LEARNING WITH MORE DATA AND BETTER MODELS FOR VISUAL SIMILARITY AND DIFFERENTIATION

Xufeng Han

A dissertation submitted to the faculty of the University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Computer Science.

Chapel Hill
2016

Approved by:

Alexander C. Berg

Tamara L. Berg

Thomas Leung

Leonard McMillan

Marc Niethammer

ABSTRACT

Xufeng Han: Learning with More Data and Better Models for Visual Similarity and Differentiation
(Under the direction of Alexander C. Berg)

This thesis studies machine learning problems involved in visual recognition on a variety of computer vision tasks. It attacks the challenge of scaling-up learning to efficiently handle more training data in object recognition, more noise in brain activation patterns, and learning more capable visual similarity models.

For learning similarity models, one challenge is to capture from data the subtle correlations that preserve the notion of similarity relevant to the task. Most previous work focused on improving feature learning and metric learning separately. Instead, we propose a unified deep-learning modeling framework that jointly optimizes the two through back-propagation. We model the feature mapping using a convolutional neural network and the metric function using a multi-layer fully-connected network. Enabled by large datasets and a sampler to handle the intrinsic imbalance between positive and negative samples, we are able to learn such models efficiently. We apply this approach to patch-based image matching and cross-domain clothing-item matching.

For analyzing activation patterns in images acquired using functional Magnetic Resonance Imaging (fMRI), a technology widely used in neuroscience to study human brain, challenges are small number of examples and high level of noise. The common ways of increasing the signal to noise ratio include adding more repetitions, averaging trials, and analyzing statistics maps solved based on a general linear model. In collaboration with neuroscientists, we developed a machine learning approach that allows us to analyze individual trials directly. This approach uses multi-voxel patterns over regions of interest as feature representation, and helps discover effects previous analyses missed.

For multi-class object recognition, one challenge is learning a non-one-vs-all multi-class classifier with large numbers of categories each with large numbers of examples. A common approach is data parallelization in a synchronized fashion: evenly and randomly distribute the data into splits, learn a full model on each split and average the models. We reformulate the overall learning problem in a consensus

optimization framework and propose a more principled synchronized approach to distributed training. Moreover, we develop an efficient algorithm for solving the sub-problem by reducing it to a standard problem with warm start.

To my parents

ACKNOWLEDGMENTS

I would like to thank my advisor, Dr. Alex Berg, for his guidance and support over the earlier years Stony Brook and later at UNC. I am also grateful to Dr. Tamara Berg, a long-term collaborator and an inspiring mentor. I would also like to thank my thesis committee, for their thoughtful suggestions and helpful criticism, all to make this dissertation a better work.

I feel fortunate and honored to get involved in a number of exciting projects along the way, and I am very grateful to the following collaborators for the knowledge, help and joy they brought to me: Thomas Leung, Yangqing Jia, Rahul Sukthanka, Dave Silver and Abhigit Ogale, during my two internships at Google; Lukas Marti, during a fun and fruitful summer at Apple; Hoi-chung Leung, during our challenging fMRI analysis collaboration; Kota Yamaguchi, Karl Stratos, Margret Mitchell, Hal Daume III, Amit Goyal, Jesse Dodge and Alyssa Mensch, during an inspiring summer workshop at Johns Hopkins University; Rita Goldstein, Nelly Alia-Klein, Muhammad Parvaz, and Tom Maloney, during my adventurous trips to the Brookhaven National Laboratory.

I am proud to be a member of the Berg's group. Over the years I have had the fortune to know, work with and even share a house with some of the amazing members in the group, especially Kota Yamaguchi, Vicente Ordonez, Wei Liu, Hadi Kiapour, Sirion Vittayakorn, and Eunbyung Park.

I would also like to thank mentors and friends from Stony Brook, especially Dr. Dimitris Samaras, Jean Honorio, Yun Zeng, Kiwon Yun, and Yifan Peng, Hojin Choi, and Chen-ping Yu, for making my early PhD years eventful and memorable.

Also thanks to my office mate, Yunchao Gong, as well as members of the V3D Lab at UNC, especially Dr. Jan-Michael Frahm, Jared Heinly, Johannes Schoenberger, Ke Wang, Dinghuang Ji and Enliang Zheng for all the interesting discussions we had.

Special thanks to Ms. Jodie Gregoritsch for all the administrative assistance from months before I arrive at UNC to the last days I am here.

Finally, my deepest gratitude goes to Mom and Dad for all the support, my late grandfather for inspiration and encouragement, and my wife Shun for all her love and care.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Motivation

We need to imagine no further than a coffee shop scenario to see that visual similarity and differentiation are applied in the human vision system every day.

> *To make a medium coffee, a barista picks the topmost cup from a stack of medium-sized paper cups on the busy counter and turn to the coffee machine.*

Before the barista reaches for the cup, the ability to differentiate medium cups from other objects and from cups of other sizes allows her to focus on the right target. At the same time, the ability to match local parts of the images projected into each eye helps her figure out the distance to the cup and plan for the reach.

Inspired by the human vision system, an interesting question is how to make computer vision systems process visual similarity and achieve visual differentiation. Such systems have merit in at least the following three aspects: (1) the ability to differentiate between objects will improve human-computer interaction. A robot with vision could understand what is around it and who it is interacting with much easier than a robot with just speech could, because the user no longer needs to describe everything. (2) They provide models and tools for understanding how the human vision system works. For example, a recent work (Gatys et al., 2015) models artistic style perception in paintings as the correlations of internal filters responses at different processing layers within a neural network trained for object classification. The model is validated by generating highly convincing renderings of an image in the style of different painters. (3) They attract scientific and engineering effort that makes faster, more accurate, and more reliable systems than the human vision system.

Such systems often consist of two types of models: the *representation model* and the *decision model*. The representation model encodes knowledge of how to transform visual entities in their original representation into a different one that makes the task easier. It often achieves this by selecting information

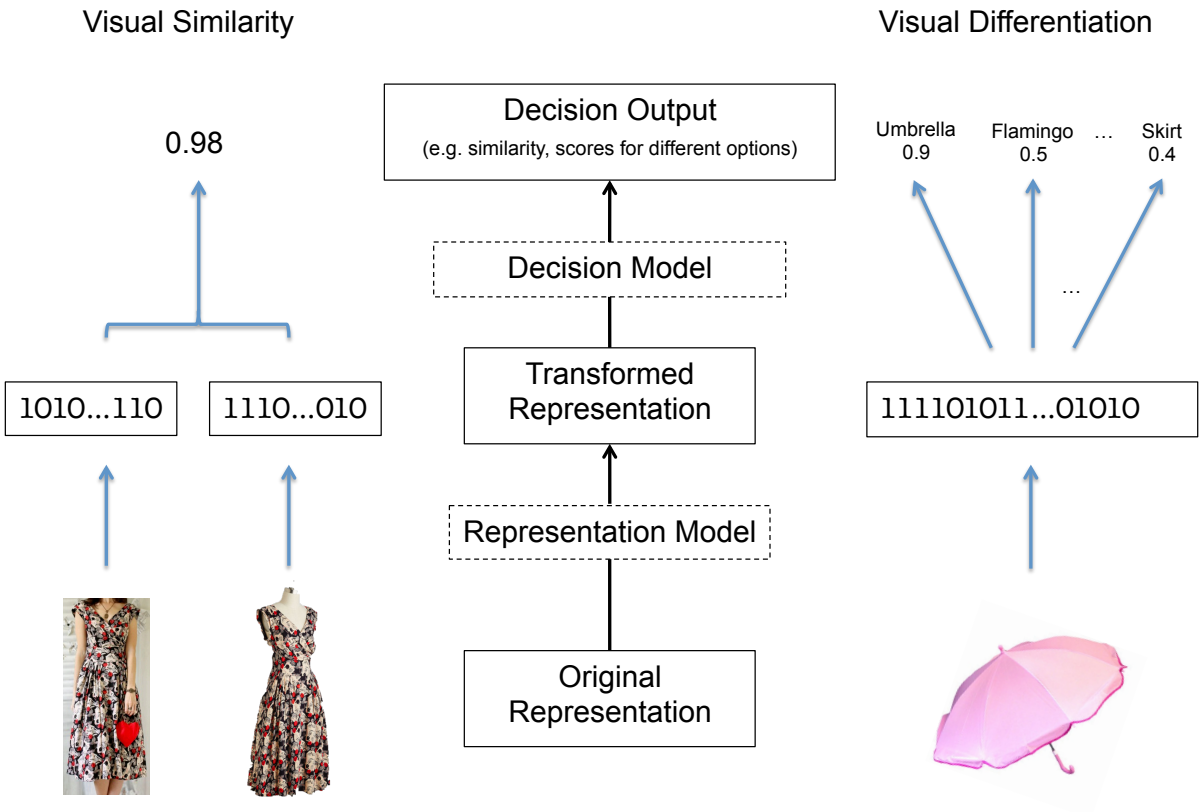

Figure 1.1: The representation model and the decision model for visual similarity and differentiation. On the left, two photos of dresses are first transformed from pixels into feature vectors, and then compared based on a decision model to obtain a similarity score. On the right, a photo of an object is also transformed from pixels into a feature vector before the decision model can compute a score for each possible labels. This thesis is concerned with identifying and tackling the challenges in designing the two types of models and learn them efficiently as we have more and more data.[1]

relevant to the task and by reducing irrelevant nuances in the original representation. The decision model encodes knowledge of how to reason about the transformed representation and produce results required by the task. Figure 1.1 shows a diagram of the two models connected together, as well as an illustration of how systems for visual similarity and differentiation can be viewed in this way.

For both visual similarity and visual differentiation, the state-of-the-art systems use machine learning to build the representation model and the decision model. The idea is to consider those models as functions, which either map a vector to a scalar or to another vector, and represent the functions in a certain parameter space. This way, building a model becomes estimating parameters such that not only

[1]Image source: `www.modcloth.com` (dresses), `www.saraglove.com` (umbrella)

does the model fit existing observations (training data) but also works for unseen, future, observations – also called *test data*.

An alternative to using machine learning throughout is to start with good intuition of the data and knowledge of the task to design the majority of each model by hand and leave a small number of hyper-parameters to be tuned quickly on a small validation dataset. For example, we can represent an image using its color histogram. This transforms the original pixel-value representation into a fixed-length vector representation, where each element in the vector is the proportion for each base color. The choice of color space (e.g. RGB, HSI, Lab, etc.) and the way we divide it can be predetermined based on domain knowledge, and assuming the whole space is covered (so all colors count), we only need to decide on one free parameter: the number of base colors. This representation may work quite well if the task is to distinguish between oranges and apples. For this task, it is harmless that the transformed representation loses most information on the spatial distribution of the colors, but for other tasks it could matter a lot.

Much research has been done on making the representation models robust to irrelevant variations in the image and work for general domains. Along the way, many effective approaches have been cleverly hand-crafted, which helped various visual similarity and differentiation tasks, for example, SIFT descriptor for matching local images (Lowe, 2003), filter banks representation for texture classification (Leung & Malik, 2001), spatial pyramid pooling for object retrieval and scene classification (Lazebnik et al., 2006), locally-constrained linear coding for robust object recognition (Yu et al., 2009; Wang et al., 2010), Histogram of Oriented Gradients (HOG) template for object detection (Dalal & Triggs, 2005), etc.

Hand-crafted representation models are fundamentally biased and thus is difficult to benefit from more data. More samples may provide a better prior distribution in the space of the transformed representation, but the effect on the overall performance is often uncertain, either because a better prior may only amplify the bias of the representation model, or because the learning algorithm for the decision model may assign a higher weight to the goodness-of-fit on the data than to the prior distribution.

An interesting set of studies carried by Zhu et al. (2015) reveal insights on how different representation models benefit differently from data. Specifically, Fig. 14 in their paper plots face detection performance against the amount of training data for a range of different representations. It is worth mentioning that those models vary more on representing the configuration of different parts than on how each part is represented. In fact, the part representation in all those models are templates of HOG features over the spatial region of the part, and the confidence score is simply the dot-product (or weighted sum of

Figure 1.2: Illustration of different representation models for face detection compared in Zhu et al. (2015). Here we simply use the face of Michelangelo's David to represent appearance templates, knowing that actual templates are in HOG space. **(A)** A single rigid template. **(B)** A single rigid template learned on frontal faces. **(C)** Mixture models for the whole face. **(D)** Deformable Parts Models (DPM). **(E)** DPM with tree configuration for parts and supervised part locations (on facial landmarks). Detection performance increases (A<B<C<D<E) as the representation model has more and more capacity to model variance of appearance in the data (C, D and E), and as better prior knowledge is used (B and E).

dot-products) between templates and HOG features from corresponding parts. In Figure 1.2 we illustrates some of the face models compared in their paper. Since there are frontal faces as well as profile ones in the dataset, a single rigid model trying to encode them all perform the worst, whereas one tuned for the just the frontal faces perform better — proper bias helps. A mixture of rigid templates have enough capacity to model different views separately and thus perform better than either one above. However the performance saturates quickly when the amount of training samples increases, because these models are too biased to handle the total variance of facial appearance. Deformable Parts Models (DPM) (Felzenszwalb et al., 2010) can model the object with several movable parts whose center positions form a star-shaped or tree-shaped (Yang & Ramanan, 2013; Zhu & Ramanan, 2012) spatial configuration. Since the parts are allowed to move around, more accurate appearance models can be learned. The configuration is also parameterized (e.g. as 2D Gaussian distributions of relative location for each part) and can either be learned. As expected, DPMs outperform rigid template mixtures. The intuition here is that a rigid template is good at handle local appearance variance between faces at part scale, but insufficient to model the variance in configurations of parts, which DPMs can model separately. Moreover, the authors show that models learned with supervising the part location using facial landmarks (e.g. tip of the nose, corners of eyes, etc) perform the best, which demonstrated again that good prior helps when applied properly.

Nonetheless, those representation models can be viewed as sophisticated ways of spatial pooling on HOG features. Although Zhu et al. (2015) conclude that developing better models appear the most promising to improve detection performance, they agree that "HOG is certainly limited.". One limitation

demonstrated directly using HOGgles (Vondrick et al., 2013), is that in the space of HOG template, albeit high dimensional, transformed random natural background patches could be very similar to actual objects.

To overcome similar limitations with hand-crafted representation models, much research has been done on learning representation models directly from pixels. A very influential model is the AlexNet (Krizhevsky et al., 2012), a multiple-hidden-layer (deep) neural network, in which each layer performs a differentiable non-linear transformation on the output of the previous layer. The layers can be one of the following types: convolution with Rectified Linear Unit (ReLU) [2] as activation, max-pooling, normalization and affine transformation followed by ReLU [3]. This model won the ILSVRC'12 Image Classification Challenge and pushed visual differentiation to new state-of-the-art.

1.2 Thesis Statement

Visual similarity concerns the identity relation between two visual entities. A similarity model maps two visual entities in certain representations to a scalar value, which indicates the, e.g. semantic, difference/similarity between the two. Visual differentiation concerns the identity relation between a visual entity and a concept. A differentiation model maps a visual entity to a vector whose elements indicate the strength of the identity relation between the entity and a set of predefined concepts.

To fully exploit the increasing amount of data available for visual similarity and differentiation, we need to address various challenges in seeking models that scale with the data and seeking algorithms that scale with both. One of them is how to design the space of similarity models such that both the representation model and the decision model benefit from the data together. Most previous work on visual similarity learning fix one and improve the other. We argue that jointly training the two scales better with the data, and developed such a model using deep convolutional neural networks for the task of local image patch matching. An associated challenge is how to make use of potentially quadratic number of pairs from a list of entities in an efficient way in terms of both space and time. We propose a sampling approach, which avoids explicitly generating training pairs (space efficient) and also avoids random reading of the database (time efficient).

[2] $\text{ReLU}(x) = \max(0, x)$

[3] Also called "fully-connected" layer, since every node in the output layer outputs a weighted sum of every node in the input layer followed by ReLU.

Another challenge happens when similarity models are applied in a retrieval setting, where the meaning of "relevant" is strict (e.g. finding the exact dress from a picture taken with a phone among a million stock photos) and a generic similarity model may not be sufficient. The training pairs might not be easy to collect too. The challenge is to design a training algorithm that can deal with limited data at an early stage and kickstart a positive feedback loop. We propose to use deep neural network models for both its ability to approximate non-linear mappings (a good property for decision models), and its potential to couple with a neural representation model. We also propose to train models only on a short list of candidates to alleviate data imbalance.

More data can mean more dimensions rather than more independent samples. For example, functional magnetic resonance imaging (fMRI) allows neural scientists to measure activity of different brain regions while the subject is performing a certain task, but it is expensive to run fMRI. Therefore most study involves only a small number of subjects, each going through a small number of "trials". On the other hand, a brain scan can have 10^5 voxels. Moreover, the measurement is prone to subject's head motion and is quite noisy. In such cases, choosing a good bias in the vast space of potential representation and decision models is crucial. We propose a system that combines domain knowledge, regularization, linear max-margin model and cross-validation for an fMRI study of visual working memory.

For visual differentiation, both the number of concepts and the number of samples for each concept can grow. When the total amount of data too large enough to be hold in memory on one machine, not only the model but the learning algorithm as well will need to scale with the data. The challenge is how to distribute the computation involved in the decision model to different machines. One aspect of the challenge is how to formulate the original problem as connected subproblems; another aspect is how to solve the subproblems efficiently. We study this problem in the context of multi-class support vector machine for large scale object recognition, and propose a principled approach using consensus optimization to address the first aspect, and a sequential dual solver to address the second.

Motivated by the above considerations, the main thesis that this dissertation aims to support is the following:

> *Models and algorithms for visual similarity and differentiation should scale up as more data becomes available. This requires that the models used for both representation decision making to have matching capacity with the increase in the number of samples and concepts,*

increase in data imbalance, increase in the dimension of representation and increase in noise. It can be achieved through (1) joint feature and metric modeling using neural networks, (2) systematic model selection, and (3) objective reformulation for data parallelization using consensus optimization.

1.3 Contributions

In this section, we briefly summarize the contributions presented in the following chapters.

1.3 MatchNet

As its name suggests, MatchNet is neural network for image matching. Following the theme of this thesis, we make use of large patch datasets to learn both the representation model and the decision model for the task of matching local image patches. Both models consist of layers of neural networks, which allows us to build an end-to-end training system from raw image patches to "match/non-match" labels using standard error back-propagation.

The original patch dataset comes with groups of matching matches as well as some fixed set of non-matching patches. Instead of using the predefined set, we develop a sampling approach to generate non-matching pairs (negatives) on the fly during training. This significantly increases the quantity and variety of negative pairs, and also allows us to balance the label distribution. The sampler keeps a buffer in the memory, and thus eliminating the need of random access to the entire database in search of negative pairs, making the learning compatible with mini-batch stochastic gradient descent in a streaming training setting.

MatchNet achieves state-of-the-art matching performance in terms of accuracy. We cite relevant previous results and run home brew baseline comparisons to show that coupling feature and metric learning indeed helps.

We also study trade-off between the accuracy and multiple other factors including feature dimension, metric network capability, and feature quantization, which provides guidance for tuning this type of model for different applications.

1.3 Exact Street-to-Shop

Exact Street-to-shop is a retrieval task of finding in large amount of stock photos exact items in a query consumer photo. It requires highly specific similarity models across the two domains: *street photos* and *shop photos*. One important characteristic of the shop photos is that we do not know where the item of interest is, and in many cases there are multiple items in single image. Therefore localization of the item becomes particularly important. Nonetheless, localization of clothing item is an open research problem by itself. One of our contribution is showing that applying an off-the-shelf high-recall localization method "over-generate" item candidates in every shop photos followed by a coarse retrieval and a fine reranking can be quite effective.

Another contribution is on the similarity model. Although in our experiment the course retrieval step uses a fixed representation models, the fine reranking step employ a learned decision model model specifically trained for the exact matching task. Both the representation model and the decision model are neural network based, so when more data comes in the future, we can easily switch on back-propagation in the representation model to make it learnable, just like in MatchNet. Therefore this approach has the potential to scale well with more data.

1.3 Multi-Voxel Pattern Analysis

This is a differentiation task in the space of fMRI images. This is a successful collaboration with neural scientists who use fMRI to study selective maintenance[4] in human visual working memory. One contribution is that machine learning with voxel activity patterns helps reveal selective maintenance in several regions where scientists predicted it exists but has not confirmed by using traditional methods of analysis. Another contribution is that we demonstrate that combining feature selection, regularization, linear max-margin model and cross-validation can make effective use of information in the fMRI dataset where the dimension of representation is high, the sample size small and the signal noisy.

1.3 DCMSVM

We propose a new algorithm named Distributed Consensus Multiclass SVM (DCMSVM) for efficient distributed parallel training of "single machine" or direct multi-class support vector machines, making

[4]The ability of choosing to hold certain information

this decision model much more scalable with data and thus much more widely usable. We benchmark on multi-class image classification methods including some using high dimensional descriptors from competitive systems. Results show significant improvements in wall-clock time and accuracy versus the very efficient implementation of Crammer & Singers algorithm in the Liblinear package.

CHAPTER 2

VISUAL SIMILARITY LEARNING USING DEEP NEURAL NETWORKS[5]

2.1 Introduction

Solutions to many computer vision problems rely on computing the similarity between pairs of visual entities. At the scale of local image patches around key points (e.g. corners), computing the similarity between patches leads to putative matching of key-points, which can be used for testing whether two images are geometrically related, and if they are, computing the transformation between them. Similarity can also be computed at object scale for retrieving from an objects database instances that match or are similar to a given query.

An interesting and challenging aspect of computing similarity that the notion of similarity is task-dependent. For example, we might consider a white dog more similar to a gray wolf than to a white cat, even though the color of the dog and the cat are more similar. Assuming collecting labeled data for the task is either necessary or relatively straightforward compared to crafting a customized similarity, this task-dependent nature calls for a data-driven approach posted as the following learning problem: given a dataset where pairs between elements are either implicitly or explicitly labeled as matching (very similar) or non-matching (not so similar), how to learn a pair-wise function that models a notion of similarity that conforms to the train data and generalizes in the given domain.

In this chapter, we begin by proposing a deep neural network model and a learning approach in the context of image patch matching. We demonstrate that coupling the learning of feature representation and distance metric is effective. Finally we present an application of such model in the context of exact-match clothing item retrieval.

---

[5]Majority of this chapter previously appeared in two proceedings (Han et al., 2015; Kiapour et al., 2015).

2.2 Similarity Learning for Patch-based Image matching

Patch-based image matching is used extensively in computer vision. Finding accurate correspondences between patches is instrumental in a broad variety of applications including wide-baseline stereo (*e.g.*, (Matas & Chum, 2004)), object instance recognition (*e.g.*, (Lowe, 1999), fine-grained classification (*e.g.*, (Yao et al., 2012)), multi-view reconstruction (*e.g.* (Seitz et al., 2006)), image stitching (*e.g.* (Brown & Lowe, 2007)), and structure from motion (*e.g.* (Molton et al., 2004)).

Since 1999, the advent of the influential SIFT descriptor (Lowe, 1999), research on patch-based matching has attempted to improve both accuracy and speed. Early efforts focused on identifying better affine region detectors (Mikolajczyk et al., 2005), engineering more robust local descriptors (Mikolajczyk & Schmid, 2005; Heinly et al., 2012), and exploring improvements in descriptor matching using alternate distance metrics (Jain et al., 2012; Jia & Darrell, 2011).

Early efforts at unsupervised data-driven learning of local descriptors (*e.g.*, (Ke & Sukthankar, 2004)) were typically outperformed by modern engineered descriptors, such as SURF (Bay et al., 2006), ORB (Rublee et al., 2011). However, the greater availability of labeled training data and increased computational resources has recently reversed this trend, leading to a new generation of learned descriptors (Brown et al., 2011; Trzcinski et al., 2012, 2013; Simonyan et al., 2014) and comparison metrics (Jia & Darrell, 2011). These approaches typically train a nonlinear model discriminatively using large datasets of patches with known ground truth matches and serve as motivation for our work.

Concurrently, approaches based on deep convolutional neural networks have recently made dramatic progress on a range of difficult computer vision problems, including image classification (Krizhevsky et al., 2012), object detection (Erhan et al., 2014), human pose estimation (Toshev & Szegedy, 2014), and action recognition in video (Karpathy et al., 2014; Simonyan & Zisserman, 2014). This line of research highlights the benefits of jointly learning a feature representation and a classifier (or distance metric), which to our knowledge has not been adequately explored in patch-based matching.

In the following sections, we propose a unified approach that jointly learns a deep network for patch representation as well as a network for robust feature comparison. In our system, dubbed MatchNet, each patch passes through a convolutional network to generate a fixed-dimensional representation reminiscent of SIFT. However, unlike in SIFT, where two descriptors are compared in feature space using the

Euclidean distance, in MatchNet, the representations are compared using a learned distance metric, implemented as a set of fully connected layers.

Our contributions include: 1) A new state-of-the-art system for patch-based matching using deep convolutional networks that significantly improves on the previous results. 2) Improved performance over the previous state of the art(Simonyan et al., 2014) using smaller descriptors (with fewer bits). 3) A careful set of experiments using standard datasets to study the relative contributions of different parts of the system, showing that MatchNet improves over both hand-crafted and learned descriptors plus comparison functions. 4) Finally we provide a public release of MatchNet trained using our own large collection of patches.

2.3 Background and Prior Work

Much previous work considers improving some components in the detector-descriptor-similarity pipeline for matching patches. Here we address the most related work that considers learning descriptors or similarities, organized by goal and the types of non-linearity used.

Feature learning methods such as (Brown et al., 2011), (Trzcinski et al., 2012) and (Simonyan et al., 2014) encode non-linearity into the procedure for mapping intensity patches to descriptors. Their goal is to learn descriptors whose similarity with respect to a chosen distance metric match the ground truth. For (Brown et al., 2011) and (Simonyan et al., 2014), the procedure includes multiple parameterized blocks of gradient computation, spatial pooling, feature normalization and dimension reduction. (Trzcinski et al., 2012) uses boosting with weak learners consisting of a family of functions parameterized by gradient orientations and spatial location. Each weak learner represents the result of feature normalization, orientation pooling and thresholding in its $+1/-1$ output. Weighting and combining multiple weak learners builds a highly non-linear mapping from gradients to robust descriptors. Different types of learning algorithms are proposed to find the optimal parameters: Powell minimization, boosting and convex optimization for (Brown et al., 2011), (Trzcinski et al., 2012) and (Simonyan et al., 2014), respectively. In (Brown et al., 2011) and (Simonyan et al., 2014) the similarity functions are simply the Euclidean distance. (Trzcinski et al., 2012) uses a Mahalanobis distance and jointly learns the descriptors and the metric. In comparison, our proposed feature extraction uses a deep convolutional network with

multiple convolutional and spatial pooling layers plus an optional bottle neck layer to obtain feature vectors, followed by a similarity measure also based on neural nets.

Metric learning methods such as (Jain et al., 2012) and (Jia & Darrell, 2011) learn a similarity function between descriptors that approximates a ground truth notion of which patches should be similar, and achieve results that improve on simple similarity functions, most often the Euclidean distance. Jain *et al*. (Jain et al., 2012) introduces non-linearity with a predefined kernel on patches. A Mahalanobis metric is learned on top of that similarity. Jia *et al*. (Jia & Darrell, 2011) uses a parametric distance based on a heavy-tailed Gamma-Compound-Laplace distribution, which approximates the empirical distribution of elements in the difference of matching SIFT descriptors. The parameters for this distance are estimated using the training data. In comparison, we use a two-layer fully connected neural network to learn the pairwise similarity, which has the potential to embrace more complex similarity functions beyond distance metrics such as Euclidean.

Semantic hashing or embedding learning methods learn non-linear mappings to generate low dimensional representations, whose similarity in some easy-to-compute distance metric (e.g., Hamming distance) correlates with the semantic similarity. This can be done using neural networks, e.g., (Chopra et al., 2005) and (Salakhutdinov & Hinton, 2007) with a two-tower structure and recently (Wang et al., 2014) that samples triplets for training. Spectral hashing (Weiss et al., 2008) or boosting (Shakhnarovich, 2006; Trzcinski et al., 2013) can also be used to learn the mapping. This approach can be applied to raw image input (Salakhutdinov & Hinton, 2007) as well as to local feature descriptors (Strecha et al., 2012). In comparison, although we do not map input to an intermediate embedding space explicitly, the representation provided by our feature extraction network naturally serves the purpose, and the dimensionality can be controlled depending on the accuracy vs. storage and computation tradeoff. We explore and analyze such effects in Section 2.6.1.

Our network structure is similar to the recent preprint (Zbontar & LeCun, 2014) for stereo matching, with a notable difference that we use pooling layers to learn compact representations from patches. Our approach, MatchNet, is designed for general wide-baseline viewpoint invariant matching, a significantly different problem than the more local matching problem in stereo. As one example, for wide-baseline matching, scale estimation from the key point descriptor may not be accurate. The pooling layers increase the robustness of the network robust to such variation. MatchNet has several other architectural differences, an additional convolutional layer, two fewer fully connected layers, and various differences

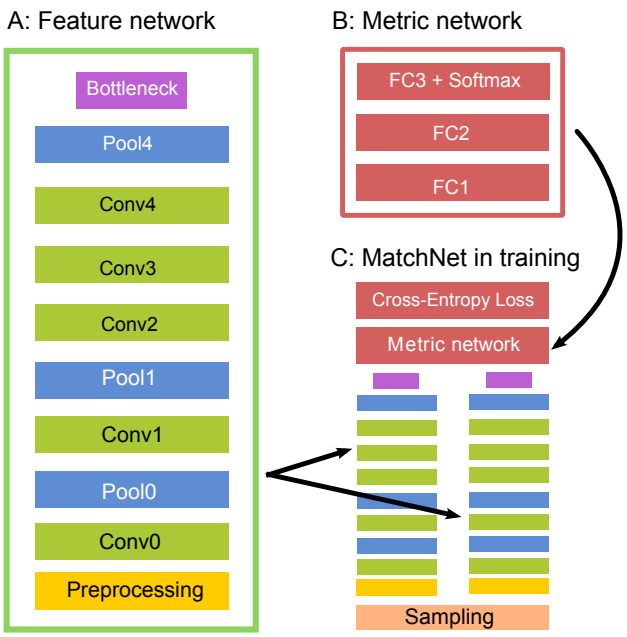

Figure 2.1: The MatchNet architecture. A: The feature network used for feature encoding, with an optional bottleneck layer to reduce feature dimension. B: The metric network used for feature comparison. C: In training, the feature net is applied as two "towers" on pairs of patches with shared parameters. Output from the two towers are concatenated as the metric network's input. The entire network is jointly trained on labeled patch-pairs generated from the sampler to minimize the cross-entropy loss. In prediction, the two sub-networks (A and B) are conveniently used in a two-stage pipeline (See Section 2.5.2).

in filter supports and layer complexity compared to (Zbontar & LeCun, 2014). We evaluate architectural variations in Section 2.6.1.

2.4 Deep Neural Network Architecture

MatchNet is a deep-network architecture (Fig. 2.1 C) for jointly learning a feature network that maps a patch to a feature representation (Fig. 2.1 A) and a metric network that maps pairs of features to a similarity (Fig. 2.1 B). It consists of several types of layers commonly used in deep-networks for computer vision. We show details of these layer in Table 2.1, and discuss some of the high-level architectural choices in this section.

The feature network: The feature network is influenced by AlexNet (Krizhevsky et al., 2012), which achieved good object recognition performance. We use many fewer parameters and do not use

Table 2.1: Layer specification of MatchNet. The output dimension is given by (height $\times$ width $\times$ depth). Patch size: patch size for convolution and pooling layers; Layer types: C: convolution, MP: max-pooling, FC: fully-connected. We always pad the convolution and pooling layers so the output height and width are those of the input divided by the stride. For FC layers, their size B and F are chosen from: $B \in \{64, 128, 256, 512\}$, $F \in \{128, 256, 512, 1024\}$. All convolution and FC layers use ReLU activation except for FC3, whose output is normalized with Softmax (Equation 2.2).

Name	Type	Output dimension	Patch size	Stride
Conv0	C	$64 \times 64 \times 24$	7×7	1
Pool0	MP	$32 \times 32 \times 24$	3×3	2
Conv1	C	$32 \times 32 \times 64$	5×5	1
Pool1	MP	$16 \times 16 \times 64$	3×3	2
Conv2	C	$16 \times 16 \times 96$	3×3	1
Conv3	C	$16 \times 16 \times 96$	3×3	1
Conv4	C	$16 \times 16 \times 64$	3×3	1
Pool4	MP	$8 \times 8 \times 64$	3×3	2
Bottleneck	FC	B	-	-
FC1	FC	F	-	-
FC2	FC	F	-	-
FC3	FC	2	-	-

Local Response Normalization or Dropout. We use Rectfied Linear Units (ReLU) as non-linearity for the convolution layers.

The metric network: We model the similarity between features using three fully-connected layers with ReLU non-linearity. FC3 also uses Softmax. Input to the network is the concatenation of a pair of features. We output two values in $[0, 1]$ from the two units of FC3, These are non-negative, sum up to one, and can be interpreted as the network's estimate of probability that the two patches match and do not match, respectively.

Two-tower structure with tied parameters: The patch-based matching task usually assumes that patches go through the same feature encoding before computing a similarity. Therefore we need just one feature network. During training, this can be realized by employing two feature networks (or "towers") that connect to a comparison network, with the constraint that the two towers share the same parameters. Updates for either tower will be applied to the shared coefficients.

This approach is related to the Siamese network (Bromley et al., 1994; Chopra et al., 2005), which also uses two towers, but with carefully designed loss functions instead of a learned metric network. A recent preprint on learning a network for stereo matching has also used the two-tower-plus-fully-connected comparison-network approach (Zbontar & LeCun, 2014). In contrast, MatchNet includes

max-pooling layers to deal with scale changes that are not present in stereo reconstruction problems, and it also has more convolutional layers compared to (Zbontar & LeCun, 2014).

In other settings, where similarity is defined over patches from two significantly different domains, the MatchNet framework can be generalized to have two towers that share fewer layers or towers with different structures.

The bottleneck layer: The bottleneck layer can be used to reduce the dimension of the feature representation and to control overfitting of the network. It is a fully-connected layer of size B, between the 4096 ($8 \times 8 \times 64$) nodes in the output of Pool4 and the final output of the feature network. We evaluate how B affects matching performance in Section 2.6.1 and plot results in Figure 2.5.

The preprocessing layer: Following a previous convention, for each pixel in the input grayscale patch we normalize its intensity value x (in $[0, 255]$) to $(x - 128)/160$. The resulting range of the input pixels is thus $[-0.8, 0.8]$.

2.5 Algorithms for Learning and Prediction

The feature and metric networks are trained jointly in a supervised setting using a two-tower structure illustrated in Figure 2.1-C. We minimize the cross-entropy error,

$$E = -\frac{1}{n}\sum_{i=1}^{n}\left[y_i \log(\hat{y_i}) + (1 - y_i)\log(1 - \hat{y_i})\right], \tag{2.1}$$

over a training set of n patch pairs using stochastic gradient descent (SGD) with a batch size of 32. Here y_i is the 0/1 label for input pair x_i. 1 indicates match. $\hat{y_i}$ and $1 - \hat{y_i}$ are the Softmax activations computed on the values of the two nodes in FC3, $v_0(x_i)$ and $v_1(x_i)$ as follows.

$$\hat{y_i} = \frac{e^{v_1(x_i)}}{e^{v_0(x_i)} + e^{v_1(x_i)}}. \tag{2.2}$$

$\hat{y_i}$ is used as the probability estimate for label 1 in Equation 2.1.

We experimented with different learning rates and momentum values and found using plain SGD with 0.01 for the learning rate yields better validation accuracy than using larger learning rates and/or with momentum, even though convergence in the latter settings is faster. Depending on the network architecture, it takes between 18 hours to 1 week to train the full network. Using a learning rate schedule can speed up the training significantly.

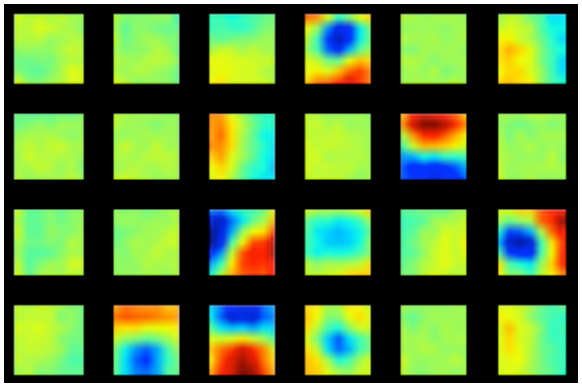

Figure 2.2: All 24 of the 7×7 filters learned in Conv0 from the liberty dataset. The pseudo-colors represent intensity.

Figure 2.2 visualizes Conv0 filters MatchNet learned on the Liberty dataset. Figure 2.3 visualizes the network's response to an example patch at different layers in the feature network.

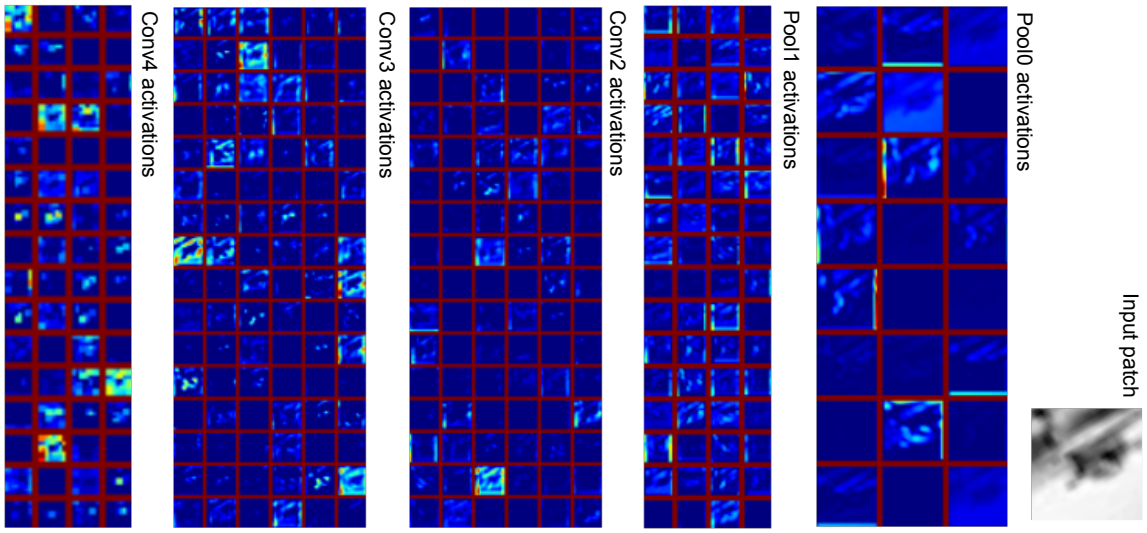

Figure 2.3: Visualization for the activations in response to an example input patch at different layers in the feature network. The input 64×64 patch is shown on the right. As sub-plot goes from right to left, we move up in the feature hierarchy. For each layer, we tile its K $H \times W$ activation maps to form a 2D image. H, W and K are the height, width and depth of the 3D activation array respectively. Red margins separates these tiles. Pseudo-colors in the tiles represent response intensity. Although border artifacts can occur, we keep our padding scheme, for it retains half of the information on the original border.

2.5 Reservoir Sampling for Label Balancing

Sampling is important in training, as the matching (+) and non-matching (-) pairs are highly unbalanced. We use a sampler to generate equal number of positives and negatives in each mini-

Algorithm 2.1 Generate a mini batch of size 2S with balanced pairs using a reservoir sampler.

for $b = 0 \dots S-1$ **do**
  Extract all patches $p_1 \dots p_k$ from the next group;
  Randomly choose p_i and p_j, $i \neq j$, $i, j \in \{1 \dots k\}$;
  $\text{Samples}(2b) \leftarrow (1, p_i, p_j)$;
  for $m = 0 \dots k$ **do**
    Consider adding p_m to the reservoir;[1]
  end for
  repeat at most 1000 times
    Randomly draw p_u and p_v from the reservoir;
  until p_u and p_v are from different group;[2]
  if negative sampling is successful **then**
    $\text{Samples}(2b+1) \leftarrow (0, p_u, p_v)$;
  else
    $\text{Samples}(2b+1) \leftarrow (1, p_i, p_j)$;
  end if
end for
return Samples;

batch so that the network will not be overly biased towards negative decisions. The sampler also enforces variety to prevent overfitting to a limited negative set.

Particularly, in our setting, the training set has already been grouped into matching patches; e.g. The UBC patch dataset has an average group size around 3. The learner streams through the training set by reading one group at a time. For positive sampling, we randomly pick two from the group; for negative sampling, we use a reservoir sampler (Vitter, 1985) with a buffer size of R patches. At any time T the buffer maintains R patches as if uniformly sampled from the patch stream up to T, allowing a variety of non-matching pairs to be generated efficiently. The buffer size controls the trade-off between memory and negative variety. In our experiments, $R = 128$ was too small and led to severe overfitting; $R = 16384$ has worked consistently. This procedure is detailed in Algorithm 2.1.

For instance, if the batch size is 32, in each training iteration we feed SGD 16 positives and 16 negatives. The positives are obtained by reading the next 16 groups from the database and randomly picking one pair in each group. Since we go through the whole dataset many times, even though we only pick one positive pair from each group in each pass, the network still gets good positive coverage,

[1]Following (Vitter, 1985), if the sampler's reservoir is not full, the candidate is always added; otherwise for the T-th candidate, with probability R/T it is added and replaces a random element in the reservoir and with probability 1-R/T it gets rejected. R is the reservoir size.

[2]We store meta data along with the patches in the buffer so it is efficient to check whether two patches match or not.

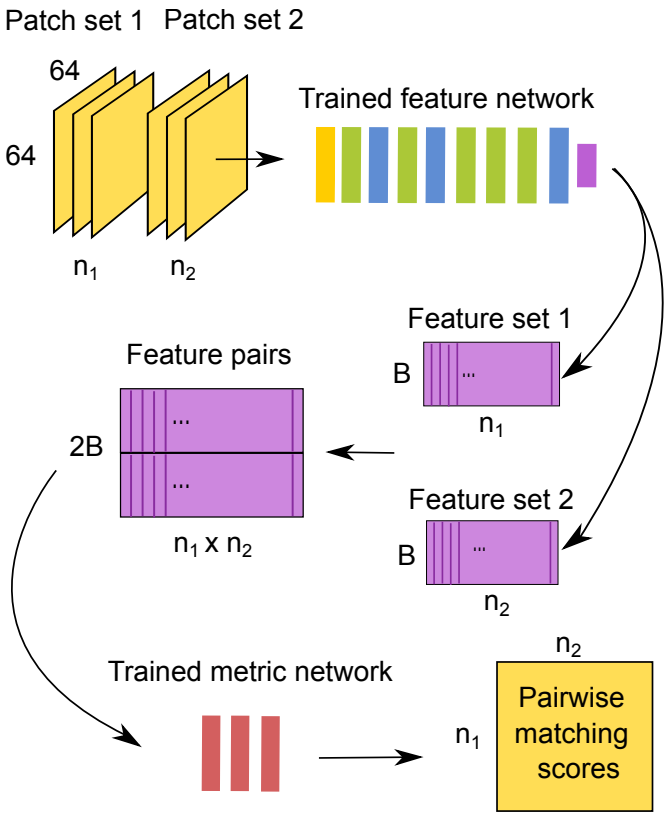

Figure 2.4: A Two-Stage prediction pipeline for computing pairwise matching score. MatchNet is disassembled during prediction. The feature network and the metric network run in a pipeline.

especially when the average group size is small. The 16 negatives are obtained by sampling two pairs from different groups from the reservoir buffer that stores previous loaded patches. At the first few iterations, the buffer would be empty or contain only matching patches. In that case we simply fill the slot in the batch with the most recent positive pair.

2.5 A Two-Stage Prediction Pipeline

A common scenario for patch-based matching is that there are sets of patches each extracted from two images. The goal is to compute a $N_1 \times N_2$ matrix of pairwise matching scores, where N_1 and N_2 are the number of patches in from image. Pushing each pair through the full network is not efficient because the feature tower would run on the same patch multiple times. Instead, we can use the feature tower and the metric network separately and in two stages (Figure 2.4). First we generate feature encodings for all patches. Then we pair the features and push them through the metric network to get the scores. In our

experiment, on one NVIDIA K40 GPU, after tuning batch size, the feature net without bottleneck runs at 3.56K patch/sec; the metric net (B=128, F=512) runs at 416.6K pair/sec. The computation can be further pipelined and distributed for large-scale applications.

2.6 Evaluation on Image Patch Matching

2.6 Dataset and Evaluation Protocol

The UBC patch dataset (UBC) [6] was collected by Winder et al. (Winder et al., 2009) for learning descriptors. The patches were extracted around real interest points from several internet photo collections published in (Snavely et al., 2008). The dataset includes three subsets with a total of more than 1.5 million patches. It is suitable for discriminative descriptor or metric learning, and has been used as a standard benchmark dataset by many (Brown et al., 2011; Jia & Darrell, 2011; Trzcinski et al., 2012, 2013; Simonyan et al., 2014). The dataset comes with patches extracted using either Difference of Gaussian (DoG) interest point detector or multi-scale Harris corner detector. We use the DoG set.

There are three subsets in UBC: Liberty, Notredame and Yosemite. Each comes with pre-generated labeled pairs of 100k, 200k and 500k, all with 50% matches. Each also provides all unique patches and their corresponding 3D point ID. The number of unique patches in each dataset is 450k for Liberty, 468k for Notredame and 634k for Yosemite.

Following the standard protocol established in (Brown et al., 2011), people train the descriptor on one subset and test on the other two subsets. Although people may use any of the pre-generated pair sets or the grouped patches in the training subset for training and validation, the testing is done on the 100k labeled pairs in the test subset. The commonly used evaluation metric is the false positive rate at 95% recall (Error@95%), the lower the better.

2.6 Baseline Experiments with SIFT features

We use VLFeat (Vedaldi & Fulkerson, 2010)'s `vlsift()` with default parameters and custom frame input to extract SIFT descriptor on patches. The frame center is the center of the patch at (32.5, 32.5). The scale is set to be 16/3, where 3 is the default magnifying coefficient, so that the bin size for the descriptor will be 16. With 4 bins along each side, the descriptor footprint covers the entire

[6] `http://www.cs.ubc.ca/~mbrown/patchdata/patchdata.html`

patch. In our preliminary experiments we found that normalized SIFT (nSIFT), which is raw SIFT scaled so its L2-norm is 1, gives slightly better performance than SIFT, so nSIFT is used for all our baseline experiments.

For a pair of nSIFT, we compute similarity using L2, linear SVM on 128d element-wise squared difference features (Squared diff.) and a two-layer fully-connected neural networks on 256d nSIFT concatenation (Concat.). For nSIFT Square diff.+ linearSVM, we use Liblinear (Fan et al., 2008) to train the SVM and search the regularization parameter C among $\{10^{-4}, 10^{-3} \ldots, 10^{4}\}$ using 10% of the training set for validation. For nSIFT Concat.+ NNet, the network has the same structure (with F=512) as the metric network in MatchNet (Figure 2.1-B) and is trained using plain SGD with learning_rate=0.01, batch_size=128 and iteration=150k.

2.6 Variations of MatchNet

We evaluate MatchNet under different architectural variation. We train MatchNet using techniques described in Section 2.5 and evaluate the performance under different (F, B) combinations, where F and B are the dimension of fully-connected layers (F1 and F2) and the bottleneck layer respectively. $F \in \{128, 256, 512, 1024\}$. $B \in \{64, 128, 256, 512\}$. We also evaluate using the feature network without the bottleneck layer.

We also evaluate MatchNet with feature quantization. The output features of the bottleneck layer in the feature tower (Figure 2.1-A) are represented as floating point numbers. They are the outputs of ReLu units, thus the values are always non-negative. We quantize these feature values in a simplistic way. For a trained network, we compute the maximum value M for the features across all dimensions on a set of random patches in the training set. Then each element v in the feature is quantized as $q(v) = \min(2^n - 1, \lfloor (2^n - 1)v/M \rfloor)$, where n is the number of bits we quantize the feature to. When the feature is fed to the metric network, v is restored using $q(v)M/(2^n - 1)$. We evaluate the performance using different quantization levels.

The quantized features give us a very compact representation. The ReLU output of the bottleneck layer is not dense. For example, for the (B=64, F=1024) network, the average density over all the UBC data is 67.9%. Using a naive representation: 1 bit to encode whether the value is zero or not, quantizing the features to 6 bits yields a representation of $64 + 6 \times 64 \times 0.679 = 324.7$ bits on average. As discussed below, this compact representation yields similar performance achieved by state-of-the-art methods with

Table 2.2: Patch matching results on UBC dataset. Numbers are Error@95% in percentage. See Section 2.6.1 for descriptions of different settings. F and B are dimensions for fully-connected and bottleneck layers in Table 2.1. **Bold** numbers are the best results across all conditions. Underlined numbers are better than the previous state-of-the-art results with similar feature dimension.

Training		Notr.	Yos.	Lib.	Yos.	Lib.	Notr.	
Test	Feature Dim.	Lib.		Notr.		Yos.		Mean
nSIFT + L2 (no training)	128d	29.84		22.53		27.29		26.55
nSIFT squared diff. + linearSVM	128d	26.54	27.07	19.65	19.87	25.12	24.71	23.82
nSIFT concat. + NNet (F=512)	256d	20.44	22.23	14.35	14.84	21.41	20.65	18.99
Trzcinski et al (2012)	64d	18.05	21.03	14.15	13.73	19.63	15.86	17.08
Brown et al (2011) w/ PCA	29d	16.85	18.27	–	11.98	–	13.55	15.16
Simonyan et al (2014) PR	<640d	16.56	17.32	9.88	9.49	11.89	11.11	12.71
Simonyan et al (2014) discrim. proj.	<80d	12.42	14.58	7.22	6.17	11.18	10.08	10.28
Simonyan et al (2014) discrim. proj.	<64d	12.88	14.82	7.52	7.11	11.63	10.54	10.75
MatchNet (F=1024, B=64)	64d	9.82	14.27	5.02	9.15	14.15	13.20	10.94
MatchNet (F=512, B=128)	128d	9.48	15.40	5.18	8.27	14.40	12.17	10.82
MatchNet (F=512, B=512)	512d	8.84	13.02	4.75	7.70	13.58	11.00	9.82
MatchNet (F=512, w/o bottleneck)	4096d	**6.90**	**10.77**	**3.87**	**5.67**	**10.88**	**8.39**	**7.75**

1024 bits. Of course, employing a more sophisticated encoding mechanism should further improve compactness.

2.6 Results and Discussion

We follow the evaluation protocol and evaluate MatchNet along with several SIFT baselines and other learned float descriptors. Results for SIFT baselines and MatchNet with floating point features are listed in Table 2.2. Our best model 4096-512x512 (feature dim.-FxF) achieves best performance over all evaluation pairs. It achieves 7.75% average error rate vs. (Simonyan et al., 2014)'s <80f at 10.38%. With a bottleneck of 64d, our 64-1024×1024 model achieves 10.94% average error rate vs. (Simonyan et al., 2014)'s 10.75% using features with about the same dimension.

One advantage of our approach is that we can easily vary feature dimension and matching complexity and jointly optimize them. The trade off between storage/computation vs. accuracy is plotted in Figure 2.5. Not suprisingly, increasing F and B leads to lower error rate, but the absolute gain is diminishing exponentially.

The results for MatchNet with quantized features are presented in Table 2.3. We use the (B=64, F=1024) model and quantize the output of the feature tower into different number of bits. 6-bit quantization achieves an average error of 11.01% and with 3 train/test pairs even better than (Simonyan et al., 2014)'s <80f. The performance is almost exactly the same as (Simonyan et al., 2014)'s 1024 bit

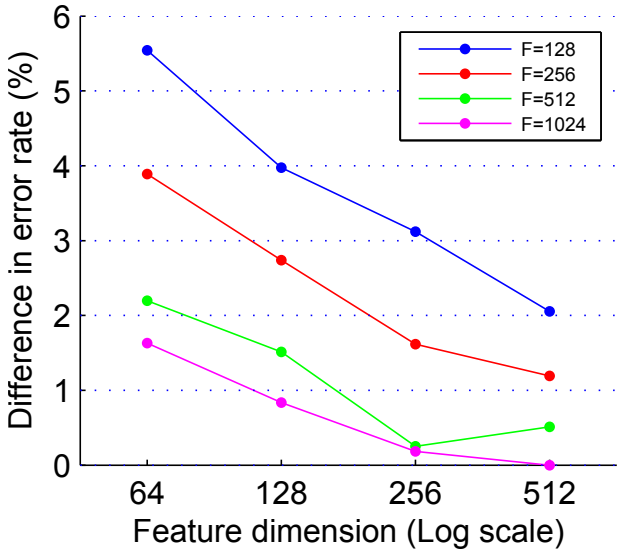

Figure 2.5: Accuracy vs. dimensionality tradeoff for different fully-connected layer sizes. we plot the average difference in Error@95% between other (F, B) combinations and $(F = 1024, B = 512)$ across all 6 train-test pairs in UBC. Features are unquantized.

representation while using only 325 bits instead of 1024. For 6-bit quantization, the feature has 324.7 bits (40.6 bytes) per patch on average. Our results also show that going from a 32-bit floating point representation to a 6-bit one yields little degradation in performance. We attribute this to the robustness of the matching network.

Our baseline experiments with SIFT features confirms that a better metric can significantly improve performance. For instance, in Yosemite-Liberty, nSIFT concat.+NNet performs better than nSIFT+L2 by 7.61% in absolute error rate, and MatchNet with the same feature dimension (128) and fully-connected layer dimension (512) achieves a further improvement of 6.7% in absolute error rate. The benefit of coupling the descriptor and the metric has been explored in different forms in the past. For instance, (Simonyan et al., 2014) learns weights for each pooling regions, so does (Trzcinski et al., 2012) for each weak gradient map learner to form a Mahalanobis type of similarity. Our proposed unification through neural networks is a simple and powerful alternative.

Our best model is trained without a bottleneck and it learns a high-dimensional patch representation coupled with a discriminatively trained metric. It improves on the previous state-of-the-art performance across all train-test pairs by up to 3.4% in absolute error rate. Fig. 3 in Simonyan et al. (Simonyan et al., 2014) showed increasing feature dimension reduces the error rate. However this effect may not

Table 2.3: Accuracy vs. quantization tradeoff for the 64-1024×1024 network. Names in the first row and second row are names of the training set and test set respectively. In the first column, the first value is bits per dimension. The second value is average bits per feature vector. It is computed using $64 + 64 \times 0.679 \times b$, where b is the number of bits per dimension, and the average density (non-zeros) of the feature vector is 67.9%. Numbers in the middle are Error@95%. The first row is for the un-quantized features.

	Notredame	Yosemite	Liberty	Yosemite	Liberty	Notredame
# of bits	Liberty		Notredame		Yosemite	
32 (1456)	9.82	14.27	5.02	9.15	14.15	13.20
8 (411.7)	9.84	14.33	5.06	9.21	14.17	13.21
7 (368.6)	9.82	14.20	5.04	9.23	14.21	13.19
6 (324.7)	9.81	14.22	5.15	9.30	14.27	13.29
5 (281.3)	10.19	14.58	5.33	9.59	14.66	13.39
4 (237.8)	11.37	15.27	6.27	10.93	15.59	14.07

be enough to explain our $\sim$3% absolute gain for Liberty-Notredame. On one hand, the top curve in (Simonyan et al., 2014)'s Fig. 3 shows a diminishing gain. Without discriminative projection, at around 1500d, the error rate is still above 9%, more than twice as much as MatchNet's error rate (3.87%) with 4096d patch representation. On the other, with a 512d bottleneck and quantization, MatchNet still outperforms (Simonyan et al., 2014)'s PR (<640d) results in 4 out of 6 train-test pairs with up to 7% improvement in absolute error rate.

Supported by the tradeoff results in Figure 2.5 and Table 2.3, we provide the following guidelines to enable users to choose a configuration based on their storage/computation constraints: The 4096-512x512 model should be used if the feature dimension is not a concern, or if accuracy takes priority. This model outperforms others by a large margin on UBC. If extra compression is needed, the 64-1024x1024 one with quantization should be used.

More sophisticated quantizations could enable a better accuracy vs. dimensionality tradeoff. Potential improvements include: (a) Using a per-dimension max value; (b) better zero encoding (currently we use 1 bit per dimension to indicate whether the dimension is 0); (c) better compression algorithm; and (d) Reduce the range from $[0, \text{MaxValue}]$ to $[0, \text{95th percentile}]$.

2.7 Cross-domain Similarity for Exact-Match Clothing Item Retrieval

2.7 Exact Street to Shop

In this section, we discuss how to apply similarity learning to a challenging clothing item retrieval problem, which we call "Exact Street to Shop". Given a real-world photo of a clothing item, e.g. taken on the street, the goal of this task is to find that clothing item in an online shop. This is extremely challenging due to differences between depictions of clothing in real-world settings versus the clean simplicity of online shopping images. For example, clothing will be worn on a person in street photos, whereas in online shops, clothing items may also be portrayed in isolation or on mannequins. Shop images are professionally photographed, with cleaner backgrounds, better lighting, and more distinctive poses than may be found in real-world, consumer-captured photos of garments. To deal with these challenges, we introduce a deep learning based methodology to learn a similarity measure between street and shop photos.

The street-to-shop problem has been recently explored (Liu et al., 2012). Previously, the goal was to find similar clothing items in online shops, where performance is measured according to how well retrieved images match a fixed set of attributes, e.g. color, length, material, that have been hand-labeled on the query clothing items. However, finding a similar garment item may not always correspond to what a shopper desires. Often when a shopper wants to find an item online, they want to find *exactly* that item to purchase. Therefore, we define a new task, *Exact Street to Shop*, where our goal is for a query street garment item, to find exactly the same garment in online shopping images.

2.7 Item Localization

As we mentioned at the beginning of this chapter, similarity can be computed at different scales. For the *Exact Street to Shop task*, the relevant scales are image-level and item-level. The former computes representation using the whole image; the latter using only the bounding box of the item of interest. We can assume the street photos have clothing items annotation in the form of bounding boxes. This is a reasonable assumption because the user is assumed motivated to search for a particular item in the photo (e.g. a dress) that is likely to contain multiple items, and thus may spend a small amount of effort annotating the item of interest with a bounding box to form a query for the retrieval system. In contrast, we should not assume the shop photos have bounding box annotations, because the items are always

listed under the correct category and shoppers can always locate the item of interest, which means the shops have no motivation to annotate the exact location of items. Therefore, the distinction of computing similarity at image-level and item-level affects how we process the shop images. We show in Section 2.8 that using item-level similarity significantly improves the retrieval success rate. Therefore localization matters.

Since clothing parsing is still an ongoing research (Yamaguchi et al., 2013), we cannot simply rely on a clothing parser to get bounding boxes for items in shop images. Instead, we use the selective search method (van de Sande et al., 2011), a high-recall, automatic, generic object proposal method to generate multiple candidate bounding boxes. This method is expected to output boxes that contain the clothing item or contain part of them, while rejecting a large number of boxes that contains only the background. The idea is that we improve the localization of the item of interest at the cost of more comparison.

More specifically, we use the selective search algorithm and filter out proposals with a width smaller than $\frac{1}{5}$ of the image width since these usually correspond to false positive proposals. From this set, the 100 most confident object proposals are kept. This remaining set of object proposals has an average recall of 97.76%, evaluated on an annotated subset of 13,004 shop item photos.

2.7 Similarity Learning

For similarity, our hypothesis is that the cosine similarity on existing convolutional neural network (CNN) features may be too general to capture the underlying differences between the street domain and the shop domains. Therefore, we explore methods to learn a data-driven similarity.

Inspired by recent work on deep similarity learning for matching image patches between images of the same scene (Han et al., 2015; Zagoruyko & Komodakis, 2015; Zbontar & LeCun, 2015), we model the similarity between a query feature descriptor and a shop feature descriptor with a three-layer fully-connected network and learn the similarity parameters for this architecture. Here, labeled data for training consists of positive samples, selected from exact street-to-shop pairs, and negative samples, selected from non-matching street-to-shop items.

Specifically, the first two fully-connected layers of our similarity network have 512 outputs and use Rectified Linear Unit (ReLU) as their non-linear activation function. The third layer of our network has two output nodes and uses the soft-max function as its activation function. The two outputs from this final layer can be interpreted as estimates of the probability that a street and shop item "match",

Category	Average boxes	Train+	Train−	Val+	Val−
Bags	326	5.6	65.5	1.9	20.7
Belts	384	1.2	28.0	0.4	12.1
Dresses	426	99.7	1456.2	46.1	670.1
Eyewear	276	1.7	69.4	0.1	3.9
Footwear	255	4.3	92.3	5.1	113.2
Hats	356	3.0	82.0	0.4	12.4
Leggings	344	6.5	145.1	2.0	52.1
Outerwear	274	6.5	118.7	2.8	52.4
Pants	443	5.4	75.2	2.3	27.2
Skirts	559	59.2	901.2	17.3	276.1
Tops	349	15.4	186.2	6.9	92.1

Table 2.4: Size statistics of the training and validation sets for similarity learning. Average boxes shows the number of selective search boxes used, averaged across query images with at least one successful retrieval within the short list. Numbers in the last four columns are in units of 1,000.

or "do not match", which is consistent with the use of cross-entropy loss during training. Once we have trained our network, during the test phase, we use the "match" output prediction as our similarity score. Previous work has shown that this type of metric network has the capacity for approximating the underlying non-linear similarity between features. For example, Han et al. (Han et al., 2015) showed that the learned similarity for SIFT features, modeled by such a network, is more effective than L2-distance or cosine-similarity for matching patches across images of a scene.

We formulate the similarity learning task as a binary classification problem, in which positive/negative examples are pairs of CNN features from a query bounding box and a shop image selective-search based item proposal, for the same item/different items. We minimize the cross-entropy error

$$E = -\frac{1}{n}\sum_{i=1}^{n}\left[y_i \log(\hat{y_i}) + (1-y_i)\log(1-\hat{y_i})\right] \tag{2.3}$$

over a training set of n bounding box pairs using mini-batch stochastic gradient descent. Here, $y_i = 1$ for positive examples; $y_i = 0$ for negative examples; and $\hat{y_i}$ and $1 - \hat{y_i}$ are the two outputs of the metric network. One complication is that we do not have hand-labeled bounding boxes for shop images. We could use all object proposals for a shop image in a matching street-to-shop pair as positive training data, but because many boxes returned by the selective-search procedure will have low intersection-over-union (IoU) with the shop item of interest, it would introduce too many noisy training examples. Another source of noisy examples for similarity training is that, due to large pose differences in images for an

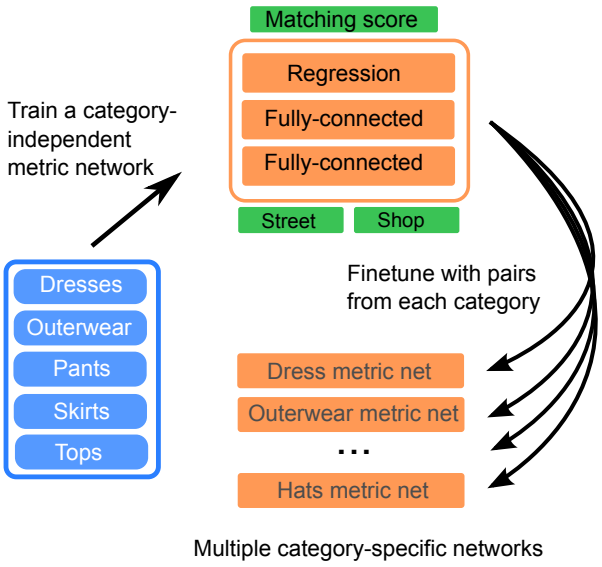

Figure 2.6: Illustration of the training, followed by fine-tuning procedure for training category-specific similarity for each category. To deal with limited data, we first train a generic similarity using five large categories and then fine-tune it for each category individually. See Section 2.7.3 for more description.

item, some images on the shop side will bear little similarity in appearance to a particular query item view. Labeling such visually distinct pairs as positives would likely confuse the classifier during training.

We handle these challenges by training our metric network on a short list of top retrieved shop bounding boxes using the object proposal retrieval approach described above. At test time, we perform the same primitive retrieval to obtain a short list of candidate retrievals and then re-rank the list using our learned similarity. This has an added benefit of improving the efficiency of our retrieval approach since the original cosine similarity measure is faster to compute than the learned similarity. Moreover, for real application, the entire primitive retrieval could be implemented efficiently using a proper binary hashing mechanism such as muilt-index hashing (Norouzi et al., 2014).

More specifically, to construct training and validation sets for similarity learning, for each training query item, q, we retrieve the top 1000 selective search boxes from shop images using cosine similarity. For each bounding box b from a shop image in this set, (q, b) is a positive sample if the shop image is a street-to-shop pair with q. Otherwise, (q, b) is used as a negative sample[7].

We randomly split the queries into `train` and `val` sets in a 2:1 ratio. such that `val` contains half the number of items as the test set. Statistics for `train` and `val` are shown in Table 2.4 for our retrieval experiments.

[7]Note, here we use only shop bounding boxes for training belonging to the top-K ($K = 75$) items in the retrieval set

Intuitively, we might want to train a different similarity measure for each garment category, for example, objects such as hats might undergo different deformations and transformations than objects like dresses. However, we are limited in the number of positive training examples for each category and by the large negative-to-positive ratio. Therefore, we employ negative sampling to balance the positive and negative examples in each mini-batch. We train a general street-to-shop similarity measure, followed by fine-tuning for each garment category to achieve *category-specific* similarity (See Figure 2.6).

In the first stage of training, we select five large categories from our garment categories: Dresses, Outerwear, Pants, Skirts, and Tops and combine their training examples. Using these examples, we train an initial *category-independent* metric network. We set the learning rate to 0.001, momentum to 0.9, and train for 24,000 iterations, then lower the learning rate to 0.0001 and train for another 18,000 iterations. In the second stage of learning, we fine-tune the learned metric network on each category independently (with learning rate 0.0001), to produce category-dependent similarity measures. In both stages of learning, the corresponding validation sets are used for monitoring purposes to determine when to stop training.

2.8 Evaluation on Exact-Match Clothing Item Retrieval

2.8 Dataset and Evaluation Protocol

The *Exact Street2Shop Dataset* [8] is collected by M. Hadi Kiapour (Kiapour et al., 2015). It contains street photos, shop photos and street-to-shop correspondences in 11 garment categories: *Bags, Belts, Dresses, Eyewear, Footwear, Hats, Leggings, Outerwear, Pants, Skirts and Tops*. Street photos are amateur fashion photographs of people wearing clothing items. Shop photos are professional photographs listed in online shopping websites representing a specific clothing item. On the shop side, a clothing item is usually represented in multiple shots including but not limited to a frontal shot, a back shot and a close-up shot. Examples of the street photos and shop photos can be seen in Figure 2.7.

[8] Available at `http://www.tamaraberg.com/street2shop`

Figure 2.7: Example of street photos (the six ones on the left) and shop photos (the six ones on the right).

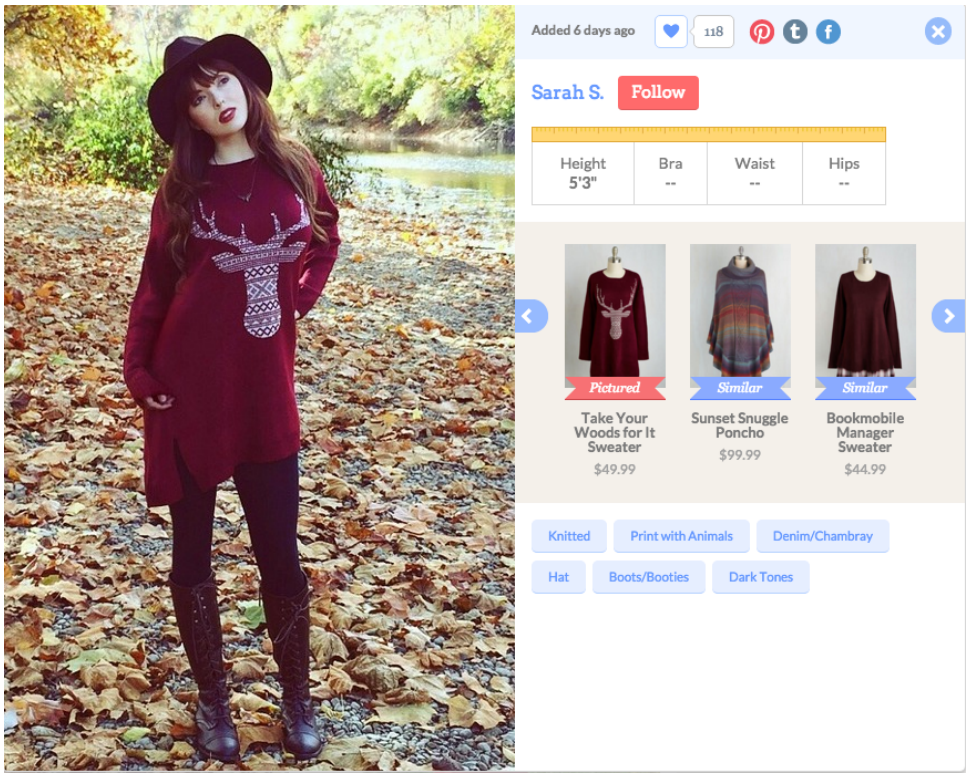

Figure 2.8: A screenshot of an example post in ModCloth's Style Gallery. On the left is the street photo. On the right are similar items (with blue "Similar"labels) and exact matches (with red "Pictured" labels). Clicking on any shop photos opens the shopping page for that item.

Street-to-shop correspondences are collected from the style galleries of ModCloth website [9]. These style galleries contain user-uploaded posts with street photos as well links to the shopping page on ModCloth where items in the photos can be purchased. Figure 2.8 shows a screenshot of an example post in the gallery, where the street-to-shop correspondence occurs.

The dataset also includes bounding boxes over items with corresponding shop photos and shop images from 24 other online stores, such as TheRealReal, Amazon, NordStrom, etc.

For experiments that involves learning, we split the exact matching pairs into two disjoint sets such that there is no overlap of items in street and shop photos between train and test. In particular, for each category, the street-to-shop pairs are distributed into train and test splits with a ratio of approximately 4:1. For our retrieval experiments, a query consists of two parts: 1) a street photo with an annotated bounding box indicating the target item, and 2) the category label of the target item. We view these as simple annotations that a motivated user could easily provide, but this could be generalized to use automatic detection methods. Since the category is assumed to be known, retrieval experiments are performed within-category. Street images may contain multiple clothing items for retrieval. We consider each instance as a separate query.

Performance is measured in terms of *top-k accuracy*, the percentage of queries with at least one matching item retrieved within the first k results.

2.8 Main Results

Following the above protocol, we compare across all categories four different similarity models: Cosine similarity on whole image (Whole Image), Cosine similarity on selective search boxes (Selective Search), Learned category-independent similarity on selective search boxes (Similarity), Learned category-specific similarity (F.T. Similarity). We use pre-trained AlexNet (Krizhevsky et al., 2012) FC6 activations as feature representation across all four models.

Table 2.5 (right) presents the exact matching performance of our baselines and learned similarity approaches (before and after fine-tuning) for k=20. Whole image retrieval performs the worst on all categories. The object proposal method improves over whole image retrieval on all categories, especially on categories like eyewear, hats, and skirts, where localization in the shop images is quite useful. Skirts,

[9] `www.modcloth.com/style-gallery`

Category	Queries	Query Items	Shop Images	Shop Items	Whole Im.	Sel. Search	Sim.	F.T. Sim.
Bags	174	87	16,308	10,963	23.6	32.2	31.6	**37.4**
Belts	89	16	1,252	965	6.7	6.7	11.2	**13.5**
Dresses	3,292	1,112	169,733	67,606	22.2	25.5	36.7	**37.1**
Eyewear	138	15	1,595	1,284	10.1	**42.0**	27.5	35.5
Footwear	2,178	516	75,836	47,127	5.9	6.9	7.7	**9.6**
Hats	86	31	2,551	1,785	11.6	36.0	24.4	**38.4**
Leggings	517	94	8,219	4,160	14.5	17.2	15.9	**22.1**
Outerwear	666	168	34,695	17,878	9.3	13.8	18.9	**21.0**
Pants	130	42	7,640	5,669	14.6	21.5	28.5	**29.2**
Skirts	604	142	18,281	8,412	11.6	45.9	**54.6**	**54.6**
Tops	763	364	68,418	38,946	14.4	27.4	36.6	**38.1**

Table 2.5: Test dataset statistics and top-20 item retrieval accuracy for the Exact-Street-to-Shop task. The last four columns report performance using whole-image features, selective search bounding boxes, and re-ranking with learned generic similarity or fine-tuned similarity.

for example, are often depicted on models or mannequins, making localization necessary for accurate item matching. We also trained category-specific detectors (Girshick et al., 2014) to remove the noisy object proposals from shop images. Keeping the top 20 confident detections per image, we observe a small drop of 2.16% in top-20 item accuracy, while we are able to make the retrieval runtime up to almost an order of magnitude more efficient (e.g. 7.6x faster for a single skirt query on one core).

Our final learned similarity after category-specific fine-tuning achieves the best performance on almost all categories. The one exception is eyewear, for which the object proposal method achieves the best top-20 accuracy. The initial learned similarity measure before fine-tuning achieves improved performance on categories that it was trained on, but less improvement on the other categories.

Example retrieval results are shown in Figure 2.9. The top three rows show success, where the exact matches are among the top results. Failure examples are shown in the bottom rows. Failures can happen for several reasons, such as visual distraction from textured backgrounds (e.g. 4^{th} row). A more accurate but perhaps more costly localization of the query item, might be helpful in these cases. Sometimes, items are visually too generic to find the exact item in shop images (e.g. blue jeans in the 5^{th} row). Finally, current deep representations may fail to capture some subtle visual differences between items (last row). We also observe errors due to challenging street item viewpoints.

Figure 2.9: Example retrievals using category-specific similarity. Top and bottom three rows show example successful and failure cases respectively.

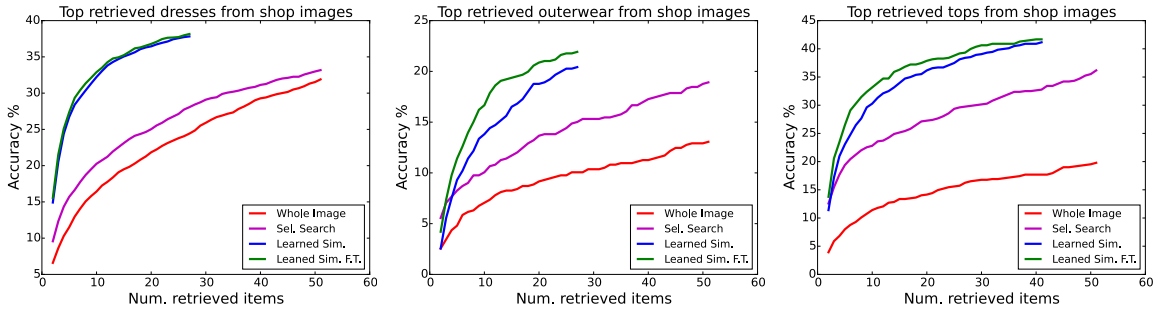

Figure 2.10: Top-k item retrieval accuracy for different numbers of retrieved items.

In Figure 2.10 we plot the top-k retrieval accuracy over values of k for three example categories (dresses, outerwear and tops). For similarity learning, we vary k from 1 to the number of available items in the retrieved short list. For the baseline methods, we plot accuracy for k=1 to 50. We observe that

the performance of our similarity network grows significantly faster than the baseline methods. This is particularly useful for real-world search applications, where users rarely look beyond the first few highly ranked results.

2.9 Summary

We propose and evaluate a unified approach for patch-based image matching that jointly learns a deep convolutional neural network for local patch representation as well as a network for robust feature comparison. Our system trains models that achieve state-of-the-art performance on a standard dataset for patch matching. We also evaluate a suite of architectural variations to study the tradeoff between accuracy vs. storage/computation. This work demonstrates that deep convolutional neural networks can be effective for general wide-baseline patch matching. In addition, an important feature of these results is that MatchNet can produce state-of-the-art accuracies while using significantly fewer bits per feature even than very recent work on compact feature representations, even with very simple quantization. This suggests that using deep learning approaches—and more advanced quantization—can make even more significant improvements in the accuracy/feature size trade-off.

We propose to model similarity using deep neural networks for an exact-match clothing item retrieval task called Exact Street to Shop. Our system learns category-specific similarity at object scale, and use it to re-rank initial retrieval results. Experiments show this approach significantly improves the retrieval accuracy over several baselines. These methods provide an initial step towards enabling accurate retrieval of clothing items from online retailers.

CHAPTER 3

MULTIPLE-VOXEL PATTERN CLASSIFIER LEARNING FOR FMRI IMAGES[10]

3.1 Introduction

Studies of human and nonhuman primates have consistently shown that the ventral temporal and occipital regions are involved in the perception and recognition of visual stimuli (see review by Ungerleider & Haxby (1994)). These visual association regions in the posterior cortex show functional divisions specializing in categorical representation of objects such as faces, tools, words, etc. (e.g., Epstein & Kanwisher (1998); Chao et al. (1999)). It has been proposed that these regions are also involved in supporting visual working memory the short-term representation of visual stimuli that are no longer physically available (Postle, 2006; Ranganath & D'Esposito, 2005). Neuroimaging findings, however, have been inconsistent thus far. Some showed that the inferior temporal region (e.g., lateral fusiform gyrus) was active in tasks requiring holding faces (e.g., Druzgal & D'esposito (2003); Postle et al. (2003); Ranganath et al. (2004)) and in tasks requiring refreshing recently seen faces (e.g., Johnson et al. (2007)). Others, however, showed that the activity in the inferior temporal region was not long lasting (Jha & McCarthy, 2000) and subject to interference (Miller et al. (1993); Sreenivasan et al. (2007), but see Yoon et al. (2006) for different results).

Some investigators further examined the selectivity of the posterior visual association regions in representing specific visual working memory. Face and/or scene images were used as task stimuli in neuroimaging studies since the fusiform (FG) and parahippocampal gyri (PHG) are known to be more specialized in processing faces and scenes, respectively (e.g., Kanwisher et al. (1997); Epstein & Kanwisher (1998)). Participants were cued to remember a particular category of visual stimuli (e.g., Remember face but ignore scene, and vice versa), with the cue presented either prior to stimulus presentation for selective encoding (Gazzaley et al., 2005; Nobre et al., 2004) or after, for selective maintenance (Oh & Leung, 2010; Lepsien et al., 2005). Across studies, the PHG consistently showed elevated activity during selective encoding and selective maintenance of scene images. The FG, however,

[10]This chapter previously appeared as an article (Han et al., 2013) in the Journal of NeuroImage.

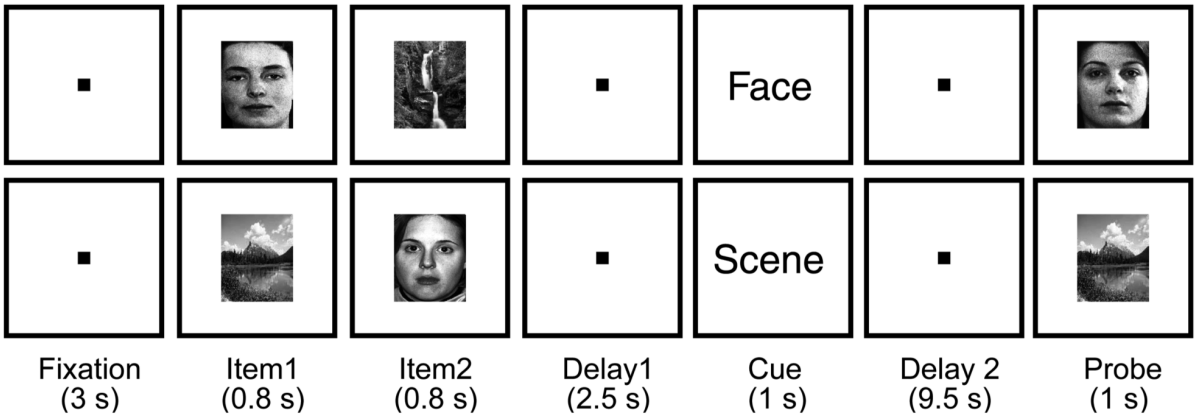

Figure 3.1: Visual working memory task with a selection cue. A face and a scene picture were presented (each for 800 ms counterbalanced in order) at the beginning of each trial for encoding. A word cue ("Face"/ "Scene") was then presented for 1000 ms to indicate the relevant picture category for selective maintenance. After a 9.5-s delay, a probe stimulus was presented for 1000 ms for the participants to make a match/non-match recognition judgment.

did not always show differential activity for selective processing of faces (compare: Oh & Leung (2010); Gazzaley et al. (2005)). A recent fMRI study reported that neither PHG nor FG was modulated by the number of face/scene images to be selectively maintained in working memory (Lepsien et al., 2011). Thus, it is unclear to what extent the different posterior association regions are involved in representing task-relevant visual working memory.

Most previous studies reviewed above applied univariate analysis to determine whether or not a brain region is activated while particular visual information is assumed to be held in working memory. Using multiple voxel pattern analysis (MVPA), recent studies successfully showed differential spatial patterns of activation in both striate and extrastriate areas for holding visual features (e.g., orientations, Harrison & Tong (2009)) and visual categories (e.g., faces, scenes and objects; Lewis-Peacock & Postle (2008); Lewis-Peacock et al. (2012) [Experiment 1]). Through reanalyzing data from the second experiment of the Lewis-Peacock et al. (2012) study, Lewis-Peacock & B.R. (2012) showed that their results on classification of task-relevant category (out of three potential categories: pseudowords, words, and line orientations) during the delay period were not affected even after excluding the suprathreshold voxels identified by the GLM as category-specific. Here, we further examined the activation patterns of the FG and PHG as well as other temporal/occipital regions in response to cued selective maintenance of task-relevant visual working memory in the presence of no-longer-relevant working memory.

We applied MVPA to previously published data (Oh & Leung, 2010) and conducted within-subject analysis to examine the activation patterns in the FG, PHG and other ventral temporal and occipital regions during selective maintenance of face/scene images. The task (Figure 3.1) comprised three phases: initial encoding (remembering two pictures, a face and a scene), selective maintenance (maintaining one of the two pictures according to a text cue), and recognition (judging whether the probe image is an exact match of the cued picture). We first trained and tested classifiers using activation patterns from the cue phase and examined classification performance across time during selective maintenance. In addition, we trained classifiers using activation patterns from the probe phase and from a separate localizer task, and tested these different classifiers on the cue-phase data to confirm that classification results for selective maintenance of faces/scenes are not due to the word cue itself. We were particularly interested in the FG and other ventral temporal and occipital regions involved in face processing since many of these regions did not show differential activity during selective maintenance in previous univariate analysis (see Figure 3.2).

3.2 Behavioral Tasks and Image Data

We used the 12 datasets from a study published by Oh & Leung (2010). A detailed description of the experimental procedure and image preprocessing can be found in that paper. Here, we provide a brief summary on the task design and image acquisition and processing procedures.

3.2 Working Memory and Localizer Tasks

The fMRI data were collected while participants performed a visual working memory task and a localizer task. For the main visual working memory task, we used a variant of the delayed recognition paradigm with a cue inserted during the delay period to study selective maintenance of faces or scenes. At the beginning of each trial, a fixation point (a small green square) was presented for 3 s and, as a warning, it turned into red color briefly before stimulus presentation. Two pictures (a face and a scene) were presented sequentially in random order, each for 800 ms, with a 200-ms gap in between. A mask (black-and-white checkerboard) was displayed for 200 ms after the offset of the second stimulus. After a delay of 2.5 s, a cue word (e.g., ”face”, ”scene”) was presented in the center of the screen for 1 s. This cue indicated the picture category relevant for the recognition test 9.5-s later. All cues were 100% valid. For

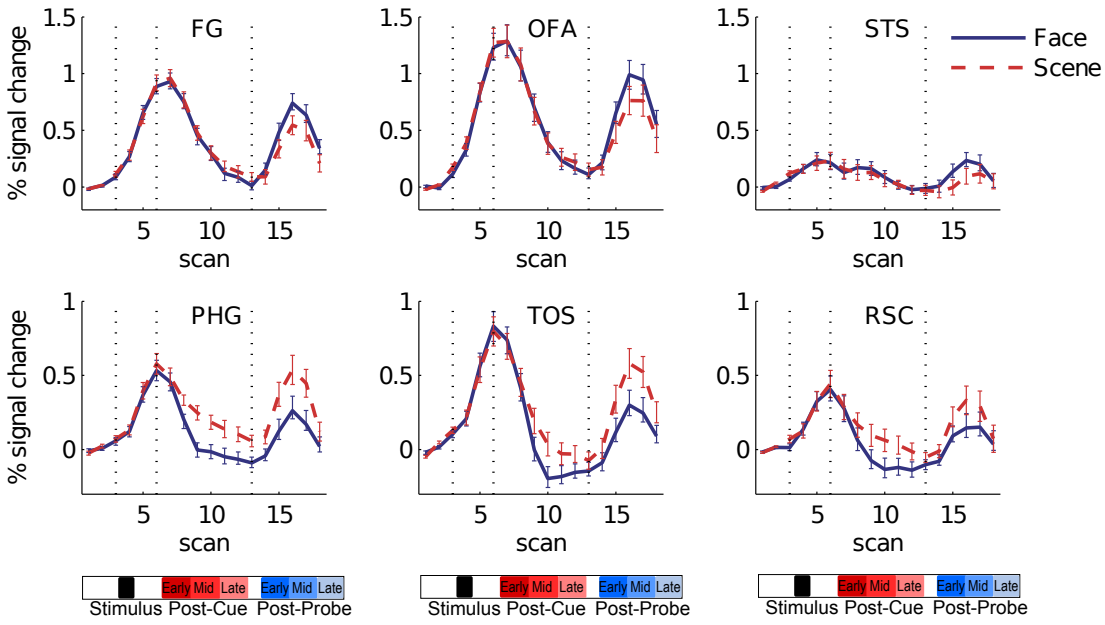

Figure 3.2: Average percent signal change in fMRI signal across time in regions associated with face and scene processing during the working memory task. The vertical dotted lines mark the onset of stimulus presentation, cue, and probe. The scales in the bottom show the division of the cue and probe phases into early, middle and late time segments, which were used in training and testing SVM classifiers.

trials with the face cue, the participants would only need to continually hold the memorized face picture as the probe would be either the cued face or a new face. It was the opposite for trials with the scene cue. The participants made button presses to indicate whether or not the probes matched the to-be-remembered picture. The inter-trial interval (ITI) varied between 8 and 14 s with a mean of 11 s. There were 20 trials with the face cue and 20 trials with the scene cue. The localizer task was in a 1-back working memory format. There were 8 task blocks (4 face blocks and 4 scene blocks) separated by resting fixation blocks. Each block was 16 s long. Within each task block, eight pictures were sequentially presented, each for 800 ms, with a gap of 1.2 s gap between the stimuli. The participants made a same/different response to each picture indicating whether or not it matched the preceding one.

3.2 Image Data Acquisition, Preprocessing and Defining ROIs

Anatomical and functional MR images were acquired with a 3 T Philips Achieva system using the standard quadrature head coil (8 channels). The acquisition parameters for the echo- planar (EPI) images were as follows for the main working memory task: 24 axial-oblique 5-mm slices/volume, 245

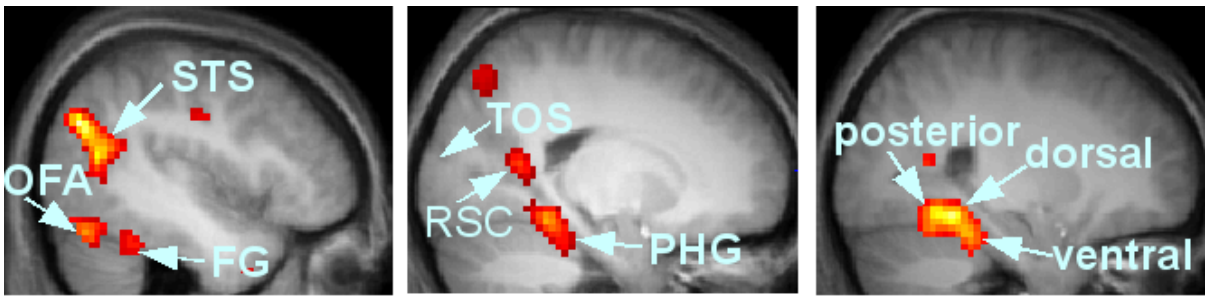

Figure 3.3: Functionally defined regions of interest. The rightmost image shows the three subdivisions of the PHG for finer scale MVPA. Abbreviations: FG, fusiform gyrus; OFA, occipitotemporal cortex or occipital face area; STS, superior temporal sulcus; PHG, parahippocampal gyrus; TOS, transverse occipital sulcus; RSC, retrosplenial cortex.

volumes/run, TR = 1.5 s, TE = 30 ms, flip angle = 80 degrees, FOV = 220x220 mm, matrix = 64x64 and ascending acquisition from the bottom slice. Similar parameters were applied for the localizer task, except a 2-s TR was used instead. All preprocessing steps were conducted using SPM2 (Wellcome Department of Cognitive Neurology, London, UK.) as reported in Oh & Leung (2010). Functional images were corrected for differences in slice timing and head motion. Images were normalized to the MNI gray matter template (Friston et al., 1995). We used smoothed images (8 mm Gaussian kernel), since we found little or no differences in our classification results using either nonsmoothed or smoothed images in a preliminary test.

Image data from the localizer task were used to define the visual association regions for each subject, using the MarsBar matlab toolbox (Brett et al., 2002) [11]. Figure 3.3 illustrates the locations of the regions of interest (ROIs). The primary regions of interest included three inferior temporal and occipital areas that showed greater activation in the Face¿Scene contrast (FG, OFA [occipitotemporal face area], STS [posterior superior temporal sulcus]), and three areas that showed greater activation in the Scene¿Face contrast (PHG, TOS [transverse occipital sulcus], and RSC [retrosplenial cortex]. These areas were defined in each hemisphere following the literature (Fox et al., 2009; Nasr et al., 2011) and guided by anatomy and group-level contrasts. Contrasts were thresholded at t>3. ROIs were defined as spheres (radius=3 voxels or 10.5 mm; approximately 123 voxels in volume) centered on the coordinates of the peak of the suprathreshold clusters in each individual. For a few subjects, we either used a lower threshold or used the contrast with fixation baseline to identify the coordinates; this was the case for TOS (1 right, 1 left), RSC (3 right, 3 left), OFA (3 left, 3 right), FG (1 left, 1 right), STS (3 right, 2 left). For

[11] http://marsbar.sourceforge.net

the two subjects where we could not identify activations in the RSC even at a lower threshold (t>1), we used the mean coordinates from the other subjects in the group.

3.3 Multiple-Voxel Pattern Analysis

3.3 Model and Features

We applied linear Support Vector Machines (SVMs) to examine the spatial response patterns in specific brain regions during the cue phase of the main task for predicting the Face/Scene cues. Regions were selected from the 6 ROIs (FG, OFA, STS, PHG, TOS and RSC) and some of their combinations, e.g., all face-related ROIs or all scene-related ROIs. All classification experiments used binary classification designed to distinguish between trials where subjects were cued to remember faces (face trials) and trials where subjects were cued to remember scenes (scene trials). The training features were voxel responses within an ROI extracted either from the cue or probe phase of the main task or from the separate localizer task. The test features were extracted from the cue phase of the main task. For comparison purposes, we also applied SVM to examine brain activity during the probe phase.

Our features are baseline-corrected voxel responses, specified by time and brain region. For the main task, we divided the cue phase into three time segments: early (7th and 8th scan), middle (9th and 10th scan) and late (11th and 12th scan), and similarly the probe phase into three segments: early (14th and 15th scan), middle (16th and 17th scan) and late (18th and 19th scan). (Scan numbers are counted from the beginning of a trial.) We normalized scans in every segment by subtracting the per voxel average of two baseline scans (1st and 2nd scan) and then averaged the two baseline-subtracted scans. We also constructed features using all segments in the cue or probe phase, in which case we concatenated all 6 baseline-corrected scans in the phase (e.g., Tables 1 and 2). For a particular time choice, voxels were selected from a particular ROI or a combination of ROIs in the baseline-corrected average scan of that time segment. For the localizer task, each face/scene block comprised 8 scans. The first two scans were averaged to form the baseline. The 5th and 6th scans were baseline corrected, and the averaged voxels of the two scans were selected for a region and used as features.

3.3 Cross-validation

The dataset for each subject consisted of 40 trials, 20 face trials and 20 scene trials. We evaluated linear classification accuracy using cross validation, leaving out one trial of each type. For each fold in cross validation, a classifier was trained on data from the remaining 38 trials and evaluated on the left out trial of each type. Each classification task specified the time segment for training and testing as well as the ROI(s) where features were extracted. For instance, one experiment–train with probe data and test with cue data for FG for the middle time segment–used measurements from the voxels in the left and right FG ROI extracted from the 16th and 17th scan, during the middle segment of the probe phase (and normalized as described above) as training data, and measurements from the same voxels in the 9th and 10th scan, during the middle segment of the cue phase, for testing. Classifiers were trained using the LibLinear SVM package [12] with L2-regularization, L2-loss function, and bias=1. The regularization vs. loss tradeoff parameter C was determined (from the set [0.001, 0.1, 1, 10, 1000]) for each cross-validation fold by using a subset of the training samples of each fold for nested cross validation. The accuracy of a classifier on a dataset is the mean across the accuracy on each fold.

The reported classification accuracy and standard error of the mean for an experiment are computed over the accuracies for each subjects dataset. Statistical significance of classification accuracy above chance (50%) was assessed using a 1-tail one-sample t-test. Bonferroni correction was applied when determining the significance of the p-values for classification tests across the three time segments of the cue/probe phase. Bonferroni correction was also applied when comparing classification performance across regions (e.g., the three face-related ROIs). In addition, we also conducted several permutation tests to evaluate the significance of classification results (following the procedures in Pereira & Botvinick (2011)). In general these tests resulted in confidence estimates similar to those from the t-test. Here we present only results from the t-tests.

3.4 Results and Discussion

3.4 Classification of Activation Patterns during Probe Recognition

To compare with previous findings, we first tested how accurately we could classify the probe category (face vs. scene) using activation patterns of the FG and ventral PHG from both hemispheres

[12] http://www.csie.ntu.edu.tw/~cjlin/liblinear/

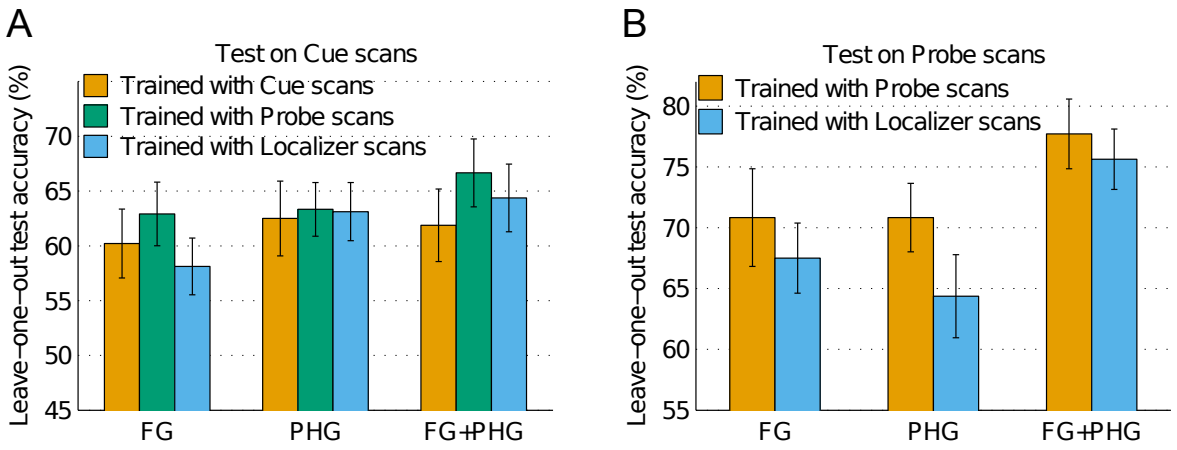

Figure 3.4: Mean accuracy in classification of face/scene during selective maintenance and probe recognition using voxels in the FG and ventral PHG. (A) SVM classifiers were trained with scans from the middle time segment of the cue phase (orange), probe phase (green) and localizer blocks (blue). Classifiers were tested on scans from the middle time segment of the cue phase. (B) Classifiers were trained with scans from the middle time segment of the probe phase (orange) and localizer blocks (blue), and tested on scans from the middle time segment of the probe phase.

Table 3.1: Average accuracy in classification of activation patterns of the FG and PHG across time (early, middle and late) for face and scene probes. Classifiers were trained and tested with data from the probe phase, with features combined from both hemispheres for each brain region. Note training and testing were done only on the corresponding time segments and never across time segments. Standard errors of mean are shown in parentheses. All values are significantly above chance (50%) except for the late time segment of vPHG. FG, fusiform gyrus; vPHG, ventral parahippocampal gyrus.

	All	Early	Middle	Late
FG	71.88% (4.21%)	64.17% (3.32%)	71.46% (4.51%)	58.75% (2.37%)
vPHG	65.21% (3.26%)	59.38% (3.29%)	70.63% (3.23%)	50.24% (3.56%)
FG+vPHG	78.12% (2.52%)	70.00% (2.04%)	76.88% (2.70%)	56.67% (2.23%)

during the probe stage of the trials. Table 3.1 shows the average within-subject classification accuracies for classifiers trained and tested using fMRI data from the three time segments (early, middle and late) of the probe phase. Classification performance was well above chance (50%) for discriminating activation patterns during the middle time segment of the probe phase (>70% mean accuracy with p's<0.001; see Figure 3.4B, yellow bars). These results were comparable to previous MVPA results for picture recognition (e.g., Morgan et al. (2011); Walther et al. (2009)). It should also be noted that classification performance varied across the three time segments of the probe phase, with lower classification accuracy for the early and late time segments (though all above chance, p's¡0.025 with Bonferroni correction, except for PHG during the late time segment). Similar results on classification of probe category were

Table 3.2: Average accuracy in classification of activation patterns in the FG and PHG across time (early, middle and late) for selective maintenance of face and scene working memory. Features were combined from both hemispheres for each brain region. Standard errors of mean are shown in parentheses. Values significantly above chance (50%) are shown in bold.

(A) Classifiers were trained and tested with data from the cue phase.

	All	Early	Middle	Late
FG	55.21% (3.58%)	49.17% (2.91%)	**62.29% (3.25%)**	52.50% (3.40%)
vPHG	**60.21% (3.61%)**	52.29% (2.81%)	**62.71% (3.50%)**	53.96% (4.43%)
FG+vPHG	**60.83% (3.61%)**	50.62% (2.84%)	**62.08% (3.34%)**	56.46% (3.70%)

(B) Classifiers were trained with data from the probe phase and tested on data from the cue phase.

	All	Early	Middle	Late
FG	**59.58% (2.34%)**	49.58% (2.42%)	**62.92% (3.18%)**	**57.71% (1.86%)**
vPHG	**57.29% (2.27%)**	53.54% (2.86%)	**63.75% (2.14%)**	53.54% (2.47%)
FG+vPHG	**61.46% (2.27%)**	53.12% (2.65%)	**66.04% (3.22%)**	54.17% (2.35%)

obtained by training classifiers with data from the localizer task (see Figure 3.4B, blue bars). It should be mentioned that the differences in mean classification accuracies produced by training on localizer data versus probe data were not significant (p's$>$0.15), while some differences were expected as the localizer task was a completely separate task with a different design and smaller total number of scans.

3.4 Classification of Activation Patterns during Selective Maintenance

A primary goal of this study was to examine whether the FG and PHG carry information about which visual category is selectively maintained in working memory. Using the leave-one- out cross-validation method, classifiers were trained and tested using data from the three time segments of the cue phase. Classification performance was significantly above chance for distinguishing activation patterns during the middle time segment of the cue phase ($>$62% mean accuracy with p's$<$0.006 with Bonferroni correction), but was around chance for the early and late time segments (see Table 3.2A).

To further test whether the neural activity patterns were representing the cued visual category and to rule out the potential confound of word classification in the above test results, additional classifiers were trained with data from the localizer task and the probe phase where no text cue was presented; these classifiers were tested on data from the cue phase. Classifiers trained with the probe data showed well above chance performance on differentiating the activation patterns in both FG and PHG by visual category during the middle time segments of the cue phase ($>$62% mean accuracy with p$<$0.001 and

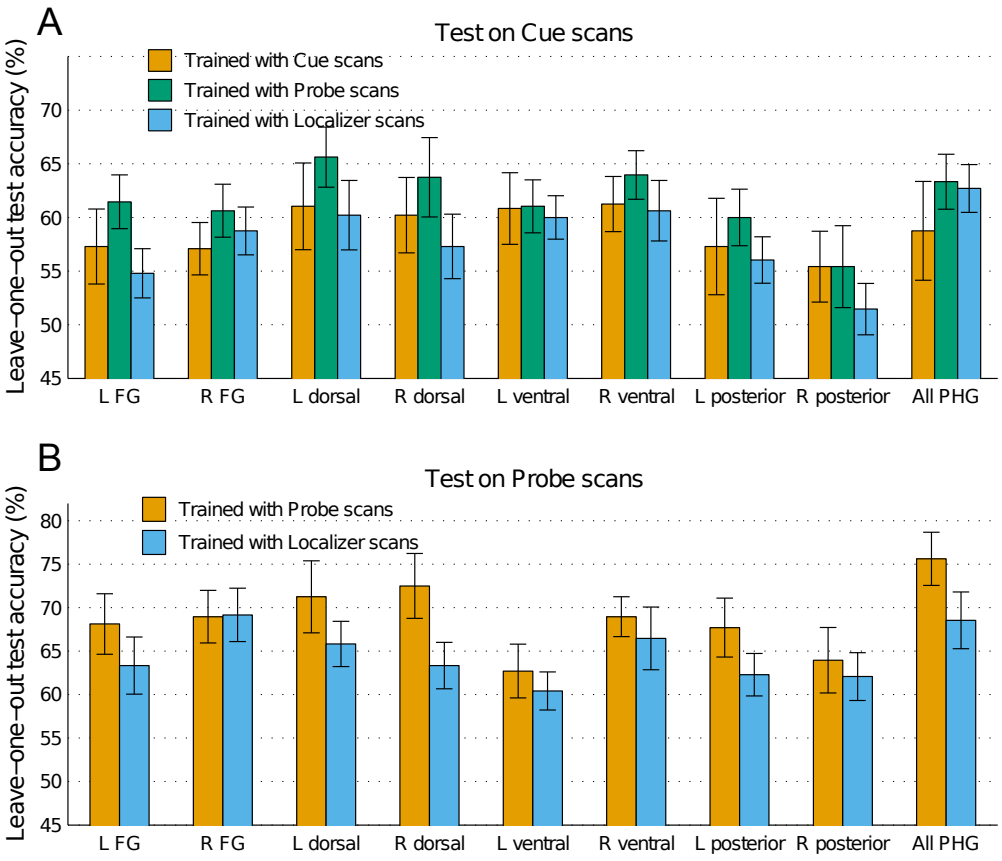

Figure 3.5: Mean accuracy in classification of face/scene during selective maintenance (A) and probe recognition (B) using voxels in the different parts (dorsal, ventral and posterior) of the PHG in the left and right hemisphere. The left and right FG are shown for comparison. Abbreviations: L:left, R:right.

p<0.0001, respectively; see Table 3.2B). Classifiers trained with the localizer data showed slightly lower but still above chance performance (>58% mean accuracy with ps¡0.01). See Figure 3.2A for a summary of the classification results using different training and testing combinations.

We further tested whether selective maintenance might be restricted to the left or right hemisphere of the FG and PHG (see Figure 3.5A). Since the PHG activation cluster was big, we also examined other subdivisions of the PHG (a more dorsal and a more posterior part) in comparison to the ventral PHG data described above (Arcaro et al., 2009; Sewards, 2011). For classifiers trained with probe data and tested on cue data (middle time segment), classification performance was significantly above chance for both the right and the left FG (both $\tilde{6}1\%$, p's<0.001). Classification performance for the dorsal-lateral and posterior-lateral portions of the PHG was comparable to the ventral-medial portion (reported above)

(all p's<0.007, except for the right posterior part, p>0.17). Not surprisingly, classification of the probe data by visual category was well above chance for all subdivisions of FG and PHG (all ps ¡0.001; see Figure 3.5B). Taken together, there were no clear hemispheric differences in classification results for either FG or PHG.

In sum, while previous univariate analysis did not yield significant differential activity in the FG for selective maintenance of faces, the MVPA analysis was able to use the spatial activation patterns from the middle time segment of the post-cue period in most individuals to differentiate the task relevant visual category.

3.4 Classification of Activation Patterns in other Visual Association Regions

Besides the FG and PHG, we examined several other temporal and occipital regions that have been previously implicated in face and scene processing (see Section 3.2.2). In terms of the average activity amplitude, only scene-related regions but not face-related regions showed differential effects by the cued visual category during selective maintenance (see Figure 3.1). Using the same approach described above, we conducted visual-category classification tests to examine the spatial response patterns of these visual cortical regions during the middle time segment of the probe and cue phases. Classifiers were trained with data from the probe phase. Figure 3.6 shows the classification results. Besides plotting the overall classification accuracy for each region as in the other figures, we plotted the classification accuracies by trial type (face/scene cue) in this figure.

Among face-related regions, classification of FG activity for selective maintenance of face/scene visual category was on average more accurate than OFA and STS (e.g., 63%, 56%, and 55%, respectively, as shown by the blue bars in Figure 3.6A). The mean classification accuracy was significantly above chance for FG (p<0.003) but not for OFA and STS (p's>0.08), after correction for multiple comparison. The overall classification accuracy for probe category, however, was comparable between FG and OFA (both about 70%; Figure 3.6D) and lower for STS (61%), though all well above chance (p's<0.002, corrected for multiple comparison). The overall classification results for the scene-related regions were more comparable, with classification of post-cue activation pattern slightly more accurate for ventral PHG than TOS and RSC (64%, 60%, 60%, respectively; blue bars in Figure 3.6A; all above chance, p's<0.02 after correction for multiple comparison). It is worth mentioning that relative to FG or PHG alone, combining data from the three face-related regions or the three scene-related regions did not change

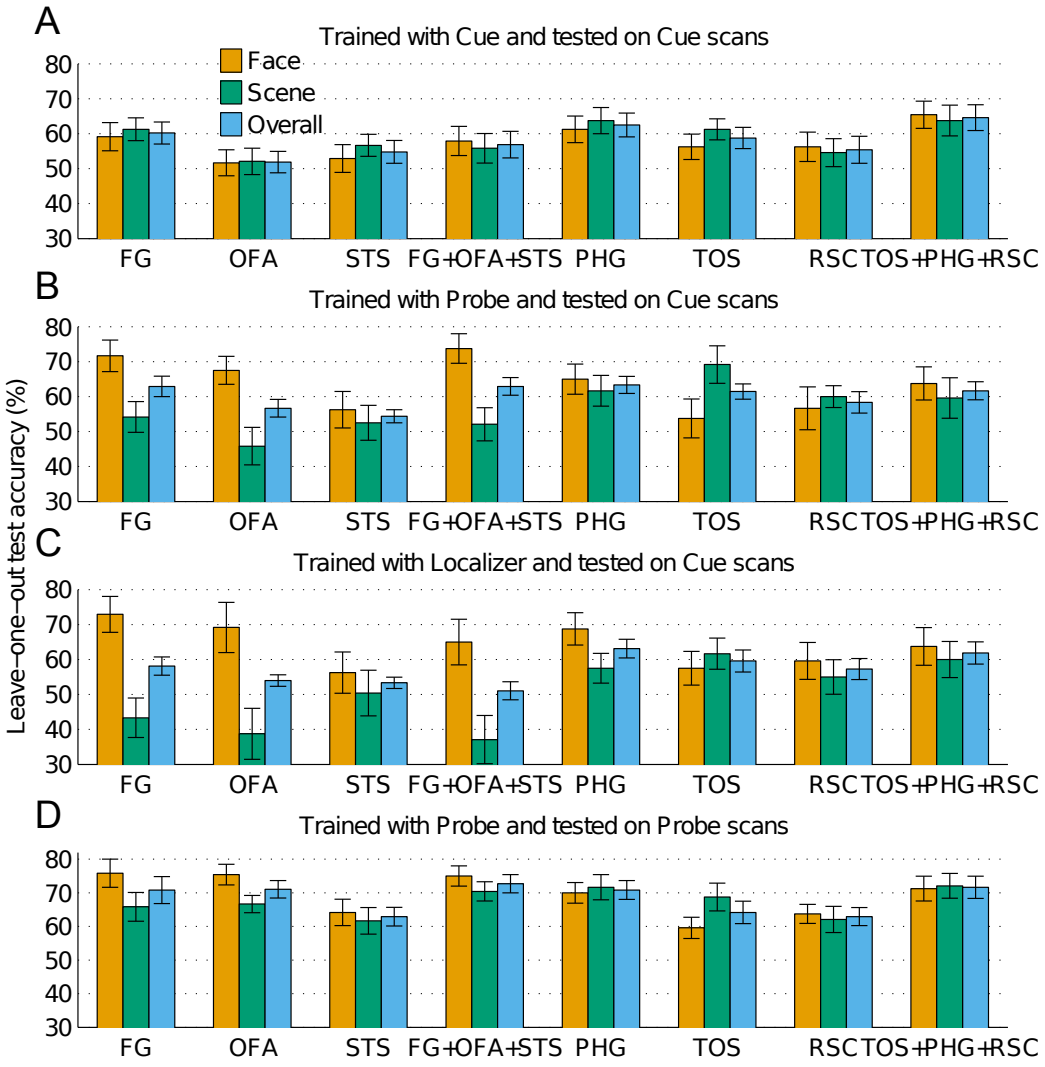

Figure 3.6: Mean accuracy in classification of face/scene during selective maintenance (A) and probe recognition (B) using voxels in regions associated with face and scene processing as shown in previous studies. Three bars are shown for each region, indicating classification performance for the face trials (orange; i.e., number of correctly classified face trials out of total number of face trials), the scene trials (green; i.e., number of correctly classified scene trials out of total number of scene trials) and the overall accuracy (average of the two, blue).

classification accuracy by much (¡0.5% difference in most cases). Intriguingly, for face-related regions such as the FG and OFA, better classification accuracy was achieved for their post-cue response patterns on face trials (65% or higher) while classification was around chance level for their response patterns on scene trials (50%); and such differences in classification were significant (pair t-tests: p's<0.04). In other words, most face trials were correctly classified as face trials while less than half of the scene trials were correctly classified as scene trials in these face-related ROIs. The opposite was not observed with the scene-related regions (p's>0.1), suggesting different ways of representing visual category information in working memory in the PHG versus FG.

Lastly, we conducted additional classification tests using activation patterns from brain areas that showed similar level of activity on face and scene trials. The mask included primarily frontal and parietal regions, defined using the conjunction map from our previous analysis (Oh & Leung, 2010). The classification accuracy for activation patterns of these regions during the cue phase, albeit significantly above chance (p's < 0.05), was all lower than 60% (between 55% to 60%, with classifiers trained by either cue or probe phase data; data not shown).

3.5 Discussion

A primary feature of visual working memory is the flexible representation of visual information most relevant to the current task demand. We utilized MVPA to examine whether the spatial response patterns of the individually defined ventral temporal and occipital regions reflect the selected visual category in working memory. Although previously we only found differential level of activity in the PHG for selective representation of scene images and no significant effects for the FG, present results from MVPA revealed differentiable spatial response patterns in both regions in correspondence to the cued visual category during post-cue maintenance and probe recognition. Classification of activation patterns for selective maintenance of face/scene category was well above chance even when classifiers were trained with data samples from the probe phase and the separate localizer task. For some other temporal and occipital regions that also showed differential response patterns to face versus scene probes, the classification of these regions' activity patterns during selective maintenance was less consistent (i.e., not always significantly above chance). These findings suggest that the ventral occipital and temporal

regions, especially the FG and PHG carry visual category representations relevant to the task, and their activity patterns during working memory maintenance may resemble those during probe recognition.

Findings from this study support the previous assertion that the ventral temporal and occipital areas contribute to keeping visual information in working memory. In particular, the present results from MVPA address the inconsistent results from previous univariate analysis on the involvement of FG in maintaining face working memory (Druzgal & D'esposito, 2003; Jha & McCarthy, 2000; Oh & Leung, 2010). By examining the spatial patterns of brain activation, we found that not only the PHG but also the FG exhibits differential activity patterns in correspondence to the cued visual category during the post-cue delay period. This was not likely caused by pure verbalization of the cue stimulus as classification was well above chance even when classifiers were trained with fMRI data from the probe phase and the completely separate localizer task. In fact, classification results on post-cue scans were worse when training with post-cue scans than when training with probe scans. This could be due to neural activity during the post-cue period being typically lower in amplitude and more variable across trials, whereas neural activity in response to the probe is usually stronger and more consistent across trials. The reliability of a classifier is dependent on the signal-to-noise ratio of the training data.

A recent study manipulated three categories of visual stimuli in a working memory task (line segments, pseudowords and words), and found brain activation patterns reflect the category currently relevant to the task demand but not the irrelevant category (Experiment 2 data from Lewis-Peacock et al. (2012); Lewis-Peacock & B.R. (2012). Our investigation extended such findings to specific visual association regions known to be involved in categorical processing. While our analysis cannot completely rule out whether or not the no-longer-relevant visual category was still represented in the visual association regions, examining the classification results for the two types of cue trials separately allowed us to gain some insight on how a region behaves on trials where the cued category matched its presumed categorical preference compared to trials of the non-preferred category (see Figure 3.6). On the one hand, the classification accuracy of activation patterns of the FG reached 70% for classifying the face trials as face trials but was at chance for classifying the scene trials as scene trials. Differences in classification of FG activation patterns on scene trials across subjects ranged from 20-85% accuracy, which suggest that some subjects might still maintain a face representation in the FG on scene trials while others might simply show some sort of non-face response patterns (or degradation in face representation) on scene trials. On the other hand, classification of activation patterns of the PHG on face trials and scene trials was similarly

above chance. These findings together with our previous finding of significantly reduced average activity in the PHG on face trials relative to scene trials suggest that the PHG's activity pattern might have changed on face trials (e.g., degraded) leading to the correct classification its activity on face trials as "non-scene" trials. Taken together, our results support the notion that the FG and PHG are specialized in representing the task-relevant visual category in working memory (Ranganath & D'Esposito, 2005). Importantly, our results further suggest that the FG and PHG probably use fundamentally different mechanisms in representing visual information and show different levels of degradation in representing its preferred visual category depending on the task goal. Perhaps reduction of face representation primarily occurs on situations demanding greater attention to a different category as in some other studies (Lewis-Peacock et al., 2012; Seidl et al., 2012). However, these previous studies did not examine the individual face- and scene-related regions so it is unclear whether or not there are differences in visual representation across different visual association regions. It is also possible that the SVM could be further optimized to distinguish multiple representations held in working memory. Further studies with different task designs are required to resolve this issue.

The present findings also suggest that the activation patterns of the posterior association regions during selective maintenance in the absence of the physical stimuli are similar to the patterns during probe recognition in the presence of the physical stimuli. Our findings are in line with an fMRI study by Lewis-Peacock & Postle (2008). They trained their participants to remember face-scene, face-object, object-scene pairs, and examined brain activation patterns in correspondence to the retrospective and prospective representation of recently presented visual images and the associated images, respectively. Their findings suggested that visual working memory is manifested as a reactivation of visual representations from long term memory. The posterior association regions indeed showed differential activation patterns during memory search of famous people, famous places and common objects from past experiences (Polyn et al., 2005). In our task, however, the face and scene target images were trial unique and novel or unfamiliar to the subjects, so the working memory representation during the post-cue delay period likely resemble the perceptual representation during the probe recognition stage, at least in terms of visual category. Thus, our findings in general support the hypothesis that the posterior association regions carry similar representations for visual perception, short-term, and long-term memory (Courtney & Ungerleider, 1997; Postle, 2006). The working memory representation of visual stimuli nonetheless may not be identical to the perceptual representation of visual stimuli, as the classification of activation

patterns during probe recognition was 10% higher than the classification of activation patterns during post-cue maintenance. It could be that the presence of a physical stimulus during the probe phase made a difference in category-specific pattern activity (but see above for alternative explanations). Future studies should investigate the cause of this degradation in classification performance for working memory maintenance as opposed to object recognition.

3.6 Summary

We applied MVPA to show that the spatial response patterns of specific posterior visual association regions including the fusiform gyrus and the parahippocampal gyrus carry information reflecting the task-relevant visual category in working memory. These findings of selective visual representation revealed by the multi-voxel activation patterns in the fusiform gyrus complimented our previous analysis which showed no differences in this region's average activity across the task conditions (Oh & Leung, 2010). In addition, our findings suggest that the activation patterns in these higher-order visual cortical regions associated with visual working memory representations are similar to the patterns associated with visual perception and recognition of different visual categories. It is possible that the visual representation maintained in working memory is a reinstatement of recent perceptual experience (see Postle (2006); Magnussen (2000)).

CHAPTER 4

DISTRIBUTED PARALLEL LEARNING FOR OBJECT RECOGNITION[13]

4.1 Introduction

As recognition in computer vision improves, researchers are pushing to recognize larger spaces of *labels*, from the 20 classes in Pascal VOC (Everingham et al., 2010) to 1000 classes in the Pascal LSVRC [14] to 10,000+ classes in ImageNet (Deng et al., 2010) to over 100,000+ classes in work on learning similarity functions for web scale retrieval (Chechik et al., 2010). Even these numbers may seem small in the future when fine grained recognition using attributes effectively increases the label space by orders of magnitude (Kumar et al., 2009; Parikh & Grauman, 2011). At the same time, high quality classification seems to require ever higher dimensional features whether for retrieval (Lin et al., 2011) or detection (Felzenszwalb et al., 2010). Together these factors present a formidable computational challenge.

This paper presents a new parallel algorithm for learning such large scale multiclass classifiers. In particular we present the first algorithm for efficiently distributing training of single machine classifiers, sometimes referred to as direct multiclass approaches. Such approaches offer potential advantages over the more commonly used *one versus all* multiclass techniques, but until now have been somewhat less scalable as no effective techniques were available for distributed parallelization.

This paper makes both a theoretical contribution – deriving a surprisingly simple sequential dual algorithm for parallel optimization of a direct multiclass SVM decision rule – and an experimental contribution – demonstrating that an implementation of this approach can significantly improve accuracy for a given amount of wall clock training time compared to the state of the art for single-machine methods using current computer vision data and features. As a result, single-machine methods can be trained on larger datasets than had been impractical in the past, and can be used more and more broadly.

To summarize our contributions we present:

---

[13] This chapter previously appeared in proceedings (Han & Berg, 2012).

[14] `http://www.image-net.org/challenges/LSVRC/2011`

- A new algorithm called Distributed Consensus Multiclass SVM (DCMSVM) for efficient distributed parallel training of "single machine" or direct multiclass SVMs. Making them much more widely usable. To our knowledge, this is the first such algorithm.
- An implementation of our algorithm.[15]
- Benchmarks on multiclass image classification including some using current high dimensional descriptors from state of the art systems. Results show significant improvements in wall-clock time and accuracy versus the very efficient implementation of Crammer & Singer's algorithm (Cramer & Singer, 2001) in Liblinear (Keerthi et al., 2008).

4.2 Background and Prior Work

Single-machine methods train a multiclass classifier by setting up a single large optimization problem tying together all parameters and as a result are computationally expensive. There have been a number of recent papers exploring parallelization of training for models in related contexts – for conditional maxent models (Mann et al., 2009), structured prediction(McDonald et al., 2010), stochastic gradient descent (Zinkevich et al., 2010), and others. Albeit to varying degrees, many of these approaches share a common idea of training models in parallel and combining the trained parameters. In fact the first stage of processing in the method proposed here follows the same procedure. One perhaps surprising result here is that, at least in some settings, it is possible to improve upon this initial step – in terms of the time / accuracy trade-off – by using the combined model as regularization for further optimization. Furthermore this can produce better classifiers than using more data (cf (Zinkevich et al., 2010)) or more time in the initial step.

We begin by considering a simple formulation for multiclass classification where a function f_c is learned for each class c and a data item x is classified into class $\mathrm{argmax}_c f_c(x)$. There are a number of ways to learn the f_c, but we first focus on the distinction between one-vs-all based methods and single-machine [16] methods. In one-vs-all, each f_c is trained *independently* to give a large response to data items from class c (the "one") and a small response to all other classes (the "all" sometimes called

[15] `http://www.cs.unc.edu/~xufeng/dcmsvm/DCMSVM_export.zip`

[16] The term "single-machine Multiclass SVM" was mentioned in (Chen & Tseng, 2011), and it refers to the method of solving k-class problem by constructing a single decision function, proposed in (Weston & Watkins, 1998)

the "rest"). The constraints for one-vs-all training generally have the form $f_c(x_p) > f_c(x_n)$ where x_p is any training item with label c and x_n is any training item with label $c' \neq c$. Single-machine approaches link the training of the functions f_c together so that for a training item x of class c, $f_c(x) > f_{c'}(x)$ for $c' \neq c$. There are many variations on those ideas – especially some considering the effects of different ways to regularize learning using margins – but the basic form of the constraints are often similar.

The single-machine approach has some potential advantages in weighing different types of errors during *training*. For instance it is possible to put more weight on avoiding confusing class a with class b than avoiding confusing class a with class c. The one-vs-all approach can only adjust weights on avoiding confusing class a with "any other class". This ability becomes more important when there is a natural hierarchical structure to classes (*e.g.* "animals", "dogs", and "black lab") where some are more similar than others and there is a notion of some labels being less different than others (*e.g.* "brown lab" and "black lab" vs "dog" and "cat"). In addition when there are many classes or classes are sampled sparsely or unevenly single-machine approaches may have an advantage. See Figure 4.5 for an example of the single-machine approach being more robust to some classes having a smaller number of training examples. Performance for the under-sampled classes is significantly higher for the single-machine approach as compared to one-vs-all (middle vs right of Figure 4.5).

Despite these potential advantages, it is much less straightforward to parallelize single-machine learning. This is in stark contrast to the one-vs-all approach, which for n classes easily distributes across as many as n machines or processors – simply training a single one-vs-all classifier on each node. Of course intermediate solutions where k one-vs-all classifiers are trained on each of $\frac{n}{k}$ nodes are also possible.[17]

In this work we explore an alternative approach that looks at training a multiclass classifier as a convex optimization problem in a general framework for parallelizing convex optimization, the alternating direction method of multipliers (ADMM). The result is a new algorithm distributing training of a single-machine classifier across multiple nodes. Training is distributed across nodes by partitioning the data, but terms are added to encourage the individual nodes toward a common (consensus) solution. This approach (exactly) linearly decreases the amount of data processed per node and (nearly) linearly decreases the

[17] We note that this easy parallelization of one-vs-all approaches still requires *all* the data to be available to train each classifier. See (Deng et al., 2011) for a clever solution requiring only part of the data on each node at the cost of some communication overhead.

computation time per node. Generally training proceeds in stages, where parameters from the parallel computations are averaged after each stage. The key difference from other work is that the averages from the previous round are used to regularize the following round. There are several attractive properties of the specific realization of this approach – it is easy to adjust the total amount of computation, there is low communication overhead (infrequent communication of model parameters, never data), and it is straightforward to minimize or even eliminate synchronization delays. The *key* mathematical insight making this efficient is that there is a simple formulation for a dual version of the training objective augmented with the consensus terms. This formulation is not only simple, but is also amenable to a very similar treatment as that utilized in efficient sequential dual methods for SVM training on a single machine.

Generally there is wide interest in scaling discriminative classifiers like SVMs to deal with increasingly large datasets. Computation becomes too great to fit on a single machine and there have been many pieces of work designed scale SVM training by carefully distributing kernel computations and values *e.g.*, (Chang et al., 2007) or splitting classification into a network of decisions (Graf et al., 2005; Collobert et al., 2002) and others. We note that these approaches and the methods in this chapter address distributing computation across a standard cluster of computers as opposed to specialized compute infrastructures where tighter parallelism is effective e.g. multi-core (Chu et al., 2007) or GPU style approaches. Nevertheless such techniques may indeed be used to speed up computation on individual nodes for our method.

In addition to the methods for efficient parallel learning of models mentioned in the introduction, Bengio *et al.* (Bengio et al., 2010) (*c.f.* (Deng et al., 2011)) present an approach for learning a tree structured classifier to improve efficiency for evaluation with a large number of image classes – a simpler approach but with similar motivation to Filter-Trees from (Beygelzimer et al., 2009). This is entirely compatible with our approach which could be seen as an effective way to scale the learning required for individual nodes in the decision tree which may still consider several hundred classes and can benefit from training on large amounts of data. In addition (Bengio et al., 2010) describe the landscape for efficient large scale classification.

The alternating direction method of multipliers approach is well researched, see Bertsekas & Tsitsiklis (D.Bertsekas & J.Tsitsiklis, 1997) using ADMM for consensus optimization and Boyd *et al.* (Boyd et al., 2011) for a wonderfully clear overview including summary and citations for convergence results.

We know of only one previous piece of work using an ADMM approach to develop a consensus algorithm for distributed SVM training from Forero *et al.* (Forero et al., 2010). Although the motivation and high level approach have similarities in spirit to our own, there are several differences. The main ones being that (Forero et al., 2010) do not derive or consider algorithmic solutions to the subproblem to be run on each node, and that they do not consider multiclass, either weighted or un-weighted. Finally they do not report results on the accuracy/time trade-off focusing more on high level capabilities for robust distributed optimization. In particular they look carefully at convergence to the non-distributed result (after a large number of iterations), which we find is far less relevant than accuracy after a small number of iterations – at least in the pursuit of reaping a wall clock time advantage from distributed computation.

In some recent work (Rifkin & Klautau, 2004) there is ongoing debate about whether single-machines are obviously better than one-vs-all approaches to multiclass classification. We note that they explicitly exempt large scale problems with very many non-uniformly sampled classes or classes with varying levels of distinction from consideration (hierarchies of classes) – exactly the cases we are interested in pursuing for many computer vision problems.

Finally as part of this work we have implemented our approach inside the Liblinear framework (Keerthi et al., 2008) which not only provides a solid baseline, but also a very flexible and useful environment for developing new algorithms that can be evaluated at large scale. It includes shrinking which we used and out-of-core processing which we did not use, but is compatible with out approach and would allow additional scaling with lower in-core memory requirements.

4.3 Parallelizing Multi-class Linear SVM Learning using Consensus Optimization

4.3 Consensus Formulation of the Learning Problem

We start from Crammer & Singer's K-class multiclass SVM formulation (Cramer & Singer, 2001):

$$\begin{aligned} \underset{w_1,\cdots,w_K}{\text{minimize}} \quad & \frac{\lambda}{2}\sum_{c=1}^{K} w_c^T w_c + \sum_{i=1}^{N}\xi_i, \\ \text{subject to} \quad & (w_{y_i} - w_c)^T x_i \geq 1 - \xi_i - \delta_{y_i,c} \\ & \forall i = 1,\cdots,N, c = 1,\cdots,K. \end{aligned} \tag{4.1}$$

Here each class c has a weight vector w_c, and each of the N training items x_i has label y_i. The indicator function $\delta_{y_i,c} = 1$ if $y_i = c$ and 0 otherwise. The variables ξ_i are slack variables for each data item, so

that only the margin between the correct class and the most confusing class is penalized – this is the main novelty of the Crammer & Singer formulation versus some others. Note that the constraint also compactly enforces that $\xi_i \geq 0$ when $c = y_i$.

One attractive aspect of the Crammer & Singer work is an efficient sequential dual algorithm for solving the problem. However the number of dual variables grows linearly with the number of training samples and with the number of classes (K). In the context of image classification, both K and the number of samples per class can be large, so we would like to split the data into smaller sets that are each tractable on a single computing node. We can break up the objective function over splits of the data. Let

$$f(\lambda, w_1, \ldots, w_K, \xi_1, \ldots, \xi_N) = \frac{\lambda}{2} \sum_{c=1}^{K} w_c^T w_c + \sum_{i=1}^{N} \xi_i \tag{4.2}$$

so that we can write the objective function in terms of S splits of the data:

$$f(\lambda, w_1, \ldots, w_K, \xi_1, \ldots, \xi_N) = \sum_{s=1}^{S} f(\frac{\lambda}{S}, w_1, \ldots, w_K, \xi_{(s-1)\frac{N}{S}+1}, \ldots, \xi_{s\frac{N}{S}}) \tag{4.3}$$

We can see that each split will solve for a multiclass classifier on a subset of the data with a smaller regularization parameter.

At the same time we want the solutions to all be the same, a consensus optimization problem.

4.3 Alternating Direction Method of Multipliers for Consensus Optimization

The idea of consensus optimization is to decompose the original problem into subproblems and solve each of the subproblems while at the same time constraining the solution to the subproblem to be equal. For instance if we can split our objective function f into S functions f_s so that $f(w) = \sum_{s=1}^{S} f_s(w)$, then we want to solve the following problem, where z is also the solution:

$$\text{minimize} \sum_{s=1}^{S} f_s(w_s) \tag{4.4}$$

$$\text{subject to } w_s = z, \quad s = 1, \ldots, S. \tag{4.5}$$

Although a simple dual decomposition followed by a dual ascent method can be used to solve the problem, it converges slowly. To help convergence, a linear and quadratic term are introduced, forming

the *augmented Lagrangian*:

$$L_\rho(w_1, \ldots, w_S, z, y) = \sum_{s=1}^{S}(f_s(w_s) + y_s^T(w_s - z) + (\rho/2)\|w_s - z\|_2^2), \tag{4.6}$$

where y are the dual variables. Note that the augmentation, effectively a smoothing term that aids convergence scaled by ρ, has no effect on the solution, only on the convergence of the algorithm to follow. It is shown in (Boyd et al., 2011) that at the k-th iteration z^k equals $\overline{w}^k$, the mean of w_s, so that we can optimize by alternately solving for:

$$w_s^{k+1} := \underset{w_s}{\text{argmin}}(f_s(w_s) + (y_s^k)^T(w_s - \overline{w}^k) + (\rho/2)\|w_s - \overline{w}^k\|_2^2) \tag{4.7}$$

$$y_s^{k+1} := y_s^k + \rho(w_s^{k+1} - \overline{w}^{k+1}). \tag{4.8}$$

These have a relatively simple interpretation: At the first part of each iteration, S separate optimizations are performed on the original objective functions f_s along with two terms that encourage consensus — one weighted by the dual variables y — and the augmenting term weighted by ρ. The second part of each iteration consists of calculating new dual variables y_s^{k+1}, and calculating $\overline{w}^{k+1}$ the mean of the w_s^{k+1}. We can consider $w^k = \overline{w}^k$ a practical solution after iteration k [18]. Note that only the w^k, $\overline{w}^{k+1}$ need to be communicated between the independent optimizations — hence the low communication overhead. So far we have introduced the standard consensus optimization framework using ADMM algorithm. We refer the reader to (Boyd et al., 2011) for a very clear review.

Substituting one of the objective functions in Equation 4.3 into the augmented consensus optimization in Equation 4.6, and renumbering the data items for each split as $1 \ldots N/S$ for notational simplicity we have

$$\begin{aligned} \underset{w_1,\cdots,w_K,\xi_1,\cdots,\xi_{N/S}}{\text{minimize}} \quad & \frac{\lambda}{2S}\sum_{c=1}^{K} w_c^T w_c + \sum_{i=1}^{N/S}\xi_i + \sum_{c=1}^{K}\alpha_c^T(w_c - \overline{w}_c) + \frac{\rho}{2}\sum_{c=1}^{K}\|w_c - \overline{w}_c\|^2, \\ \text{subject to} \quad & (w_{y_i} - w_c)^T x_i \geq 1 - \xi_i - \delta_{y_i,c} \\ & \forall i = 1, \cdots, N/S, c = 1, \cdots, K. \end{aligned} \tag{4.9}$$

Each subproblem above differs from the original Crammer & Singer problem due to the extra linear and a quadratic terms measuring how close the solution is to the consensus. The problem is still quadratic and convex, and ρ determines how much to weigh consensus.

[18] Projection of w^k to the constraints can be done by shrinking it, although it will not affect decision in test time.

4.3 A Sequential Dual Solver for the Sub-problem

A number of recent methods devised especially for Crammer & Singer multiclass SVM have been proposed (Teo et al., 2007; Collins et al., 2008; Joachims, 2006; Joachims et al., 2009; Keerthi et al., 2008). Keerthi et al's sequential dual method follows Crammer & Singer's original approach closely and demonstrated good performance in experiments (Keerthi et al., 2008). In search of a similar method to solve our consensus formulation, we derive the Langrangian dual (see Appendix A for details) of the subproblem Equation 4.9. The dual turns out to have the following simple form:

$$\begin{aligned} \underset{\beta}{\text{minimize}} \quad & h(\beta) = \frac{1}{2}\sum_{c=1}^{K} \|w_c(\beta)\|^2 + \sum_{i,c} \beta_{i,c} e_{i,c}, \\ \text{subject to} \quad & \beta_{i,c} \leq C\delta_{y_i,c}, \quad \sum_{c=1}^{K} \beta_{i,c} = 0, \\ & \forall i = 1, \cdots, N/S, \quad c = 1, \cdots, K, \end{aligned} \tag{4.10}$$

where $C = 1/(\lambda + \rho)$, $e_{i,c} = 1 - \delta_{y_i,c}$, and $w_c(\beta) = \sum_{i=1}^{N} \beta_{i,c} x_i - Ct_c$. Here, $t_c = \alpha_c - \rho\overline{w}_c$ is a constant vector for each class. Although the relation between the primal and dual variables, respectively w_c and β, are different from those in Crammer and Singer's dual formulation (Cramer & Singer, 2001), the resemblance in form suggests the sequential dual method (SDM) introduced in (Keerthi et al., 2008).

We develop a sequential dual method (SDM) to solve the subproblem defined by Equation 4.10, following the general procedure from Keerthi *et al*, with the inner-most optimization following Crammer and Singer's algorithm as described in Section 6 of (Cramer & Singer, 2002). The idea is to iteratively consider a data item x_j and solve for the corresponding dual variables $\beta_j = (\beta_{j,1}, \ldots, \beta_{j,K})$ (K is the number of classes) that minimize $h(\beta)$ with respect to β_j. The primal solution, $w = (w_1, \ldots, w_K)$, is then incrementally updated using the difference between the new β_j and the old. These along with other efficient implementation techniques such as shrinking are covered in (Keerthi et al., 2008).

The gradient of $h(\beta)$ is the key to whether a SDM is possible. In our problem, the gradient, expressed in terms of w as follows,

$$\begin{aligned} g_i^c = & \frac{\partial h(\beta)}{\partial \beta_{i,c}} = w_c(\beta)^T x_i + e_{i,c}, \\ & \forall i = 1, \cdots, N/S, \quad c = 1, \cdots, K. \end{aligned} \tag{4.11}$$

Algorithm 4.1 DCMSVM

- Divide the training data into S splits, $T_1 \dots T_S$;
- $\overline{w} \leftarrow 0$;
- FOR $k := 1$ TO max_iter DO
  FOR $s := 1$ TO S DO IN PARALLEL
  $\beta \leftarrow 0; \quad w^s \leftarrow -\frac{\rho}{\lambda/S+\rho}\overline{w}$
  UNTIL $v_i < \epsilon$ (tolerance parameter), $\forall i$ DO
  FOR $i := 1$ TO $|T_s|$ DO
  Compute g_i^c using Equation 4.11;
  $v_i \leftarrow \max_c g_i^c - \min_{c:\beta_i^c < C_i^c} g_i^c$ (See Equation 6 in (Keerthi et al., 2008));
  If $v_i < \epsilon$, find β_i' using FixedPointAlgorithm(Cramer & Singer, 2001)
  $\Delta\beta_i \leftarrow \beta_i' - \beta_i; \quad \beta_i \leftarrow \beta_i';$
  $w_c^s \leftarrow w_c^s + \Delta\beta_{i,c} x_i;$
  END
  END
  $\overline{w} \leftarrow \frac{1}{S}\sum_s w^s$; $\alpha_c \leftarrow \alpha_c + \rho(w_s - \overline{w})$;
  END
- RETURN $\overline{w}$;

has the same general form as that in (Keerthi et al., 2008). As a result the sequential update can be applied naturally and the only change in implementation necessary is to initialize $\beta = 0$ and $w_c = -Ct_c$. Details are covered in Appendix A. This has a simple, rough, interpretation as adding the consensus and augmenting terms by changing the "center" of the dual variables – effectively pushing the solution toward the average.

We write the the overall algorithm in Algorithm 4.1.

4.4 Evaluation

We perform a series of experiments to verify that our DCMSVM algorithm is effective for distributed parellelization of training a "single-machine" multiclass classifier, and that it compares favorably with respect to both Liblinear's Cramer & Singer implementation on a single node, as well as the standard one-vs-all, winner take all, approach to multiclass classification distributed across nodes.

For these experiments we use three datasets. All are subsets of the PASCAL Large Scale Visual Recognition Challenge dataset (LSVRC) [19], which in turn is part of ImageNet (Deng et al., 2009):

- **LSVRC100** consists of 100 classes randomly sampled from LSVRC with 800 images per class. Features are 1000 dimensional SIFT BOW released by LSVRC.
- **LSVRC1000** consists of all 1000 classes in LSVRC with 600 imagees per class. Features are 1000 dimensional SIFT BOW released by LSVRC.
- **LSVRC100-HD** consists of 100 classes randomly sampled from LSVRC with 600 images per class. Features are **210,000-dimensional** local coordinate code (LCC) features (Lin et al., 2011).

Each dataset is evenly divided into four folds for cross validation, and each training split is further divided into four "train"/"validation" subsets for parameter selection. For all methods, except where noted, the regularization parameter λ for each experiment was chosen to be the best on the train/validation sets for each experiment cross validation fold, and the tolerance parameter $\epsilon = 0.1$.

In the first ADMM iteration of our DCMSVM algorithm, each node solves the multiclass problem on its own split of the data in parallel. These parameters are averaged and used in the second iteration following to initialize the subsequent iteration following Equations 4.7&4.8.

Result Plots: We present most of the results in time/accuracy plots. In those plots, larger markers connected by thick solid lines show results after ADMM iterations in DCMSVM. Smaller markers connected by dotted lines show progress of the optimization inside each iteration, and are plotted after a certain number of passes of the sequential dual method through the data (after shrinking) in each problem. Since sequential dual methods like our DCMSVM and Liblinear's Cramer & Singer solver can be stopped at any given point and output the current estimate on w, those intermediate results are not difficult to obtain.

The first order result is that for all datasets, our DCMSVM produces classifiers that are at least as accurate (and usually more than as accurate) as the baseline 1-vs-all winner take all approach, and Liblinear's Cramer & Singer implementation. Furthermore in terms of wall-clock-time our approach trains these models more quickly than either Liblinear on a single node or 1-vs-all distributed on the same number of nodes as DCMSVM. See Figs. 4.1 and the LSVRC100-HD results in Sec. 4.4.4.

[19] `http://www.image-net.org/challenges/LSVRC/2010/`

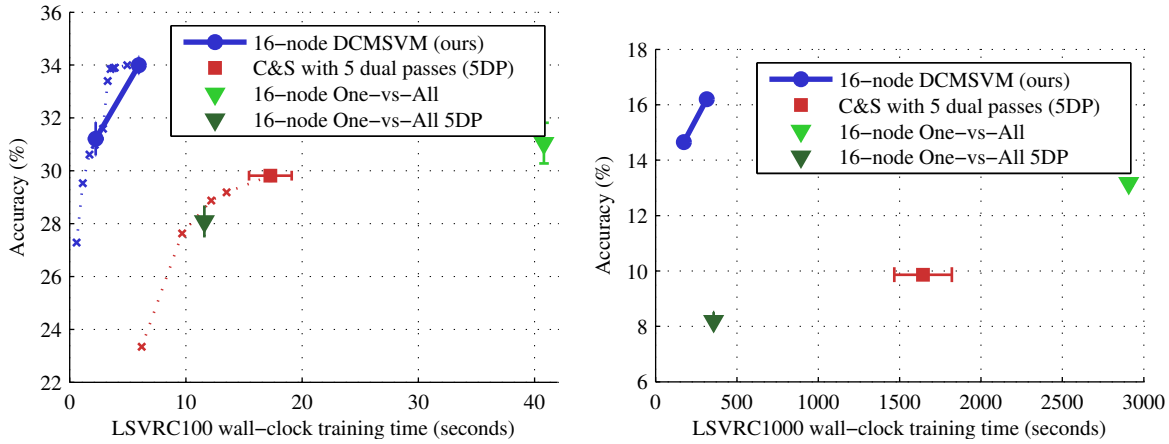

Figure 4.1: **Left:** Image classification on LSVRC100 (100 classes). **Right:** Image classification on LSVRC1000 (1000 classes). Comparison between our DCMSVM multiclass method and One-vs-all on 16 nodes, and Liblinear's Crammer & Singer implementation on a single node. Dashed lines in the left plot indicate progress during optimization. For instance, the first cross mark indicates the performance of the consensus averaged from subproblems solutions after one dual pass. (See Sec. 4.4 "Result Plots" for more explanation). Accuracy is measured on held out test data. Error bars indicate variation in time and accuracy across 4 different train/test splits (they are too small to see for DCMSVM in the right plot).

4.4 Time-Accuracy Trade-off under Different Number of Splits

As can be seen in all the figures, there is a significant reduction in wall clock time from distributing training. We see in Figure 4.1 (left) that DCMSVM with 16 splits after two iterations has higher accuracy than Cramer & Singer on a single node and is about 2.5 times faster in wall clock time, while doing about 6.5 times as much total computation. The progress of optimization is shown in many of the plots and indicates that more aggressive early stopping could produce even larger speedups. Note that for more than 16 splits we begin to see a decrease in accuracy after 2 iterations – although solutions are still significantly better than a single split. In Figure 4.2 we include a plot using 16 splits where each split has 1/8 of the data (instead of 1/16th of the data). This does not improve the accuracy vs wall-clock time trade-off, and careful examination of the progress during optimization shows that earlier stopping would not help. Even more impressive speedups are achieved for 1000 classes as shown in Figure 4.1 (right).

To make a fair comparison, we split the one-vs-all training tasks across the same number of nodes as used for DCMSVM. Note that one-vs-all training requires all the data to train each classifier, unlike DCMSVM which requires only a subset of the data in each split. In this paper all algorithms run single threaded on a node for the sake of comparison. All three can be accelerated by taking advantage of parallelism on each node.

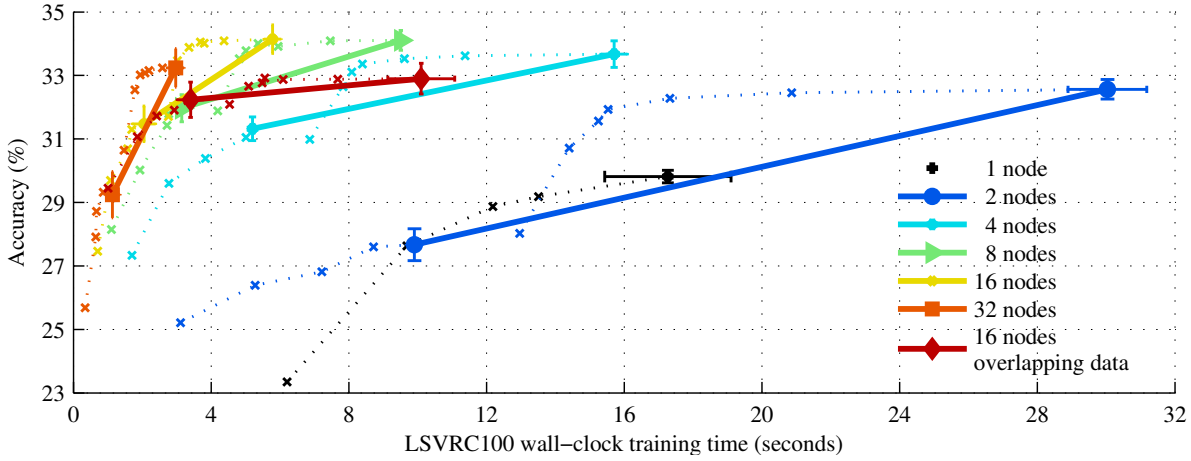

Figure 4.2: Comparison between our DCMSVM split across varying numbers of nodes and Liblinear's Crammer & Singer implementation on ILSVRC100. Dashed lines indicate progress during optimization. See Sec. 4.4.1 for details.

4.4 Convergence and Regularization

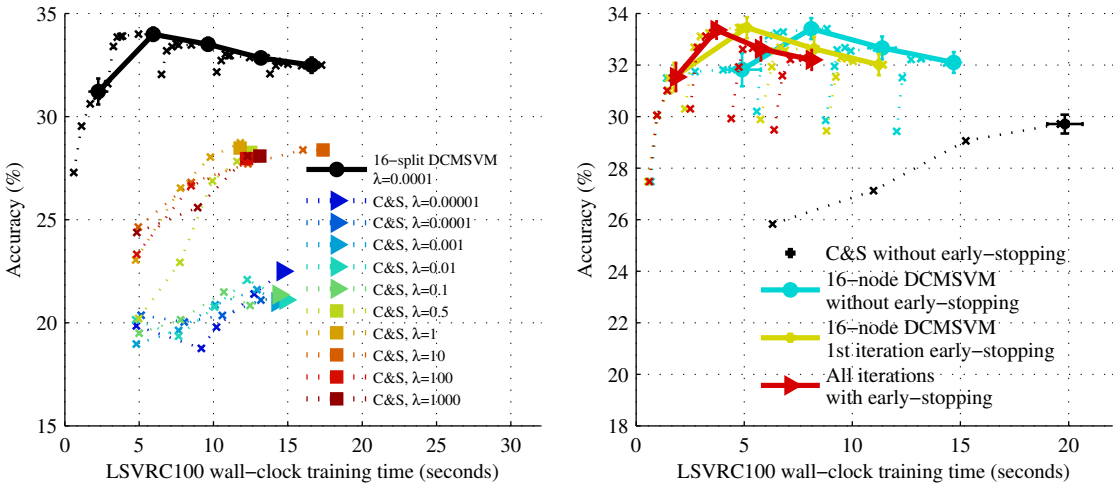

Figure 4.3: Comparison between our solution split across 16 nodes and Liblinear's Crammer & Singer implementation on the ILSVRC100 data. Dashed lines indicate progress during optimization. Accuracy is measured on held out data. Error bars indicate variation in time and accuracy across different train/test splits. **Left** Shows that no λ allows a single split to match 16 splits in generalization accuracy, see Sec. 4.4.2. **Right** Effects of early stopping over 6 validation splits ($\lambda = 10$) see Sec. 4.4.3.

ADMM methods are known to converge (Boyd et al., 2011) in the limit, although sometimes many iterations are required to achieve high accuracy in approximating the non-distributed objective. In practice only a few iterations of DCMSVM are necessary to get good solutions – in all our experiments, solutions after a few iterations were consistently superior to the non-distributed solution. A natural question would

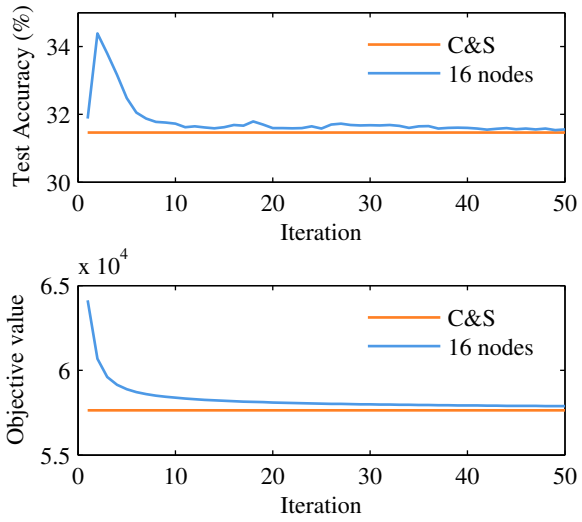

Figure 4.4: Convergence of DCMSVM (on 16 nodes) in terms of test accuracy (top) and objective value (bottom).

be whether the boost in performance is a result of some (positive) artifact of effective regularization from averaging solutions based on subsets of the data, and whether the same effect is achievable by adjusting λ in the original single machine formulation. Our answer to the second question is "No". Figure 4.3 (top) shows that no choice of λ allows the single-split version (Crammer & Singer) to match the 16 split version of DCMSVM[20]. We can choose the number of iterations to run as part of parameter search during cross-validation on training data – in our experiments cross validation almost always chooses 2 iterations. In Figure 4.4 we show more than two iterations to provide an idea of what happens as optimization progresses.

4.4 Early Stopping

In almost all experiments early stopping after 5 passes through the data (with shrinking) was used in the sequential dual optimization for DCMSVM. In Figure 4.3 we show results without early stopping and with early stopping. We also show the detailed progress of the Cramer & Singer optimization, showing that, while early stopping might help a little, the effects are not as dramatic as for DCMSVM. We also experimented with early stopping after five passes for one-vs-all training and found improved speed at the cost of lower classifier accuracy for both LSVRC100 and LSVRC1000 datasets. See Fig. 4.1.

[20]Fig. 4.3 also indicates some numerical instability for Liblinear's C&S implementation when using small λ.

4.4 High Dimensional Image Features

High dimensional features such as local coordinate coding used by (Lin et al., 2011) in their winning Pascal Large Scale Visual Recognition Challenge results have shown good discriminative power in large scale image classification tasks. We ran DCMSVM on our ILSVRC100-HD dataset to test its behavior in handling high dimensional features. For this experiment we use DCMSVM split over 8 nodes, achieving accuracy of 61.2% in 1509 seconds vs 61.7% in 2959 seconds for Liblinear's Cramer & Singer implementation.

4.4 Unbalanced Training Set

The LSVRC data provides an almost uniform number of examples per class – and we use an exactly uniform number of examples per class in most of our experiments for the purpose of benchmarking – but this is unrealistic in many scenarios. In fact unbalanced training data is one of the settings where a single machine, direct training, method for multiclass classifiers can have an advantage. To demonstrate this we train a classifier for the 100 class LSVRC100 dataset using Cramer & Singer and one-vs-all. We separate the classes into a set of 80 and a set of 20. Figure 4.5 top-left and bottom-left, show histograms of accuracy for the two sets of classes with equal training per class. When only one fifth of the training data is used for the 20 classes, then accuracy on those classes reduces dramatically, with *larger decrease* in performance for the one-vs-all approach (top-right) than for the single machine approach (bottom-right).

4.5 Summary

We derived an efficient consensus based distributed parallel training algorithm for Crammer & Singer's multiclass single-machine SVM formulation. The key mathematical insight was an efficient dual method for the consensus augmented optimization problem. The approach is validated with an implementation and extensive evaluation on image classification, showing advantages in both accuracy and wall-clock training time. This method allows single machine methods to scale to larger problems than had previously been impossible.

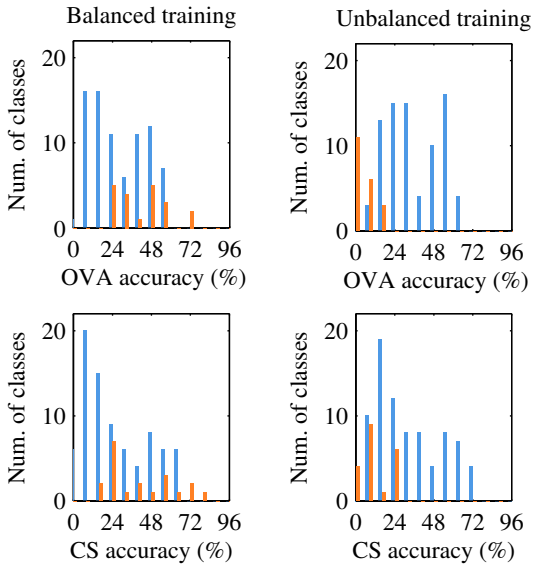

Figure 4.5: Solutions from Crammer and Singer (C&S) formulation is more stable for unbalanced training than those form One-vs-all (OVA). See Sec. 4.4.5.

CHAPTER 5

CONCLUSION

The main objectives of the research presented in this dissertation are to address the challenges introduced in Section 1.2 for scaling up models and algorithms for visual similarity and differentiation. The general methodology developed in this thesis is designing machine learning models and algorithms for both representation and decision with capacity matching the data. More specifically, we build MatchNet, a unified neural network framework for jointly modeling and learning both representation and decision for image patch similarity; we developed a learning based fMRI analysis approach combining multiple model selection approaches to achieve robust differentiation in the context of small sample size, high dimension features and high level of noise; we propose DCMSVM, a distributed algorithm for parallelizing multiclass support vector machine training to handle more categories and more samples in each category. In the following, we summarize our results (Section 5.1), and conclude (Section 5.2).

5.1 Summary of Results

Our research makes contributions in three areas, namely the design and implementation of neural network models for image similarity learning, the design and selection of models for fMRI differentiation analysis, and the design and evaluation of a parallelization algorithm for multiclass support vector machine. In the following, we recapitulate the key findings of Chapter 2–4.

5.1 MatchNet

The design and implementation of MatchNet are described in detail in Chapter 2. MatchNet is a unified approach for patch-based image matching that jointly learns a deep convolutional neural network for local patch representation as well as a network for robust similarity comparison. Our system trains models that achieve state-of-the-art performance on a standard dataset for patch matching, which provide a solution to custom similarity modeling as more and more domain specific data become available in practice.

We evaluate the feature-dimension-vs-accuracy trade-off by varying the number of nodes in the bottleneck layer (Note: the bottleneck layer is used for feature dimension reduction and performs a learned affine transformation followed by a ReLU activation) and found that the gain in accuracy diminishes exponentially as the dimension of the feature representation increases linearly.

We also evaluate the feature-precision-vs-accuracy tradeoffs and found MatchNet can produce state-of-the-art accuracy while using significantly fewer bits per feature even than very recent previous work on compact feature representations, even under simplistic quantization.

We show online sampling can be applied efficiently during training as a way of dealing with overfitting. This approach is effective because increasing sample variety and balancing training labels. We develop an efficient two-stage evaluation pipeline for a common matching scenario, which reduces the impact from expensive neural network *forward pass* computation on the system's efficiency.

We also apply this framework to "Exact Street-to-Shop", a novel cross-domain clothing item retrieval task, and show that the learned similarity outperforms standard cosine similarity. Our experiment also show that item localization is crucial for good performance. Retrieving item patches (obtained from object proposal methods) perform much better than retrieving whole images, even though current object proposal methods produces many noisy proposals.

These results supports the thesis that jointly modeling the representation and decision models results in models scalable with data for visual similarity tasks. They also serve as reference and guide for developing models or building systems for similar tasks.

5.1 MVPA

The details of our MVPA approach are described in Chapter 3. We develop a machine learning based approach for analyzing highly noisy, high dimensional featured, low sample size fMRI datasets. Models are trained with various regularization and model selection approaches that maximizes generalization accuracy. Performance is measured using cross-validation.

We apply the approach to analyzing the fMRI images and show that the spatial response patterns in certain brain regions carry information reflecting the task-relevant visual category in working memory. These finding complement previous analysis which show no differences in the region's average activity across the task conditions.

In addition, our findings also suggest that the activation patterns in these higher-order visual cortical regions associated with visual working memory representation are similar to the patterns associated with visual perception and recognition of different visual categories. It is possible that the visual perception maintained in working memory is a reinstatement of recent perceptual experience.

5.1 DCMSVM

The details of DCMSVM, a distributed method for training linear support vector machines for object classification, are described in Chapter 4 and Appendix A. This method addresses the challenge of training multi-class linear decision models on large number of categories and large number of samples per category by reformulating the learning objective such that the computation can be distributed across multiple computers and carried on in parallel.

We derived an efficient consensus optimization based algorithm for Crammer & Singer's multiclass single-machine SVM formulation. We showed the reduction from the core optimization problem in the new formulation to a simpler form which has efficient solution. The approach is validated with an implementation and extensive evaluation on image classification. It shows advantages in both accuracy and wall-clock training time in practice. The method allows single machine methods to scale to larger problems.

5.2 Closing Remarks

Visual similarity and differentiation are essential for computer vision systems. Both problems are often posted as modeling problems in two stages: representation and decision modeling. As acquisition technology improves, more and more data are available and are potentially useful for building these models. More data naturally helps estimate free parameters in the model, and yet it is still challenging to design models that make full use of the data and learning algorithms that scale with both. In this dissertation, we identify specific challenges in designing and learning these models as the quantity, dimension, imbalance, noise of the data increases, and provide systematic treatment for each one. It is our hope that the research is helpful for future exploration in this topic.

APPENDIX A

DERIVATION OF DCMSVM SUB-PROBLEM AND ITS SEQUENTIAL DUAL SOLVER[21]

A.1 Dual derivation of DCMSVM subproblem

The subproblem can be formulated in primal form as

$$\underset{w_1,\cdots,w_K,\xi_1,\cdots,\xi_N}{\text{minimize}} \frac{\lambda}{2}\sum_{c=1}^{K} w_c^T w_c + \sum_{c=1}^{K}\alpha_c^T(w_c-\overline{w}_c) + \frac{\rho}{2}\sum_{c=1}^{K}\|w_c-\overline{w_c}\|^2 + \sum_{i=1}^{N}\xi_i, \tag{A.1}$$

$$\text{subject to} \quad (w_{y_i}-w_c)^T x_i \geq 1-\xi_i-\delta_{y_i,c} \quad \forall i=1,\cdots,N, c=1,\cdots,K \tag{A.2}$$

where $\delta_{y_i,c}=1$ if $y_i=c$, 0 otherwise. Notice for $c=y_i$ the inequality constraints become $\xi_i \geq 0$. Remove the constant terms in (A.1), we have the following equivalent problem.

$$\underset{w_1,\cdots,w_K,\xi_1,\cdots,\xi_N}{\text{minimize}} \frac{\lambda+\rho}{2}\sum_{c=1}^{K} w_c^T w_c + \sum_{c=1}^{K}(\alpha_c-\rho\overline{w}_c)^T w_c + \sum_{i=1}^{N}\xi_i, \tag{A.3}$$

$$\text{subject to} \quad (w_{y_i}-w_c)^T x_i \geq 1-\xi_i-\delta_{y_i,c} \quad \forall i=1,\cdots,N, c=1,\cdots,K \tag{A.4}$$

We introduce multipliers μ for inequality constraints and form the Lagrangian.

$$\begin{aligned} L(w_1,\cdots,w_K,\xi_1,\cdots,\xi_N,\mu) = {} & \frac{\lambda+\rho}{2}\sum_{c=1}^{K} w_c^T w_c + \sum_{c=1}^{K}(\alpha_c-\rho\overline{w}_c)^T w_c \\ & + \sum_{i=1}^{N}\xi_i - \sum_{i,c}\mu_{i,c}\left((w_{y_i}-w_c)^T x_i - 1 + \xi_i + \delta_{y_i,c}\right). \end{aligned} \tag{A.5}$$

The dual function is

$$g(\mu) = \inf_{w_1,\cdots,w_K,\xi_1,\cdots,\xi_N} L(w_1,\cdots,w_N,\xi_1,\cdots,\xi_N,\mu). \tag{A.6}$$

Setting the derivatives of the Lagrangian with respect to w_c and ξ_i to zero, we get

$$\frac{\partial L}{\partial \xi_i} = 1 - \sum_{c=1}^{K}\mu_{i,c} = 0 \quad \Rightarrow \sum_{c=1}^{K}\mu_{i,c} = 1. \tag{A.7}$$

[21] This chapter previously appeared as supplementary materials for the paper (Han & Berg, 2012).

Similarly,

$$\frac{\partial L}{\partial w_c} = (\lambda+\rho)w_c + \alpha_c - \rho\overline{w}_c - \left(-\sum_{i=1}^{N} \mu_{i,c}x_i + \sum_{i=1}^{N} \delta_{y_i,c} \left(\sum_{q=1}^{K} \mu_{i,q} \right) x_i \right) \tag{A.8}$$

$$= (\lambda+\rho)w_c + \alpha_c - \rho\overline{w}_c + \sum_{i=1}^{N} (\mu_{i,c} - \delta_{y_i,c})x_i = 0, \tag{A.9}$$

which results in

$$w_c = \frac{1}{\lambda+\rho} \left(\rho\overline{w}_c - \alpha_c + \sum_{i=1}^{N} (\delta_{y_i,c} - \mu_{i,c})x_i \right). \tag{A.10}$$

Substitute (A.7) into the Lagrangian, we obtain the dual function represented only using dual variables.

$$\begin{aligned} g(\mu) = & \overbrace{\frac{\lambda+\rho}{2} \sum_{c=1}^{K} w_c^T w_c}^{S_3} \\ & + \overbrace{\sum_{c=1}^{K} \left(\alpha_c - \rho\overline{w}_c + \sum_{i=1}^{N} \mu_{i,c}x_i \right)^T w_c}^{S_1} \\ & - \overbrace{\sum_{i,c} \mu_{i,c} x_i^T w_{y_i}}^{S_2} - \sum_{i,c} \mu_{i,c}\delta_{y_i,c} + N \end{aligned} \tag{A.11}$$

Next we substitute (A.10) into the dual objective function (A.11). The constant vector $\alpha_c - \rho\overline{w}_c$ is denoted by t_c.

$$S_1 = \frac{1}{\lambda+\rho} \sum_{c=1}^{K} \left(t_c + \sum_{i=1}^{N} \mu_{i,c}x_i \right)^T \left(\sum_{j=1}^{N} (\delta_{y_j,c} - \mu_{j,c})x_j - t_c \right) \tag{A.12}$$

$$= \frac{1}{\lambda+\rho} \sum_{c=1}^{K} \left(\sum_{i,j} x_i^T x_j \mu_{i,c}(\delta_{y_j,c} - \mu_{j,c}) + \sum_{i=1}^{N} x_i^T t_c(\delta_{y_i,c} - 2\mu_{i,c}) - \|t_c\|^2 \right) \tag{A.13}$$

$$= \frac{1}{\lambda+\rho} \left(\sum_{i,j} x_i^T x_j \sum_{c=1}^{K} \mu_{i,c}(\delta_{y_j,c} - \mu_{j,c}) + \sum_{i=1}^{N} x_i^T (t_{y_i} - 2\sum_{c=1}^{K} t_c\mu_{i,c}) - \sum_{c=1}^{K} \|t_c\|^2 \right) \tag{A.14}$$

$$S_2 = \frac{1}{\lambda+\rho}\sum_{c,i}\mu_{i,c}x_i^T\left(\sum_{j=1}^{N}(\delta_{y_j,y_i}-\mu_{j,y_i})x_j - t_{y_i}\right) \tag{A.15}$$

$$= \frac{1}{\lambda+\rho}\left(\sum_{i,j}x_i^T x_j\sum_{c=1}^{K}\mu_{i,c}(\delta_{y_j,y_i}-\mu_{j,y_i}) - \sum_{i=1}^{N}x_i^T\sum_{c=1}^{K}\mu_{i,c}t_{y_i}\right) \tag{A.16}$$

$$= \frac{1}{\lambda+\rho}\left(\sum_{i,j}x_i^T x_j(\delta_{y_j,y_i}-\mu_{j,y_i}) - \sum_{i=1}^{N}x_i^T t_{y_i}\right) \tag{A.17}$$

$$= \frac{1}{\lambda+\rho}\left(\sum_{i,j}x_i^T x_j\sum_{c=1}^{K}\delta_{y_i,c}(\delta_{y_j,c}-\mu_{j,c}) - \sum_{i=1}^{N}x_i^T t_{y_i}\right) \tag{A.18}$$

$$S_3 = \frac{1}{2(\lambda+\rho)}\sum_{c=1}^{K}\left(\sum_{i=1}^{N}(\delta_{y_i,c}-\mu_{i,c})x_i - t_c\right)\left(\sum_{j=1}^{N}(\delta_{y_j,c}-\mu_{j,c})x_j - t_c\right) \tag{A.19}$$

$$= \frac{1}{\lambda+\rho}\left(\frac{1}{2}\sum_{i,j}x_i^T x_j\sum_{c=1}^{K}(\delta_{y_i,c}-\mu_{i,c})(\delta_{y_j,c}-\mu_{j,c}) - \sum_{i=1}^{N}x_i^T(t_{y_i}-\sum_{c=1}^{K}t_c\mu_{i,c}) + \frac{1}{2}\sum_{c=1}^{K}\|t_c\|^2\right) \tag{A.20}$$

Therefore, we have

$$S_3 + S_1 - S_2$$
$$= \frac{1}{\lambda+\rho}\left(-\frac{1}{2}\sum_{i,j}x_i^T x_j\sum_{c=1}^{K}(\delta_{y_i,c}-\mu_{i,c})(\delta_{y_j,c}-\mu_{j,c}) + \sum_{i=1}^{N}x_i^T(t_{y_i}-\sum_{c=1}^{K}t_c\mu_{i,c}) - \frac{1}{2}\sum_{c=1}^{K}\|t_c\|^2\right) \tag{A.21}$$

Substitue (A.21) into the dual objective function (A.11), we have the dual objective function

$$g(\mu)$$
$$= \frac{1}{\lambda+\rho}\left(-\frac{1}{2}\sum_{i,j}x_i^T x_j\sum_{c=1}^{K}(\delta_{y_i,c}-\mu_{i,c})(\delta_{y_j,c}-\mu_{j,c}) + \sum_{i=1}^{N}x_i^T(t_{y_i}-\sum_{c=1}^{K}t_c\mu_{i,c}) - \frac{1}{2}\sum_{c=1}^{K}\|t_c\|^2\right)$$
$$-\sum_{i,c}\mu_{i,c}\delta_{y_i,c} + N \tag{A.22}$$

Finally, after removing the constants we have the dual problem

$$\underset{\mu}{\text{maximize}} \quad g(\mu) = -\frac{1}{2(\lambda+\rho)} \sum_{i,j} x_i^T x_j \sum_{c=1}^{K} (\delta_{y_i,c} - \mu_{i,c})(\delta_{y_j,c} - \mu_{j,c}) - \sum_{i,c} \mu_{i,c} (\frac{x_i^T t_c}{\lambda+\rho} + \delta_{y_i,c}), \tag{A.23}$$

$$\text{subject to} \quad \mu_{i,c} \geq 0, \quad \sum_{c=1}^{K} \mu_{i,c} = 1, \quad \forall i = 1, \cdots, N, \quad c = 1, \cdots, K. \tag{A.24}$$

This problem is slightly different from the dual problem for Crammer & Singer SVM formulation(Crammer et al., 2001), where the coefficient for $\mu_{i,c}$ in the last term of the objective is just $\delta_{y_i,c}$.

A.2 Sequential dual method for the subproblem

Let $C = \frac{1}{\lambda+\rho}$, $e_{i,c} = 1 - \delta_{y_i,c}$, $\beta_{i,c} = C(\delta_{y_i,c} - \mu_{i,c})$. Notice

$$\sum_{c=1}^{K} \mu_{i,c} = 1, \qquad \sum_{c=1}^{K} \delta_{y_i,c} = 1, \qquad \sum_{c=1}^{K} \beta_{i,c} = 0. \tag{A.25}$$

Also multiplying $g(\mu)$ by C and adding constant terms will not change the optimal solution. We can rewrite (A.23) and (A.24) as

$$\underset{\beta}{\text{maximize}} \quad h(\beta) = -\frac{1}{2} \sum_{i,j} x_i^T x_j \sum_{c=1}^{K} \beta_{i,c}\beta_{j,c} + \sum_{i,c} \beta_{i,c}(C x_i^T t_c - e_{i,c}), \tag{A.26}$$

$$\text{subject to} \quad \beta_{i,c} \leq C\delta_{y_i,c}, \quad \sum_{c=1}^{K} \beta_{i,c} = 0, \quad \forall i = 1, \cdots, N, \quad c = 1, \cdots, K. \tag{A.27}$$

Rewrite (A.10) as

$$w_c(\beta) = \sum_{i=1}^{N} \beta_{i,c} x_i - C t_c, \tag{A.28}$$

and put it in the dual formulation, which gives (we change the sign of the objective so maximization becomes minimization).

$$\underset{\beta}{\text{minimize}} \quad h(\beta) = \frac{1}{2} \sum_{c=1}^{K} \|w_c(\beta)\|^2 + \sum_{i,c} \beta_{i,c} e_{i,c}, \tag{A.29}$$

$$\text{subject to} \quad \beta_{i,c} \leq C\delta_{y_i,c}, \quad \sum_{c=1}^{K} \beta_{i,c} = 0, \quad \forall i = 1, \cdots, N, \quad c = 1, \cdots, K. \tag{A.30}$$

The gradient of h is given by

$$h_i^c = \frac{\partial h(\beta)}{\partial \beta_{i,c}} = w_c(\beta)^T x_i + e_{i,c}, \quad \forall i = 1, \cdots, N, \quad c = 1, \cdots, K. \tag{A.31}$$

Now our ADMM sub–problem has been reduced to a form very close to what Keerthi *et al* used in their paper (Keerthi et al., 2008). In fact the only difference between the two is that we have an extra constant term Ct_c for each $w_c(\beta)$. Given that these terms are independent of βs, and w is incrementally updated, if we initialize Keerthi *et al*'s algorithm 2.1 with $\beta_{i,c} = 0$ (or $\alpha = 0$ by their notation) and with $w_c = -Ct_c$, the solution it gives will be the solution to our ADMM sub–problem.

Based on this reduction, with a little modification to the part of code for solving the Crammer&Singer SVM, LibLinear package is ready to solve our ADMM sub–problem.

REFERENCES

Arcaro, M. J., McMains, S. A., Singer, B. D., & Kastner, S. (2009). Retinotopic organization of human ventral visual cortex. *The Journal of Neuroscience*, 29(34), 10638–10652.

Bay, H., Tuytelaars, T., & Van Gool, L. (2006). SURF: Speeded up robust features. In *ECCV*.

Bengio, S., Weston, J., & Grangier, D. (2010). Label embedding trees for large multi-class tasks. In *NIPS*.

Beygelzimer, A., Langford, J., & Ravikumar, P. (2009). Error-correcting tournaments. *International Conference on Algorithmic Learning Theory (ALT)*, (pp. 247–262).

Boyd, S. P., Parikh, N., Chu, E., Peleato, B., & Eckstein, J. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1), 1–122.

Brett, M., Anton, J. L., Valabregue, R., & Poline, J. B. (2002). Region of interest analysis using an spm toolbox. *NeuroImage*, 16, 497.

Bromley, J., Guyon, I., Lecun, Y., Säckinger, E., & Shah, R. (1994). Signature verification using a "Siamese" time delay neural network. In *NIPS*.

Brown, M., Hua, G., & Winder, S. A. J. (2011). Discriminative learning of local image descriptors. *IEEE TPAMI*, 33(1), 43–57.

Brown, M. & Lowe, D. (2007). Automatic panoramic image stitching using invariant features. *IJCV*, 74(1).

Chang, E. Y., Zhu, K., Wang, H., Bai, H., Li, J., Qiu, Z., , H, & Cui. (2007). Psvm: Parallelizing support vector machines on distributed computers. In *NIPS*.

Chao, L. L., Haxby, J. V., & Martin, A. (1999). Attribute-based neural substrates in temporal cortex for perceiving and knowing about objects. *Nature Neuroscience*, 2, 912–919.

Chechik, G., Sharma, V., Shalit, U., & Bengio, S. (2010). Large Scale Online Learning of Image Similarity Through Ranking. *JMLR*, 11, 1109–1135.

Chen, C. C. & Tseng, Y.-D. (2011). Quality evaluation of product reviews using an information quality framework. *Decision Support Systems*, 50(4), 755–768.

Chopra, S., Hadsell, R., & LeCun, Y. (2005). Learning a similarity metric discriminatively, with application to face verification. In *CVPR*.

Chu, C.-T., Kim, S. K., Lin, Y.-A., Yu, Y., Bradski, G., Ng, A. Y., & Olukotun, K. (2007). Map-reduce for machine learning on multicore. In *NIPS*.

Collins, M., Globerson, A., Koo, T., Carreras, X., & Bartlett, P. L. (2008). Exponentiated gradient algorithms for conditional random fields and max-margin markov networks. *JMLR*, 9, 1775–1822.

Collobert, R., Bengio, S., & Bengio, Y. (2002). A parallel mixture of svms for very large scale problems. In *NIPS*.

Courtney, S. M. & Ungerleider, L. G. (1997). What fmri has taught us about human vision. *Current Opinion in Neurobiology*, 7(4), 554 – 561.

Cramer, K. & Singer, Y. (2001). On the algorithmic implementation of multiclass kernel-based vector machines. *JMLR*, 2, 265–292.

Cramer, K. & Singer, Y. (2002). On the learnability and design of output codes for multiclass problems. *Machine Learning*, 47.

Crammer, K., Singer, Y., Cristianini, N., Shawe-Taylor, J., & Williamson, B. (2001). On the algorithmic implementation of multiclass kernel-based vector machines. *JMLR*, 2.

Dalal, N. & Triggs, B. (2005). Histograms of oriented gradients for human detection. In *CVPR* San Diego, USA.

D.Bertsekas & J.Tsitsiklis (1997). *Parallel and Distributed Computation:Numerical Methods*. AthenaScientific.

Deng, J., Berg, A., Li, K., & Fei-Fei, L. (2010). What does classifying more than 10,000 image categories tell us? In *ECCV*.

Deng, J., Dong, W., Socher, R., Li, L., Li, K., & Fei-Fei, L. (2009). ImageNet: A large-scale hierarchical image database. In *CVPR*.

Deng, J., Satheesh, S., Berg, A., & Fei-Fei, L. (2011). Fast and balanced: Efficient label tree learning for large scale object recognition. In *NIPS*.

Druzgal, T. J. & D'esposito, M. (2003). Dissecting contributions of prefrontal cortex and fusiform face area to face working memory. *J. Cognitive Neuroscience*, 15(6), 771–784.

Epstein, R. & Kanwisher, N. (1998). A cortical representation of the local visual environment. *Nature*, 392, 598–601.

Erhan, D., Szegedy, C., Toshev, A., & Anguelov, D. (2014). Scalable object detection using deep neural networks. In *CVPR*.

Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., & Zisserman, A. (2010). The pascal visual object classes (voc) challenge. *IJCV*, 88(2), 303–338.

Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., & Lin, C.-J. (2008). Liblinear: A library for large linear classification. *Journal of Machine Learning Research*.

Felzenszwalb, P. F., Girshick, R. B., McAllester, D., & Ramanan, D. (2010). Object detection with discriminatively trained part-based models. *TPAMI*, 32(9), 1627–1645.

Forero, P. A., Cano, A., & Giannakis, G. B. (2010). Consensus-based distributed support vector machines. *JMLR*, 11, 1663–1707.

Fox, C. J., Iaria, G., & Barton, J. J. S. (2009). Defining the face processing network: optimization of the functional localizer in fmri. *Human Brain Mapping*, 30, 1637–1651.

Gatys, L. A., Ecker, A. S., & Bethge, M. (2015). A neural algorithm of artistic style. http://arxiv.org/abs/1508.06576.

Gazzaley, A., Cooney, J. W., McEvoy, K., Knight, R. T., & D'esposito, M. (2005). Top-down enhancement and suppression of the magnitude and speed of neural activity. *J. Cognitive Neuroscience*, 17(3), 507–517.

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*.

Graf, H. P., Cosatto, E., Bottou, L., Dourdanovic, I., & Vapnik, V. (2005). Parallel support vector machines: The cascade svm. In *NIPS*.

Han, X. & Berg, A. C. (2012). Dcmsvm: Distributed parallel training for single-machine multiclass classifiers. In *CVPR*.

Han, X., Berg, A. C., Oh, H., Samaras, D., & chung Leung, H. (2013). Multi-voxel pattern analysis of selective representation of visual working memory in ventral temporal and occipital regions. *NeuroImage*, 73, 8–15.

Han, X., Leung, T., Jia, Y., Sukthankar, R., & Berg, A. C. (2015). Matchnet: Unifying feature and metric learning for patch-based matching. In *CVPR*.

Harrison, S. A. & Tong, F. (2009). Decoding reveals the contents of visual working memory in early visual areas. *Nature*, 458, 632–635.

Heinly, J., Dunn, E., & Frahm, J.-M. (2012). Comparative evaluation of binary features. In *ECCV*.

Jain, P., Kulis, B., Davis, J. V., & Dhillon, I. S. (2012). Metric and kernel learning using a linear transformation. *Journal of Machine Learning Research*, 13, 519–547.

Jha, A. P. & McCarthy, G. (2000). The influence of memory load upon delay-interval activity in a working-memory task: an event-related functional mri study. *Journal of Cognitive Neuroscience*, 12, 90–105.

Jia, Y. & Darrell, T. (2011). Heavy-tailed distances for gradient based image descriptors. In *NIPS* (pp. 397–405).

Joachims, T. (2006). Training linear SVMs in linear time. KDD (pp. 217–226).: ACM.

Joachims, T., Finley, T., & Yu, C.-N. J. (2009). Cutting-plane training of structural svms. *Machine Learning*, 77(1), 27–59.

Johnson, M. R., Mitchell, K. J., Raye, C. L., D'Esposito, M., & Johnson, M. K. (2007). A brief thought can modulate activity in extrastriate visual areas: Top-down effects of refreshing just-seen visual stimuli. *NeuroImage*, 37, 290–299.

Kanwisher, N., Mcdermott, J., & Chun, M. M. (1997). The fusiform face area: A module in human extrastriate cortex specialized for face perception. *Journal of Neuroscience*, 17, 4302–4311.

Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., & Fei-Fei, L. (2014). Large-scale video classification with convolutional neural networks. In *CVPR*.

Ke, Y. & Sukthankar, R. (2004). PCA-SIFT: A more distinctive representation for local image descriptors. In *CVPR*.

Keerthi, S. S., Sundararajan, S., wei Chang, K., jui Hsieh, C., & jen Lin, C. (2008). A sequential dual method for large scale multi-class linear svms. In *KDD*.

Kiapour, M. H., Han, X., Lazebnik, S., Berg, A. C., & Berg, T. L. (2015). Where to buy it: Matching street clothing photos in online shops. In *ICCV*.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. In *NIPS*.

Kumar, N., Berg, A. C., Belhumeur, P. N., & Nayar, S. K. (2009). Attribute and Simile Classifiers for Face Verification. In *ICCV*.

Lazebnik, S., Schmid, C., & Ponce, J. (2006). Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR* (pp. 2169–2178).

Lepsien, Griffin, I. C., Devlin, J. T., & Nobre, A. C. (2005). Directing spatial attention in mental representations: Interactions between attentional orienting and working-memory load. *NeuroImage*, 26(3), 733 – 743.

Lepsien, J., Thornton, I., & Nobre, A. C. (2011). Modulation of working-memory maintenance by directed attention. *Neuropsychologia*, 49, 1569–1577.

Leung, T. & Malik, J. (2001). Representing and recognizing the visual appearance of materials using three-dimensional textons. *International Journal of Computer Vision*, 43(1), 29–44.

Lewis-Peacock, J. & B.R., P. (2012). Decoding the internal focus of attention. *Neuropsychologia*, 50(4), 470–478.

Lewis-Peacock, J. A., Drysdale, A. T., Oberauer, K., & Postle, B. R. (2012). Neural evidence for a distinction between short-term memory and the focus of attention. *J. Cognitive Neuroscience*, 24(1), 61–79.

Lewis-Peacock, J. A. & Postle, B. R. (2008). Temporary activation of long-term memory supports working memory. *The Journal of Neuroscience*, 28(35), 8765–8771.

Lin, Y., Lv, F., Zhu, S., Yang, M., Cour, T., & Yu, K. (2011). Large-scale Image Classification:Fast Feature Extraction and SVM Training. In *CVPR*.

Liu, S., Song, Z., Liu, G., Xu, C., Lu, H., & Yan, S. (2012). Street-to-shop: Cross-scenario clothing retrieval via parts alignment and auxiliary set. In *CVPR*.

Lowe, D. (1999). Object recognition from local scale-invariant features. In *ICCV*.

Lowe, D. (2003). Distinctive image features from scale-invariant keypoints. In *International Journal of Computer Vision*, volume 20.

Magnussen, S. (2000). Low-level memory processes in vision. *Trends in Neurosciences*, 23(6), 247 – 251.

Mann, G., McDonald, R., Mohri, M., Silberman, N., & Walker, D. (2009). Efficient large-scale distributed training of conditional maximum entropy models. In *NIPS*.

Matas, J. & Chum, O. (2004). Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, 22(10).

McDonald, R., Hall, K., & Mann, G. (2010). Distributed training strategies for the structured perceptron. In *NAACL*.

Mikolajczyk, K. & Schmid, C. (2005). A performance evaluation of local descriptors. *TPAMI*, 27(10), 1615–1630.

Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., & Gool, L. J. V. (2005). A comparison of affine region detectors. *IJCV*, 65.

Miller, E. K., Li, L., & Desimone, R. (1993). Activity of neurons in anterior inferior temporal cortex during a short term memory task. *Journal of Neuroscience*, 13(4), 1460–1478.

Molton, N., Davison, A., & Reid, I. (2004). Locally planar patch features for real-time structure from motion. In *BMVC*.

Morgan, L. K., MacEvoy, S. P., Aguirre, G. K., & Epstein, R. A. (2011). Distances between real-world locations are represented in the human hippocampus. *The Journal of Neuroscience*, 31(4), 1238–1245.

Nasr, S., Liu, N., Devaney, K. J., Yue, X., Rajimehr, R., Ungerleider, L. G., & Tootell, R. B. H. (2011). Scene-selective cortical regions in human and nonhuman primates. *The Journal of Neuroscience*, 31(39), 13771–13785.

Nobre, A. C., Coull, J. T., Maquet, P., Frith, C. D., Vandenberghe, R., & Mesulam, M. M. (2004). Orienting attention to locations in perceptual versus mental representations. *Journal of Cognitive Neuroscience*, 16, 363–373.

Norouzi, M., Punjani, A., & Fleet, D. J. (2014). Fast exact search in hamming space with multi-index hashing. *TPAMI*, 36(6).

Oh, H. & Leung, H.-C. (2010). Specific and nonspecific neural activity during selective processing of visual representations in working memory. *J. Cognitive Neuroscience*, 22(2), 292–306.

Parikh, D. & Grauman, K. (2011). Relative attributes. In *ICCV*.

Pereira, F. & Botvinick, M. (2011). Information mapping with pattern classifiers: A comparative study. *NeuroImage*, 56(2), 476–496.

Postle, B. R. (2006). Working memory as an emergent property of the mind and brain. *Neuroscience*, 139, 23–38.

Postle, B. R., Druzgal, T. J., & D'Esposito, M. (2003). Seeking the neural substrates of visual working memory storage. *Cortex*, 39(4â5), 927 – 946.

Ranganath, C., Cohen, M. X., Dam, C., & DEsposito, M. (2004). Inferior temporal, prefrontal, an hippocampal contributions to visual working memory maintenance and associative memory retrieval. *Journal of Neuroscience*, 24(16), 3917–3943.

Ranganath, C. & D'Esposito, M. (2005). Directing the minds eye: prefrontal, inferior and medial temporal mechanisms for visual working memory. *Current Opinion in Neurobiology*, 15, 175–182.

Rifkin, R. & Klautau, A. (2004). In defense of one-vs-all classification. *JMLR*, 5, 101–141.

Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. (2011). ORB: An efficient alternative to SIFT or SURF. In *ICCV*.

Salakhutdinov, R. & Hinton, G. E. (2007). Learning a nonlinear embedding by preserving class neighbourhood structure. In *AISTATS*.

Seidl, K. N., Peelen, M. V., & Kastner, S. (2012). Neural evidence for distracter suppression during visual search in real-world scenes. *The Journal of Neuroscience*, 32(34), 11812–11819.

Seitz, S., Curless, B., Diebel, J., Scharstein, D., & Szeliski, R. (2006). A comparison and evaluation of multi-view stereo reconstruction algorithms. In *CVPR*.

Sewards, T. V. (2011). Neural structures and mechanisms involved in scene recognition: A review and interpretation. *Neuropsychologia*, 49(3), 277 – 298.

Shakhnarovich, G. (2006). *Learning Task-Specific Similarity*. PhD thesis, MIT.

Simonyan, K., Vedaldi, A., & Zisserman, A. (2014). Learning local feature descriptors using convex optimisation. *TPAMI*.

Simonyan, K. & Zisserman, A. (2014). Two-stream convolutional networks for action recognition in videos. arXiv preprint arXiv:1406.2199.

Snavely, N., Seitz, S. M., & Szeliski, R. (2008). Modeling the world from internet photo collections. *International Journal of Computer Vision*, 80(2), 189–210.

Sreenivasan, K. K., Katz, J., & Jha, A. P. (2007). Temporal characteristics of top-down modulations during working memory maintenance: An event-related potential study of the n170 component. *J. Cognitive Neuroscience*, 19(11), 1836–1844.

Strecha, C., Bronstein, A. A., Bronstein, M. M., & Fua, P. (2012). LDAHash: Improved matching with smaller descriptors. *TPAMI*, 34(1), 66–78.

Teo, C. H., Smola, A., Vishwanathan, S., & Le, Q. V. (2007). A scalable modular convex solver for regularized risk minimization. In *ACM SIGKDD* (pp. 727–736).

Toshev, A. & Szegedy, C. (2014). DeepPose: Human pose estimation via deep neural networks. In *CVPR*.

Trzcinski, T., Christoudias, C. M., Fua, P., & Lepetit, V. (2013). Boosting binary keypoint descriptors. In *CVPR*.

Trzcinski, T., Christoudias, C. M., Lepetit, V., & Fua, P. (2012). Learning image descriptors with the boosting-trick. In *NIPS* (pp. 278–286).

Ungerleider, L. G. & Haxby, J. V. (1994). âwhatâ and âwhereâ in the human brain. *Current Opinion in Neurobiology*, 4(2), 157 – 165.

van de Sande, K. E. A., Uijlings, J. R. R., Gevers, T., & Smeulders, A. W. M. (2011). Segmentation as selective search for object recognition. In *ICCV*.

Vedaldi, A. & Fulkerson, B. (2010). Vlfeat: an open and portable library of computer vision algorithms. In *ACM Multimedia*.

Vitter, J. S. (1985). Random sampling with a reservoir. *ACM Trans. Math. Softw.*, 11(1), 37–57.

Vondrick, C., Khosla, A., Malisiewicz, T., & Torralba, A. (2013). Hoggles: Visualizing object detection features. In *ICCV* (pp. 1–8).: IEEE.

Walther, D. B., Caddigan, E., Fei-Fei, L., & Beck, D. M. (2009). Natural scene categories revealed in distributed patterns of activity in the human brain. *The Journal of Neuroscience*, 29(34), 10573–10581.

Wang, J., Song, Y., Leung, T., Rosenberg, C., Wang, J., Philbin, J., Chen, B., & Wu, Y. (2014). Learning fine-grained image similarity with deep ranking. In *CVPR*.

Wang, J., Yang, J., Yu, K., Lv, F., Huang, T. S., & Gong, Y. (2010). Locality-constrained linear coding for image classification. In *CVPR* (pp. 3360–3367).

Weiss, Y., Torralba, A. B., & Fergus, R. (2008). Spectral hashing. In *NIPS*.

Weston, J. & Watkins, C. (1998). *Multi–class support vector machines*. Technical Report CSD-TR-98-04, Department of Computer Sciences, Royal Holloway, University of London.

Winder, S. A. J., Hua, G., & Brown, M. (2009). Picking the best DAISY. In *CVPR*.

Yamaguchi, K., Kiapour, M. H., & Berg, T. L. (2013). Paper doll parsing: retrieving similar styles to parse clothing items. *ICCV*.

Yang, Y. & Ramanan, D. (2013). Articulated human detection with flexible mixtures of parts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(12), 2878–2890.

Yao, B., Bradski, G., & Fei-Fei, L. (2012). A codebook-free and annotation-free approach for fine-grained image categorization. In *CVPR*.

Yoon, J. H., Curtis, C. E., & D'Esposito, M. (2006). Differential effects of distraction during working memory on delay-period activity in the prefrontal cortex and the visual association cortex. *NeuroImage*, 29, 1117–1126.

Yu, K., Zhang, T., & Gong, Y. (2009). Nonlinear learning using local coordinate coding. In *NIPS* (pp. 2223–2231).

Zagoruyko, S. & Komodakis, N. (2015). Learning to compare image patches via convolutional neural networks. In *CVPR*.

Zbontar, J. & LeCun, Y. (2014). Computing the stereo matching cost with a convolutional neural network. *http://arxiv.org/abs/1409.4326*.

Zbontar, J. & LeCun, Y. (2015). Computing the stereo matching cost with a convolutional neural network. In *CVPR*.

Zhu, X. & Ramanan, D. (2012). Face detection, pose estimation, and landmark localization in the wild. In *CVPR* (pp. 2879–2886).: IEEE Computer Society.

Zhu, X., Vondrick, C., Fowlkes, C., & Ramanan, D. (2015). Do we need more training data? *IJCV*.

Zinkevich, M., Weimer, M., Smola, A., & Li, L. (2010). Parallelized stochastic gradient descent. In *NIPS*.