Survey : Improving Recall using CNNs

Swapnil Taneja
University Of California, San Diego
9450 Gilman Drive, La Jolla
San Diego, CA , USA
swtaneja@ucsd.edu

Arun Kumar
University Of California, San Diego
9450 Gilman Drive, La Jolla
San Diego, CA , USA
arunkk@ucsd.edu

ABSTRACT

This paper provides a survey of a number of research papers and an insight into the approaches that can be made for improving recall in an image similarity search using Convolutional Neural Networks. Various researches are being done to understand the features that are learned during training . Neural Networks learn internal representations in the service of the task. In the previous approaches for image instance retrieval , features were extracted from the images and were used to improve the recall. SIFT [11], HOG [4] , SURF [2], Fisher Vectors [15] , to name a few were the features that provided a means to compare images from the dataset with the features of the query image. This paper also surveys briefly the challenges that may come up in addressing this research problem. For example, the internal representations of the hidden units have large number of parameters. Neural Networks since the very beginning have faced this challenge where there is utmost requirement of a Model and Database Management System or Versioning System for addressing this storage problem. Nevertheless, more focus will be on the motivations to gain insights into the distributed features which can be used for improving recall.

Categories and Subject Descriptors

H.4 [**Convolutional Neural Networks**]: Miscellaneous; D.2.8 [**Database Management System**]: Recall—*performance measures*

1. INTRODUCTION

In this project, we explored the concepts of Convolutional Neural Networks. CNNs can be involved in numerous tasks . Image Classification , Object Detection , Object Localization, Image Segmentation , Text Categorization, Spectral Graph Theory, Optical Flow etc. In the ILSVRC challenge of 2010 , Alex Krizhevsky proposed ALexNet [10]. In 2014 [17], VGG 16, and 19 models were proposed. They achieved remarkable results in Top-1 and Top-5 accuracy . The concept behind this survey is to identify methods that can potentially help to improve recall of images in image instance retrieval . The purpose of identifying the taxonomy is to understand and explore the ways these tasks have been performed using internal representations of these neural networks. The next section considers a few of these tasks and explore the exploitable features. Section 2.1 and , 2.3 present the tasks of the taxonomy. Section 2.2 , 2.4 and 2.5 do a survey of existing researches in this field.

2. IMAGE CLASSIFICATION , LOCALISATION AND SEGMENTATION

Image Classification is essentially identifying a class label for an image from a set of pre-determined classes. In the ImageNet ILSVRC-2010 contest, Alex Krizhevsky et. al [10] trained a large deep convolutional neural network to classify 1.2 million high resolution images. On the test data they achieved top-1 and top-5 error rates of 37.5 % and 17 % respectively. Figure 1 shows the model and network architecture of AlexNet[10]. They used non linear hidden units - RELU units and devised a methodology called Dropout for reducing overfitting .

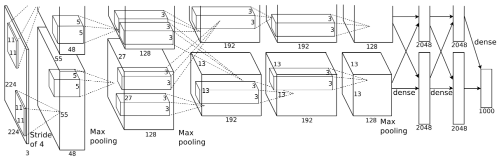


Figure 2: An illustration of the architecture of our CNN, explicitly showing the delineation of responsibilities between the two GPUs. One GPU runs the layer-parts at the top of the figure while the other runs the layer-parts at the bottom. The GPUs communicate only at certain layers. The network's input is 150,528-dimensional, and the number of neurons in the network's remaining layers is given by 253,440–186,624–64,896–64,896–43,264–4096–4096–1000.

Figure 1:

IN ILSVRC 2014 , Karen Simonyan and Andrew Zisserman [17] proposed 6 configurations . They released the two best performing models - one with 16 layers and other with 19 layers. During training, the input to the ConvNets is a fixed-size 224 ÃŮ 224 RGB image .The only pre-processing they do is subtracting the mean RGB value, computed on the training set, from each pixel. A stack of convolutional layers (which has a different depth in different architectures) is followed by three Fully-Connected (FC) layers: the first two have 4096 channels each, the third performs 1000-way ILSVRC classification and thus contains 1000 channels (one for each class). The final layer is the soft-max layer. The configuration of the fully connected layers is the same in all networks. Their best single-network performance on the validation set is 24.8 % / 7.5 % top-1/top-5 error . **Convnet Fusion** : - In another part of the experiments, they combined the outputs of several models by averaging their soft-max class posteriors. This improved the performance due to complementarity of the models, and was used in the top ILSVRC submissions in 2012 (Krizhevsky et al., 2012) and 2013 (Zeiler and Fergus, 2013; Sermanet et al.,2014). Figure 2 shows the results after performing this fusion .

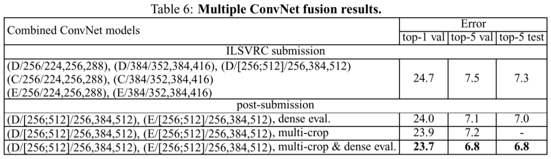


Table 6: **Multiple ConvNet fusion results.**

Combined ConvNet models	Error		
	top-1 val	top-5 val	top-5 test
ILSVRC submission			
(D/256/224,256,288), (D/384/352,384,416), (D/[256;512]/256,384,512) (C/256/224,256,288), (C/384/352,384,416) (E/256/224,256,288), (E/384/352,384,416)	24.7	7.5	7.3
post-submission			
(D/[256;512]/256,384,512), (E/[256;512]/256,384,512), dense eval.	24.0	7.1	7.0
(D/[256;512]/256,384,512), (E/[256;512]/256,384,512), multi-crop	23.9	7.2	-
(D/[256;512]/256,384,512), (E/[256;512]/256,384,512), multi-crop & dense eval.	**23.7**	**6.8**	**6.8**

Figure 2:

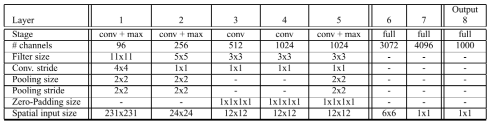
Localization [17] can be seen as a special case of object detection, where a single object bounding box should be predicted for each of the top-5 classes, irrespective of the actual number of objects of the class.To perform object localization, Karen Simonyan et. al used a Very Deep Convnet, where the last fully connected layer predicts the bounding box location instead of the class scores. A bounding box is represented by a 4-D vector storing its center coordinates, width, and height. Training of localization ConvNets is similar to that of the classification ConvNets. The main difference is that the logistic regression objective is replaced with a Euclidean loss, which penalizes the deviation of the predicted bounding box parameters from the ground-truth. They had two testing protocols[17]. In one, the bounding box is obtained by applying the network only to the central crop of the image.In the second, fully-fledged, testing procedure is based on the dense application of the localization ConvNet to the whole image, similarly to the classification task . The difference is that instead of the class score map, the output of the last fully-connected layer is a set of bounding box predictions.

To come up with the final prediction, they utilized the greedy merging procedure of Sermanet et al. (2014), which first merges spatially close predictions (by averaging their coordinates), and then rates them based on the class scores, obtained from the classification ConvNet.

In another model Overfeat [16] , Sermanet et. al present an integrated framework for using Convolutional Networks for classification, localization and detection. This integrated framework is the winner of the localization task of the ImageNet Large Scale Visual Recognition Challenge 2013 (ILSVRC 2013) and obtained very competitive results for the detection and classifications tasks.

Sermanet et. al [16] cited that the simplest approach of predicting segments (regions and not the bounding boxes) consists of training the ConvNet to classify the central pixel (or voxel for volumetric images) of its viewing window as a boundary between regions or not. But when the regions must be categorized, it is preferable to perform semantic segmentation. The main idea is to train the ConvNet to classify the central pixel of the viewing window with the category of the object it belongs to, using the window as context for the decision. The advantage of this approach is that the bounding contours need not be rectangles, and the regions need not be well-circumscribed objects.

The localization task is similar to classification in that 5 guesses are allowed per image, but in addition, a bounding box for the predicted object must be returned with each guess [16].The detection task differs from localization in that there can be any number of objects in each image (including zero), and false positives are penalized by the mean average precision . Feature Extractor model Overfeat specifications can be seen in figure 3.



Layer	1	2	3	4	5	6	7	Output 8
Stage	conv + max	conv + max	conv	conv	conv + max	full	full	full
# channels	96	256	512	1024	1024	3072	4096	1000
Filter size	11x11	5x5	3x3	3x3	3x3	-	-	-
Conv. stride	4x4	1x1	1x1	1x1	1x1	-	-	-
Pooling size	2x2	2x2	-	-	2x2	-	-	-
Pooling stride	2x2	2x2	-	-	2x2	-	-	-
Zero-Padding size	-	-	1x1x1x1	1x1x1x1	1x1x1x1	-	-	-
Spatial input size	231x231	24x24	12x12	12x12	12x12	6x6	1x1	1x1

Table 1: **Architecture specifics for *fast* model.** The spatial size of the feature maps depends on the input image size, which varies during our inference step (see Table 5 in the Appendix). Here we show training spatial sizes. Layer 5 is the top convolutional layer. Subsequent layers are fully connected, and applied in sliding window fashion at test time. The fully-connected layers can also be seen as 1x1 convolutions in a spatial setting.

Figure 3:

The above mentioned state of the art CNNs have been used for image classification, localization, segmentation and detection . The purpose of introducing them here is to gain insight into or find evidence of usage of the internal representation of features. As the fusion of final layers of the ConvNets improved the performance of this task, we hypothesize that fusion of internal layers can provide better performance in improving precision or recall. The questions to ponder here are - How transferable are the features ? What layers are transferable ? How much of the fusion is useful ? How should the fusion be done ? Does weighted averaging provide better results rather than simply averaging ? Can we merge different layers of different ConvNets ? Can we perform the merge simply or by transforming into different domains ? Should the principal components be merged ? Should we worry about these questions or should we create additional layers in the neural network and let Backpropagation take control ? In other words should we employ neural encoders for performing transformation of internal features ? , etc. A few of these questions are answered to a certain extent in the next sections. In the next few sections, we provide a few research works and give insight into how can they be useful .

2.1 Content-Based Image Retrieval

Earlier approaches for content-based image retrieval were based on encoding the images into binary codes. This is equivalent to saying that images were hashed into a binary code so that image similarity measures can be formulated along with dealing in memory constraints. In the paper [18], Torralba et. al proposed methods to hash the GIST descriptors of images. Perhaps the state-of-the-art method to obtain compact binary descriptors for querying a large database is Locality Sensitive Hashing (LSH), which finds nearest neighbors of points lying in a high dimensional Euclidean space in constant time. LSH does this by computing a hash function for a point by rounding a number of random projections of that point into R^1.

Given an input image, a GIST descriptor [18] is computed by first convolving the image with 32 Gabor filters at 4 scales, 8 orientations, producing 32 feature maps of the same size of the input image, then by dividing each feature map into 16 regions (by a 4x4 grid), and then average the feature values within each region. Thereafter, the 16 averaged values of all 32 feature maps are concatenated, resulting in a 16x32=512 GIST descriptor. Intuitively, GIST summarizes the gradient information (scales and orientations) for different parts of an image, which provides a rough description (the gist) of the scene [20]. These descriptors were not practically usage on large datasets due to high dimensionality. The GIST descriptors were converted to a compact binary code using RBMs (Restricted Boltzmann Machines) or Boosting [18].

This algorithm using RBMs is based on the dimensionality reduction framework of Salakhutdinov and Hinton [14] , which uses multiple layers of restricted Boltzmann machines (RBMs).An RBM models an ensemble of binary vectors with a network of stochastic binary units arranged in two layers, one visible, one hidden. Units v in the visible layers are connected via a set of symmetric weights W to units h in the hidden layer. The joint configuration of visible and hidden units has an energy . RBMs can be trained using Contrastive Divergence Sampling Scheme [14].

Their goal was to identify what was the minimal number of bits that they needed to encode an image so that the nearest neighbor defined using a Hamming distance is also a semantically similar image.

Alex Krizhevsky and Georey E. Hinton in their paper [9] used very deep autoencoders to map small color images to short binary codes.They called such models DBNs . Deep Belief Networks are multilayer, stochastic generative models that are created by learning a stack of Restricted Boltzmann Machines (RBMs), each of which is trained by using the hidden activities of the previous RBM as its training data. In the first RBM, most of hidden units learned to be high-frequency monotone Gabor like filters that were balanced in the RGB channels and most of the remaining units became lower frequency filters that responded to color edges. Exploiting the model architecture, they were able to generate 28 bit encodings of the images.

The state of the art SIFT [11], GIST [20], HOG [4], SURF [2] , etc. provide various mechanisms to tap the convolutional features of CNN. To gain further insight into hidden layer features, hierarchical Cluster Analysis, Principal Component Analysis were proposed by elman in paper [6].

Such proposals provide answers to a few questions such as - What are the right transformations of the internal Convolutional features that are not only good for improving precision-recall but are also cost effective in terms of computation and memory ? RBMs and hashing mechanisms can provide a compact binary representation of internal features . For image search, these transformations can provide a cost effective approach.

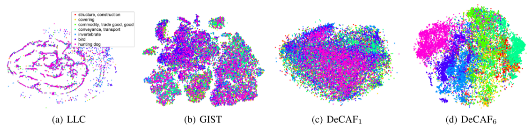
2.2 DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition

In the paper [5] , Jeff Donahue et. al present an approach of tapping out features at different layers of a trained model . The activations of each layer l is referred as DECAF$_l$. They present a methodology of using activations at a particular layer to perform tasks such as Object Recognition, Scene Recognition and Domain Adaptation. They also hypothesized that the activations of the neurons in its late hidden layers might serve as very strong features for a variety of object recognition tasks. In doing so, the following questions were essentially answered - Are the features or activations transferable? It turns out, the answer is ’Yes’ but with certain condition. The condition is that there should be semantic relationship between the data on which it is trained and the data it is tested on.

The deep convolutional model is trained in a fully supervised setting using a state-of-the-art method Krizhevsky et al. (2012). While exploring they additionally address the following questions - Do the features extracted from this model generalize to other datasets ? How do these tapped features perform versus depth ?

Figure 4 shows the visualization from various approaches . Visualizations are done by running the t-SNE algorithm (van der Maaten and Hinton, 2008) [19] to get a 2-dimensional embedding of the high-dimensional feature space, and by plotting them as points colored depending on their semantic category in a particular hierarchy .

The figure shows the features extracted on the validation set using the first pooling layer, and the second to last fully connected layer, showing a clear semantic clustering in the latter but not in the former. This is compatible with common deep learning knowledge that the first layers learn ”low-level” features, whereas the latter layers learn semantic or ”high- level” features. Furthermore, other features such as GIST or LLC fail to capture the semantic difference in the image (although they show interesting clustering structure).



(a) LLC (b) GIST (c) DeCAF$_1$ (d) DeCAF$_6$

This figure shows several t-SNE feature visualizations on the ILSVRC-2012 validation set. (a) LLC , (b) GIST, and features derived from our CNN: (c) DeCAF$_1$, the first pooling layer, and (d) DeCAF$_6$, the second to last hidden layer (best viewed in color).

Figure 4:

More interestingly, in Figure 5 we can see the top performing features (DeCAF$_6$) on the SUN-397 dataset. Even there, the features show very good clustering of semantic classes (e.g., indoor vs. outdoor). The authors claim that these features cluster several intermediate nodes of WordNet implying that these features are an excellent starting point for generalizing to unseen classes.

Additionally , they also give time analysis on different layers. They conclude that the convolution and fully-connected layers take most of the time to run, which is understandable as they involve large matrix-matrix multiplications . In Figure 6 they laid out the computation time spent on individual layers with the most time-consuming layers labeled. Also, the time distribution over different layer types in figure 6 reveals an interesting fact: in large networks such as the current ImageNet CNN model, the last few fully-connected layers require the most computation time as they involve large transform matrices.

To conclude, earlier convolutional layers are unlikely to contain a richer semantic representation than the later higher level features which form higher-level hypotheses. Convolutional layers transition from the low to mid-level local information in their activations. These features could be fused

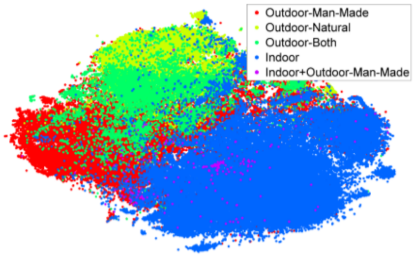


Figure 2. In this figure we show how our features trained on ILSVRC-2012 generalized to SUN-397 when considering semantic groupings of labels (best viewed in color).

Figure 5:

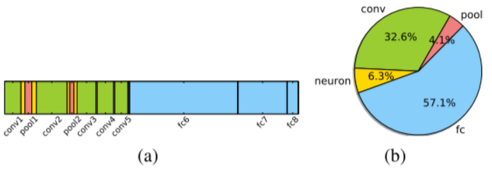


Figure 3. (a) The computation time on each layer when running classification on one single input image. The layers with the most time consumption are labeled. (b) The distribution of computation time over different layer types. In the piechart, fc = fully connected layers, conv = convolution layers, pool = pooling layers, and neuron = neuron layers such as ReLU, sigmoid, and dropout.

Figure 6:

to generate higher representative features in our main goal.

2.3 CAS-CNN: A Deep Convolutional Neural Network for Image Compression Artifact Suppression

Lossy image compression algorithms are used to reduce the size of images transmitted over the web and recorded on data storage media [3]. However, we pay for their high compression rate with visual artifacts degrading the user experience. Deep convolutional neural networks have become a widespread tool to address high-level computer vision tasks very successfully. Recently, they have found their way into the areas of low-level computer vision and image processing to solve regression problems mostly with relatively shallow networks[3].

In this work [3], Lukas Cavigelli et al. present 1) the construction of a new deep convolutional neural network architecture to remove compression artifacts in JPEG compressed image data, 2) a strategy to train this deep network, adaptable to other low-level vision tasks, and 3) extensive evaluations on the LIVE1 dataset, highlighting the properties of the network and showing that this is the current state-of-the-art performance ConvNet for compression artifact suppression (CAS).

Figure 7 shows the model architecture employed for performing compression. The blocks A, . . . ,D each consist of two convolutional layers, increasing the number of channels from 1 to 128 and later to 256, the deeper they are in the network. At the same time the resolution is reduced by down-sampling (DS), which is implemented with 2 X 2 pixel average-pooling layers with 2 X 2 stride. The main path through the ConvNet (marked blue in Figure 7) then proceeds through the full-convolution (also known as up-convolution, deconvolution, backwards convolution, or fractional-strided convolution) layers D, . . . , B (tilde caps) and the normal convolution layer A (cap) .

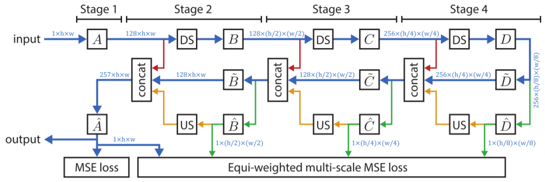


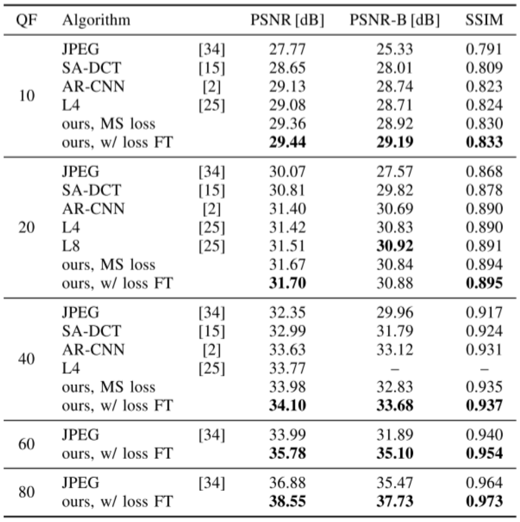
Fig. 1: Structure of the proposed ConvNet. The paths are color coded: main path (bold), concatenation of lower-level features, multi-scale output paths, re-use of multi-scale outputs.

Figure 7:

During the training of the ConvNets they minimize the MSE criterion, penalizing deviations from the reference image by the squared distance [3]. However, in order to improve the training procedure they include not only the full-resolution output, but also the low-resolution outputs from within the network. The reference for these is computed by down-sampling the input image, averaging across 4, 16 and 64 pixels, respectively. Each of these outputs' MSE contributes equally to the overall multi-scale (MS) loss function. These give rise to hierarchical skip connections[3].

The result of their work is a new state-of-the-art ConvNet achieving a boost of up to 1.79 dB in PSNR over ordinary JPEG and showing an improvement of up to 0.36 dB over the best previous ConvNet result[3].

TABLE II: Restoration Quality Comparison on LIVE1



QF	Algorithm		PSNR [dB]	PSNR-B [dB]	SSIM
10	JPEG	[34]	27.77	25.33	0.791
	SA-DCT	[15]	28.65	28.01	0.809
	AR-CNN	[2]	29.13	28.74	0.823
	L4	[25]	29.08	28.71	0.824
	ours, MS loss		29.36	28.92	0.830
	ours, w/ loss FT		**29.44**	**29.19**	**0.833**
20	JPEG	[34]	30.07	27.57	0.868
	SA-DCT	[15]	30.81	29.82	0.878
	AR-CNN	[2]	31.40	30.69	0.890
	L4	[25]	31.42	30.83	0.890
	L8	[25]	31.51	**30.92**	0.891
	ours, MS loss		31.67	30.84	0.894
	ours, w/ loss FT		**31.70**	30.88	**0.895**
40	JPEG	[34]	32.35	29.96	0.917
	SA-DCT	[15]	32.99	31.79	0.924
	AR-CNN	[2]	33.63	33.12	0.931
	L4	[25]	33.77	–	–
	ours, MS loss		33.98	32.83	0.935
	ours, w/ loss FT		**34.10**	**33.68**	**0.937**
60	JPEG	[34]	33.99	31.89	0.940
	ours, w/ loss FT		**35.78**	**35.10**	**0.954**
80	JPEG	[34]	36.88	35.47	0.964
	ours, w/ loss FT		**38.55**	**37.73**	**0.973**

Figure 8:

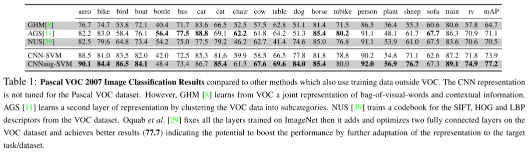
Figure 8 shows us the comparison against different approaches for reconstruction quality on the LIVE1 dataset. The structural similarity index (SSIM)[3]is the mean of the product of three terms assessing similarity in luminance, contrast and structure over multiple localized window. The quality of the restored image is compared with the original High quality image . It is clear from the results that that their model outperforms other approaches . Their work give us an insight that lower level features are transferable to higher layers. By doing they improve on the performance measures and provide a hint that these combined or concatenated features could act as intermediate features for fetching images from the database.

2.4 CNN Features off-the-shelf: an Astounding Baseline for Recognition

In this work [13] Ali Sharif Razavian et. al used the publicly available trained CNN called OverFeat [16] . The structure of this network follows that of Krizhevsky et al [10]. The convolutional layers each contain 96 to 1024 kernels of size 3X3 to 7X7. Half-wave rectification is used as the nonlinear activation function. Max pooling kernels of size 3X3 and 5X5 are used at different layers to build robustness to intra-class deformations. OverFeat[16] was trained for the image classification task of ImageNet ILSVRC 2013 and obtained very competitive results for the classification task of the 2013 challenge and won the localization task. ILSVRC13 contains 1.2 million images which are hand labelled with the presence/absence of 1000 categories. The images are mostly centered and the dataset is considered less challenging in terms of clutter and occlusion than other object recognition datasets such as PASCAL VOC. The thing to remember is that the CNN features used are trained only using ImageNet data though the simple classifiers are trained using images specific to the task's dataset[13].

For all the experiments, they used the first fully connected layer (layer 22) of the network as their feature vector.The feature vector is further L2 normalized to unit length for all the experiments. They used the 4096 dimensional feature vector in combination with a Support Vector Machine (SVM) to solve different classification tasks (CNN-SVM). They further augment the training set by adding cropped and rotated samples and doing component-wise power transform and report separate results (CNNaug+SVM). The tasks tested are - Image Classification, Object Detection, Attribute detection, Fine grained Recognition and Visual Instance Retrieval .

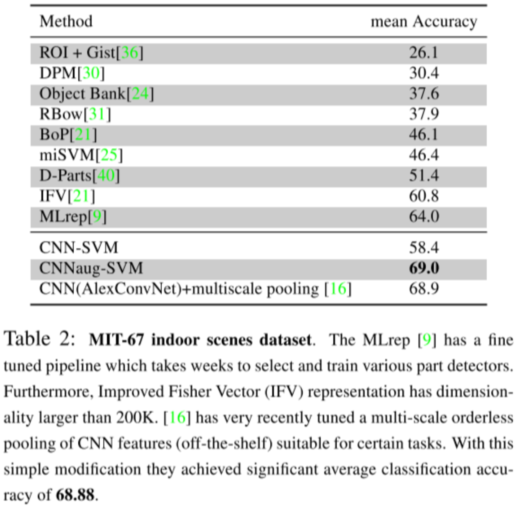
Figure 9 and 10 show the results compared to other approaches for Image Classification on two different datasets Pascal VOC and MIT indoors dataset .



	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
GHM[8]	76.7	74.7	53.8	72.1	40.4	71.7	83.6	66.5	52.5	57.5	62.8	51.1	81.4	71.5	86.5	36.4	55.3	60.6	80.6	57.8	64.7
AGS[11]	82.2	83.0	58.4	76.1	**56.4**	**77.5**	**88.8**	69.1	**62.2**	61.8	64.2	51.3	**85.4**	**80.2**	91.1	48.1	61.7	**67.7**	86.3	70.9	71.1
NUS[39]	82.5	79.6	64.8	73.4	54.2	75.0	77.5	79.2	46.2	62.7	41.4	74.6	85.0	76.8	91.1	53.9	61.0	67.5	83.6	70.6	70.5
CNN-SVM	88.5	81.0	83.5	82.0	42.0	72.5	85.3	81.6	59.9	58.5	66.5	77.8	81.8	78.8	90.2	54.8	71.1	62.6	87.2	71.8	73.9
CNNaug-SVM	**90.1**	**84.4**	**86.5**	**84.1**	48.4	73.4	86.7	**85.4**	61.3	**67.6**	**69.6**	**84.0**	**85.4**	80.0	**92.0**	**56.9**	**76.7**	67.3	**89.1**	**74.9**	**77.2**

Table 1: **Pascal VOC 2007 Image Classification Results** compared to other methods which also use training data outside VOC. The CNN representation is not tuned for the Pascal VOC dataset. However, GHM [8] learns from VOC a joint representation of bag-of-visual-words and contextual information. AGS [11] learns a second layer of representation by clustering the VOC data into subcategories. NUS [39] trains a codebook for the SIFT, HOG and LBP descriptors from the VOC dataset. Oquab *et al*. [29] fixes all the layers trained on ImageNet then it adds and optimizes two fully connected layers on the VOC dataset and achieves better results (**77.7**) indicating the potential to boost the performance by further adaptation of the representation to the target task/dataset.

Figure 9:

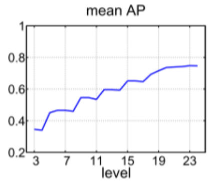


Method	mean Accuracy
ROI + Gist[36]	26.1
DPM[30]	30.4
Object Bank[24]	37.6
RBow[31]	37.9
BoP[21]	46.1
miSVM[25]	46.4
D-Parts[40]	51.4
IFV[21]	60.8
MLrep[9]	64.0
CNN-SVM	58.4
CNNaug-SVM	**69.0**
CNN(AlexConvNet)+multiscale pooling [16]	68.9

Table 2: **MIT-67 indoor scenes dataset**. The MLrep [9] has a fine tuned pipeline which takes weeks to select and train various part detectors. Furthermore, Improved Fisher Vector (IFV) representation has dimensionality larger than 200K. [16] has very recently tuned a multi-scale orderless pooling of CNN features (off-the-shelf) suitable for certain tasks. With this simple modification they achieved significant average classification accuracy of **68.88**.

Figure 10:

The figure 11 shows us the variation of mean average precision as the we go deeper. We can observe that the mAP is increasing for image classification as we increase the level of the layer .



(a)

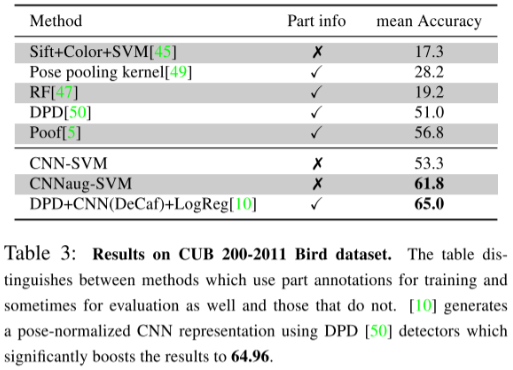
Figure 2: **a)** Evolution of the mean image classification AP over PASCAL VOC 2007 classes as we use a deeper representation from the OverFeat CNN trained on the ILSVRC dataset. OverFeat considers convolution, max pooling, nonlinear activations, etc. as separate layers. The re-occurring decreases in the plot is of the activation function layer which loses information by half rectifying the signal.

Figure 11:

Fine grained recognition[13] has recently become popular due to its huge potential for both commercial and cataloging applications. Fine grained recognition is specially interesting because it involves recognizing subclasses of the same object class such as different bird species, dog breeds, flower types, etc. The figures 12 and 13 show the results on Caltech UCSD birds dataset and Oxford 102 Flowers dataset for Fine grained Recognition.

An attribute within the context of computer vision is defined as some semantic or abstract quality which different instances/categories share. The figures 14 and 15 show the results for attribute detection on UIUC 64 object attributes dataset and H3D Human Attributes dataset .

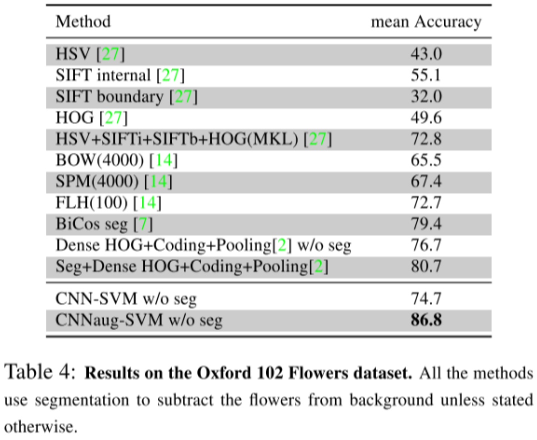
In case of Spatial search [13], the items of interest can appear at different locations and scales in the test and reference images making some form of spatial search necessary. In their crude search for each image they extract multiple sub-patches of different sizes at different locations. For each extracted sub-patch they compute its CNN representation.



Method	Part info	mean Accuracy
Sift+Color+SVM[45]	✗	17.3
Pose pooling kernel[49]	✓	28.2
RF[47]	✓	19.2
DPD[50]	✓	51.0
Poof[5]	✓	56.8
CNN-SVM	✗	53.3
CNNaug-SVM	✗	**61.8**
DPD+CNN(DeCaf)+LogReg[10]	✓	**65.0**

Table 3: **Results on CUB 200-2011 Bird dataset.** The table distinguishes between methods which use part annotations for training and sometimes for evaluation as well and those that do not. [10] generates a pose-normalized CNN representation using DPD [50] detectors which significantly boosts the results to **64.96**.

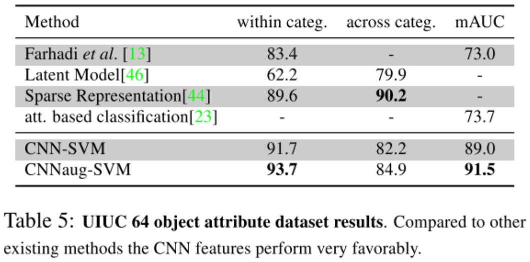
Figure 12:



Method	mean Accuracy
HSV [27]	43.0
SIFT internal [27]	55.1
SIFT boundary [27]	32.0
HOG [27]	49.6
HSV+SIFTi+SIFTb+HOG(MKL) [27]	72.8
BOW(4000) [14]	65.5
SPM(4000) [14]	67.4
FLH(100) [14]	72.7
BiCos seg [7]	79.4
Dense HOG+Coding+Pooling[2] w/o seg	76.7
Seg+Dense HOG+Coding+Pooling[2]	80.7
CNN-SVM w/o seg	74.7
CNNaug-SVM w/o seg	**86.8**

Table 4: **Results on the Oxford 102 Flowers dataset.** All the methods use segmentation to subtract the flowers from background unless stated otherwise.

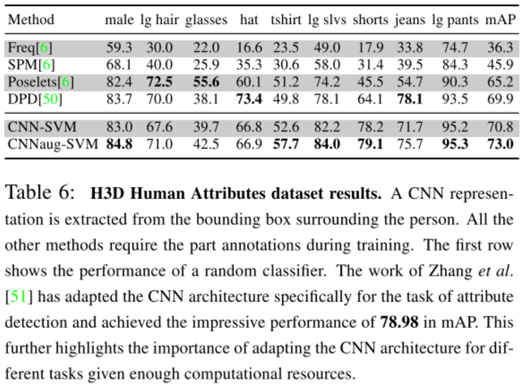
Figure 13:



Method	within categ.	across categ.	mAUC
Farhadi *et al.* [13]	83.4	-	73.0
Latent Model[46]	62.2	79.9	-
Sparse Representation[44]	89.6	**90.2**	-
att. based classification[23]	-	-	73.7
CNN-SVM	91.7	82.2	89.0
CNNaug-SVM	**93.7**	84.9	**91.5**

Table 5: **UIUC 64 object attribute dataset results**. Compared to other existing methods the CNN features perform very favorably.

Figure 14:



Method	male	lg hair	glasses	hat	tshirt	lg slvs	shorts	jeans	lg pants	mAP
Freq[6]	59.3	30.0	22.0	16.6	23.5	49.0	17.9	33.8	74.7	36.3
SPM[6]	68.1	40.0	25.9	35.3	30.6	58.0	31.4	39.5	84.3	45.9
Poselets[6]	82.4	**72.5**	**55.6**	60.1	51.2	74.2	45.5	54.7	90.3	65.2
DPD[50]	83.7	70.0	38.1	**73.4**	49.8	78.1	64.1	**78.1**	93.5	69.9
CNN-SVM	83.0	67.6	39.7	66.8	52.6	82.2	78.2	71.7	95.2	70.8
CNNaug-SVM	**84.8**	71.0	42.5	66.9	**57.7**	**84.0**	**79.1**	75.7	**95.3**	**73.0**

Table 6: **H3D Human Attributes dataset results.** A CNN representation is extracted from the bounding box surrounding the person. All the other methods require the part annotations during training. The first row shows the performance of a random classifier. The work of Zhang *et al.* [51] has adapted the CNN architecture specifically for the task of attribute detection and achieved the impressive performance of **78.98** in mAP. This further highlights the importance of adapting the CNN architecture for different tasks given enough computational resources.

Figure 15:

Successful instance retrieval methods have many feature processing steps. They process the extracted 4096 dim features in the following way: L2 normalize , PCA dimensionality reduction , whitening , L2 renormalization [13]. The figure 16 shows the result of mAP on 5 different datasets for this task of instance retrieval .

2.5 Exploiting Local Features from Deep Networks for Image Retrieval

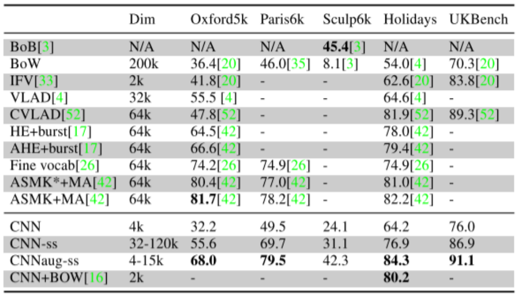
Deep convolutional neural networks have been successfully applied to image classification tasks. When these same networks have been applied to image retrieval, the assumption has been made that the last layers would give the best performance, as they do in classification. Joe Yue-Hei Ng et. al [12] show that for instance-level image retrieval, lower layers often perform better than the last layers in convolutional neural networks. Experiments demonstrate that intermediate layers or higher layers with finer scales produce better results for image retrieval, compared to the last layer. When using compressed 128-D VLAD descriptors, their method obtains state-of-the-art results and outperforms other VLAD and CNN based approaches on two out of three test datasets.

Most existing approaches adopt low-level visual features [12], i.e., SIFT [11] descriptors, and encode them using bag-of-words (BoW) [21], vector locally aggregated descriptors (VLAD) [8] or Fisher vectors (FV) [15] and their variants. Since SIFT descriptors capture local characteristics of objects, such as edges and corners, they are particularly suitable for matching local patterns of objects for instance-level image retrieval.

By default CNNs are trained for classification tasks, where features from the final layer (or higher layers) are usually used for decision because they capture more semantic features for category-level classification. However, local characteristics of objects at the instance level are not well preserved at higher levels [12] . Therefore, it is questionable whether it is best to directly extract features from the final layer or higher layers for instance-level image retrieval, where different objects from the same category need to be separated.

Unlike image classification, which is trained with many labeled data for every category, in instance retrieval generally there is no training data available. Therefore, a pre-trained network is likely to fail to produce good holistic representations that are invariant to translation or viewpoint changes while preserving instance level information. In contrast, local features, which focus on smaller parts of images, are easier to represent and generalize to other object categories while capturing invariance.

The figure 17 captures the overall idea where the features of the internal layers are extracted and transformed into VLAD descriptors .



	Dim	Oxford5k	Paris6k	Sculp6k	Holidays	UKBench
BoB[3]	N/A	N/A	N/A	**45.4**[3]	N/A	N/A
BoW	200k	36.4[20]	46.0[35]	8.1[3]	54.0[4]	70.3[20]
IFV[33]	2k	41.8[20]	-	-	62.6[20]	83.8[20]
VLAD[4]	32k	55.5 [4]	-	-	64.6[4]	-
CVLAD[52]	64k	47.8[52]	-	-	81.9[52]	89.3[52]
HE+burst[17]	64k	64.5[42]	-	-	78.0[42]	-
AHE+burst[17]	64k	66.6[42]	-	-	79.4[42]	-
Fine vocab[26]	64k	74.2[26]	74.9[26]	-	74.9[26]	-
ASMK*+MA[42]	64k	80.4[42]	77.0[42]	-	81.0[42]	-
ASMK+MA[42]	64k	**81.7**[42]	78.2[42]	-	82.2[42]	-
CNN	4k	32.2	49.5	24.1	64.2	76.0
CNN-ss	32-120k	55.6	69.7	31.1	76.9	86.9
CNNaug-ss	4-15k	**68.0**	**79.5**	42.3	**84.3**	**91.1**
CNN+BOW[16]	2k	-	-	-	**80.2**	-

Table 7: **The result of object retrieval on 5 datasets.** All the methods except the CNN have their representation trained on datasets similar to those they report the results on. The spatial search result on Oxford5k,Paris6k and Sculpture6k, are reported for $h_r = 4$ and $h_q = 3$. It can be seen that CNN features, when compared with low-memory footprint methods, produce consistent high results. ASMK+MA [42] and fine-vocab [26] use in order of million codebooks but with various tricks including binarization they reduce the memory foot print to 64k.

Figure 16:

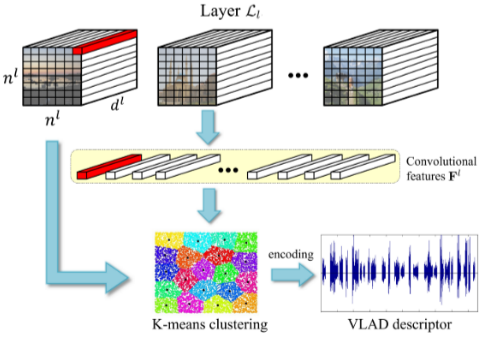


Figure 1: Overview of our feature extraction and encoding.

Figure 17:

At any particular layer l , they extract pixels horizontally in the sense that each pixel (i,j) is picked up from n_l feature maps to formulate a single feature vector. These vectors are transformed using VLAD [8].

Image retrieval is done by calculating the L2 distance between the VLAD descriptors of the query image and database images. They used PCA to compress the original VLAD [8]descriptors to relatively low-dimensional vectors (128-D), so that the computation of L2 distance can be done efficiently [12].

The figure 18 shows the trend observed for mean Average Precision using VLAD encoding at different layers.[12] There is a clear trend in the results of both networks (OxfordNet and GoogleNet) on the first scale (solid lines in the figure). The mAP first increases as we go deeper into the network because the convolutional features achieve more invariance, until reaching a peak. However, the performance at higher layers gradually drops since the features are becoming too generalized and less discriminative for instance-level retrieval.

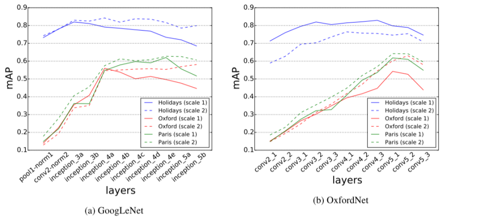


Figure 2: Performance of different layers on both scales: Solid and dash lines correspond to the original and second scale respectively. Fully-connected layers of OxfordNet are omitted due to incompatible size of the last convolutional layer at scale 2.

Figure 18:

There is no reason why this can not be observed for recall. The insight gained here is that depending on the task, the layers deem themselves useful . In other words, Image classification did not show this trend but Image instance retrieval did and hence, an experiment in this domain can most certainly prove innovative.

3. CONCLUSIONS

We come to the conclusion that an experiment in this domain leveraging internal features should most likely provide a better recall. Hopefully, the approaches outlined in this paper provide enough motivation for the project. A couple of other works such as Multi Task CNNs [1] and Hypercolumns [7] give us some more hints to leverage internal distributed representations of the inner layers. Multi task CNNs [1] employed a shared layer of weights across many CNNs. The task in that paper is to perform Attribute Detection for many attributes. In order to do that, authors propose a single CNN for a single attribute. Attributes are also grouped into different groups. The task for each CNN is to make a binary prediction of whether an attribute is present in an image or not. This approach highlights the transferability of CNNs. The transferability comes from a fact that the attributes within the same group share some semantic meaning . This led them to build a CNN with a shared layer . This also made it possible to train some CNN measurably better even if the training samples are not enough. It is because other CNNs already trained the layer to some extent during their own training.

In another work by Bharath Hariharan et. al [7], they proposed a new concept of Hypercolumns. These hypercolumns are essentially the vectors formulated across layers for a single pixel. The number of neurons in each layer are different and hence they perform upsampling of the feature maps. The advantage of these hypercolumns is that they also capture the location semantics of an artifact in an image. We know that the higher layers ,as we go deeper ,do not hold information regarding the specifics of location due to pooling operations. These feature maps get more invariant as they go deep. Hence, they proved that hypercolumns can improve performance for segmentation tasks[7]. This paper discussed various research papers and proposed a few questions to ponder upon . Content Based Image Retrieval gave us a state of the art approach for retrieving images using RBMs . These techniques can be applied to convolutional feature maps for improving recall. Decaf gave us a tool for exploring internal layers and provided some great insight of representative power of hidden units across layers. CAS-

CNN provided a means to compress images while introducing hierarchical skip connections . This raised a question as to whether we can fuse part or whole of the layer representations for generating a better descriptor for a better recall. CNN features off the shelf showed the trend for mean Average Precision for image classification while Exploiting local features showed the trend for Image Instance Retrieval. It is learned that higher level features over generalize and it is better to use intermediate and not high layers for similarity search.

4. REFERENCES

[1] A. H. Abdulnabi, G. Wang, and J. Lu. Multi-task cnn model for attribute prediction.

[2] H. Bay1, T. Tuytelaars, and L. V. Gool. Surf: Speeded up robust features.

[3] L. Cavigelli, P. Hager, and L. Benini. Cas-cnn: A deep convolutional neural network for image compression artifact suppression. *ETH Zurich*.

[4] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection.

[5] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition.

[6] J. L. ELMAN. Distributed representations, simple recurrent networks, and grammatical structure. 1990.

[7] B. Hariharan, P. ArbelÂt'aez, and R. Girshick. Hypercolumns for object segmentation and fine-grained localization.

[8] H. JÂt'egou, M. Douze, C. Schmid, and P. PÂt'erez. Aggregating local descriptors into a compact image representation.

[9] A. Krizhevsky and G. E. Hinton. Using very deep autoencoders for content-based image retrieval.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. 2010.

[11] D. G. Lowe. Distinctive image features from scale-invariant keypoints. January 2004.

[12] J. Y.-H. Ng, F. Yang, and L. S. Davis. Exploiting local features from deep networks for image retrieval.

[13] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition.

[14] R. Salakhutdinov and G. Hinton. *University Of Toronto*.

[15] J. Sancheza, F. Perronnin, T. Mensink, and J. Verbeek. Image classification with the fisher vector: Theory and practice.

[16] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Integrated recognition, localization and detection using convolutional networks. 2014.

[17] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. 2014.

[18] A. Torralba and R. Fergus. Small codes and large image databases for recognition. .

[19] L. van der Maaten. https://lvdmaaten.github.io/tsne/.

[20] Zhang. https://www.quora.com/computer-vision-what-is-a-gist-descriptor.
.

[21] Y. Zhang, R. Jin, and Z.-H. Zhou. Understanding bag-of-words model: A statistical framework.