Hierarchical Spectral Clustering based Large Margin Classification of Visually Correlated Categories

Digbalay Bose [*]
IBM Research, Bangalore
digbbose@in.ibm.com

Subhasis Chaudhuri
Department of Electrical Engineering
Indian Institute of Technology, Bombay
sc@ee.iitb.ac.in

ABSTRACT

Object recognition is one of the challenging tasks in computer vision and the problem becomes increasingly difficult when the image categories are visually correlated among themselves i.e. they are visually similar and only fine differences exist among the categories. This paper has a two-fold objective which involves organization of the image categories in a hierarchical tree like structure using self tuning spectral clustering for exploiting the correlations among them. The organization phase is followed by a node specific large margin nearest neighbor classification scheme, where a Mahalnobis distance metric is learnt for each non-leaf node. Further a procedure for hyperparameters selection has been discussed w.r.t two strategies i.e. grid search and Bayesian optimization. The proposed algorithm's effectiveness is tested on selected classes of the popular Imagenet dataset.

CCS Concepts

•**Computing methodologies** → **Object recognition;** *Supervised learning by classification; Cluster analysis;*

Keywords

Object recognition, Visually correlated categories, Large margin nearest neighbor classification, Self Tuning Spectral Clustering, Hierarchical Organization

1. INTRODUCTION

Object recognition is a difficult task in the computer vision domain due to wide variations in the pose, shape and color. Even though humans are able to recognize multitude of objects inspite of these variations , accurate object recognition based on algorithms is a formidable task. The problem of object recognition is more challenging when the objects from different categories are visually similar to each other. For example, the cat category in Imagenet [4] when compared with computers is radically different but when compared with margay(a form of wild cat native to Central America) the differences are very subtle.

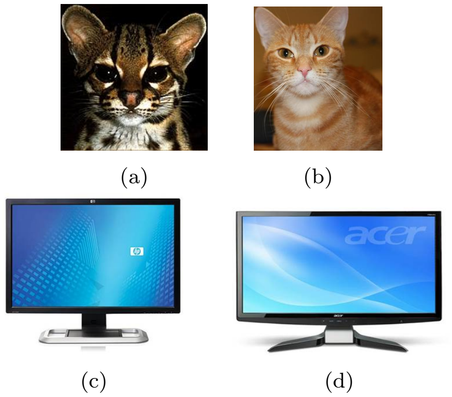
(a) (b) (c) (d)

Figure 1: (a) Sample image of the margay category (b) Sample image of the cat category (c) Sample image of the monitor category (d) Sample image of the computer screen category. High degree of visual similarity exists between images (a) and (b). Also images (c)and (d) are also visually similar. Image courtesy: Imagenet database[4]

.

In this paper we present an novel approach of combining both self tuning spectral clustering and large margin nearest neighbor algorithms for classifying visually correlated categories. Since our goal is to automatically determine the hierarchy among the categories, the self tuning variant is taken into account because of its ability to determine the number of clusters based on structure of eigenvectors of the normalized affinity matrix. Further the second objective involves learning of a Mahalnobis distance metric at each non leaf node of the hierarchical tree like strucure.

The distance metrics thus obtained are utilized in the energy based classification scheme of a test sample, which occurs in a top-down fashion, starting from root to a leaf node. The selection of the hyperparameters is performed using both grid search and Bayesian optimization, where both grid search and Bayesian optimization are applied on the entire tree structure.

Experiments have been performed on the popular benchmark i.e. Imagenet database, where two groups of visually similar categories, which are highly distinct from each other are considered. Results have been compared with state of

[*] The work was done at the Department of Electrical Engineering, Indian Institute of Technology, Bombay

ICVGIP, December 18-22, 2016, Guwahati, India

DOI: http://dx.doi.org/10.1145/3009977.3010064

the art algorithms like jDL [20], IMDL [20], FDDL [17], ScSPM [16], DKSVD [19] .

The rest of the paper is organized as follows. Section 2 introduces the works on different feature extraction schemes and object recognition. In Section 3 the Large Margin Nearest neighbor classification scheme has been detailed. Section 4 gives a brief idea about Self Tuning Spectral Clustering with specific emphasis on scale and cluster number determination. Section 5 includes the details of our proposed algorithm HSpecLMNN where the subsections include the methods for hierarchical organization of the image categories, node specific large margin nearest neighbor scheme and a top-down energy based classification scheme, respectively. Section 6 details the experimental setup including the features used, hyperparameter selection schemes and comparison with other algorithms, followed by conclusions in section 7.

2. RELATED WORK

The success of object recognition algorithms can be attributed to different state of the art feature extraction schemes which are currently in use. Most common technique involves dense sampling of the SIFT [11] descriptors followed by orderless representation in a BOW model [3].This BOW model has shown to be quite effective in certain object recognition tasks like PASCAL VOC challenge by Everingham [6] and scene recognition by Fei-Fei[7]. Since BOW is an orderless representation of an image Lazebnik extended it further by computing a spatial pyramid representation of the image and concatenating the BOW histogram in each bin of the pyramid to obtain a pyramid feature vector for the entire image. In order to use a linear SVM for classification, Yang [16] developed an extension of the Spatial Pyramid Scheme called ScSPM by using a sparse coding scheme for learning the vocabulary (instead of using Kmeans) and max pooling the sparse codes across multiple scales in the spatial pyramid to obtain feature vector representation of the images. The computational complexity of ScSPm was tackled in Wang's [14] work,where instead of sparse codes locality constrained linear coding was used followed by similar max-pooling strategy to obtain the feature vectors.

With the advent of the advanced feature extraction techniques, many state of the art algorithms have been proposed for object recognition problems. Attempts have been made to improve the standard K-NN classification scheme by incorporating distance metric learning in Chopra [2], Goldberger [9]. Weinberger [15] took the idea of distance metric learning further by obtaining a Mahalnobis distance metric and testing on problems like facial recognition, handwriting recognition and text categorization. In the field of supervised dictionary learning methods, Fisher discriminant criteria FDDL [17] has been incorporated on the sparse coefficients in order to learn class specific discriminative dictionaries which are applied on the problems of facial recognition and object categorization. Recent efforts have been made in the direction of large scale visual recognition challenge, made popular by the large scale Image database known as Imagenet, which contains images for 22K categories of varying types. Krizhevsky [10] trained a deep convolutional neural network model for classifying the 1.2 million images in the ImageNet LSVRC 2010 challenge. Bengio [1] proposed a tree structure based classifier by minimizing overall tree loss and applied it to Imagenet dataset. Deng[5] also proposed a label tree scheme which simultaneously learnt the tree structure and classifiers per node and obtained balanced trees as compared to Bengio's scheme. With the increasing popularity of the supervised dictionary learning schemes in classification tasks, Zhou[20] proposed a joint dictionary learning scheme with the aim of classifying visually correlated categories.

3. LARGE MARGIN NEAREST NEIGHBOR CLASSIFICATION

Large Margin Nearest Neighbor(LMNN) Algorithm proposed by Weinberger [15] aims at improving the k-NN classification scheme by minimizing the number of differently labelled examples in the k-nearest neighborhood of the training samples. It learns a linear transformation $\mathbf{T}$ such that in the transformed space the differently labelled examples are far apart and the k-nearest neighbors consist of examples having same labels. For a particular training sample s_i(label y_i) two types of neighbors are considered i.e. target neighbors (k nearest neighbors having same labels) and impostors (training samples having different labels but lying within the region marked by target neighbors).
Considering a target neighbor of s_i to be denoted by s_j, then a training sample s_l such that the label $y_l \neq y_i$ is called an impostor if the given inequality holds:

$$\|\mathbf{T}(s_i - s_l)\|_2^2 \leq \|\mathbf{T}(s_i - s_j)\|_2^2 + 1 \tag{1}$$

The cost function consists of two parts, where the first part is responsible for attracting the target neighbors closer whereas the second component aims at driving the impostors away from the boundary set by target neighbors.
The respective components are given by $Cost_{pull}(\mathbf{T})$ and $Cost_{push}(\mathbf{T})$ which are defined as follows:

$$Cost_{pull}(\mathbf{T}) = \sum_{i,j \rightsquigarrow i} \|\mathbf{T}(s_i - s_j)\|_2^2 \tag{2}$$

$$Cost_{push}(\mathbf{T}) = \sum_{i,j \rightsquigarrow i} \sum_{l} (1 - z_{il}) max(0, dist_{set}) \tag{3}$$

Here $dist_{set} = 1 + D_{\mathbf{T}}(s_i, s_j) - D_{\mathbf{T}}(s_i, s_l)$. Further $D_{\mathbf{T}}(s_i, s_j) = \|\mathbf{T}(s_i - s_j)\|_2^2$ and $j \rightsquigarrow i$ indicates that s_j is the target neighbor of s_i. Also $z_{il} = 0$ if $y_l \neq y_i$ and $z_{il} = 1$ if otherwise. Thus $Cost_{push}(\mathbf{T})$ incorporates the standard hinge loss formulation and takes into account a non-zero loss when the inequality in (1) is satisfied. The overall cost function is given by the convex combination of the two components $Cost_{pull}(\mathbf{T})$ and $Cost_{push}(\mathbf{T})$ as follows:

$$Cost_{total}(\mathbf{T}) = \alpha(Cost_{pull}(\mathbf{T})) + (1 - \alpha)(Cost_{push}(\mathbf{T})) \tag{4}$$

Here α is the weight associated with the pull component $Cost_{pull}(\mathbf{T})$ and $(1 - \alpha)$ is the weight associated with the push component $Cost_{push}(\mathbf{T})$.

Since $\|\mathbf{T}(s_i - s_j)\|_2^2 = (s_i - s_j)^t \mathbf{T}^t \mathbf{T}(s_i - s_j)$ and $\mathbf{T}^t \mathbf{T} = M$ we have $(s_i - s_j)^t \mathbf{T}^t \mathbf{T}(s_i - s_j) = (s_i - s_j)^t \mathbf{M}(s_i - s_j) = D_{\mathbf{M}}(s_i, s_j)$. By including the variable change $\mathbf{T}^t \mathbf{T} = \mathbf{M}$ the loss function in 4 can be rewritten as :

$$\begin{aligned} Cost_{total}(\mathbf{M}) = \alpha \sum_{i,j \rightsquigarrow i} D_{\mathbf{M}}(s_i, s_j) \\ + (1 - \alpha) \sum_{i,j \rightsquigarrow i} \sum_{l} (1 - z_{il}) max(0, d_{best}) \end{aligned} \tag{5}$$

where $d_{best} = 1 + D_{\mathbf{M}}(s_i, s_j) - D_{\mathbf{M}}(s_i, s_l)$. The cost function in 5 is a convex function of the elements in the matrix $\mathbf{M}$ and can be solved by posing as a Semidefinite Programming problem(SDP). Actual solver implemented by Weinberger [15] instead of using a SDP formulation directly solves (5) by an iterative sub-gradient based method and since the matrix $\mathbf{M}$ is positive semidefinite, projection operation is done onto the set S_+ (cone of positive semidefinite matrices). After learning of the Mahalnobis distance metric, energy based classification model inspired by Chopra [2] has been used instead of K-NN classification due to higher classification accuracy.

4. SELF-TUNING SPECTRAL CLUSTERING

Standard clustering techniques [12] including spectral based methods require determination of number of clusters beforehand. Many applications necessitate the determination of cluster numbers automatically since the manual specification of cluster number might not be an optimal choice. Self tuning version of spectral clustering proposed by Manor [18] addresses the issue of automatically selecting the number of clusters along with handling of the data having multiple scales.

4.1 Scale determination

Standard spectral clustering techniques determine the affinity between two sample points s_i and s_j using the following relation:

$$Af(i,j) = exp(\frac{\|s_i - s_j\|_2^2}{2\sigma^2}) \tag{6}$$

Here a single value of the scale parameter σ is used for computing the pairwise affinities. When the data has multiple scales i.e. a tight cluster consisting of smaller number of points exists within a sparse background cluster, a single value of σ is not an optimal choice . A scaling parameter σ_i is determined for each data member s_i by considering the distance from the pth neighbor of s_i.

$$\sigma_i = \|s_i - s_p\|_2 \tag{7}$$

The parameter p can be considered as the neighbor number associated with any datapoint. After determination of the local scaling parameter σ_i for datapoint s_i the distance from s_i to s_j as seen by s_i is given by $\frac{\|s_i - s_j\|_2}{\sigma_i}$. Conversely the distance from s_j to s_i is $\frac{\|s_j - s_i\|_2}{\sigma_j}$. Using both the distances the entries of the affinity matrix $Af(i,j)$ are computed as follows:

$$\begin{aligned} Af(i,j) &= exp(-\frac{\|s_i - s_j\|_2^2}{\sigma_i \sigma_j}) \quad & i \neq j \\ &= 0 & i = j \end{aligned} \tag{8}$$

4.2 Cluster number determination

Automatic determination of the number of clusters depend upon the structure of the eigen vectors associated with the normalized affinity matrix $L = D^{-\frac{1}{2}} Af D^{\frac{1}{2}}$ where the affinity matrix Af is calculated using (8) and the entries of degree matrix D are computed using the following:

$$D(i,i) = \sum_{j=1}^{nd} Af(i,j) \tag{9}$$

In the ideal case when the clusters are widely separated from each other the matrix L has a block diagonal structure with the number of blocks equal to number of clusters. Due to block structure of the matrix L overall eigenvalues are obtained by the union of the eigenvalues of the individual blocks and the eigenvectors are obtained after stacking the eigenvectors of the individual blocks with zeros in specific locations. By stacking the first c_{num} eigen vectors of matrix $L(L$ having a block diagonal structure) in E_c, the matrix E_c is given as follows:

$$E_c = \begin{bmatrix} e^1 & 0 & \dots & 0 \\ 0 & e^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & e^{c_{num}} \end{bmatrix} \tag{10}$$

Here e^k is the eigenvector associated with the sub-matrix L_k , which coressponds to cluster k. Given the matrix E_c and a rotation matrix $\tilde{R}$, the matrix $E = E_c\tilde{R}$ is such that its rows remain orthogonal to each other and the columns of E has the same span as E_c. Thus it is possible to obtain E_c from E using the relation $E_c = ER$, where $R = \tilde{R}^t$

For each cluster number c_{num} upto maximum number of clusters C_{max}, the rotation matrix R is obtained which results in best alignment of the columns of E with the standard basis system. Considering the matrix after rotation to be denoted by $M_r = ER$ and $m_i = \underset{j}{max}\ (M_r)_{ij}$, the quality of alignment of the set of eigenvectors in E is given by :

$$Cost_{rotate} = \sum_{i=1}^{nd} \sum_{j=1}^{c_{num}} \frac{(M_r)_{i,j}^2}{m_i^2} \tag{11}$$

Here nd is the number of datapoints considered for clustering. The rotation matrix R is recovered such that there is a single non-zero entry in each row of the matrix M_r. The optimum cluster number is selected to be that value of c_{num} which provides the minimum value of cost function in (11).

5. PROPOSED ALGORITHM

In this section the proposed algorithm HSpecLMNN has been detailed with specific emphasis on the hierarchical organization of the image categories, node specific Mahalnobis metric learning and energy based classification model.

5.1 Hierarchical organization of visually correlated categories

The first step in our proposed method is to obtain a hierarchical organization of the given image categories in a tree like structure. One possible alternative is to apply top down hierarchical clustering where one starts with the entire data in the root node and uses a flat clustering technique like k-means at each level to split the clusters in one level into finer clusters. But such a technique involves prespecifying the number of clusters into which a particular group of data points must be split. Our goal is to automatically learn the number of clusters from the given data points and apply the same technique in a recursive manner to determine the underlying tree structure .

Since the self tuning clustering algorithm can determine the number of clusters from a given set of datapoints, it is used to obtain the hierarchical structure of the set of given N image categories $C = \{C_1, C_2, \dots C_N\}$. Since the goal is to determine which image categories can be grouped together, instead of applying the clustering operation on the entire set of images of different classes, it is done on the set of characteristic members of the image categories. For the c^{th} image category C_c with N_c members (feature vectors) $[feat_1, feat_2, feat_3, \ldots feat_{N_c}]$ the characteristic member of the c^{th} image category is given by :

$$char_c = \frac{\sum_{i=1}^{N_c} feat_i}{N_c} \tag{12}$$

After l_2 normalization of the characteristic members, the self tuning spectral clustering technique is applied on N such characteristic members. The basic steps involved in determination of the hierarchical structure are listed below:

- Start from the root node which contains the characteristic members of all the N image categories computed using (12). Apply self tuning spectral clustering on the set of characteristic members and obtain the clusters, where each cluster is a child of the root node. If a particular non leaf node has greater than 2 image categories, apply the same technique recursively until all the child nodes have atmost 2 image categories associated with them. Given a non leaf node n at level l containing the set of image categories $Class_n = \{C_k, C_{k+1}, \ldots C_r\}$ and the set of classes associated with the ith child node of n denoted by $child_i$, then the following property holds:

$$\begin{gathered} Class_n = \bigcup_{i=1}^{n_c} child_i \\ child_i \bigcap child_j = \phi \end{gathered} \tag{13}$$

  Here n_c determines the number of child nodes associated with the node n. Thus the division of the image categories among the child nodes is non-overlapping in nature.

- If the node of the tree has only two associated class labels (two image categories) then instead of applying self tuning spectral clustering on the characteristic members of the two categories , split the current node into two leaf nodes with each leaf node having an associated class label.

The above process of determining the hierarchical structure stops when the number of leaf nodes equal the number of image categories. The advantage of clustering using characteristic members is that it can be used for learning the hierarchical structure in databases having large number of categories like Imagenet which has around 22K image categories.

5.2 Node specific large margin nearest neighbor algorithm

The organization of the different classes of images in a hierarchical tree structure is followed by learning of a Mahalnobis metric at each non-leaf node. Considering the ith child node , a broad class label is assigned to it such that for n_c child nodes, there are n_c possible broad classes available for selection at the location of the parent node. Thus the selection of the appropriate child node is cast as a multi class classification problem at the parent node. After the assignment of the broad class labels to the child nodes, a Mahalnobis metric $\mathbf{M_{par}}$ is learnt at the parent node. The main idea behind learning a Mahalnobis metric is to ensure that the number of impostors in the perimeter of the considered training sample (the perimeter being set up by its target neighbors) is minimized after learning the metric. In this case the target neighbors of a training sample indicate the k nearest neighbors among the members of the child node, that the training sample is a part of. The impostors refer to the training samples having different broad class labels i.e. part of different child nodes, but invading the perimeter set by the target neighbors. Thus the formulation for learning $\mathbf{M_{par}}$ at a non leaf node is given by minimizing the following function:

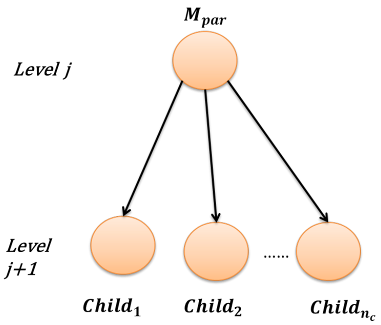

Figure 2: Node specific learning of the Mahalnobis distance metric M_{par} at level j with the corresponding child nodes $(child_1, child_2, \ldots child_{n_c})$at level $j+1$. n_c broad class labels are available for selection at the parent node's location.

.

$$\begin{aligned} Cost(\mathbf{M_{par}}) = \alpha \sum_{i,j \rightsquigarrow i} D_{\mathbf{M_{par}}}(s_i, s_j) \\ + (1-\alpha) \sum_{i,j \rightsquigarrow i} \sum_{l} (1 - z_{il}) d_{bestparent} \end{aligned} \tag{14}$$

Here $d_{bestparent} = max(0, 1 + D_{\mathbf{M_{par}}}(s_i, s_j) - D_{\mathbf{M_{par}}}(s_i, s_l))$ Here s_i refers to a particular training sample, whose class label is one of the broad class labels given to the child nodes. s_j and s_l denote its target neighbor and impostor respectively , and they are defined previously on a per node basis . Further $D_{\mathbf{M_{par}}}(s_i, s_j) = (s_i - s_j)^t \mathbf{M_{par}} (s_i - s_j)$.

5.3 Classification Scheme

For the classification of test sample y, the energy based model proposed by Chopra [2] and considered by Weinberger [15] was used on a per node basis instead of standard K-NN classification scheme. Standard energy based models consider a test sample as an extra training sample and evaluate the cost function given by eqn. 5 for all possible labels that are assigned to the test sample. That particular label which results in the minimum value of the cost function is the predicted label for the test sample.

In case of the hierarchical model considered, the energy based classification is performed at each non leaf node in order to select the child node for traversal in the next level. Since the child nodes have associated broad class labels, the test sample while arriving at the non leaf node is assigned each of the broad class labels and the child node whose label gives the minimum value of cost function is selected for next level. For a particular non leaf node at level l of the tree(associated Mahalnobis metric $\mathbf{M_{par}}$) with n_c number of child nodes , the broad class label of a child node y_{child} is such that $y_{child} \in \{1, 2, ..., n_c\}$. For the test sample s_t with an assigned class label $y_t \in \{1, 2, ..., n_c\}$, the cost function in (14) when evaluated has three terms

- $Cost_1 = \alpha \sum_{j \rightsquigarrow t} D_{\mathbf{M_{par}}}(s_t, s_j)$. Here $j \rightsquigarrow t$ refers to the k nearest neighbors of s_t having the same label y_t i.e. the k-nearest members of child node having broad class label y_t.

- $Cost_2 = (1-\alpha) \sum_{j \rightsquigarrow t} \sum_l (1 - z_{tl}) d_{best1}$. Here $d_{best1} = max(0, 1 + D_{\mathbf{M_{par}}}(s_t, s_j) - D_{\mathbf{M_{par}}}(s_t, s_l))$. This term takes into account the hinge loss over all the impostors that invade the boundary set by the target neighbors of s_t. Here the impostors refer to the members of the child nodes having broad class labels different from y_t and lying within the periphery set by the target neighbors.

- $Cost_3 = (1-\alpha) \sum_i \sum_{j \rightsquigarrow i} (1 - z_{it}) d_{best2}$. Here $d_{best2} = max(0, 1 + D_{\mathbf{M_{par}}}(s_i, s_j) - D_{\mathbf{M_{par}}}(s_i, s_t))$. This term includes the contribution of the test sample s_t, when it acts as an impostor for other training samples i.e. members of other child nodes(having broad class labels different from y_t) .

Thus the predicted broad class label for the test sample y_t is given by :

$$y_{pred} = \arg\min_{y_t}(Cost_1 + Cost_2 + Cost_3) \tag{15}$$

Here $y_{pred} \in \{1, 2, ..., n_c\}$. For determining the node number to be considered for traversal in the next level, the tree nodes are marked in level order fashion and an array $node_{mark}$ is maintained for each level of the tree . For the next level $l + 1$ the numbering of the node selected by (15) is given by $node_{mark}(y_{pred})$. This process of tree traversal starts from the root node and continues until a leaf node is reached, whose associated class label is the final predicted label for the test sample s_t.

6. EXPERIMENTAL RESULTS

For experiments the Imagenet[4] database consisting of multiple visually correlated categories is considered. The Imagenet dataset was considered due to the greater correlation among multiple classes as compared to other datasets. The categories are organized using Wordnet[8] hierarchy and each category has a unique wordnet id($wnid$). The entire Imagenet database has a massive collection of data (22K image categories overall and 1000 categories in the ILSVRC challenges)and learning of hyperparameters for the entire set will take months unless one has access to enormous computational resources. Hence,for the purpose of our experiments two groups of highly visually correlated 11 classes which are distinct from each other but correlated among themselves are considered. The image categories chosen were the same considered in [20]. The categories and the synset ids are listed below:

- **Group 1:**
  1. **Dog**($wnid : n$02084071)
  2. **Hound**($wnid : n$02087551),
  3. **Whippet**($wnid : n$02091134),
  4. **Cat**($wnid : n$02121620),
  5. **Margay**($wnid : n$02126640)
- **Group 2:**
  1. **Computer Monitor**($wnid : n$03085219),
  2. **Computer Screen**($wnid : n$03086502),
  3. **Desktop Computer**($wnid : n$03180011),
  4. **Keyboard**($wnid : n$03614007),
  5. **Laptop**($wnid : n$03642806),
  6. **Television**($wnid : n$04404412)

Before feature extraction , the images were cropped to obtain the object parts by using the bounding boxes given in XML format.The XML files were parsed using PASCAL development toolkit [6]. After cropping, the multiple objects in the images are saved resulting in 6313 images of the 11 image categories.

For feature extraction, a dense sampling strategy was used to obtain the SIFT[11] descriptors of the images. The patch and step sizes were fixed at 16 and 6 respectively and the codebook size considered was 1024. The encoding of the SIFT descriptors was performed using LLC[14] scheme followed by max-pooling of the LLC codes across multiple scales and locations in order to obtain the spatial pyramid feature vector of the image. Further the dimensions of the spatial pyramid feature vectors were reduced using PCA.

6.1 Hyperparamter selection

The set of hyperparameters to be selected include: **K**(number of target neighbors), **outdim**(number of rows of linear transformation matrix **T**),**maxiter**(number of iterations required for training) The value of α in (14) was fixed at 0.5. For hyperparameter selection two strategies were used i.e.grid search and Bayesian optimization[13].

For grid search techniques, hyperparameters were found for the entire tree instead of individual non leaf nodes. **K** was varied in the range $\{3, \ldots, 15\}$ and **outdim** was varied in the range $\{1000, \ldots, 2000\}$ in steps of 100. The maximum iterations , **maxiter** was fixed to 200.

For Bayesian optimization the ranges of target neighbors **K** and **outdim** were $\{1, \ldots, 15\}$ and $\{2, \ldots, rval\}$ respectively. $rval$ refers to the number of rows in the training sample. The **maxiter** range was varied between 10 and 200. Selection of the hyperparameters using Bayesian optimization was performed for the entire tree instead of each individual node.

The entire dataset was split randomly($80 - 20\%$) into a training and test set. The training set was further split randomly($80 - 20\%$) into a validation and training set. We considered 30 such splits of training and validation sets and the results of the best performing tree structure is reported. The best performing tree structure is shown in Figure 3. Using grid search and Bayesian optimization the optimal hyperparameters obtained are $\mathbf{K} = 4, \mathbf{outdim} = 1600$ and $\mathbf{K} = 2, \mathbf{outdim} = 3463, \mathbf{maxiter} = 16$ respectively. In order to compare the performances of hyperparameters selection using grid search and Bayesian optimization, the best performing tree structure is kept fixed and the for the same training-test split, the recognition accuracies are listed.

Grid Search	Bayesian Optimization
58.31%	69.23%

Table 1: **Comparison of the best recognition accuracies for the tree structure($tree_1$) in case of grid search and Bayesian optimization**

Since Grid search's performance is inferior as compared with Bayesian optimization, the hyperparameters selected by the later are considered to be optimal for the entire tree and used in our subsequent experiments.

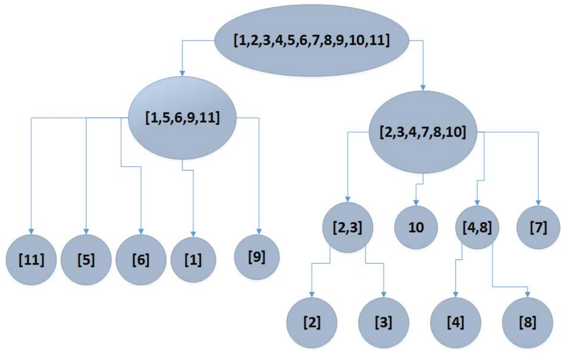

Figure 3: Best performing tree structure ($tree_1$) in terms of classification accuracy. ([1]:Cat, [2]:Monitor, [3]:Screen, [4]:Desktop, [5]:Dog, [6]:Hound, [7]:Keyboard, [8]:Laptop, [9]:Margay, [10]:Television, [11]:Whippet)

In the first level of the tree the characteristic members of all the classes are present as denoted by the groupings of the 11 classes. In the second level of the tree the classes of the Group 2(composed of the monitor, screen, desktop, keyboard, laptop and television) are separated from the classes of Group 1(composed of the dog, hound, whippet, cat, margay), thus ensuring that the visually correlated classes are only grouped together. In the third level finer groupings are obtained i.e. $[2, 3]$ (composed of computer monitor and screen classes) and $[4, 8]$(composed of desktop and laptop classes).

6.2 Effect of neighbor number p

The variation of the tree structure with the neighbor number p as mentioned in eq. (7) is discussed here. In [18] the neighbor number $p = 7$ was found to be suitable for their different datasets. In our case the neighbor number was varied in two different ways:

- Neighbor number p was fixed at 7. In that case the tree structure consists of all leaf nodes at the third level and there are no finer groupings obtained for the different classes at the third level. The resulting tree structure($tree_2$) is shown in Fig 4.
- Neighbor number p was varied depending on the number of classes in each non leaf node since each non leaf node has the number of characteristic members equal to some number of classes. A rule was considered based on the number of classes associated with each non-leaf node in order to obtain finer groupings:
  - $numclass > 15$**:** $p = 7$
  - $numclass \geqslant 10$ & $numclass < 15$**:** $p = 5$
  - $numclass \geqslant 5$ & $numclass < 10$**:** $p = 3$
  - $numclass < 5$**:** $p = 2$

Here the $numclass$ refers to the number of classes in each non-leaf node. The tree structure shown in Fig 3($tree_1$) is obtained by varying the neighbor number p according to $numclass$, instead of fixing a single value of p. For comparing the two tree structures, the same training and test split was considered and the overall classification accuracies were reported after hyperparameters selection using Bayesian optimization .

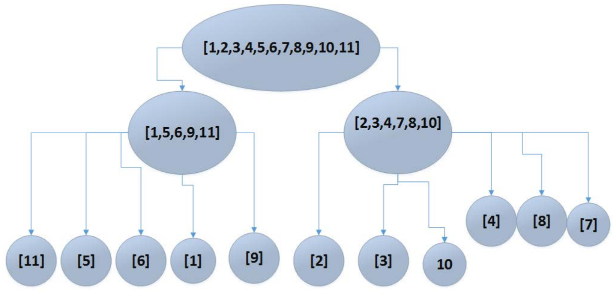

Figure 4: Tree structure ($tree_2$) obtained when the neighbor number p is kept fixed at 7 for the entire tree.

$tree_1$	$tree_2$
69.23%	66.21%

Table 2: **Comparison of the overall best classification accuracies of the tree structure $tree_1$(Figure 3) and $tree_2$(Figure 4)**

In the tree structure $tree_2$ the third level is composed of leaf nodes whereas the tree structure $tree_1$ consists of finer groupings between the monitor and screen classes[2,3] and desktop and laptop classes [4,8]. As evident from the results in Table 2, the finer groupings in the third level of $tree_1$ resulted in the increase of classification accuracy, when compared with the tree structure $tree_2$.

6.3 Comparison with state of the art

The results of our proposed algorithm HSpecLMNN has been compared with FDDL[17], JDL[20],ScSPM[16],IMDL [20] and DKSVD [19]. The results reported for these algorithms are those listed in JDL[20]where the average results of 10 random training and test splits are considered. In their setup they have considered 4723 images(1491 images of Group 1 and 3723 images of Group 2) and used sparse coding scheme for feature descriptors instead of LLC and used a codebook of 1024 atoms. For comparison purposes we considered the class specific recognition accuracies.

In our setup we have considered all 6313 images obtained as a result of cropping operation. Further we also fix the best performing tree structure and test its effectiveness by reporting the average accuracies of 10 random training-test splits(80-20%). The motivation behind using LLC instead of sparse coding in our case was the reduced complexity in LLC process without causing any significant decrease in the performance of the algorithm. The tables 3 and 4 list the

Algorithm	Cat	Dog	Hound	Margay	Whippet
HSpecLMNN	43.74 ± 6.76	**58.26 ± 6.08**	54.32 ± 7.62	79.65 ± 7.07	45.56 ± 3.82
FDDL	66.26 ± 7.08	44.67 ± 10.09	57.52 ± 6.91	68.06 ± 6.67	**62.57 ± 7.34**
ScSPM	71.35 ± 6.71	45.43 ± 3.55	58.85 ± 4.63	80.71 ± 9.56	41.63 ± 6.55
jDL	**71.67 ± 3.22**	57.14 ± 4.69	59.62 ± 4.21	**89.29 ± 2.70**	53.06 ± 3.09
DKSVD	59.46 ± 5.23	38.57 ± 6.34	57.69 ± 9.84	87.50 ± 8.51	38.78 ± 5.77
IMDL	71.62 ± 3.94	54.57 ± 6.73	**61.54 ± 7.69**	86.79 ± 3.70	42.04 ± 4.69

Table 3: **Recognition accuracies(%) of the Cat, Dog Hound, Margay and Whippet categories**

Algorithm	Monitor	Screen	Desktop	Keyboard	Laptop	Television
HSpecLMNN	**52.05 ± 4.91**	**63.26 ± 6.32**	78.94 ± 4.32	93.10 ± 5.01	**57.83 ± 3.10**	70.71 ± 4.01
FDDL	43.75 ± 7.98	43.75 ± 8.39	48.99 ± 7.52	98.04 ± 0.31	41.19 ± 8.60	61.60 ± 5.61
ScSPM	24.58 ± 4.00	39.22 ± 3.92	80.77 ± 6.01	96.65 ± 1.25	54.88 ± 6.73	77.38 ± 3.88
jDL	41.67 ± 6.97	53.85 ± 5.38	**83.08 ± 5.02**	92.31 ± 2.28	57.14 ± 2.87	**81.48 ± 0.67**
DKSVD	22.92 ± 11.28	33.33 ± 5.11	82.69 ± 4.13	**98.05 ± 1.01**	46.34 ± 4.98	73.83 ± 3.68
IMDL	29.58 ± 4.56	43.14 ± 6.73	82.69 ± 1.52	96.11 ± 0.63	56.13 ± 6.98	80.37 ± 3.19

Table 4: **Recognition accuracies(%) of the Monitor, Screen, Desktop, Keyboard, Laptop and Television categories**

category specific recognition accuracies of the 11 image categories. It can be seen from the results in Tables 3 and 4 that HSpecLMNN achieved the highest recognition accuracies in 4 classes out of 11 classes considered. The performance improvements in cases of dog, monitor and screen classes were significant. Further HSpecLMNN was also able to improve the classification accuracy of the laptop class and showed comparable performance in case of desktop .

6.4 Training times and convergence plots

The training times for each non leaf node of the tree structure are listed . The training times are mentioned in seconds and reported only after the best hyperparameters are found for the entire tree using Bayesian Optimization. Since the nodes are marked in level-order manner, the root node of the tree (displayed in Fig (3))is marked 1 in level 1. Similarly, the node containing the classes [2,3] is marked node 6 of level 3. Thus the notation a/b denotes the node numbered a in level b. Since we fix the best performing tree structure and consider 10 random training-test splits(80-20%), the training times are listed as average of 10 such runs for each non-leaf node. Using the notation a/b as mentioned above, the training times are given as:

Node number/level number	Training time (in seconds)
1/1	286.73
1/2	234.72
2/2	436.66
6/3	77.75
8/3	99.94

Table 5: **Average training times associated with each non-leaf node in seconds after the hyperparameters are fixed using bayesian optimization**

The convergence plots for the non leaf nodes of the tree structure($tree_1$) are given in the Figure 5.

7. CONCLUSIONS

In this paper we consider the problem of classifying visually correlated categories using a two step process where the first step involves organization of the image categories on the basis of visual similarities in a hierarchical tree like structure followed by utilization of the tree structure for classification. For organization purpose the self tuning variant of the spectral clustering is applied on the set of characteristic members of the image categories in a recursive manner in order to determine the tree structure. The tree structure is further utilized by learning a Mahalnobis distance metric at each non-leaf node via the LMNN framework . The classification of the test samples are done in a top-down fashion starting from the root node to the leaf nodes by using an energy based model.

Future work involves testing the proposed model for larger number of visually correlated image categories since the procedure for determining the tree structure can be scaled to increasing number of categories. Further this scheme of hierarchy determination can be extended to cross domain recognition tasks.

8. REFERENCES

[1] S. Bengio, J. Weston, and D. Grangier. Label embedding trees for large multi-class tasks. In *Advances in Neural Information Processing Systems*, pages 163–171, 2010.

[2] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 539–546, 2005.

[3] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV*, volume 1, pages 1–2. Prague, 2004.

[4] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

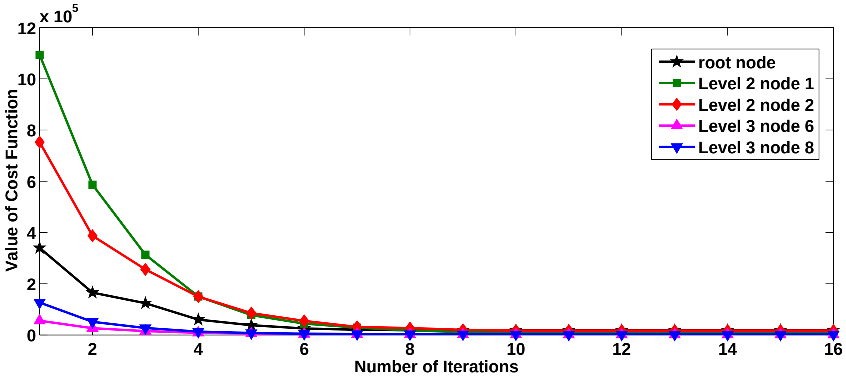

Figure 5: Convergence plot of the non leaf nodes of the tree. Variation of the cost function value in eq.(14) for each non leaf node.

[5] J. Deng, S. Satheesh, A. C. Berg, and F. Li. Fast and balanced: Efficient label tree learning for large scale object recognition. In *Advances in Neural Information Processing Systems*, pages 567–575, 2011.

[6] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

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

[8] I. Feinerer and K. Hornik. *wordnet: WordNet Interface*, 2016. R package version 0.1-11.

[9] J. Goldberger, G. E. Hinton, S. T. Roweis, and R. Salakhutdinov. Neighbourhood components analysis. In *Advances in neural information processing systems*, pages 513–520, 2004.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[11] D. G. Lowe. Object recognition from local scale-invariant features. In *IEEE international conference on Computer vision*, volume 2, pages 1150–1157, 1999.

[12] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems*, pages 849–856. MIT Press, 2001.

[13] J. Snoek, H. Larochelle, and R. P. Adams. Practical bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*, pages 2951–2959, 2012.

[14] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained linear coding for image classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3360–3367, 2010.

[15] K. Q. Weinberger, J. Blitzer, and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. In *Advances in neural information processing systems*, pages 1473–1480, 2005.

[16] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1794–1801, 2009.

[17] M. Yang, L. Zhang, X. Feng, and D. Zhang. Sparse representation based fisher discrimination dictionary learning for image classification. *International Journal of Computer Vision*, 109(3):209–232, 2014.

[18] L. Zelnik-Manor and P. Perona. Self-tuning spectral clustering. 2005.

[19] Q. Zhang and B. Li. Discriminative k-svd for dictionary learning in face recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2691–2698. IEEE, 2010.

[20] N. Zhou, Y. Shen, J. Peng, and J. Fan. Learning inter-related visual dictionary for object recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3490–3497, 2012.