests were very similar

under the fisher vector SIFT representation. This matches our intuition— forests have generally uniform backgrounds and scaling between high-to-low, unlike living rooms which can have all sorts of different objects, and thus these local similarities work well with t-SNE.

The fastest method was NMF at 3 minutes, due to the lax stopping threshold: gradient norm = 20. When we ran with gradient norm = 0.0001, NMF did not finish, likely because our data was so high-dimensional (¿= 20000) that achieving that gradient norm would require prohibitively small gradient values across all the features. followed by PCA at 8 and t-SNE at 10. PCA has the advantage of using randomized projections to speed things up. t-SNE has the advantage of taking PCA as input. Both methods outperform the manifold methods, which usually must construct some sort of weighted graph on top of calculating eigenvalues.

We could only run Isomap, MDS, and spectral embedding for 2-3 components, as our computers crashed otherwise. These algorithms typically scale linearly in the components number and linearithmic/quadratic in sample number. See section 3.2 for further discussion of their Orders of Growth.

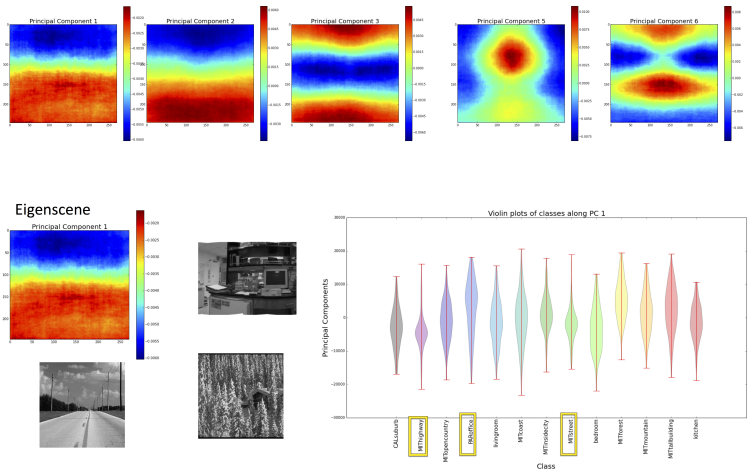

Figure 5: Top: several Eigenscenes on raw images. These explained 50% of the variance. The first two seem to be generic horizons, and explain 33% of the variance. Since PCA assumes that the data are a linear combination of the underlying scenes, and the scenes could tend to be noisy and have different illuminations, one would expect such generic "horizon" scenes. Bottom: Violin Plot of the scene classes across the first Principal Component. According the decision tree, this component was important for classifying the boxed classes: highway, forest, and office. For highway (the purple violin), this is explained by its concentrated values across this component (as one would expect similar horizons across highways).

We have plotted the eigenscenes retrieved by PCA in Figure 5. Since PCA assumes the scenes are linear combinations of the underlying components, we find generic eigenscenes like horizons and, in Eigenscene 3, horizons interspersed with something in between (for example, in a picture of a white house, where the driveway/pavement and the sky above are darker than the house in between). The horizon eigenscene was important for classifying highways, forests, and offices.

We were surprised that NMF performed so poorly, as it has often been used for complex computer vision tasks. We plotted the top several factors we found in Figure 6, both the violin plots and the underlying factors. NMF seems to find similar factors to PCA: the first factor is very bright in the top of the image, suggesting that it may be the sky. Highways have concentrated distributions around the first factor, since they have uniform skies. The third factor is similar to the third principal component: intense in the top and bottom and less in the middle, suggesting the same difference in brightness. Most of the scenes are concentrated around 0, however, suggesting that there is less of a range of values compared to PCA, and thus would result in poor classification. We had consider

5 Discussion and Conclusion

We explored how dimensionality reduction techniques impact scene recognition for the 13Scenes dataset. After generating a host of features, both high-level (e.g. HOG and SIFT) and naive (e.g. flattening greyscaled pixels), we tested how different dimensionality reduction techniques impacted classification accuracy. PCA-reduced HOG features classified best across dimensionality reduction techniques – especially when classifying with support vector machines with RBF kernel.

The best-performing reduction techniques were PCA and t-SNE. We think t-SNE performed well since it focused on local distance preservation and was not sensitive to outliers; thus it focused on scenes that were most similar to each other. The fastest dimensionality reduction techniques were NMF, PCA, and t-SNE. Many of the other methods crashed our computers for higher parameter (e.g. dimension) values, likely because they use graph algorithms that had high computational complexity. NMF did poorly compared to PCA possibly because its factors did not allow a large amount of variance in values.

We consider several possible extensions. We could examine newer and larger labeled scene datasets (like MIT's Places2), to see how these techniques represent even more highly varying images. A larger dataset would potentially make VLAD a much more viable encoding for the images, for VLAD has primarily been used for extremely large scale image classification due its extremely compact representation. [2] We could also try to use power spectral features, which use convolution across the principal components. These may outperform PCA because the convolution is orientation and translation invariant. This would also have the advantage of having interpretable components, unlike many of the other reduction techniques we tried. It would be interesting, also, to try the spatial pyramidal features as in [14] and compare that method to SIFT and HoG which seem similar in the spirit of computing histograms over the image.

Analysis we'd like to have done ourselves include: interpreting the feature importances of the reduced higher level features. E.g. if fisher Vector index 1000 was important, recover the gaussian from which it was calculated and the keypoint descriptor of that gaussian's mean. We also wonder whether concatenating various high-level features, like those derived from techniques such as HOG and SIFT, and could have classified scenes more reliably. We also would like to have compared performance of reduced concatenated features versus features separated reduce, which were then concatenated. Finally, we'd like to have compared the performance of the reduced features to the original features (tens of thousands long) to see how much of a difference the dimension reduction made.

Acknowledgments

We would like to thank the COS424 course instructors and AIs for their invaluable help in conceiving of and shaping this analysis.

References

[1] R. Albright J. Cox A. Langville, C. Meyer and D. Duling.

[2] R. Arandjelovic and A. Zisserman. All about vlad. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 1578–1585, June 2013.

[3] The authors of VLFeat. Vlfeat.org, 2013.

[4] Boris Babenko. Computer vision: What is the difference between hog and sift feature descriptor?

[5] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduciton and data representation, 2002.

[6] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893 vol. 1, June 2005.

[7] Pietro Perona Fei-Fei Li. A bayesian hierarchical model for learning natural scene categories. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 524–531, 2005.

[8] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(9):1627–1645, 2010.

[9] Chris Harris and Mike Stephens. A combined corner and edge detector. Citeseer, 1988.

[10] D. G. Lowe. Object recognition from local scale-invariant features. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 2, pages 1150–1157 vol.2, 1999.

[11] Aude Oliva and Antonio Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42(3):145–175, 2001.

[12] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[13] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. Cnn features off-the-shelf: An astounding baseline for recognition. In *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 512–519, June 2014.

[14] J. Ponce S. Lazebnik, C. Schmid. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 2169–2178, 2006.

[15] Karen Simonyan, Omkar Parkhi, Andrea Vedaldi, and Andrew Zisserman. Fisher vector faces in the wild. *Procedings of the British Machine Vision Conference 2013*, 2013.

[16] Laurens van der Maaten. Visualizing data using t-sne. Youtube video, 2013.

[17] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *Computer vision and pattern recognition (CVPR), 2010 IEEE conference on*, pages 3485–3492. IEEE, 2010.

Supplemental Information

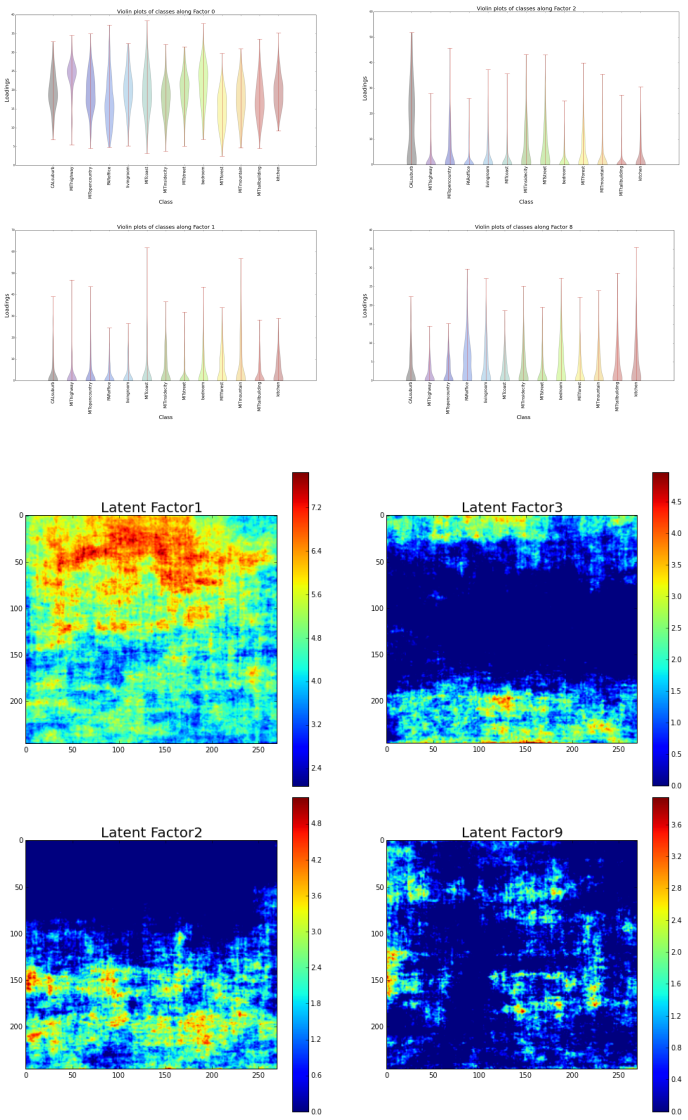

Figure 6: Top: Violin plots of classes over 4 of the NMF factors. Bottom: The corresponding NMF factors, mapped onto greyscale. The first factor is generally bright at the top; this suggests that the first factor represents how bright the sky is in these images. This is consistent with how highways have generally uniform skies. Some of the classes have high loadings, such as CALsuburb in the top-right factor: this seems similar to the factor found in PCA with intense pixels at the top and bottom (sky and pavement) and lighter pixels in the middle (the house). However, most of them had low encoding values with high variance. 10 factors were used in total.