Deep Segments: Comparisons between Scenes and their Constituent Fragments using Deep Learning

Jigar Doshi[1], **Celeste Mason**[2], **Alan Wagner**[3] and **Zsolt Kira**[3]
School of Electrical & Computer Engineering[1]
College of Computing[2]
Georgia Tech Research Institute[3]
jdoshi8@gatech.edu, celeste.m@gatech.edu, alan.wagner@gtri.gatech.edu, zsolt.kira@gtri.gatech.edu

Abstract

We examine the problem of visual scene understanding and abstraction from first person video. This is an important problem and successful approaches would enable complex scene characterization tasks that go beyond classification, for example characterization of novel scenes in terms of previously encountered visual experiences. Our approach utilizes the final layer of a convolutional neural network as a high-level, scene specific, representation which is robust enough to noise to be used with wearable cameras. Researchers have demonstrated the use of convolutional neural networks for object recognition. Inspired by results from cognitive and neuroscience, we use output maps created by a convolutional neural network as a sparse, abstract representation of visual images. Our approach abstracts scenes into constituent segments that can be characterized by the spatial and temporal distribution of objects. We demonstrate the viability of the system on video taken from Google Glass. Experiments examining the ability of the system to determine scene similarity indicate a $\rho\left(384\right) = +0.498$ correlation to human evaluations and 90% accuracy on a category match problem. Finally, we demonstrate high-level scene prediction by showing that the system matches two scenes using only a few initial segments and predicts objects that will appear in subsequent segments.

1 Introduction

Scene understanding from video imagery is a fundamental problem with applications ranging from scene classification to video retrieval and robotics. While a great deal of progress has been made for tasks such as scene classification from single images (Russakovsky et al. 2014) or small video clips (Yang, Liu, and Shah 2009), there is an opportunity to expand scene understanding to encompass a deeper analysis of environments in terms of the types and patterns of human interactions. For example, at a fast-food restaurant customers typically enter the establishment, order food, fill drinks and wait for their orders to be filled, eat, and finally leave. In other words, scenes often afford regular scene fragments (or segments) that can be analyzed, described, compared, and used for prediction. Such regularity allows for additional information beyond objects and their spatial relationships to be used when performing tasks such as scene classification and scene abstraction. More importantly, this analysis could also allow one to predict the objects and scene fragments that will follow, an important task for several applications within robotics.

Recently, it has become possible to collect video from first-person perspective using wearable cameras such as Google Glass. Videos taken from a first-person perspective present a more realistic view of the visual scene in terms of movement and activities within an environment. The video information collected via Google Glass could potentially be applied to robotics or human computer interaction problems.

In order to reason about environments at multiple levels, however, a hierarchical representation that spans from low-level features to high-level objects is necessary. Recently, progress has been made in solving the general problems of object, scene, and activity classification using feature representation learning techniques, where hierarchical feature representations are learned in a data-driven way using methods such as deep learning (Zeiler and Fergus 2013). Specifically, convolutional neural networks have shown great promise in learning a hierarchy of features ranging from low-level filters to object parts and entire objects themselves. Further, these techniques are neuroscientifically inspired and result in features that are similar to those seen in the visual cortex, lending themselves to be used for cognitive tasks where a human will interact with the resulting representation.

In this paper, we leverage these advances in deep learning to analyze scene fragments from first-person video using the spatial and temporal distribution of outputs maps that results from a convolutional neural network. Specifically, we employ a publicly available network that has been pre-trained on a different dataset and used for a completely separate object recognition task (Jia 2013), and develop a novel approach that uses the output of this network to define a distance metric which is then used to evaluate the similarity of different scene segments. Due to the neuro-inspired nature of the algorithm and the fact that a robust set of feature hierarchies are learned, we hypothesize that the resulting distance metric will closely agree with human evaluations in a segment comparison task. Further, we hypothesize that a segment-based view of scenes together with the distance metric can be used to perform scene classification, even using only the first few segments alone. We confirm both of these hypotheses on real-world data from multiple environments recorded using Google Glass. Taken together, these two capabilities result in a system that can robustly represent an environment by spatial and temporal distributions of objects across multiple segments of a scene. Such a representation holds promise for performing higher-level inference of scenes, including prediction of upcoming segments and the corresponding distribution of objects that are likely to be encountered.

2 Related Work

Research in the area of scene classification and recognition has recently become an important field of focus. Researchers have predominately focused on the task of classifying single images in terms of a predefined labeled category (Fei-Fei and Perona 2005; Oliva and Torralba 2001; Juneja et al. 2013; Lazebnik, Schmid, and Ponce 2006; Feichtenhofer, Pinz, and Wildes 2014). Traditionally, these approaches tend to use low-level feature descriptors such as SIFT (Lowe 2004) or HOG (Dalal and Triggs 2005) to characterize the target scene. Xiao et al., for example, utilizes an extensive dataset of different scenes derived from over 130,000 images to test a variety of low-level feature descriptors on the task of scene classification (Xiao et al. 2010).

More recently, convolutional neural networks have been used in which these low-level features are learned (Girshick et al. 2013; Krizhevsky, Sutskever, and Hinton 2012; LeCun, Kavukcuoglu, and Farabet 2010; Sermanet et al. 2013; Simard, Steinkraus, and Platt 2003). Donahue et al., for instance used deep convolutional neural networks to examine the task of scene classification (Donahue et al. 2013). They used *Caffe* to classify scenes from the SUN-397 scene category database obtaining a recognition rate of 40.94% which is currently state-of-the-art. Our approach differs from these other methods in that we use the output maps from *Caffe* convolutional neural network as means for representing the visual locations that one has experienced. This important difference captures the spatial and temporal distributions of objects as a sparse representation. Moreover, we focus on the task of comparing scenes one has experienced in the past to one's current scene in order to use the system for prediction and scene abstraction, rather than simply for recognition.

Researchers have also used object tracking within stationary videos to categorize scenes (Stauffer and Grimson 2000; Wang, Tieu, and Grimson 2006). Our approach differs in that our use output maps generated from a convolutional neural network which are relatively stable across the frames of a video, providing temporal continuity and robustness to blur and camera motion. As shown below, this facet of the research makes our system implementable with a wearable camera. Because one primary goal of this research is to implement this system on a moving first-person camera, we felt that it was inappropriate to compare this work to video and images taken from a stationary camera included within these datasets (Xiao et al. 2010; Lazebnik, Schmid, and Ponce 2006; Quattoni and Torralba 2009). We conjecture that the performance of our system would be similar to that of Donahue et al. had we used the dataset to test the performance of our system.

3 Approach

In this paper, we focus on characterizing scenes at the object and then segment level in order to create a distance metric for scene classification and inference. Specifically, we represent scenes by the identity, strength, and spatial distribution of objects in their constituent segments. In order to do this, we begin with a pre-trained convolutional neural network, which represents the current state of the art in object recognition (Krizhevsky, Sutskever, and Hinton 2012). As is described in Section 3.1, these networks are capable of learning a hierarchy of features from supervised data. The spatially-localized object-level features in the last layers of the network are input to our system, which summarizes their distributions using a Fisher vector representation (Perronnin and Dance 2007). A distance metric is then used to compare these vectors which incorporates segments consisting of multiple frames. This process is described in more detail in Section 3.2.

Intuitively, this system transforms low-level video images into high-level, scene specific, representations which are resistant to noise. Inspired by recent cognitive science research related to concept cells (McClelland, McNaughton, and O'Reilly 1995), we sought to explore the possibility of using the output maps created by the last layer of a deep neural network as a sparse, abstract representation for comparing and understanding categories of locations, such as restaurants .

3.1 Convolutional Neural Networks

Convolutional Neural Networks (CNN) are a class of deep learning architectures that alternate between two stages: 1) The inputs are convolved with learned filters that are then fed through a non-linear function, which results in projecting the input to a high-dimensional space and 2) pooling, which summarizes the output of a group of spatially-coherent output neurons. This technique was first popularized by (LeCun et al. 1998) showing state-of-the-art performance in text recognition and has received significant attention recently because of it substantially higher performance than existing techniques in a variety of tasks including object classification, speech recognition (Dahl et al. 2012), and even text analysis (Simard, Steinkraus, and Platt 2003). In computer vision, high accuracy has been shown on the ImageNet Large Scale Visual Recognition Challenge in particular.

We use a pre-trained neural network as defined in (Krizhevsky, Sutskever, and Hinton 2012) which is trained on the ImageNet dataset made publicly available as part of the *Caffe* (Jia 2013) software package. This network is defined by *60* million parameters and *650,000* neurons, it consists of five convolutional layers and three fully-connected layers. The first convolutional layer filters the 224×224×3 input image with 96 kernels of size 11×11×3 with a stride of 4 pixels (this is the distance between the receptive field centers of neighboring neurons in a kernel map). The second convolutional layer takes as input the (response-normalized and pooled) output of the first convolutional layer and filters it with 256 kernels of size $5 \times 5 \times 48$. The third, fourth, and fifth convolutional layers are connected to one another without any intervening pooling or normalization layers. The third convolutional layer has 384 kernels of size 3 × 3 × 256 connected to the (normalized, pooled) outputs of the second convolutional layer. The fourth convolutional layer has 384 kernels of size 3 × 3 × 192, and the fifth convolutional layer has 256 kernels of size 3 × 3 × 192. The fully-connected layers have 4096 neurons each.

3.2 Characterizing Scene Fragments using CNN Outputs

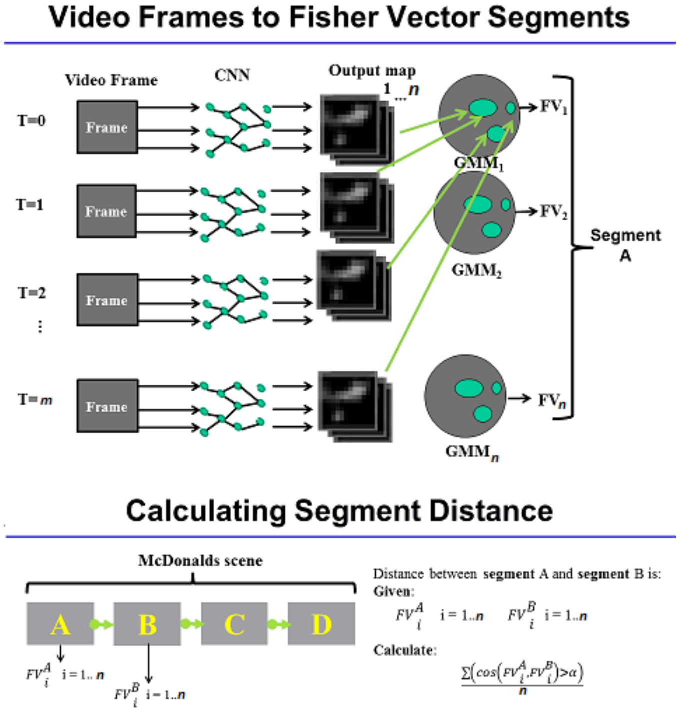

Video Frames to Fisher Vector Segments
Video Frame
CNN
Output map
1 ... n
T=0
T=1
T=2
T=m
Frame
GMM$_1$
GMM$_2$
GMM$_n$
FV$_1$
FV$_2$
FV$_n$
Segment A
Calculating Segment Distance
McDonalds scene
A
B
C
D
FV_i^A i = 1.. n
FV_i^B i = 1..n
Distance between segment A and segment B is:
Given:
FV_i^A i = 1..n FV_i^B i = 1..n
Calculate:
$\frac{\sum(cos(FV_i^A, FV_i^B) > \alpha)}{n}$

Figure 1: The diagram above depicts the process of converting a frame into a Fischer Vector and then our method for calculating the distance between segments

We believe that in order to build a general high-level representation of environments, a rich set of scene elements must be taken into account and that the convolutional neural network's output is able to capture this quality of information. Others have visualized different features that are learned at each layer of the network, starting from simple edge-like features all the way to entire objects (Zeiler and Fergus 2013). While the last layer of the network outputs the final object classification, we use the output maps from the fifth convolutional layer of the network to capture some of the richness of the representation, namely the identity, strength, and spatial distribution of objects throughout the image after pooling. Formally, layer i of a convolutional layer consists of n output maps, corresponding to the n convolutional kernels or filters that are used to transform the input. We denote output map j of layer i as O_j^i, where $j = \{1, 2, ..., n\}$ and this is in general a matrix representing the output of convolving the filter corresponding to a particular feature (i.e. edge filters at the lowest level, object parts higher up, and entire objects at the last layer) with the input. Without losing spatial information, we vectorize this matrix to produce one vector representing the output map. The set of output maps at layer i is then $\mathbf{O^i}$ ={O_1^i,O_2^i,..., O_n^i}.

The output maps described above capture the output of one frame. However, we would like to represent a series of frames in a temporal segment of the video, and later use multiple segments to represent an entire scene. Specifically, we would like to capture the distribution of these output maps across the segment. In order to do that, we first create clusters for each of the output maps by learning a Gaussian Mixture Model (GMM), denoted as $GMM_j^i = EM(\mathbf{O^i})$ for a set of output maps derived from a training dataset, where EM is the Expectation Maximization algorithm. Note that in this work, as described in the experiments section, we use the fifth layer and so $i = 5$, but the framework is not specific to any one layer. The GMM clustering captures the dominant spatial distributions and strengths in the output maps for a particular filter over the training data. Given these cluster means and variances, we then use an Improved Fisher vector (Paolacci, Chandler, and Ipeirotis 2010) encoding which summarizes the strength of association between a *set* of vectors to the different modes in the GMM. In our case, the set of vectors consists of a set of particular output maps j across *multiple frames* which are the segments (or fragments) of the scene. We denote a segment as a set of output maps $\mathbf{S} = \{O_{j,t}^i, O_{j,t+1}^i, ..., O_{j,t+m}^i\}$, where t is the starting time of the segment and $t+m$ is the end time of the segment. We then calculate $F_{j,\mathbf{S}}^i = Fisher(\mathbf{S})$.

Once normalized, the result is an extremely sparse representation with only a few values in the Fisher vector being much larger than zero. Given n of these vectors, we can then define a distance metric between each pair of vectors. As has been argued in the literature (Choi et al. 2014), a good choice for a distance metric for such sparse vectors is the cosine distance, which we use here. Hence, we define the distance between two Fisher vectors as follows:

$$D(F_{\mathbf{S_1}}, F_{\mathbf{S_2}}) = 1 - \frac{F_{\mathbf{S_1}} \cdot F_{\mathbf{S_2}}}{\|F_{\mathbf{S_1}}\| \|F_{\mathbf{S_2}}\|} \tag{1}$$

Since we get a Fisher vector for each output map, we have n distance values that must be combined. Here, we use a simple percentage of the distance values above a threshold, although in future work we will look at more sophisticated methods. Specifically, we define the distance between two segments as:

$$D(\mathbf{S_1}, \mathbf{S_2}) = \frac{\sum_j \varphi(D(F_{j,\mathbf{S_1}}^i, F_{j,\mathbf{S_2}}^i))}{n} \tag{2}$$

where $\varphi(x)$ is an indicator variable that is one if $x < \tau$ and zero otherwise for some threshold $\tau (= 0.0708)$.

As shown in Figure 1, this architecture starts by subsampling video frames which are fed through a pre-trained network. Next, we extract and cluster the output maps using a set of Gaussian Mixture Models. Finally, using the GMMs and Fisher vector encoding, we encode each segment (consisting of multiple frames) as one Fisher vector. We then

Table 1: Recorded Scenes

Category	Scene	# of Scenes
Fast Food	Burger King, McDonalds, Krystal	3
Library	Georgia Tech (x2), Emory	3
Park	University Park, Public Park	2
Cafeteria	Hospital Cafeteria	1

use the cosine distance between the vectors and accumulate them across all output maps to derive a final distance metric between two segments. We show in the experiments below that this metric validates well with distances provided by humans. For two entire scenes, each of which consists of multiple segments, we can compute a distance matrix. This distance matrix was converted to a similarity matrix $(Similarity = 1 - D(S_1, S_2))$ as it was more intuitive for human subject tests. This similarity matrix is used in the experiments below to demonstrate scene classification.

4 Experiments

4.1 Experimental Setup

We used Google Glass to record videos that captured the interactions and objects that occur in a particular scene from a first-person point of view. For example, when one enters a fast-food restaurant, orders a drink, fills the drink, sits down and consumes the drink, and then leaves. This platform was chosen mainly because of its small, unobtrusive design which allows a high degree of natural interaction. The Google Glass camera operates at a resolution of 1280×720 and a frame rate of 30 fps. The videos that resulted were not altered or preprocessed to improve quality. Nine different videos between 3-7 minutes long were recorded by two different experimenters acting independently and at different times and days. These experimenters acted naturally as they interacted with people and objects in the scene. We chose scenes that matched different categories of locations, such as fast-food resturants, parks, and libraries. Table(1) describes each of the nine scenes. The videos of each of the scenes were then divided into 20 second segments. We divided the scenes into segments because early experiments indicated that portions of the 3-7 minute videos matched more strongly than others. Twenty second segments were used because it was felt that this length was long enough to capture a meaningful, moment-to-moment snapshot of the scene.

Once the videos were collected, they served as test data for the experiments described below. In order to reduce the processing time, we only selected the first frame for every second of video. Each frame was then passed as input to the *Caffe* convolutional neural network (Jia 2013). The runtime to feed a frame through the network was approximately 350 milliseconds. *Caffe* is pretrained on the ImageNet dataset allowing the neural network to recognize 1000 objects (Deng et al. 2012).

4.2 Comparisons to Human Evaluations

The first experiment compared the system's estimate of scene segment similarity to evaluations made by people. Because there are many different ways to judge similarity, the purpose of this experiment was to establish, to the extent possible, ground truth on which we could gauge the accuracy of the system. We hypothesized that the system's estimate of similarity would strongly correlate to the estimates made by human subjects.

Crowdsourcing was used to obtain human subject evaluations of segment similarity. Crowdsourcing is a method for collecting data from a relatively large, diverse set of people (Paolacci, Chandler, and Ipeirotis 2010). Crowdsourcing sites, like Amazon's Mechanical Turk, post potential jobs for crowdworkers, manage worker payment, and worker reputation. The use of crowdworkers offers a quick and efficient complement to traditional laboratory experiments. Moreover, the population of workers tends to be somewhat more diverse than traditional American university undergraduates. In order to ensure the best possible data, individuals were required to have a 95% acceptance rate for their past work and were only allowed to participate once. To ensure thoughtful evaluations, each worker was asked to briefly describe their rationale behind each score they assigned. Participants were paid an effective hourly rate of $8.87 and IRB approval for the experiment was granted. Approximately 10% of the surveys were rejected because of a failure to follow the instructions or accept the consent agreement. In the end, the data used originated from 224 different people.

The nine scenes from Table(1) were divided into 20 second segments resulting in a total of 161 different segments. Subjects were presented with one 20 second target video segment and four randomly chosen segments. They were asked to rate the similarity of each randomly chosen segment to the target on a scale of (1-10) and to briefly describe their rationale. Once this task was complete they were then presented with a different target and set of different randomly choose segments for evaluation. Each participant evaluated 2 groupings of target and random segments. Due to human subject limitations, only 64 of the possible 161 segments were used in this this experiment. Overall, each target-random segment combination was evaluated once by seven different people.

We compared the evaluations made by the study's participants to the evaluations made by our system in several different ways. For direct comparisons of whether or not one segment matched the target more or less than another segment, we found a 70% agreement between our system and the participants' evaluations. This level of agreement is statistically significant from a random baseline $p < 0.001$. We found a $\rho\,(384) = +0.498$, correlation between the similarity scores generated by our system and those of the participants. For data involving human subjects, this represents a strong, positive correlation (Hemphill 2003) and supports our first hypothesis.

As expected, for some target-to-segment comparisons, there was little or no consensus across the participants in terms of similarity score. We arbitrarily defined high consensus target-to-segment evaluations as those in which the inter-rater standard deviation was less than 2. For these high consensus evaluations, the correlation with our system was $\rho\,(249) = +0.609$. The percent match was 76.3%, which is statistically significant $p < 0.001$.

The results support our contention that the segment similarity evaluations made by the proposed system correlate to those made by people. Yet, the results do not show the extent to which segments and scenes clusters around a category, such as restaurant. We therefore conducted a segment matching experiment.

4.3 Segment Matching

Given the correlation to human evaluations, we wondered if the distances generated by the system could be used for segment matching. If so, it might then be possible to use the system to match one's current scene to a previous scene and predict upcoming objects and events. We hypothesized that segments that originate from the same general category (restaurant, park, etc.) would be more similar than segments from different general categories. In other words, we believed that the system would generate clusters of segments which matched the general categories listed in Table(1). In order to test this hypothesis the system was used to generate distances between the 161 segments created in the experimental setup from Section 4.1.

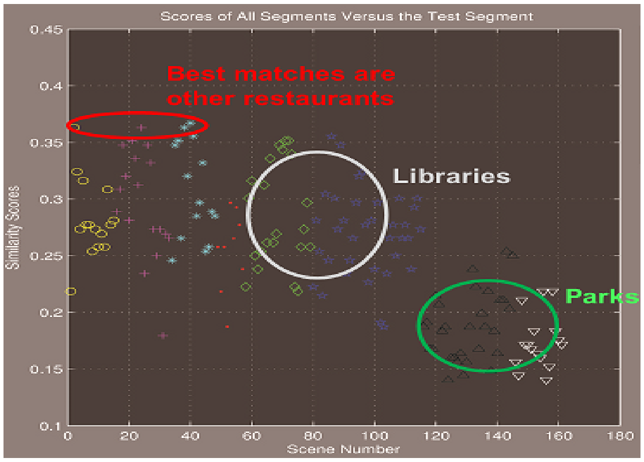
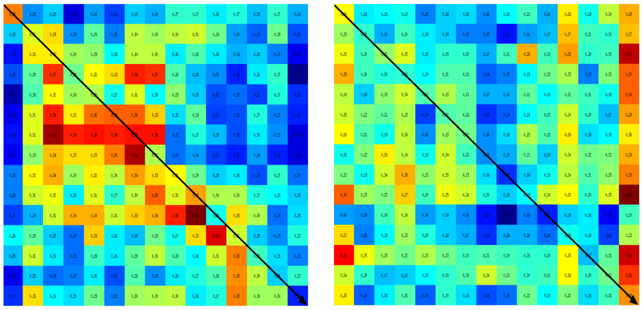


Figure 2: Segment Clustering across all 9 scenes. The graph depicts similarity scores calculated using our system comparing a segment taken from a McDonalds scene to all other segments. The best matches are other fast food resturants.

Figure 2 depicts a representative target segment compared to all 161 segments across all 9 scenes. The x-axis depicts the segment number and the marker type depicts the scene category. The y-axis indicates the similarity score. The target segment was from the McDonalds restaurant scene. As depicted in the figure, the best matches are to the other fast food restaurants. The worst matches are from segments from park scenes. The figure also shows a skew of similarity across the scene. In other words, a range of match similarities occur with respect to the scene itself, the best of which occur in scenes from the same category. For the McDonalds scene, the best match was from a member of the restaurant category in 13 of the 15 segments.

Figure(3a) compares the Burger King scene to every segment from the McDonalds scene. The top left of the graph depicts segment 1 from each scene and bottom right depicts segment n for both scenes. The highest matching scores are located along the diagonal. This indicates that the strongest matches occur in corresponding segments from each of these scenes. By contrast, the Figure(3b) compares the Burger King scene to one of the Park scenes. In this case, the top left to bottom right diagonal does not indicate a strong match. Strong matches do arise, however, in the first and last segments of the Burger King video. These matches result from video taken in the parking lot as the experimenter exits their car and enters the restaurant or leaves the restaurant. This Burger King restaurant's parking lot had several trees and bushes which were matched to the trees and bushes typically seen in a park.

The results support our hypothesis that scenes originating from the same category generate the best matches and that the best matches correspond to similar segments within a scene. One of the goals of this project is to explore how well a segment or a group of segments match to previously observed segments. Furthermore, we would like to predict future environments that will be encountered. Our preliminary experiment shows promising results.



(a) Restaurant vs Restaurant

(b) Restaurant vs Park

Figure 3: Pairwise Segment by Segment Scene Comparison

4.4 Episode Matching and Prediction

A system which can match its current scene to previously encountered scenes and is able to use this information to predict which objects it will encounter in the near future would be an important step towards creating systems which can use their previous first-person video experience to inform future plans. With this goal in mind, we conducted a final experiment that examined if the initial segments from a scene could be used to match a series of segments to a scene from the same category. We hypothesized that the initial segments from a scene could be used to match the currently experienced scene to the correct category. We then tested whether the system could be used to predict which objects the individual will encounter and if those objects are indeed encountered.

This experiment followed the same general setup as before (Section 4.1). In this case, however, one target scene from Table(1) was withheld to represent the current scene. The system used the first *x* segments from the current scene to locate a match to the other 8 scenes. A correct match was determined by calculating the similarity scores to all of the other 161 segments. The scene that the most similar segment originated from was considered a correct match if the scene originated from the same category as the current scene. In order to simulate a real-world use case we varied the number of segments in the current scene. We hypothesized that as the number of segments in the current scene increased the percent match would also increase.

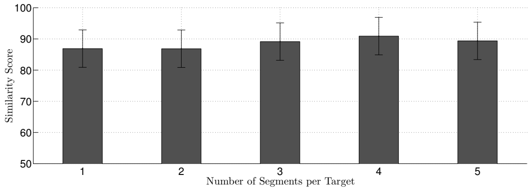


Figure 4: The x-axis presents the number of segments that constituted the target. The y-axis is a similarity score.

Figure 4 depicts the results from this experiment. The results show that as the number of segments used to select a match increase the percent match increases slightly from 89.90% to 90.91 %. Although the increase is slight, the percent match across all cases far exceeds a random baseline of 33%. We believe that the high initial match rate and low number of categories, prevented improvement as segments were added.

In a final experiment we examined the system's ability to predict future objects from a currently experienced scene. Here again one target scene from Table(1) was withheld to represent the current scene. The system used only a single segment from the scene to select the closest matching scene. The matched scene was then used to predict which of 1000 possible objects would occur in the future segments of the current scene. With respect to object labels, the output from the convolutional neural network produces the probability that a particular object appeared. We only considered objects as appearing in a segment if they were assigned a probability greater than 0.001. Next, for each remaining segment, we calculated the intersection of object labels predicted by the convolutional neural network and the object labels generated by the next segment from the target scene. This intersection was divided by their union to get a score. When averaged over all segments' scores, we found an overall $94.76 \pm 0.18\%$ correct prediction rate. This result strongly supports our contention that, at least for the setup tested, the system can be used to predict which objects will occur.

Overall, the results from these experiments are meant to convey the breadth of what we expect this approach to achieve. The results hint that the system can be used for prediction and to match one's experiences in a particular environment to a category of environments and previous experiences. Moreover, because the experiments used raw data taken from a wearable computer, we have reason to believe that practical applications derived from the system are achievable in the near-term.

5 Conclusions

This article has presented a system that uses the output maps generated from a convolutional neural network with camera video input as a representation of the objects and their spatial location in a scene. Methods for calculating the distance between these output maps afford a means for determining scene similarity and/or distance. We have demonstrated that this similarity metric correlates to the similarity judgments of people, that if the scene is broken into segments then these similarity scores can be used to cluster scenes with respect to abstract categories, and that the system offers a potential means for predicting future encounters with objects.

The experimental results that we present are based on nine different scenes. Currently it takes approximately a day to process additional scenes for matching, which is a limitation of the current system. The time to find a match for a scene, on the other hand, is approximately 2-4 seconds, which may be a limitation for applications that require fast scene recognition. The system does not, however, appear to be very sensitive to blur or noise. Because of naturally occurring head motion, the videos we captured tended to be blurry. Yet we choose not to deblur the videos or to remove blurred frames. The video was input to the system without further processing. The fact that we tested on a consumer grade hardware and that the system did not require preprocessing to reduce blur is testament to the robustness of our approach.

To the best of our knowledge, this work represents the first time that the output maps from a convolutional neural network have been used as a sparse representation of the visual environment for the purpose of scene understanding. We believe that this innovation will afford a means of using visual information to reason about one's location, for visual planning, and, perhaps, for higher-level scene abstraction.

The results suggest several potential practical applications. For instance, a search engine which takes scene segments and returns similar scenes appears easily achievable. The system might also afford a new method for vision and landmark based navigation to aid a person, or for a robot. We intend to develop and test such a system in the coming months. The possibility of understanding one's context and of using one's experience to reason about the consequences of an action would a large step towards developing an artificially intelligent system. The research presented here offers a novel method aimed in that direction.

6 Acknowledgment

This work was funded by award #N00141210484 from the Office of Naval Research. We would like to thank Vinay Bettadapura in helping collect our initial batch of videos.

References

Choi, J.; Cho, H.; Kwac, J.; and Davis, L. S. 2014. Toward sparse coding on cosine distance. In *International Conferencerence on Pattern Recognition*.

Dahl, G. E.; Yu, D.; Deng, L.; and Acero, A. 2012. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *Audio, Speech, and Language Processing, IEEE Transactions on* 20(1):30–42.

Dalal, N., and Triggs, B. 2005. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, 886–893 vol. 1.

Deng, J.; Berg, A.; Satheesh, S.; Su, H.; Khosla, A.; and Fei-Fei, L. 2012. Imagenet large scale visual recognition competition 2012 (ilsvrc2012). `http://www.image-net.org/challenges/LSVRC/2012/`.

Donahue, J.; Jia, Y.; Vinyals, O.; Hoffman, J.; Zhang, N.; Tzeng, E.; and Darrell, T. 2013. Decaf: A deep convolutional activation feature for generic visual recognition. *arXiv preprint arXiv:1310.1531*.

Fei-Fei, L., and Perona, P. 2005. A bayesian hierarchical model for learning natural scene categories. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, 524–531. IEEE.

Feichtenhofer, C.; Pinz, A.; and Wildes, R. P. 2014. Bags of spacetime energies for dynamic scene recognition. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, 966–974. IEEE.

Girshick, R. B.; Donahue, J.; Darrell, T.; and Malik, J. 2013. Rich feature hierarchies for accurate object detection and semantic segmentation. *CoRR* abs/1311.2524.

Hemphill, J. F. 2003. Interpreting the magnitudes of correlation coefficients.

Jia, Y. 2013. Caffe: An open source convolutional architecture for fast feature embedding. `http://caffe.berkeleyvision.org/`.

Juneja, M.; Vedaldi, A.; Jawahar, C.; and Zisserman, A. 2013. Blocks that shout: Distinctive parts for scene classification. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, 923–930. IEEE.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In Pereira, F.; Burges, C.; Bottou, L.; and Weinberger, K., eds., *Advances in Neural Information Processing Systems 25*. Curran Associates, Inc. 1097–1105.

Lazebnik, S.; Schmid, C.; and Ponce, J. 2006. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, 2169–2178. IEEE.

LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86(11):2278–2324.

LeCun, Y.; Kavukcuoglu, K.; and Farabet, C. 2010. Convolutional networks and applications in vision. In *Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on*, 253–256. IEEE.

Lowe, D. G. 2004. Distinctive image features from scale-invariant keypoints. *International journal of computer vision* 60(2):91–110.

McClelland, J. L.; McNaughton, B. L.; and O'Reilly, R. C. 1995. Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory. *Psychological review* 102(3):419.

Oliva, A., and Torralba, A. 2001. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International journal of computer vision* 42(3):145–175.

Paolacci, G.; Chandler, J.; and Ipeirotis, P. G. 2010. Running experiments on amazon mechanical turk. *Judgment and Decision making* 5(5):411–419.

Perronnin, F., and Dance, C. 2007. Fisher kernels on visual vocabularies for image categorization. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, 1–8. IEEE.

Quattoni, A., and Torralba, A. 2009. Recognizing indoor scenes. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, 413–420.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; Berg, A. C.; and Fei-Fei, L. 2014. Imagenet large scale visual recognition challenge.

Sermanet, P.; Eigen, D.; Zhang, X.; Mathieu, M.; Fergus, R.; and LeCun, Y. 2013. Overfeat: Integrated recognition, localization and detection using convolutional networks. *CoRR* abs/1312.6229.

Simard, P. Y.; Steinkraus, D.; and Platt, J. C. 2003. Best practices for convolutional neural networks applied to visual document analysis. In *2013 12th International Conference on Document Analysis and Recognition*, volume 2, 958–958. IEEE Computer Society.

Stauffer, C., and Grimson, W. E. L. 2000. Learning patterns of activity using real-time tracking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 22(8):747–757.

Wang, X.; Tieu, K.; and Grimson, E. 2006. Learning semantic scene models by trajectory analysis. In *Computer Vision–ECCV 2006*. Springer. 110–123.

Xiao, J.; Hays, J.; Ehinger, K. A.; Oliva, A.; and Torralba, A. 2010. Sun database: Large-scale scene recognition from abbey to zoo. In *Computer vision and pattern recognition (CVPR), 2010 IEEE conference on*, 3485–3492. IEEE.

Yang, Y.; Liu, J.; and Shah, M. 2009. Video scene understanding using multi-scale analysis. In *Computer Vision, 2009 IEEE 12th International Conference on*, 1669–1676.

Zeiler, M. D., and Fergus, R. 2013. Visualizing and understanding convolutional neural networks. *arXiv preprint arXiv:1311.2901*.