Open World Recognition

by

Abhijit Bendale

B.E., Electronics and Telecommunications, University of Pune, India 2007

M.S., Computer Science, University of Colorado at Colorado Springs, Colorado Springs 2009

A dissertation submitted to the Graduate Faculty of the

University of Colorado at Colorado Springs

in partial fulfillment of the

requirements for the degree of

Doctor of Philosophy

Department of Computer Science

2015

This dissertation for the Doctor of Philosophy degree by

Abhijit Bendale

has been approved for the

Department of Computer Science

by

Terrance Boult, Chair

Jugal Kalita

Rory Lewis

Sébastian Marcel

Xiaobo Zhou

Date

Bendale, Abhijit (Ph.D., Computer Science)

Open World Recognition

Dissertation directed by El Pomar Professor, Chair Terrance Boult

As humans, we encounter countless objects daily. We effortlessly recognize object across variations despite the fact that the objects might vary in size, scale, translation or rotation. Humans can identify previously seen objects and posses the ability to learn new instances with minimal or no supervision. Human visual system continues to learn and adapt to ever changing surroundings. In recent years, there have been significant advances in the field of computer based recognition systems. While significant strides have been made towards building automated recognition systems, these systems face multiple challenges when operating in evolving environments. Operational issues such as changing data distributions, perturbations in input/output conditions and ever changing requirements of the system users, pose challenges in operational environments. In this work we highlight specific operational challenges such as handling partial information, incremental model adaptation, large-scale classification and propose solutions towards addressing these challenges

Dedication

This thesis is dedicated to my family for their love, constant support, encouragement and patience.

Acknowledgements

I would like to thank Prof. Terrance Boult for giving me the opportunity to work with him. Terry provided right amount of mentoring, freedom, endless support and constantly encouraged me to push myself to redefine my abilities. I continue to admire his endless enthusiasm, his fearlessness in the face of hard research problems and selflessness for shielding me from real-life worries like grants/funding. Terry enabled me to have a first rate research career, and for this, I will be forever grateful to him.

I would like to thank my committee members Prof. Jugal Kalita, Prof. Rory Lewis, Prof. Sébastian Marcel and Prof. Xiaobo Zhou for encouraging me to see the "big picture" at various stages in thesis. Special thanks to Prof. Sébastian Marcel for inviting me to spend a memorable winter in Switzerland, arranging logistics and providing excellent research atmosphere. I would also like to thank Patricia Rhea and Ali Langfels for administrative support.

Over the years, many friends had a profound impact on my thinking as a researcher. I would like to thank Pranav Mistry for constantly inspiring me and teaching me everything I know about innovation. I got to learn a lot from Lining Yao about research and life through our countless conversations. I would like to thank Nikhil Naik for keeping me grounded and giving a real-life perspective about research. Archana Sapkota provided endless support over the years, both in research and life in general. This thesis would be incomplete without mentioning her. Walter Scheirer served as an excellent fellow student and and inspiring mentor over the years. Special thanks to Ginger Boult for making countless ski trips, conferences and house parties memorable. Special thanks for Radhika Marathe for being my agony aunt over the years. Dr. Manuel Güenther was an excellent collaborator during my stay in Switzerland and continues to be a great friend and collaborator at UCCS. Special thanks for Dr. Bill Triggs for providing valuable insights about research and academia in

general.

Many friends and lab mates over the years made PhD very enjoyable. Friends from VAST lab Lalit and Ankita Jain, Michael Gohde, Michael Wilber, Steve Cruz, Svati Dhamija, Chloe Bradley, Ethan Rudd and Anthony Magee made life fun and rewarding. MIT - BCS friends Nicolas Pinto, Elias Issa, Grant Mulliken and Kailyn Schmidt continued to provide wisdom over the years. MIT Media Lab friends Otkrisht (Samuel) Gupta and Roarke Horstmeyer were awesome over the years. IDIAP friends Elie Khoury, Laurent El Shafey and members of Biometrics Group at Idiap were very helpful in making my stay in IDIAP memorable. Special mention to Dr. Connor McGann and Dr. Jeffrey Davitz for helping with all the logistics at Solariat and Lingonautics.

Thanks to Charudatta Katdare and Prashant Bagale for being my lifelong friends and for providing me right perspective in life. I am indebted to them for everything they have done for me. Abhijit Bhoite provided insights at key stages in life for which I am forever grateful to him. Ashish Gupta provided constant support over the years and has been a great friend. Svati for everything she has done over the years. Finally, I would like to thank my sister for being constant source of inspiration and role model throughout life, Jonathan and Palak for loving me unconditionally. This thesis is dedicated to my parents for all the quiet sacrifices they have made over the years.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

> Originality is the art of concealing the source. Nothing is original, everything is inspired.
>
> Franklin P. Jones

1.1 Recognition System

As humans, we encounter countless objects daily. We effortlessly recognize object across variations despite the fact that the objects might vary in size, scale, translation or rotation. We can identify previously seen objects and learn new object instances with minimal or no supervision. Human visual system continues to learn and adapt to ever changing surroundings. In recent years, there have been significant advances in the field of computer based recognition systems. Automated computer based recognition has found applications in wide variety of areas from wafer inspection systems in a chip manufacturing plant to self driving cars and from a face recognition system at border control to identifying category of a plant. These applications have been possible due to advances in fundamental problems in computer vision such as motion analysis, object detection, camera calibration, scene reconstruction, learning methods, decision making etc. While these advances have made possible in creating stand alone systems that can automatically detect tumors [159], recognition systems face multiple challenges when operating in “evolving” environments. Operational challenges such as changing data distributions, perturbations in input/output conditions and ever changing requirements of the system users makes recognition in the wild extremely challenging. For instance, Even partial perturbation in input space [90] can cause a system to fail and thereby demand for alternate approach for system building.

1.1.1 Operational Challenges of Recognition System

Building automated recognition system is challenging primarily because the conditions under which the system was built will differ from those in which the system will be used. In the real world, environments are non-stationary and is often impractical to match the development scenario (either due to high costs, operational infeasibility etc.)[122]. Majority of off-the-shelf recognition systems ignore operational scenarios: they either presume the test domain and training domain (in classification stage of recognition system) match or make no difference if these scenarios do not match [61]. Let us consider this issue more systematically. The goal of the recognition system is to learn a model Θ, given some data, which can then be used to make predictions $P(y|x)$ for some targets y given some new observation x_{new}. In many real-world applications, non-stationary environments exist. A face recognition system might have been trained on data collected from a busy street in a city (e.g. New York) and is then deployed for operation is a deserted snow street in a small Alaskan town. A plant recognition system might contain training data from North American species of plants, though its application environment might be a tropical rainforest. With such scenarios being a commonplace, it is important to ask *if a different predictive model Θ_{new} should be used, a different learning method should be used or is there just some ad-hoc post-processing that can be done to the learned model to account for such changes?*

In order to manage such ill-posed scenarios, it is important to develop a critical understanding of operational scenarios. In some cases, these changes can be session dependent [101]. In others, the changes could be domain dependent [61], [54]. While in yet another case, it might be advisable to perturb the learned model or keep some operational data to re-estimate the model parameters. These operational changes in the context of face recognition could be capturing images of a person across multiple days (session variability), across different domains (labelled faces in the wild [59] or to adapt a pre-trained model to incoming data [66]. The knowledge of how best to model the potential changes will allow creation of better recognition systems suitable for widely varying operational scenarios. Modeling for such domain shifts involves estimating the map between representations from source and target domain. This is a challenging problem, but one with significant real-world consequences. The issue of dataset bias and the damage it can cause to trained vision systems has been documented in the works of [72], [118] and [163]. Failing to model for these changes often leaves the system with dataset bias with a huge impact on real-world performance. Furthermore, models that work well in static scenarios (such as the use of a conditional model) can fail in situations of shift. By characterizing the different forms of dataset shift, we can begin to get a handle on the ways the data can be expected to change [122]. Development of such methods will help to automate the process of adaptation. While this problem is not specific to a particular learning method or inability to model a particular data distribution shift, it is important to take a system's perspective. Such systems perspective will allow us to understand the inter-play of various aspects of the operational change (whether it is change in distribution of data, its characteristics, its properties, learning methodology or all of them).

The problem of learning models in evolving domains has been studied from multiple perspectives in the fields of machine learning and computer vision. In the field of domain adaptation the aim is to build classifiers (or learning methods) that are robust to mismatched distributions. It has been noted previously that domain adaptation methods suffer from sample selection bias. The assumption in domain adaptation is that the task to accomplish is same in both the source and target domain [61]. In an related field of transfer learning, the task to accomplish between source domain and target domain might be different, but related. In case of transfer learning three major research issues are: 1) what to transfer, 2) how to transfer and 3) what to transfer [113]. In both transfer learning and domain adaptation problems it is imperative to have thorough knowledge of source/target domains and of the tasks. This knowledge is crucial to develop the learning problem which can help devise systems that operate well during prediction phase. While it might be possible to model non-stationary environments in certain scenarios [185], evolving data can also lead to negative transfer and thereby affect system performance. In case of multi-category systems, especially when the categories are related, automatic transfer learning is extremely difficult. In an environment where object categories are continuously added and removed (e.g. a face recognition system with continuous enrolling of new identities) existing adaptation methods face major challenge. Recent work of Kuzborskij et. al [80] demonstrate the challenges faced in the domain of object detection for multi-class transfer learning.

The field of online/incremental learning takes into account the notion of learning [21], [17] in the presence of changing data by continuously updating the stored model. This can be done by adding one sample at a time or updating models from small batches. The general assumption in case of these problems is the data is drawn from a fixed distribution. This assumption often leads to sample selection bias. Posterior probabilities are often used to develop confidence of the system conditioned on training data $P(x_{new}|y, \Theta)$. However, these methods were developed mostly in the context of batch learning and specific study from the point of view of online/incremental learning in missing. Another related problem is modeling changing properties of data over time. In the filed of predictive analytics and machine learning, this problem is referred to as concept drift. The core assumption when dealing with the concept drift problem is uncertainty about the future [50]. While it is possible to develop methods that can handle drifts in data over time, most of the existing methods can handle a specific kind of drift. In real-world scenarios, it is not just important to understand the nature of drift, but to be also aware of associated risk when classifying under such circumstances. Given the variability occurring in the real-world, it is impossible to devise one-size-fits all approach to the problem of recognition. In the past, various aspects of recognition systems working in operational environments have been considered in isolation. The problem of evolving data has been considered in the domain of concept drift [50], the problem of evolving tasks and changes in source and operational domains from learning problem's perspective have been considered in the areas of transfer learning, covariate shift and domain adaptation [61], [113], [122]. The problem of meta-recognition and considering the problem from user's perspective has been seen in the works of various post-recognition analysis methods [136], [169], [1].

A question to ask at this stage is why a systems perspective is critical to consider the problem of recognition under change. As noted by Scheirer et. al. [141], in classification, one assumes there is a given set of classes between which we must discriminate. For recognition, we assume there are some classes we can recognize in a much larger space of things we do not recognize. Methods developed for classifying in evolving domains in the area of domain adaptation, concept drift , incremental learning or covariate shift almost explicitly consider the classification viewpoint of recognition. One thing is certain though, study of evolving recognition systems i.e. evolving nature of data to be processed, evolving nature of learning problem to be considered and evolving nature of desired outcome from system by the user cannot be done in isolation of one another. *Moving forward, it is imperative to consider a systems approach that can consider challenges seen by various parts of a recognition system in isolation, as a whole and inter-play of these issues.*

The goal of this thesis is not to present a unified framework that claims to solve the problem of recognition. It will be very ambitious and very pre-mature. Rather, the goal of the thesis is to identify and address issues faced by a recognition system, develop understanding of these issues and propose solution. This thesis can be viewed as a step towards building robust recognition systems that are suited to operate in ever changing operational scenarios.

We first discuss the general notion of recognition system in detail. This is followed by a discussion on identification of some specific challenges faced by current day recognition systems. Finally we discuss our contributions related to these challenges. Throughout the course of the thesis we maintain a focus on image based recognition systems. Major part of the thesis is dedicated to the problem of face recognition. In the latter part of the thesis we show application of our method to the more general problem of object recognition.

1.1.2 What is Recognition?

Why is the problem of recognition so hard? The world is made of clutter of objects which vary in scale, position, orientation leading to significantly different manifestation of same 3D object when projected on a 2D surface. Furthermore, the variability intrinsic within an object category (e.g. faces, dogs etc.), due to complex non-rigid articulation and extreme variations in shape and appearance makes it impossible that one could simply perform exhaustive matching against a database of all the representative exemplars [159].

Recognition in context of computer vision is commonly referred to understanding and inferring the objects presented in any visual representation (image, video etc.). Recognition is commonly defined as submitting an unknown "object", say o_t to an algorithm which will compare the object to a known set of classes, thus producing a similarity measure to each. For any recognition system, maximizing the performance of recognition is a primary goal [136]. Shakhnarovich et. al. [145] define the task of a recognition system in statistical framework is to find a class label c^*, where p_k is the underlying probability rule and p_0 is the input distribution satisfying

$$c^* = Pr(p_0 = p_c) \tag{1.1}$$

In the above definition, it is assumed that an object representation o_t is given to a recognition system, which in turn provides a decision c^* to the user. Though this might be an important element in the recognition system, it is far from complete. The recognition system can be broken down across multiple axes: each playing a supplementary or complementary role depending on the system design. A recognition system usually consists of a image pre-processing module, object detection module, object recognition module and post-processing module. Image pre-processing usually consists of normalization for illumination variation or sub-sampling if dealing with video. This often serves as an input to detection and tracking system.

Detection (whether it is for faces or more general object categories) is the problem of segmenting the object of interest from background. Detection is a necessary step for most face recognition application and its reliability has a major influence on performance and usability of the entire recognition pipeline. The problem of detection is extremely hard. An ideal detector should be able to locate a face or an object irrespective of its position, scale, orientation, age and expression (in case of faces). Tracking based systems often work in conjunction with a detection module. The problem of detection has received significant attention in last couple of decades. [64], [132], [89]. The problem of detection is an important step in automated face recognition systems. Though an important operational challenge, in this work we do not explore the problem of detection in detail. The problem of detection by itself is an extremely important issue and has received significant attention from computer vision community. The problem of face detection has been solved to meet the "minimum" requirements of most practical applications [89]. Addressing the problem of detection in this thesis, would have made the scope of the thesis very ambitious and impractical. For majority of the thesis, we focus mainly on the classification aspect of recognition system from a systems perspective.

The object recognition module usually consists of feature extraction, feature matching and classification/learning module. After successful detection, the image is transformed into a feature space. Following this transformation a learning methodology called classfication is adopted to separate the data into multiple categories using a training set. The process of discriminating set of observation into multiple categories on the basis of predefined rules is called *classification*. In the terminology of machine learning, classification is considered an instance of supervised learning, i.e. learning where a training set of correctly identified observations is available. The corresponding unsupervised procedure is known as clustering.

Lets try to understand in more formal terms as to why the problem of classification is difficult. The fundamental problem in statistical learning seeks to find a classification function f that minimizes the ideal risk $\mathcal{R}_{\mathcal{I}}$

$$\underset{f}{\text{argmin}} \left\{ \mathcal{R}_{\mathcal{I}}(f) := \int_{\mathbb{R}^d x \mathbb{N}} \phi(x, y, f(x)) P(x, y) \right\} \tag{1.2}$$

$\mathcal{R}_{\mathcal{I}}$ is composed of two terms, the joint distribution $P(x, y)$ of the data x and labels y, and the loss function $\phi(x, y, f(x))$, which assigns the cost of misclassification. Since the joint distribution $P(x, y)$ is unknown during training time completely, the problem is unsolvable in the fundamental formulation. The traditional approach at this point is to change the problem to use only things we do know. As Smola et. al. notes"The only way out is to approximate $P(x, y)$ by the empirical probability density function.." [152]. In chapter 2, we present a framework to extend the definition of recognition function (1.2) to incorporate the broader notion of **open world recognition**.

1.1.3 Open World Recognition

With the of advent rich classification models and high computational power visual recognition systems have found many operational applications. Over the past decade, datasets for building and evaluating visual recognition systems have increased both in size and variation. The size of datasets has increased from a few hundred images to millions of images, and the number of categories within the datasets has increased from tens of categories to more than a thousand categories. Co-evolution of rich classification models along with advances in datasets have resulted in many commercial applications. Recognition in the real world poses multiple challenges that are not apparent in controlled lab environments. A multitude of operational challenges are posed while porting recognition systems from controlled lab environments to the real world. A recognition system in the ?open world? has to continuously update with additional object categories and be robust to unseen categories and have minimum downtime. Despite the obvious dynamic and open nature of the world, a vast majority of recognition systems assume a static and closed world model of the problem where all categories are known a priori. To address these operational issues, we formalize and present steps towards the problem of open world recognition.

1.2 Contributions of the Thesis

The contribution of this work can be summarized as follows:

1. **Towards Open World Recognition:** In chapter 2 we formalize the problem of open world recognition. A recognition system in the "open world" has to continuously update with additional object categories, be robust to unseen categories and have minimum downtime. Despite the obvious dynamic and open nature of the world, a vast majority of recognition systems assume a static and closed world model of the problem where all categories are known apriori. To address these operational issues, we formalize and presents steps towards the problem of Open World Recognition.

2. **Learning with Streaming Data:** In chapter 4 we address the issue of recognition in streaming environment. Recognition in streaming environment poses multiple challenges : resources are limited, obtaining ground truth is expensive and data distribution is continuously evolving and demands for alternative learning strategies such as incremental/online learning. Increasing access to large, non-stationary face datasets and corresponding demands to process , analyze and learn from this data has lead to a new class of online/incremental face recognition problems. While it is advantageous to build large scale learning systems when resources permit, a counter problem of learning with limited resources in presence of streaming data arises. We present a budgeted incremental support vector learning method suitable for online learning applications. Our system can process one sample at a time and is suitable when dealing with large streams of data.

3. **Incremental Model Adaptaion:** In chapter 5 we address the issue of incremental model adaptation for enrolled client identities. We develop tools and techniques with specific focus on face recognition applications in the wild. We compare two popular classification methods, generative methods (Gaussian Mixture Models) and discriminative methods (Support Vector Machines) in the context of incremental learning. We first develop tools and techniques for incremental support vector learning and incremental gaussian mixture modeling. We then present a protocol suited for large scale incremental learning for

face recognition. Finally we discuss strategies for adapting learned models in the presence of limited labelled data: semi-supervised incremental model adaptation.

4. **Operational Adaptation for Visual Recognition:** In chapter 6 we formalize the problem of operational adaptation and propose a novel solution for recognition with partial data. We consider a constrained environment for recognition, wherein only trained model is available during test time with access to no additional data. Further, we develop the concept of adaptation risk and show that the proposed risk estimation is better predictor of system performance.

In the following chapters, the above mentioned problems and proposed solutions are discussed in detail[1].

1.2.1 Publications

This thesis has resulted in following publications (current and in progress)

1. A. Bendale, T. Boult “Reliable Posterior Probability Estimation for Streaming Face Recognition” Biometrics Workshop, CVPR 2014 (28.81% acceptance rate, 4)
2. A. Bendale, T. Boult “Towards Open World Recognition” CVPR 2015 (oral, 3.3 % acceptance rate, 2)
3. A. Bendale, T. Boult “Operational Adaptation for Recognition: A case study on Missing Data” (in review, 6)
4. A. Bendale, M. Güenther, T. Boult, S. Marcel “Incremental Model Adaptation for Face Recognition” IEEE Trans. on Information Forensics and Security (in preparation, 5)

1.2.2 Acknowledgements

This work was carried out with the generous support of following grants:

1. UCCS Graduate School Fellowship
2. ONR MURI N00014-08-1-0638 on Maritime Biometrics.

[1] Each chapter mentioned in this thesis is presented as a “stand alone” techniques. The symbols and notations are specific to each chapter, unless specified otherwise.

3. NSF Research Grant IIS-1320956 on Open Vision - Tools for Open Set Computer Vision and Learning.
4. European project Biometrics Evaluation and Testing (BEAT) under the European Commission 7th Framework Programme (FP7)

Chapter 2

Towards Open World Recognition

Over the past decade, datasets for building and evaluating visual recognition systems have increased both in size and variation. The size of datasets has increased from a few hundred images to millions of images, and the number of categories within the datasets has increased from tens of categories to more than a thousand categories. Co-evolution of rich classification models along with advances in datasets have resulted in many commercial applications [35, 160, 111]. A multitude of operational challenges are posed while porting recognition systems from controlled lab environments to the real world. A recognition system in the "open world" has to continuously update with additional object categories and be robust to unseen categories and have minimum downtime. Despite the obvious dynamic and open nature of the world, a vast majority of recognition systems assume a *static* and closed world model of the problem where all categories are known a priori. To address these operational issues, this paper formalizes and presents steps towards the problem of open world recognition. The key steps of the problem are summarized in Fig. 2.1.

As noted by [141], "when a recognition system is trained and is operational, there are finite set of known objects in scenes with myriad unknown objects, combinations and configurations – labeling something new, novel or unknown should always be a valid outcome." One reason for the domination of "closed world" assumption of today's vision systems is that matching, learning and classification tools have been formalized as selecting the most likely class from a closed set. Recent research [141, 142, 63], has re-formalized learning for recognition as open set recognition. However, this approach does not explicitly require that inputs be as known

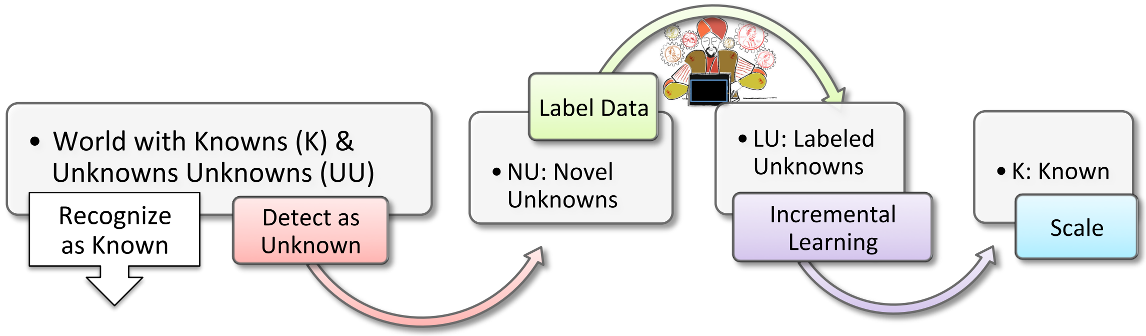


Figure 2.1: In open world recognition, the system must be able to recognize objects and associate them with known classes while also being able to label classes as unknown. These “novel unknowns” must then be collected and labeled (e.g. by humans). When there are sufficient labeled unknowns for new class learning, the system must incrementally learn and extend the multi-class classifier, thereby making each new class “known” to the system. Open World recognition moves beyond just being robust to unknown classes and toward a scalable system that is adapting itself and learning in an open world.

or unknown. In contrast, for open world recognition, we propose the system explicitly label novel inputs as unknown and then incrementally incorporate them into the classifier. Furthermore, open set recognition as formulated by [141] is designed for traditional one-vs-all batch learning scenario. Thus, it is open set but not incremental and does not scale gracefully with the number of categories.

While there is a significant body of work on incremental learning algorithms that handle new instances of known classes [21, 29, 178], open world requires two more general and difficult steps: continuously detecting novel classes and when novel inputs are found updating the system to include these new classes in its multi-class open set recognition algorithm. Novelty detection and outlier detection are complex issues in their own right with long histories [97, 58] and they are still active vision research topics [15, 96]. After detecting a novel class, the requirement to add new classes leaves the system designer with the choice of re-training the entire system. When the number of categories are small, such a solution may be feasible, but unfortunately, it does not scale. Recent studies on ImageNet dataset using SVMs or CNN require days to train their system [117, 75], e.g. 5-6 CPU/GPU days in case of CNN for 1000 category image classification task. Distance based classifiers like Nearest Class Mean (NCM) [71, 103, 124] offer a natural choice for building scalable system that can learn new classes incrementally. In NCM-like classifiers, incorporating new images or classes in implies adjusting the existing means or updating the set of class means. However, NCM classifier in its current formulation is not suited for open set recognition because it uses close-set assumptions for probability normalization. Handling unknowns in open world recognition requires gradual decrease in the value of probability (of class membership) as the test point moves away from known data into open space. The softmax based probability assignment used in NCM does not account for open space.

The *first contribution* of this paper is a formal definition of the problem of open world recognition, which extends the existing definition of open set recognition which was defined for a static notion of set. In order to solve open world recognition, the system needs to be robust to unknown classes, but also be able to move through the stages and knowledge progression summarized in Fig. 2.1. *Second contribution* of the work is a recognition system that can continuously learn new object categories in an open world model. In particular, we show how to extend Nearest Class Mean type algorithms (NCM) [103], [124], to a Nearest Non-Outlier (NNO) algorithm that can balance open space risk and accuracy.

To support this extension, our *third contribution* is showing that thresholding sums of monotonically decreasing functions of distances of linearly transformed feature space can have arbitrarily small "open space risk". Finally, we present a protocol for evaluation for open world recognition, and use this protocol to show our NNO algorithm perform significantly better on open world recognition evaluation using Image-Net [9].

2.1 Related Work

Our work addresses an issue that is related to and has received attention from various communities such as incremental learning, scalable learning and open set learning.

Incremental Learning: As SVMs rose to prominence in for object recognition applications [182, 92], many incremental extensions to SVMs were proposed. Cauwenberghs *et al.* [21] proposed an incremental binary SVM by means of saving and updating KKT conditions. Yeh *et al.* [178] extended the approach to object recognition and demonstrated multi-class incremental learning. Pronobis [121] proposed a memory-controlled online incremental SVM for visual place recognition. Although incremental SVMs might seem natural for large scale incremental learning for object recognition, they suffer from multiple drawbacks. The update process is extremely expensive (quadratic in the number of training examples learned [83]) and depends heavily on the number of support vectors stored for performing updates [83]. To overcome the update expense, Crammer *et al.* [29] and Shalev-Shwartz *et al.* [146] proposed classifiers with fast and inexpensive update process along with their multi-class extensions. However, the multi-class incremental learning methods and other incremental classifiers [29, 146, 171, 88] are incremental in terms of additional training samples but not additional training categories..

Scalable Learning: Researchers like [99, 94, 38] have proposed label tree based classification methods to address scalability (# of object categories) in large scale visual recognition challenges [42, 9]. Recent advances in deep learning community [75, 148], has resulted in state of the art performance on these challenges. Such methods are extremely useful when the goal is to obtain maximum classification/recognition performance. These systems assume a priori availability of entire training data (images and categories). However, adapting such methods to a dynamic learning scenario becomes extremely challenging. Adding object categories requires retraining the entire system, which could be infeasible for many applications. Thus, these methods are scalable but not incremental (Fig 4.1)

Open Set Learning: Open set recognition assumes there is incomplete knowledge of the world at training time, and unknown classes can be submitted to an algorithm during testing [87, 141]. Scheirer *et al.* [141] formulated the problem of open set recognition for static one-vs-all learning scenario by balancing open space risk while minimizing empirical error. Scheirer *et al.* [142, 63] extended the work to multi-class settings by introducing compact abating probability model. Their work offers insights into building robust methods to handle unseen categories. However, class specific Weibull based calibration of SVM decision scores does not scale. Fragoso *et al.* [47] proposed a scalable Weibull based calibration for hypothesis generation for modeling matching scores, but they do not address it in the context of general recognition problem.

The final aspect of related work is nearest class mean (NCM) classifiers. NCM classification, in which samples undergo a Mahalanobis transform and then are are associated with a class/cluster mean, is a classic pattern recognition approach [49]. NCM classifiers have a long history of use in vision systems [30] and have multiple extensions, adaptations and applications [34, 151, 177, 79, 95]. Recently the technique has been adapted for use in larger scale vision problems [166, 165, 103, 124], with the most recent and most accurate approaches combining NCM with metric learning [103] and with random forests[124].

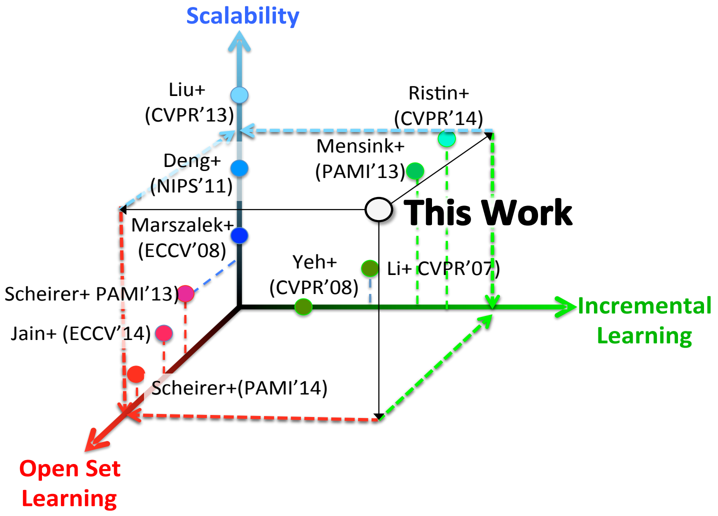


Figure 2.2: Putting the current work in context by depicting locations of prior work with respect to three axes of the major issues for open world recognition: open set learning, incremental learning and scalability. In this work, we present a system that is scalable, can handle open set recognition and can learn new categories incrementally without having to retrain the system every time a new category arrives. The works depicted include Ristin *et al.* [124], Mensink *et al.* [103], Scheirer *et al.* . [141], [142], Jain *et al.* . [63], Yeh *et al.* , [178], Marszalek *et al.* . [99], Liu *et al.* [94], Deng *et al.* [38], and Li *et al.* [88]. This papers advances the state of the art in open set learning and incremental learning while providing reasonable scalability.

Since we extend NCM classification, we briefly review the formulation including a probabilistic interpretation. Consider an image represented by a d-dimensional feature vector $x \in \mathbb{R}^d$. Consider $\mathcal{K}$ object categories with their corresponding centroids μ_k, where $k \in \mathcal{K}$. Let $\mathcal{I}_k$ be images for each object category. The centroid is given by $\mu_k = \frac{1}{|\mathcal{I}_k|} \sum_{i \in \mathcal{I}_k} x_i$. NCM classification of a given image instance I with a feature vector x is formulated as searching for the closest centroid in feature space as $c^* = \underset{k \in \mathcal{K}}{\operatorname{argmin}} \ \mathbf{d}(x, \mu_k)$. Here $\mathbf{d}(.)$ represents a distance operator usually in Euclidean space. Mensink *et al.* [103] replace Euclidean distance with a low-rank Mahalanobis distance optimized on training data. The Mahalanobis distance is induced by a weight matrix $W \in \mathbb{R}^{d \times D}$, where D is the dimensionality of the lower dimensional space. Class conditional probabilities $p(c|x)$ using an NCM classifier are obtained using a probabilistic model based on multi-class logistic regression as follows:

$$p(c|x) = \frac{exp(-\frac{1}{2}\mathbf{d}_W(x,\mu_k))}{\sum_{k'=1}^{\mathcal{K}} exp(-\frac{1}{2}\mathbf{d}_W(x,\mu_{k'}))} \tag{2.1}$$

In the above formulation, class probabilities $p(c)$ are set to be uniform over all classes. During metric learning optimization, Mensink *et al.* [103] considered non-uniform probabilities given by:

$$p'(c|x) = \frac{1}{Z} exp(x^T W^T W \mu_c + s_c) \tag{2.2}$$

where Z denotes the normalizer and s_c is a per class bias.

2.2 Open World Recognition

We first establish preliminaries related to open world recognition, following which we formally define the problem. Let classes be labeled by positive integers $\mathbb{N}^+$ and let $\mathcal{K}_t \subset \mathbb{N}^+$ be the set of labels of known classes at time t. Let the zero label (0) be reserved for (temporarily) labeling data as unknown. Thus $\mathbb{N}$ includes unknown and known labels.

Let our features be $x \in \mathbb{R}^d$. Let $f \in \mathcal{H}$ be a measurable recognition function, i.e. $f_y(x) > 0$ implies recognition of the class y of interest and $f_y(x) \leq 0$ when y is not recognized, where $\mathcal{H} : \mathbb{R}^d \mapsto \mathbb{R}$ is a suitably smooth space of recognition functions.

The objective function of open set recognition, including multi-class formulations, must balance open space risk against empirical error. As a preliminary we adapt the definition of open space and open space risk used in [141]. Let open space, the space sufficiently far from any known positive training sample $x_i \in \mathcal{K}, i = 1 \ldots N$, be defined as:

$$\mathcal{O} = S_o - \bigcup_{i \in N} B_r(x_i) \tag{2.3}$$

where $B_r(x_i)$ is a closed ball of radius r centered around any training sample x_i. Let S_o be a ball of radius r_o that includes all known positive training examples $x \in \mathcal{K}$ as well as the open space $\mathcal{O}$. Then probabilistic *Open Space Risk* $R_{\mathcal{O}}(f)$ for a class y can be defined as

$$R_{\mathcal{O}}(f_y) = \frac{\int_{\mathcal{O}} f_y(x)dx}{\int_{S_o} f_y(x)dx} \tag{2.4}$$

That is, the open space risk is considered to be the relative measure of positively labeled open space compared to the overall measure of positively labeled space.

Given an empirical risk function $R_{\mathcal{E}}$, e.g. hinge loss, the objective of *open set recognition* is to find a measurable recognition function that manages (minimizes) the **Open Set Risk**:

$$\underset{f \in \mathcal{H}}{\text{argmin}} \left\{ R_{\mathcal{O}}(f) + \lambda_r R_{\mathcal{E}}(f) \right\} \tag{2.5}$$

where λ_r is a regularization constant.

With the background in place, we formalize the problem of open world recognition.

Definition 1 (Open World Recognition)**.** *A solution to open world recognition is a tuple* $[F, \varphi, \nu, L, I]$ *with:*

1. *A* **multi-class open set recognition function** $F(x) : \mathbb{R}^d \mapsto \mathbb{N}$ *using a vector function* $\varphi(x)$ *of* i *per-class measurable recognition functions* $f_i(x)$*, also using a* **novelty detector** $\nu(\varphi) : \mathbb{R}^i \mapsto [0, 1]$*. We require the per class recognition functions* $f_i(x) \in \mathcal{H} : \mathbb{R}^d \mapsto \mathbb{R}$ *for* $i \in \mathcal{K}_t$ *to be open set recognition functions that manage open space risk as Eq.2.4. The novelty detector* $\nu(\varphi) : \mathbb{R}^i \mapsto [0, 1]$ *determines if results from vector of recognition functions is from an unknown (*0*) class.*

2. *A labeling process* $L(x) : \mathbb{R}^d \mapsto \mathbb{N}^+$ *applied to novel unknown data* U_t *from time* t*, yielding labeled data* $D_t = \{(y_j, x_j)\}$ *where* $y_j = L(x_j) \forall x_j \in U_t$*. Assume the labeling finds* m *new classes, then the set of known classes becomes* $\mathcal{K}_{t+1} = \mathcal{K}_t \cup \{i + 1, \ldots i + m\}$*.*

3. *An incremental learning function* $I_t(\varphi; D_t) : \mathcal{H}^i \mapsto \mathcal{H}^{i+m}$ *to scalably learn and add new measurable functions* $f_{i+1}(x) \ldots f_{i+m}(x)$*, each of which manages open space risk, to the vector* φ *of measurable recognition functions.*

Ideally, all of these steps should be automated, but herein we presume supervised learning with labels obtained by human labelling.

If we presume that each $f_k(x)$ reports a likelihood of being in class k and $f_k(x)$ is normalized across the respective classes. Let $\varphi = [f_1(x), \ldots, f_k(x)]$. For this paper we let the multi-class open set recognition function be given as

$$y^* = \underset{y \in \mathcal{K}, f_y(x) \in \varphi(x)}{\text{argmax}} f_y(x), \tag{2.6}$$

$$F(x) = \begin{cases} 0 & \text{if } \nu(\varphi(x)) = 0 \\ y^* & otherwise \end{cases} \tag{2.7}$$

With these definitions, a simple approach for the novelty detection is to set a minimum threshold τ for acceptance, e.g. letting $\nu(\varphi(x)) = f_{y^*}(x) > \tau$. In the following section we will prove this simple approach can manage open space risk and hence provide for item 1 in the open world recognition definition.

2.3 Opening existing algorithms

The series of papers [141, 142, 63] formalized the open set recognition problem and proposed 3 different algorithms for managing open set risk. It is natural to consider these algorithms for open world recognition. Unfortunately, these algorithms use EVT-based calibration of 1-vs-rest RBF SVMs and hence are not well suited for incremental updates or scalability required for open world recognition. In this paper we pursue an alternative approach better suited to open world using non-negative combinations of abating distance. Using this we develop the Nearest Non-Outlier (NNO) algorithm to inexpensively extend NCM for open world recognition.

The authors of [142] show if a recognition function is decreasing away from the training data, a property they call abating, then thresholding the abating function limits the labeled region and hence can manage/limit open space risk. The Compact Abating Probability (CAP) model presented in that paper is a sufficient model, but it is not necessary. In particular we build on the concept of a CAP model but generalize the model showing that any non-negative combination of abating functions, e.g., a convex combination of decreasing functions of distance, can be thresholded to have zero open space risk. We further show we can work in linearly transformed spaces, including projection onto subspaces, and still manage open space risk and NCM type algorithms manage open space risk.

Theorem 1 (**Open space risk for model combinations).** *Let $M_{\tau,y}(x)$ be a recognition function that thresholds a non-negative weighted sum of η CAP models ($M_{\tau,y}(x) = \sum_{j=1}^{\eta} c_j M_{j,\tau_j,y}(x)$) over a known training set for class y, where $1 \geq c_j \geq 0$ and $M_{j,\tau,y}(x)$ is a CAP model. Then for $\delta \geq 0$ $\exists \tau^*$ s.t. $R_{\mathcal{O}}(M_{\tau^*,y}) \leq \delta$, i.e. one can threshold the probabilities $M_{\tau,y}(x)$ to limit open space risk to any level.*

Proof: It is sufficient to show it holds for $\delta = 0$, since similar to Corr. 1 of [142], larger values of δ allow larger labeled regions with larger open space risk. Considering each model $M_{j,\tau_j,y}(x) j = 1..\eta$ separately, we can apply Theorem 1 of [142] to each $M_{j,\tau_j,y}(x)$ yielding a τ_j such that the function $M_{j,\tau_j,y}(x) > 0$ defines a labeled region $l_j(\tau_j) \subset X$ with zero open space risk. Letting $\tau^* = \min_j \tau_j$ it follows that $M_{\tau^*,y}(x) > 0$ is contained within $\cup_j l_j(\tau^*)$, which as a finite union of compact regions with zero risk, is itself a compact labeled region with zero open space risk. *Q.E.D*

The theorem/proof trivially holds for a max over classes but can be generalized to combinations via product or to combinations of monotonic transformed recognition functions, with appropriate choice of thresholds. For this paper we need max over models using data from metric learned tranformed features, i.e. lower-dimensional projected spaces.

Theorem 2 (Open Space Risk for Transformed Spaces). *Given a linear transform $T : \mathbb{R}^n \to \mathbb{R}^m$ let $x' = T(x), \forall x \in X$, yields X' a linearly transformed space of features derived from feature space $X \subset \mathbb{R}^n$. Let $\mathcal{O}' = \cup_{x \in \mathcal{O}} T(x)$ be the transformation of points in open space $\mathcal{O}$. Let $M'_{\tau,y}(x')$ be a probabilistic CAP recognition function over $x' \in X'$ and let $M_{\tau,y}(x) = M'_{\tau,y}(Tx)$ be a recognition function over $x \in X$. Then $\exists \epsilon : R_{\mathcal{O}'}(M'_{\tau',y}) \leq \delta \implies R_{\mathcal{O}}(M_{\tau,y}) < \epsilon\delta$, i.e. managing open set risk in X' will also manage it in the original feature space X.*

Proof: If T is dimensionality preserving, then the theorem follows from the linearity of integrals in the definition of risk. Thus we presume T is projecting away $n - m$ dimensions. Since the open space risk in the projected space is δ we have $\lambda_m(M'_{\tau',y} \cap \mathcal{O}') = c\delta$ where λ_m is the Lebesgue measure in $\mathbb{R}^m$ and $c < \infty$. Since $\mathcal{O} \subset S_o$, i.e. $\mathcal{O}$ is contained within a ball of radius r_o, it follows from the properties of Lebesgue measure that $\lambda_n(M_{\tau,y} \cap \mathcal{O}) \leq \lambda_m \left(M'_{\tau',y} \cap (\mathcal{O}' \times [-r_o, r_o]^{n-m})\right) = c * \delta * (2r_o)^{n-m} = 0$ and hence the open space risk in $\mathbb{R}^m$ is bounded. *Q.E.D.*

It is desirable for open world problems that we consider the error in the original space. We note that ϵ varies with dimension and the above bounds are generally not tight. While the theorem gives a clean bound for zero open space risk, for a solution with non-zero δ risk in the lower dimensional space, when considered in the original space, the solution may have open space risk that increases exponentially with the number of missing dimensions.

We note that these theorems are not a license to claim that any algorithms with rejection manage open space risk. While many algorithms can be adapted to compute a probability estimate of per class inclusion and can threshold those probabilities to reject, not all such algorithms/rejections manage open space risk. Thresholding Eq 2.2, which [103] minimizes in place of 2.1, will not manage risk because the function does not always decay away from known data. Similarly, rejecting a decision close to the plane in a linear SVM does not manage open space risk, nor does the thresholding layers in a convolution neural network [144].

On the positive side, these theorems show that one can adapt algorithms that linearly transforms feature space and use a *probability/score mapping that combines positive scores that decrease with distance from a finite set of known samples*. In the following section, we demonstrate how to generalize an existing algorithm while managing open space risk. Open world performance, however, greatly depends on the underlying algorithm and the rejection threshold. While theorems 1 and 2 say there exists a threshold with zero open space risk, at that threshold there may be minimal or no generalization ability.

2.3.1 Nearest Non-Outlier (NNO)

As discussed previously one of the significant contributions of this paper is combining theorems 1 and 2 to provide an example of open space risk management and move toward a solution to open world recognition. Before moving on to defining open world NCM, we want to add a word of caution about "probability normalization" that presumes all classes are known. e.g. softmax type normalization used in eqn 2.1. Such normalization is problematic for open world recognition where there are unknown classes. In particular, **in open world recognition the Law of Total Probability and Bayes' Law cannot be directly applied** and hence cannot be used to normalize scores. Furthermore, as one adds new classes, the normalization factors and hence probabilities, keep changing and thereby limiting interpretation of the probability. For an NCM type algorithm, normalization with the softmax makes thresholding very difficult since for points far from the class means the nearest mean will have a probability near 1. Since it does not decay, it does not follow Theorem 1.

To adapt NCM for open world recognition, we introduce Nearest Non-Outlier (NNO) which uses a measurable recognition function consistent with Theorems 1 and 2. Let NNO represent its internal model as a vector of means $\mathcal{M} = [\mu_1, \ldots \mu_k]$. Let $W \in \mathbb{R}^{d \times m}$ be the linear transformation dimensional reduction weight matrix learned by the process described in [103]. Then given τ, let

$$\hat{f}_i(x) = \frac{\Gamma(\frac{m}{2}+1)}{\pi^{\frac{m}{2}}\tau^m}(1 - \frac{1}{\tau}\|W^\top x - W^\top \mu_i\|) \tag{2.8}$$

be our measurable recognition function with $\hat{f}_i(x) > 0$ giving the probability of being in class i, where Γ is the standard gamma function which occurs in the volume of a m-dimensional ball. Intuitively, the probability is a tent-like function in the sphere and the first fraction in eqn 2.8 comes from volume of m-sphere and ensures that the probability integrates to 1.

Let $\hat{\varphi} = [\hat{f}_1(x), \ldots, \hat{f}_k(x)]$ with y^* and $F(x)$ given by Eq. 2.7. Let with $\hat{\nu}(\hat{\varphi}(x)) = \hat{f}_{y^*}(x) > 0$. That is, NNO rejects x as an outlier for class i when $\hat{f}_i(x) = 0$, and NNO labels input x as unknown/novel when all classes reject the input.

Finally, after collecting novel inputs, let D_t be the human labeled data for a new class $k + 1$ and let our incremental class learning $I_t(\hat{\varphi}; D_t)$ compute $\mu_{k+1} = mean(D_t)$ and append μ_{k+1} to $\mathcal{M}$.

Corollary 1 (NNO solves open world recognition)**.** *The NNO algorithm with human labeling $L(x)$ of unknown inputs is a tuple $[F(x), \hat{\varphi}, \hat{\nu}(\hat{\varphi}(x)), L, I_t(\hat{\varphi}; D_t)]$, consistent with Definition 1, hence NNO is a open world recognition algorithm.*

By construction theorems 1 and 2 apply to the measurable recognition functions $F(x)$ from Eq. 2.7 when using a vector of per classes functions given eq. 2.8. By inspection the NNO definitions of $\hat{\nu}(\hat{\varphi}(x))$ and $I_t(\hat{\varphi}; D_t)$ are consistent with Definition 1 and are scalable. *Q.E.D.*

2.4 Experiments

In this section we present our protocol for open world experimental evaluation of NNO, and a comparison-withmultiple baseline classifiers including NCM, a liblinear SVM [43] and our liblinear version of the 1vSet algorithm of [141][1].

Dataset and Features: Our evaluation is based on the ImageNet Large Scale Visual Recognition Competition 2010 dataset. ImageNet 2010 dataset is a large scale dataset with images from 1K visual categories. The dataset contains 1.2M images for training (with around 660 to 3047 images per class), 50K images for validation and 150K images for testing. The large number of visual categories allow us to effectively gauge the performance of incremental and open world learning scenarios. In order to effectively conduct experiments

[1] Code and data partitions for experiments can be found at `http://vast.uccs.edu/OpenWorld`

using open set protocol, we need access to ground truth. ILSVRC'10 is the only ImageNet dataset with full ground truth, which is why we selected that dataset over later releases of ILSVRC (e.g. 2011-2014).

We used densely sampled SIFT features clustered into 1K visual words as given by Berg *et al.* [9]. Though more advanced features are available [117, 75, 149], extensive evaluation across features is beyond the scope of this work [2]. Each feature is whitened by its mean and standard deviation to avoid numerical instabilities. We report performance in terms of average classification accuracy obtained using top-1 accuracy as per the protocol provided for the ILSVRC'10 challenge. As our work involves initially training a system with small set of visual categories and incrementally adding additional categories, we shun top-5 accuracy.

Algorithms: The proposed Nearest Non-Outlier (NNO) extension of NCM classifier is compared with the baseline NCM algorithm in both incremental and open world settings. We use the code provided by Mensink *et al.* [103] as the NCM baseline. This algorithm has near state of the art results and while recent extension with random forests[124] improved accuracy slightly, [124] does not provide code. While not incremental, we also include a comparison with the state of the art open set algorithm by extending liblinear to provide a 1vSet SVM [141]. Details about our extension can be found in the supplemental material.

2.4.1 Open World Evaluation Protocol

Closed set evaluation is when a system is tested with allclassesknown during training and testingi.e. training, and testing use the same classes but different instances. In open set evaluation,training uses known classes and testing uses both known and unknownclasses.The open set recognition evaluation protocol proposed by Scheirer *et al.* [141] does not handle the open world scenario in which object categories are beingincrementallyadded to the system. Ristin *et al.* [124] presented an incremental closed set learning scenario where novel object categories are added continuously. We combined ideas from both of these approaches and propose a protocol that is suited for open world recognition in which categories are being added to the system continuously while the system is also tested with unknown categories.

Training Phase: The training of the NCM classifier is divided into two phases: an initial metric learning/training phase and a growth/incremental learning phase. In the metric learning phase, a set of object

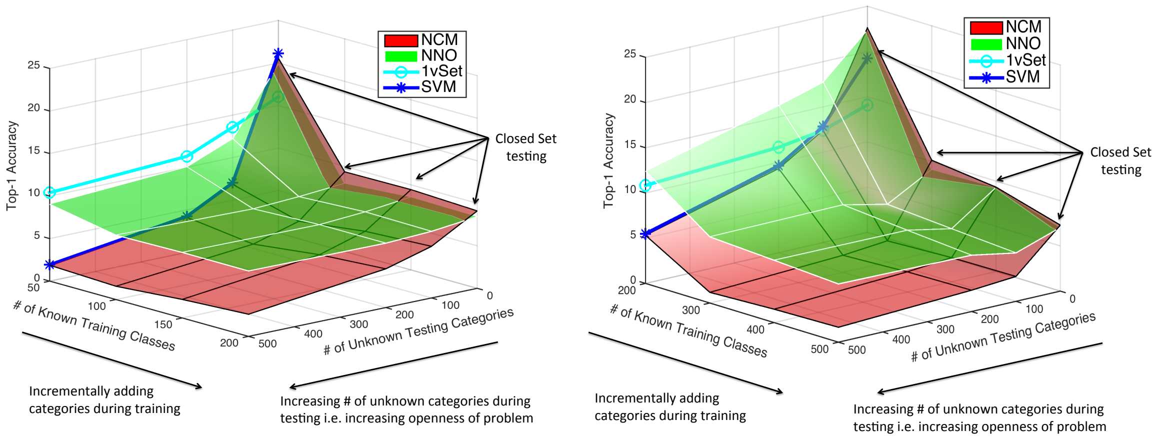
[2] The supplemental material presents experiments with additional ILSVRC'13 features, showing the gains of NNO are not feature dependent

categories are provided to the system uses iterative metric learning on these categories. Once the metric learning phase is completed, the incremental learning phase uses the fixed metrics and parameters. During the incremental learning phase, object categories are added to the system one-by-one. While for scalability one might measure time, both NCM and NNO add new categories in the same way, and it is extremely fast since it only consists of computing the means. Thus, so we do not measure/report timing here.

Nearest Non-Outlier (NNO) is based on the CAP model and requires estimation of τ for eq. 2.8. To estimate τ, during theparameter estimation phase using the metric learned in that phase,we use a 3-fold cross-class validation [63] wherein ech fold dividesthetrainingdatainto two sets: training categories and validation categories. The τ for NNO is estimated with 3-fold cross-class validation optimizing for F1-measure over values for which there is at least 90% recall in a fold, yielding a value of 5000 – see the supplemental material for more details. An important point to note about estimating τ is that one has to balance the classification errors between known set of categories along with the errors between known and unknown set of categories. One could obtain high accuracy when testing with large number of samples from unknown categories by rejecting everything, but this compromises accuracy on the known set of categories. Hence our requirement of high recall rate and optimization over F1-measure rather than accuracy.

Testing Phase: To ensure proper open world evaluation, we do cross-class folds that split the ImageNet test data into two sets of 500 categories each: the known set and the unknown set. At every stage, the system is evaluated with a subset of the known set and the unknown set to obtain closed set and open set performance. This process is repeated as we continue to add categories to the system. The whole process is repeated across multiple dataset splits to ensure fair comparisons and estimate error. While [141] suggest a particular openness measure, it does not address the incremental learning paradigm. We fixed the number of unknown categories and report performance as series of known categories are incrementally added. Thus, open world evaluation involves varying of two variables: number of known categories in training (incremental learning) and number of unknown categories during testing (open set learning) leading to surface plots as shown in Fig 2.3.

Multi-class classification error [28] for a system $F_{\mathcal{K}}(.)$ with test samples $\{(x_i, y_i)\}_{i=1}^{N}, y_i \in \mathcal{K}$ is given as $\epsilon_{\mathcal{K}} = \frac{1}{N}\sum_{i=1}^{N} [\![F_{\mathcal{K}}(x_i) \neq y_i]\!]$. For open world testing the evaluation must keep track of the errors which occur due to standard multi-class classification over known categories as well as errors between known and



(a) 50 categories in metric learning phase.

(b) 200 categories in metric learning phase.

Figure 2.3: Open World learning on data from ILSVRC'10 challenge. Top-1 accuracy is plotted as a function of known classes in the system and unknown classes used during testing. NNO performs at par with NCM in closed set testing (marked with arrows in above figure) as categories are added incrementally to the system. As number of unknown categories are increased during testing phase, NNO continues to remain robust while performance of NCM suffers significantly. The proposed Nearest Non-Outlier (NNO) approach of handling unknown categories based on extending NCM with Compact Abating Probabilities remains robust in both circumstances: as more number of categories are added to the system and as the system is tested with more unknown categories. The current state-of-the-art on open set recognition 1vSet algorithm [141] and standard SVM [43] is shown above as a line, as neither of them possess incremental learning capabilities. Fig 2.3a and Fig 2.3b shows results when 50 and 200 categories were used for initial metric learning and parameter estimation.

unknown categories. Consider evaluation of N samples from $\mathcal{K}$ known categories and N' samples from $\mathcal{U}$ unknown categories leading to $(N + N')$ test samples and $\mathcal{K} \cup \mathcal{U} \in X$. Thus, open world error ϵ_{OW} for a system $F_{\mathcal{K}}(.)$ trained over $\mathcal{K}$ categories is given as:

$$\epsilon_{OW} = \epsilon_{\mathcal{K}} + \frac{1}{N'} \sum_{j=N+1}^{N'} [\![F_{\mathcal{K}}(x_j) \neq unknown]\!] \tag{2.9}$$

2.4.2 Experimental Results

In the first experiment, we do incremental learning from a base of relatively few (50) categories and we add 50 categories incrementally. For NCM and NNO systems, we update the means of the categories. We repeat that expansion 3 times growing from 50 to 100 to 150 and finally 200 known classes. For close-set testing we therefore have training and testing with 50, 100, 150 and 200 categories. To test open world performance, we considering an additional set of 100, 200 and 500 unknown categories showing up during testing. For example, open world testing with 100 unknown categories for a system that is trained with 50 categories would have a total of 50 + 100 i.e. 150 categories in testing. The experiment is thus a function of two variables : total number of known set of categories learned during training (both metric learning and incremental learning phase) and unknown set of categories seen by the system during testing phase. Varying both of these leads to performance being shown as the surface plots shown in 2.3. The plot shows showing top-1 accuracy where we treat unknown as its own class.

We note that each testing phase is independent and for these experiments we do not provide feedback for the results of testing as labels for the incremental training – i.e. we still presume perfect labels for each incremental training round. In this model, misclassifications in testing only impact the workload level of human labelers. If the open-world results of testing were used in a semi-supervised manner, the compounding of errors would significantly amplify the difference in the algorithm accuracy.

The back edges of the plot provide 2 easy to separate views. To see pure incremental learning, we incrementally add categories to the system in closed set testing paradigm. This is shown in the back right portion of Fig. 2.3a, where the performance of NCM (red) and NNO (green) drop similarly and rather gracefully, which is expected. However, as we increase openness for a fixed number of training classes, the back left of edge of Fig. 2.3a, the impact on NCM is a dramatic loss of performance even for the non-incremental growth case. This is caused by errors for the unknown categories, something NCM was not designed to handle and the NCM error is dominated by the second term in Eqn 2.9. As we can see the standard SVM also has a dramatic drop in accuracy as openness increases. Both the NNO and 1vSet algorithm, designed for open set recognition, degrade gracefully. The 1vSet and SVM don't have a way to incrementally add classes and are curves not surfaces in the plots. So the 1vSet, while slight better than NNO for pure open set on 50 categories, does not support open world recognition.

Open world recognition needs to support increasing classes while handling unknowns, so is can be viewed as the performance as known training classes increase for non-zero number of unknowns. At all such points in the plot, NNO significantly dominates the NCM algorithm.

In second experiment, we consider 200 categories for metric learning and parameter estimation, and successively add 100 categories in each of three incremental learning phases. By the end of the learning process, the system needs to learn a total of 500 categories. Open world evaluation of the system is carried out as before by considering with 100, 200 and 500 additional unknown categories with results show in Fig 2.3b. In final stage of the learning process i.e 500 categories for training and 500 (known) + 500 (unknown) categories for open set testing, we use all 1000 categories from ImageNet for our evaluation process. We observe that NNO again dominates the baselines for open world recognition; this time even outperforming 1vSet for open set testing on 200 classes. On the largest scale task involving 500 categories in training and 1000 categories in testing, we observe that NNO provides almost 74% improvement over NCM. Also note performance starting with 200 classes (2.3b) is better than starting with 50 classes (2.3a), i.e. increased classes for the metric learning improves both NNO and NCM performance. We repeated the above experiments over three cross-class folds and found the standard deviation to be on the order of ± 1% which is not visible in the figure.

The training time required for the initial metric learning process depends on the SGD speed and convergence rate. We used close to 1M iterations which resulted in metric-learning time of 15 hours in case of 50 categories and 22 hours in case of metric learning for 200 categories. Given the metric, the learning of new classes via the update process is extremely fast as it is simply computation of means from labeled data. For this work we fix the metric, though future work might explore incrementally updating the metric as well. The majority of time in update process is dominated by feature extraction and file I/O. However, these operations could be easily optimized for real-time operations. The NNO Multi-class recognition and detecting novel classes is also easily done in real time.

2.5 Discussion

In this work, we formalize the problem of open world recognition and provide an open world evaluation protocol. We extend existing work on NCM classifiers and show how to adapt it for open world recognition. The proposed NNO algorithm consistently outperforms NCM on open world recognition tasks and is comparable to NCM on closed set – we gain robustness to the open world without much sacrifice.

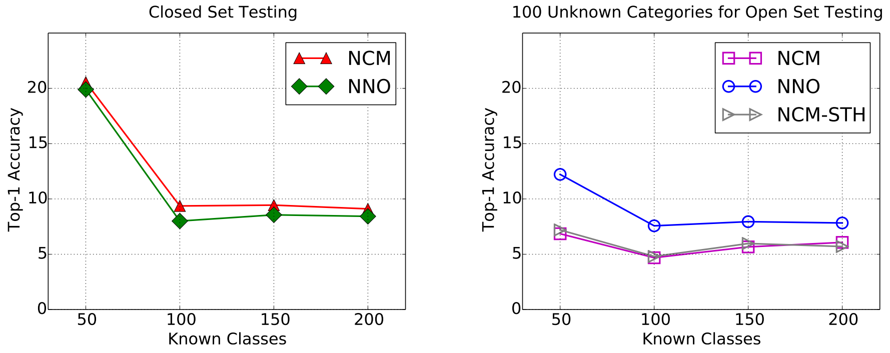
There are multiple implications of our experiments. First, we demonstrate suitability of NNO for large scale recognition tasks in dynamic environments. NNO allows construction of scalable systems that can be updated incrementally with additional classes and that are robust to unseen categories. Such systems are suitable where minimum downtime is desired.

Second, as can be seen in Figs 2.3a and Fig 2.3b NNO offers significant improvement for open world recognition while for close set recognition NNO remains relatively close to NCM in performance.

We also noted that as the number of classes incrementally grew, the closed and open set performance NNO seems to converge, i.e. the front right edge of the plots in Figs. 2.3a an 2.3b are very flat. This observation suggests that adding classes in a system may also be limited by open space risk. We conjecture that as the number of classes grows, the close world performance converges to an open world performance and thus open world recognition is a more natural setting for building scalable systems.

While we provide one viable open world extension, the theory herein allows a broad range of approaches; more expressive models, improved CAP models and better open set probability calibration should be explored.

Open world evaluation across multiple features for a variety of applications is an important future work. Recent advances in deep learning and other areas of visual recognition have demonstrated significant improvements in absolute performance. The best performing systems on such tasks use a parallel system and train for days. Extending these systems to support incremental open world performance may allow one to provide a hybrid solution where one reuses the deeply learned features with a top layer of an open world multi-class



(a)

(b)

Figure 2.4: Effect of open set performance of thresholding softmax probabilities. Fig 2.4a shows performance with closed set testing and 2.4b shows performance on open set testing with 100 unknown categories. Metric learning was performed on 50 categories, followed by incremental learning phase with 50 categories in each phase. NCM-STH denotes NCM algorithm with open set testing with thresholded softmax probabilities. As can be seen clearly, just thresholding a probability estimates does not produce good open set performance

algorithm. While scalable learning in the open world is critical for deploying computer vision applications in the real world, high performing systems enable adoption by masses. Pushing absolute performance on large scale visual recognition challenges [9], and development of scalable systems for the open world are essentially two sides of the same coin.

In this supplemental section, we provide additional material to further the reader's understanding of the work on Open World Recognition, CAP models and the Nearest Non-Outlier algorithm that we presented in the main paper. We present additional experiments on ILSVRC 2010 dataset. We then present experiments on ILSVRC 2012 dataset to demonstrate the performance gain of NNO over NCM (see fig 3 in the main paper) are not feature/dataset specific. We then provide algorithmic pseudocode for implementing the NNO algorithm. Finally we discuss the 1vsSet extension to liblinear, its parameter tuning and its computational savings.

2.6 Experiments on ILSVRC 2010

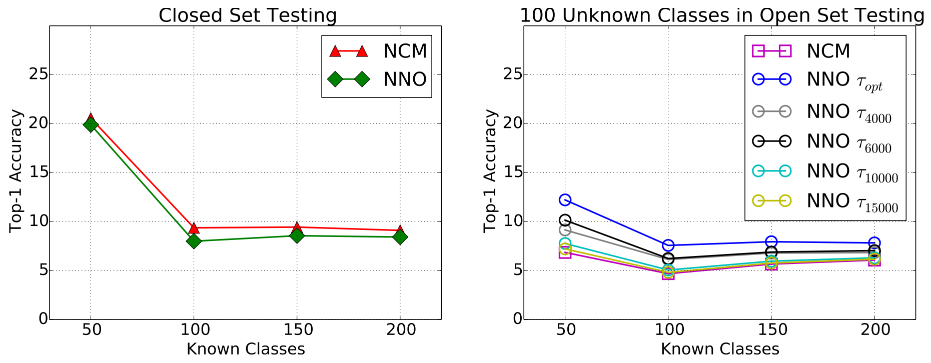
2.6.1 Thresholding NCM-Softmax for ILSVRC 2010

In section 4.1 of the main paper, we explain the process of rejecting samples from unseen categories to balance open space risk and defined in Eq. 8, a probability function which is thresholded at zero. At first it might seem like a viable idea to just threshold the original softmax probability used in NCM. As explained in the main paper this will fail for open set because the normalization is improper and hence the softmax probability calibration will bias results. To convince the skeptical reader, we add a small experiment, similar to fig 3 in the main paper, and show the performance of classifying samples as unknown by directly thresholding softmax probabilities. As this is a smaller experiment, we show 2D plots instead of a 3D surface as the system is tested in closed set settings (fig 2.4a) and open set settings with 100 unknowns (fig 2.4b). NNO algorithm performs comparable with NCM in closed set settings. The reader can observe the performance of NCM-STH is similar to NCM and significantly worse than NNO on open set testing with 100 unknowns. Just thresholding the softmax probability is not enough, because its normalization keeps it from decaying as one move away from known data. This result confirms the suitability of balancing open set risk with Eq 8,using transformed learned Mahalanobis distance to the NCM. The results from this experiment are shown in fig 2.4.

2.6.2 Performance of NNO for different values of τ

Section 4.1 and 5.1 in the main paper describes NNO algorithm in detail and steps involved in estimating optimal τ required to balance open space risk. It is natural to ask how sensitive are the results to the choice of τ. In this section, we show the effect of different values of τ on the performance of NNO tested in open set settings with 100 unknown categories, to provide reader the feeling for the sensitivity to the parameter.

In the experimental results shown in Fig 3 in the main paper, we used optimal τ for evaluation purpose, which was approximately 5000. The optimal value near the beginning of a broad peak and small changes in τ have minimal impact. Even increasing it by 20% has only a small impact on open set testing. Fig 2.5 shows performance for varying set of τ. τ_{opt} is the optimal threshold that was selected. We observe that performance of NNO continues to improve as we near the optimal threshold from above. For a threshold value lower than



(a)

(b)

Figure 2.5: The above figure shows the effect of varying threshold τ on top-1 accuracy on ILSVRC'10 data. The results from closed set testing are shown in fig 2.5a and results from open set testing with 100 unknown categories are shown in fig 2.5b. Here $\tau_{opt} = 5000$, which was the selected threshold for experiments in fig 3a. For a threshold value lower than τ_{opt}, the number of correct predictions retained reduces significantly.

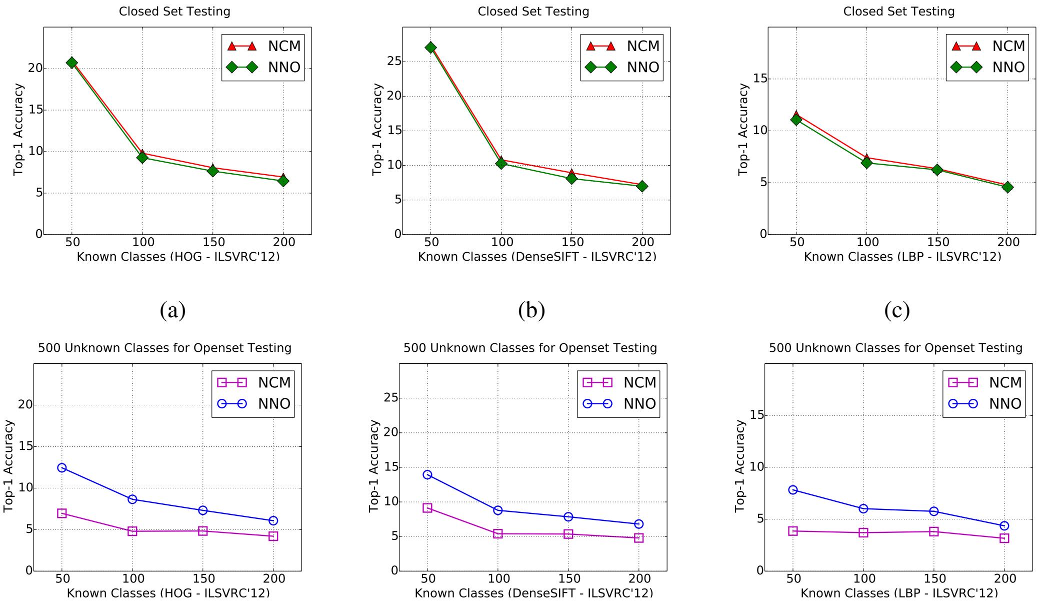
τ_{opt} (e.g. 4000), the number of false rejects raises significantly. Thus, a balance between correct predictions retained and unknown categories rejected has to be maintained by the selected τ_{opt}. The results are obtained on ILSVRC'10 dataset, similar to fig 3a in the main paper. In our experiments, we observed similar trends for all other experiments – poor performance below the optimal τ and insensitivity to modest changes above it.

2.7 Experiments on ILSVRC 2012 Dataset

As noted in section 5 (Experiments) in the main paper, we used ILSVRC 2010 dataset because we needed access to ground truth labels. Ground truth is necessary to perform the open world recognition test protocol, which includes selecting known and unknown set of categories. In this section, we perform additional experiments on the training subset of ILSVRC 2012 [127] [3] dataset across multiple features to show that the effectiveness of NNO algorithm for closed set and open set tasks does not significantly depend on feature type.

Since ground truth is not available for ILSVRC'12 dataset, we split the training data provided by the authors into training and test split. The number of categories is the same, this just limits the number of images per class used. We use 70% of training data to train models and 30% of the data for evaluation. This process

[3] ILSVRC dataset remained unchanged between 2012, 2013 and 2014. Ground truth labels are available for training data only.



(a) (b) (c)

(d) (e) (f)

Figure 2.6: The above figure shows experiments on ILSVRC'12 data with 50 classes used for metric learning. The top row shows performance on closed set testing and bottom row shows performance on open set testing with 500 unknown categories. Figs 2.6a, 2.6d are for HOG features [32], figs 2.6b, 2.6e are for DenseSIFT features [85] and figs 2.6c, 2.6f are for LBP features [109]. The training data for ImageNet'12 was split into train (70%) and test split (30%). This is similar to experiment shown in fig 3a in the main paper. The absolute performance varies from feature to feature, however we see similar trends in performance as we saw on ILSVRC'10 data.

is repeated over multiple folds. Once the data is split into training and test split the remaining procedure for metric learning and incremental learning is similar to that in section 5 (Experiments) in the main section. We conduct two sets of similar experiments on ILSVRC'12 data: metric learning with 50 and 200 initial categories as shown in Figs 3 and 4 in the main paper. The closed set and open set testing is conducted in similar manner as well. While the open world experimental setup for ILSVRC'12 is not ideal because of the smaller number of images per class, the goal of this experiment is to show that the advantages of NNO are not feature dependent.

We use pre-computed features as provided on cloudcv.org [2]. We consider three set of features as follows:

1. **DenseSIFT:** SIFT descriptors are densely extracted [85] using a flat window at two scales (4 and 8 pixel radii) on a regular grid at steps of 5 pixels. The three descriptors are stacked together for each HSV color channels, and quantized into 300 visual words by k-means. The features used in the main paper are similar to these features, except in the main paper, denseSIFT features were quantized into 1000 visual words by k-means.

2. **Histogram of Oriented Gradients (HOG):** HOG features are used in wide range of visual recognition tasks [32]. HOG features are densely extracted on a regular grid at steps of 8 pixels. HOG features are computed using code provided by [46]. This gives a 31-dimension descriptor for each node of the grid. Finally, the features are quantized into 300 visual words by k-means.

3. **Local Binary Patterns (LBP):** Local Binary Patterns (LBP) [109] is a texture feature based on occurrence histogram of local binary patterns. It has been widely used for face recognition and object recognition. The feature dimensionality used was 59.

Results on HOG [32] are shown in 2.6a, 2.6d, on DenseSIFT [85] are shown in 2.6b, 2.6e and on LBP features [109] are shown in 2.6c, 2.6f respectively. The absolute performance with DenseSIFT features is the best, followed by HOG and LBP. The DenseSIFT is very similarly to the results on ILSVRC 2010. Moreover, from these experiments we observe similar trends across all features to the trends seen in Fig 3 in the main paper. We see that as closed set performance of NCM and NNO is comparable while NCM with open set suffers significantly when tested with unknown set of categories. We continue to see significant gains of NNO over NCM with open set testing across HOG and denseSIFT features. We also observe the trend where as

Algorithm 1 Nearest Non-Outlier Algorithm

```
Require: $X_k, \mu_k$    ▷ Initial Training Data $X_k$ from $k$ categories and their means $\mu_k$
  function METRICLEARN($X_k, \mu_k$)
    $W$ = NCMMetricLearn($X_k, \mu_k$)    ▷ Train NCM Classifier
    for $i = 1 \to m$ do    ▷ Over multiple folds
      $X_{k_K}, X_{k_U}$ = SplitKnownUnknown($X_k$)    ▷ Split Training Data into known and unknown set
      $\tau_i$ = OpenSetThresh($X_{k_K}, X_{k_U}$)    ▷ Estimate optimal $\tau_i$ for each split
    end for
    $\tau = \frac{1}{m} \sum_{i=1}^{m} \tau_i$    ▷ Use average $\tau$
    NNOModel$_k$ = $[W, \mu_k, \tau]$
  end function
Require: NNOModel$_k$, $X_n, \mu_n$    ▷ Add additional data $X_n$ from $n$ categories with means $\mu_n$
  function INCREMENTALLEARN( NNOModel$_k$, $X_n, \mu_n$)

    NNOModel$_{k+n}$ = $[W, [\mu_k, \mu_n], \tau]$    ▷ Update model with means $\mu_n$
  end function
```

we add more categories in the system, the closed set and open set performance begin to converge. Thus, it is reasonable to conclude that the performance gain seen in terms of NNO on open set testing is not feature dependent. These observations are consistent with our observations from experiments on ILSVRC'10 data.

2.8 Algorithmic Pseudocode for Nearest Non-Outlier (NNO)

In this section, we provide pseudocode for Nearest Non-Outlier algorithm as described in section 4.1 in the main paper. The algorithm proceeds in multiple steps. In the first step, features are normalized by the mean and standard deviation over the starting subset. The initial set of features is used to perform metric learning. Following this step, threshold τ for open set NNO is estimated using per class decisions using per Eq. 8 in the main paper and a cross class validation procedure of [63] training data splits. The complete pseudocode is given in Alg 1

2.9 Liblinear 1-vs-set extension Baseline

For this paper, we needed a baseline from an existing open set algorithm, so the performance of NNO was properly placed in context. Unfortunately, the 1-vs-all nature of the computations used in the various libsvm open set extensions developed to date ([141], [142], and [63]) make them very expensive for use in the scale of experiments in this paper. We tried the linear 1-vs-set machine of [141] and abandoned when it was not done with one fold of the smallest experiment after 3 weeks of computing. Recognizing that to scale we needed a more efficient implementation we adapted the concept/code from [141] into a liblinear ([43]) implementation. While still a 1-vs-all implementation, the liblinear library uses a much more efficient algorithm for estimation of the hyperplane than the libsvm and reduced the computation from more than 30,000 minutes (3weeks) to about 5 minutes for processing one fold of 50 classes. The default liblinear solver (L2-regularized logistic regression (primal)) did not converge so we switched to L2-regularized L2-loss support vector classification (primal).

A second issue we address in the extension is supporting searching for the "pressure" parameters discussed in the original 1-vs-set paper [141]. There is a parameter for the near plane, close to the negative data and a second parameter for the far plane. That paper provided no formal process for setting the pressures, saying it is problem dependent. Initially, we used strict 1-vs-set slabs with "-G 0 1" and the performance was quite weak with only about 5% top-1 accuracy on closed set testing. The out of the box liblinear was doing much better. Upon examination we found that liblinear was choosing the class with largest score, even if that score was negative. Thus, we needed to move beyond the default 1-vs-set machine parameters and add some near pressure to capture the negatives. The obvious process is a grid search of parameters using training data. For that process, we choose to mimic the process for selection of τ in the NNO algorithm. In particular we split the 50 classes into 3 folds of 32 classes each: first 32 classes, last 32 classes and first 16 + last 16). We trained a 1-vs-set machine with both near and far pressure and then used them to predict performance on the full 50 class training sample. Among the parameters that achieved at 90% or better recall on at least one fold, we choose the set of parameters that optimized F-measure. Our grid search included: near pressure = [0 .05 .1 .2 .3 .4 .5 .6. 7. 8. .9 1 2 3 4 5 6] and far pressure = [1 2 3 4 5 10 15 20 25 30 35 40 45 50 55 60 65]. We found the optimal parameters to be near pressure =.2, and far pressure = 55, i.e. a small expansion inside the margin space but a large expansion on the back side of the training data. We used the same parameters for 200 classes.

With that many parameters to grid search, we decided that retraining a new model with liblinear, even though it is efficient, was still computationally expensive at 56 hours for our grid search. Recognizing that other researchers may face a similar issue, we decided to develop a more efficient way to search. We adapted the liblinear 1-vs-set extension to support a mode that reads an existing liblinear model (raw or 1-vs-set), as well as the training data and 1-v-set parameters such as pressure, then computes the optimized 1-vs-set data from that starting model. The result is that for one fold it takes about 7 seconds to train model for a given set of pressures and 2 second to predict with that model on the full training data. Thus the full grid search of 225 values and 3 folds, was reduced to under 2 hours. This mode allows anyone that has liblinear models to quickly adapt existing models for 1-vs-set testing without the costs of retraining the original model. The 1-vs-set liblinear extension discussed in this section is available as open-source from `https://github.com/Vastlab/liblinear.git`.

2.10 Numerical Values for Results

In this section, we provide numerical values for clarity. The following table shows numerical values for the surface plot presented in Fig 3a in the main paper for each algorithm. Metric learning/parameter estimation was performed on 50 categories followed by updating classifier with 50 additional categories. As 1vSet and SVM do not have incremental learning capabilities, results with only varying number of unknown categories in testing are shown.

The following tables show results shown in Fig 3b in the main paper. Metric learning/parameter estimation was performed with 200 initial categories. Similarly, we provide results for 1vSet and SVM algorithms with varying number of unknown categories. Incremental learning results for 1vSet and SVM are not provided as they do not possess those capabilities.

NCM		# Known Categories in Training			
		50	100	150	200
# Unknown Categories in testing	0	20.5600	9.3667	9.4400	9.1033
	100	6.8533	4.6833	5.6640	6.0689
	200	4.1120	3.1222	4.0457	4.5517
	500	1.8691	1.5611	2.1785	2.6010

Table 2.1: NCM with metric learning performed on 50 categories

NNO		# Known Categories in Training			
		50	100	150	200
# Unknown Categories in testing	0	19.8933	8.0000	8.5600	8.4267
	100	12.2178	7.5700	7.9440	7.8267
	200	10.0267	7.2667	7.5752	7.5667
	500	9.0412	7.5656	7.7015	7.6867

Table 2.2: NNO with metric learning performed on 50 categories

2.11 Acknowledgement

We would like to thank Thomas Mensink (ISLA, Informatics Institute, University of Amsterdam) for sharing code and Marko Ristin (ETH Zurich) for sharing features. The work support in part by NSF Research Grant IIS-1320956 (Open Vision - Tools for Open Set Computer Vision and Learning) and a UCCS Graduate School Fellowship.

1vsSet		# Known Categories in Training
		50
# Unknown Categories in testing	0	16.0267
	100	13.5689
	200	11.2827
	500	9.0412

Table 2.3: 1vSet algorithm tested with varying number of unknown categories. Model trained on 50 categories

SVM		# Known Categories in Training
		50
# Unknown Categories in testing	0	21.12
	100	7.04
	200	4.224
	500	1.92

Table 2.4: SVM algorithm tested with varying number of unknown categories. Model trained on 50 categories

NCM		# Known Categories in Training			
		200	300	400	500
# Unknown Categories in testing	0	22.6133	10.1400	9.3300	7.2987
	100	12.4089	2.3550	2.6640	2.7489
	200	9.3067	1.8840	2.2200	2.3562
	500	5.3181	1.1775	1.4800	

Table 2.5: NCM with metric learning performed on 200 categories

NNO		# Known Categories in Training			
		200	300	400	500
# Unknown Categories in testing	0	22.4033	9.1000	9.2833	7.0307
	100	17.7378	6.3933	5.8520	5.3478
	200	16.2900	7.4627	6.8178	6.2276
	500	12.4343	7.3167	6.8926	1.6493

Table 2.6: NNO with metric learning performed on 200 categories

1vsSet		# Known Categories in Training
		200
# Unknown Categories in testing	0	14.0933
	100	12.4044
	200	11.6617
	500	10.8448

Table 2.7: 1vSet algorithm tested with varying number of unknown categories. Model trained on 200 categories

SVM		# Known Categories in Training
		200
# Unknown Categories in testing	0	19.25
	100	12.8333
	200	9.625
	500	5.5

Table 2.8: SVM algorithm tested with varying number of unknown categories. Model trained on 200 categories

Chapter 3

Towards Open Set Deep Networks

Deep networks have produced significant gains for various visual recognition problems, leading to high impact academic and commercial applications. Recent work in deep networks highlighted that it is easy to generate images that humans would never classify as a particular objectclass, yet networks classify such images high confidence as that given class – deep network are easily fooled with images humans do not consider meaningful. The closed set nature of deep networks forces them to choose from one of the known classes leading to such artifacts. Recognition in the real world is open set, i.e. the recognition system should reject unknown/unseen classes at test time. We present a methodology to adapt deep networks for open set recognition, by introducing a new model layer, OpenMax, which estimates the probability of an input being from an unknown class. A key element of estimating the unknown probability is adapting Meta-Recognition concepts to the activation patterns in the penultimate layer of the network. OpenMax allows rejection of "fooling" and unrelated open set images presented to the system; OpenMax greatly reduces the number of *obvious errors* made by a deep network. We prove that the OpenMax concept provides bounded open space risk, thereby formally providing an open set recognition solution. We evaluate the resulting open set deep networks using pre-trained networks from the Caffe Model-zoo on ImageNet 2012 validation data, and thousands of fooling and open set images. The proposed OpenMax model significantly outperforms open set recognition accuracy of basic deep networks as well as deep networks with thresholding of SoftMax probabilities.

3.1 Introduction

Computer Vision datasets have grown from few hundred images to millions of images and from few categories to thousands of categories, thanks to research advances in vision and learning. Recent research in deep networks has significantly improved many aspects of visual recognition [157, 24, 76]. Co-evolution of rich representations, scalable classification methods and large datasets have resulted in many commercial applications [35, 160, 111, 44]. However, a wide range of operational challenges occur while deploying recognition systems in the dynamic and ever-changing real world. A vast majority of recognition systems are designed for a static closed world, where the primary assumption is that all categories are known a priori. Deep networks, like many classic machine learning tools, are designed to perform closed set recognition.

Recent work on open set recognition [138, 139] and open world recognition [8], has formalized processes for performing recognition in settings that require rejecting unknown objects during testing. While one can always train with an "other" class for uninteresting classes (*known unknowns*), it is impossible to train with all possible examples of unknown objects. Hence the need arises for designing visual recognition tools that formally account for the "unknown unknowns"[126]. Altough a range of algorithms has been developed to address this issue [31, 138, 139, 155, 16], performing open set recognition with deep networks has remained an unsolved problem.

In the majority of deep networks [76, 157, 24], the output of the last fully-connected layer is fed to the SoftMax function, which produces a probability distribution over the N known class labels. While a deep network will always have a most-likely class, one might hope that for an unknown input all classes would have low probability and that thresholding on uncertainty would reject unknown classes. Recent papers have shown how to produce "fooling" [107] or "rubbish" [53] images that are visually far from the desired class but produce high-probability/confidence scores. They strongly suggests that thresholding on uncertainty is not sufficient to determine what is unknown. In Sec. 3.3, we show that extending deep networks to threshold SoftMax probability improves open set recognition somewhat, but does not resolve the issue of fooling images. Nothing in the theory/practice of deep networks, even with thresholded probabilities, satisfies the formal definition of open set recognition offered in [138]. This leads to the first question addressed in this paper, *"how to adapt deep networks support to open set recognition?"*

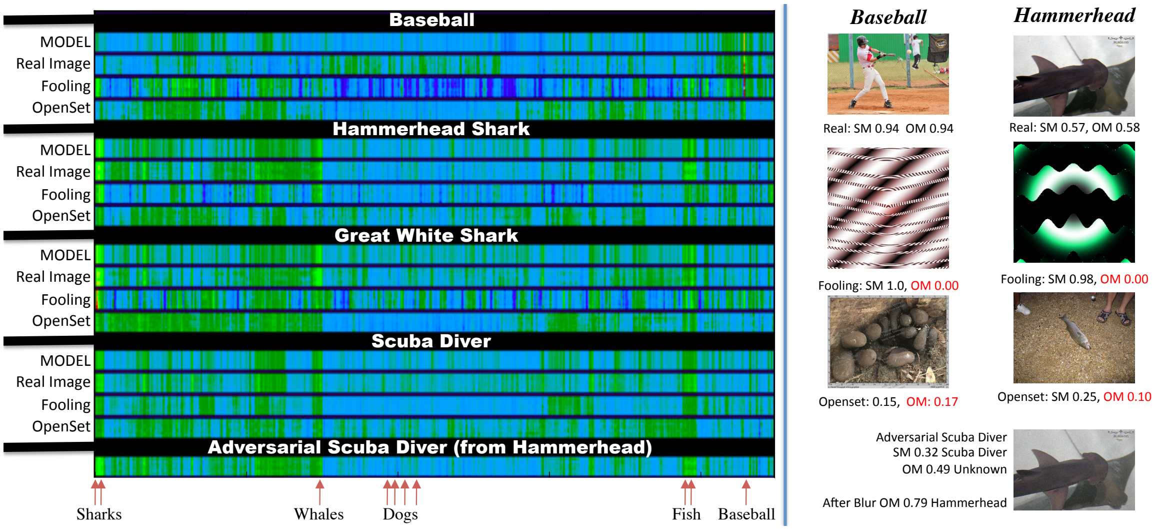


Figure 3.1: Examples showing how an activation vector model provides sufficient information for our Meta-Recognition and OpenMax extension of a deep network to support open-set recognition. The OpenMax algorithm measures distance between an *activation vector (AV)* for an input and the model vector for the top few classes, adjusting scores and providing an estimate of probability of being unknown. The left side shows activation vectors (AV) for different images, with different AVs separated by black lines. Each input image becomes an AV, displayed as 10x450 color pixels, with the vertical being one pixel for each of 10 deep network channel activation energy and the horizontal dimension showing the response for the first 450 ImageNet classes. Ranges of various category indices (sharks, whales, dogs, fish, etc.) are identified on the bottom of the image. For each of four classes (baseball, hammerhead shark, great white shark and scuba diver), we show an AV for 4 types of images: the model, a real image, a fooling image and an open set image. The AVs show patterns of activation in which, for real images, related classes are often responding together, e.g., sharks share many visual features, hence correlated responses, with other sharks, whales, large fishes, but not with dogs or with baseballs. Visual inspection of the AVs shows significant difference between the response patterns for fooling and open set images compared to a real image or the model AV. For example, note the darker (deep blue) lines in many fooling images and different green patterns in many open set images. The bottom AV is from an "adversarial" image, wherein a hammerhead image was converted, by adding nearly invisible pixel changes, into something classified as scuba-diver. On the right are two columns showing the associated images for two of the classes. Each example shows the SoftMax (SM) and OpenMax (OM) scores for the real image, the fooling and open set image that produced the AV shown on the left. The red OM scores implies the OM algorithm classified the image as unknown, but for completeness we show the OM probability of baseball/hammerhead class for which there was originally confusion.

The SoftMax layer is a significant component of the problem because of its closed nature. We propose an alternative, OpenMax, which extends SoftMax layer by enabling it to predict an unknown class. OpenMax incorporates likelihood of the recognition system failure. This likelihood is used to estimate the probability for a given input belonging to an unknown class. For this estimation, we adapt the concept of Meta-Recognition[140, 183, 67] to deep networks. We use the scores from the penultimate layer of deep networks (the fully connected layer before SoftMax, e.g., FC8) to estimate if the input is "far" from known training data. We call scores in that layer the *activation vector*(AV). This information is incorporated in our OpenMax model and used to characterize failure of recognition system. By dropping the restriction for the probability for known classes to sum to 1, and rejecting inputs far from known inputs, OpenMax can formally handle unknown/unseen classes during operation. Our experiments demonstrate that the proposed combination of OpenMax and Meta-Recognition ideas readily address open set recognition for deep networks and reject high confidence fooling images [107].

While fooling/rubbish images are, to human observers, clearly not from a class of interest, adversarial images [53, 158] present a more difficult challenge. These adversarial images are visually indistinguishable from a training sample but are designed so that deep networks produce high-confidence but incorrect answers. This is different from standard open space risk because adversarial images are "near" a training sample in input space, for any given output class.

A key insight in our opening deep networks is noting that "open space risk" should be measured in feature space, rather than in pixel space. In prior work, open space risk is not measured in pixel space for the majority of problems [138, 139, 8]. Thus, we ask "is there a feature space, ideally a layer in the deep network, where these adversarial images are *far away* from training examples, i.e., a layer where unknown, fooling and adversarial images become outliers in an open set recognition problem?" In Sec. 3.2.1, we investigate the choice of the feature space/layer in deep networks for measuring open space risk. We show that an extreme-value meta-recognition inspired distance normalization process on the overall activation patterns of the penultimate network layer provides a rejection probability for OpenMax normalization for unknown images, fooling images and even for many adversarial images. In Fig. 4.1, we show examples of activation patterns for our model, input images, fooling images, adversarial images (that the system can reject) and open set images.

In summary the contributions of this paper are:

1. Multi-class Meta-Recognition using Activation Vectors to estimate the probability of deep network failure
2. Formalization of open set deep networks using Meta-Recognition and OpenMax, along with the proof showing that proposed approach manages open space risk for deep networks
3. Experimental analysis of the effectiveness of open set deep networks at rejecting unknown classes, fooling images and obvious errors from adversarial images, while maintaining its accuracy on testing images

3.2 Open Set Deep Networks

A natural approach for opening a deep network is to apply a threshold on the output probability. We consider this as rejecting uncertain predictions, rather than rejecting unknown classes. It is expected images from unknown classes will all have low probabilities, i.e., be very uncertain. This is true only for a small fraction of unknown inputs. Our experiments in Sec. 3.3 show that thresholding uncertain inputs helps, but is still relatively weak tool for open set recognition. Scheirer *et al.* [138] defined open space risk as the risk associated with labeling data that is "far" from known training samples. That work provides only a general definition and does not prescribe how to measure distance, nor does it specify the space in which such distance is to be measured. In order to adapt deep networks to handle open set recognition, we must ensure they manage/minimize their open space risk and have the ability to reject unknown inputs.

Building on the concepts in [139, 8], we seek to choose a layer (feature space) in which we can build a compact abating probability model that can be thresholded to limit open space risk. We develop this model as a decaying probability model based on distance from a learned model. In following section, we elaborate on the space and meta-recognition approach for estimating distance from known training data, followed by a methodology to incorporate such distance in decision function of deep networks. We call our methodology OpenMax, an alternative for the SoftMax function as the final layer of the network. Finally, we show that the overall model is a compact abating probability model, hence, it satisfies the definition for an open set recognition.

3.2.1 Multi-class Meta-Recognition

Our first step is to determine when an input is likely not from a known class, i.e., we want to add a meta-recognition algorithm [140, 183] to analyze scores and recognize when deep networks are likely incorrect in their assessment. Prior work on meta-recognition used the final system scores, analyzed their distribution based on Extreme Value Theory (EVT) and found these distributions follow Weibull distribution. Although one might use the per class scores independently and consider their distribution using EVT, that would not produce a compact abating probability because the fooling images show that the scores themselves were not from a compact space close to known input training data. Furthermore, a direct EVT fitting on the set of class post recognition scores (SoftMax layer) is not meaningful with deep networks, because the final SoftMax layer is intentionally renormalized to follow a logistic distribution. Thus, we analyze the penultimate layer, which is generally viewed as a per-class estimation. This per-class estimation is converted by SoftMax function into the final output probabilities.

We take the approach that the network values from penultimate layer (hereafter the *Activation Vector (AV)*), are not an independent per-class score estimate, but rather they provide a distribution of what classes are "related." In Sec. 3.2.2 we discuss an illustrative example based on Fig. 4.1.

Our overall EVT meta-recognition algorithm is summarized in Alg. 2. To recognize outliers using AVs, we adapt the concepts of Nearest Class Mean [162, 102] or Nearest Non-Outlier [8] and apply them per class within the activation vector, as a first approximation. While more complex models, such as nearest class multiple centroids (NCMC) [104] or NCM forests [124], could provide more accurate modeling, for simplicity this paper focuses on just using a single mean. Each class is represented as a point, a *mean activation vector (MAV)* with the mean computed over only the correctly classified training examples (line 2 of Alg. 2).

Given the MAV and an input image, we measure distance between them. We could directly threshold distance, e.g., use the cross-class validation approach of [8] to determine an overall maximum distance threshold. In [8], the features were subject to metric learning to normalize them, which makes a single shared threshold viable. However, the lack of uniformity in the AV for different classes presents a greater challenge

Algorithm 2 EVT Meta-Recognition Calibration for Open Set Deep Networks, with per class Weibull fit to η largest distance to mean activation vector. Returns libMR models ρ_j which includes parameters τ_i for shifting the data as well as the Weibull shape and scale parameters:κ_i, λ_i.

```
Require: FitHigh function from libMR
Require: Activation levels in the penultimate network layer $\mathbf{v}(\mathbf{x}) = v_1(x) \dots v_N(x)$
Require: For each class $j$ let $S_{i,j} = v_j(x_{i,j})$ for each correctly classified training example $x_{i,j}$.
1: for $j = 1 \dots N$ do
2:     Compute mean AV, $\mu_j = mean_i(S_{i,j})$
3:     EVT Fit $\rho_j = (\tau_j, \kappa_j, \lambda_j) =$ FitHigh($\|\hat{S}_j - \mu_j\|, \eta$)
4: end for
5: Return means $\mu_j$ and libMR models $\rho_j$
```

and, hence, we seek a per class meta-recognition model. In particular, on line 3 of Alg. 2 we use the libMR [140] FitHigh function to do Weibull fitting on the largest of the distances between all correct positive training instances and the associated μ_i. This results in a parameter ρ_i, which is used to estimate the probability of an input being an outlier with respect to class i.

Given ρ_i, a simple rejection model would be for the user to define a threshold that decides if an input should be rejected, e.g., ensuring 90% of all training data will have probability near zero of being rejected as an outlier. While simple to implement, it is difficult to calibrate an absolute Meta-Recognition threshold because it depends on the unknown unknowns. Therefore, we choose to use this in the OpenMax algorithm described in Sec. 3 which has a continuous adjustment.

We note that our calibration process uses only correctly classified data, for which class j is rank 1. At testing, for input $\mathbf{x}$ assume class j has the largest probability, then $\rho_j(\mathbf{x})$ provides the MR estimated probability that $\mathbf{x}$ is an outlier and should be rejected. We use one calibration for high-ranking (e.g., top 10), but as an extension separate calibration for different ranks is possible. Note when there are multiple channels per example we compute per channel per class mean vectors $\mu_{j,c}$ and Weibull parameters $\rho_{j,c}$. It is worth remembering that *the goal is not to determine the training class of the input, rather this is a meta-recognition process used to determine if the given input is from an unknown class and hence should be rejected.*

3.2.2 Interpretation of Activation Vectors

In this section, we present the concept of activation vectors and meta-recognition with illustrative examples based on Fig. 4.1.

Closed Set: Presume the input is a valid input of say a hammerhead shark, i.e., the second group of activation records from Fig. 4.1. The activation vector shows high scores for the AV dimension associated with a great white shark. All sharks share many direct visual features and many contextual visual features with other sharks, whales and large fish, which is why Fig. 4.1 shows multiple higher activations (bright yellow-green) for many ImageNet categories in those groups. We hypothesize that for most categories, there is a relatively consistent pattern of related activations. The MAV captures that distribution as a single point. The AVs present a space where we measure the distance from an input image in terms of the activation of each class; if it is a great white shark we also expect higher activations from say tiger and hammerhead sharks as well as whales, but very weak or no activations from birds or baseballs. Intuitively, this seems like the right space in which to measure the distance during training.

Open Set: First let us consider an open set image, i.e., a real image from an unknown category. These will always be mapped by the deep network to the class for which SoftMax provides the maximum response, e.g., the images of rocks in Fig. 4.1 is mapped to baseball and the fish on the right is mapped to a hammerhead. Sometimes open set images will have lower confidence, but the maximum score will yield a corresponding class. Comparing the activation vectors of the input with the MAV for a class for which the input produced maximum response, we observe it is often far from the mean. However, for some open set images the response provided is close to the AV but still has an overall low activation level. This can occur if the input is an "unknown" class that is closely related to a known class, or if the object is small enough that it is not well distinguished. For example, if the input is from a different type of shark or large fish, it may provide a low activation, but the AV may not be different enough to be rejected. For this reason, it is still necessary for open set recognition to threshold uncertainty, in addition to directly estimating if a class is unknown.

Fooling Set: Consider a fooling input image, which was artificially constructed to make a particular class (e.g., baseball or hammerhead) have high activation score and, hence, to be detected with high confidence. While the artificial construction increases the class of interest's probability, the image generation process did not simultaneously adjust the scores of all related classes, resulting in an AV that is “far” from the model AV. Examine the 3rd element of each class group in Fig. 4.1 which show activations from fooling images. Many fooling images are visually quite different and so are their activation vectors. The many regions of very low activation (dark blue/purple) are likely because one can increase the output of SoftMax for a given class by reducing the activation of other classes, which in turn reduces the denominator of the SoftMax computation.

Adversarial Set: Finally, consider an adversarial input image [53, 158, 180], which is constructed to be close to one class but is mislabeled as another. An example is shown on the bottom right of Fig. 4.1. If the adversarial image is constructed to a nearby class, e.g., from hammerhead to great white, then the approach proposed herein will fail to detect it as a problem – fine-grained category differences are not captured in the MAV. However, adversarial images can be constructed between any pair of image classes, see [158]. When the target class is far enough, e.g., the hammerhead and scuba example here, or even farther such as hammerhead and baseball, the adversarial image will have a significant difference in activation score and hence can be rejected. We do not consider adversarial images in our experiments because the outcome would be more a function of that adversarial images we choose to generate – and we know of no meaningful distribution for that. If, for example, we choose random class pairs (a, b) and generated adversarial images from a to b, most of those would have large hierarchy distance and likely be rejected. If we choose the closest adversarial images, likely from nearby classes, the activations will be close and they will not be rejected.

The result of our OpenMax process is that open set as well as fooling or adversarial images will generally be rejected. Building a fooling or adversarial image that is not rejected means not only getting a high score for the class of interest, it means maintaining the relative scores for the 999 other classes. At a minimum, the space of adversarial/fooling images is significantly reduced by these constraints. Hopefully, any input that satisfies all the constraints is an image that also gets human support for the class label, as did some of the fooling images in Figure 3 of [107], and as one sees in adversarial image pairs fine-grain separated categories such as bull and great white sharks.

One may wonder if a single MAV is sufficient to represent complex objects with different aspects/views. While future work should examine more complex models that can capture different views/exemplars, e.g., NCMC [104] or NCM forests [124]. If the deep network has actually achieved the goal of view independent recognition, then the distribution of penultimate activation should be nearly view independent. While the open-jaw and side views of a shark are visually quite different, and a multi-exemplar model may be more effective in capturing the different features in different views, the open-jaws of different sharks are still quite similar, as are their side views. Hence, each view may present a relatively consistent AV, allowing a single MAV to capture both. Intuitively, while image features may vary greatly with view, the relative strength of "related classes" represented by the AV should be far more view independent.

3.2.3 OpenMax

The standard SoftMax function is a gradient-log-normalizer of the categorical probability distribution – a primary reason that it is commonly used as the last fully connected layer of a network. The traditional definition has per-node weights in their computation. The scores in the penultimate network layer of Caffe-based deep networks [68], what we call the activation vector, has the weighting performed in the convolution that produced it. Let $\mathbf{v}(\mathbf{x}) = v_1(x), \ldots, v_N(x)$ be the activation level for each class, $y = 1, \ldots, N$. After deep network training, an input image $\mathbf{x}$ yields activation vector $\mathbf{v}(\mathbf{x})$, the SoftMax layer computes:

$$P(y = j|\mathbf{x}) = \frac{e^{\mathbf{v_j}(\mathbf{x})}}{\sum_{i=1}^{N} e^{\mathbf{v_i}(\mathbf{x})}} \tag{3.1}$$

where the denominator sums over all classes to ensure the probabilities over all classes sum to 1. However, in open set recognition there are unknown classes that will occur at test time and, hence, it is not appropriate to require the probabilities to sum to 1.

To adapt SoftMax for open set, let ρ be a vector of meta-recognition models for each class estimated by Alg. 2. In Alg. 3 we summarize the steps for OpenMax computation. For convenience we define the *unknown unknown* class to be at index 0. We use the Weibull CDF probability (line 3 of Alg. 3) on the distance between $\mathbf{x}$ and μ_i for the core of the rejection estimation. The model μ_i is computed using the images associated with category i, images that were classified correctly (top-1) during training process. We expect the EVT function of distance to provide a meaningful probability only for few top ranks. Thus in line 3 of Alg. 3, we compute weights for the α largest activation classes and use it to scale the Weibull CDF probability. We then compute revised activation vector with the top scores changed. We compute a pseudo-activation for the unknown unknown class, keeping the total activation level constant. Including the unknown unknown class,

Algorithm 3 OpenMax probability estimation with rejection of unknown or uncertain inputs.

```
Require: Activation vector for $\mathbf{v}(\mathbf{x}) = v_1(x), \ldots, v_N(x)$
Require: means $\mu_j$ and libMR models $\rho_j = (\tau_i, \lambda_i, \kappa_i)$
Require: $\alpha$, the numer of "top" classes to revise
1: Let $s(i) = \mathrm{argsort}(v_j(x))$; Let $\omega_j = 1$
2: for $i = 1, \ldots, \alpha$ do
3:     $\omega_{s(i)}(x) = 1 - \frac{\alpha-i}{\alpha} e^{-\left(\frac{\|x-\tau_{s(i)}\|}{\lambda_{s(i)}}\right)^{\kappa_{s(i)}}}$
4: end for
5: Revise activation vector $\hat{v}(x) = \mathbf{v}(\mathbf{x}) \circ \omega(\mathbf{x})$
6: Define $\hat{v}_0(x) = \sum_i v_i(x)(1 - \omega_i(x))$.
7:
       $\hat{P}(y = j|\mathbf{x}) = \frac{e^{\hat{\mathbf{v}}_{\mathbf{j}}(\mathbf{x})}}{\sum_{i=0}^{N} e^{\hat{\mathbf{v}}_{\mathbf{i}}(\mathbf{x})}}$    (3.2)
8: Let $y^* = \mathrm{argmax}_j\, P(y = j|\mathbf{x})$
9: Reject input if $y^* == 0$ or $P(y = y^*|\mathbf{x}) < \epsilon$
```

the new revised activation compute the OpenMax probabilities as in Eq. 3.2.

OpenMax provides probabilities that support explicit rejection when the unknown unknown class ($y = 0$) has the largest probability. This Meta-Recognition approach is a first step toward determination of unknown unknown classes and our experiments show that a single MAV works reasonably well at detecting fooling images, and is better than just thresholding on uncertainty. However, in any system that produces certainty estimates, thresholding on uncertainty is still a valid type of meta-recognition and should not be ignored. The final OpenMax approach thus also rejects unknown as well as uncertain inputs in line 9 of Alg.3.

To select the hyper-parameters ϵ, η, and α, we can do a grid search calibration procedure using a set of training images plus a sampling of open set images, optimizing F-measure over the set. The goal here is basic calibration for overall scale/sensitivity selection, not to optimize the threshold over the space of unknown unknowns, which cannot be done experimentally.

Note that the computation of the unknown unknown class probability inherently alters all probabilities estimated. For a fixed threshold and inputs that have even a small chance of being unknown, OpenMax will reject more inputs than SoftMax. Fig. 3.2 shows the OpenMax and SoftMax probabilities for 100 example images, 50 training images and 50 open set images as well as for fooling images. The more off-diagonal the more OpenMax altered the probabilities. Threshold selection for uncertainty based rejection ϵ, would find a balance between keeping the training examples while rejecting open set examples. Fooling images were not used for threshold selection.

While not part of our experimental evaluation, note that OpenMax also provides meaningful rank ordering via its estimated probability. Thus OpenMax directly supports a top-5 class output with rejection. It is also important to note that because of the re-calibration of the activation scores $\hat{v}_i(x)$, OpenMax often does not produce the same rank ordering of the scores.

3.2.4 OpenMax Compact Abating Property

While thresholding uncertainty does provide the ability to reject some inputs, it has not been shown to formally limit open space risk for deep networks. It should be easy to see that in terms of the activation vector, the positively labeled space for SoftMax is not restricted to be near the training space, since any increase in the maximum class score increases its probability while decreasing the probability of other classes. With sufficient increase in the maximum directions, even large changes in other dimension will still provide large activation for the leading class. While in theory one might say the deep network activations are bounded, the fooling images of [107], are convincing evidence that SoftMax cannot manage open space risk.

Theorem 3 (Open Set Deep Networks)**.** *A deep network extended using Meta-Recognition on activation vectors as in Alg. 3, with the SoftMax later adapted to OpenMax, as in Eq. 3.2, provides an open set recognition function.*

Proof. The Meta-Recognition probability (CDF of a Weibull) is a monotonically increasing function of $\|\mu_i - x\|$, and hence $1 - \omega_i(x)$ is monotonically decreasing. Thus, they form the basis for a compact abating probability as defined in [139]. Since the OpenMax transformation is a weighted monotonic transformation of the Meta-Recognition probability, applying Theorems 1 and 2 of [8] yield that thresholding the OpenMax probability of the unknown manages open space risk as measured in the AV feature space. Thus it is an open set recognition function. □

3.3 Experimental Analysis

In this section, we present experiments carried out in order to evaluate the effectiveness of the proposed OpenMax approach for open set recognition tasks with deep neural networks. Our evaluation is based on ImageNet Large Scale Visual Recognition Competition (ILSVRC) 2012 dataset with 1K visual categories. The dataset contains around 1.3M images for training (with approximately 1K to 1.3K images per category), 50K images for validation and 150K images for testing. Since test labels for ILSVRC 2012 are not publicly available, like others have done we report performance on validation set [76, 107, 150]. We use a pre-trained AlexNet (BVLC AlexNet) deep neural network provided by the Caffe software package [68]. BVLC AlexNet is reported to obtain approximately 57.1% top-1 accuracy on ILSVRC 2012 validation set. The choice of pre-trained BVLC AlexNet is deliberate, since it is open source and one of the most widely used packages available for deep learning.

To ensure proper open set evaluation, we apply a test protocol similar to the ones presented in [139, 8]. During the testing phase, we test the system with all the 1000 categories from ILSVRC 2012 validation set, fooling categories and previously unseen categories. The previously unseen categories are selected from ILSVRC 2010. It has been noted by Ruskovsky *et al.* [128] that approximately 360 categories from ILSVRC 2010 were discarded and not used in ILSVRC 2012. Images from these 360 categories as the *open set* images, i.e., unseen or unknown categories.

Fooling images are generally totally unrecognizable to humans as belonging to the given category but deep networks report with near certainty they are from the specified category. We use fooling images provided by Nguyen *et al.* [107] that were generated by an evolutionary algorithm or by gradient ascent in pixel space. The final test set consists of 50K closed set images from ILSVRC 2012, 15K open set images (from the 360 distinct categories from ILSVRC 2010) and 15K fooling images (with 15 images each per ILSVRC 2012 categories).

Training Phase: As discussed previously (Alg. 2), we consider the penultimate layer (fully connected layer 8 , i.e., *FC8*) for computation of mean activation vectors (MAV). The MAV vector is computed for each class by considering the training examples that deep networks training classified correctly for the respective class. MAV is computed for each crop/channel separately. Distance between each correctly classified training example and MAV for particular class is computed to obtain class specific distance distribution. For these experiments we use a distance that is a weighted combination of normalized Euclidean and cosine distances. Supplemental material shows results with pure Euclidean and other measures that overall perform similarly. Parameters of Weibull distribution are estimated on these distances. This process is repeated for each of the 1000 classes in ILSVRC 2012. The exact length of tail size for estimating parameters of Weibull distribution is obtained during parameter estimation phase over a small set of hold out data. This process is repeated multiple times to obtain an overall tail size of 20.

Testing Phase: During testing, each test image goes through the OpenMax score calibration process as discussed previously in Alg. 3. The activation vectors are the values in the *FC8* layer for a test image that consists of 1000x10 dimensional values corresponding to each class and each channel. For each channel in each class, the input is compared using a per class MAV and per class Weibull parameters. During testing, distance with respect to the MAV is computed and revised OpenMax activations are obtained, including the new unknown class (see lines 5&6 of Alg. 3). The OpenMax probability is computed per channel, using the revised activations (Eq. 3.2) yielding an output of 1001x10 probabilities. For each class, the average over the 10 channel gives the overall OpenMax probability. Finally, the class with the maximum over the 1001 probabilities is the predicted class. This maximum probability is then subject to the uncertainty threshold (line 9). In this work we focus on strict top-1 predictions.

Evaluation: ILSVRC 2012 is a large scale multi-class classification problem and top-1 or top-5 accuracy is used to measure the effectiveness of a classification algorithm [128]. Multi-class classification error for a closed set system can be computed by keeping track of incorrect classifications. For open set testing the evaluation must keep track of the errors that occur due to standard multi-class classification over known categories as well as errors between known and unknown categories. As suggested in [155, 138] we use F-measure to evaluate open set performance. For open set recognition testing, F-measure is better than accuracy because it is not inflated by true negatives.

For a given threshold on OpenMax/SoftMax probability values, we compute true positives, false positives and false negatives over the entire dataset. For example, when testing the system with images from validation set, fooling set and open set (see Fig. 3.3), true positives are defined as the correct classifications on the validation set, false positives are incorrect classifications on the validation set and false negatives are images from the fooling set and open set categories that the system incorrectly classified as known examples. Fig. 3.3 shows performance of OpenMax and SoftMax for varying thresholds. Our experiments show that the proposed approach of OpenMax consistently obtains higher F-measure on open set testing.

3.4 Discussion

We have seen that with our OpenMax architecture, we can automatically reject many unknown open set and fooling images as well as rejecting some adversarial images, while having only modest impact to the true classification rate. One of the obvious questions when using Meta-Recognition is "what do we do with rejected inputs?" While that is best left up to the operational system designer, there are multiple possibilities. OpenMax can be treated as a novelty detector in the scenario presented open world recognition [8] after that human label the data and the system incrementally learn new categories. Or detection can used as a flag to bring in other modalities[154, 48].

A second approach, especially helpful with adversarial or noisy images, is to try to remove small noise that might have lead to the miss classification. For example, the bottom right of Fig. 4.1, showed an adversarial image wherein a hammerhead shark image with noise was incorrectly classified by base deep network as a scuba diver. OpenMax Rejects the input, but with a small amount of simple gaussian blur, the image can be reprocessed and is accepted as a hammerhead shark by with probability 0.79.

We used non-test data for parameter tuning, and for brevity only showed performance variation with respect to the uncertainty threshold shared by both SoftMax with threshold and OpenMax. The supplemental material shows variation of a wider range of OpenMax parameters, e.g. one can increase open set and fooling rejection capability at the expense of rejecting more of the true classes. In future work, such increase in true class rejection might be mitigated by increasing the expressiveness of the AV model, e.g. moving to multiple MAVs per class. This might allow it to better capture different contexts for the same object, e.g. a baseball on a desk has a different context, hence, may have different "related" classes in the AV than say a baseball being thrown by a pitcher.

Interestingly, we have observe that the OpenMax rejection process often identifies/rejects the ImageNet images that the deep network incorrectly classified, especially images with multiple objects. Similarly, many samples that are far away from training data have multiple objects in the scene. Thus, other uses of the OpenMax rejection can be to improve training process and aid in developing better localization techniques [167, 110]. See Fig. 3.5 for an example.

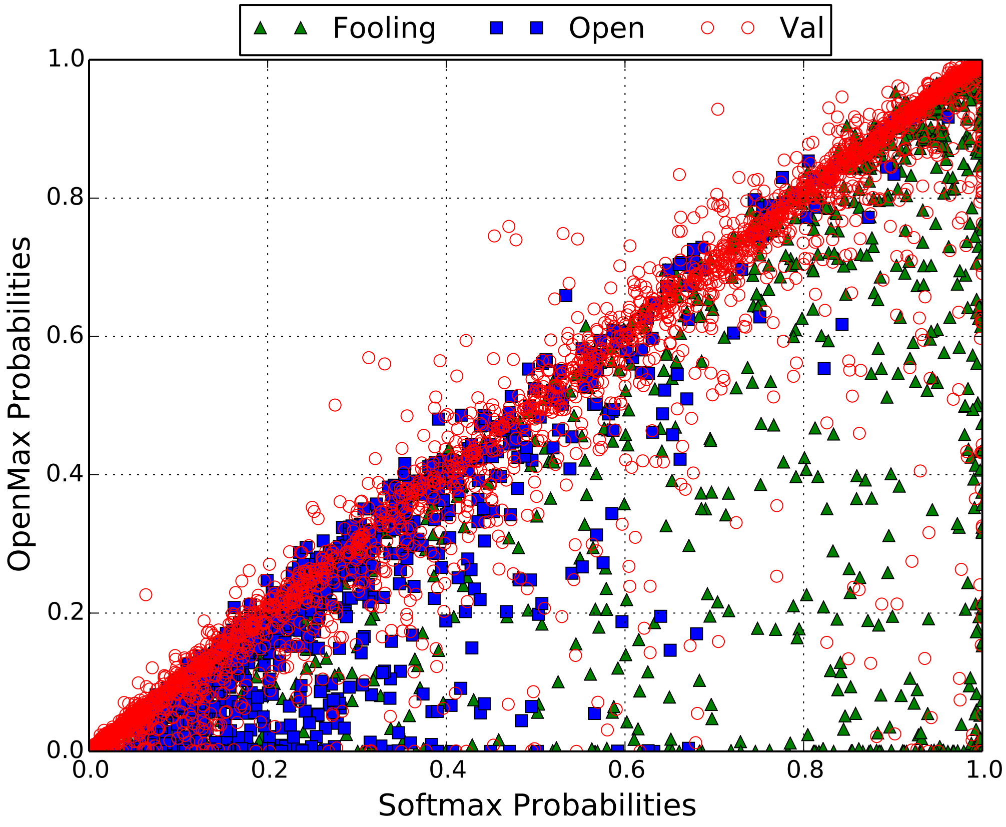


Figure 3.2: A plot of OpenMax probabilities vs SoftMax probabilities for the fooling (triangle), open set (square) and validation (circle) for 100 categories from ImageNet 2012. The more off-diagonal a point, the more OpenMax altered the probabilities. Below the diagonal means OpenMax estimation reduced the inputs probability of being in the class. For some inputs OpenMax increased the classes probability, which occurs when the leading class is partially rejected thereby reducing its probability and increasing a second or higher ranked class. Uncertainty-based rejection threshold (ϵ) selection can optimize F-measure between correctly classifying the training examples while rejecting open set examples. (Fooling images are not used for threshold selection.) The number of triangles and squares below the diagonal means that uncertainty thresholding on OpenMax threshold (vertical direction), is better than thresholding on SoftMax (horizontal direction).

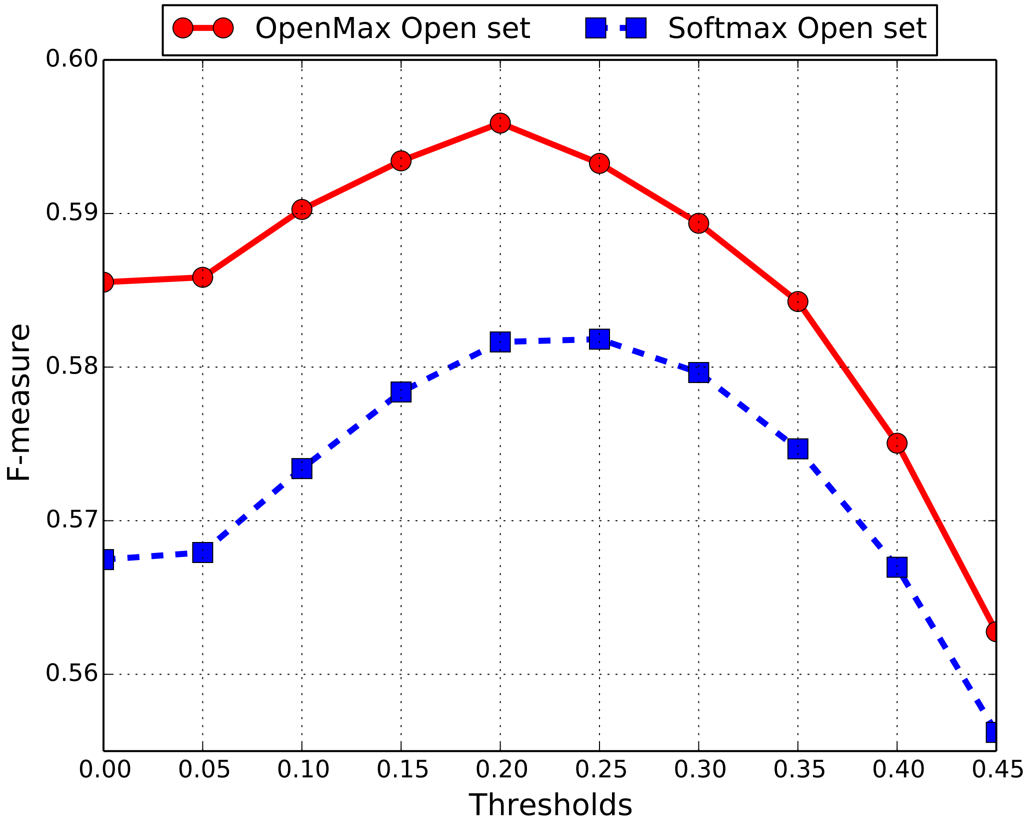


Figure 3.3: OpenMax and SoftMax-w/threshold performance shown as F-measure as a function of threshold on output probabilities. The test uses 80,000 images, with 50,000 validation images from ILSVRC 2012, 15,000 fooling images and 15,000 "unknown" images draw from ILSVRC 2010 categories not used in 2012. The base deep network performance would be the same as threshold 0 of SoftMax-w/threshold. OpenMax performance gain is nearly 4.3% improvement accuracy over SoftMax with optimal threshold, and 12.3% over the base deep network. Putting that in context, over the test set OpemMax correctly classified 3450 more images than SoftMax and 9847 more than the base deep network.

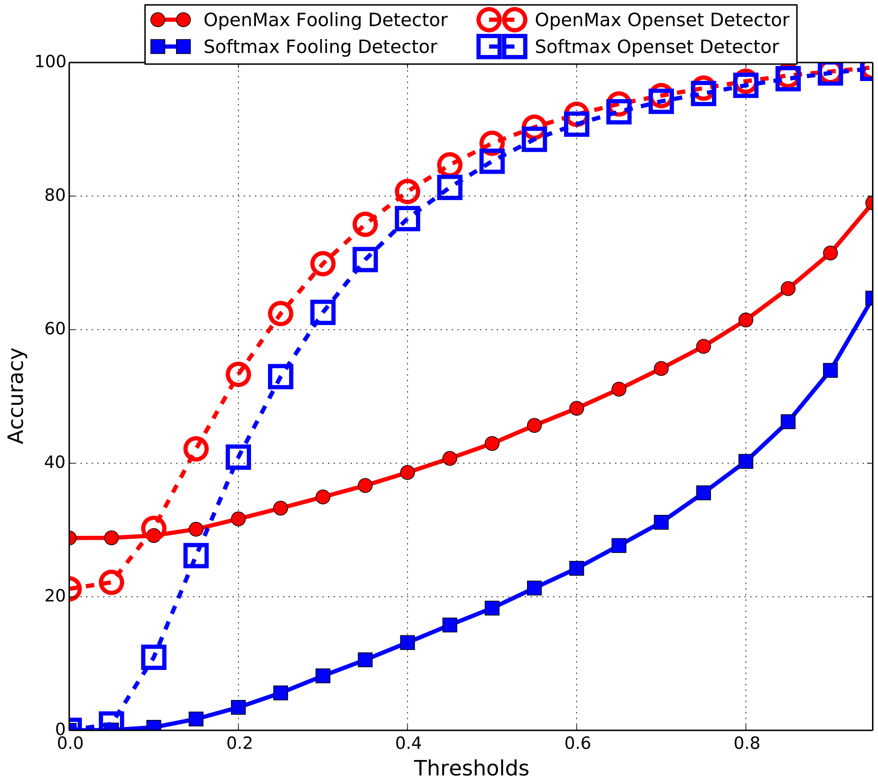


Figure 3.4: The above figure shows performance of OpenMax and SoftMax as a detector for fooling images and for open set test images. F-measure is computed for varying thresholds on OpenMax and SoftMax probability values. The proposed approach of OpenMax performs very well for rejecting fooling images during prediction phase.

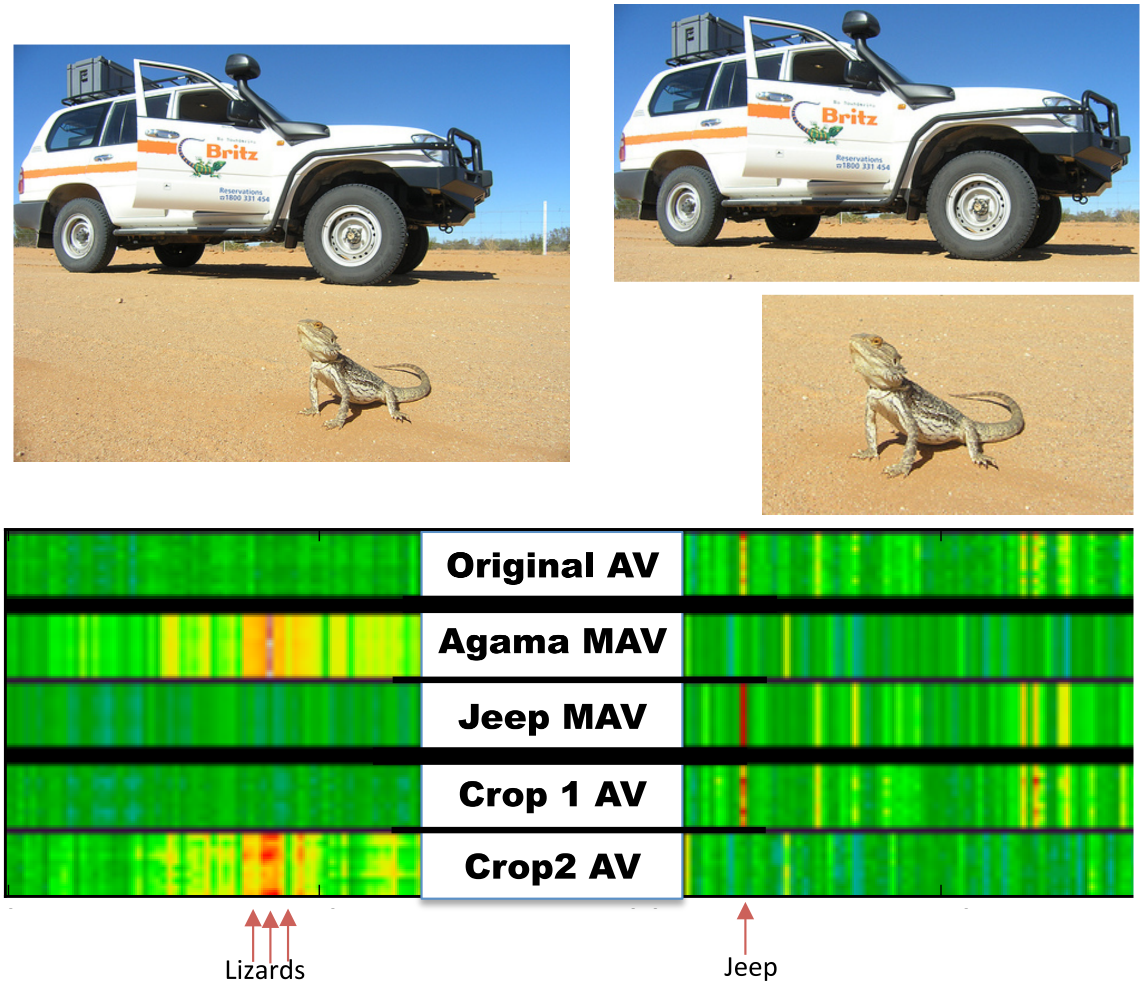


Figure 3.5: OpenMax also predict failure during training as in this example. The official class is agama but the MAV for agama is rejected for this input, and the highest scoring class is jeep with probability 0.26. However, cropping out image regions can find windows where the agama is well detected and another where the Jeep is detected. Crop 1 is the jeep region, crop 2 is agama and the crops AV clearly match the appropriate model and are accepted with probability 0.32 and 0.21 respectively.

Chapter 4

Streaming Face Recognition

The performance of a face recognition system relies heavily on its ability to handle spatial, temporal and operational variances [184]. Large scale face recognition systems have to adapt with changing environments. Face recognition systems often achieve excellent performance on benchmark datasets, but performance degrades significantly in operational environments [118]. One could build an application specific dataset, but this process is extremely cumbersome. Recently [112], [176] have adopted the approach to incrementally adapt learned models for face recognition with new data. As hardware (e.g. surveillance cameras) becomes cheaper, it is easier to capture more data to address problems such as self-occlusion, motion blur and illumination [55]. A typical image captured from a surveillance dataset is about 70 KB [14]. Analyzing a short video stream of about 10 minutes at 30fps amounts to processing of 1.2 GB of data. Updating existing learned models with such large and rapidly produced data poses significant challenge for a system with limited processing resources. While processing data on the cloud might be possible for some applications, many surveillance applications have constrained operational environments.

A common notion in many learning based vision systems is to learn with as much data as possible, to achieve best possible prediction performance during testing phase. This approach is useful for large scale face recognition when resources permit [175], [10], however poses multiple challenges when learning with limited resources [70]. The feature descriptors used for image representation are often high dimensional and data is generated faster than it can be processed/learned. Thus from practical application point of view it

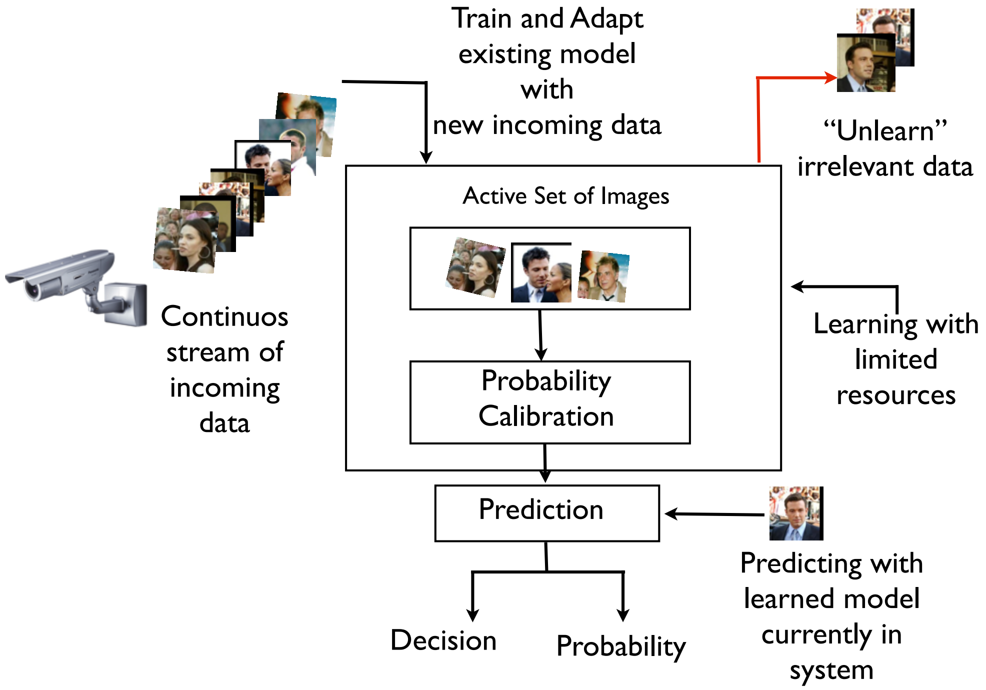

Train and Adapt existing model with new incoming data
"Unlearn" irrelevant data
Active Set of Images
Continuos stream of incoming data
Probability Calibration
Learning with limited resources
Prediction
Predicting with learned model currently in system
Decision
Probability

Figure 4.1: In streaming face recognition with limited resources, updating the existing system with incoming data, gradually adapting to variations and unlearning already learned samples, without compromising on accuracy can be extremely challenging. We present a system that can incrementally adapt to incoming samples and provide reliable posterior probability estimates.

becomes important to handle not just every incoming sample, but all the "important" samples. This problem is further constrained in case of standalone or portable face recognition systems [100]. Problem of online learning occurs in many consumer applications as well [71]. Updating the existing face recognition system with incoming data, gradually adapting to variations and possibly forgetting already learned samples can be extremely challenging. In this work, we study online learning on a fixed budget in the context of unconstrained face recognition. We consider a specific operational scenario where system is presented one sample at a time with a user-defined upper limit on maximum number of samples (budget size) it can retain in the memory.

We propose an incremental and online Support Vector Machine (SVM) learning system on a budget. Our system can process one example at a time and is based on work of Cauwenberghs et al [21]. This method maintains optimal SVM solution on all previously seen examples during training by incrementally updating

Karush-Kuhn-Tucker (KKT) conditions. The system learns incoming data one sample at a time (Fig 4.1) until it reaches the maximum allowable budget size. In context of SVMs, the budget size implies maximum number of support vectors retained by the system. This set of support vectors is termed as active set. Once this size is reached, existing samples are incrementally removed from the active set. This process is called "unlearning".

In many face recognition applications, it is important to predict well calibrated probabilities [108], [136], [134] along with SVM decision scores (or class predictions). The problem of probability calibration is more pronounced for budgeted online learning, since calibration data is limited. Further, the calibration data changes regularly when model gets updated. While one can always re-calibrate with data present in the active set at a given time, a counter question about reliability of the calibration arises. Scheirer *et al.* [136] have demonstrated that the broader problem of recognition is consistent with the assumptions of statistical Extreme Value Theory (EVT). They show EVT provides a way to determine probabilities, regardless of the overall distribution of the data. They have shown the extremes or tail of a score distribution produced by a recognition/classification algorithm can always be modeled by an Extreme Value Theory (EVT) distribution. This distribution is shown to produce a reverse Weibull distribution when the data is bounded. This observation makes EVT a natural choice for probability calibration for budgeted online SVMs.

In the following sections, we discuss our modifications to the approach of [21] to make it suitable for budgeted online learning. We show via extensive experiments that our method works comparable (and often better) at really small budget sizes when compared with many off-the-shelf machine learning packages [41]. We develop EVT based calibration models for online budgeted learning for posterior probability estimation. We compare our method with a de-facto standard in the community proposed by Platt [119]. We perform rigorous comparison of the proposed probability calibration techniques for extreme budget sizes to assess the reliability of estimated probabilities. Finally, we quantify our results by methods inspired from meteorology: reliability plots [174] and Brier Score analysis [93]. Our study suggests EVT calibration is well suited for online learning applications as it consistently yields more reliable probabilities. We test our methods on Labeled Faces in the Wild [59] dataset and show suitability of the proposed approach for large scale face verification/recognition. The contributions of this work can be summarized as follows:

1. A probability based budgeted incremental support vector learning and unlearning method.

2. A novel posterior probability estimation model based on EVT.

3. Analysis of posterior probability estimation models with limited calibration data.

4. Reliability analysis of various probability estimation techniques.

4.1 Related Work

Ozawa *et al.* [112] use incremental principal component analysis with resource allocating network with long term memory for constrained face recognition problem. Yan *et al.* [176] used incremental linear discriminant analysis with spectral regression. Their approach is suitable for incremental model adaptation but does not provide posterior probability estimates, as required in multiple applications.

Incremental Learning described in this work draws inspiration from work done by Cauwenberghs et al [21]. It maintains optimal SVM solution on all previously seen examples during training by incrementally updating Karush-Kuhn-Tucker (KKT) conditions. Wang et al [170] provide a thorough analysis of various budgeted online learning techniques in their work. In section 4.5, we compare our method with the ones mentioned in [170].

Posterior probability estimation from SVM decision scores is a well studied problem in computer vision and machine learning. Platt [119] proposed probability calibration for SVMs, which has also been applied to other learning algorithms. A comprehensive analysis of probability calibration techniques can be found in work of [108]. These methods were devised for batch supervised learning. They have been found to be effective when the entire training set is available for calibration. Calibration with limited data is a challenging problem as noted by Zadrozny et al [181] and [108]. Niculescu-Mizil et al [108] found that isotonic regression based calibration is prone to over-fitting. It performs worse than Platt scaling, when data is limited (less than 1000 samples). Majority of the online learning software packages [114], [143], [41] provide either a default Platt Scaling for posterior probability estimation or just decision score as output.

Recent work by Scheirer et al [136] has shown the extremes or *tail* of a score distribution produced by any recognition algorithm can always be modeled by an EVT distribution. These approaches were found to be useful for attribute fusion in image retrieval applications [134], scene classification in remote sensing applications [147] and biometric verification systems [135]. Probability calibration for budgeted online learning for SVMs presented in this work builds on top of the work of Scheirer *et al.* [136]. Reliability diagrams [174] are frequently used for assessing reliability of probability forecasts for binary events such as the probability of measurable precipitation in the area of weather forecasting. We introduce the tools such as reliability diagrams and Brier Scores [93] to assess the reliability of posterior probability estimation obtained by Platt calibration and EVT based calibration.

4.2 Incremental Support Vector Machines

Let us assume we have a set of training data $D = \{(x_i, y_i)\}_{i=1}^{k}$, where $x_i \in \mathcal{X} \subseteq \mathcal{R}^n$ is input and $y_i \in \{+1, -1\}$ is the output class label. Support Vector Machines learn the function $f(x) = w^T \phi(x) + b$, where $\phi(x)$ denotes a fixed feature space transformation. The dual formulation of this problem involves estimation of α_i, where α are the Lagrange multipliers associated with the constraints of the primal SVM problem. These coefficients are obtained by minimizing a convex quadratic objective function under the constraints

$$\min_{0 \leq \alpha_i \leq C} : W = \frac{1}{2} \sum_{i,j} \alpha_i Q_{ij} \alpha_j - \sum_i \alpha_i + \sum_i y_i \alpha_i \tag{4.1}$$

where b is the bias (offset), Q_{ij} is the symmetric positive definite kernel matrix $Q_{ij} = y_i y_j K(x_i, x_j)$ and C is the nonnegative user-specified slack parameter that balances model complexity and loss of training data. The first order conditions on W reduce to the KKT conditions, from which following relationships are obtained:

$$y_i f(x_i) > 1 \Rightarrow \alpha_i = 0 \tag{4.2}$$

$$y_i f(x_i) = 1 \Rightarrow \alpha_i \in [0, C] \tag{4.3}$$

$$y_i f(x_i) < 1 \Rightarrow \alpha_i = C \tag{4.4}$$

and $\sum_{i=1}^{k} y_i \alpha_i = 0$. These conditions partition the training data into three discrete sets: margin support vectors ($\alpha_i \in [0, C]$), error support vectors ($\alpha_i = C$) and ignored vectors. Decision score for test sample x_t is obtained using $f(x) = w^T\phi(x) + b$ where

$$w = \sum_{i=1}^{l} y_i \alpha_i \phi(x_i) \tag{4.5}$$

and l is total number of support vectors (consisting of margin support vectors and error support vectors). This is the traditional batch learning problem for SVM [164].

The incremental extension for SVM was suggested by Cauwenberghs *et al.* [21]. In this method, the KKT conditions are preserved after each training iteration. For incremental training, when a new training sample (x_c, y_c) is presented to the system, the Lagrangian coefficients (α_c) corresponding to this sample and positive definite matrix Q_{ij} from the SVM currently in the memory undergo a small change Δ to ensure maintenance of optimal KKT condition (details of these increments can be found in [83], [21]).

4.3 SVM for Streaming Face Recognition

In this section, we describe how we extend the incremental SVM for streaming face recognition application. Throughout the course of this work, we consider the case of binary classification. As noted earlier, operational face recognition systems have to learn from a continuous stream of data. The incoming data is processed continuously till the user prescribed budget size B (i.e. Active Set) is reached. At every stage, the classifier is updated by methodology described in section 4.2. Once the prescribed budget size is reached (e.g. if the system runs out of memory), the process of "unlearning" starts. Two system design questions at this stage are: (i) How to select a sample from current active set to unlearn? (ii) How to update existing SVM solution?

The process of unlearning starts with determining a particular sample to unlearn. In the past, machine learning researchers have considered methods like randomly removing support vector [22], removing the oldest support vector [36] or removing support vector that yields smallest prediction error using leave one out error methodology [21]. Removing oldest sample from memory have found to be useful for applications that exploit temporal coherence [4] (eg. tracking). A related study in this area [170] analyzed multiple budget maintenance strategies. They concluded that removing random SV (support vector) or removing oldest SV from the current active set yields poor performance. They suggested removing support vector with smallest norm with respect to current SVM solution (a comparison with this method is shown in Fig 4.4). We consider an alternative budget maintenance strategy for our work.

When the prescribed limit is reached, the training samples currently in the active set are used to calibrate probabilities based on Platt scaling [119]. For two class classification, a probability of 0.5 determines random chance (before threshold estimation process). The most extreme probability is obtained by the equation $max(abs \mid 0.5 - p(f(x_i)) \mid_{i=1}^{l})$, where l is the total number of samples in active set, $p()$ is calibrated probability and $f()$ is the current SVM solution in memory. The corresponding sample is determined by the system as the sample to unlearn. Once the training sample to unlearn is determined, it is incrementally unlearned from the existing model. This process is combined as part of the entire update process, thus at each update stage probabilities are calibrated using the training data currently in the active set. There are a number of reasons for using calibrated probabilities to determine sample to unlearn. If samples closest to decision boundary are removed, system becomes less robust at handling noisy data. Sigmoid nature of Platt scaling helps avoid these issues [11]. As we compute absolute distance from the mean, even in case of imbalanced data, the sample that is most probable to be included in a particular class is removed. This process helps the system to focus on samples around the decision boundary: an area considered to be most informative for discriminative learning methods [164]

To answer the second question (how to update existing SVM solution), we make a key observation in the method of [21]. Cauwenberghs *et al.* [21] note that update process is reversible: thus when a sample is to be removed from the system, the Lagrangian corresponding to the training sample is assigned to zero. The matrix Q_{ij} is decremented to maintain KKT condition. After every incremental/decremental stage, these optimality conditions (i.e. α_i, Q_{ij}) are saved as the part of the learned model. Although this process adds a small overhead on disk space it guarantees an optimal solution on previously seen data.

In sec 4.5 we compare the proposed SVM based unlearning method with budgeted stochastic gradient descent method proposed by Wang *et al.* [170]. The method maintains a fixed number of support vectors in the model and incrementally updates them during stochastic gradient descent (SGD) training. Budget maintenance is achieved by removal of support vector with smallest norm. The goal is to minimize degradation of weight vector Δ_i after removal of support vector (here Δ_i is the degradation obtained by removing ith support vector from the system).

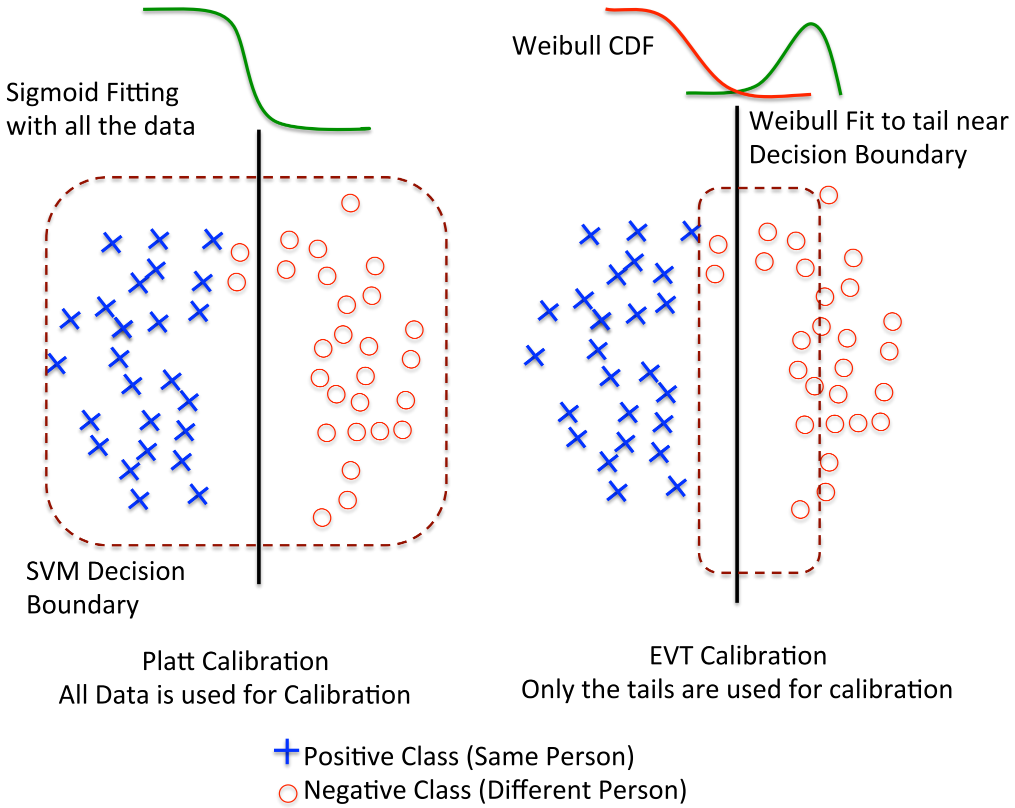

Sigmoid Fitting
with all the data
SVM Decision
Boundary
Platt Calibration
All Data is used for Calibration
Weibull CDF
Weibull Fit to tail near
Decision Boundary
EVT Calibration
Only the tails are used for calibration
+ Positive Class (Same Person)
○ Negative Class (Different Person)

Figure 4.2: Platt posterior probability estimation [119] method uses all the available training data for calibrating the model. EVT based posterior probability estimation [136] uses the tail of the data near decision boundary for calibration. In streaming face recognition applications when limited data is available for calibration, EVT based methods yield robust posterior probability estimations

4.4 Probability calibration for Streaming Face Recognition

4.4.1 Sigmoid Based Calibration

Many computer vision applications require prediction of posterior class probability $p(y = 1|x)$ [69], [134]. Probability calibration method suggested by Platt [119] is the most commonly adopted method for many machine learning tasks. The method proposes approximating the posterior by a sigmoid function. The parameters for calibration are estimated using the entire training set. Maximum likelihood estimation is used to solve for the parameters A and B. When a test sample x_i is to be tested with respect to a learned model

$f(x)$, the posterior probability is given by

$$p(y = 1|x) = \frac{1}{1 + exp(Af(x) + B)} \tag{4.6}$$

In our experiments, we use a version of Platt's method modified by [91] to avoid issues related to numerical difficulties.

4.4.2 EVT based Calibration

For streaming face recognition the data available in the active set is constantly changing as the system learns from new incoming samples and unlearns existing samples (see Sec 4.3). This implies data available for calibration purpose for posterior probability estimation changes when the model undergoes an update process. Niculescu-Mizil *et al.* [108] carried out extensive experiments with various learning methods and calibration methods. They noted that posterior probability estimation was more reliable for methods like Platt [119] and isotonic regression when large number of training samples were available for calibration. *For streaming face recognition applications, when data is scarce for smaller budget sizes, how does one obtain reliable posterior probability estimation ?*

In this work, we build on top of a probability calibration model based on Extreme Value Theory for SVMs first proposed by [136], [134]. They noted that the general recognition problem itself is consistent with the assumptions of statistical extreme value theory (EVT), which provides a way to determine probabilities, regardless of the overall distribution of the data. The extremes or tail of a score distribution produced by any recognition algorithm can always be modeled by an EVT distribution, which is a reverse Weibull if the data are bounded. Fig 4.2 shows the difference between Platt calibration and EVT based calibration. The figure illustrates a toy scenario for a batch learning application. For batch learning, when all the data is available for calibration, Platt calibration methods uses entire data for building posterior probability estimation model. For EVT based calibration, only the tail of the data is used for building estimation model. We use this key observation to build a posterior probability estimation model based on EVT for streaming face recognition.

Given a SVM decision function $f(x_i)$, and a test sample x_i, we have two independent estimates for posterior probability estimation $P(c \mid f(x_i))$, where c is the class under consideration. The first P_η is based on Weibull cumulative distribution function (CDF) derived from positive class (match $\mathcal{M}$) data. The second,

P_ψ, is based on reverse Weibull CDF from negative (non-match $\mathcal{N}$) estimate, which is equivalent to rejecting the Weibull fitting on 1-vs-all negative (non-match) data. We consider the case of $P_\eta(c \mid f(x_i))$ for our experiments. The Weibull distribution has 3 parameters: scale λ_c, shape κ_c and location τ_c and is given by:

$$W_c(z) = \frac{\tau_c}{\lambda_c}(\frac{z-\tau_c}{\lambda_c})^{\kappa_c - 1} e^{(\frac{z-\tau_c}{\lambda_c})^{\kappa_c}} \tag{4.7}$$

with $z > \tau_c$, $\lambda_c > 0$ and $\kappa_c > 0$. For this work we use LibMR provided by [136], which uses maximum likelihood estimate to estimate the Weibull parameters. To estimate the probability at any point x_i belonging to a class c, we use CDF derived from $\mathcal{M}$, given below:

$$P_\eta(c \mid f(x_i)) = 1 - e^{-(\frac{-f(x_i)-\tau_\eta}{\lambda_\eta})^{\kappa_\eta}} \tag{4.8}$$

It is interesting to note that Platt's [119] original observation *The class-conditional densities between the margins are exponential*, is roughly consistent with the Weibull-based CDFs in eqn 4.7 . The difference is that the Weibull CDF, while generally exponential, 1) has more parameters, 2) will have different parameter fits for the positive and negative classes using only positive or negative data respectively. In addition, EVT allows us to use a very different fitting process using only the extrema, which is more suitable when calibrating with limited data. Eqn 4.8 models the likelihood of the sample x_i being from match distribution ($\mathcal{M}$).

The overall approach can be summarized as follows:

1. Step 1: Incrementally learning all incoming samples till predefined budget size B is reached
2. Step 2: Once budget size B is reached
   (a) Calibrate data currently in buffer Platt probabilities. Get sample with probability farthest from random chance ($max||0.5 - p(x_i)||$).
   (b) Incrementally unlearn sample with max difference from 0.5 from system
3. Step 3: Incrementally learn incoming training sample. Go to step 2.
4. Step 4: Calibrate probabilities with samples currently in the system. Obtain calibration parameters (3 parameters $z > \tau_c$, $\lambda_c > 0$ and $\kappa_c > 0$ for EVT calibration and 2 parameters A, B for Platt calibration)
5. At any stage, perform prediction with model currently in the system. Obtain probabilities based on calibration parameters wrt model currently in the system.

4.5 Experiments

Figure 4.3: The figure shows examples of images from Labeled Faces in the Wild [59] dataset. The images considered in this dataset are taken in unconstrained settings with no control on pose, illumination, gender or context. The dataset contains images of about 5749 individuals with a total of 13233 images.

Dataset: In this section, we discuss our experiments on the problem of face verification in the wild. In face verification, given two face images, goal is to predict whether the images are from the same person. For evaluation, we used Labeled Faces in the Wild (LFW) [59] dataset which contains 13233 images of 5749 individuals, developed for face verification. View 1 of LFW is used for building models, feature selection and finding optimal operating parameters. View 2 consists of 6000 pairs of face images on which performance is to be reported. The 6000 image pairs are divided into 10 sets to allow 10-fold cross-validation. Overall classification performance is reported on "View 2" of the dataset. Fig 4.3 shows examples of images from LFW dataset.

Features: We use classifier scores obtained from attribute classification method described in [77]. Kumar *et al.* [77] compute visual describable visual attributes for face verification problem. Describable visual attributes are labels given assigned to an image to describe any aspect of its appearance (e.g. gender, hair color, ethnicity: Asian, ethnicity: European etc.). Kumar et al, compute attributes from each pair of images in LFW dataset. Image-pairs can be effectively verified based on presence/absence confidence on variety of these attributes. The total number of features per image used were 146.

Protocol: We follow the protocol as proposed for LFW dataset with minor modification. Training samples are incrementally added to the system, one sample at a time. A budget size is specified for each iteration. Once the prescribed budget size is reached, samples are incrementally unlearned from the system by the method described in Sec 4.3. For e.g. in a typical run, the system learns with 5400 training samples incrementally. At

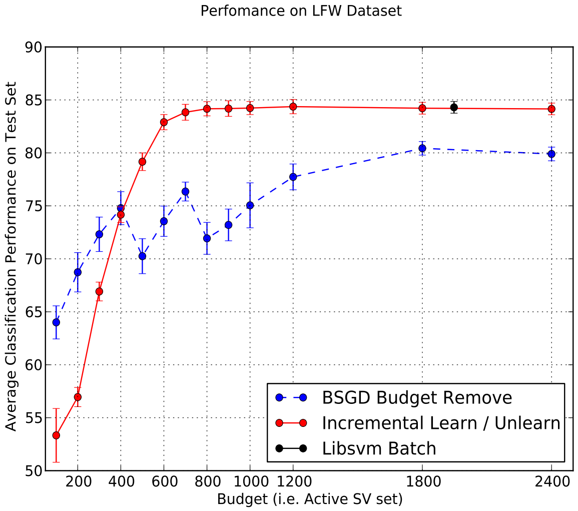


Figure 4.4: Performance of a leading budgeted stochastic gradient descent method with removal strategy for budget maintenance compared with incremental learning method proposed in this work on LFW face dataset. For each method, average and standard over 10 splits of LFW is shown. Performance obtained with LIBSVM [23] in batch mode is shown. See Sec 4.5

the end of the learning process, the data present in the active set is saved for probability calibration. Calibration for unlearning is done with data in the system at any given point of time. Test performance is predicted on 600 pairs, as proposed in LFW protocol. At no time instant, the system is allowed to exceed the prescribed budget size B.

We conducted a number of experiments to assess the performance of the proposed system. Fig 4.4 shows comparison of performance of incremental learning/unlearning algorithm proposed and a leading off-the-shelf technique for budgeted online learning [41]. In their previous work, Wang *et al.* [170] found budgeted stochastic gradient descent (BSGD-remove) algorithm with removal strategy (sample to be removed based on smallest norm with respect to current SVM solution) to be a leading method. The performance of BSGD-remove and the proposed algorithm improves as the size of budget is increased. For very small budget sizes, proposed incremental learning/unlearning method performs worse than BSGD-remove. The performance saturates around budget size 1200 for incremental learning and 1800 for BSGD. For reference, the performance obtained with batch learning (1950 support vectors) with LIBSVM [114] is plotted. The performance of both the methods is significantly worse when budget size is less than the number of feature dimensions used for training. Thus, the proposed methods perform better at most budget sizes compared to a BSGD-remove. Henceforth, in our probability calibration experiments, we show the performance on data obtained from incremental learning/unlearning method presented in this work.

Probability Calibration: In earlier section 4.4, we discussed posterior probability estimation methods for streaming face recognition problem. We first discuss a methodology to evaluate calibration methods followed by our experiments. Reliability diagrams are frequently used in meteorology to assess the probability forecasts for binary events such as probability of measurable precipitation. On X-axis of reliability diagram, mean predicted value is plotted and on Y-axis fraction of positives is plotted. This chart effectively tells the user how often (as a percentage) of predicted probability actual event occurs. Thus, the diagonal line joining $[0, 0]$ and $[1, 1]$ represents ideally calibrated system [174] [56]. Discrete Brier skill score (BSS_D) measures the accuracy of probabilistic predictions. This measure is often used in weather forecasting to assess performance of a system plotted on a reliability diagram. The general formulation of the score is

$$BSS_D = \frac{1}{N}\sum_{t=1}^{N}(f_t - o_t)^2 \tag{4.9}$$

where f_t is the predicted probability and o_t is the actual outcome (binary) and N is total number of events. The score ranges from $[0, 1]$ with 0 representing ideally calibrated system and 1 representing worst possible calibration. Fig 4.5 shows BSS_D plotted for different budget sizes for different calibration methods. For a fixed budget size, model was learned incrementally with all the 5400 training samples for each LFW split. The data in the active set at the end of the complete run is used to calibrate model (i.e. in case of Platt, this data is used to estimate A and B params as described in sec 4.4.1 and estimating the Weibull CDF parameters scale λ_c, shape κ_c and location τ_c in Sec 4.4.2). Once the models for both methods are calibrated, posterior probability is estimated for each example in the test set. This process is repeated for each test set and each budget size. The figure 4.5 represents average and standard error over all the splits for these runs.

When data available is limited for calibration, EVT-based calibration gives more reliable posterior probability estimates compared to Platt's method. When more data is available, the performance of both the method converges. EVT calibration focusses only on tails of distribution. As budget size increases the amount of calibration data needed by Platt calibration reaches closer to entire training set. In case of EVT, only tails of distribution are required (irrespective of whether all or only part of the data is available for calibration). These

Iteration Type	Calibration Method	BSS_D
All Training Data	Platt	**0.1024**
All Training Data	EVT	0.1385
Support Vectors	Platt	0.1345
Support Vectors	EVT	**0.1057**
Budget 600	Platt	0.1285
Budget 600	EVT	**0.1080**
Budget 400	Platt	0.4939
Budget 400	EVT	**0.1769**

trends are reflected more clearly in reliability diagram shown in Fig 4.6. When all training data is available for calibration, both Platt and EVT calibration oscillates around ideal calibration line, suggesting well calibrated models. When the models are calibrated using only support vectors (this was obtained when budget size was equal to total training samples and using only support vectors for calibration), the calibration process, appears to oscillate further from the ideal calibration line. For smaller budget size, this phenomenon is amplified.

In the following table, BSS_Ds for the examples plotted in 4.6 are given for reference.

4.6 Discussion

In this work , we presented a method suitable for streaming face recognition problem. We build on existing work on incremental learning and adapt it to incrementally unlearn training samples. Our system can operate on user-specified budget and still yield comparable (and often better) performance to existing off-the-shelf budgeted online learning methods. We proposed a novel posterior probability estimation method based on statistical extreme value theory for streaming face recognition problem. We carry out thorough analysis of the proposed method with respect to state-of-the-art estimation technique and show our method provides more reliable estimates for lower budget sizes. Finally we demonstrate our results on a unconstrained face verification problem and show the suitability of the proposed method to handle long streams of data.

Although there many advantages to the proposed incremental unlearning and EVT based calibration method, there are some limitations. The incremental learning algorithm of [21] scales with Budget size $O(B^2)$. Hence, for really large budget sizes, other alternative learning approaches might be practical. The space requirement of the algorithm is primarily due to saving the state of the optimality (KKT) conditions. Approaches such as Pegasos [146] save only SVM decision boundary, however compromise significantly on accuracy [170]. The proposed calibration techniques should be useful for online learning, irrespective of the choice underlying budget SVM learning algorithm.

We considered a particular approach for unlearning of training sample (described in 4.3). For streaming face recognition, when the system gains input from multiple cameras [55], a fast filtering mechanism could be devised based on correlation or Nearest Neighbour approach. This could ensure fast processing of incoming data, without explicitly adding all the training samples to the learned model to adapt. A combination of proposed calibration technique with methods in consumer face recognition offer an extremely interesting future direction [71]

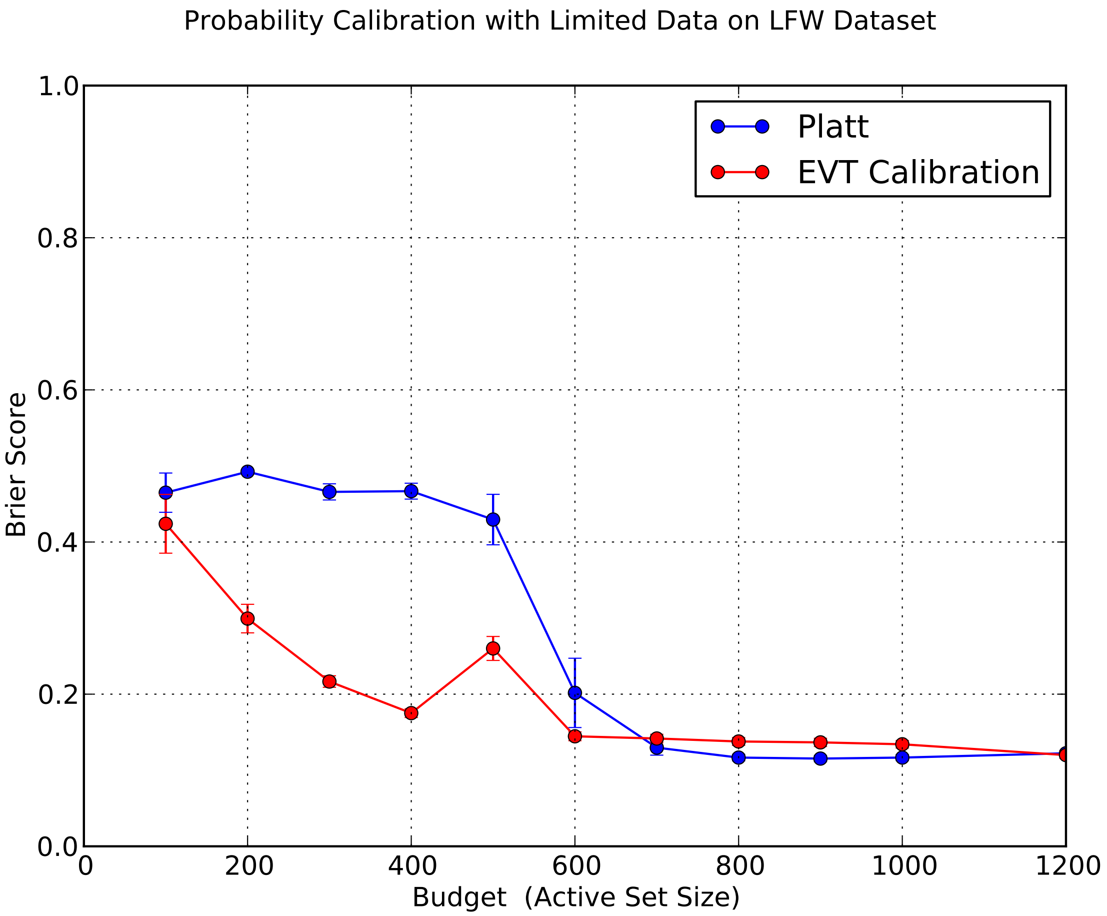


Figure 4.5: Brier Score Analysis for varying budget sizes. When limited data is available for calibration EVT based calibration gives more reliable posterior probability estimates compared to Platt's method. When more data is available, the performance of both the method converges. (The discrete Brier Skill Score ranges from $[0, 1]$ with 0 representing ideally calibrated system and 1 representing worst possible calibration.)

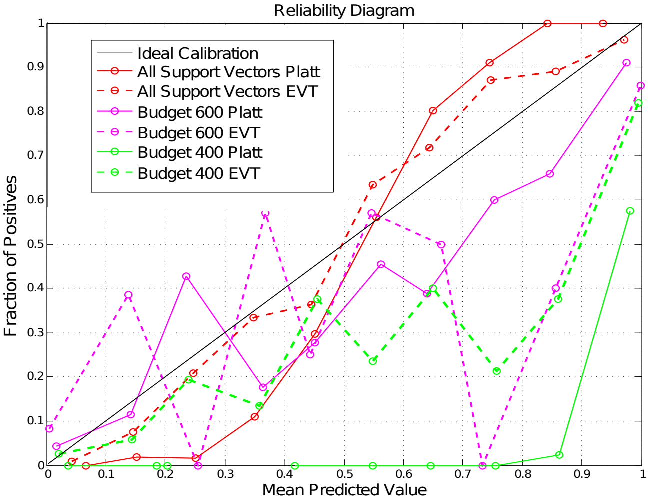


Figure 4.6: As the amount of calibration data reduces, the reliability of calibration for both Platt and EVT decrease but the EVT degrades most slowly, e.g. consider the green solid (platt) vs green dashedEVT). The EVT is much closer to the diagonal, which is ideal reliability. EVT calibration provides robust posterior estimations when limited data is available for calibration. See BSS_Ds for methods mentioned above in table

Chapter 5

Incremental Model Adaptation for Face Recognition in the Wild

5.1 Introduction

Traditionally, the problem of face recognition is considered to be stationary one and has been extensively studied in the domain of batch learning. A face recognition system is usually developed in two phases: training/development phase and testing/evaluation phase. In training phase, domain specific dataset is collected, algorithms are tuned to this training data and evaluated on a hold out set. The system is then deployed for its application (identification/verification). The performance of face recognition system is good if the system designers are able to model the operational conditions well. A general assumption behind this mode of design is that the domain of the problem is stationary i.e. the distribution from which the training and testing data is drawn is fixed, but unknown. However, these assumptions often do not hold true in many real-world application. In a surveillance application, one might want to add new identities in the system based on detection of an event. A system vendor might want to develop a general face recognition system and then adapt it to its operational environment: e.g. different backgrounds. In yet another application, system resources might be limited and one would want to continously add and remove enrolled identities without taking the entire system down. A face recognition system designed to operate in the "wild" needs to have the ability to continuously adapt to changing environments. As hardware (e.g. surveillance cameras) become cheaper, it is easier to capture more application specific data on the fly than to replicate every operational environment.

A preferred way to learn from continuos influx of data is with incremental learning. Incremental learning is the process of learning one (or multiple) sample at a time and adapting the learned model. An incremental learning method should have some essential components: it should perform comparably with its batch learning counterpart, it should quick and easy to add training samples to already learned models and it should be able to learn from long streams of data [125]. A more challenging scenario for incremental model adaptation is learning in semi-supervised setting from limited labelled data [131] when such data is drawn from a non-stationary distribution.

The two major paradigms to handle classification problems are generative methods and discriminative methods. Generative models are built to explain how samples have been generated and specifies a joint probability distribution over observations and label sequences [116]. Popular among these methods are Gaussian mixture models, Latent Dirichlet Allocation, Hidden Markov Models etc. Generative models explain the data in a way that the model parameters link hidden variables to observations so as to fit the probability density of the observed data [12]. Discriminative methods focus the boundaries between categories rather than the entire density function over data. In recent years, discriminative methods such as Support Vector Machines, Linear Discriminant Analysis, Logistics Regression have gained significant attention [12]. Researchers in face recognition have used both these paradigms with considerable success for developing large scale face recognition systems.

Developing incremental learning framework is difficult for both generative and discriminative methods. For generative methods like GMMs the challenge lies in selection of model complexity. Arandjelovic et. al. [4] noted that when learning one sample at a time, the incoming sample never carries enough information to cause an increase in number of Gaussian components. In case of discriminative learning methods like incremental SVMs, while a single observation near the boundary might be enough to shift the hyperplane significantly it also adds additional overhead of storing optimality conditions of the existing model [21] [156]. If one tries to circumvent the storage requirement and move towards approximate solutions [146], the performance of the system is significantly affected. In many online discriminative methods, optimization is carried out in primal domain instead of the dual like their batch learning counterparts. In primal domain, the optimization depends on number of feature dimensions. For face recognition applications where feature dimensions are often large and number of training samples per client is often limited. Thus, from a face recognition system designer's perspective it is imperative to ask the question: ***which incremental learning methodology is most suited for face recognition applications in the wild?***

Moving forward, it is important to note the difference between online and incremental learning frameworks [57]. In online setting, data is available for learning either sequentially or in bursts. This is different than offline setting where entire data is available for the training algorithm. A training algorithm can learn incrementally or in batch setting. An incremental algorithm can learn one sample at a time, as opposed to batch learning. Thus, an incremental algorithm can work in both online and offline settings. In this work, we consider incremental algorithms with online settings: a difficult but an important problem to make large scale face recognition in wild practical.

In this work we present a comparison of incremental learning methods suited for face recognition in nonstationary and evolving environments. We consider a generative method of Gaussian Mixture Models and a discriminative method of Support Vector Machine for our comparison. We first present a protocol suited for incremental learning in the wild for face recognition. We present a comparison between incremental SVMs and incremental GMMs. We present a semi-supervised approach to add unlabelled data in previously learned models in incremental learning framework. Finally, we present performance analysis of the proposed learning methods and their suitability for face recognition in the wild.

The contributions of the work can be summarized as follows:

1. Detailed comparison of GMMs and SVMs in the context of incremental face recognition.

2. Incremental model adaptation in the presence of limited labelled data and unlabeled data.

3. A novel protocol for incremental model adaptation with unlabeled data.

4. Performance analysis on three unconstrained face recognition in the wild datasets.

5.2 Related Work

5.2.0.1 Discriminative vs Generative Classifiers Literature

Ng et. al. [106] carried out one of the first studies comparing generative and discriminative classifiers. They compared naive Bayes and logistic regression methods empirically and theoretically and noted that while discriminative learning has a lower asymptotic errors, a generative classifier approaches its asymptotic error at much faster rate. More recently Bishop et. al. [13] proposed a hybrid approach based on discriminative training of generative classifiers. A more thorough treatment on hybrid of generative discriminative methods can be found in the work of Lassere et. al. [84]. Although researchers have compared properties of generative and discriminative classifiers, a thorough comparison in incremental settings, especially for the problem of face recognition has been overlooked.

5.2.0.2 Incremental Learning for Face Recognition

In recent years number of approaches for incremental learning have been explored. Deniz et. al. [39] developed a method based on incremental refinement of decision boundaries by actively using input from an expert user. Yan et. al. [176] developed a method based on incremental Linear Discriminant Analysis in spectral regression framework for face recognition. Arandjelovic et. al. [4] studied the problem of face detection by incrementally learning GMMs with temporal coherency. They noted that the task of updating GMM was surprisingly difficult. In their work, they were able to exploit temporal coherency, a commonplace in tracking and detection application. For recognition application, coherency becomes less relevant and extreme variations are often important to model for superior recognition performance [136]. Verification applications involve constructing of a universal background model (UBM), along with a density estimation process (GMM). While Kristan et. al. [74] has some interesting properties from the perspective of one-dimensional data, it is unclear how these methods can be directly applied to verification applications. Kim et. al. [73] present incremental linear discriminant analysis for face recognition with some of the leading constrained face recognition datasets such as BANCA [5]. Their method is based on incremental updates of total scatter matrix, between class scatter matrix and projected data matrix. However, the problem of incremental face recognition in unconstrained environment is still an open problem, especially under unsupervised settings. In medical imaging, Carneiro et. al. [20] devised incremental online semi-supervised learning for segmenting left ventricle of heart from ultrasound data. While overall learning theme is similar to the methodology adopted in this work, their application environment is orthogonal to this work.

5.2.0.3 Incremental Client Model Adaptation

: GMMs for face verification draw inspiration from the area of speaker verification [168]. The problem of incremental model adaptation in speaker verification is often referred to as progressive speaker adaptation [179]. In this work, we build on some protocols used in progressive speaker adaptation for unsupervised model adaptation for face recognition.

5.3 Incremental Model Adaptation

5.3.1 Support Vector Machines

The primary intuition behind SVM classification is to map the data into a high dimensional space and find a max-margin separating hyperplane for efficient classification. We assume that we are given a set of training vectors $x_i \in \mathbb{R}^n, i = 1, ..m$ in two classes, and a vector of labels **y** such that $y_i \in \{1, -1\}$.

$$\max_{\boldsymbol{w},b,\xi} \mathcal{P}(\boldsymbol{w}, b, \xi) = \frac{1}{2}\boldsymbol{w}^2 + C\sum_{i=1}^{m} \xi_i \tag{5.1}$$

$$\text{subject to} \left\{ \; y_i(\boldsymbol{w}^\intercal \boldsymbol{\phi}(x_i) + b) \geq 1 - \xi_i, \forall i \; \xi_i \geq 0, \forall i \right. \tag{5.2}$$

where training data is mapped to a higher dimensional space by the kernel function $\boldsymbol{\phi}(.)$, and C is a penalty parameter on the training error (trade-off between accuracy and model complexity), ξ_i are the slack variables used when training instances are not linearly separable and b is the classifier bias. In the formulation of SVM in equation 6.1, the term w defines the orientation of hyperplane with respect to origin of the feature space ($\mathbb{R}^n$) and the bias term b defines the distance of the hyperplane from the origin[18]. This is a quadratic programming problem and can be solved efficiently for linear and non-linear kernels

5.3.2 Incremental Support Vector Machine

Let us assume we have a set of training data $D = \{(x_i, y_i)\}_{i=1}^{k}$, where $x_i \in \mathcal{X} \subseteq \mathcal{R}^n$ is input and $y_i \in \{+1, -1\}$ is the output class label. Support Vector Machines learn the function $f(x) = w^T \phi(x) + b$, where $\phi(x)$ denotes a fixed feature space transformation. The dual formulation of this problem involves estimation of α_i, where α are the Lagrange multipliers associated with the constraints of the primal SVM problem. These coefficients are obtained by minimizing a convex quadratic objective function under the constraints

$$\min_{0 \leq \alpha_i \leq C} : W = \frac{1}{2} \sum_{i,j} \alpha_i Q_{ij} \alpha_j - \sum_i \alpha_i + \sum_i y_i \alpha_i \tag{5.3}$$

where b is the bias (offset), Q_{ij} is the symmetric positive definite kernel matrix $Q_{ij} = y_i y_j K(x_i, x_j)$ and C is the nonnegative user-specified slack parameter that balances model complexity and loss of training data. The first order conditions on W reduce to the KKT conditions, from which following relationships are obtained:

$$\begin{aligned} y_i f(x_i) &> 1 \Rightarrow \alpha_i = 0 \\ y_i f(x_i) &= 1 \Rightarrow \alpha_i \in [0, C] \\ y_i f(x_i) &< 1 \Rightarrow \alpha_i = C \end{aligned} \tag{5.4}$$

and $\sum_{i=1}^{k} y_i \alpha_i = 0$. These conditions partition the training data into three discrete sets: margin support vectors ($\alpha_i \in [0, C]$), error support vectors ($\alpha_i = C$) and ignored vectors. Decision score for test sample x_t is obtained using $f(x) = w^T \phi(x) + b$ where

$$w = \sum_{i=1}^{l} y_i \alpha_i \phi(x_i) \tag{5.5}$$

where l is total number of support vectors (consisting of margin support vectors and error support vectors). This is the traditional batch learning problem for SVM [164].

The incremental extension for SVM suggested by [21]. In this method, the KKT conditions are preserved after each training iteration. For incremental training, when a new training sample (x_c, y_c) is presented to the system, the Lagrangian coefficients (α_c) corresponding to this sample and positive definite matrix Q_{ij} from the SVM currently in the memory undergo a small change Δ to ensure maintenance of optimal KKT condition (details of these increments can be found in [83], [21]).

5.3.3 Incremental Gaussian Mixture Models

5.3.3.1 Gaussian Mixture Models

A GMM is a generative model that consists of K multivariate Gaussian components [123]. Each component k is defined by a mean vector μ_k, a co-variance matrix $\sum_k$, which can be assumed to be diagonal and a weight vector w_k. A GMM is described by a set of parameters $\Theta = \{w_k, \mu_k \sum_k\}_{k=1..K}$. Given the parameters Θ, the likelihood of feature vectors O is

$$P(O|\Theta) = \prod_b \sum_k w_k \mathcal{N}[o^{-b}|\mu_k, w_k] \tag{5.6}$$

where $\mathcal{N}$ is multivariate gaussian with mean μ_k and covariance matrix $\sum_k$. A GMM $\mathcal{M}^{(c)}$is created for each client identity c with the help of a Universal Background Model (UBM). The UBM is used as a prior and is adapted for each enrolled client c. UBM ($\mathcal{M}^{(ubm)}$) is a gaussian mixture model, which is trained on feature vectors extracted from the world model set by using an iterative *expectation-maximization* (EM) algorithm [37]. The client model $\mathcal{M}^{(c)}$ is adapted from $\mathcal{M}^{(ubm)}$ towards the enrollment features using *maximum a posteriori* (MAP) estimate [51]. This process facilitates generation of client models from limited number of enrollment images, as often the case in face recognition.

Once the model is trained, a probe image O can be authenticated against a client model $\mathcal{M}^{(c)}$ by calculating a log likelihood ratio (LLR) as follows:

$$S_{GMM}(O|\mathcal{M}^{(c)}) = log\left(\prod_{b=1}^{B} \frac{P(o|m^{(o)})}{P(o|m^{(ubm)})}\right) \tag{5.7}$$

By applying a threshold value τ (usually obtained from development set), the LLR (or score) can then be used in a decision rule where O is said to match to client model $\mathcal{M}^{(c)}$ if and only if $S_{GMM}(O|\mathcal{M}^{(c)}) \geq \tau$

5.3.3.2 Incremental GMM Adaptation

As seen in section 5.2, the problem of incremental model adaptation for face recognition in Gaussian Mixture Models framework has been overlooked in the past. While there exist works from tracking domain and incremental updates for one-dimensional data, it is unclear how these methods can be applied to problem of face verification. The problem is especially challenging in the presence of limited or unlabeled data. In this section, we present multiple strategies for performing incremental model updates with GMMs for face verification. As discussed earlier, client specific model is denoted by $\mathcal{M}^{(c)}$. Session specific client model is denoted as $\mathcal{M}_i^{(c)}|_{i=0..n}$, where i denotes the number of times adaptation process takes place. In the initial state (with no client specific training data), the model is denoted as $\mathcal{M}_0^{(c)}$. After first adaptation process (i.e. when the system receives first set of training images for a given client), the model is denoted as $\mathcal{M}_1^{(c)}$ and so on. At any given time, prediction is carried out with $\mathcal{M}_k^{(c)}$ where k is the current state of model in system.

Adapt UBM and replace: Everytime when training samples are available, the UBM $\mathcal{M}^{(ubm)}$ is adapted using MAP estimate to create session specific client model $\mathcal{M}_k^{(c)}$. The client model that existed in the memory previously $\mathcal{M}_{k-1}^{(c)}$, is destroyed. The prediction phase is carried out using the new client model $\mathcal{M}_k^{(c)}$. For a given probe image O LLR is computed as follows:

$$S_{GMM}(O|\mathcal{M}_k^{(c)}) = log\left(\prod_{b=1}^{B}\frac{P(o|m_k^{(o)})}{P(o|m^{(ubm)})}\right) \tag{5.8}$$

Adapt UBM and combine scores: When system is presented with incremental client specific training samples, the UBM $\mathcal{M}^{(ubm)}$ is adapted in similar manner as with replace strategy. However, instead of replacing the existing client model $\mathcal{M}_{k-1}^{(c)}$, the new client model is stored in the memory. At any time instant after the adaptation process is completed, all the previous client model are stored in the system $\mathcal{M}_K^{(c)} = \{\mathcal{M}_1^{(c)}, \mathcal{M}_2^{(c)}, ..., \mathcal{M}_{k-1}^{(c)}, \mathcal{M}_k^{(c)}\}$ (note $\mathcal{M}_0^{(c)}$ indicates initial state of the client model with no training data). Let score computed by eqn 5.8 with respect to each adapted client model $\mathcal{M}_k^{(c)}$ be S_k. During prediction phase after K increments, for a probe image LLR is computed as follows:

$$S_{GMM}(O|\mathcal{M}_K^{(c)}) = \frac{1}{k}\sum_{k=1}^{K} S_k \tag{5.9}$$

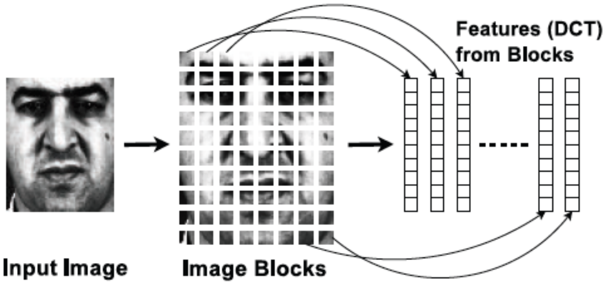


Figure 5.1: Parts Based DCT Feature extraction:After preprocessing, the input image is divided into multiple blocks. From each block DCT features are extracted

5.4 Experiments

5.4.1 Features

5.4.1.1 Parts Based Features

In this section we discuss the preprocessing and feature extraction applied for discriminative and generative learning methods. To minimize the effect of variation in illumination across different image capture conditions, we apply multi-stage preprocessing algorithm of Tan et. al [161]. The algorithm applies gamma correction followed by difference in gaussian filtering and finally applies contrast equalization. We use the default parameters as reported in [161].

Following the preprocessing step, parts-based features [130] are extracted by decomposing preprocessed images into B overlapping blocks. For each block, the pixel intensity is normalized to zero mean and unit variance. A 2D discrete cosine transform (DCT) is applied to each block, before extracting the F lowest-frequency DCT coefficients that form the descriptor of a given block. For a given image the resulting block-based feature vectors are normalized to zero mean and unit variance in each dimension [3]. Each preprocessed image is finally described by a set of B features vectors. Fig 5.1 shows computation of parts-based features from image to set of blocks and extracting DCT features from each block. For SVMs we vectorize all the DCT components into a single feature vector.

In case of MOBIO dataset, we use 45 DCT coefficients with total block overlap of 11 pixels and block size being a total of 12. These block features (as shown in Fig 5.1) are used for GMM training. The blocks are vectorized into a single feature vector for SVM training. The total size of the feature vector amounts to a total of 160908. Thus, SVM is trained in a feature space with 160908 dimensions. It might be possible that an alternate DCT feature representation might be better suited for SVM training. However, in this work we use same feature set for both GMM and SVM training for fair comparison between the two learning paradigms.

5.4.2 Evaluation

A biometric verification system can make two types of mistakes: when either real access is rejected (false rejection) or an impostor is falsely accepted by the system. These mistakes are quantified with the help of false rejection rate (FRR) and false acceptance rate (FAR). These performance numbers can be combined by using half total error rate defined as

$$HTER(\tau, \mathcal{D}) = \frac{FRR(\tau, \mathcal{D}) + FAR(\tau, \mathcal{D})}{2} \tag{5.10}$$

where $\mathcal{D}$ denotes the dataset used. It is important to note that both FRR and FAR are dependent on the operational threshold τ and hence are strongly related to each other: increasing FAR will reduce FRR and vice-versa.

We consider a protocol that splits the data in three sets: a training set, a development set and an evaluation set. Scores and FAR/FRR are computed for both the development and the evaluation set independently. Then, a threshold τ^* is obtained based on the intersection point of FAR and FRR curves of the development set. This threshold is used to compute the equal error rate (EER) on the development set and the half total error rate (HTER) on the evaluation set as follows:

$$EER = \frac{FAR_{dev}(\tau^*) + FRR_{dev}(\tau^*)}{2} \tag{5.11}$$

$$HTER = \frac{FAR_{eval}(\tau^*) + FRR_{eval}(\tau^*)}{2} \tag{5.12}$$

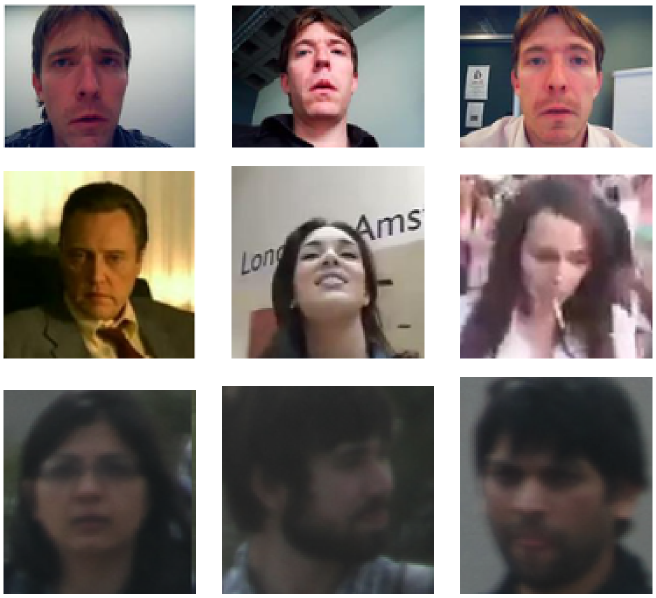


Figure 5.2: Example images from various datasets used in this work. The top row contains images from MOBIO [3] dataset and the bottom row contains images from Youtube faces [175] dataset. All three datasets are unconstrained face datasets obtained in the wild

5.4.3 Datasets

5.4.3.1 MOBIO Face Dataset

The mobile biometry (MOBIO) database [3] consists of video data of 152 people taken with mobile devices like mobile phones or a laptop. In this work, we consider data collected from mobile phones only. The dataset contains two separate gender specific splits consisting of male and female. For each client 12 different sessions were recorded. The MOBIO protocol is supplied with the database[1] and defines three non-overlapping partitions: world (training), development and testing. The development and testing partitions are defined in a gender dependent manner, such that subjects models are only probed by images from subjects of the same gender. The dataset is challenging since it offers significant variations in facial expressions, pose, illumination conditions and occlusion. The dataset is divided into three sets: training set, development set and evaluation set. A subset (full training data consists of 9579 images) of training set of 1224 images from 34 subjects (36 images each) was used for UBM training. The development set consists of 24 clients for male split and 18

[1] http://www.idiap.ch/dataset/mobio

clients for female split. The evaluation set consists of 38 clients for male split and 20 clients for female split. In each session, 5 images per client are available for enrolling client models.

5.4.3.2 Youtube Face Dataset

YouTube face dataset (YTF) [175] is one of the largest unconstrained video based face dataset available. The dataset contains videos from a total of 1595 individuals with 3425 videos in total. For each individual anywhere from 1-6 videos are provided with the dataset[2]. The shortest video clip is of 48 frames and longest clip contains 6070 frames, with each video containing an average of 181.3 frames. The authors also provide meta-data such as bounding box co-ordinates for detected faces. The bounding box around the face is expanded by 2.2 of its original size and cropped from the frame. This is further resized to standard dimensions of 200x200 pixels. Finally the image is cropped again leaving an image centered on the face of size 100x100 pixels.

5.4.3.3 UnConstrained College Student Face Dataset

UnConstrained College Student (UCCS) face dataset[3] [131] is a large scale dataset captured designed for openset face recognition at a distance. The release 1 of the dataset (which we consider in this work) consists of 6,337 images from 308 individuals. The camera is placed inside an office room and is focused on the outdoor sidewalk at 100m distance from the office room, resulting in 18 Megapixels scene images. Images are captured at an interval of 100msec, resulting in around 10 pictures of a person at different focal points, with multiple views and expressions at each particular interval. As it is collected on a college campus, the chances of the same person appearing in front of the camera the next day at the same interval is high. For example, a student taking Monday-Wednesday classes at 12:30 PM will show up in the camera on almost every Monday and Wednesday during that time interval. This results in multiple sequences of an individual on multiple days. The images contain various weather conditions such as sunny versus snowy days. They also contain various occlusions such as sunglasses, winter caps, fur jackets, etc., and occlusion due to tree branches, poles, etc.

[2] http://www.cs.tau.ac.il/ wolf/ytfaces/

[3] http://vast.uccs.edu/uccsfaces

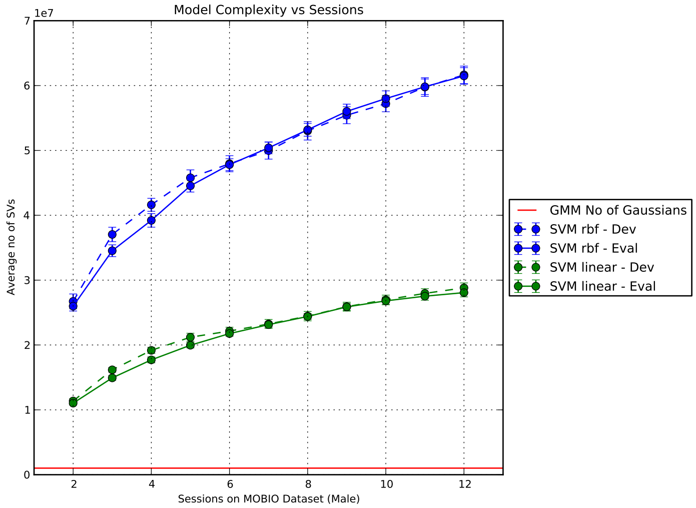


Figure 5.3: Model Complexity as a function of sessions. As data is added, model complexity for SVMs increase, while that for GMM remains constant. The above plot shows data obtained on MOBIO dataset (male-split). Performance numbers are shown for SVMs with linear kernel and RBF Kernels

5.4.4 Experiments on MOBIO Dataset

For our experiments we use the mobile split of MOBIO dataset i.e. data obtained from mobile phones. The data is divied into 12 sessions. For each session a set of training images are available with which client models are created. In case of GMMs, client images for each session are used to adapt client specific GMM from a common UBM (discussed later). For SVMs, the client images form the positive class (as in a standard 1-vs-all binary classification problem). Per session, 5 images per client are available for model creation. At session 12, the system is exposed to a total of 55 images (note in MOBIO dataset enrollement starts from session 2). This setting is similar for both development and evaluation split of the dataset and for all the clients. The same process is carried out for female split of MOBIO (only difference being the number of identities). A common world model consisting of 1224 images is used for both GMM and SVM. In case of GMMs, world model is used to train UBM. For SVMs, this set is used as a common negative set.

The experiments are conducted in two modes: Batch mode and incremental mode. In batch mode, all training samples are provided at same time, with no possibility for adaptation. We use batch mode SVM and batch mode GMM as baseline. In batch mode for each session a client model is trained with images from

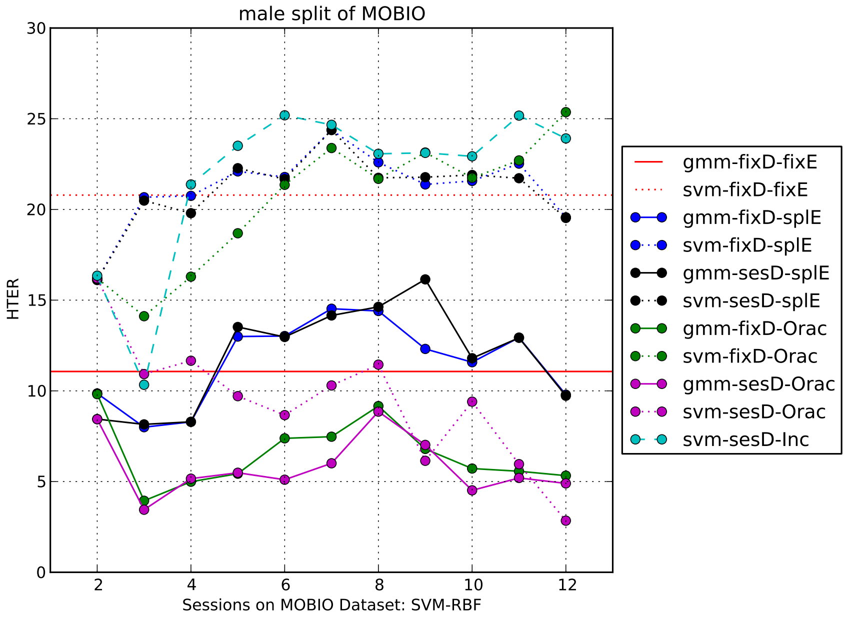


Figure 5.4: Performance on MOBIO dataset (male split) as a function of sessions. Data is added in each session to the learned models. The models are updated with incremental SVMs and GMMs.

current session and all the previous sessions. Thus for any given client for session 2, 5 images are used to enroll For session 3 images from both session 2 and session 3 are used for enrollment. In incremental mode, enrollment procedure for session 2 is similar. For session 3, only images from session 3 and statistical model (either SVM model or GMM model) is available. The goal is to adapt the learned statistical model with images from session 3. This process is repeated for all successive sessions in incremental setting. A set of independent images is used for probing. The process of probing is same for both batch and incremental setting to allow fair comparison between the two algorithms.

Fig 5.3 shows model complexity for SVM and GMMs. The experiments are performed for batch setting only, to understand the effect of model complexity as a function of sessions (i.e. as number of training images are increased). For SVMs, we conducted experiments with linear kernel and RBF kernel. On Y-axis model complexity is plotted. In case of SVM, model complexity is denoted by number of support vectors multiplied by number of feature dimensions. For GMM, model complexity is number of gaussians multiplied by feature dimensions. In case of GMMs, the model complexity remains constant irrespective of number of training images. We note that for SVM with linear kernel and RBF kernel, model complexity increases with number of training sessions.

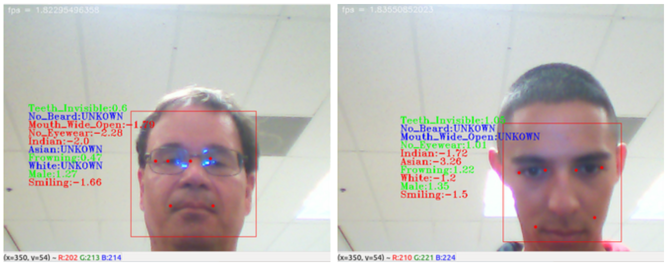
Fig 5.4 shows accuracy as a function of training sessions. For both GMM and SVM, we first eastablish a baseline performance with batch learning. The baseline performance is obtained by training models on development set, eastablishing τ and then applying these parameters on evaluation set. HTER measure obtained on probe set is plotted on Y-axis. In incremental settings, SVM model is adapted from the model from previous session and training images from current session. We note that on DCT features, GMM perform better compared to SVM: both in batch and incremental mode. The performance for batch SVM improves as more training images are obtained. In incremental setting, SVM performance degrades. The difference in performance between incremental and batch setting is not as drastic as SVM, but still far from perfect.

Chapter 6

What do you do when you know that you don't know?

Real-world biometrics recognition problems often have two unknowns: the person be recognized, as well as a hidden unknown - missing data. If we choose always to ignore data that is occasionally missing, we sacrifice accuracy. In this paper, we present a novel technique to address the problem of handling missing data in biometrics systems without having to make implicit assumptions on the distribution of the underlying data. We introduce the concept of "operational adaptation" for biometric systems and formalize the problem. We present a solution for handling missing data based on refactoring on Support Vector Machines for large scale face recognition tasks. We also develop a general approach to estimating SVM refactoring risk. We present experiments on large-scale face recognition based on describable visual attributes on LFW dataset. Our approach consistently outperforms state-of-the-art methods designed to handle missing data.

Biometrics systems have been widely adopted in various walks of life, thanks to significant progress in various sub-fields in the past decade. Cheap sensors, models learned with large amounts of data, an abundance of processing power have all led to development and deployment of biometrics systems beyond the narrow scope of research labs [62]. Biometric recognition in unconstrained settings imposes little restrictions on the data acquisition and processing procedure. Biometric recognition in the open world leads to multiple challenges. Failing assumptions, failing code, or missing inputs can then lead to missing data in higher-level



```
fps = 1.82295496358
Teeth_Invisible:0.6
No_Beard:UNKOWN
Mouth_Wide_Open:-1.79
No_Eyewear:-2.28
Indian:-2.0
Asian:UNKOWN
Frowning:0.47
White:UNKOWN
Male:1.27
Smiling:-1.66
(x=350, y=54) ~ R:202 G:213 B:214
```

```
fps = 1.83550852023
Teeth_Invisible:1.05
No_Beard:UNKOWN
Mouth_Wide_Open:UNKOWN
No_Eyewear:1.01
Indian:-1.72
Asian:-3.26
Frowning:1.22
White:-1.2
Male:1.35
Smiling:-1.5
(x=350, y=54) ~ R:210 G:221 B:224
```

Figure 6.1: A system for describable visual attributes for faces based [172], extended for open-set recognition with attributes as "unknown" . In the image, green text is a positive attribute, red text is negative attributes and blue color signifies unknown/missing attribute. In the above images, the left shows how bad lighting/feature-detection led to "UNKNOWN" labels for *Asian, While* attributes. The example on the right shows occlusion leading to the *no beard* attribute being labeled "UNKNOWN". Handling such unknowns at run-time, in a learning-based biometrics system poses considerable operational challenges. This paper is about what we do when we know we don't know some feature.

feature descriptions. Matching models (especially learned-models) with missing data is challenging. How to do recognition in the face of these "unknowns" is the fundamental problem that we address in this paper.

In recent years, describable visual attributes have emerged as a powerful low-level feature representation for a wide range of face recognition applications [134, 78]. Kumar *et al.* [78] demonstrated a system to automatically train several attribute classifiers for faces, such as "brown hair", "mustache", "blonde", "pointy nose", "thin eyebrows", "wearing lipstick" etc. Attribute classifiers take an image as input and return a real-valued score representing the presence of the given attribute in the image. These real-valued scores can then be used in a full-fledged face recognition system for identification/retrieval [134]. While the system designed by Kumar *et al.* was primarily for closed set face verification task, more recently, Wilber *et al.* [172] have proposed open set extensions for such systems. As noted by Scheirer *et al.* [141] "when a recognition system is trained and is operational, there are finite set of known objects in scenes with myriad unknown objects, combinations and configurations - labeling something new, novel or unknown should always be a valid outcome". In open-set systems, a specific face attribute is named unknown if the system is either unable to classify with sufficient confidence or is presented with an image/feature that it has not seen before. Such open set “unknown” labeling thus leads to known missing status for the respective attribute (see Fig 6.1). Systems designed to handle open set recognition have demonstrated excellent performance on many biometrics and computer vision systems in wild [172, 7, 26, 173].

This paper introduces and addresses a novel and practical problem **Operational Adaptation**, where given only a previously trained operational system and a test instance $\boldsymbol{x}_t$ with some described difference from the normal instances, the system must adapt to the constraints of data to make predictions and to provide estimates of the risk of adaptation. While there is a growing body of work in domain adaptation and transfer learning that work towards adapting classifier during the training phase, such approaches are not practical for a machine that may take days or weeks to train. In this work, we focus our attention on the more common and prevalent missing data problem, what [52, 81] calls the “nightmare at test time”, where at test time the operational data is corrupted or missing. This is a nightmare because it cannot be avoided. We contend there are two important subproblems within the nightmare. The first is the obvious one, making decisions using partial data. The second, and generally overlookedproblem, is estimating how scared we should be using that partial data. Intuitively, many users would assume that losing 70% of features yields a nearly useless classifier while losing only one feature is probably not bad. However, even one missing feature can lead to horrible performance if that is a critical feature, while the 70% missing may have little impact.

There are multiple contributions of this work. We formally define the problem of operational adaptation and present a novel solution for handling missing data with SVMs based on SVM re-factoring with bias re-optimization. Our solution offers superior results to many state-of-the-art approaches both in terms of accuracy and storage space. Further, we develop a general approach to estimating SVM Refactoring Risk. Our risk estimation process provides the associated risk when performing predictions with missing data. We show the proposed adaptation risk estimation is a better predictor of success/failure [133] than percent missing data. We use describable visual attribute representation on large scale face verification tasks for our experiments. The proposed approach consistently performs Labelled Faces in the Wild [59] and other machine learning datasets such as USPS [60] and MNIST [86]. Our new method is the first step toward addressing an important problem for operational use of machine learning for large scale biometrics recognition systems.

6.1 Related Work

Handling missing data in biometrics is an important problem and has been addressed by multiple researchers in the past. Ding *et al.* [40] performed a detailed study comparing multiple imputation methods for score fusion in biometrics. Poh *et al.* [120] proposed a framework for addressing kernel based multi-modal biometric fusion using neutral point substitution. Other notable works in the domain of handling missing data for biometric score fusion are by Fatukasi *et al.* [45] and Damer *et al.* [33]. Our work differs from these works in multiple aspects. Ding *et al.* showed promising results for score fusion with generative models with relatively lower feature dimensions. In our work, we focus mainly on large-scale discriminative models such as SVMs. Further, the problem of interest of our work is run-time [81] adaptation of learned models for verification/recognition systems unlike the works of Fatukasi *et al.* , Poh *et al.* and Damer *et al.* where the focus is primarily on fusion rules for biometric score fusion.

Researchers in machine learning and statistics communities [98, 25, 129] have also addressed the problem of learning from missing data. Chechik *et al.* [25] proposed a max-margin learning framework that is based on geometric interpretation of the margin and aims to maximize the margin of each sample in its own relevant subspace. The work of [129] presents a comprehensive evaluation framework comparing imputation based methods and reduced feature models. Reduced feature models are constructed for each type of missing pattern separately making the problem computationally extremely expensive. Such methods are unsuitable for biometrics where feature dimensionality tends to be very high as it requires storing reduced model for every permutation of missing feature space[1].

6.2 Operational Adaptation

Given a training set $\{\mathcal{D}_S = x_i, y_i \in \mathcal{X}$ x $\mathcal{Y} : \{+1, -1\}\}$, where $\mathcal{X}$ is the input space and $\mathcal{Y}$ is a finite training set. The learning problem is to find a function $f \ : \ \mathcal{X} \rightarrow \mathcal{Y}$ that obtains high predictive accuracy. In this work, we focus our attention on run-time adaptation of a pre-trained classifier to new operational domains, in the presence of limited data and model parameters. We assume existence of a Source Domain $\mathcal{D}_S$ and a family

[1] e.g. for LBP like features where feature dimensionality is around 200 features, total number of reduced models that need to be stored would be $2^2 00$

of Operational Domains $\mathcal{D}_{o_j} \neq \mathcal{D}_S$. We assume the training set $\mathcal{D}_S \in \mathbb{R}^n$ and each test sample belongs to an operational domain $\mathcal{D}_{o_j}$, where $\mathcal{D}_{o_j} \in \mathbb{R}^{k_j}$ where, in general, $k_j \leq n$. Let $\boldsymbol{M}_j$ be an operator such that $\boldsymbol{M}_j : \mathcal{D}_S \rightarrow \mathcal{D}_{o_j}$, i.e. it map items in the source domain to the operational domain. While the definition of operational adaptation can be more general, in this work, we focus on problems where the operational domain $\mathcal{D}_o$ has missing features compared to source domain $\mathcal{D}_S$, in which case, $\boldsymbol{M}$ projects away dimensions associated with missing features. For operational adaptation, we enforce that, during operational time, we only have access only to the operational data O_D, which includes the learned prediction function f and its associated parameters $(\theta_1..\theta_f)$. In terms of SVMs, one could view f, b, type of kernel, α and the support vectors as the operational data O_D.

Definition 2. Operational Adaptation: *Given a learned prediction function $f(\theta_1, ..\theta_f)$ over some source (training) domain $\mathcal{D}_S$ defined by operational data O_D, an operational domain $\mathcal{D}_o$, a transformation operator $\boldsymbol{M}$ relating $\mathcal{D}_S$ to $\mathcal{D}_o$, and test point $x \in \mathcal{D}_o$, the problem of Operational Adaptation is:*

1. *to obtain adapted prediction function $f_o()$ and an effective prediction function over the operational domain $\mathcal{D}_o$*
2. *obtain an associated operational adaptation risk measure $\mathcal{R}_o : (f_o(), x) \rightarrow [0, 1]$, which estimates the likelihood of failure of the prediction function f_o.*

In this paper, we focus on the difference between source domain $\mathcal{D}_S$ and operational domain $\mathcal{D}_o$ as difference in dimensionality, in particular in the remainder, we presume that $\boldsymbol{M}$ is linear projection. However, the idea of operational adaptation applies to any problem which satisfies the constraints mentioned in definition 1, e.g. the general definition includes operational domains that involve linear basis transformations or even non-linear remapping. In this particular definition, while we presume that $\boldsymbol{M}$ is given, a more general form may involve estimating $\boldsymbol{M}$.

6.2.1 SVM Refactoring and Run Time Bias Estimation

The primary intuition behind SVM classification is to map the data into a high dimensional space and find a max-margin separating hyperplane for efficient classification. In this section, we present a methodology

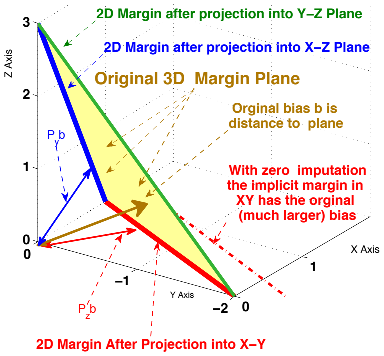

2D Margin after projection into Y–Z Plane
2D Margin after projection into X–Z Plane
Original 3D Margin Plane
Orginal bias b is distance to plane
With zero imputation the implicit margin in XY has the orginal (much larger) bias
$P_y b$
$P_z b$
2D Margin After Projection into X–Y
Z Axis
Y Axis
X Axis
3
2
1
0
0
−1
−2
0
1

Figure 6.2: With Bias re-factoring, the data and the bias vector are both projected and distances are computed in the lower-dimensional subspace. For this example, presume 3D features with the original margin plane in 3D with bias b. Classic imputation by zero, if Z is missing, computes distances in the X-Y plane but keeping the original bias sets a much higher threshold as show in the dashed red line. When features are missing, the Projection of the Bias decomposed vector is like using a lower-dimensional margin for decision making. If the Z feature is missing, P_z project the data and the bias vector the X-Y plane, effectively using the Red margin, but if y is missing, P_y projects into the X-Z plane effectively using the blue margin. Bias-decomposition seeks to adjust the bias from the margin to the original plane to appropriate the projection subspace.

to modify support vector machines and introduce the idea of operationally adapted instance specific bias. The estimation for bias in operational domain is based on modifying the independent variables in dual of objective function of SVM to minimize the classification error over support vectors. Our SVM refactoring method is computationally efficient compared to reduced model methods, and more accurate than zero or mean imputations. The proposed method for bias estimation at prediction time is termed as Run Time Bias Estimation (RTBE). The intuition behind this method is explained in detail in Fig 6.2

We assume that we are given a set of training vectors $x_i \in \mathbb{R}^n, i = 1, ..m$ in two classes, and a vector of labels **y** such that $y_i \in \{1, -1\}$.

$$\max_{\boldsymbol{w},b,\xi} \mathcal{P}(\boldsymbol{w}, b, \xi) = \frac{1}{2}\boldsymbol{w}^2 + C\sum_{i=1}^{m} \xi_i \tag{6.1}$$

$$\text{subject to} \left\{ \; y_i(\boldsymbol{w}^\intercal \boldsymbol{\phi}(x_i) + b) \geq 1 - \xi_i, \forall i \; \xi_i \geq 0, \forall i \right. \tag{6.2}$$

where training data is mapped to a higher dimensional space by the kernel function $\boldsymbol{\phi}(.)$, and C is a penalty parameter on the training error (trade-off between accuracy and model complexity), ξ_i are the slack variables used when training instances are not linearly separable and b is the classifier bias [105]. In the formulation of SVM in equation 6.1, the term w defines the orientation of hyperplane with respect to origin of the feature space ($\mathbb{R}^n$) and the bias term b defines the distance of the hyperplane from the origin[18].

The dual of problem in equation 6.1 is given as

$$\max \mathcal{F}(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} y_i \alpha_i y_j \alpha_j \phi(x_i, x_j) \tag{6.3}$$

$$\text{subject to} \left\{ \ \forall i, 0 \leq \alpha_i \leq C \ \sum y_i \alpha_i = 0 \right. \tag{6.4}$$

where $K(x_i, x_j) = \phi(x_i)^\intercal \phi(x_j)$ is matrix of kernel values. Using positive Lagrange coefficients $\alpha + i \geq 0$, the Lagrangian of the dual problem is given as

$$\begin{aligned} \mathcal{L}(\boldsymbol{w}, b, \xi, \alpha) &= \frac{1}{2}\boldsymbol{w}^2 + C\sum_{i=1}^{m} \xi_i - \\ & \sum_{i=1}^{m} \alpha_i (y_i(\boldsymbol{w}^\intercal \phi(x_i) + b) - 1 + \xi_i) \end{aligned} \tag{6.5}$$

which leads to the formal dual objective function $\mathcal{F}(\boldsymbol{\alpha})$ as:

$$\mathcal{F}(\boldsymbol{\alpha}) = \min_{\boldsymbol{w}, b, \xi} \mathcal{L}(\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}) \ \text{ subject to } \ \forall i, \xi_i \geq 0 \tag{6.6}$$

The optimization of the dual objective function directly produces $\boldsymbol{\alpha}*$, yielding w.

Let $\boldsymbol{\alpha}^* = (\alpha_1^* .. \alpha_m^*)$ be a solution of the dual problem of equation 6.6 where $\boldsymbol{\alpha}^*$ satisfies the dual constraints. The vector $\boldsymbol{\alpha}^*$ is generally sparse, with many zero elements. Let v be the number of non-zero elements α^*, let $\boldsymbol{s} = [s_i], i = 1 \ldots v$, the support vectors, be a remaining of the training points x_j associated with the non-zero elements of α^*. The operational data O_D for the SVM is thus $\{\boldsymbol{w}, \boldsymbol{\alpha}, \boldsymbol{s}, \boldsymbol{y}, b, K\}$ and $\boldsymbol{M}$. Given these, the optimal value of b can be obtained via 1-dimensional optimization over the decision function (i.e. the primal problem) using the training data [18].

Let us now derive our approach to SVM refactoring, an operational adaptation approach that provides both improved classification as well as risk estimation. Our challenge is to define a solution in the reduced dimensional space using only operational data. A plausible solution for operational adaptation would be to could be to treat the support vectors as a training set, project them into the operational domain and train a new optimal SVM solution. While plausible, our results show that this approach does not provide acceptable rate of classification on test data, e.g. in Figure 6.3, it is evaluated on the USPS dataset where it is only slightly better than zero imputation. On some other examples, it did much worse than zero imputation.

Thus, we seek an approach that will reuse more of learned structure than just the knowledge of the support vectors, in particular, to adapt the optimal weights. If we revisit the dual of objective function in equation 6.6, we note that re-optimizing values of α, ξ or $\boldsymbol{w}$ for the operational domain would the require projection of all the training data which would violate the definition of operational adaptation. Thus, the only reasonable perturbation/optimization that can be performed is re-optimization of the bias term b.

Letting $\ell(x, y)$ be the loss function for estimate x given label y, and let L_n be the empirical loss, over full support vectors $\boldsymbol{s}$ using original SVM f in $\mathbb{R}^n$. Then we first define our refactored projected error as:

$$\varepsilon(\hat{f}(\cdot; b)) = \frac{1}{v} \sum_{j=1}^{v} \ell(\sum_{i=1}^{v} \alpha_i y_i \phi(\boldsymbol{M} s_i, \boldsymbol{M} s_j) + b), y_j) \tag{6.7}$$

And using this we define our refactor risk as:

$$R_r(\hat{f}(\cdot; b), \boldsymbol{s}) = 1 - \frac{\min\left(1 - L_n,\ 1 - \varepsilon(\hat{f}(\cdot; b))\right)}{1 - L_n} \tag{6.8}$$

where we normalize by $1 - L_n$, so the refactor risk is relative risk compared to the original loss. We include the $\min()$ because noise or irrelevant variables may result in the projected support vectors having lower empirical risk than the original dimensional version. In the paper, we generally exclusively used 0-1 error, thus $1 - L_n$ is average accuracy. The changes to use absolute, square, or other loss functions should be minimal because we are only measuring the loss on the projected support vectors.

6.2.2 Adaptation Risk Estimation

Since the classification accuracy depends on dimensions of the missing features and the technique used to reclassify data in the reduced dimensions, we refer to our re-factored risk model as the **Adaptation Risk Estimator** (eqn 6.8). It applies to *any* reduced dimensional model $\hat{f}$ for dealing with the partial data, not just re-factored SVM with factored bias. In particular, if the projection matrix $\boldsymbol{M}_0$ (transformation operator $\boldsymbol{M}$ between $\mathcal{D}_S$ and $\mathcal{D}_o$) is N x N and fills in "missing" data with zero, then the model is zero imputation and we can estimate the risk of using zero-imputation. It equally applies to mean-imputation.

Intuitively, our risk model is conservative as it uses difficult examples to estimate risk of failure. The actual performance can be much better if the example is far from the boundary, suggesting better risk estimators could be developed. In case of classifiers like decision trees or random forests [19], where only thresholds of tree splits are retained, operational adaptation could be achieved by retaining an "operational validation set" at run-time. Note the set of support vectors is an extremely biased and correlated set and hence it might violate the assumptions of statistical tests for distributional shifting such as those considered in [27].

Returning to the re-optimization bias b_o for the refactored machine. Note that our refactor risk estimation applies to any machine f, and in particular we explicitly called out that it is a function of the underlying bias b. Using this, our refactored approach, we will will seek a new operational bias b_o for equation 6.7, which is obtained from the 1-D optimization problem minimizing refactor risk:

$$b_o = \operatorname*{argmin}_b R_r(f_o(\cdot; b), \boldsymbol{s}) \tag{6.9}$$

i.e., the re-optimization of the bias is performed based on minimization of risk over the support vectors projected into $\mathbb{R}^k$. If using 0-1 loss, as we have in the paper, the loss function is non-convex, however explicit 1-D optimization is still very efficient. We also note that, in practice, for operational adaptation with just missing variables, the computations of both the classification f_o and optimization of b_o can replace the matrix multiplication by $\boldsymbol{M}$, which is mostly zeros, with a selection operation that simply selects the relevant dimensions. With that, the cost to optimize b_o is dominated by estimating risk at the values of b associated with the v projected support vector locations. In our experiments, we found that minimizing refactor risk was infact a good predictor of performance on test-set. To show this relationship, we plot results of varying bias for a particular operational domain and noting the associated refactor risk and test set classification accuracy 6.4. However, it should be noted that this experiment is done to show the relationship between refactor risk and test accuracy and during optimization process, we do not assume any knowledge about test set apart from the test-sample under consideration.

6.3 Experiments

In this section, we evaluate the proposed algorithm for SVM based re-factoring (i.e. Run Time Bias Estimation - RTBE) on USPS [60], MNIST [86] and Labelled Faces in the Wild [59] datasets (Fig 6.5). USPS and MNIST are leading handwriting recognition datasets and results proposed on those can be compared against wide variety of methods across different disciplines (e.g. biometrics, computer vision, machine learning, statistics etc.). The feature dimensionality considered for all datasets is high to show the suitability of the proposed methods on large-scale recognition tasks. USPS dataset [60] contains 9298 handwritten digits (0 - 9) (7291 for training and 2007 for testing) collected from mail envelopes in Buffalo. Each image is represented as a 256 dimensional feature vector. The MNIST database consists of 60,000 training samples and 10000 testing samples for digits between (0-9). The digits are size-normalized and centered in a fixed-size image. Size of each image is 28 x 28 leading to a feature vector of size 784. Scaled pixel values are provided for performing supervised classification task.

LFW face dataset [59] is designed for large scale face verification task and contains 13233 images of 5749 individuals. View 1 of LFW is used for building models, feature selection and finding optimal operating parameters. View 2 consists of 6000 pairs of face images on which performance is to be reported. The 6000 pairs are further divided in 10 splits to allow 10-fold cross-validation. Overall classification performance is reported on View 2 by using only the signs of the outputs and counting the number of errors in classification. We use describable visual attributes [78] on LFW dataset for face verification task. Attribute classifiers are created by using describable visual traits such as gender, race, hair color etc. These visual traits are used to construct classifiers C_K. These classifiers are then used to detect presence/absence of attribute in a given face image and a score is assigned to it. Each image in LFW dataset is thus represented as a vector $C(I_i) = \langle C_1(I_i), C_2(I_i)..C_K(I_i)\rangle$ (where K is total number of attributes/traits). To decide if the image belongs to the same person, these classifier scores are compared $\{C(I_i), C(I_j)\}$. Verification classifier for a pair of images is given as $v(I_i, I_j) = (|C_i - C_j|, (C_i.C_j), \frac{1}{2}(C_i + C_j))$. These classifier scores are used as input features for face verification task.

We systematically delete features from testing as percentages of total features present for each dataset. Each set of missing feature leads to a new operational domain $\mathcal{D}_o$. We consider percentage of missing features in range of 10%, 20%, 30%, 40% and 50%. The process is kept similar for all three datasets. For each dataset, we trained SVM with linear and RBF kernel essentially training model in $\mathbb{R}^n$ (where n = $\{$ 256, 146 and 784$\}$ for USPS, Attributes and MNIST respectively). During test phase, for zero imputation method, all the missing features are substituted by zero and classification is carried out. With RTBE approach, we detect the missing features, project the Support Vectors in corresponding operational domain and obtain new optimal bias b_o by minimizing refactor risk over support vectors (operational data). As a baseline, we also obtain results on respective datasets without deleting any features. It is obvious that the performance of the system would be best when all the features are present. We observe that the proposed approach of RTBE consistently outperforms zero imputation across multiple datasets. We also note that rate of performance degradation for RTBE is lower compared to zero imputation. For USPS and MNIST dataset, the performance obtained with RTBE continues to remain stable even under extreme cases (e.g. 40% and 50% missing data).

6.3.1 Comparison with Other Methods for Handling Missing Data

The state of the art for handling missing data for USPS dataset is multi-class Gaussian Process [153] yielding 94.2% (error of 5.8 %) at 25% features missing (64 pixels out of 256). In the same work, authors noted Zero imputation resulted in classification accuracy of 94.15 (5.85% Error) and mean imputation yielded 93.92 (6.08% error). On the same dataset with similar train/test protocol, our method of RTBE achieves overall classification accuracy of 95.11 % (4.89 % error) using linear kernel and 98.26 % (error 1.76 %) using RBF kernel (a 69.65 % reduction of error over the state of the art). Chechik *et al.* [25] reported their results on MNIST dataset by considering classification on digits ‘5’ and ‘6’. They remove a square patch randomly from the image and report a performance of 95% using their geometric margin method (similar performance is reported for their averaged norm method in the same study). Our approach on RTBE yields 96.6% on similar problem of classifying ‘5’ and ‘6’. To the best of our knowledge, no study has been done on handling missing data on attributes on LFW [78].

6.3.2 Risk Estimation for Missing Data

To evaluate the effectiveness of the Adaptation Risk Estimation, we use the meta-recognition evaluation paradigm, MRET (Meta-Recognition Error Trade-off Curves), proposed in [137, 133], which considers how often the risk estimator correctly predicts the failure/success of the underlying classifier. We consider two risk estimation approaches: percentage of missing features (wrt to total features) and the risk estimator from Eq. 6.8. For risk estimation with percentage of missing features, we drop features in steps (e.g. 10%, 20% etc.) and at each step we predict success/failure. When using refactor risk $\mathcal{R}(f_o)$ as risk estimator (from Eq. 6.8), the range of refactor risk is divided into steps and at each step success and failure is computed using formulae from 6.10. For each of risk estimators we consider both zero-imputation and the SVM refactoring via bias factoring. One can threshold the risk estimator and predict any instance below threshold to be successfully classified and predict failure for those above it.

In particular, we define

C_1 = # instance when the risk estimator is below threshold yet the adapted SVM misclassifies

C_2 = # instance when the risk estimator is above threshold yet the adapted SVM correctly classifies

C_3 = # instance when the risk estimator is below threshold and the adapted SVM correctly classifies

C_4 = # instance when the risk estimator is above threshold yet the adapted SVM misclassifies

Finally, we calculate the Meta-Recognition False Accept Rate (MRFAR), the rate at which thresholded risk estimator incorrectly predicts success, and the Meta-Recognition Miss Detection Rate (MRMDR), the rate at which the thresholded risk estimator incorrectly predicts failure, as

$$MRFAR = \frac{|C_1|}{|C_1| + |C_4|}, \;\; MRMDR = \frac{|C_2|}{|C_2| + |C_3|}. \tag{6.10}$$

and then vary our threshold to compute the curves shown in Fig. 6.7. The resulting MRET curves shows the proposed adaptation risk estimator is superior, and is more effective when combined with our novel SVM re-factoring. At operation time, just as one uses a traditional DET or ROC curve to set verification system parameters, the threshold on the risk parameter $\mathcal{R}(f_o)$ on MRET curve can be used to tune the rejection for an acceptable risk due to missing data. The results of this experiments[2] are shown in Fig 6.7.

6.4 Discussion and Conclusion

We noted that support vectors is an extremely biased and correlated set, and hence it might violate the assumptions of statistical tests for distributional shifting [27]. Detailed analysis of such correlation is an important future work. In streaming settings (incremental SVMs) for face recognition [6], the operational data available is always changing as support vectors are continuously updated. Handling missing data in such settings is another important aspect of operational adaptation. Some other problems such as adapting pre-trained classifiers for face-detection [66, 65] can be viewed as operational adaptation problems.

[2] We obtained similar results on USPS and MNIST dataset, but are not shown here

This paper provides theory and a novel solution for handling missing data in large-scale recognition problems. It adapts the solution at testing time, with virtually no loss in computational speed/efficiency, but significant improvements in accuracy compared to the state of the art. Further, it does not require apriori knowledge of missing features. SVM refactoring with bias factoring performed consistently well on leading datasets compared to current de-facto methods, and when only modest data was missing, significantly outperformed the competition. Our method is suitable for large scale recognition tasks for many applications in computer vision like object recognition, feature tracking, action recognition, etc. that use supervised learning in the form of SVMs when features are missing. The second significant contribution of the is a technique for estimating the risk associated with classification with missing data, using only the data in the operational SVM. Our approach reclassifies the SVM in the reduced space and estimates the associated risk. Experiments show this risk measure is a better estimator of expected performance on the reduced dataset than just using the fraction of data missing. Finally, we show that the concept of operational adaptation is broader and applies to multiple areas beyond the domain of handling missing data.

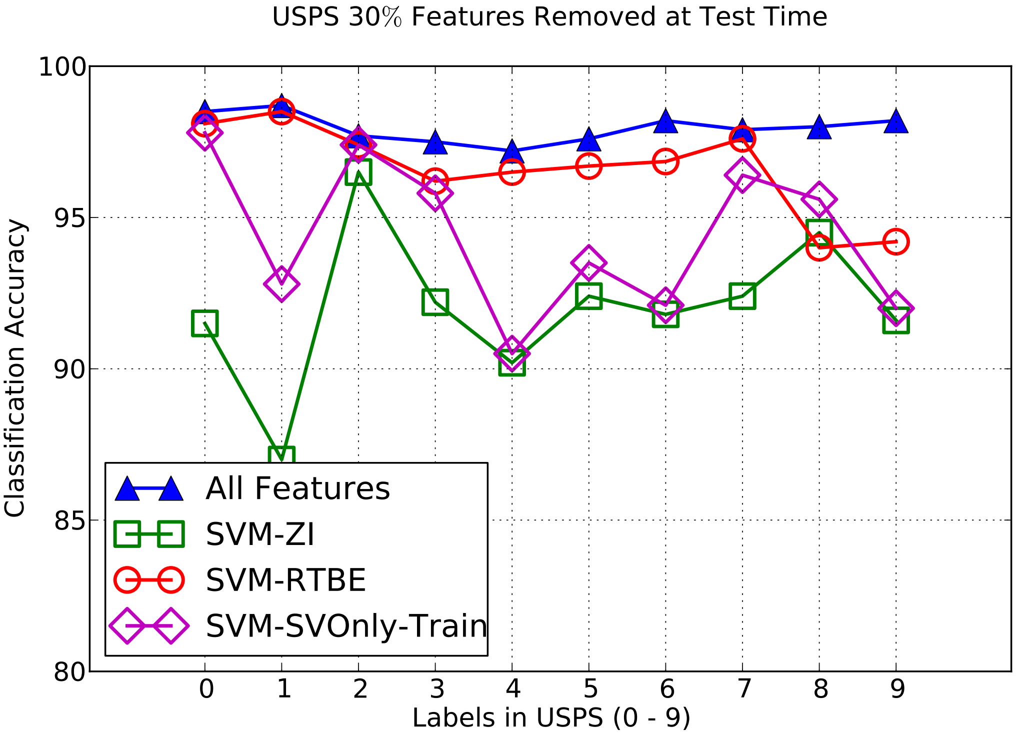


Figure 6.3: The above figure contains test classification accuracy for each digit in USPS dataset when 30% features were deleted from test samples. The results were obtained by training SVM with RBF kernel. The methods shown correspond to nature of test set. a) All Features Present : All the features were present during test time. b) Zero Imputation: 30% features were removed at test time and missing features were imputed with zero. c) Training with SVs only: 30% features were removed from test samples. Corresponding features were removed from Support Vectors and a new model with these support vectors was trained. Results shown are classification results obtained with this new trained model d) RTBE: 30% features were removed at test time. SVM bias was re-optimized using our approach and classification results were obtained using optimized bias for operational domain $\mathcal{D}_o$.

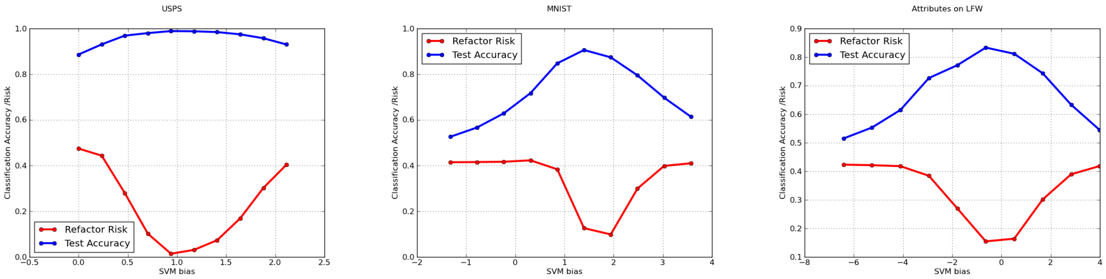


Figure 6.4: Effect of Varying SVM Bias b on classification accuracy on test set and associated refactor risk. It can be observed that when our refactor risk is minimum, maximum classification accuracy over test is obtained (Test Accuracy was scaled between 0 - 1 to fit this plot). Our objective is to obtain value of SVM bias b for which refactor risk is minimum. In the above experiment, 30% features were removed for each dataset during testing phase. The datasets shown are (from L-R) USPS [60], MNIST [86] and Attributes on LFW [59]

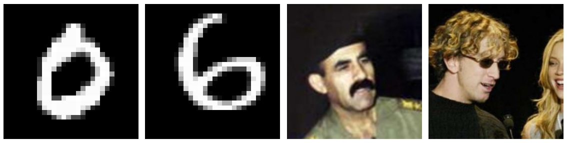


Figure 6.5: Example images from MNIST [86])(left) and LFW datasets [59] (right)

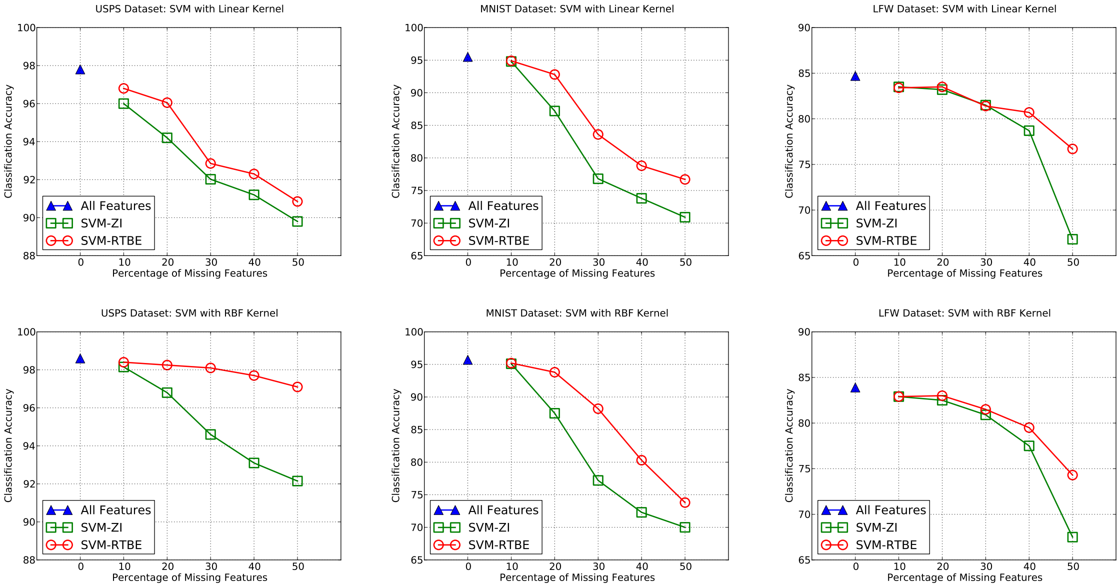


Figure 6.6: The above figure shows results on three image datasets used for recognition: USPS, MNIST and LFW. The top row corresponds to results with SVM with linear kernel and bottom row corresponds to SVM with RBF kernel. Classification accuracy (Y - axis) is plotted as a function of percentage of missing features (X - Axis). It can be observed that RTBE consistently performs better than imputation with zero when features are missing. The difference is especially large when percentage of missing features increases

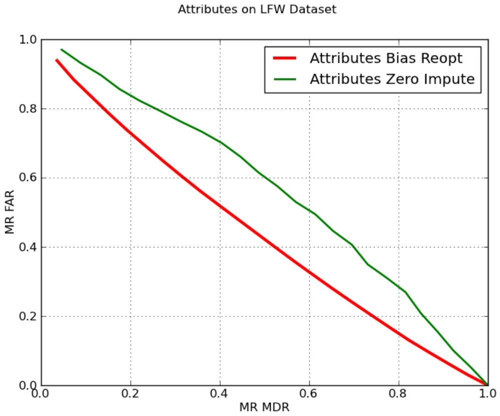


Figure 6.7: Meta-recognition comparison curve for evaluation adaptation risk estimators on partial data. The ideal location is the lower-left. Meta-recognition False accept rate (Y axis) is the fraction of time the risk was low but the SVM classification failed, and the Meta-recognition miss detection rate is the fraction of time the risk was high yet the SVM corrected classified the partial data. The plot shows the adaptation risk estimator for SVM refactoring (RTBE) is better than risk estimation for zero-imputation.

Chapter 7

Moving Forward

7.1 In Summary

Recognition in the open world is a challenging problem. In this work, we investigated multiple challenges involved in building recognition system that can operate in robustly in changing and evolving operational conditions. In the work on open world recognition, we extend existing work on NCM classifiers and show how to adapt it for open world recognition. The proposed Nearest Non-Outlier (NNO) algorithm consistently outperforms NCM on open world recognition tasks and is comparable to NCM on closed set – we gain robustness to the open world without much sacrifice. NNO allows construction of scalable systems that can be updated incrementally with additional classes and that are robust to unseen categories. Such systems are suitable where minimum downtime is desired.

In the work on streaming face recognition , we presented a system that can incrementally learn and unlearn under budget constraints of operational conditions. We proposed a novel posterior probability estimation method based on statistical extreme value theory for streaming face recognition problem. y we demonstrate our results on a unconstrained face verification problem and show the suitability of the proposed method to handle long streams of data.

In the work on incremental model adaptation, We compared discriminative and generative approaches for model adaptation, more specifically support vector machines and gaussian mixture models. We show that learned models can be adapted with minimal downtime without significantly compromising on accuracy. We investigated various properties of generative and discriminative models for incremental learning on large scale face recognition tasks.

Finally, in the chapter on handling missing data, we proposed a refactoring based approach for Support Vector Machines that allows the user to make predictions and asses associated risk in the presence of partially missing features. We showed that the proposed approach did not require apriori knowledge of the missing feature dimensions. Further, we proposed a technique for estimating the risk associated with classification with missing data, using only the data in the operational SVM. Our approach reclassifies the SVM in the reduced space and estimates the associated risk. Finally, we show that the concept of operational adaptation is broader and applies to multiple areas beyond the domain of handling missing data.

7.2 Open Questions and Future Directions

The proposed approaches denote significant advances in handling some operational challenges faced by current state-of-the-art recognition systems. However, the task is far from done. In order to build intelligent machines that operate similar to human intelligence, a recognition system needs to be able to do lot more than merely assign a label from bag of available labels in an image classification task. Recognition system needs to be self-aware about how much it has learned so far, how it can improve the learned models and how it can adapt with the changing operational scenarios.

In the work on open world recognition, we explored a strategy to learning novel concepts for large scale recognition settings. Open world evaluation across multiple features for a variety of applications is an important future work. Recent advances in deep learning and other areas of visual recognition have demonstrated significant improvements in absolute performance. The best performing systems on such tasks use a parallel system and train for days. Extending these systems to support incremental open world performance may allow one to provide a hybrid solution where one reuses the deeply learned features with a top layer of an open world multi-class algorithm. While scalable learning in the open world is critical for deploying computer vision applications in the real world, high performing systems enable adoption by masses. Pushing absolute performance on large scale visual recognition challenges, and development of scalable systems for the open world are essentially two sides of the same coin.

In the work of open world recognition and streaming recognition we assumed presence of an Oracle for identifying novel concepts. However, there is a need for detailed investigation of unsupervised and weakly supervised learning methodologies for incorporating novel concepts in a recognition system. It is extremely hard to have reliable evaluation mechanism for traditional clustering based unsupervised learning frameworks. In the work on open world recognition, we proposed a threshold based novelty detection technique to identifying if the concept is not learned by the system. However, the proposed technique does not provide any additional details about the nature of novelty. It might be beneficial to encode a hierarchical knowledge base in an open world recognition system that provides more details about the novelty of the incoming image. Instead of labelling an image of an unseen dog category as merely an "unknown" image, it might be useful to let the user know that it is an "unknown dog" image. This will allow the user to send such an image to a dog expert for labelling.

There are number of open questions to consider in the quest for building intelligent vision systems. How to learn concepts that build on top of existing concepts, without explicit knowledge of hierarchy? How to learn relationships between existing concepts? How can we incorporate additional human knowledge about the visual world besides specifying training examples? [115, 82]

Bibliography

[1] G Agarwal, S Biswas, P Flynn, and K Boyer. Predicting performance of face recognition systems: An image characterization approach. *IEEE CVPR Biometrics Workshop*, 2011. 4

[2] H Agrawal, N Chavali, C Mathialagan, Y Goyal, A Alfadda, P Banik, and D Batra. Cloudcv: Large-scale distributed computer vision as a cloud service. 2013. 32

[3] A. Anjos, L. El Shafey, R Wallace, M. Gunther, C. McCool, and S. Marcel. Bob: a free signal processing and machine learning toolbox for researchers. *ACM-MM*, 2012. xvi, 84, 86

[4] O Arandjelovic and R Cipolla. Incremental learning of temporally-coherent gaussian mixture models. *BMVC*, 2005. 64, 77, 79

[5] E. Bailly-Bailliere, S. Bengio, F. Bimbot, M. Hamouz, J. Kittler, J. Mariethoz, J. Matas, K. Messer, V. Popovici, F. Poree, B. Ruiz, and J Thiran. The banca database and evaluation protocol. *International Conference on Audio- and Video-Based Biometric Person Authentication*, 2003. 79

[6] A Bendale and T Boult. Reliable posterior probability estimation for streaming face recognition. *CVPR Biometrics Workshop*, 2014. 102

[7] A Bendale and T Boult. Towards open world recognition. *CVPR*, 2015. 93

[8] Abhijit Bendale and Terrance E. Boult. Towards open world recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1893–1902, June 2015. 40, 42, 43, 44, 50, 51, 53

[9] A Berg, J Deng, and L Fei-Fei. Large scale visual recognition challenge, 2010. `http://image-net.org/challenges/LSVRC/2010/index`[Online; accessed 1-Nov-2013]. 13, 14, 22, 28

[10] L Best-Rowden, B Klare, J Klontz, and A Jain. Video-to-video face matching: Establishing a baseline for unconstrained face recognition. *BTAS*, 2013. 59

[11] Battista Biggio, Blaine Nelson, and Pavel Laskov. Support vector machines under adversarial label noise. *ACML*, 2011. 65

[12] C Bishop. Pattern recognition and machine learning. *Springer*, 2006. 77

[13] C Bishop and J Lasserre. Generative or discriminative? getting the best of both worlds. *ISBA Eighth World Meeting on Bayesian Statistics*, 2006. 79

[14] S Blunsden and R Fisher. The behave video dataset: ground truthed video for multi-person behavior classification. *Annals of BMVA*, 2010. 59

[15] Paul Bodesheim, Alexander Freytag, Erik Rodner, Michael Kemmler, and Joachim Denzler. Kernel null space methods for novelty detection. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 3374–3381. IEEE, 2013. 12

[16] Paul Bodesheim, Alexender Freytag, Erik Rodner, and Joachim Denzler. Local novelty detection in multi-class recognition problems. In *Winter Conference on Applications of Computer Vision, 2015 IEEE Conference on*. IEEE, 2015. 40

[17] A Bordes, S Ertekin, J Weston, and L Bottou. Fast kernel classifiers with online and active learning. *JMLR*, 2005. 4

[18] L Bottou and Chih-Jen Lin. Support vector machine solvers. *Large Scale Kernel Machines - MIT Press*, 2007. 80, 97

[19] L Breiman, J Friedman, R Olshen, and C Stone. Classification and regression trees. *Wadsworth*, 1984. 98

[20] Gustavo Carneiro and Jacinto Nascimento. Incremental on-line semi-supervised learning for segmenting the left ventricle of the heart from ultrasound data. *ICCV*, 2011. 80

[21] Gert Cauwenberghs and Tomaso Poggio. Incremental and decremental support vector machine learning. In *Advances in Neural Information Processing Systems 13: Proceedings of the 2000 Conference*, volume 13, page 409. MIT Press, 2001. 4, 12, 13, 60, 61, 62, 64, 65, 73, 77, 81, 82

[22] N. Cesa-Bianchi and C. Gentile. Tracking the best hyperplane with a simple budget perceptron. *Machine Learning*, 2007. 64

[23] C Chang and C Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. xv, 70

[24] Ken Chatfield, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *Proceedings of the British Machine Vision Conference,(BMVC)*, 2014. 40

[25] G Chechik, G Heitz, G Elidan, P Abbeel, and D Koller. Max-margin classification of data with absent features. *J. of Machine Learning Research*, pages 1–21, 2008. 94, 101

[26] G Chiachia, AX Falcao, N Pinto, A Rocha, and D Cox. Learning person-specific representations from faces in the wild. *IEEE TIFS*, 2014. 93

[27] David A Cieslak and Nitesh V Chawla. A framework for monitoring classifiers performance: when and why failure occurs? *Knowledge and Information Systems*, 18(1):83–108, 2009. 99, 102

[28] K Crammer and Y Singer. On the algorithmic implementations of multiclass kernel-based vector machines. *JMLR*, 2001. 23

[29] Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. Online passive-aggressive algorithms. *The Journal of Machine Learning Research*, 7:551–585, 2006. 12, 14

[30] Jill D Crisman and Charles E Thorpe. Color vision for road following. In *1988 Robotics Conferences*, pages 175–185. International Society for Optics and Photonics, 1989. 14

[31] Qing Da, Yang Yu, and Zhi-Hua Zhou. Learning with augmented class by exploiting unlabeled data. In *AAAI Conference on Artificial Intelligence*. AAAI, 2014. 40

[32] N Dalal and B Triggs. Histogram of oriented gradient for object detection. *CVPR*, 2005. xii, 31, 32

[33] N Damer, B Fuhrer, and A Kuijper. Missing data estimation in multi-biometric identification and verification. *IEEE Biometric Measurements and Systems for Security and Medical Applications Workshop*, 2013. 94

[34] Piew Datta and Dennis Kibler. Symbolic nearest mean classifiers. In *AAAI/IAAI*, pages 82–87, 1997. 14

[35] Thomas Dean, Mark A Ruzon, Mark Segal, Jonathon Shlens, Sudheendra Vijayanarasimhan, and Jay Yagnik. Fast, accurate detection of 100,000 object classes on a single machine. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 1814–1821. IEEE, 2013. 11, 40

[36] O. Dekel, S. S. Shwartz, and Y. Singer. The forgetron: A kernel based perceptron on a budget. *SIAM Journal of Computation*, 2008. 64

[37] A Dempster, N Laird, and D Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society B*, 1977. 82

[38] Jia Deng, Sanjeev Satheesh, Alexander C Berg, and Fei Li. Fast and balanced: Efficient label tree learning for large scale object recognition. In *Advances in Neural Information Processing Systems*, pages 567–575, 2011. x, 14, 15

[39] O Deniz, M Castrillon, J Lorenzo, and M Hernandez. An incremental learning algorithm for face recognition. *Biometric Authentication, LNCS, Springer*, 2002. 79

[40] Y Ding and A Ross. A comparison of imputation methods for handling missing scores in biometric fusion. *Pattern Recognition*, pages 919–933, 2012. 94

[41] Nemanja Djuric, Liang Lan, Slobodan Vucetic, and Zhuang Wang. Budgetedsvm: A toolbox for scalable svm approximations. *JMLR*, 2013. 61, 62, 70

[42] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The PASCAL Visual Object Classes (VOC) challenge. *International Journal of Computer Vision (IJCV)*, 88(2):303–338, 2010. 14

[43] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. Liblinear: A library for large linear classification. *The Journal of Machine Learning Research*, 9:1871–1874, 2008. xi, 21, 24, 34

[44] Hao Fang, Saurabh Gupta, Forrest Iandola, Rupesh Srivastava, Li Deng, Piotr Dollar, Jianfeng Gao, Xiaodong He, Margaret Mitchell, Platt John, Lawrence Zitnick, and Geoffrey Zweig. From captions to visual concepts and back. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2015. 40

[45] O Fatukasi, J Kittler, and N Poh. Estimation of missing values in multimodal biometric fusion. *IEEE BTAS*, 2008. 94

[46] P Felzenszwalb, R Girshick, D McAllester, and D Ramanan. Object detection with discriminatively trained part based models. *IEEE TPAMI*, 2010. 32

[47] Victor Fragoso, Pradeep Sen, Sergio Rodriguez, and Matthew Turk. EVSAC: accelerating hypotheses generation by modeling matching scores with extreme value theory. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 2472–2479. IEEE, 2013. 14

[48] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Tomas Mikolov, et al. Devise: A deep visual-semantic embedding model. In *Advances in Neural Information Processing Systems*, pages 2121–2129, 2013. 53

[49] Keinosuke Fukunaga. *Introduction to statistical pattern recognition*. Academic press, 1990. 14

[50] J Gama, I Zliobaite, A Bifet, M Pechenizkiy, and M Bouchachia. A survey on concept drift adaptation. *ACM Computing Surveys*, 2014. 4

[51] J Gauvain and C H Lee. Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains. *IEEE Trans. Speech and Audio Processing 2*, 1994. 82

[52] Amir Globerson and Sam Roweis. Nightmare at test time: robust learning by feature deletion. In *Proc. ICML*, pages 353–360. ACM, 2006. 93

[53] Ian Goodfellow, Jonathon Shelns, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*. Computational and Biological Learning Society, 2015. 40, 42, 47

[54] Raghuraman Gopalan, Ruonan Li, and Rama Chellappa. Domain adaptation for object recognition: An unsupervised approach. *ICCV*, 2011. 2

[55] Josh Harguess, Changbo Hu, and J Aggarwal. Fusing face recognition from multiple cameras. *WACV*, 2009. 59, 73

[56] H Hartmann, T Pagano, S Sorooshiam, and R Bales. Confidence builder: evaluating seasonal climate forecasts from user perspectives. *Bull Amer. Met. Soc.*, 2002. 71

[57] S Ho and H Wechsler. Learning from data streams via online transduction. *ICDM Workshop on Temporal Data Mining: Algorithms, Theory and Applications*, 2004. 78

[58] Victoria J Hodge and Jim Austin. A survey of outlier detection methodologies. *Artificial Intelligence Review*, 22(2):85–126, 2004. 12

[59] G Huang, M ramesh, T Berg, and E Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. *University of Massachusetts, Amherst Tech Report*, 2007. xv, xvii, 2, 61, 69, 93, 99, 100, 105

[60] J. J. Hull. A database for handwritten text recognition research. *IEEE TPAMI.*, 16(5):550–554, May 1994. xvii, 93, 99, 100, 105

[61] H. Daume III and D. Marcu. Domain adaptation for statistical classiers. *J. of Artificial Intelligence Research*, 2006. 2, 3, 4

[62] A Jain and A Kumar. Biometrics of next generation: An overview. *Springer*, 2010. 91

[63] Lalit P Jain, Walter J Scheirer, and Terrance E Boult. Multi-class open set recognition using probability of inclusion. In *Computer Vision–ECCV 2014*, pages 393–409. Springer, 2014. x, 11, 14, 15, 18, 23, 33, 34

[64] V Jain and E Learned-Miller. FDDB: A benchmark for face detection in unconstrained settings. *University of Massachusetts, Amherst, Tech. Rep. UM-CS-2010-009*, 2010. 6

[65] Vidit Jain and S Farfade. Adapting classification cascades to new domains. *ICCV*, 2013. 102

[66] Vidit Jain and Erik Learned-Miller. Online domain-adaptation of a pre-trained cascade of classifiers. *CVPR*, 2011. 2, 102

[67] Natraj Jammalamadaka, Andrew Zisserman, Marcin Eichner, Vittorio Ferrari, and C Jawahar. Has my algorithm succeeded? an evaluator for human pose estimators. In *Computer Vision–ECCV 2012*. Springer, 2014. 42

[68] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014. 48, 51

[69] A. J. Joshi, F. Porikli, and N. Papanikolopoulos. Scalable active learning for multi-class image classification. *IEEE TPAMI*, 2012. 66

[70] E Kakula and F Shaw. Towards a mobile biometric test framework - presentation. *International Biometric Performance Conference*, 2012. 59

[71] Ashish Kapoor, Simon Baker, Sumit Basu, and Eric Horvitz. Memory constrained face recognition. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2539–2546. IEEE, 2012. 12, 60, 73

[72] A Khosla, T Zhou, T Malisiewicz, A Efros, and A Torralba. Undoing the damage of dataset bias. *ECCV*, 2012. 3

[73] Tae-Kyun Kim, Bjrn Stenger, Josef Kittler, and Roberto Cipolla. Incremental linear discriminant analysis using sufficient spanning sets and its applications. *IJCV*, 2010. 79

[74] Matej Kristan, Danijel Skocaj, and Ales Leonardis. Incremental learning with gaussian mixture models. *Computer Vision Winter Workshop, Slovenian Pattern Recognition Society*, 2008. 79

[75] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems (NIPS)*, pages 1097–1105, 2012. 12, 14, 22

[76] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems (NIPS)*, pages 1097–1105, 2012. 40, 51

[77] N Kumar, A Berg, P Belhumeur, and S Nayar. Attribute and simile classifiers for face verification. *ICCV*, 2009. 69

[78] Neeraj Kumar, Alexander C. Berg, Peter N. Belhumeur, and Shree K. Nayar. Describable Visual Attributes for Face Verification and Image Search. *IEEE TPAMI*, 33(10):1962–1977, October 2011. 92, 100, 101

[79] Ludmila I Kuncheva, Christopher J Whitaker, and A Narasimhamurthy. A case-study on naïve labelling for the nearest mean and the linear discriminant classifiers. *Pattern Recognition*, 41(10):3010–3020, 2008. 14

[80] I Kuzborskij, F Orabona, and B Caputo. From n to n+1: Multiclass transfer incremental learning. *CVPR*, 2013. 3

[81] A Royer C Lampert. Classifier adaptation at prediction time. *CVPR*, 2015. 93, 94

[82] C Lampert, H Nickisch, and S Harmeling. Attribute-based classification for zero-shot visual object categorization. *IEEE Trans. on Pattern Anal. and Machine Intelligence*, 2013. 110

[83] Pavel Laskov, Christian Gehl, Stefan Krüger, and Klaus-Robert Müller. Incremental support vector learning: Analysis, implementation and applications. *The Journal of Machine Learning Research*, 7:1909–1936, 2006. 13, 14, 64, 82

[84] J Lasserre. Hybrid of generative and discriminative methods for machine learning. *PhD Dissertation, University of Cambridge*, 2008. 79

[85] S Lazebnik, C Schmid, and J Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. *CVPR*, 2006. xii, 31, 32

[86] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proc. of the IEEE*, 86(11):2278–2324, 1998. xvii, 93, 99, 105

[87] Fayin Li and Harry Wechsler. Open set face recognition using transduction. *Pattern Analysis and Machine Intelligence, IEEE Transactions on (T-PAMI)*, 27(11):1686–1697, 2005. 14

[88] Li-Jia Li and Li Fei-Fei. Optimol: automatic online picture collection via incremental model learning. *International Journal of Computer Vision (IJCV)*, 88(2):147–168, 2010. x, 14, 15

[89] S Li and A Jain. Handbook of face recgonition. *Springer*, 2004. 6

[90] S Liao, A Jain, and S Li. Partial face recognition: Alignment-free approach. *IEEE Trans on Pattern Analysis and Machine Intelligence*, 2012. 2

[91] H Lin, C Lin, and R Weng. A note on platts probabilistic outputs for support vector machines. *Machine Learning*, 2007. 67

[92] Yuanqing Lin, Fengjun Lv, Shenghuo Zhu, Ming Yang, Timothee Cour, Kai Yu, Liangliang Cao, and Thomas Huang. Large-scale image classification: fast feature extraction and svm training. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1689–1696. IEEE, 2011. 13

[93] Andreas Weigel Mark Liniger and Christof Appenzeller. The discrete brier and ranked probability skill scores. *Monthly Weather Review, American Meteorological Society*, 2007. 61, 63

[94] Baoyuan Liu, Fereshteh Sadeghi, Marshall Tappen, Ohad Shamir, and Ce Liu. Probabilistic label trees for efficient large scale image classification. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 843–850. IEEE, 2013. x, 14, 15

[95] Marco Loog. Constrained parameter estimation for semi-supervised learning: the case of the nearest mean classifier. In *Machine Learning and Knowledge Discovery in Databases*, pages 291–304. Springer, 2010. 14

[96] Junyang Lu, Jiazhen Zhou, Jingdong Wang, Tao Mei, Xian-Sheng Hua, and Shipeng Li. Image search results refinement via outlier detection using deep contexts. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3029–3036. IEEE, 2012. 12

[97] Markos Markou and Sameer Singh. Novelty detection: a reviewpart 1: statistical approaches. *Signal processing*, 83(12):2481–2497, 2003. 12

[98] B Marlin. Missing data problems in machine learning. *PhD Thesis, University of Toronto*, 2008. 94

[99] Marcin Marszałek and Cordelia Schmid. Constructing category hierarchies for visual recognition. In *Computer Vision–ECCV 2008*, pages 479–491. Springer, 2008. x, 14, 15

[100] J Matai, A Irturk, and R Kastner. Design and implementation of an fpga-based real-time face recognition system. *Intern Symp on Field-Programmable Custom Computing Machines*, 2011. 60

[101] C. McCool, R. Wallace, M. McLaren, L. El-Shafey, and S. Marcel. Session variability modelling for face authentication. *IET Biometrics*, 2013. 2

[102] T Mensink, J Verbeek, F Perronnin, and G Csurka. Metric learning for large scale image classification: Generalizing to new classes at near-zero cost. *ECCV*, 2012. 44

[103] Thomas Mensink, Jakob Verbeek, Florent Perronnin, and Gabriela Csurka. Distance-based image classification: Generalizing to new classes at near-zero cost. *Pattern Analysis and Machine Intelligence, IEEE Transactions on (T-PAMI)*, 35(11):2624–2637, 2013. x, 12, 13, 14, 15, 16, 20, 22

[104] Thomas Mensink, Jakob Verbeek, Florent Perronnin, and Gabriela Csurka. Distance-based image classification: Generalizing to new classes at near-zero cost. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(11):2624–2637, 2013. 44, 47

[105] T Ming-Huang and V Kecman. Bias term b in svms again. *ESANN*, pages 441–448, 2004. 97

[106] Andrew Ng and Michael Jordan. On discriminative vs generative: A comparison of logistic regression and naive bayes. *NIPS*, 2001. 79

[107] Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*. IEEE, 2015. 40, 42, 47, 50, 51

[108] A Niculescu-Mizil and R Caruana. Predicting good probabilities with supervised learning. *ICML*, 2006. 61, 62, 67

[109] T Ojala, M Pietikainen, and T Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE TPAMI*, 2002. xii, 31, 32

[110] Maxime Oquab, Leon Bottou, Ivan Laptev, and Josef Sivic. Is object localization for free? weakly-supervised learning with convolutional neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. 54

[111] Vicente Ordonez, Vignesh Jagadeesh, Wei Di, Anurag Bhardwaj, and Robinson Piramuthu. Furniture-geek: Understanding fine-grained furniture attributes from freely associated text and tags. In *Applications of Computer Vision (WACV), 2014 IEEE Winter Conference on*, pages 317–324. IEEE, 2014. 11, 40

[112] S Ozawa, S Lee-Toh, S Abe S Pang, and N Kasabov. Incremental learning for online face recognition. *IJCNN*, 2005. 59, 62

[113] S J Pan and Q Yang. A survey on transfer learning. *IEEE Trans on Knowledge and Data Engg*, 2010. 3, 4

[114] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. 62, 70

[115] A Pentina and C Lampert. A pac-bayesian bound for lifelong learning. *ICML*, 2014. 110

[116] A Perina, M Cristani, U Castellani, V Murino, and N Jojic. Free energy score spaces: Using generative information in discriminative classifiers. *IEEE TPAMI*, 2012. 77

[117] Florent Perronnin, Zeynep Akata, Zaid Harchaoui, and Cordelia Schmid. Towards good practice in large-scale learning for image classification. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3482–3489. IEEE, 2012. 12, 22

[118] N Pinto, J DiCarlo, and D Cox. How far can you get with a modern face recognition test set using only simple features? *CVPR*, 2009. 3, 59

[119] J Platt. Probabilities for sv machines. *Advances in Large Margin Classifiers*, 1999. xv, 61, 62, 65, 66, 67, 68

[120] N Poh, D Windrige, V Mottl, A Tatarchuk, and A Eliseyev. Addressing missing values in kernel-based multimodal biometric fusion using neutral point substitution. *IEEE Trans. on Info Forensics and Security*, 2010. 94

[121] Andrzej Pronobis, Luo Jie, and Barbara Caputo. The more you learn, the less you store: Memory-controlled incremental svm for visual place recognition. *Image and Vision Computing*, 28(7):1080–1097, 2010. 13

[122] J Quionero-Candela, M Sugiyama, A Schwaighofer, and N Lawrence. Dataset shift in machine learning. *MIT Press*, 2009. 2, 3, 4

[123] D Reynolds, T Quatieri, and R Dunn. Speaker verification using adapted gaussian mixture models. *Digital Signal Processing*, 2000. 82

[124] Marko Ristin, Matthieu Guillaumin, Juergen Gall, and Luc Van Gool. Incremental learning of ncm forests for large-scale image classification. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 3654–3661. IEEE, 2014. x, 12, 13, 14, 15, 22, 44, 47

[125] R Roscher, W Forstner, and B Waske. I2vm: Incremental import vector machines. *Image and Vision Computing*, 2012. 77

[126] Donald Rumsfeld. *Known and unknown: a memoir*. Penguin, 2011. 40

[127] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge, 2014. 30

[128] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, pages 1–42, April 2015. 51, 52

[129] M Saar-Tsechansky and F Provost. Handling missing values when applying classification models. *J. of Machine Learning Research*, 2007. 94

[130] C Sanderson and K Paliwal. Fast features for face authentication under illumination direction changes. *Pattern Recognition Letters*, 2003. 84

[131] A Sapkota and T Boult. Large scale unconstrained open set face database. *IEEE BTAS*, 2013. 77, 87

[132] W Scheirer, S Anthony, K Nakayama, and D Cox. Perceptual annotation : Measuring human vision to improve computer vision. *IEEE TPAMI*, 2014. 6

[133] W Scheirer, A Bendale, and T Boult. Predicting biometric facial recognition failure with similarity surfaces and support vector machines. *CVPR Biometrics Workshop*, 2008. 93, 101

[134] W Scheirer, N Kumar, P Belhumeur, and T Boult. Multi-attribute spaces: Calibration for attribute fusion and similarity search. *CVPR*, 2012. 61, 62, 66, 67, 92

[135] W Scheirer, A Rocha, R Micheals, and T Boult. Robust fusion: Extreme value theory for recognition score normalization. *ECCV*, 2010. 63

[136] W Scheirer, A Rocha, R Micheals, and T Boult. Meta-recognition: The theory and practice of recognition score analysis. *IEEE TPAMI*, 2011. xv, 4, 5, 61, 62, 63, 66, 67, 68, 79

[137] Walter Scheirer, Anderson Rocha, Ross Michaels, and Terrance E. Boult. Meta-Recognition: The Theory and Practice of Recognition Score Analysis. *IEEE TPAMI*, 33(8):1689–1695, August 2011. 101

[138] Walter J Scheirer, Anderson de Rezende Rocha, Archana Sapkota, and Terrance E Boult. Toward open set recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(7):1757–1772, 2013. 40, 42, 43, 52

[139] Walter J Scheirer, Lalit P Jain, and Terrance E Boult. Probability models for open set recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36(11):2317–2324, 2014. 40, 42, 43, 50, 51

[140] Walter J Scheirer, Anderson Rocha, Ross J Micheals, and Terrance E Boult. Meta-recognition: The theory and practice of recognition score analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(8):1689–1695, 2011. libMR code at `http://metarecognition.com`. 42, 44, 45

[141] Walter J. Scheirer, Anderson Rocha, Archana Sapkota, and Terrance E. Boult. Towards open set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, 36, July 2013. x, xi, 4, 11, 12, 14, 15, 16, 18, 21, 22, 23, 24, 34, 92

[142] W.J. Scheirer, L.P. Jain, and T.E. Boult. Probability models for open set recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on (T-PAMI)*, 36(11):2317–2324, Nov 2014. x, 11, 14, 15, 18, 19, 34

[143] D Sculley. Combined regression and ranking. *ACM SIGKDD*, 2010. 62

[144] P Sermanet, D Eigen, Z Zhang, M Mathieu, R Fergus, and Y LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *ICLR*, 2014. 20

[145] G Shakhnarovich, J Fisher, and T Darrell. Face recognition from long term observations. *ECCV*, 2002. 5

[146] Shai Shalev-Shwartz, Yoram Singer, Nathan Srebro, and Andrew Cotter. Pegasos: Primal Estimated sub-GrAdient SOlver for SVM. *ICML*, 2007. 14, 73, 77

[147] Wen Shaoa, Wen Yangab, and Gui-Song Xiab. Extreme value theory-based calibration for the fusion of multiple features in high-resolution satellite scene classification. *International Journal of Remote Sensing*, 2013. 63

[148] K Simonyan and A Zisserman. Very deep convolutional networks for large-scale image recognition. *arxiv*, 2014. 14

[149] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep fisher networks for large-scale image classification. In *Advances in neural information processing systems*, pages 163–171, 2013. 22

[150] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large scale image recognition. In *International Conference on Learning Representations*. Computational and Biological Learning Society, 2015. 51

[151] Marina Skurichina and Robert PW Duin. Bagging, boosting and the random subspace method for linear classifiers. *Pattern Analysis & Applications*, 5(2):121–135, 2002. 14

[152] A Smola. Learning with kernels. *PhD Dissertation, Technische Universitat Berlin, Berlin, Germany*, 1998. 7

[153] A J Smola, S V N Vishwanathan, and T Hofmann. Kernel methods for missing variables. In *Proc. Wksp on Articial Intelligence and Statistics*, 2005. 101

[154] Richard Socher, Milind Ganjoo, Christopher D Manning, and Andrew Ng. Zero-shot learning through cross-modal transfer. In *Advances in neural information processing systems*, pages 935–943, 2013. 53

[155] Richard Socher, Christopher D Manning, and Andrew Y Ng. Learning continuous phrase representations and syntactic parsing with recursive neural networks. In *Proceedings of the NIPS-2010 Deep Learning and Unsupervised Feature Learning Workshop*, pages 1–9, 2010. 40, 52

[156] N Syed, H Liu, and K Sung. Incremental learning with support vector machines. *IJCAI*, 1999. 77

[157] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*. IEEE, 2015. 40

[158] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations*. Computational and Biological Learning Society, 2014. 42, 47

[159] R Szeliski. Computer vision: Algorithms and applications. *Springer*, 2012. 1, 5

[160] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 1701–1708. IEEE, 2014. 11, 40

[161] X Tan and B Triggs. Enhanced local texture feature sets for face recognition under difficult lighting conditions. *IEEE Trans. on Image Processing*, 2010. 84

[162] Robert Tibshirani, Trevor Hastie, Balasubramanian Narasimhan, and Gilbert Chu. Diagnosis of multiple cancer types by shrunken centroids of gene expression. In *Proceedings of the National Academy of Sciences*. NAS, 2002. 44

[163] A Torralba and A Efros. Unbiased look at dataset bias. *CVPR*, 2011. 3

[164] Vladimir Vapnik. The nature of statistical learning theory. *Springer Verlag*, 1995. 64, 65, 81

[165] Cor J Veenman and Marcel JT Reinders. The nearest subclass classifier: A compromise between the nearest mean and nearest neighbor classifier. *Pattern Analysis and Machine Intelligence, IEEE Transactions on (T-PAMI)*, 27(9):1417–1429, 2005. 14

[166] Cor J Veenman and David MJ Tax. A weighted nearest mean classifier for sparse subspaces. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 1171–1176. IEEE, 2005. 14

[167] Alexander Vezhnevets and Vittorio Ferrari. Object localization in imagenet by looking out of the window. In *Proceedings of the British Machine Vision Conference,(BMVC)*, 2015. 54

[168] R Wallace, M McLaren, C McCool, and S. Marcel. Cross-pollination of normalisation techniques from speaker to face authentication using gaussian mixture models. *IEEE Trans. Information Forensics and Security*, 2010. 80

[169] Peng Wang, Qiang Ji, and James Wayman. Modeling and predicting face recognition system performance based on analysis of similarity scores. *IEEE TPAMI*, 2007. 4

[170] Z Wang, K Crammer, and S Vucetic. Breaking the curse of kernelization: Budgeted stochastic gradient descent for large-scale svm training. *JMLR*, 2012. 62, 64, 65, 70, 73

[171] Zhuang Wang, Koby Crammer, and Slobodan Vucetic. Multi-class pegasos on a budget. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 1143–1150, 2010. 14

[172] M Wilber, E Rudd, B Heflin, Y Lui, and T Boult. Exemplar codes for facial attributes and tattoo recognition. *WACV*, 2014. xvi, 92, 93

[173] M Wilber, W Scheirer, P Leitner, B Heflin, J Zott, D Reinke, D Delaney, and T Boult. Animal recognition in the mojave desert: Vision tools for field biologists. *WACV*, 2013. 93

[174] D S Wilks. On the combination of forecast probabilities for consecutive precipitation periods. *Academic Press*, 1995. 61, 63, 71

[175] Lior Wolf, Tal Hassner, and Itay Maoz. Face recognition in unconstrained videos with matched background similarity. *CVPR*, 2011. xvi, 59, 86, 87

[176] J Yan, Z Lei, D Yi, and S Li. Towards incremental and large scale face recognition. *IEEE IJCB*, 2011. 59, 62, 79

[177] Yi-Hsuan Yang, Chia-Chu Liu, and Homer H Chen. Music emotion classification: a fuzzy approach. In *Proceedings of the 14th annual ACM international conference on Multimedia*, pages 81–84. ACM, 2006. 14

[178] Tom Yeh and Trevor Darrell. Dynamic visual category learning. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008. x, 12, 13, 15

[179] S Yin, R Rose, and P Kenny. A joint factor analysis approach to progressive model adaptation in text independent speaker verication. *IEEE Trans. Audio, Speech and Language Processing*, 2007. 80

[180] Jason Yosinski, Jeff Clune, Anh Nguyen, Thomas Fuchs, and Hod Lipson. Understanding neural networks through deep visualization. In *International Conference on Machine Learning, Workshop on Deep Learning*, 2015. 47

[181] B Zadrozny and C Elkan. Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. *ICML*, 2001. 62

[182] Hao Zhang, Alexander C Berg, Michael Maire, and Jitendra Malik. Svm-knn: Discriminative nearest neighbor classification for visual category recognition. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 2126–2136. IEEE, 2006. 13

[183] Peng Zhang, Jiuling Wang, Ali Farhadi, Martial Hebert, and Devi Parikh. Predicting failures of vision systems. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014. 42, 44

[184] W Zhao, R Chellappa, P. Jonathon Phillips, and A Rosenfeld. Face recognition: A literature survey. *ACM Comput. Survey*, 2003. 59

[185] W Zheng, S Gong, and T Xiang. Quantifying and transferring contextual information in object detection. *IEEE TPAMI*, 2012. 3