©2016 Society for Imaging Science and Technology
DOI: 10.2352/ISSN.2470-1173.2016.19.COIMG-152

Hierarchical Decomposition of Large Deep Networks

Sumanth Chennupati, Shagan Sah, Sai Nooka, Raymond Ptucha; Rochester Institute of Technology; Rochester, NY/USA

Abstract

Deep networks have revolutionized the image, speech, and pattern recognition communities. Despite recent evidence showing deep networks can rival the human brain for visual object recognition, the expansion of such architectures to general-purpose intelligent reasoning is intractable due to the number of training parameters. Hierarchical representations have been introduced, but either have been applied to small problems, or have been ad hoc in nature. This paper introduces a framework that automatically analyzes and configures a family of smaller deep networks as a replacement to a singular, larger network. By analyzing the linkage coefficients from confusion matrices and class boundaries from spectral clustering, class clusters and sub-clusters are automatically detected, enabling the framework to divide and conquer large classification problems. The resulting smaller networks are not only highly scalable, parallel and more practical to train, but also achieve higher classification accuracy. Numerous experiments on network classes, layers, and architecture configurations validate our results.

Introduction

Deep architectures [1] with hierarchical frameworks enable the representation of complex concepts with fewer nodes than shallow architectures. With regard to object classification, these networks have recently been shown to equal the performance of neurons in the primate inferior temporal cortex [2], even under difficult conditions such as pose, scale, and occlusions. It has been shown that network depth generally is more important than the number of nodes in each layer [3], with modern architectures containing more than 20 layers [4], requiring the solution of over 100M parameters. As the classification task becomes more difficult, the number of parameters increases exponentially.

This paper introduces a multi-layer hierarchical framework to reduce the overall number of solvable parameters by subdividing the classification task into smaller intrinsic problems. Abstract higher level networks initially determine which subnetwork a sample should be directed to, and lower level networks take on the task of finding discriminating features amongst similar classes. The proposed method is a hierarchy of scalable hierarchical networks. Each sub-network is called a mini-deep network, and mini-deep networks can recursively be split into subsequently smaller mini-deep networks. Outputs from these mini-deep networks feed a probabilistic classifier to predict a test sample's final class.

Confusion matrices infer class-wise linkage statistics by converting from similarity to dissimilarity matrices. Similarly, k-means and spectral clustering on low dimensional representations of the data offer clues to natural cluster boundaries at a coarser level. These statistics, form clusters and sub-clusters where each grouping contains classes with similar features. By viewing the resulting graph tree, such as a dendrogram graph, logical cluster boundaries can often be determined by manual inspection. This paper introduces data driven heuristics along with an iterative search algorithm to automatically detect these cluster boundaries. To ensure robustness and improved generalization, classes which are similar to multiple subgroups are encouraged to occur in multiple networks. Semantic outputs from the activated networks include softmax probabilities. The outputs of the networks, feed a final classification engine, which makes the final class decision.

The rest of the paper is organized as follows. The Background section overviews related work, followed by the Methods section which introduces the hierarchical deep network framework. The Results section presents experimental results and concluding remarks.

Background

The pioneering work of Hubel and Wiesel [5] laid the foundation for the modern hierarchical understanding of the ventral stream of the primate visual cortex. Simple receptive fields in the eye form complex cells in V1, then more abstract representations in V2 through V4, and finally into the inferior temporal (IT) cortex. The object representation in the IT cortex is amazingly robust to position, scale, occlusions, and background- the exact understanding of which still remains a mystery and marvel of the human brain [6].

Traditional computer vision techniques pair hand crafted low level features such as SIFT [7], SURF [8], or HOG [9] along with complimentary classifiers such as support vector machines (SVM) or neural networks. LeCun et al. [10] introduced convolutional neural networks (CNNs), computer vision oriented deep feed forward networks based upon a hierarchy of abstract layers. CNNs are end-to-end, learning the low level features and classifier simultaneously in a supervised fashion, giving substantial advantage over methods using independently solved vision features and classifiers.

Datasets such as Mnist [10], CalTech [11], and Pascal [12] have become more challenging over the years. The ImageNet [13] dataset has over 20,000 classes and 14M samples. In 2012, Krizhevsky and Hinton [14] beat the nearest competitor by 10% in the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) [15] competition with a seven layer CNN, taking advantage of a powerful regularization scheme called dropout [16].

Zeiler & Fergus [17] improved classification results by introducing random crops on training samples and improved parameter tuning methodologies. Simonyan and Zisserman [4] investigated the usage of network depth and C. Szegedy, et al. used banks of smaller convolutional filters [18] to simultaneously improve accuracy and lesson the number of parameters.

There are numerous works describing hierarchical decomposition of classification problems [19]. One of the earliest attempts of a CNN hierarchical approach [20] used transfer learning from sub-groups with many samples to sub-groups with few. Deng et al. [21] used a hierarchy of label relations, and further improvements were made by [22] and [23] using two and many categories respectively.

Confusion matrices can be used to determine hierarchical clusters [24, 25]. Podalak [26] increased robustness by allowing classes to fork in more than one hierarchal branch. Slakhutdinov et al. [27] combined structured hierarchical Bayesian models with deep learning to generate a framework that can learn new concepts with a minimal number of training samples.

CNN hierarchical improvements were demonstrated by [18, 28], and category hierarchy CNN based classifier was demonstrated in [23] that builds a two stage classifier to separate easy and difficult classes but the memory footprint and time constraints were a major challenge.

Methods

We propose a novel method to alleviate the computational complexity involved in training larger networks for datasets with higher number of discrete classes or concepts. Our approach uses a high-level classifier to initially determine which sub-class a sample belongs to, then passes that sample into the corresponding sub-class network to make a final class assignment. Our method automatically determines the optimal number of sub-classes, then trains each sub-class in an independent fashion. The first stage of determining the number of sub-classes is called Hierarchy Clustering. In this stage we exploit the rich information from the class to class confusion matrix (generated using a simplified conventional neural network mapping to all classes or concepts) to extract hidden correlations amongst classes. During training, a Hierarchy Classifier predicts which sub-network a sample belongs. This sample is then passed into one of *C* smaller Class Assignment Classifiers, each which is only concerned with a subset of classes to make a final classification estimate. The approach consists of three phases: 1.Hierarchical clustering, 2. Hierarchy classifier, and 3. Class assignment classifiers, which will be described next.

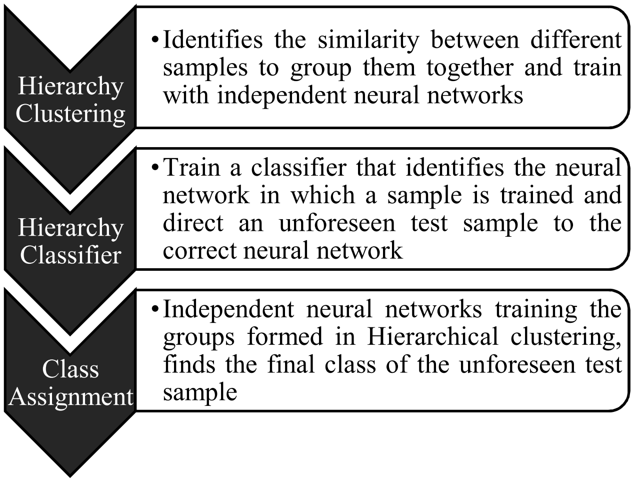

Hierarchy Clustering
•Identifies the similarity between different samples to group them together and train with independent neural networks
Hierarchy Classifier
•Train a classifier that identifies the neural network in which a sample is trained and direct an unforeseen test sample to the correct neural network
Class Assignment
•Independent neural networks training the groups formed in Hierarchical clustering, finds the final class of the unforeseen test sample

Figure 1. Flow of classification using hierarchal deep networks.

Hierarchy Clustering

To tackle problems with a large number of classes, we propose a hierarchical approach for clustering similar classes into sub-groups. This requires the training of a handful of much simpler neural networks where the number of overall parameters has been reduced. The intuition behind using a hierarchical clustering is the presence of coarse category or super classes which contain a higher number of finer classes. To categorize the given set of classes into super classes we have used spectral clustering of confusion matrix to generate a given number of clusters. The main challenge with a hierarchical clustering scheme is the selection of an optimum merge or split breakpoints, which if done improperly, can lead to low quality clusters. To address this challenge, we formulate a multi-phase technique that is based on the analysis of the confusion matrix of the classifier in the parent stage.

We use linkage statistics for getting the correlation indicators among classes in a hierarchical configuration. We define the distance matrix D, which is estimated from the confusion matrix C, and measures the dissimilarity among different classes. If a stage p has Kp clusters of classes, D has dimensions Kp × Kp, where an element Dp (Ci, Cj) represents the dissimilarity between cluster Ci and cluster Cj. An unweighted pair group method based on the arithmetic mean is used for determining the linkages between individual clusters. Dp (Ci, Ci) = 0 ∀ i ∈ K, represents the dissimilarity of a cluster with itself. We use a top-down divisive strategy to find non-overlapping classes that starts by including all classes in a single cluster. The parent cluster is subdivided into smaller class clusters until a termination criteria is met. The dissimilarity between clusters helps in dynamically determining the split points with an upper limit on the number of sub-clusters. As a result, this technique automatically adapts to the internal characteristics of the data.

Hierarchy class clustering:

Hierarchy relationships between classes are derived using the confusion matrix C_p that measures linkage distances d between classes. To form clusters with overlapping classes, we threshold class posterior probabilities *DCN* for classes originally not in cluster.

```
Input: Confusion matrix $C_p$ at classification stage $p$
Output: Overlapping class labels $Q$
Initialize: Upper limit on non-overlapping cluster size $\theta$ and overlapping factor $\gamma$
1) Compute distance matrix $D$ from $C_p$
2) Compute linkage statistics:
$d(r,s) = \frac{1}{n_r n_s} \sum_{i=1}^{n_r} \sum_{j=1}^{n_s} dist(x_{ri}, x_{sj}),$
Where $x_{ri}$ and $x_{sj}$ are dissimilar groups with $n_r$ and $n_s$ elements, respectively
3) Compute cumulative linkage values $Cum(d)$
4) for descending values $k$ in $Cum(d)$
        $\alpha$ = no. of classes with $d < k$
    if $\alpha > \theta$ then
        group classes $\alpha$ as new cluster $Q$
    repeat until all classes are assigned clusters
  end
5) Compute column normalized confusion matrix ($DCN$)
6) for each cluster $Q_i$
    if $DCN_p(C_i, C_j) \geq (\gamma . K_{p-1})^{-1} \forall C_i \notin Q_i, C_j \in Q_i$ then
            append class $j$ to cluster $Q_i$
  end
```

Figure 2. Hierarchy class clustering algorithm.

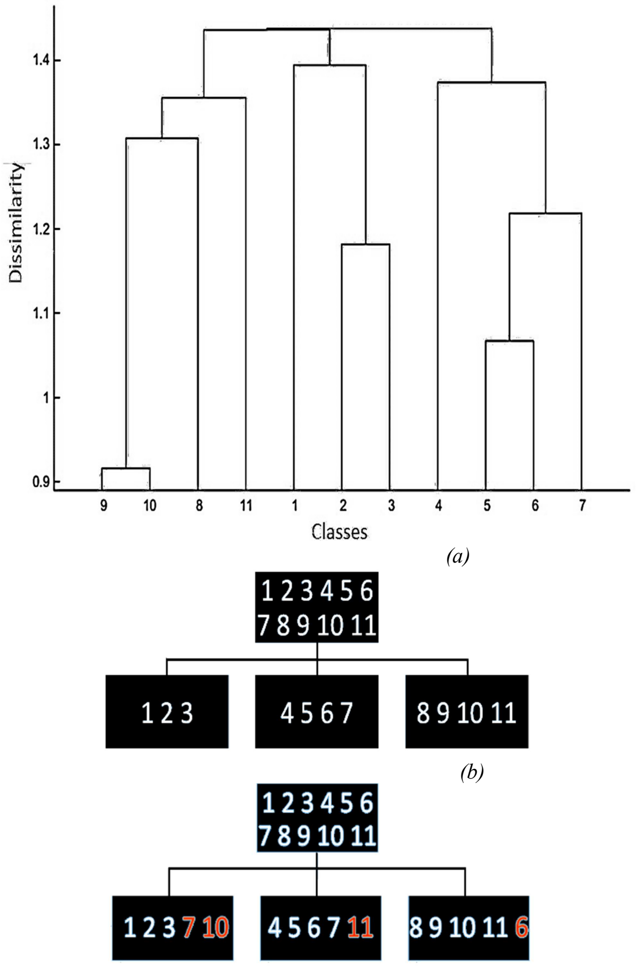
Small non-overlapping class groups are obtained by grouping similar classes together. However, in a non-overlapping setting, a sample that is misclassified at a parent level, has no chance of getting predicted correctly at the lower levels. Therefore, the small clusters are overlapped using the posterior probabilities to achieve higher generalization accuracy. The confusion matrix of the parent

©2016 Society for Imaging Science and Technology
DOI: 10.2352/ISSN.2470-1173.2016.19.COIMG-152

cluster are column normalized (DCNp) to obtain the class posterior probabilities. An element DCNp (Ci, Cj) represents the likelihood that a sample is of true class Ci given that it was predicted as class Cj. Let Qi be the collection of classes in cluster i, then the condition that certain classes are similar to this cluster can be given as,

$$DCN_p(C_i, C_j) \geq (\gamma . K_{p-1})^{-1} \; \forall \, C_i \notin Q_i, C_j \in Q_i \qquad (1)$$

We use a parametric threshold of $(\gamma.K_{p-1})^{-1}$, where γ is an overlapping hyper-parameter that determines the probability for including a class in cluster Q_i. The value of γ depends on the number of classes in the original problem and the number of clusters in the parent stage. The overlapping in the classifier allows a test sample to follow multiple paths of sub-classifiers. Figure 2 describes the pseudo code of the class clustering algorithm.



(a)

(b)

(c)

Figure 3. Illustration of Hierarchy clustering on Toy data having 11 classes. (a) Shows a dendrogram with dissimilarity among the classes. (b) Shows the 3 non-overlapping clusters formed with similar classes grouped together and (c) Shows the overlapping clusters.

Hierarchy Classifier

Let S be the training set with N classes, where C clusters are formed after Hierarchical clustering such that C clusters have $n_1, n_2 \dots n_c$ number of classes $\forall \; n_1 + n_2 + \cdots + n_c = N$. The classes associated with n_1 are labelled with class 1 and n_2 as class 2 and similarly $n_3 \dots n_c$ as class 3 … class c. In this way, the training set S is classified into C outputs instead of N classes. An unforeseen test sample when passed through the network shown in Figure 4 enters the hierarchy classifier. The hierarchy classifier directs the test sample in to one of the C networks. Once the test sample passes through a network in a class assignment classifier, the final class prediction is obtained.

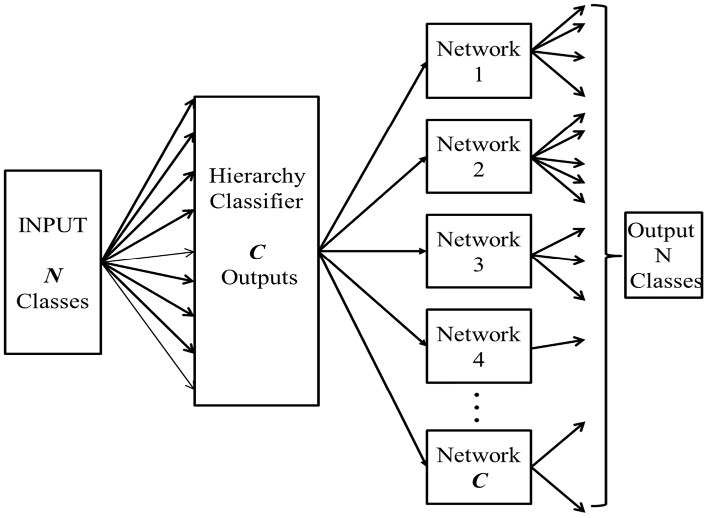


Figure 4: Illustration of Hierarchical deep network framework.

Class Assignment Classifier

The class assignment classifier consists of several simple neural networks (***C*** neural networks in Figure 4) predicting smaller number of classes at each neural network. The class assignment classifier outputs N classes, i.e., all classes of the dataset are classified at this stage of hierarchical deep network. In order to address misclassification at hierarchy classifier, overlapping clusters allow a test sample to be passed to more than one assignment classifiers. Let $p_1, p_2, \dots, p_c$ be the predictions of the hierarchy classifier for the corresponding $\boldsymbol{C}$ outputs and $q_1, q_2, \dots, q_{n_1}$ be the predictions of Network 1 for the corresponding n_1 outputs. When overlapped clusters are used, the top k predictions are used, each of which is a product of predictions from the hierarchy classifier and class assignment classifier. This product, referred to as confidence (ℂ), is the predicted final classification output:

$$Confidence(\mathbb{C}_i) = p_j * q_k$$
$$\forall \; \boldsymbol{i} \in \mathbf{1} \dots \boldsymbol{N}, \; \boldsymbol{j} \in \mathbf{1} \dots \mathbf{C}, \boldsymbol{k} \in \mathbf{1} \dots \; \boldsymbol{n_j} \qquad (2)$$

Results

Experiments are performed on CalTech101, CalTech256, and CIFAR100 datasets. This CalTech101dataset has 102 classes, with 40 to 800 images per class with image size of 300×200×3 pixels. The CalTech256 dataset has 257 classes, with 80 to 827 images per class with image size of 300×200×3 pixels. The CIFAR100 dataset has 100 classes with 500 images for training and 100 images for testing respectively per class and has an image size of 32×32×3 pixels.

Datasets were processed through multi-layer perceptron (MLP) as well as convolutional neural networks (CNNs). MLP processing used HOG input features. CNN processing used mean subtracted images resized to 64×64×3 for CalTech101 and

©2016 Society for Imaging Science and Technology
DOI: 10.2352/ISSN.2470-1173.2016.19.COIMG-152

CalTech256. The training and test splits are obtained using a 6-fold cross validation for all the datasets. The automatic clustering of large classification problems into a hierarchy of smaller classification networks offers a solution whereby the smaller networks have less parameters and are faster to train while offering increased classification accuracy. Current increase in accuracy is dependent on the number of mini-deep networks allowed, but can be upwards of 20%.

Table 1 demonstrates MLP processing on the CalTech101 dataset increases the final accuracy by approximately 16% using a non-overlapping hierarchical architecture. Similar observations were found with overlapping hierarchical architecture, but performance decreases with increasing overlap factor. This was attributed to the increase in confusion in the hierarchical stage. It should also be noted that the memory requirements increase as the overlap factor increases due to larger mini-networks.

Table 1. CalTech101 dataset- Top line indicates performance of a single large MLP neural network with two hidden layer of size [200 150]. MLP neural network used with two hidden layers of size [25 10] used in each mini-network for the rest of the lines. Hierarchical Clustering is controlled by varying the parameter Gamma (γ). HC indicates Hierarchy Classifier accuracy and FC indicates Final Classification accuracy.

C	Clustering Method	Gamma (γ)	HC (%)	FC (%)
1	NA	NA	NA	45.6
44	Non-overlap	NA	69.43	61.39
44	Overlap	3	69.05	61.56
44	Overlap	5	62.73	60.13
44	Overlap	8	52.05	58.61

In Table 2, when convolutional neural networks are used to evaluate the CalTech101 dataset, the final accuracy decreased by 4% using a non-overlapping hierarchical architecture. It is hypothesized the reason for this decline is due to 1) the identical architecture of all the mini-networks, and 2) when a cluster has fewer classes, the number of training samples for that network are also less, making them insufficient for CNNs. Both the CalTech datasets have significant variation in number of samples per class, but the results presented in this study were obtained on the entire dataset. The accuracies would be improved if the number of samples were identical across all classes.

Table 2. Caltech101 dataset- Top line indicates performance of a single large Convolutional neural network. Similarly, Convolutional neural networks are used in each mini-network.

C	Clustering Method	Gamma (γ)	HC (%)	FC (%)
1	NA	NA	NA	55.84
48	Non-overlap	NA	62.42	51.57
48	Overlap	3	50.33	50.72

In Table 3 & Table 5, it was observed that the performance is increased by 3% in case of both CalTech256 and CIFAR100 datasets when MLP neural network was used to evaluate the performance of the non-overlapping and overlapping hierarchical architectures. In Table 4, when convolutional neural networks are used to evaluate the performance of CIFAR100 dataset, final accuracy decreased due to the same reason as we have mentioned earlier for the CalTech101 dataset.

The dendrograms in Figures 5-7, represent the class grouping formed using the linkage statistics for different data sets. The colors in the graph depict groups of classes as determined by the algorithm described in Figure 2. While analyzing the groups, we observed that similar classes were grouped together which proves the efficacy of the hierarchical clustering.

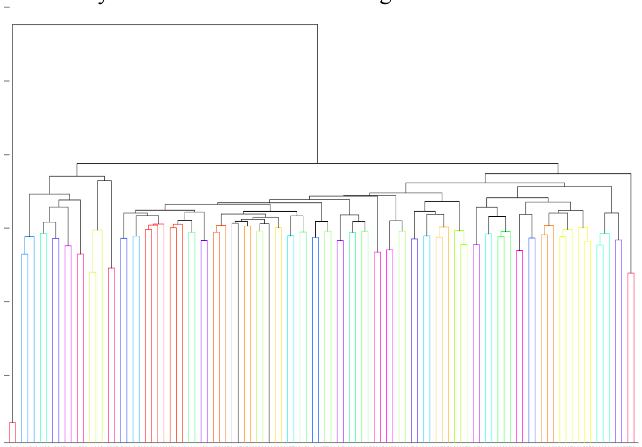


Figure 5. Example of a Dendrogram with 102 Classes of CalTech101 dataset generated using confusion matrix obtained from a single CNN (Better viewed in color).

Table 3. Caltech256 dataset- Top line indicates performance of a single large MLP neural network with two hidden layer of size [200 150]. MLP neural network used with two hidden layers of size [25 10] used in each mini-network.

C	Clustering Method	Gamma (γ)	HC (%)	FC (%)
1	NA	NA	NA	18.61
104	Non-overlap	NA	23.07	21.56
104	Overlap	3	24.49	21.96
104	Overlap	5	22.55	20.61

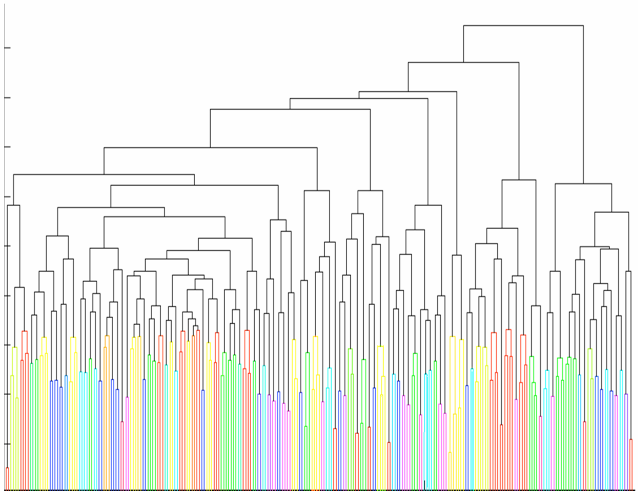


Figure 6. Example of a Dendrogram with 257 Classes of CalTech 256 dataset generated using confusion matrix obtained from a single MLP.

Table 4. CIFAR-100 dataset- Top line indicates performance of a single large Convolutional neural network. Similarly, Convolutional neural networks are used in each mini-network.

C	Clustering Method	Gamma (γ)	HC (%)	FC (%)
1	NA	NA	NA	29.45
39	Non-overlap	NA	46.91	23.09
39	Overlap	3	46.48	22.74

©2016 Society for Imaging Science and Technology
DOI: 10.2352/ISSN.2470-1173.2016.19.COIMG-152

Table 5. CIFAR100 dataset- Top line indicates performance of a single large MLP neural network with two hidden layer of size [200 150]. MLP neural network used with two hidden layers of size [25 10] used in each mini-network.

C	Clustering Method	Gamma (γ)	HC (%)	FC (%)
1	NA	NA	NA	22.83
36	Non-overlap	NA	28.37	24.96
36	Overlap	3	27.89	24.82
36	Overlap	5	26.34	23.25

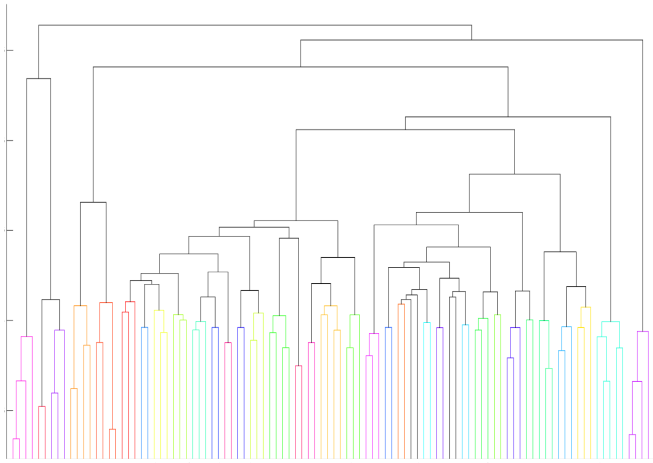


Figure 7. Example of a dendrogram with 100 Classes of CIFAR100 dataset generated using confusion matrix obtained from a single CNN.

In Table 6, it is observed that while using spectral density clustering the number of networks affect the final accuracy. An inappropriate choice of networks will lead to low quality clusters. It is observed that lower number networks improve hierarchy classifier accuracy but fail to improve final classifier accuracy. Choosing an optimal number of networks using dendrograms have already demonstrated the significant improvement in the performance.

Table 6: Spectral clustering on Caltech101 dataset- Top line indicates performance of a single large MLP neural network with two hidden layer of size [200 150]. MLP neural network used with two hidden layers of size [25 10] used in each mini-network for the rest of the lines. HC indicates Hierarchy Classifier accuracy and FC indicates Final Classification accuracy.

C	**HC (%)**	**FC (%)**
1	NA	45.6
5	80.3	35.2
10	70.1	50.2
15	65.2	51.0
17	63.2	53.7
20	62.9	49.0
25	59.9	50.8

Conclusion

An automatic hierarchical clustering method is introduced which reduces parameters while simultaneously increasing classification accuracy. This new approach borrows concepts from traditional divisive clustering techniques as well as confusion matrix dissimilarity linkage tree decomposition, to create an iterative method which methodically identifies cluster boundaries in a natural fashion. Hierarchical cluster boundary formation was tested on both MLP and CNN classifier frameworks, and shows significant benefit to the former, but not the latter. It is hypothesized that other classification frameworks such as SVM and Bayes classifiers can benefit from the hierarchical framework. Future work includes testing the hierarchy framework with larger data sets. What is most intriguing is that the proposed strategy allows for virtually unlimited number of classes in any particular classification problem.

References

[1] G. E. Hinton, S. Osindero, and T. Yee-Whye, "A fast learning algorithm for deep belief nets," *Neural Computation,* vol. 18, pp. 1527-54, 07/ 2006.

[2] D. Yamins, H. Hong, C. Cadieu, and J. J. Dicarlo, "Hierarchical modular optimization of convolutional networks achieves representations similar to Macaque IT and human ventral stream," in *27th Annual Conference on Neural Information Processing Systems,* Lake Tahoe, NV, 2013.

[3] A.-R. Mohamed, G. E. Dahl, and G. Hinton, "Acoustic modeling using deep belief networks," *IEEE Transactions on Audio, Speech and Language Processing,* vol. 20, pp. 14-22, 2012.

[4] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in *ICLR*, 2015.

[5] D. H. Hubel and T. N. Wiesel, "Receptive fields and functional architecture of monkey striate cortex," *The Journal of Physiology,* vol. 195, pp. 215-243, 1968.

[6] N. Kruger, *et al.*, "Deep Hierarchies in the Primate Visual Cortex: What Can We Learn for Computer Vision?," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 35, pp. 1847-71, 08/ 2013.

[7] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision,* vol. 60, pp. 91-110, 11/ 2004.

[8] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Computer Vision and Image Understanding,* vol. 110, pp. 346-59, 06/ 2008.

[9] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition,* San Diego, CA, 2005.

[10] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE,* vol. 86, pp. 2278-2323, 1998.

[11] F.-F. Li, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories," *Computer Vision and Image Understanding,* vol. 106, pp. 59-70, 04/ 2007.

[12] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *International Journal of Computer Vision,* vol. 88, pp. 303-338, 2010.

[13] D. Jia, D. Wei, R. Socher, L. Li-Jia, L. Kai, and F.-F. Li, "ImageNet: a large-scale hierarchical image database," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[14] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *26th Annual Conference on Neural Information Processing Systems*, Lake Tahoe, NV 2012.

[15] O. Russakovsky, *et al.*, "Imagenet large scale visual recognition challenge," *arXiv preprint arXiv:1409.0575,* 2014.

DOI: 10.2352/ISSN.2470-1173.2016.19.COIMG-152

[16] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research,* vol. 15, pp. 1929-1958, 2014.

[17] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *13th European Conference on Computer Vision,* Zurich, Switzerland, 2014.

[18] C. Szegedy*, et al.*, "Going deeper with convolutions," *arXiv preprint arXiv:1409.4842,* 2014.

[19] A.-M. Tousch, S. Herbin, and J.-Y. Audibert, "Semantic hierarchies for image annotation: A survey," *Pattern Recognition,* vol. 45, pp. 333-345, 2012.

[20] N. Srivastava and R. R. Salakhutdinov, "Discriminative transfer learning with tree-based priors," in *Advances in Neural Information Processing Systems*, 2013, pp. 2094-2102.

[21] J. Deng*, et al.*, "Large-scale object classification using label relation graphs," in *Computer Vision–ECCV 2014*, ed: Springer, 2014, pp. 48-64.

[22] T. Xiao, J. Zhang, K. Yang, Y. Peng, and Z. Zhang, "Error-Driven Incremental Learning in Deep Convolutional Neural Network for Large-Scale Image Classification," in *Proceedings of the ACM International Conference on Multimedia*, 2014, pp. 177-186.

[23] Z. Yan*, et al.*, "HD-CNN: Hierarchical Deep Convolutional Neural Network for Large Scale Visual Recognition," in *Computer Vision and Pattern Recognition*, Boston, MA, 2015.

[24] S. Godbole, "Exploiting confusion matrices for automatic generation of topic hierarchies and scaling up multi-way classifiers," *Annual Progress Report, Indian Institute of Technology–Bombay, India,* 2002.

[25] Y. Xiong, "Building text hierarchical structure by using confusion matrix," in *2012 5th International Conference on BioMedical Engineering and Informatics*, 2012.

[26] I. T. Podolak, "Hierarchical classifier with overlapping class groups," *Expert Systems with Applications,* vol. 34, pp. 673-682, 2008.

[27] R. Salakhutdinov, J. B. Tenenbaum, and A. Torralba, "Learning with hierarchical-deep models," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 35, pp. 1958-71, 08/ 2013.

[28] A. G. Howard, "Some Improvements on deep convolutional neural network based image classification," in *International Conference on Learning Representations*, 2014.