MINING WEB IMAGES FOR CONCEPT LEARNING

A THESIS

SUBMITTED TO THE DEPARTMENT OF COMPUTER ENGINEERING

AND THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

By

Eren Golge

August, 2014

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Asst.Prof.Dr. Pinar Duygulu(Advisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Asst.Prof.Dr. Oznur Tastan

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Asst.Prof.Dr. Sinan Kalkan

Approved for the Graduate School of Engineering and Science:

Prof. Dr. Levent Onural
Director of the Graduate School

ABSTRACT

MINING WEB IMAGES FOR CONCEPT LEARNING

Eren Golge
M.S. in Computer Engineering
Supervisor: Asst.Prof.Dr. Pinar Duygulu
August, 2014

We attack the problem of learning concepts automatically from noisy Web image search results. The idea is based on discovering common characteristics shared among category images by posing two novel methods that are able to organise the data while eliminating irrelevant instances. We propose a novel clustering and outlier detection method, namely Concept Map (CMAP). Given an image collection returned for a concept query, CMAP provides clusters pruned from outliers. Each cluster is used to train a model representing a different characteristics of the concept. One another method is Association through Model Evolution (AME). It prunes the data in an iterative manner and it progressively finds better set of images with an evaluational score computed for each iteration. The idea is based on capturing discriminativeness and representativeness of each instance against large number of random images and eliminating the outliers. The final model is used for classification of novel images. These two methods are applied on different benchmark problems and we observed compelling or better results compared to state of art methods.

Keywords: weakly-supervised learning, concept learning, rectifying self-organizing map, association with model evolution, clustering and outlier detection, conceptmap, attributes, object recognition, scene classification, face identification, feature learning
.

ÖZET

AĞ İMGELERİNİN KONSEPT ÖĞRENME AMACIYLA İŞLENMESİ

Eren Golge
Bilgisayar Mühendisliği, Yüksek Lisans
Tez Yöneticisi: Asst.Prof.Dr. Pinar Duygulu
Ağustos, 2014

Bu calışmada görsel konseptlerin otomatik olarak internet kaynaklı imgeler kullanılarak öğrenilmesi üzerine çalışılmıştır. Sunulan iki yeni yöntem ile bir konsept için edinilmiş imge koleksyonundaki ortak özellikleri kullanarak, ilgisiz imgeleri elemek ve imgeleri görsel bütünlük içinde gruplanmak amaçlanmıştır. İlk olarak, yeni bir veri öbekleme ve ilintisiz veri eleme yöntemi Konsept Haritası (Concept Map - CMAP) sunulmuştur. CMAP verileri öbeklere ayırırken, ilgisiz verileri bu öbeklere olan benzerliklerine göre eler. Daha sonrasında, CMAP'in bir konsept için ürettiği her bir veri öbeğinden, konseptin değişik bir alt kümesini tanımlayan, birer model öğrenilir. Diğer bir yöntem, Model Evrimi ile Eşleme (Association through Model Evolution - AME), imgelerin rasgele alınmış büyük bir imge kümesi ile farklarını yinelemeli bir yöntem ile ölçer. Bu ölçümlere dayanarak, her yinelemede, yeni bir grup ilintisiz imge elenir. AME her bir imgenin, rasgele alınmış büyük bir imge kümesine karşı, ait olduğu konsept için ayrımsallık ve temsil edebilirlik özelliklerini teşhis eder. Bu özellikleri göz önüne alarak, ilitisiz imgeleri bulur. En son aşamada, temizlenmiş imge setleri üzerinden hesaplanmış modeller ile, yeni imgeler üzerinde konseptsel sınıflandırma yapılır. Sunulan iki yeni yöntem de bilindik veri setleri ve problemler üzerinde sınanmıştır. Sonuçlar bilindik en iyi yötemler ile kıyaslanabilir değerler vermektedir.

Anahtar sözcükler: zayıf güdümlü öğrenme, konsept öğrenme, doğrulayıcı kendini örgütleyen devre, model evrimi ile ilişkilendirme, öbekleme ve ilintisiz veri bulma, konsept haritası , öznitelikler, obje öğrenimi, sahne sı nıflandırması, yüz tanıma, öznitelik öğrenimi.

Acknowledgement

”This thesis is dedicated to the great leader Mustafa Kemal Atatürk.”

Here I am crumbling an important milestone with the assitance of unique people. I should thank those people who watch over me and who made this thesis viable.

My meaningless greatitudes, regarding what they’ve done, are served to my parents Serpil Golge and Halit Golge who freed me of any bitterness throughout the life.

My dear love was the source of living. I am appriciated to Hazal Demiral owing to her great effort so as to make me a living organism.

Of course, this thesis would not be possible without my advisor Asst. Prof. Dr. Pinar Duygulu. I thank her for the guadiance and assistance.

”Friendship is friendship”. I serve great thanks to my friends; Anil Armagan, Ahmet Iscen, Ilker Sarac, Fadime Sener, Caner Mercan, Can F. Koyuncu, Acar Erdinc, Sermetcan Baysal, Gokhan Akcay and the all the people shared their valuable time. I also specially thank to Atilla Abi who makes our offices habitable.

Not but not least, I should also thank people all around the world and in the history, living not only for their greeds but the excellence of the others and willing a world living life in peace. Those are the people that insipe me to study and work hard.

This thesis is partially supported by TUBITAK project with grant no 112E174 and CHIST-ERA MUCKE project.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The need for manually labelled data continues to be one of the most important limitations in large scale recognition. Alternatively, images are available on the Web in vast amounts, even though the precise labels are missing. This fact recently attracted many researchers to build (semi-)automatic methods to learn from web data collected for a given query targetting a visual concept category. The aim is to appraise the rich but noisy crowd of web images to learn visual concepts that are more robust for various vision tasks. However, there are several challenges that makes the data collections gathered from web difficult than the hand-crafted datasets. The first is visual difficulties as the cause of variations and artificial effects, the second is overlapping verbal correspondence of the visual concepts and the third is the irrelevant images to the targeted concept category .

With the advent of huge social networks in Internet, there are many images are used as a part of daily communication between people. However those images are generally deformed by the artificial effects to make them more attractive. This causes visual variations, hence an automatic learning system needs to be tolerating all such complex visual effects on the images.

All the web images are in a weakly-labelled setting in which their category names are roughly prescribed by the given query. This uncertainty contrives another problem. Two different visual concepts might be associated with a same verbal correspondence

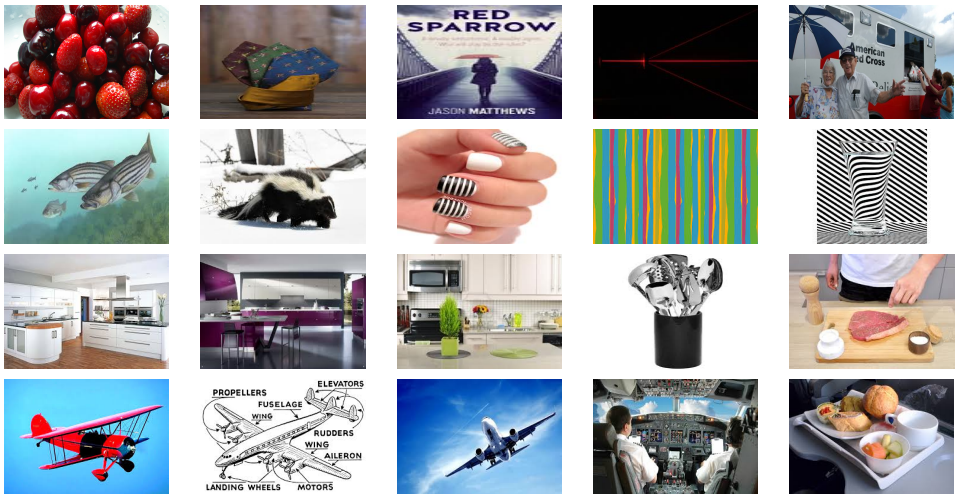


Figure 1.1: Example Web images, in the row order, collected for query keywords red, striped, kitchen, plane. Even in the relevant images, the concepts are observed in different forms requiring grouping and irrelevant ones to be eliminated.

which is called "polysem". Therefore, even though images shares the same verbal correspondence, they need be seperated where each of the groups represents a distinct visual essence of its category.

Since usually images are gathered based on the surrounding text, the collection is very noisy with several visually irrelevant images as well as variety of images corresponding to different characteristic properties of the concept. The irrelevant instances and the sub-grouping obstruct learning salient concept models from the raw collection. Then, for learning a salient set of concept models, we need to prune the data from irrelevant instances and group the images into sub-categories to handle the intrinsic variations.

For the queried data for full-automatic learning of concepts, here we propose two novel methods to obtain adequate set of images with the explained challenges are solved. Our intuition is that, given a concept category by a query, although the list of images returned include irrelevant ones, there will be common characteristics shared among subset of images that are different than the other concept categories. Our first method **ConceptMap (CMAP)** tries to capture sub-grouping in the data as it removes irrelevant images. Second method **Association through Model Evolution (AME)**

looks to problem from different perceptive and accumulate a coherent set of images by eliminating the poor ones insoecting the measure of distinctiveness against random images collected through Internet.

To retain only the relevant images that describe the concept category correctly, **CMAP** detects outliers instances by first grouping them into clusters and then uncover the poor clusters against to salient ones. it also finds the spurious instances in the salient clusters. After **CMAP** we end up with a supposedly good set of instances organized into clusters. Each latched cluster might also be thought as a sub-concept of the given concept category.

The other possible solution is presented by **AME** that evaluates quality of category images against vast amount of random images gathered from the web. The idea is to highlight the differences of the correct category images in relation to the random images per se the individual representativeness of the category properties. **AME** exploits this intuition with an iterative refining so as to find the correct set of category instances enpowering prosperous final concept models. Our models evolve through consecutive iterations to associate the category name with the correct set of images. These models are then used for labelling concepts on novel datasets. **AME** remove outlier images, while retaining the diversity as much as possible.

Chapter 2

Background

CMAP and **AME** are related to several studies in the literature. Here, we will discuss the most relevant ones by grouping them into three sections. First we give a common review of methods releated to both **CMAP** and **AME**. Then two sections are given for each method **CMAP** and **AME**. Then the last section discusses the particular differences and contributions of our methods. Notice that, reviewing the huge literature on object and scene recognition is far from the scope of this study since the concern of this thessi is to deal with noisy image collections.

Harvesting web for concept learning Several recent studies tackle the problem of building qualified training sets by using images returned from image search engines [2, 12, 13, 14, 7, 15].

Fergus *et al.* [2] propose a pLSA based method in which the spatial information is also incorporated in the model. They collected noisy images from Google as well as a validation set which consists of top five images collected in different languages which was used to pick the best topics. They experimented classification on subsets of Caltech and Pascal datasets, and re-ranking of Google results. The main drawback of the method is the dependency to the validation set. Moreover, the results indicate that the variations in the categories are not handled well.

Berg and Forsyth [12] use visual features and surrounding the text for collecting

animal images from web. Visual exemplars are obtained through clustering text. They require the relevant clusters to be identified manually, as well as an optional step of eliminating irrelevant images in clusters. Note that these steps are automatically performed in our proposed methods.

Li and Fei-Fei [7] present the OPTIMOL framework for incrementally learning object categories from web search results. Given a set of seed images, a non-parametric latent topic model is applied to categorise collected web images. The model is iteratively updated with the newly categorised images. To prevent over specialised results, a set of cache images with high diversity are retained at each iteration.

While the main focus is on the analysis of the generated collection, they also compared the learned models on the classification task on the dataset provided in [2]. The validation set is used to gather the seed images. The major drawback of the method is the high dependency to the quality of the seed images and the risk for concept drift during iterations.

Schroff *et al.* [15] first filters out the abstract images (drawings, cartoons, etc.) from the resulting set of images collected through text and image search in Google for a given category. Then, they use text and metadata surrounding the images to re-rank the images. Finally they train a visual classifier by sampling from the top ranked images as positives and random images from other categories as negatives. Their method highly depends on the filtering and text-based re-ranking as shown with the lower performances obtained by visual only based classifier.

Berg and Berg [13] find iconic images that are the representatives of the collection given a query concept. First they select the images with objects are distinct from background. Then, the high ranked images are clustered using k-medoids to consider centroid images as iconic. Due to the elimination of several images in the first step it is likely that helpful variations in the dataset are removed. Moreover, possible irrelevant instances are not targeted in this work. It makes the method success strongly releated to quality of image source.

Fan *et al.* [14] propose a graph theoretical method which is difficult to apply large scale problems because of space and time complexity.

NEIL [16] is the most similar study to ours. In NEIL, large numbers of models are learned automatically for each concept and iteratively these models are used for refining the data. It works on attributes, objects and scenes as well and localises objects in the images. The main bottle-neck of NEIL is the examplar approach that they use for learning models for each individual image taken from the web, even the image is useless. It makes the system very slow and time-consuming. Therefore NEIL demands high computational power for reasonable run-time.

Learning dicriminative and representative instances or parts: Our methods are also related to the recently emerged studies in discovering discriminativeness. [17, 18, 19, 18, 20, 21, 22, 23, 24, 25, 26]. In these studies weakly labeled datasets are leveraged for learning visual instances that are representative and discriminative. In [27], discriminative patches in images are discovered through an iterative method which alternates between clustering and training discriminative classifiers. Li *et al.* [25] solve same problem with multiple instance learning. [20] and [24] apply the idea to scene images for learning discriminative properties by embracing the unsupervised exemplar models. Moreover [21] enhances the unsupervised learning schema by more robust alternation of Mean-Shift clustering algorithm. Discriminativeness idea is also applied to video domain by [22]. We aim to discover the visual cues or the entire images representing the collected data in the best way. However, **CMAP** also want to keep the variations in the concept for allowing intra-class variations and multiple senses to be modelled through different sub-groups. **AME** thrives on high dimensional representations of instances. Thus, each category is linearly seperable from the others regardless of the grouping in each category.

2.1 Releated Work for Concept Map

Learning attributes and mid-level representations: The use of attributes has been the focus of many recent studies [28, 29, 35, 30, 31, 32, 33, 34, 35, 36]. Most of the methods learn attributes in a supervised way [37, 38] with the goal of describing object categories. Not only semantic attributes, but classemes [39] and implicit attributes [40] have also been studied. We focus on attribute learning independent of

object categories and learn different intrinsic properties of semantic attributes through models obtained from separate clusters that are ultimately combined in a single semantic. In [37], Farhadi *et al.* learn complex attributes (shape, materials, parts) in a fully supervised way focusing on recognition of new types of objects. In [38], for human labelled animal categories, semantic attribute annotations available from studies in cognitive science were used in a binary fashion for zero-shot learning. [41] learns a set of independent classifiers for different sets of attributes, including the ones that describe the overall image as well as the objects, to be used as semantic image descriptors for object classification. Images are trained on Google images with the false positives rejected manually. Torresani *et al.* [39] introduce classemes, attributes that do not have specific semantic meanings, but meanings expected to emerge from intersections of properties, and they obtain training data directly from web image search. Rastegari *et al.* [40] propose discovering implicit attributes that are not necessarily semantic but preserve category-specific traits through learning discriminative hyperplanes with max-margin and locality sensitive hashing criteria. Learning semantic appearance attributes, such as colour, texture and shape, on ImageNet dataset is attacked in [5] relying on image level human labels using AMT for supervised learning. We learn attributes from real world images collected from web with no additional human effort for labelling. Another study on learning colour names from web images is proposed in [3] where a pLSA based model is used for representing the colour names of pixels. Similar to ours, the approach of Ferrari and Zisserman [42] considers attributes as patterns sharing some characteristic properties where basic units are the image segments with uniform appearance. We prefer to work on patch level alternative to pixel level which is not suitable for region level attributes such as texture; image level which is very noisy; or segment level which is difficult to obtain clearly. Based on McRae *et al.*'s norms [43], Silberer *et al.* [44] use large number of attributes for representing large number of concepts with the goal of developing distributional models that are applicable to many words. While most of the studies focus on attributes for object categorisation, one of the early works by Vogel and Schiele [45] use attributes such as grass, rocks or foliage for categorisation of natural scenes. On the other hand, [46] uses objects as attributes of scenes for scene classification. Images are represented by their responses to a large number of object detectors/filters. From a different perspective,

the work of Quattoni *et al.* [47] makes use of images with captions to learn visual representations that reflects the semantic content of the images through utilising auxiliary training data and structural learning.

Other methods on outlier detection with SOM: [48, 49] utilise the habitation of the instances. Frequently observed similar instances excite the network to learn some regularities and divergent instances are observed as outliers. [50] benefits from weights prototyping the instances in a cluster. Thresholded distance of instances to the weight vectors are considered as indicator of being outlier. In [51], aim is to have different mapping of activated neuron for the outlier instances. The algorithm learns the formation of activated neurons on the network for outlier and inlier items with no threshold. It suffers from the generality, with its basic assumption of learning from network mapping. LTD-KN [52] performs Kohonen learning rule inversely. An instance activates only the winning neuron as in the usual SOM, but LTD-KN updates winning neuron and its learning windows decreasingly.

These algorithms only eliminate outlier instances ignoring outlier clusters. **CMAP** finds outlier clusters as well as the outlier instances in the salient clusters. Another difference of **CMAP** is the computation cost. Most of outlier detection algorithms model the data and iterate over the data again to label outliers. It is not suitable for large scale data. **CMAP** has the ability to detect outlier clusters and the items all in the learning phase. Thus, there is no need for learning a model of the data first, then detecting outliers, it is all done in a single pass in our method.

2.2 Releated Work for Association through Model Evolution

Naming faces using weakly-labeled data: **AME** is proposed as a generic method for data refining of noisy data collections. However, in this work, it is specifically used for face indeitification problem. It uses raw set of queired web images and it iteratively captivates clearner image collection and the identification models are trained over the final collection. This idea has some of task specific retroespective that we discuss some

of the important ones starting from here.

The work of Berg *et al*.is one of the first attempts in labelling large number of faces from weakly-labeled web images [53, 54] with the "Labeled Faces in the Wild" (LFW) dataset introduced. It is assumed that in an image at most one face can correspond to a name, and names are used as constraints in clustering faces. Appearances of faces are modelled through Gaussian mixture model with one mixture per name. In [53], k-PCA is used to reduce the dimensionality of the data and LDA is used for projection. Initial discriminant space learned from faces with a single associated name is used for clustering through a modified k-means. Better discriminants are then learned to re-cluster. In [54] face name associations are captured through an EM based approach.

For aligning names and faces in an (a)symmetric way, Pham *et al*.[55] cluster the faces using a hierarchical agglomerative clustering method. They use the constraint that faces in an image cannot be in the same cluster. They then use an EM based approach for aligning names and faces based on probability of reoccurrences. They use a 3D morphable model for face representation. They introduce the picturedness and namedness: the probability of a person being in the picture based on textual info, and being in the text based on visual info.

Ideally, there should be a single cluster per person. However, these methods are likely to produce clusters with several people mixed in, and multiple clusters for the same person.

In [56, 57], Ozkan and Duygulu consider the problem as retrieving faces for a single query name, and then pruning the set from the irrelevant faces. A similarity graph is constructed where the nodes are faces, and edges are the similarity between faces. With the assumption that the most similar subset of faces will correspond to the queried name, the densest component in the graph is sought using a greedy method. In [58], the method of [56, 57] is improved by introducing the constraint for each image to contain a single instance of the queried person and replacing the threshold in constructing the binary graphs with assigning non-zero weights to k nearest neighbours. The authors further generalised the graph based method for for multi-person naming, as well as null assignments. They propose a min-cost max-flow based approach to optimise face name assignments under unique matching constraints. In [59]. a logistic discriminant

approach which learns the metric from pairs of faces is proposed for identification of faces. As another approach for face identification, they propose a method where the probability of two faces belonging to the same class is computed in a nearest neighbour based approach.

In [60] face-name association problem is tackled as a multiple instance learning problem over pairs of bags. Detected faces in an image is put into a bag, and names detected in the caption are put into the corresponding set of labels. A pair of bags is labeled as positive if they share at least one label, and negative otherwise. The results are reported on Labelled Yahoo! News dataset which is obtained through manually annotating and extending LFW dataset. In [61], it is shown that the performance of graph-based and generative approaches for text-based face retrieval and face-name association tasks can be improved with the incorporation of logistic discriminant based metric learning (LDML) [59].

Kumar *et al*.[62] introduced attribute and smile classifiers for verifying the identity of faces. For describable aspects of visual appearance, binary attribute classifiers are trained with the help of AMT. Moreover, simile classifiers are trained to recognise the similarity of faces to specific reference people. Pub-Fig, dataset of public figures on the web, is presented alternative to LFW with larger number of individuals each having more instances.

Pham et al. [63] use the idea of label propogation, to name unlabelled faces in videos starting from a set of seed labeled faces. Together with visual similarities, they also make use of constraints for assigning a single name to face tracks and not labelling two faces in a single frame with the same name.

In [61], the concept of "friends" is introduced for query expansion. The names of the people frequently co-occurring with the queried person is used for extending the set of faces, and resemblance of the faces to the friends is used for better modelling of the query person.

Recently, PubFig83, a subset of PubFig dataset with near-duplicates eliminated and individuals with large number of instances are selected, is provided for face identification task [11]. Inspired from biological systems, Pinto *et al.* [11] consider V1-like

features and introduce both single- and multi-layer feature extraction architecture followed by LinearSVM classifier. In [64], person specific partial least squares (PS-PLS) approach is presented to generate subspaces for familiar faces, such as celebrities.

[65] define the open-universe face identification problem as identifying faces with one of the labeled categories in a dataset including distractor faces that do not belong to any of the labels. In [66], the authors combine PubFig83, as being the set of labeled individuals, and LFW, as being the set of distractors. On this set, they evaluate a set of identification methods including nearest neighbour, SVM, sparse representation based classification (SRC) and its variants, as well as linearly approximated SRC that they proposed in [65].

Other recent work include [67] where Fisher vectors on densely sampled SIFT features are utilised. Large margin dimensionality reduction is used to reduce high dimensionality.

2.3 Discussions

Unlike most of the recent studies that focus on learning specific types of categories from noisy images downloaded from web (such as objects [2, 7], scenes [68], and attributes [3, 42]) we do not restrict ourselves with a single domain but propose a general framework which is applicable to many domains from low level attributes to high level concepts, such as objects, and scenes.

As in [7, 16] we address three main challenges in learning visual concepts from noisy web results: (i) **Irrelevant images** returned by the search engines due to keyword based queries on the noisy textual content. (ii) **Intra-class variations** within a category resulting in multiple groups of relevant images. (iii) **Multiple senses** of the concept.

With **CMAP**, we aim to answer not only "which concept is in the image?", but also "where the concept is?" as in [16] . Local patches are considered as basic units to solve the localisation as well as to eliminate background regions.

We use only visual informations extracted from the images gathered for a given query word, and do not require any other additional knowledge such as surrounding text, metadata or GPS-tags [15, 12, 69].

The collection returned from web search is used in its **pure** form without requiring any prior supervision (manual or automatic) for cleaning, selection or organisation of the data [12, 15, 7].

AME presents a very solid and novel idea different than the all litreture by taking the advantage of unannotated random images that are far to much in web.

Chapter 3

Concept Maps

Concept Maps (CMAP) is inspired from the well-known Self Organising Maps (SOM) [70]. In the following, SOM will be revisited briefly, and then CMAP will be described. Table 3.1 summarises the notation used.

3.1 Revisiting Self Organising Maps (SOM)

Intrinsic dynamics of SOM are inspired from developed animal brain where each part is known to be receptive to different sensory inputs and which has a topographically organized structure [70]. This phenomena, i.e. "receptive field" in visual neural systems [71], is simulated with SOM, where neurons are represented by weights calibrated to make neurons sensitive to different type of inputs. Elicitation of this structure is furnished by competitive learning approach.

Consider input $X = \{x_1, .., x_M\}$ with M instances. Let $N = \{n_1, ..., n_K\}$ be the locations of neuron units on the SOM map and $W = \{w_1, ..., w_K\}$ be the associated weights. The neuron whose weight vector is most similar to the input instance x_i is called as the winner and denoted by $\hat{v}$. The weights of the winner and units in the neighbourhood are adjusted towards the input at each iteration t with delta learning

Table 3.1: Notation used for Concept Map.

$Notation$	$Description$
$X = \{x_1, .., x_M\}$	Set of M instances
x_i	an instance
$N = \{n_1, .., n_K\}$	Locations of K SOM units
n_i	Location of i^{th} SOM unit
$W = \{w_1, .., w_K\}$	Set of weight vectors of K SOM units
w_i	Weight vector of i^{th} SOM unit
$w_{\hat{v}}$	Winner SOM unit's weight vector
$h(n_i, n_{\hat{v}} : \epsilon^t, \sigma^t)$	Window function
ϵ	Learning rate
σ	Neighbour effect coefficient
$\hat{v}$	winner SOM unit
$E = \{e_1, .., e_K\}$	Set of activation scores for K SOM units
e_i	Activation score of i^{th} SOM unit
$Z = \{z_1, .., z_K\}$	Win counts for K SOM units
z_i	Win count of i^{th} SOM unit
ρ	Learning solidity coefficient s.t. $\rho = 1/\epsilon$
β_i	Total activation of i^{th} SOM unit by neighbours
δ	Outlier cluster threshold, $\delta \in [0, 1]$
τ	In-cluster outlier threshold, $\tau \in [0, 1]$
ν	Preserved PCA variance threshold, $\nu \in [0, 1]$

rule as in Eq.3.1.

$$w_j^t = w_j^{t-1} + h(n_j, n_{\hat{v}} : \epsilon^t, \sigma^t)[x_i - w_j^{t-1}] \tag{3.1}$$

The update step is scaled by the window function $h(n_i, n_{\hat{v}} : \epsilon^t, \sigma^t)$ for each SOM unit, inversely proportional to the distance to the winner (Eq. 3.2). The learning rate ϵ is a gradually decreasing value, resulting in larger updates at the beginning and finer updates as the algorithm evolves. σ^t defines the neighbouring effect so with the decreasing σ, neighbour update steps are getting smaller in each epoch. Note that, there are different alternatives for update and windows functions in SOM literature.

$$h(n_i, n_{\hat{v}} : \epsilon^t, \sigma^t) = \epsilon^t \exp \frac{-||n_j - n_{\hat{v}}||^2}{2\sigma^{t^2}} \tag{3.2}$$

3.2 Clustering and outlier detection with CMAP:

CMAP introduces excitation scores $E = \{e_1, e_2, \ldots, e_K\}$ where e_j, the score for neuron unit j, is updated as in Eq. 3.3.

$$e_j^t = e_j^{t-1} + \rho^t(\beta_j + z_j) \tag{3.3}$$

As in SOM, window function gets smaller with each iteration. z_j is the activation or win count for the unit j, for one epoch. ρ is learning solidity scalar that represents the decisiveness of learning with dynamically increasing value, assuming that later stages of the algorithm have more impact on the definition of salient SOM units. ρ is equal to the inverse of the learning rate ϵ. β_j is the total measure of the activation of jth unit in an epoch, caused by all the winners of the epoch but the neuron itself (Eq. 3.4).

$$\beta_j = \sum_{i=1}^{K} h(n_j, n_i : \epsilon^t, \sigma^t) z_i \tag{3.4}$$

At the end of the iterations, normalised e_j is a quality value of a unit j. Higher value of e_j indicates that total amount of excitation of the unit j in the entire learning period is high thus it is responsive to the given class of instances and it captures notable amount of data. Low excitation values indicate the contrary. CMAP is capable of detecting outlier units via a threshold θ in the range $[0, 1]$ on e_j

Let $C = \{c_1, c_2, \ldots, c_K\}$ be the cluster centres corresponding to each unit. c_j is considered to be a **salient cluster** if $e_j \geq \theta$, and an **outlier cluster** otherwise.

The excitation scores E are the measures for saliency of neuron units in CMAP. Given the data belonging to a category, we expect that data is composed of sub-categories that share common properties. For instance `red` images might include darker or lighter tones to be captured by clusters but they are supposed to share a common characteristics of being red. In that sense, for the calculation of the excitation scores we use individual activations of the units as well as the activations as being in a neighbourhood of another unit. Individual activations measure the saliency of being a salient cluster corresponding to a particular sub-category, such as `lighter red`. Neighbourhood activations count the saliency in terms of the shared regularity between

sub-categories. If we don't count the neighbourhood effect, some unrelated clusters would be called salient since large number of outlier instances could be grouped in a unit, e.g. noisy white background patches in `red` images.

Outlier instances of salient clusters, namely the **outlier elements** should also be detected. After the detection of outlier neurons, statistics of the distances between neuron weight w_i and its corresponding instance vectors (assuming weights prototyping instances grouped by the neuron) is used as a measure of instance divergence. If the distance between the instance vector x_j and its winner's weight $\hat{w}_i$ is more than the distances of other instances having the same winner, x_j is raised as an outlier element. We exploit box plot statistics, similar to [72]. If the distance of the instance to its cluster's weight is more than the upper-quartile value, then it is detected as an outlier. The portion of the data, covered by the upper whisker is decided by τ.

CMAP provides a good basis for cleansing poor instances whereas computing cost is relatively small since CMAP is capable of discarding items with one shot of learning phase. Thus, an additional data cleansing iteration after clustering phase is not required. All the necessary information (excitation scores, box plot statistics) for outliers is calculated at runtime of learning. Hence, CMAP is suitable for large scale problems.

CMAP is also able to estimate the number of intrinsic clusters of the data. We use PCA as a simple heuristic for that purpose, with defined variance ν to be retained by the selected first principle components. Given data and ν, principle components are found and the number of principle components describing the data with variance ν is used as the number of clusters for the further processing of CMAP. If we increase ν, CMAP latches more clusters therefore ν should be carefully chosen."

$$Num.Clusters = \max_{q} \left(\frac{\sum_{i=1}^{q} \lambda_i}{\sum_{j=1}^{p} \lambda_j} \leq \nu \right) \tag{3.5}$$

where q is the number of top principle components selected after PCA and p is the dimension of instance vectors. λ is the eigenvalue of the corresponding component.

Figure 3.2 depicts CMAP layout on a toy example with neighbourhood connections between units. As the red points show, CMAP is able to find the fringes of the given

data space with outlier units as it discovers the salient regions that are possibly reliable in a noisy environment with salient CMAP units. There are also some of empty CMAP outlier clusters. They are useful for continuous data flow. Although they are empty now, later in time new instances captured by these units are labelled outlier. This feature makes CMAP useful for online problems as well, even we do not testify it in this work. Another fact on the figure is the importance of outlier detection in the salient clusters. You might observe that some of the outlier instances are not accommodated by the outlier units but they are not coherently embraced by the closest salient unit as well. Hence the statistical threshold is able to detect such outlier instances.

CMAP is described in Algorithm 1. Note that, although in the real code vectorised implementation is used, we write down iterative pseudo-code for the favour of simplicity. The code is available at http://www.erengolge.com/pub_sites/concept_map.html.

Computational complexity: With the support of GPGPU programming CMAP scales to large data with roughly constant time matrix multiplication (implementation described in [73]). If we discern the complexity of each phase, unit to instance distance is $O(K.(N.f_{\neq 0}))$, finding winning units is $O(K)$, and update weights is $O(K.N.f_{\neq 0})$. Here K is the number of output units (clusters), N is number of instances and $f_{\neq 0}$ is number of non-zero items in input matrix. Then by applying GPU to matrix multiplication steps by keeping all the process in GPU memory, roughly (since the performance gain is depended to the hardware used) deceased complexity by constant matrix multiplication (discarding memory dispatch overhead) is as follows: unit to instance distance is $O(K + N)$, finding winning units is $O(K)$, update weights is $O(K + N)$.

Why we choose SOM over K-means: Someone might prefer to thrive CMAP on K-means instead of SOM. However, there are some of the important differences between these two algorithms that highlight SOM. First, K-means performs poorly on non-globulor cluster like chain like data distribution. This is not good for our problem of outlier detection since in that case cluster activations are not reliable with flawed cluster distribution. SOM units are not constrained unlike K-menas. Moreover, the real problem is basically mapping the units onto the data with optimal objective value instead of clustering. Therefore, some units prone to latch many instances, even some others are empty with better mapping even onto non-globulos distiributions. Second, as

Algorithm 1: CMAP

Input: $X = \{x_1, .., x_M\}$, θ, τ, R, T, ν, σ^{init}, ϵ^{init}
Output: $OutlierUnits, Membership, W$

set each item z_i in Z to 0
$K \leftarrow estimateUnitNumber(X, \nu)$
$W \leftarrow randomInit(K)$
while $t \leq T$ **do**
    $\epsilon^t \leftarrow computeLearningRate(t, \epsilon^{init})$
    $\rho^t \leftarrow 1/\epsilon^t$
    set each item β_i in B to 0
    select a batch set $X^t \subset X$ with R instances
    for *each* $x_i \in X^t$ **do**
        $\hat{v} \leftarrow min_j(||x_i - w_j||)$
        increase win count $z_{\hat{v}} \leftarrow z_{\hat{v}} + 1$
        for *each* $w_k \in W$ **do**
            $\beta_k^t = \beta_k^t + h(n_k, n_{\hat{v}})$
            $w_k = w_k + h(n_k, n_{\hat{v}})||x_i - w_{\hat{v}}||$
        end
    end
    for *each* $w_j \in W$ **do**
        $e_j^t = e_j^{t-1} + \rho^t(\beta_j^t + z_j)$
    end
    $t \leftarrow t + 1$
end
$W_{outlier} \leftarrow thresholding(E, \theta)$
$W_{inlier} \leftarrow W \setminus W_{outlier}$
$Membership \leftarrow findMembership(W_{inlier}, X)$
$Whiskers \leftarrow findUpperWhiskers(W_{inlier}, X)$
$X_{outlier} \leftarrow findOutlierIns(X, W_{inlier}, Whiskers, \tau)$
return $W_{outlier}, X_{outlier}, Membership, W$

distinct from K-means, SOM units are oriented with neighboring relations so that each winner unit can also activate (update) its neighboring units with the measure defined by the window function. Hence, if a SOM unit is mapped onto a dense instance region, although it is activated rarely as a winner, it might be defined as a salient unit because of the frequent activations of the neighborhood units.

3.3 Concept learning with CMAP

We utilise the clusters, that are obtained through clustering and outlier detection as presented above, for learning sub-models in categorisation of concepts. We exploit the proposed framework for learning of attributes, scenes, and objects. Each task requires the collection of data, selection of instances that will be fed into CMAP, clustering and outlier detection with CMAP, and finally training of sub-models from the resulting clusters. In the following, first we will describe the attribute learning, and then describe the differences in learning other concepts. Implementation details are presented in Section 3.4.

3.3.1 Learning low-level attributes:

Recently, use of visual attributes have become attractive as being helpful in describing properties shared by multiple categories and resulting in novel category recognition. However, most of the methods require learning of visual attributes from labeled data, and cannot eliminate human effort. Yet, it may be more difficult to describe an attribute than an object, and localisation may not be trivial.

Alternatively, images tagged with attribute names are available on the web in large amounts. However, data collected from web inherits all type of challenges due to illumination, reflection, scale, and pose variations as well as camera and compression effects [3]. Most importantly, the collection is very noisy with several irrelevant images as well as variety of images corresponding to different characteristic properties of the attribute (Figure 1.1). Localisation of attributes inside the images arises as another

important issue. The region corresponding to the attribute may cover only a fraction of the image, or the same attribute may be in different forms in different parts of an image.

Here, we describe our method in learning attributes from web data without any supervision.

Dataset and model construction: We collect crude web images through querying for colour and texture names. Specifically, we gathered images from Google for 11 distinct colours as in [3] and 13 textures. We included the terms "colour" and "texture" in the queries, such as "red colour", or "wooden texture". For each attribute, 500 images are collected. In total we have 12000 web images.

The data is weakly labelled, with the labels given for the entire image, rather than the specific regions. Most importantly, there are irrelevant images in the collection, as well as images with a tiny portion corresponding to the query keyword.

We aim to answer not only "which attribute is in the image?", but also "where the attribute is?". For this purpose, we consider image patches as the basic units for providing localisation.

Each image is densely divided into non-overlapping fixed-size (100x100) patches to sufficiently capture the required information. We assume that the large volume of the data itself is sufficient to provide instances at various scales and illuminations, and therefore we did not perform any scaling or normalisation. Unlike [3], we didn't apply gamma correction. For colour concepts we use 10x20x20 bins Lab space colour histograms and for texture concepts we use BoW representation for densely sampled SIFT [74] features with 4000 words. We keep the feature dimensions high to utilise from the over-complete representations of the instances with L1 norm linear SVM classifier.

The collection of all patches extracted from all images for a single attribute is then given to CMAP to obtain clusters which are likely to capture different characteristics of the attribute as removing the irrelevant image patches.

Each cluster obtained through CMAP is used to train a separate classifier for the

attribute. Positive examples are selected as the members of the cluster and negative instances are selected among the outliers removed by CMAP for that attribute and also among random elements from other attribute categories.

Attribute recognition on novel images: The goal of this task is to label a given image with a single attribute name. Although there may be multiple attributes in a single image, for being able to compare our results on benchmark data-sets we consider one attribute label per image. For this purpose, first we divide the test images into grids in three levels using spatial pyramiding [4]. Non-overlapping patches (with the same size of training patches) are extracted from each grid of all three levels. Recall that, we have multiple classifiers for each attribute trained on different salient clusters. We run all the classifiers on each grid for all patches. Then, we have a vector of confidence values for each patch, corresponding to each particular cluster classifier. We sum those confidence vectors of each patch in the same grid. Each grid at each level is labelled by the maximum confidence classifier among all the outputs for the patches. All of those confidence values are then merged with a weighted sum to a label for the entire image.

$$D^i = \sum_{l=1}^{3} \sum_{n=1}^{N_l} \frac{1}{2^{3-l}} h_i e^{-(\hat{x}-x)/2\sigma^2} \tag{3.6}$$

Here, N_l is the grid number for level l and h_i is the confidence value for grid i. We include a Gaussian filter, where $\hat{x}$ is centre of the image and x is location of the spatial pyramid grid, to give more priority to the detections around the centre of the image for reducing noisy background effect.

Attribute based scene recognition : While the results on different datasets support the ability of our approach to be generalised to different datasets, we also perform experiments to understand the effect of the learned attributes on a different task, namely for classification of scenes using entirely different collections. Experiments are performed on MIT-indoor [6], and Scene-15 [4] datasets. MIT-indoor has 67 different indoor scene with 15620 images with at least 100 images for each category and we use 100 images from each class to test our results. Scene-15 is composed by 15 different scene categories. We use 200 images from each category for our testing. MIT-indoor is extended and even harder version of Scene-15 with many additional categories.

We again get the confidence values for each grid in three levels of the spatial pyramid on the test images. However, rather than using a single value for the maximum classifier output, we keep the confidence values for all the classifiers for each grid. We concatenate these vectors for all grids in all levels to get a single feature vector of size 3xNxK for the image, which is then used for scene classification. Here N is the number of grids at each level, and K is the number of different concepts. Note that, while the attributes are learned in an unsupervised way, in this experiment scene classifiers are trained on the datasets provided (see next section for automatic scene concept learning).

This method will be referred to as **CMAP-A**.

3.3.2 Learning scene categories:

To show that CMAP is capable of being generalised to higher level concepts, we collected images for scene categories from web to learn these concepts directly. Note that, alternative to recognising scenes through the learned attributes, in this case we directly learn higher level concepts for scene categories. For this task, which we refer to as **CMAP-S**, we use the entire images as instances, and aim to discover group of images each representing a different property of the scene category, at the same time by eliminating the images that are either irrelevant, or poor to sufficiently describe any characteristics.. These clusters are then used as models similar to the attribute learning.

Specifically, we perform testing for scene classification for 15 scene categories on [4] and MIT-indoor [6] data-sets, but learn the scene concepts directly from the images collected from Web through querying for the names of the scene concepts used in these datasets. That is, we do not use any manually labelled training set (or training subset of the benchmark data-sets), but directly the crude web images which are pruned and organised by CMAP, in contrast to comparable fully supervised methods.

3.3.3 Learning object categories:

In the case of objects, we detect salient regions on each image via [1], to eliminate background noise (see Figure 3.3). Then these salient regions are fed into CMAP framework for clustering.

Salient regions extracted from images are represented with 500 word quantized SIFT [74] vector with additional 256 dimension LBP [75] vector. In total we aggregated a 756 dimension vector representation for each salient region. At the final stage of learning with CMAP, we learn L2 norm, linear SVM classifiers for each cluster with negatives are gathered from other classes and the global outliers. For each learning iteration, we also apply hard mining to cull highest rank negative instances in the amount 10 times of salient instances in the cluster. All pipeline hyper-parameters are tuned via the validation set provided by [2]. Given a novel image, learned classifiers are passed over the image with gradually increasing scales, up to a point where the maximum class confidences are stable.

3.3.4 Learning faces

We use FAN-large [10] face data-set for testing our method in face recognition problem. We use Easy and Hard subsets with the names accommodating more than 100 images (to have fair testing results). Our models are trained over web images queried from Bing Image search engine for the same names. All the data preprocessing and the feature extraction flow follow the same line of [10], that is owned from [76]. However, [10] trains the models and evaluates the results at the same collection.

We retrieve the top 1000 images from Bing results. Face are detected and face with the highest confidence is extracted from each image to be fed into CMAP. Face instances are clustered and spurious face instances are pruned. Salient clusters are used for learning SVM models for each cluster in the same settings of the object categories. For our experiments we used two different face detectors. One is cascade classifier of [77] implemented in OpenCV library [78] and another is [8] with more precise detection results, even the OpenCV implementation is very fast relatively.

3.3.5 Selective Search for Object Detection with CMAP

Even this problem is a bit out of the scope of the thesis, this is very intuitive application of CMAP to object detection. Selective search [9] has been recently of interest to speed up against brute-force sliding-window approach for object detection. The main idea here is to generate a hierarchy of image regions where the leaves are usually the super-pixels and the upper levels are the compositions of the most similar neighbouring regions at a level below. Besides being based on a simple idea, selective search gives very promising results, even better than the sliding-window based approaches on Pascal 2007 dataset as presented in [9].

We extend the idea of CMAP in order to reduce the number of candidate regions at each level of the hierarchy. First, we collect random image patches in different sizes and scales from the object category images, inside the annotation boxes. Then we train CMAP units with the same representations used in the selective search method of [9]. After training, when we apply selective search for object detection, any candidate region is rectified with CMAP relative to it matches with an outlier or an inlier unit. If it matches with the outlier, it is ignored for the level and so for the upper compositional levels as well. This further elimination removes considerable number of redundant regions at each level and more at the upper levels, additional to normal selective-search method. Figure 3.9 gives examples for the eliminations of CMAP.

3.4 Experiments

3.4.1 Qualitative evaluation of clusters

As Figure 3.5 depicts, CMAP captures different characteristics of concepts for attribute and scene categories in separate salient clusters, while eliminating outlier clusters that group irrelevant images coherent among themselves, as well as outlier elements wrongly mixed with the elements of salient clusters . On a more difficult task of grouping objects, CMAP is again successful in eliminating outlier elements and outlier clusters as shown in Figure 3.6 and Figure 3.7.

3.4.2 Implementation details

Parameters of CMAP are tuned on a small held-out set gathered for each concept class for colour, texture, and scene. We apply grid-search on the held-out set for each concept class. Best ν is selected by the optimal Mean Squared Error and threshold parameters are tuned by cross-validation accuracies of the classifiers trained by salient clusters appeared by the corresponding threshold values. Figure 3.10 depicts the effect of parameters θ, τ and ν. For each parameter the other two are fixed at the optimum value.

We use LIBLINEAR library [79] for L1 norm SVM classifiers. SVM parameters are selected with 10-fold cross validation and grid-search. We end the search process when the current accuracy is less than the average accuracy of the 5-10 step back.

CMAP implementation is powered by GPGPU programming over CUDA environment. Matrix operations observed for each iteration is kernellised by CUDA codes. It provides good reduction in time, especially if the instance vectors are large and all the data is able to fit into GPU memory. Hence, we are able to execute all the optimisation steps in the GPU memory. Otherwise some dispatching overhead is observed between GPU and global memory that sometimes hinge the GPGPU efficiency. Thus, GPU implementation should be considered in relation to specific architecture and data-matrix.

3.4.3 Attribute recognition on novel images

For evaluation we use three different datasets. The first dataset is Bing Search Images curated by ourselves from the top 35 images returned with the same queries we used for initial images. This set includes 840 images in total for testing. Second dataset is Google Colour Images [3] previously used by [3] for learning colour attributes. Google Colour Image dataset includes 100 images for each colour name. We used the entire data-sets only for testing of our models learned on a possibly different set that we collected from Google, contrary to [3]. The last dataset is sample annotated images from ImageNet [5] for 25 attributes. To test the results on a human labelled dataset, we use Ebay dataset provided by [3] which has labels for the pixels in cropped regions. It

includes 40 images for each colour name.

Our method is also utilised for retrieving images on EBAY dataset as in [3]. [3] learns the models from web images and apply the models to another set so both method study a similar problem. We utilise CMAP with patches obtained from the entire images (**CMAP**) as well as from the masks provided by [3] (**CMAP-M**). As shown in Figure 3.12, even without masks CMAP is comparable to the performance of the PLSA based method of [3], and with the same setting CMAP outperforms the PLSA based method with significant performance difference.

On ImageNet dataset, we obtained 37.4% accuracy compared to 36.8% of Russakovsky and Fei-Fei [5]. It is also significant that, our models trained from different source of information are better to generalised for some of worse performance classes (rough, spotted, striped, wood) of [5]. Recall that we globally learn the attribute models from web images, not from any partition of the ImageNet. Thus, it is encouraging to observe better results in such a large data-set against [5]'s attribute models trained by a sufficiently large training subset.

3.4.4 Comparison with other clustering methods

Figure 3.13 compares the overall accuracy of the proposed method (**CMAP**) with other methods on the task of attribute learning. As the **Baseline**, we use all the images returned for the concept query to train a single model. This case simulates a single cluster with no pruning. As expected, the performance is very low suggesting that a single model trained by crude noisy web images performs poorly and the data should be organised to train at least some qualified models from coherent clusters in which representative images are grouped. As other methods for clustering the data, we used **k-means**, original **SOM** algorithm, **MeanShift** [80] and **DBSCAN** [81]. Optimal cluster numbers are decided by cross-validation when the algorithm requires, and again models are trained for each cluster. The low results support the need for pruning of the data through outlier elimination. Results show that, CMAP's clusters are able to detect coherent and clean representative data groups so we train less number of classifiers by eliminating outlier clusters but those classifiers better in quality and also, on novel

test sets with images having different characteristics than the images used in training, CMAP can still perform very well on learning of attributes.

CMAP is a computationally efficient algorithm as well compared to the other alternatives that we experimented. For one class, running times were 15 minutes for k-means, 19 minutes for CMAP (on GPU), 42 minutes for MeanShift and 53 minutes for DBSCAN. Although k-means is fast as well, since it does not detect outliers and prune the spurious instances, it has very low performance compared to CMAP. CMAP has also almost the same computation time compared to SOM since all the required information is computed in the original SOM iterations. However, CMAP yields better results with additional data pruning. MeanShift and DBSCAN are the other common clustering techniques. These methods are computationally very intensive for especially large scale problems. We observed that in the same machine and with the same amount of data they waste three order of magnitude more time compared to CMAP. Furthermore, since we rely on long dimensional representations, MeanShift and DBSCAN suffers from curse of dimensionality. They give very high varying results for different runs because they find the number of clusters by their intrinsic properties that are not very reliable in high dimensions. For these facts, CMAP is better in mapping noisy data in large scale problems, as the comparative results are given at the Figure 3.12.

Table 3.2 depicts more detailed class-wise accuracies, comparing Concept Map with Baseline method, as well as with k-means and SOM. Results are evident that using clustering and learning separate models improve the classification results comparing to raw models. It is because, by clustering we are able to capture representative image instances through , at least, some of the clusters and the models from those representative clusters are more qualified. At the second stage, we observe impact of outlier detection. Results clearly show that removing superiors instances from the data-set, additional to clustering, increases final accuracy values substantially. Table 3.2 indicates that our final method Concept Map improves the baseline (BL) accuracy 38.5% in average.

Table 3.2: Concept classification results over the datasets with different methods. In K-means **KM**, K is found out on held-out set, (some of the values are empty since that category is not provided by the data-set) **BL** is the Baseline method with no clustering and outlier detection. For ImageNet we only use the classes used in the paper [5] for better comparison and bold values are the results we obtain better. Although [5] trains classifiers from annotated images, our results absolute some of the classes including their poor performance classes as rough, spotted, striped. For other classes we have 3.45% lower accuracy in average. Google colour and EBAY datasets cannot be compared with referred paper since they expose object retrieval results other than colour classification accuracies.

DATA	Bing				Google [3]				ImageNet [5]				EBAY [3]			
METHOD	CMAP	SOM	KM	BL	CMAP	SOM	KM	BL	CMAP	SOM	KM	BL	CMAP	SOM	KM	BL
black	0.89	0.54	0.60	0.30	0.73	0.30	0.32	0.27	0.60	0.21	0.21	0.17	0.83	0.67	0.70	0.63
blue	0.88	0.50	0.48	0.23	0.62	0.34	0.33	0.29	0.63	0.25	0.29	0.25	0.79	0.63	0.65	0.60
brown	0.88	0.51	0.53	0.27	0.64	0.23	0.27	0.21	0.62	0.29	0.32	0.21	0.87	0.74	0.72	0.51
green	0.91	0.53	0.49	0.28	0.72	0.28	0.30	0.27	**0.42**	0.25	0.28	0.22	0.84	0.65	0.70	0.57
gray	0.79	0.45	0.48	0.30	0.70	0.21	0.23	0.19	0.27	0.13	0.16	0.14	0.72	0.68	0.68	0.52
orange	0.94	0.65	0.69	0.31	0.80	0.47	0.45	0.31	**0.30**	0.23	0.19	0.18	0.83	0.74	0.71	0.52
pink	0.86	0.54	0.47	0.20	0.79	0.53	0.41	0.32	-	-	-	-	0.78	0.63	0.62	0.58
purple	0.84	0.50	0.51	0.29	0.77	0.37	0.35	0.30	-	-	-	-	0.75	0.54	0.58	0.50
red	0.80	0.57	0.53	0.32	0.64	0.24	0.22	0.21	**0.61**	0.25	0.28	0.21	0.80	0.72	0.75	0.55
white	0.81	0.54	0.57	0.37	0.57	0.28	0.30	0.22	0.56	0.33	0.32	0.30	0.91	0.80	0.83	0.68
yellow	0.90	0.63	0.64	0.43	0.73	0.21	0.22	0.19	0.40	0.24	0.19	0.17	0.81	0.63	0.67	0.46
colours	0.86	0.54	0.54	0.30	0.70	0.31	0.30	0.25	0.49	0.24	0.25	0.20	0.81	0.68	0.69	0.56
furry	0.92	0.79	0.84	0.50	-	-	-	-	0.70	0.53	0.54	0.43	-	-	-	-
grass	0.91	0.70	0.73	0.47	-	-	-	-	-	-	-	-	-	-	-	-
metallic	0.87	0.64	0.61	0.35	-	-	-	-	0.12	0.09	0.07	0.02	-	-	-	-
rough	0.78	0.58	0.57	0.23	-	-	-	-	**0.1**	0.081	0.082	0	-	-	-	-
shiny	0.64	0.57	0.52	0.27	-	-	-	-	0.31	0.23	0.27	0.22	-	-	-	-
smooth	0.57	0.45	0.49	0.13	-	-	-	-	0.35	0.23	0.24	0.21	-	-	-	-
spotted	0.64	0.41	0.45	0.22	-	-	-	-	**0.089**	0.052	0.054	0	-	-	-	-
striped	0.71	0.50	0.57	0.28	-	-	-	-	**0.09**	0.04	0.032	0.01	-	-	-	-
vegetation	0.82	0.78	0.77	0.42	-	-	-	-	-	-	-	-	-	-	-	-
wall	0.87	0.63	0.63	0.32	-	-	-	-	-	-	-	-	-	-	-	-
water	0.71	0.55	0.57	0.24	-	-	-	-	-	-	-	-	-	-	-	-
wet	0.60	0.39	0.34	0.16	-	-	-	-	0.25	0.18	0.20	0.18	-	-	-	-
wood	0.91	0.71	0.75	0.55	-	-	-	-	**0.24**	0.21	0.22	0.16	-	-	-	-
Textures	0.77	0.59	0.59	0.31	-	-	-	-	0.25	0.18	0.19	0.14	-	-	-	-
OVERALL	0.82	0.56	0.57	0.31	-	-	-	-	0.37	0.21	0.22	0.17	-	-	-	-

3.4.5 Attribute based scene recognition

As shown in Table 3.3, our method for scene recognition with learned attributes (**CMAP-A**), performs competitively with [17] while using shorter feature vectors in relatively cheaper environment, and outperforms the others. Comparisons with [6] show that using the visual information acquired from attributes is more descriptive in the cluttered nature of MIT-indoor scenes. For instance, "bookstore" images has very similar structural layout to "clothing store" images, but they are more distinct with colour and texture information around the scene. Attribute level features do not create this much difference for Scene-15 data-set since images include some obvious statistical differences.

3.4.6 Learning concepts for scene categories

As shown in Table 3.3, our method in recognising scenes directly from web images (**CMAP-S**) is competitive with the state-of-the-art studies without requiring any supervised training.

We then made a slight change on our original CMAP-S implementation by using the hard-negatives of previous iteration as a negative set of next iteration (we refer to this new method as **CMAP-S-HM**). We relax the memory needs with less but strong negative instances. As the results in Table 3.3 and Figure 3.13 show, we achieve better performances in Scene-15 than the state-of-the-art studies with this simple addition, still without requiring any supervisory input. However, on a harder MIT-indoor dataset, without using attribute information, low-level features are not very distinctive.

3.4.7 Comparisons with discriminative visual features

In order to understand the effect of discriminative visual features, which aim to capture representative and discriminative mid-level features, we also compare our method with the work of Singh *et al.*[18]. As seen in Table 3.3, our performances are better than both their reported results on MIT-indoor, and our implementation on Scene-15.

-	MIT-indoor	Scene-15
CMAP-A	46.2%	82.7%
CMAP-S	40.8%	80.7%
CMAP-S+HM	41.7%	81.3%
Li et al. [17] VQ	47.6%	82.1%
Pandey et al. [82]	43.1%	-
Kwitt et al. [83]	44%	82.3%
Lazebnik et al. [4]	-	81%
Singh et al. [18]	38%	77%

Table 3.3: Comparison of our methods on scene recognition in relation to state-of-the-art studies on MIT-Indoor [6] and Scene-15 [4] datasets. CMAP-A uses attributes for learning scenes. CMAP-S learns scenes directly and CMAP-S+HM uses hard mining for the final models.

We also tried to use the other alternative method like [17] for our datasets. However, these methods are not applicable at our computer configurations because of high memory demands for reliable results.

3.4.8 Learning concepts of object categories

We learn object concepts from Google web images used in [2] and compare our results with [2] and [7] (see Table 3.4).

[2] provides a data-set from Google with 7 classes and total 4088 grey scale images, 584 images in average for each class with many "junk" images in each class as they indicated. They test their results in a manually selected subset of Caltech Object data-set. Because of its raw nature of the Google images and adaptation to the Caltech subset, it is a good experimental ground for our pipeline.

Among class confidences, maximum confidence indicates the final prediction for that image. We observe 6.3 salient clusters in average for all classes and 69.4 instances for each salient clusters. That is, CMAP eliminates 147 instances for each class as supposedly outlier instances. Results support that elimination of "junk" images gives significant improvements, especially for the noisy classes in [2].

	CMAP	[2]	[7]		CMAP	[2]	[7]
airplane	0.63	0.51	0.76	car	0.97	0.98	0.94
face	0.67	0.52	0.82	guitar	**0.89**	0.81	0.60
leopard	0.76	0.74	0.89	motorbike	**0.98**	0.98	0.67
watch	**0.55**	0.48	0.53	overall	**0.78**	0.72	0.75

Table 3.4: Classification accuracies of our method in relation to [2] and [7].

Method	GBC+CF(half)[10]	CMAP-1	CMAP-2	BaseLine
Easy	0.58	0.63	0.66	0.31
Hard	0.32	0.34	0.38	0.18

Table 3.5: Face learning results with detecting faces using OpenCV(CMAP-1) and [8](CMAP-2).

3.4.9 Learning Faces

CMAP results are compared with [10] and the aforementioned baseline in EASY and HARD subsets of the dataset. Results are depicted at Table3.5 with two different face detection method and baseline result with models trained on raw Bing images for each person.

3.4.10 Selective Search for Object Detection with CMAP

We compare the selective search [9] idea using CMAP on Pascal 2007 test set with two other methods. As seen in Table 3.6 our method achieves better results with lower number of candidate windows. Here MABO refers to Mean Average Biggest Overlap rate. CMAP reduces number of candidate regions from 10.097 to 6.753 in average which is

	MABO	Recall	No. of Windows
Objectness [84]	0.69	0.94	1.853
Selective Search [9]	0.879	0.991	10.097
Selective Search + CMAP	0.891	0.993	6.753

Table 3.6: Object Detections results on Pascal 2007 TEST set. The best result of [9] is provided here. We applied the same training pipeline as suggested in [9].

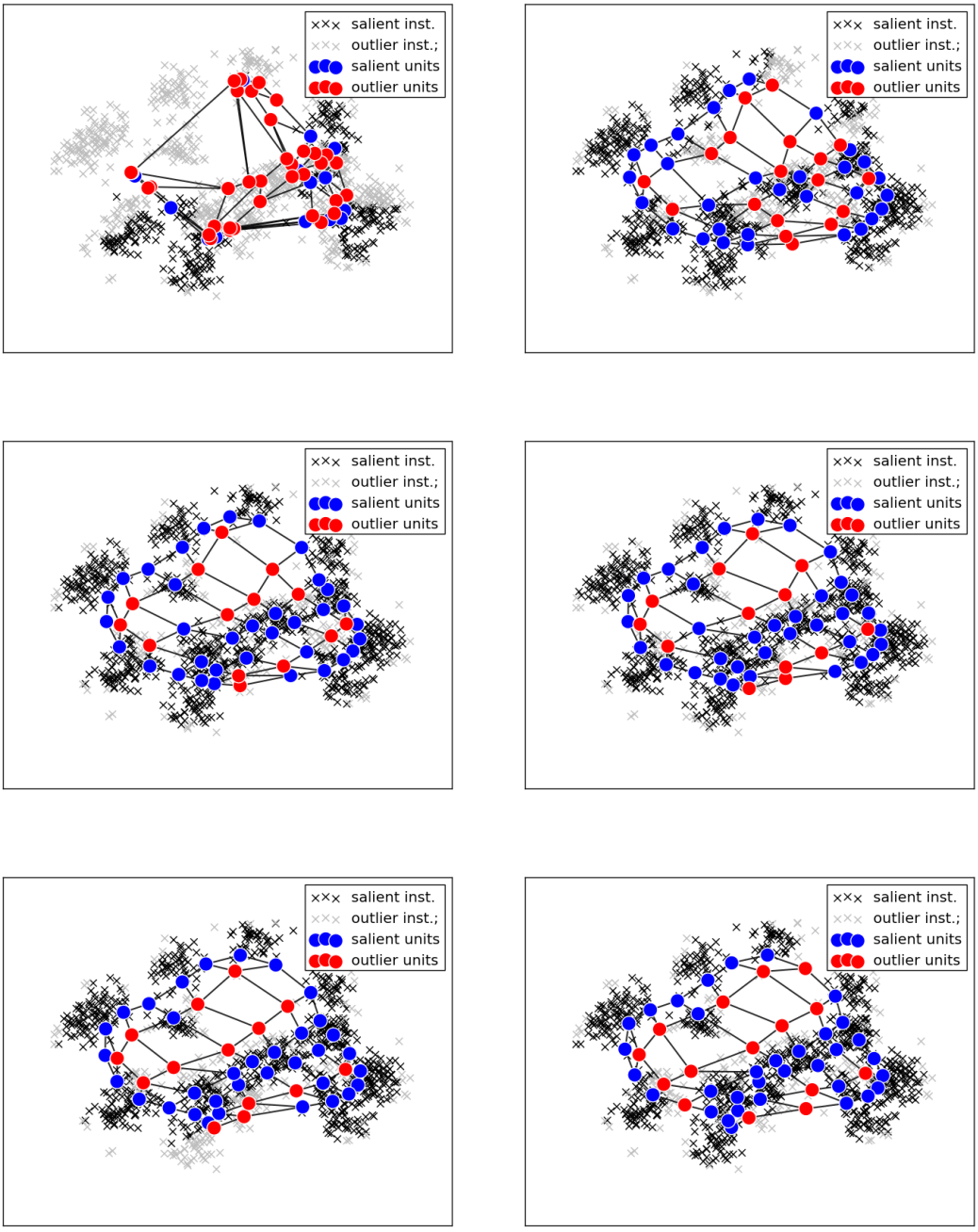


Figure 3.1: first to sixth iterations of CMAP from left top to bottom right. After the initial iterations CMAP just has small changes of the unit positions.

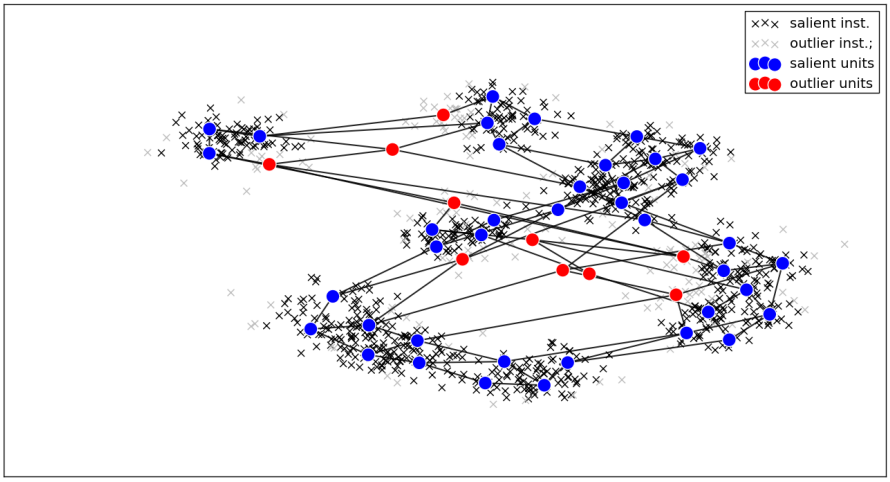


Figure 3.2: SOM units superimposed over a toy dataset with neighbourhood edges. It is clear that the outlier clusters (red points) are at the fringes of the given data space and some of them even has no member instances.

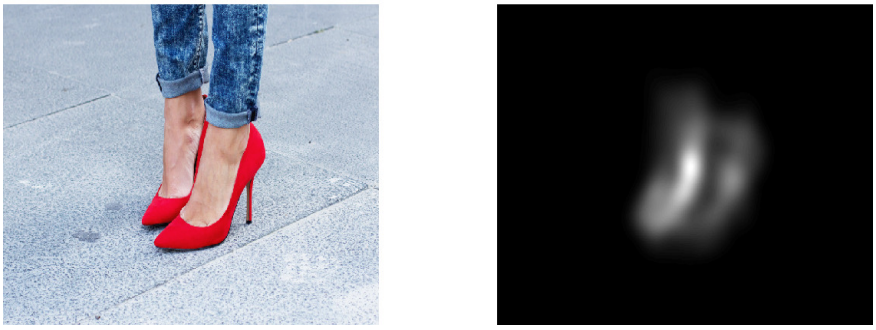


Figure 3.3: **Left:** Given object image with red attribute and **Right:** Salient object regions highlighted by the method of [1]. We apply our learning pipeline after we find salient object regions of the object category images by [1] and we only use top 5 salient regions from each image.

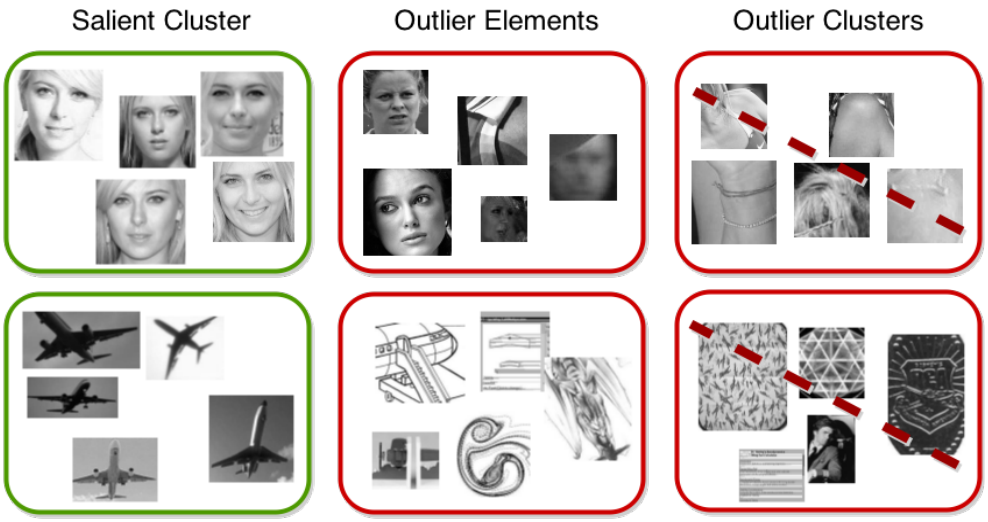


Figure 3.4: CMAP results for object and face examples. Left columns shows one example of salient cluster. Middle column shows outlier instances captured from salient clusters. Right column is the detected outlier clusters.

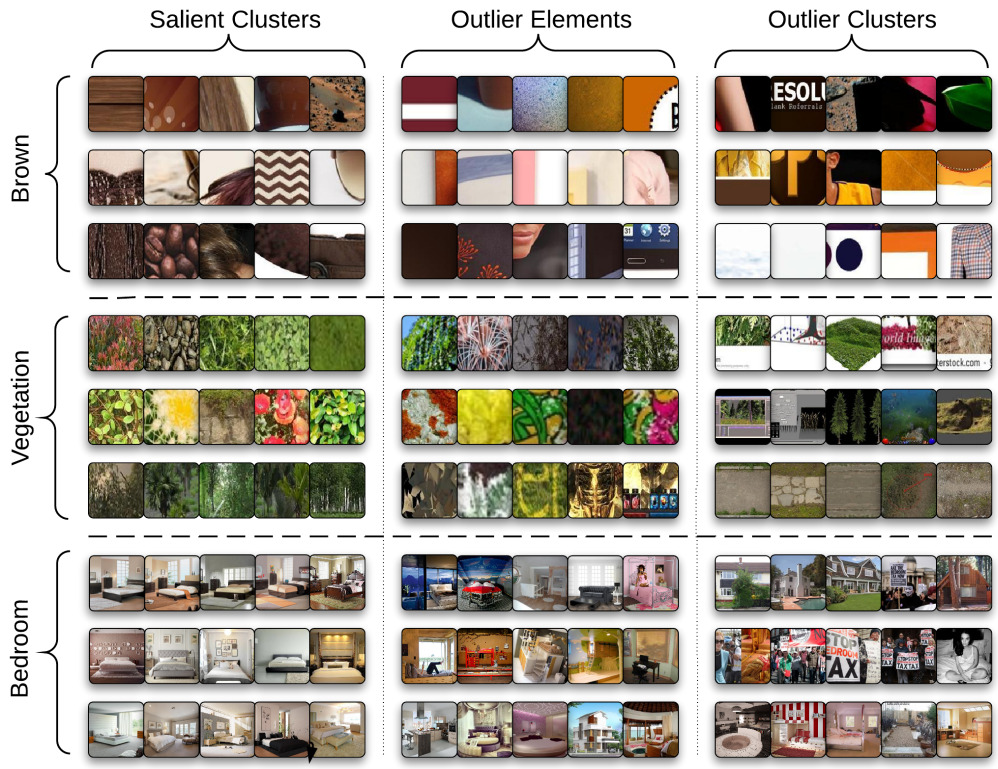


Figure 3.5: For colour and texture attributes *brown* and *vegetation* and scene concept *bedroom*, randomly sampled images detected as (i) elements of **salient clusters**, (ii) elements of **outlier clusters**, and (iii) **outlier elements** in salient clusters. CMAP detects different shades of Brown and eliminates some superior elements belonging the different colours. For the Vegetation and Bedroom , CMAP again divides the visuals elements with respect to structural and angular properties. Especially for Bedroom, each cluster is able to capture different view-angle of the images as it successfully removes outlier instances with some of little mistakes that are belonging to the label but not representative for the concept part.

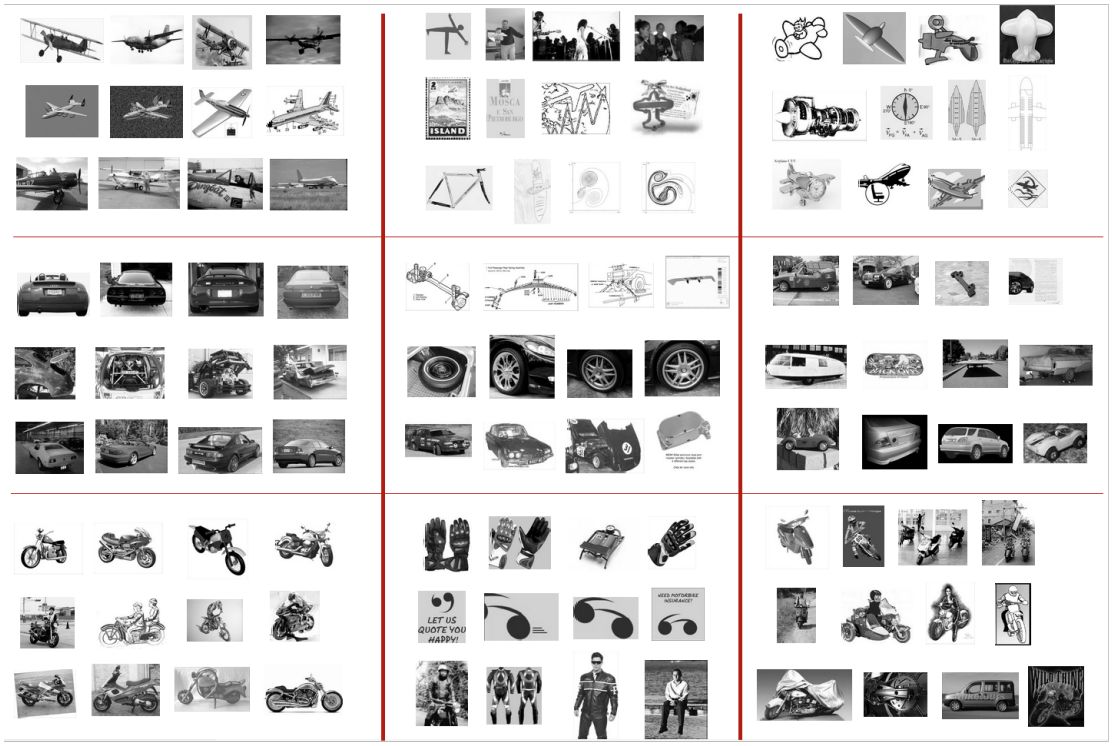


Figure 3.6: Examples of object clusters gathered from the Google images data-set of [2]. We give randomly selected sampled of three object classes; airplane, cars_rear, motorbike. Each class depicted with three salient clusters, three outlier clusters and three set of outlier instances -outliers detected in the salient clusters-. Each set of outlier instances are from the salient cluster shown at the same row. In the data-set there are duplicates and we eliminate those when we select the figure samples.

Figure 3.7: Examples of face clusters. We give randomly selected sampled of three face categories; Andy Roddick, Paul Gasol, Barrack Obama. Each category depicted with three salient clusters, three outlier clusters and three set of outlier instances -outliers detected in the salient clusters-. Each set of outlier instances are from the salient cluster shown at the same row. In the data-set there are duplicates and we eliminate those when we select the figure samples.

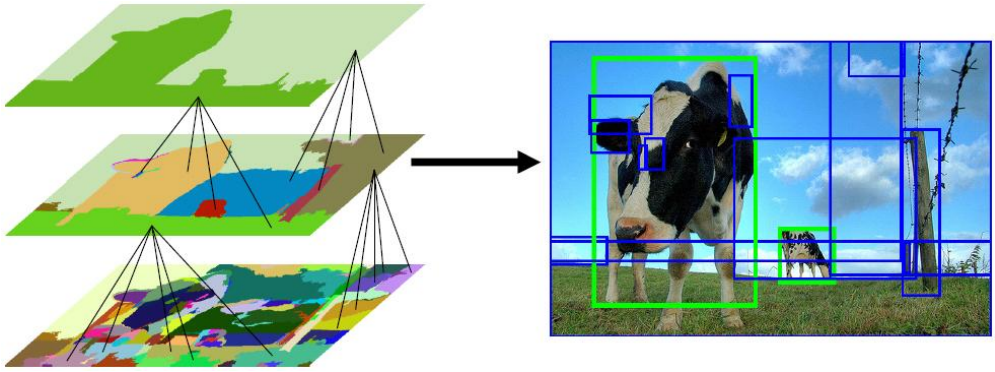


Figure 3.8: Object detection with Selective Search [9]. At the left, there is the superpixel hierarchy where each superpixel is merged with the visually most similar neigbouring superpixel for the upper layer. CMAP removes outlier superpixel for each of layers before the merging.

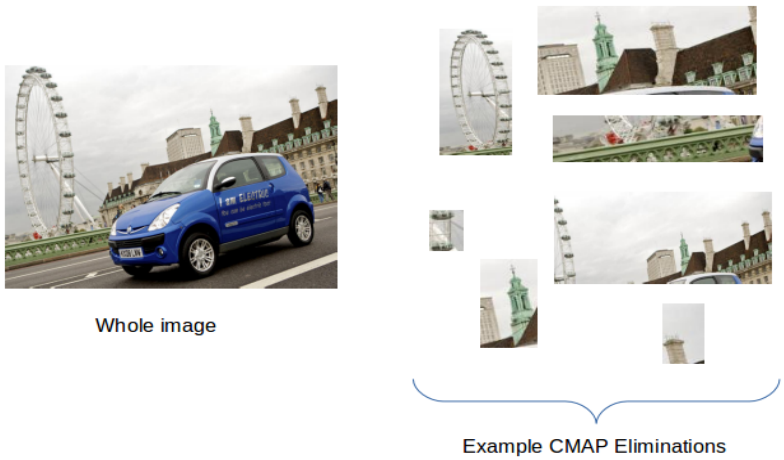


Figure 3.9: Example of CMAP elimination in Selective Search for “car” category.

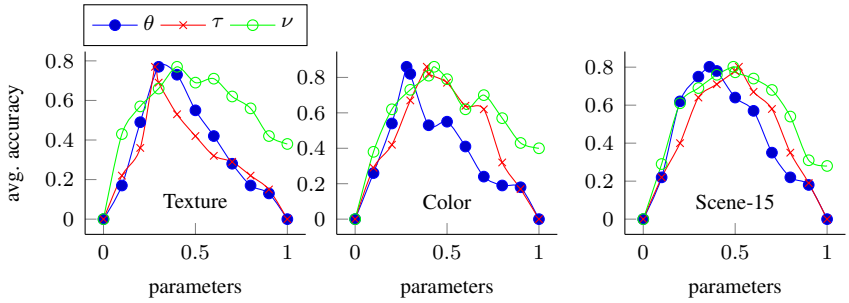


Figure 3.10: Effect of parameters on average accuracy. For each parameter, the other two are fixed at their optimal values. θ is outlier cluster threshold, ν is PCA variation used for the estimation of number of clusters,τ is the upper whisker threshold for the outliers in salient clusters.

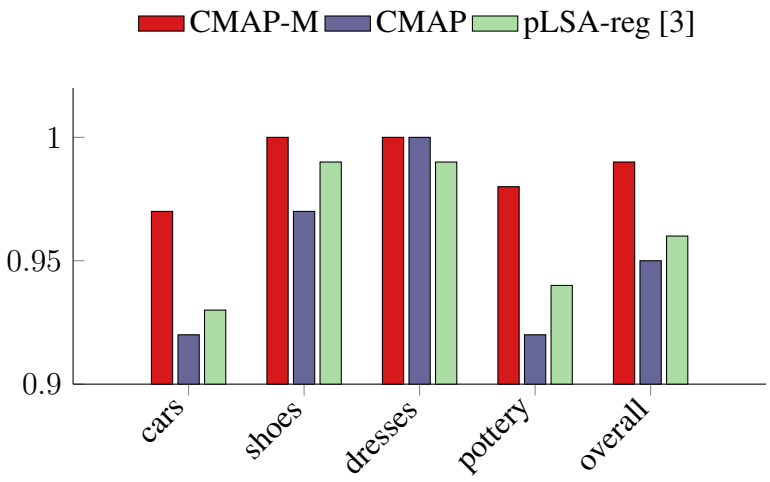


Figure 3.11: Equal Error Rates on EBAY dataset for image retrieval using the configuration of [3]. CMAP does not utilise the image masks used in [3], while CMAP-M does.

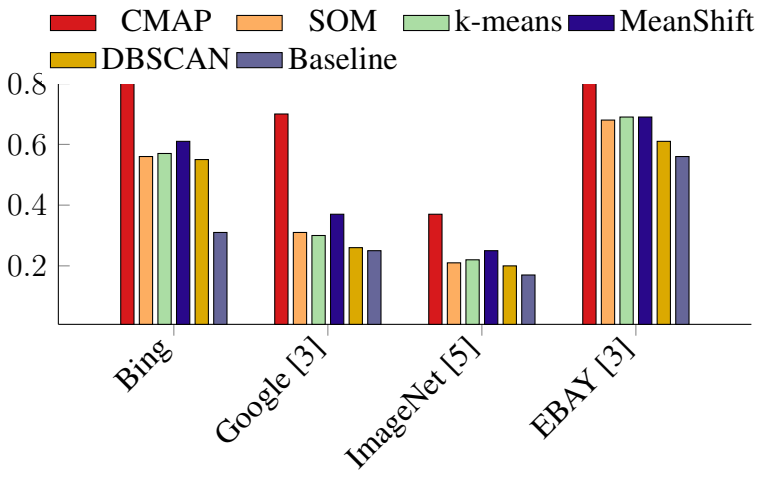


Figure 3.12: Attribute recognition performances on novel images compared to other clustering methods.

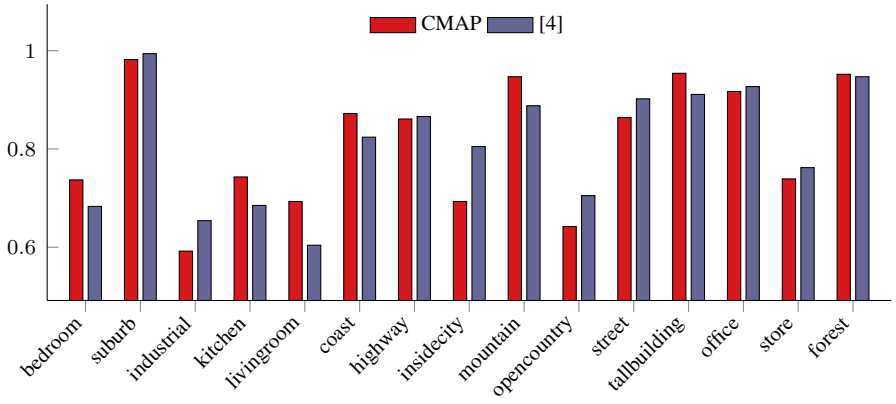


Figure 3.13: Comparisons on Scene-15 dataset. Overall accuracy is 81.3% for CMAP-S+HM , versus 81% for [4] . Classes “industrial”, “insidecity”, “opencountry” results very noisy set of web images, hence trained models are not strong enough as might be observed from the chart.

Chapter 4

AME:
Association through Model Evolution

To label faces of friends in social networks or celebrities and politicians in news, automatic methods are indispensable to manage large number of face images piling up on the web. On the other hand, unlike their counterparts in controlled datasets, faces on the web inherit all type of challenges naturally, resulting in the traditional methods incapable to recognise.

We challenge the identification of faces for famous people. The famous people tend to change their make-up, hair style/colour, and accessories more often compared to regular people, resulting in large number of varieties in face images. Moreover, they are likely to appear with others in photographs, causing faces of irrelevant people to be retrieved.

In this chapter, we present a new approach for learning better models through iteratively pruning the data (Figure4.1). First, we benefit from large number of global negatives representing the rest of the world against the class of interest. Next, among the candidate in-class examples we try to separate the most confident instances from the others. These two successive steps are repeated to eliminate outlier instances iteratively. To consider intra-class variability, we use a representation that results in large dimensional feature vectors to make each class linearly separable from others despite

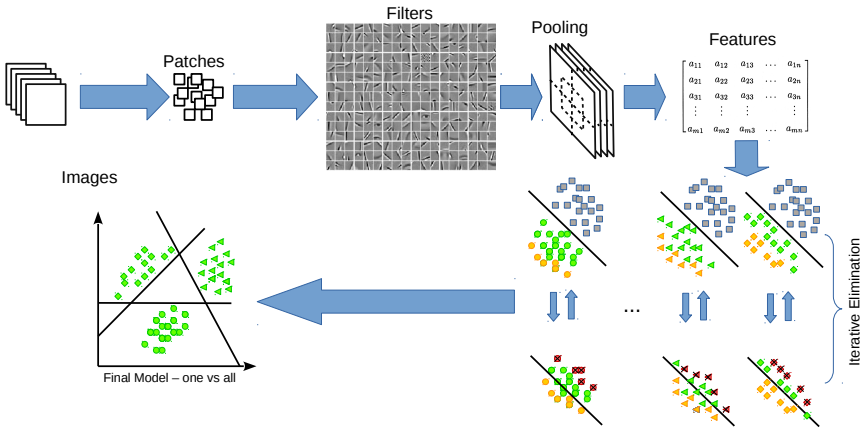


Figure 4.1: Overview of the proposed method.

of possible high level intra-class variance.

4.1 Model Evolution

In this chapter, we discuss our first proposed system AME and its experimental evaluation on the problem of learning face categories from web weakly-labelled images gathered from image search engines. Even we point one of possible use case of AME, it is a agnostic method that can be applied to other kind of tasks and domains.

We believe that in order to learn salient models we need to have a train set including instances that are different from the rest of the world and similar to other instances belonging to same category. This intuition emerges AME as a method that eliminating the spurious instances with successive linear classifiers that measure indicated qualities. First, we learn a hyperplane that separates the initial set of candidate class instances from the large set of global negatives. Global negative set is curated by the instances of other classes and the random face images collected from Web. Then, we select some fraction of the class instances that are distant from the separating hyperplane. We use these instances as the discriminative seed set, since they are confidently classified against the rest of the world. We consider the rest of the class data as possible negatives. We then learn another model that try to capture in-class dissimilarities

between discriminative examples and possible negatives. At the final step, we combine the confidence scores of the first and the second models. By combining the two scores, that respectively correspond to the confidence of being different from the rest of the world, and in-class affinity of the instance, we get a measure of instance saliency. Over these confidence scores we detect instances with the lowest scores as the outliers for that iteration. These steps are iterated multiple times up to a desired level of pruning. The representation that we use might cause computational burden with complicated learning models. Therefore, we leverage simple linear regression (LR) models with L1 norm regularisation performing sparse feature selection as the learning evolves. Sparsity makes categories more distinct and performs category specific feature selection implicitly.

Algorithm 2 summarises our data elimination procedure. $C = \{c_1, c_2, \ldots c_m\}$ refers to the examples collected for a class and $N = \{n_1, n_2, ..., n_l\}$ refers to the the vast numbers of global negatives. Each vector is a d dimensional representation of a single face image. At each iteration t, the first LR model M^1 learns a hyperplane between the candidate set of class instances C and global negatives N. Then the current C is divided into two subsets: p instances in C that are farthest from the hyperplane are kept as the candidate positive set (C^+) and the rest is considered as the negative set (C^-) for the next model. C^+ is the set of salient instances for the class and C^- is the set of possible spurious instances. The second LR model M^2 uses C^+ as positive and C^- as the negative set to learn best possible hyperplane separating them. For each instance in C^-, by aggregating the confidence values of both models, we eliminate o instances with the lowest scores as the outliers. At the next iterations, we run all the steps again and end up with a clean set of class instances C.

This iterative procedure continues until it satisfies a stopping condition. We use M^1's objective as the measure of data quality. As we incrementally remove poor instances, we expect to have better separation against the negative instances. Alternatively, when we have very large number of class instances, we can divide data into two independent subset and apply the iterative elimination to both as we measure the quality of one set's M^1 over the other set's C at each iteration t. It is similar to co-training approach and more robust to over-fitting, albeit it requires very large number of instances for convincing results.

Algorithm 2: AME

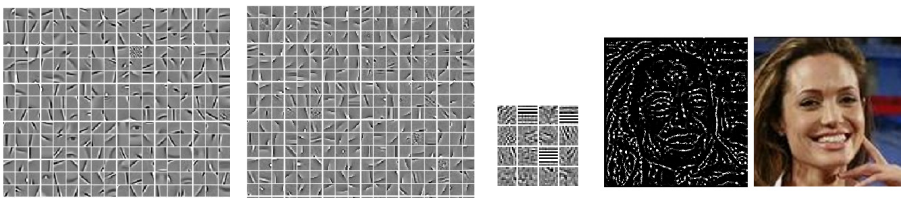
In the real code we use vectorized implementation whereas we write down iterative pseudo-code for the favour of simplicity.
Input: C, N, o, p
Output: C
$C_0 \leftarrow C$
$t \leftarrow 1$
while $stoppingConditionNotSatisfied()$ **do**
    $M_t^1 \leftarrow LogisticRegression(C_{t-1}, N)$
    $C_t^+ \leftarrow selectTopPositives(C_{t-1}, M_t^1, p)$
    $C_t^- \leftarrow C_{t-1} - C_t^+$
    $M_t^2 \leftarrow LogisticRegresstion(C_t^+, C_t^-)$
    $[S_1^-, S_2^-] \leftarrow getConfidenceScores(C_t^-, M_t^1 M_t^2)$
    $O_t \leftarrow selectOutliers(C_t^-, S_1^-, S_2^-, o)$
    $C_t \leftarrow C_{t-1} - O_t$
    $t \leftarrow t + 1$
end
$C \leftarrow C_t$
return C

4.2 Representation

To represent face images we learn two distinct set of filters by an unsupervised method similar to [85] (see Figure4.2(d)). First set is learned from the raw-pixel random patches extracted from grey-scale images. The second set is learned from LBP [86] encoded images. First set of learned filters are receptive to edge- and corner-like structural points and the second set is more sensitive to textural commonalities of the LBP histogram statistics.

LBP encoded images are invariant to illumination since the intensity relations between pixels are considered instead of exact pixel values. We use rotation invariant LBP encoding [87] that gives binary codes for each pixel. Then, we convert these binary codes into corresponding integer values. A Gaussian filter is used to smooth out the heavy-tailed locations.

The pipeline in order to learn filters from both raw-pixel and LBP images is as follows. First we extract a set of randomly sampled patches in the size of predefined



(a) (b) (c) (d)

Figure 4.2: Random set of filters learned from (a) whitened raw image pixels, and (b) LBP encoded images. (c) Outlier filters of raw-image filters. (d) LBP encoding of a given RGB image. We might observe eye or mount shaped filters from the raw image filters and more textural information from the LBP encoded filters. Outlier filters are very cluttered and observe low number of activations mostly from background patches.

receptive field. Then contrast normalisation is applied to each patch (for only raw-image filters) and patches are whitened to reduce the correlations among dimensions. These patches are clustered using k-means into K groups. We perform thresholding to centroids with box-plot statistics over the activations counts to remove the outlier centroids that are supposedly not representative for the face images but background clutters. After the learning phase, centroid activations are collected from receptive fields with small striding. We applied spatial average pooling onto five different grids including a grid at the center of the image additional to 4 equal-sized quadrants since face images includes important spatial regularities at the center. We use triangular activation function to map each receptive field to learned centroids. This yields a $5xK$ dimensional representation for each face. However, since we use two different set of filters, at the end, each image presented by $2x5xK$ dimensions. Thresholding of centroid activations provides a implicit removal of outlier patches as well as the salient set of centroids. We use those outlier centroids to eliminate patches at the feature extraction step by assuming the patches assigned to outlier centroids are not relevant thus avoiding them in pooling.

4.3 Experiments

4.3.1 Datasets

Images are collected using Bing to train models. Then, two recent benchmark datasets, FAN-large [10] and PubFig83[11], are used for testing.

Bing collection: For a given name, 500 images are gathered using Bing image search [1]. Categories are chosen as the people having more than 50 annotated face images in FAN-large or PubFig83 datasets. In total, 226691 images are collected corresponding to 365 name categories in FAN-large, and 83 name categories in PubFig83. Additional 2500 face images for queries "female face", "male face", "face images" are collected to construct the global negatives. Face detector of [8] is used for detecting faces. Only the most confident detection is selected from each image to be put into the initial pool of faces associated with the name (on the average 450 faces per category). Other detections are added to global negatives.

Test collection: We use two sets from FAN-large face dataset [10]: EASY and ALL. EASY subset includes faces larger than 60x70 pixels. ALL includes all names without any size constraint. We use 138 names from EASY, and 365 from ALL subsets, with 23952 and 199295 images respectively. On the average there are 541 images for each name.

We also use PubFig83[11] dataset, which is the subset of well-known PugFig dataset with 83 different celebrities having at least 100 images. PubFig83 is more convenient set for face identification problem with near-duplicate images and the ones that are no longer available at Internet are removed[88] . We shaped a controlled test environment by using PubFig83+LFW [88]: extending PubFig83 with some distract images from LFW [89] not belonging to any of the selected categories (distractors are six percent of correct instances). We use these distract images to extend our global negatives. For the controlled experiment, we select name categories with more than 270 images and mixed them with random set of distract images. Then we apply full

[1] $https://www.bing.com/$

stack of AME with 5-fold cross-validation.

4.3.2 Implementation Details

The dataset is expanded with horizontally flipped images. Before learning filters from raw-pixel images, each grey-level face image is resized to 60 pixels height and LBP images resized to 120 pixels height. LBP encoding has been done by 16 different filter orientation and at radius 2. We sample random patches from images and apply contrast normalization to only raw-pixel patches. Then, we perform ZCA whitening transform and set ϵ_{ZCA} to 0.5.

We use receptive field of 6x6 regions with 1 stride and learn 2400 centroids for both raw-pixel images and LBP encoded images. Hence, we conclude 2 (raw-pixel + LBP) x 5 (pooling grids) x 2400 (centroids) dimensional feature representation of each image. For instance to centroid distances we used Euclidean Distance. We detect the outliers by a threshold at the 99% upper whisker of the centroid activations. Our implementation of feature learning framework aggregated upon the code furnished by [85].

For iterative elimination, we train L1 norm Logistic Regression model with *Gauss-Seidel* algorithm [90] and final classification is done with Linear SVM through *grafting* algorithm [91] that learns sparse set of important features incrementally by using gradient information. At each AME iteration we eliminate five images. We stop when there is no improvement on the accuracy. If the classifier saturates so quickly, iteration continues until 10% of the instances are pruned. If we encounter memory constraints due to large number of global negatives, at each iteration we sample a different set of negative instances, to provide slightly different linear boundaries that are able to detect different spurious instances.

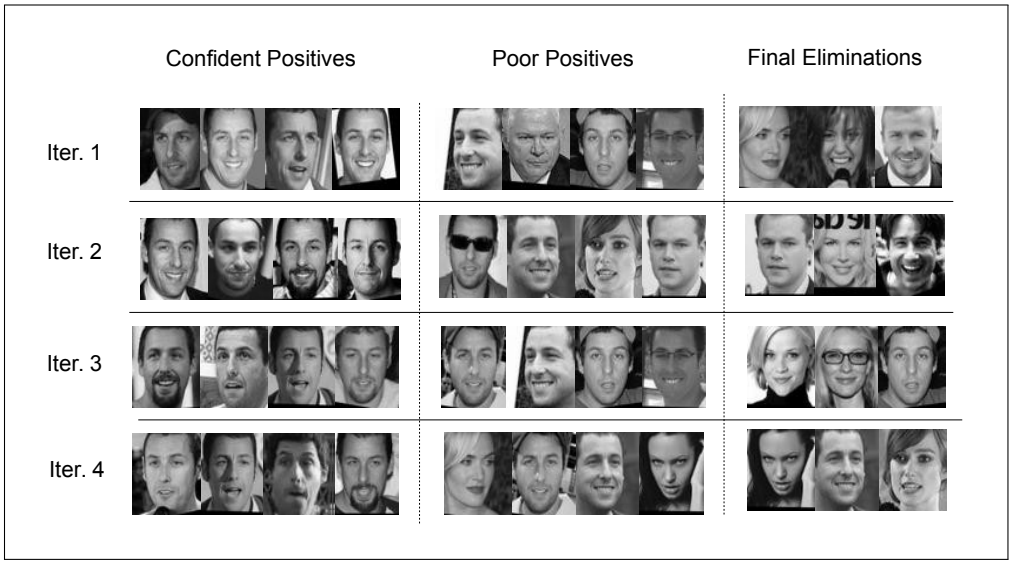


Figure 4.3: Some of the instances selected for C^+ (Confident Positives) that are selected as the most reliable instances by M_1, C^- (Poor Positives) that are close or wrong classification of M_1 and O final eliminations of the iteration. Figure depitcs iterations $t = 1 \ldots 4$.

4.3.3 Evaluations

We conduct controlled experiments over PubFig83+LFW. We select classes with at least 270 instances and inject 10% (27 instances) noise instances. There are six classes conforming that criterion. Noisy images are randomly chosen from global negatives consisting of "distract" set of PubFig83+LFW and FAN-large faces that we collected. As a result, we have 297x6 training instances. We apply AME to this data while applying cross-validation at each iteration step, between these six classes.

Figure4.3 helps to visualise the model evolution in AME. As shown on the left, at each iteration dataset is divided into candidate positives and possible negatives: candidate positives are selected as the most representative instances of the class and true outliers are found among the possible negatives. As shown on the right, AME is able to learn models from noisy weakly label sets, while eliminating the outliers at successive iterations for a variety of people.

As Figure4.3.3 shows with the increasing number of iterations, more outliers are eliminated. Although some correct instances are also eliminated, the ratio is very low

Figure 4.4: At the left column, random final images are depicted and at the following columns 2 iteration of elimination results are shown.

Feature	Accuracy
LBP filters	60.7
raw-pixel filters	71.6
LBP+raw-pixel filters	79.3

Num. Centroids	1500	2000	2400
Accuracy(%)	84.9	88.60	90.75

Table 4.1: (Left:) This table compares the performances obtained with different features on PubFig83 dataset with the models learned from web. As the figure suggests, even LBP filters are not competitive with raw-pixel filters, its textural information is subsidiary to raw-pixel filters with increasing performance. (Right:) Accuracy versus number of centroids k.

compared to the spurious instances. Moreover, our observations show that the eliminated positive examples might be overseened versus to the elimination of malicious instance as supported with the results in Figure4.3.3. As seen in Figure4.3.3, we can achieve up to 75.2 on FAN-Large (EASY) and 79.8 on PubFig83 by removing one outlier at each iteration: we prefer to eliminate five outliers for the efficiency. However, if you don't want to bother yourself with the best possible value, one elimination per iteration is the result guaranteeing setting with a bit of computational latency.

We compare AME with baseline method that learns models from the raw collection gathered through querying the name without any pruning. As seen in Table4.3.3

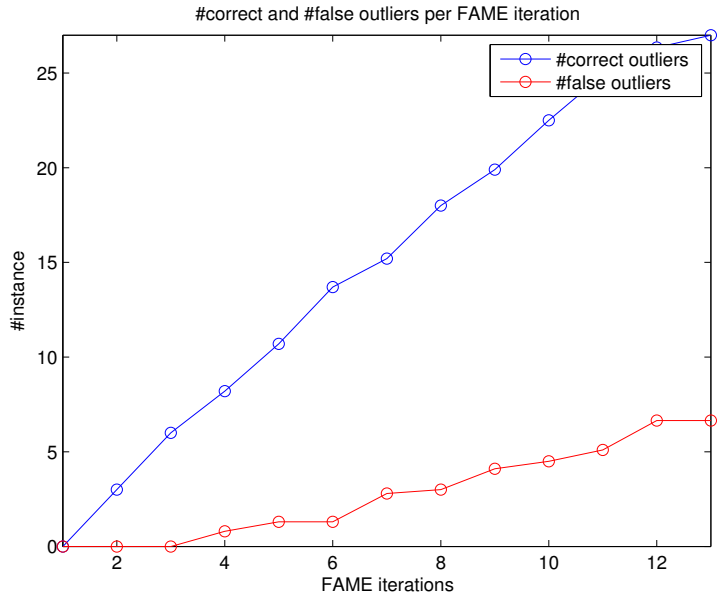


Figure 4.5: Incremental plot of correct versus false outlier detections until AME finds all the outliers for all classes. Each iteration values are aggregated by the previous iteration. For instance for iteration 6, there is no wrong elimination versus all true eliminations. We stop AME for the saturated classes before the end of the plot causing a bit of attenuation at the end of the plot.

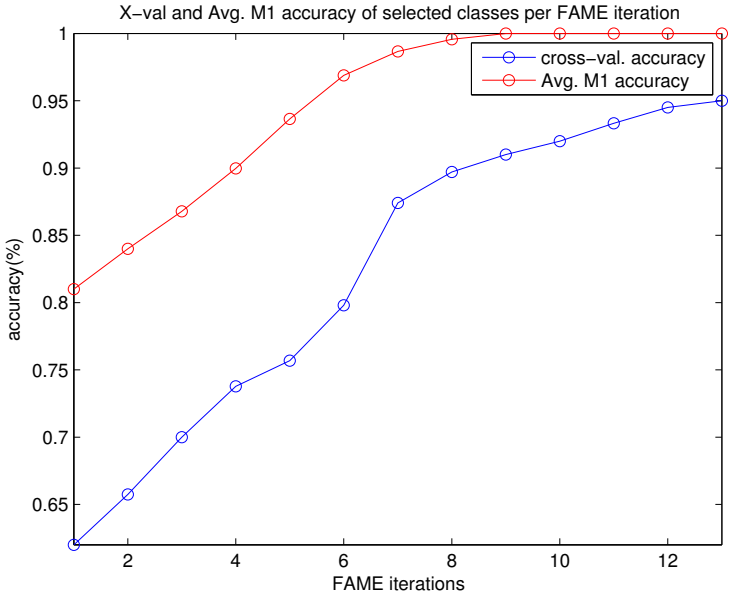


Figure 4.6: Cross-validation and M_1 accuracies as the algorithm proceeds. This shows the salient correlation between cross-validation classifier and M_1 models, without M_1 models incurring over-fitting.

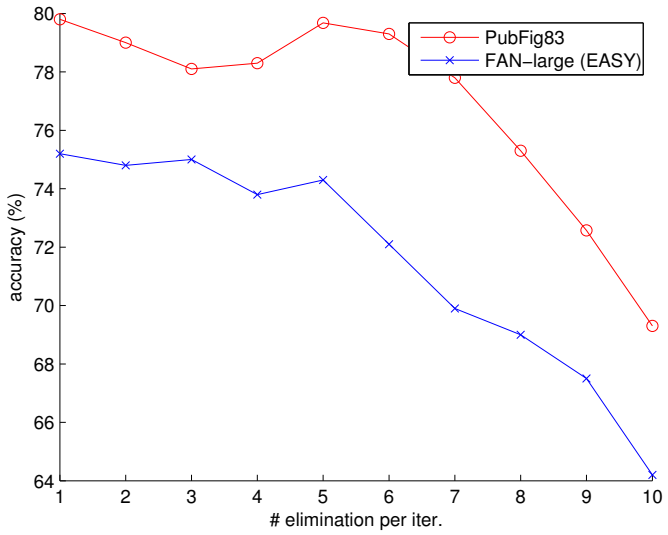


Figure 4.7: Effect of number of outliers removed at each iteration versus final test accuracy. It is observed that elimination after some limit imposes degradation of final performance and eliminating 1 instance per iteration is the salient selection without any sanity check.

with one vs all L1 norm Linear SVM model on the raw data, the performance is very low on all datasets. Note that, on the datasets FAN-Large EASY and ALL, as well as PubFig83, we learn the models from web images and tested them on these novel datasets for the same categories. We also divided the collected Bing images into two subsets to test the effect of training and testing on the same type of dataset. AME leads encouraging results even the model is susceptible to domain shifting problem, with a significant improvement over baseline.

The most similar data handling approach to ours is the method of Singh *et al.*[18], although there are important differences. First, [18] clusters the data to capture intra-cluster variance and uncover the representative instances. However, it requires to decide the optimal cluster number in advance and divides the problem into multiple homologous pieces which need to be solved separately. This increase the complexity of the proposed system. Second difference lies in the philosophy. They aim to discover representative and discriminative set of instances whereas we aim to prune spurious ones. Hence, they need to keep all vast negative instances on memory but we can sample different subsets of global negatives and find corresponding outlier instances.

-	Bing	FAN-Large (EASY)	FAN-Large (ALL)	PubFig83
Baseline	62.5	56.5	52.7	52.8
Singh *et al.*[18]	74.7	65.9	62.3	71.4
AME-M1	78.6	68.3	60.2	71.7
AME-SVM	81.4	73.1	65.4	76.8
AME-LR	83.7	74.3	67.1	79.3

Table 4.2: Accuracies (%) on FAN-Large [10] (EASY and ALL), PubFig83 and on the held-out set of our Bing data collection. There are three alternative AME implementations. AME-M1 uses only the model M1 which removes instances regarding global negatives. AME-SVM uses SVM in training and AME-LR is the proposed method using linear regression.

It provides faster and easier way of data pruning. They divide each class into two sets and apply their scheme by interchanging data after each iteration like in the case of co-training learning procedure. Nevertheless, co-training demands large number of instances for reliable results. In our methodology, we prefer to use all the class data at once in our particular scheme. We evaluate the method of Sing *et al.*on the same datasets, and show that AME is superior to their method (see Table4.3.3). We use the released code by Singh *et al.*[18] with up-limit settings that our resources allow.

To test the effectiveness of the proposed linear regression based model learning, we also compare our results by using only the M^1 model (AME-M1) and using SVM for classification (AME-SVM). As shown in Table4.3.3, all AME models outperforms the baseline method as well as the method of [18] with a large improvement with the proposed LR model.

Method	Pinto *et al.*[11] (S)	Pinto et al.[11](M)	face.com [11]	Becker *et al.*[88]	AME
Accuracy	75.6	87.1	82.1	85.9	90.75

Table 4.3: Accuracies (%) of face identification methods on PubFig83. [11] proposes single layer (S) and multi-layer (M) architectures. `face.com` API is also experienced in [11]. Note that, here AME is learned from the same dataset.

Finally, we compare the performance of AME on the benchmark PubFig83 dataset with the other state-of-the-art studies on face identification. In this case, unlike the previous experiments where we learned the models from noisy images, in order to

make a fair comparison we learned the models from the same dataset. As seen in Table4.3 AME achieves the best accuracy in this setting. Referring back to Table4.3.3 even with the domain adaptation setting where the model is learned from the noisy web images our results are comparable to the most recent studies on face identification that train and test on the same dataset. Note that, the method of Pinto *et al*.[92] is similar to our classification pipeline but we prefer to learn the filters in an unsupervised way with the method of Coastes *et al*.[85].

Chapter 5

Conclusion

This thesis presents two new methods to learn visual concepts through exploiting large volume of weakly labeled data on the web. We propose Concept Maps to organise the data and prune it from outliers. Multiple classifiers are then built for each group sensitive to a different visual variation of the concept. AME relies on large number of negative instances in selecting a set of good instances which are then used to learn models to eliminate the bad ones. The proposed methods outperforms the baseline and are comparable to state-of-the-art methods.

CMAP has the ability to categorise images and regions across datasets without being limited to a single source of data. Detailed evaluations show that CMAP is able to recognise low level attributes on novel images and has a good basis for higher level recognition tasks like scene recognition with inexpensive setting. It can also directly learn scene and object categories. Comparisons with the state-of-the-art studies in show that CMAP achieves competitive results to the other methods which use the same/similar web data for training or which require supervision.

AME contrives a new idea to data cleansing problem by taking the advantage of large amount of random images collected from the web. It evaluates disciminativeness and representativeness of each candiate category image of the given data with two different linear models and eliminates poor ones. Albeit the idea is very simple and easy to implements, it brings about very compelling result even in cheap hardware

configurations. Furthermore, the proposed method is tested for identification of faces but it is a general method that could be used for other domains as we aim to attack as our future work.

Bibliography

[1] E. Erdem and A. Erdem, "Visual saliency estimation by nonlinearly integrating features using region covariances," *Journal of Vision*, vol. 13, no. 4, pp. 1–20, 2013.

[2] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman, "Learning object categories from google's image search," in *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, vol. 2, pp. 1816–1823, IEEE, 2005.

[3] J. Van De Weijer, C. Schmid, J. Verbeek, and D. Larlus, "Learning color names for real-world applications," *Image Processing, IEEE*, 2009.

[4] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 2, pp. 2169–2178, IEEE, 2006.

[5] O. Russakovsky and L. Fei-Fei, "Attribute learning in large-scale datasets," in *Trends and Topics in Computer Vision*, pp. 1–14, Springer, 2012.

[6] A. Quattoni and A. Torralba, "Recognizing indoor scenes," *CVPR*, 2009.

[7] L.-J. Li and L. Fei-Fei, "Optimol: automatic online picture collection via incremental model learning," *International journal of computer vision*, vol. 88, no. 2, pp. 147–168, 2010.

[8] X. Zhu and D. Ramanan, "Face detection, pose estimation, and landmark localization in the wild," in *Computer Vision and Pattern Recognition (CVPR)*, 2012.

[9] J. R. Uijlings, K. E. van de Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," *International journal of computer vision*, vol. 104, no. 2, pp. 154–171, 2013.

[10] M. Özcan, J. Luo, V. Ferrari, and B. Caputo, "A large-scale database of images and captions for automatic face naming.," in *BMVC*, pp. 1–11, 2011.

[11] N. Pinto, Z. Stone, T. Zickler, and D. Cox, "Scaling up biologically-inspired computer vision: A case study in unconstrained face recognition on facebook," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2011 IEEE Computer Society Conference on*, pp. 35–42, IEEE, 2011.

[12] T. L. Berg and D. A. Forsyth, "Animals on the web," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 2, pp. 1463–1470, IEEE, 2006.

[13] T. L. Berg and A. C. Berg, "Finding iconic images," in *Computer Vision and Pattern Recognition Workshops, 2009. CVPR Workshops 2009. IEEE Computer Society Conference on*, pp. 1–8, IEEE, 2009.

[14] J. Fan, Y. Shen, N. Zhou, and Y. Gao, "Harvesting large-scale weakly-tagged image databases from the web.," in *CVPR*, pp. 802–809, 2010.

[15] F. Schroff, A. Criminisi, and A. Zisserman, "Harvesting image databases from the web," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 4, pp. 754–766, 2011.

[16] X. Chen, A. Shrivastava, and A. Gupta, "Neil: Extracting visual knowledge from web data," in *Proc. 14th International Conference on Computer Vision*, vol. 3, 2013.

[17] Q. Li, J. Wu, and Z. Tu, "Harvesting mid-level visual concepts from large-scale internet images," *CVPR*, 2013.

[18] S. Singh, A. Gupta, and A. A. Efros, "Unsupervised discovery of mid-level discriminative patches," in *European Conference Computer Vision (ECCV)*, 2012.

[19] G. Kim and A. Torralba, "Unsupervised detection of regions of interest using iterative link analysis.," in *NIPS*, vol. 1, pp. 4–2, 2009.

[20] C. Doersch, S. Singh, A. Gupta, J. Sivic, and A. A. Efros, "What makes paris look like paris?," *ACM Transactions on Graphics (TOG)*, vol. 31, no. 4, p. 101, 2012.

[21] C. Doersch, A. Gupta, and A. A. Efros, "Mid-level visual element discovery as discriminative mode seeking," in *Advances in Neural Information Processing Systems*, pp. 494–502, 2013.

[22] A. Jain, A. Gupta, M. Rodriguez, and L. S. Davis, "Representing videos using mid-level discriminative patches," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pp. 2571–2578, IEEE, 2013.

[23] I. Endres, K. J. Shih, J. Jiaa, and D. Hoiem, "Learning collections of part models for object recognition," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pp. 939–946, IEEE, 2013.

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

[25] Q. Li, J. Wu, and Z. Tu, "Harvesting mid-level visual concepts from large-scale internet images," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pp. 851–858, IEEE, 2013.

[26] F. Couzinié-Devy, J. Sun, K. Alahari, and J. Ponce, "Learning to estimate and remove non-uniform image blur," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pp. 1075–1082, IEEE, 2013.

[27] S. Singh, A. Gupta, and A. A. Efros, "Unsupervised discovery of mid-level discriminative patches," in *European Conference on Computer Vision (ECCV)*, 2012.

[28] A. Farhadi, I. Endres, and D. Hoiem, "D.: Attribute-centric recognition for cross-category generalization," 2010.

[29] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar, "Attribute and simile classifiers for face verification," *ICCV*, 2009.

[30] J. Choi, M. Rastegari, A. Farhadi, and L. S. Davis, "Adding unlabeled samples to categories by learned attributes," *CVPR*, 2013.

[31] S. Ma, S. Sclaroff, and N. Ikizler-Cinbis, "Unsupervised learning of discriminative relative visual attributes," in *2nd International Workshop on Parts and Attributes, in conjunction with 12th international conference on Computer Vision (ECCV'12)*, pp. 61–70, 2012.

[32] F. X. Yu, "Weak attributes for large-scale image retrieval," in *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, CVPR '12, (Washington, DC, USA), pp. 2949–2956, 2012.

[33] K. Duan, D. Parikh, D. Crandall, and K. Grauman, "Discovering localized attributes for fine-grained recognition," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pp. 3474–3481, June 2012.

[34] V. Sharmanska, N. Quadrianto, and C. Lampert, "Augmented attribute representations," in *European Conference on Computer Vision (ECCV 2012)*, pp. 242–255, 2012.

[35] D. Parikh, A. Kovashka, A. Parkash, and K. Grauman, "Relative attributes for enhanced human-machine communication," in *AAAI*, 2012.

[36] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid, "Label-embedding for attribute-based classification," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'13)*, pp. 819–826, June 2013.

[37] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth, "Describing objects by their attributes," *CVPR*, 2009.

[38] C. H. Lampert, H. Nickisch, and S. Harmeling, "Learning to detect unseen object classes by between-class attribute transfer," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 951–958, IEEE, 2009.

[39] L. Torresani, M. Szummer, and A. Fitzgibbon, "Efficient object category recognition using classemes," *ECCV*, 2010.

[40] M. Rastegari, A. Farhadi, and D. Forsyth, "Attribute discovery via predictable discriminative binary codes," *ECCV*, 2012.

[41] Y. Su, M. Allan, and F. Jurie, "Improving object classification using semantic attributes," in *Proceedings of the British Machine Vision Conference*, 2010.

[42] V. Ferrari and A. Zisserman, "Learning visual attributes," *NIPS*, 2008.

[43] K. McRae, G. S. Cree, M. S. Seidenberg, and C. McNorgan, "Semantic feature production norms for a large set of living and nonliving things," *Behavior research methods*, vol. 37, no. 4, pp. 547–559, 2005.

[44] C. Silberer, V. Ferrari, and M. Lapata, "Models of semantic representation with visual attributes.," in *ACL (1)*, pp. 572–582, 2013.

[45] J. Vogel and B. Schiele, "Natural scene retrieval based on a semantic modeling step," in *Image and Video Retrieval* (P. Enser, Y. Kompatsiaris, N. O'Connor, A. Smeaton, and A. Smeulders, eds.), vol. 3115 of *Lecture Notes in Computer Science*, pp. 207–215, Springer Berlin Heidelberg, 2004.

[46] L.-J. Li, H. Su, Y. Lim, and L. Fei-Fei, "Objects as attributes for scene classification," in *Trends and Topics in Computer Vision* (K. Kutulakos, ed.), vol. 6553 of *Lecture Notes in Computer Science*, pp. 57–69, Springer Berlin Heidelberg, 2012.

[47] A. Quattoni, M. Collins, and T. Darrell, "Learning visual representations using images with captions," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR '07)*, 2007.

[48] S. Marsland, U. Nehmzow, and J. Shapiro, "A model of habituation applied to mobile robots," 1999.

[49] S. Marsland, U. Nehmzow, and J. Shapiro, "Novelty Detection for Robot Neotaxis," *Proceedings 2nd NC*, 2000.

[50] T. Harris, "A kohonen som based, machine health monitoring system which enables diagnosis of faults not seen in the training set," *Neural Networks. IJCNN'93-Nagoya.*, 1993.

[51] A. Ypma, E. Ypma, and R. P. Duin, "Novelty detection using self-organizing maps," *In Proc. of ICONIP'97*, 1997.

[52] D. Theofilou, V. Steuber, and E. D. Schutter, "Novelty detection in a kohonen-like network with a long-term depression learning rule," *Neurocomputing*, vol. 52, no. 54, pp. 411–417, 2003.

[53] T. L. Berg, A. C. Berg, J. Edwards, M. Maire, R. White, Y. W. Teh, E. Learned-Miller, and D. A. Forsyth, "Names and faces in the news,"

[54] T. L. Berg, A. C. Berg, J. Edwards, and D. A. Forsyth, "Who's in the picture?,"

[55] P. Pham, M. Moens, and T. Tuytelaars, "Cross-media alignment of names and faces," *IEEE Transactions on Multimedia*, vol. 12, no. 1, pp. 13–27, 2010.

[56] D. Ozkan and P. Duygulu, "A graph based approach for naming faces in news photos,"

[57] D. Ozkan and P. Duygulu, "Interesting faces: A graph based approach for finding people in news," *Pattern Recognition*, vol. 43, pp. 1717–1735, May 2010.

[58] M. Guillaumin, T. Mensink, J. Verbeek, and C. Schmid, "Automatic Face Naming with Caption-based Supervision," in *Computer Vision and Pattern Recognition (CVPR)*.

[59] M. Guillaumin, J. Verbeek, and C. Schmid, "Is that you? Metric learning approaches for face identification," in *International Conference on Computer Vision (ICCV 2009)*, 2009.

[60] M. Guillaumin, J. Verbeek, and C. Schmid, "Multiple instance metric learning from automatically labeled bags of faces," in *European Conference on Computer Vision (ECCV)* , 2010.

[61] M. Guillaumin, T. Mensink, J. Verbeek, and C. Schmid, "Face recognition from caption-based supervision," *International Journal of Computer Vision*, vol. 96, pp. 64–82, Jan. 2012.

[62] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar, "Attribute and Simile Classifiers for Face Verification," in *International Conference on Computer Vision (ICCV)*, 2009.

[63] P. Pham, M. Moens, and T. Tuytelaars, "Naming persons in news videos with label propagation," *IEEE MultiMedia*, vol. 18, no. 3, pp. 44–55, 2011.

[64] G. Chiachia, N. Pinto, W. R. Schwartz, A. Rocha, A. X. Falcão, and D. D. Cox, "Person-specific subspace analysis for unconstrained familiar face identification.," in *BMVC*, pp. 1–12, 2012.

[65] E. G. Ortiz and B. C. Becker, "Face recognition for web-scale datasets," *Computer Vision and Image Understanding*, vol. 118, pp. 153–170, Jan. 2014.

[66] B. C. Becker and E. G. Ortiz, "Evaluating open-universe face identification on the web," in *Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2013.

[67] K. Simonyan, O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Fisher Vector Faces in the Wild," in *British Machine Vision Conference (BMVC)*, 2013.

[68] K. Yanai and K. Barnard, "Probabilistic web image gathering," in *Proceedings of the 7th ACM SIGMM international workshop on Multimedia information retrieval*, pp. 57–64, ACM, 2005.

[69] J. Hays and A. A. Efros, "Im2gps: estimating geographic information from a single image," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pp. 1–8, IEEE, 2008.

[70] T. Kohonen, *Self-organizing maps*. Springer, 1997.

[71] D. H. Hubel and T. N. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *The Journal of physiology*, vol. 160, no. 1, p. 106, 1962.

[72] A. Muñoz and J. Muruzábal, "Self-organizing maps for outlier detection," *Neurocomputing*, vol. 18, no. 1, pp. 33–60, 1998.

[73] J. Li, S. Ranka, and S. Sahni, "Gpu matrix multiplication," *Multicore Computing: Algorithms, Architectures, and Applications*, p. 345, 2013.

[74] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[75] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 7, pp. 971–987, 2002.

[76] M. Everingham, J. Sivic, and A. Zisserman, "Hello! my name is... buffy–automatic naming of characters in tv video," 2006.

[77] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 1, pp. I–511, IEEE, 2001.

[78] G. Bradski *Dr. Dobb's Journal of Software Tools*.

[79] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "Liblinear: A library for large linear classification," *The Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008.

[80] Y. Cheng, "Mean shift, mode seeking, and clustering," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 17, no. 8, pp. 790–799, 1995.

[81] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise.," in *Kdd*, vol. 96, pp. 226–231, 1996.

[82] M. Pandey and S. Lazebnik, "Scene recognition and weakly supervised object localization with deformable part-based models," *ICCV*, 2011.

[83] R. Kwitt, N. Vasconcelos, and N. Rasiwasia, "Scene recognition on the semantic manifold," *ECCV*, 2012.

[84] B. Alexe, T. Deselaers, and V. Ferrari, "Measuring the objectness of image windows," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 11, pp. 2189–2202, 2012.

[85] A. Coates, A. Y. Ng, and H. Lee, "An analysis of single-layer networks in unsupervised feature learning," in *International Conference on Artificial Intelligence and Statistics*, pp. 215–223, 2011.

[86] T. Ahonen, A. Hadid, and M. Pietikainen, "Face description with local binary patterns: Application to face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 12, pp. 2037–2041, 2006.

[87] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Gray scale and rotation invariant texture classification with local binary patterns," in *Computer Vision-ECCV 2000*, pp. 404–420, Springer, 2000.

[88] B. C. Becker and E. G. Ortiz, "Evaluating open-universe face identification on the web," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2013 IEEE Conference on*, pp. 904–911, IEEE, 2013.

[89] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," tech. rep., Technical Report 07-49, University of Massachusetts, Amherst, 2007.

[90] S. K. Shevade and S. S. Keerthi, "A simple and efficient algorithm for gene selection using sparse logistic regression," *Bioinformatics*, vol. 19, no. 17, pp. 2246–2253, 2003.

[91] S. Perkins, K. Lacker, and J. Theiler, "Grafting: Fast, incremental feature selection by gradient descent in function space," *The Journal of Machine Learning Research*, vol. 3, pp. 1333–1356, 2003.

[92] N. Pinto, Z. Stone, T. Zickler, and D. Cox, "Scaling up biologically-inspired computer vision: A case study in unconstrained face recognition on facebook," in *Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2011.