CEN: Classifier Ensemble Networks based on Joint Optimization of Hyperplanes

Jiman Kim
Samsung Research
jiman14.kim@samsung.com

Dongha Bahn
Samsung Research
dongha.bahn@samsung.com

Chanjong Park
Samsung Research
cj710.park@samsung.com

Abstract

Recent studies have demonstrated that the classification accuracy of single convolutional networks can be improved by incorporating model ensemble methods, which combine results from independently learned single networks. In this study, we propose a Classifier Ensemble Network (CEN), which substitutes model ensemble methods by embedding the information of a full hierarchy of classes into a single network. CEN's multiple classifiers, which are defined at various category levels, share feature layers and are simultaneously learned. The classifiers generate different semantic hyperplanes, and the joint optimization of hyperplanes alleviates overfitting on target classes. We evaluate our proposed architecture using the ImageNet dataset with a clear label hierarchy. We performed extensive experiments, specifically for various category levels, to evaluate the effectiveness of CEN architecture when compared to a baseline model and its ensemble models. The CEN obtains meaningful improvements to the top-1 accuracy. The proposed ensemble concept can be applied to most existing convolutional networks.

1 Introduction

Since convolutional neural networks (CNNs) were introduced in the image recognition domain, a notable number of network architectures were proposed, and these have significantly advanced performance levels by overcoming the training conundrum of deep architecture. Within the various CNNs, a particular main architecture stream recorded significant performance improvements on the ImageNet challenge (6; 27) and derived many structural variants: AlexNet (17), ZFNet (22), VGGNet (29), GoogleNet (32), InceptionNets (33; 31), ResNets (13; 36; 34), DenseNet (14), and so on. The single network architectures achieved human-level classification error rates by increasing the number of layers through convolution filter modification, layer connection change, various regularization techniques, and efficient learning algorithms. These methods focus on network architecture improvement. To overcome the performance of single networks, ensemble methods also have been studied. Ensemble methods combine the results of single networks by techniques such as bagging (3) and boosting (28). They simulate the decision process of a domain-specific expert group.

In this paper, we propose a single network architecture that has an ensemble effect. Partly inspired by the studies (35; 37; 1), which use the partial information of label hierarchy, we embed the information of a full hierarchy of class labels into deep network training. By the joint optimization of hyperplanes of different grouping levels using the proposed loss function, we develop a generic feature extractor which exhibits various grouping perspectives. This constitutes a major difference between CEN and the existing CNNs. The final accuracy of CEN surpasses the backbone single network and is

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

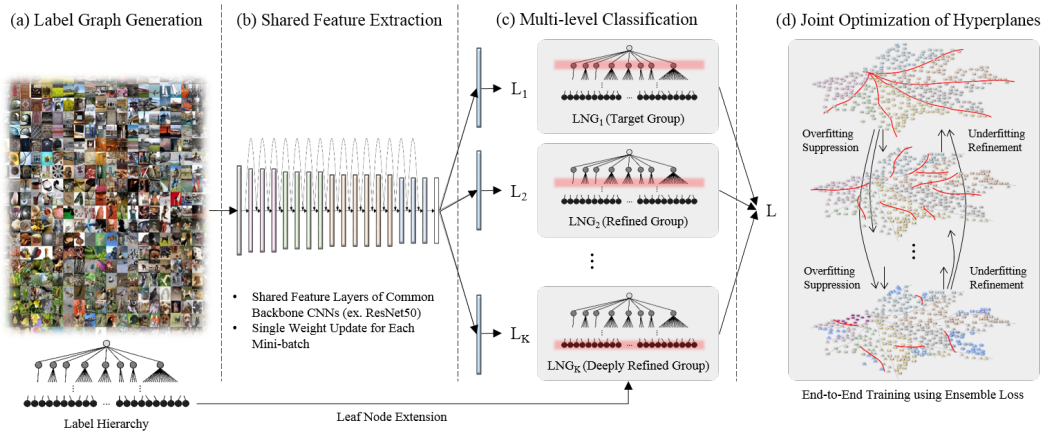

Figure 1: Pipeline of the proposed classifier ensemble network. From left to right: (a) ImageNet 1K dataset sample and a hierarchical tree graph of their semantic labels. (b, c) The proposed CEN architecture consisting of shared feature layers and ensembled classifier layers, where LNG_k means kth Leaf Node Group (red shaded region). (d) Joint Learning of multiple hyperplanes and interactions in the optimization process among different grouping levels.

comparable to model ensemble methods. This indicates that embedding various grouping perspectives in network training improves the network's overall performance. Most existing CNN networks can be used as backbone networks for feature extraction, and their performance can be improved by applying the proposed CEN architecture (Fig. 1).

2 Related Work

One of the core techniques to improve the accuracy of deep learning based visual object recognition is to efficiently modify the architecture of the deep neural networks. For efficient feature extraction and accurate classification of objects in images, CNN architecture has evolved in different ways.

Single CNN Architecture. After AlexNet (17) won in the ImageNet Large Scale Visual Recognition Competition (ILSVRC) 2012 overwhelmingly, many researchers, who use the ImageNet dataset and powerful GPUs, increased the number of layers, thereby solving vanishing gradients, exploding gradients, overfitting, and degradation problems; AlexNet (8 layers), VGGNet (19 layers), GoogleNet (22 layers), ResNet (152 layers), DenseNet (169 layers). Unlike increasing the depth of a layer, some attempts have been made to increase the width. GoogleNet proposed an inception module to the width as well as the depth. ResNet in ResNet (RiR) (34) generalized the previous ResNet architecture and wide ResNet (36) improved the image classification accuracy by expanding the number of filters of each layer. FractalNet (18) constructed a wide network through factorizing convolutions and aggressive dimension reductions. Meanwhile, there are studies that have achieved performance improvement utilizing the hierarchical structure of object categories. Before deep learning, a label hierarchy had been used to construct the taxonomy of linear classifiers (23; 12; 24; 2; 19; 9; 7; 16; 21). To improve the classification accuracy of deep neural networks (DNN), label tree based priors were used as DNN's parameters (30) or semantic relations between labels were encoded into the DNN's classifier (5). HD-CNN is a scalable network, which consists of a two-level category hierarchy and it conditionally executes fine category classifiers for ambiguous categories (35). For network embedding of label structure, a generalized triplet loss was proposed, which efficiently learn fine-grained feature representations (37). Recently, a hierarchy-aware CNN was proposed, where a maximum number of separable groups are selected from each of AlexNet's convolution layer and each layer is optimized using the loss function of a corresponding group (1).

Model Ensemble of Multiple CNNs To train more than two DNNs to have different representation properties, we can create different training sets, use different initial weights, or set the order of input images randomly for each DNN (3; 28; 8; 26; 20; 4). Lately deep learning studies commonly use an approach based on random ordering of input images or random initial weights to efficiently obtain adequate variances (15). The trained models are combined using bagging methods (3) to creat an

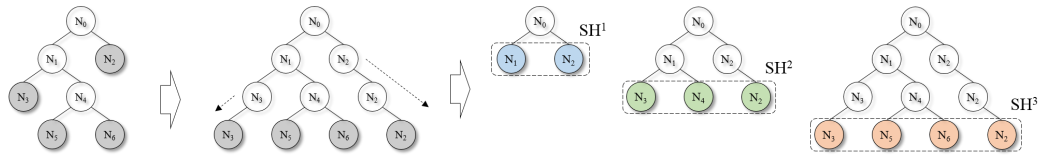

Figure 2: Process of leaf node extension and definition of semantic hyperplanes. (a) An example of original label graph before leaf node extension. Gray nodes mean leaf nodes of the hierarchy tree structure. (b) After leaf node extension, the leaf nodes are changed which are at the lowest level. (c) For each grouping level, multiple semantic hyperplanes are defined according to target object groups. They consider different kinds of features on the same feature space.

ensemble model. The goal of bagging methods based on parallel training is to develop a generic model, where overfitting phenomenon on specific training data is minimized. By contrast, boosting methods aim to develop a higher accurate model by sequential learning and minimize the number of falsely classified images. In this study, we compare a bagging method and our CEN architecture, focused on improving the general performance.

3 Semantic Label Graph

3.1 Label Hierarchy

In order to implement the proposed CEN architecture, a label hierarchy of target objects should be established. In this study, we include 1K classes on ImageNet (6; 27). The hierarchy was verified and the related sample images were downloaded from the website. For ease of verification the complete hierarchy of the 1K classes was manually arranged as a tree-graph. The order of classes was arranged to a certain degree taking into consideration the ImageNet's explorer page. The full hierarchy consists of twelve levels. In the tree-graph, deeper levels have more leaf nodes and the number nodes at the final level is 1K; 8 (first level), 48, 102, 252, 460, 677, 824, 935, 985, 997, 1,000, and 1,000 (twelfth level). Rapid increase in the number of leaf nodes indicates that the features trained on the two levels are distinct from each other. To generate different classifiers on each level of the label hierarchy graph, every leaf node should be extended to the final level. Consequently, the features of all 1K classes are identified on different grouping levels. In other words, all classifiers perform a coarse-to-fine classification of 1K classes on different grouping levels. The process of a leaf node extension is illustrated in Fig. 2. For example, the node N_2, which is a leaf node of level-1, is copied and connected up to level-3 using the same class name. The entire process of building of the extended graph is; a leaf node is inherited to a lower level using the same class name and the node is further divided if there are corresponding refined sub-classes.

3.2 Semantic Hyperplanes

To generate a separate classifier for each level of a label hierarchy graph, we consider the sub-graphs, and not the whole hierarchy graph with extended leaf nodes. Specifically, nodes which belong to lower parts are marked as unnecessary and removed in contrast to the target category group that are utilized to learn. As a result, after the leaf node extension (Fig. 2 (b)) of a full hierarchy graph (Fig. 2 (a)), semantic hyperplanes are defined as shown in Fig. 2 (c). Total 12 separated sub-graphs represent different target object groups, and this results in 12 different hyperplanes on the same feature space.

4 Classifier Ensemble Network

4.1 Shared Feature Layers

Most of the existing convolutional deep networks can be used for verification of our classifier ensemble concept. In this study, we use ResNet50 as a backbone network. All hidden layers of the backbone model, except the last layer which performs classification, are used as a common feature

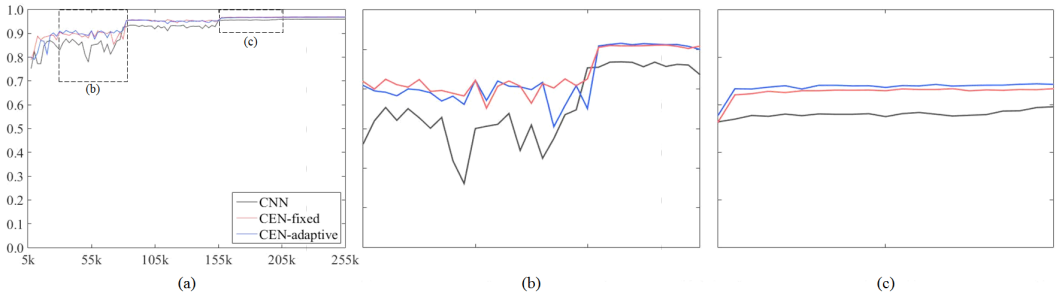

Figure 3: Validation results of the first grouping level (LNG^1) on the label hierarchy. (a) Top-1 classification accuracy over iterations. Ensembled classifiers using proposed losses reduce the overfitting before convergence (b) and achieved an improved accuracy (c).

extractor of multiple classifiers. In other words, a same feature space is shared on different semantic hyperplanes of the label hierarchy graph (Fig. 1 (b)). In our experiments, the feature layers consists of convolution and pooling layers with residual connections and the last feature layer is connected with each of the 12 classifiers. The weights of the feature layers are updated simultaneously with the weights of classifiers through the error backpropagation algorithm in a training process.

4.2 Ensembled Classifier Layer

To construct the CEN architecture, we appended a classifier layer to follow the feature layers. The classifier layer consists of 12 total parallel classifiers. The number of output nodes of each classifier is the same as the number of leaf nodes. Classifiers generate different hyperplanes, which segregate the categories of different grouping levels on the same feature space. This means each classifier will learn the sub-hierarchy's context (Fig. 2 (c)). Unlike the existing coarse-to-fine classification methods, the proposed CEN discriminates both of the refined sub-categories and other unrefined categories (N_2, N_3 in Fig. 2 (b)) at the same grouping level. This approach plays an important role in fine-tuning hyperplanes for refined categories preserving the hyperplane context of the upper levels. When classifiers of all levels are trained together, the training progress is more stable and fewer iterations are needed to achieve the same accuracy when compared to single classifier training (Fig. 3). In conclusion, optimizing multiple hyperplanes means that features of various grouping levels reflect on a network's training, and it prevents overfitting.

4.3 Joint Optimization of Hyperplanes

To simultaneously optimize multiple hyperplanes, a joint loss function is proposed. Joint loss is a combination of independent losses $L_i(k)$, $k = 1, \ldots, K$, where $L_i(k)$ represents the cross-entropy loss based on the softmax function of kth classifier, and K is the number of classifiers. The whole hierarchy of labels can be learned using the proposed joint losses; fixed loss and adaptive loss. Shared feature layers and ensembled classifier layer are updated based on the proposed loss functions. Regardless of the number of classifiers, only one update is performed for each mini-batch.

4.3.1 Fixed Ensemble Loss

Fixed loss function applies fixed weights to combine independent losses of all classifiers and it is defined as

$$L = \sum_{k=1}^{K} \lambda_k L_k, \tag{1}$$

where λ_k is a fixed weight of kth classifier. Fixed weights imply that we use the same weight values throughout the entire training process after initialization. It is further divided based on the process used to include the independent loss of each classifier into a joint loss, and is calculated as,

$$\lambda_k = \frac{1}{K} \quad \text{or} \quad \frac{1 - acc_k}{\sum_{k=1}^{K}(1 - acc_k)}, \tag{2}$$

where acc_k means the classification accuracy of the pretrained kth CNN. If fixed weights are used based on a constant K, then the training tendency of the CEN follows the average training tendency of all independently trained CNNs because the same value of each independent loss is reflected on the joint loss. The fixed uniform weight, $1/K$, scales the joint loss as a bounded value. If fixed weights, based on the classification accuracy, are used, then the independent loss of the CNN with lower discrimination ability has a higher impact on the joint loss. Each CNN's classifier has a different number of output nodes, and for a fair comparison the classification accuracy is measured on the smallest number of categories of d_1. Therefore, training the CEN tends to mitigate complex classification problems.

4.3.2 Adaptive Ensemble Loss

When the joint optimization of hyperplanes is performed, their discrimination ability changes over iterations. To exhibit the training tendency of the classifiers on the CEN's learning process, an adaptive loss function is proposed. The adaptive loss accepts adaptive weights and a regularization loss, in which an iteration index t is added as

$$L(t) = \left(\sum_{k=1}^{K} \lambda_k(t) L_k(t) \right) + \lambda_r L_r(t), \tag{3}$$

where $\lambda_k(t)$ represents a weight of kth classifier at iteration t and the other weight λ_r is the weight of a regularization loss $L_r(t)$. Adaptive weights imply that the classification accuracy of all classifiers for each iteration is measured and the weights are readjusted based on the accuracy values as follows, $\lambda_k \longmapsto \lambda_k(t), acc_k \longmapsto acc_k(t)$, where $acc_k(t)$ means the classification accuracy of kth classifier at iteration t. The weight $\lambda_k(t)$ of a classifier with low classification accuracy increases for each epoch, to comprehensively enhance its discrimination ability. Additionally, to prevent excessive overfitting of a specific classifier and reduce the variation of the overall performance distribution, a regularization loss was incorporated and $\lambda_r = 1$ was empirically set. The criterion function of Fisher's Linear Discriminant (FLD) (25) was utilized to define the regularization loss. Because the distribution of $L_k(t)$ is essentially required to be dense and separate from the original point, the regularization loss is inversely proportional to the FLD's criterion function. Thus, the regularization loss is defined as,

$$L_r(t) = \frac{1}{J(t)} = \left. \frac{{s_1}^2 + {s_2}^2}{|\mu_1 - \mu_2|^2} \right|_t, \tag{4}$$

where $J(t)$ represents a criterion function of FLD at iteration t, and it consists of pairs of between-class scatter (μ_1, μ_2) and within-class scatter (s_1, s_2). Between-class scatter indicates the mean of a distribution and within class scatter is an equivalent of the variance. We take into consideration two non-normal distributions, $L_k(t)$ and the original zero point. Because the mean of variance of the original point is zero, μ_2 and s_2 are zero. Hence, if the distribution $L_k(t)$ has a small variance and large distance from the original zero point, then the regularization loss decreases.

5 Experiments

We used ImageNet 2012 classification dataset (6; 27) consisting of 1K classes. The training set includes 1.28 million images and 50 thousand images are available for validation. However, at present, the dataset is in use at the Kaggle community as a combined task for detection, localization, and classification. As it is difficult to only measure the classification accuracy, we evaluated the proposed CEN architecture on validation data. By using fixed hyper-parameters on the complete training process, we utilized the validation set as the test set. Additionally, we solely considered a top-1 accuracy on 1 crop condition to observe the comprehensive improvement of a backbone deep network. For training, randomly sampled images are cropped to 224x224 randomly sampled, with the per-pixel mean subtracted (17). Furthermore, for validation, we center cropped each images to 224x224 and subtract the per-pixel mean. We trained each model for 100 epochs, including five epochs for warmup (11), and the learning rate starts from 0.128 and reduces by a divisor of 10 at epochs 30, 60, 80, 90. Moreover, we used SGD optimizer with momentum on a mini-batch size of 256, where a weight decay of 0.0001 and momentum of 0.9 were used. We used four Tesla P40 (24GB) for each training and on average it takes 90 hours for each model to train 100 epochs.

Method	Top-1 Accuracy (%) - 1 Crop										
	Level 1	Level 2	Level 3	Level 4	Level 5	Level 6	Level 7	Level 8	Level 9	Level 10	Level 11
ResNet50	95.94	90.08	85.77	83.20	81.82	79.71	78.54	77.43	76.73	76.57	76.70
CEN$_{i_top}$	96.75	90.86	85.96	82.69	81.00	79.04	78.10	76.96	76.46	76.61	76.74
CEN$_{i_bottom}$	96.90	91.86	87.68	84.18	81.78	79.32	78.25	77.02	76.49	76.62	76.73
CEN$_{f1_top}$	97.12	91.82	87.11	83.90	82.06	79.78	78.64	**77.53**	77.01	**76.72**	76.66
CEN$_{\mathbf{f1_bottom}}$	**97.17**	**92.42**	88.04	84.83	82.63	**80.17**	78.86	**77.53**	77.01	**76.72**	76.66
CEN$_{f2_top}$	96.99	91.63	86.95	84.41	82.01	79.70	78.75	77.46	**77.04**	76.60	76.64
CEN$_{f2_bottom}$	97.04	91.97	87.78	84.11	82.51	79.96	**78.89**	77.52	**77.04**	76.60	76.64
CEN$_{a1_top}$	96.76	91.28	86.87	83.79	82.02	79.81	78.68	77.38	76.98	76.71	76.62
CEN$_{a1_bottom}$	97.16	92.27	88.20	84.86	82.51	80.08	78.83	77.45	77.00	76.71	76.62
CEN$_{a2_top}$	93.63	88.54	84.00	82.17	81.25	79.57	78.50	77.44	76.85	76.54	**76.80**
CEN$_{\mathbf{a2_bottom}}$	96.94	92.23	**88.09**	**84.94**	**82.72**	80.12	78.88	**77.53**	76.88	76.54	**76.80**

Table 1: Classification results of each grouping level. Comparison networks; ResNet50 backbone network, independent loss i, fixed ensemble loss with uniform weights $f1$ and non-uniform loss $f2$, adaptive ensemble loss without the regularization loss $a1$ and without that $a2$.

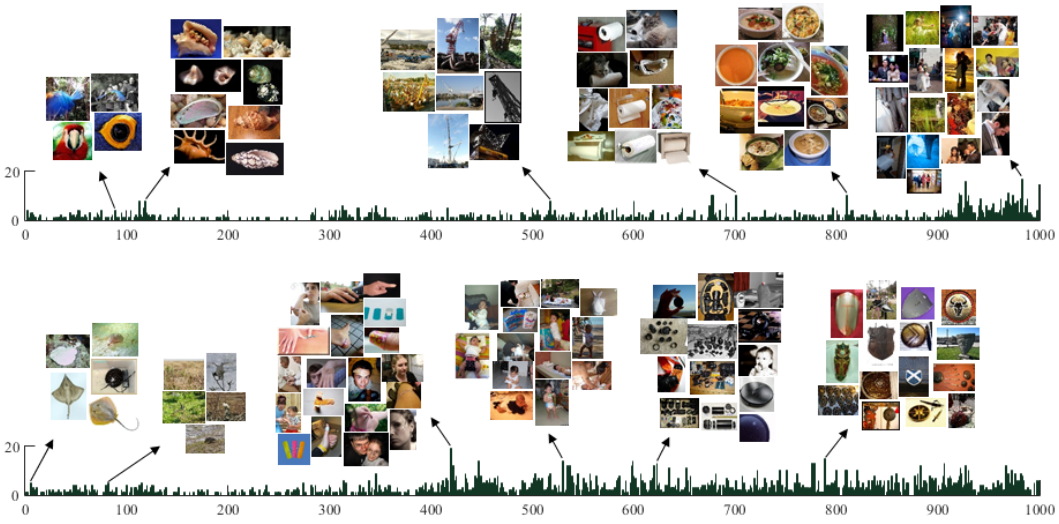

Figure 4: Examples of improved classification results on level 2 (top) and level 5 (bottom). Horizontal axis; 1K class index value which is defined at ImageNet dataset. Vertical axis; the number of images that are properly classified on the CEN architecture but falsely classified on the ResNet50 architecture.

5.1 Full Evaluation of Loss Functions

We performed a full test for all grouping levels of the label hierarchy graph, which is constructed from ImageNet (Table 1). The results of level 12 is the same as the one of level 11. In fact, evaluation on level 12 is meaningless because level 12 has no sub-hierarchy of labels. We measured the top-1 accuracy of the backbone network, ResNet50, and various versions of CEN architectures. Independent loss was proposed in (10), where each classifier has a cross-entropy loss function and, weight update for each mini-batch is performed based on the number of classifiers. The $a1$ is an adaptive ensemble loss when λ_r is zero. Subscript top represents the inference result of the classifier, with as many output nodes as the number of target category groups, and $bottom$ means the result of merging the inference of the classifier that has the largest number of output nodes among the learned classifiers collectively. Although the best network is different on each level, CEN$_{\text{f1_bottom}}$ and CEN$_{\text{a2_bottom}}$ demonstrate the best accuracy on average. The degree of accuracy improvement decreases when the level increases. As a result of the decrease in number of classifier or hyperplanes, the diversity of the grouping context information exhibited in network learning is reduced. Thus, to achieve significant improvement of accuracy it is essential to comprehensively refine and hierarchize the target category groups. In other words, if we subdivide the 1K classes of ImageNet and commensurately increase the number of ensembled classifiers, considerably improved top-1 accuracy can be obtained. Sample images that are misclassified on ResNet50 but properly classified on CEN$_{\text{f1_bottom}}$ are represented in Fig. 4. The images are grouped based on the index of 1K classes. We observed that the class with higher number of improved examples has various transforms and complex backgrounds. It

Level	Accuracy	Backbone Architecture					
		VGG19	VGG19$_{CEN}$	Res152	Res152$_{CEN}$	Den169	Den169$_{CEN}$
Level 2	Top1(%)	87.50	90.42	89.93	92.71	89.64	92.14
	Top5(%)	99.38	99.51	99.45	99.65	99.41	99.66
Level 5	Top1(%)	77.35	77.15	81.22	82.63	80.46	81.30
	Top5(%)	92.51	92.31	94.39	94.74	93.93	94.42

Table 2: Classification Results of various backbone networks.

Level	Loss	Top-1 Accuracy (%) on Different Training Configurations									
		~ 12	~ 11	~ 10	~ 9	~ 8	~ 7	~ 6	~ 5	~ 4	~ 3
Level 2	CEN$_{f1}$	**92.42**	92.38	92.31	92.33	92.21	92.17	92.11	91.81	91.30	90.88
	CEN$_{a2}$	**92.23**	92.20	92.18	92.20	92.21	92.19	92.05	91.99	91.42	90.97
Level 5	CEN$_{f1}$	82.63	**82.70**	82.44	82.54	82.53	82.49	82.26	-	-	-
	CEN$_{a2}$	**82.72**	82.68	82.65	82.66	82.58	82.62	82.40	-	-	-

Table 3: Classification results of various ensemble configurations at level 2 and 5. For instance, 'Classifier 2$\sim$12' means the hyperplanes from level 2 to level 12 are jointly optimized using ensemble losses.

indicates that we can accurately classify more images on variations of objects and backgrounds when the classifier ensemble architecture is applied on a backbone network. We also applied the CEN architecture to various backbone networks and compared the results to demonstrate the sufficient generalizability of our CEN concept (Table 2). We performed the experiments on two sample levels. The classification results of various CNNs were improved regardless of backbone network type.

5.2 Ablation Test on Various Configurations

The accuracy values of Section 5.1 are the results of the joint optimization of classifiers that can be utilized at each level. Out of all levels, we selected two levels and performed an ablation test, with constraints of the number of classifiers ensembled (Table 3). By means of this experiment, we analyze the performance distribution according to the number of classifiers and select the optimal number to be ensembled. In the ablation test of level 2 (48 categories), the classification accuracy tends to decrease when the number of classifiers is reduced. Shallow configurations, from 'Classifier 2$\sim$7' to 'Classifier 2$\sim$3', have a large degree of the accuracy degradation. This is because of the essential role performed by each hyperplane in a network training, when the total amount of the context information for feature discrimination is small. By contrast, deep configurations, from 'Classifier 2$\sim$12' to 'Classifier 2$\sim$8', already have a large amount of accumulated context information. Furthermore, because the categories are no longer not subdivided, the degree of the accuracy degradation is relatively small. This means that if our target categories are refined, for instance 1K classes, we can expect further accuracy improvement. In the ablation test of level 5 (460 categories), there is no obvious accuracy improvement because the number of jointly optimized classifiers is relatively small. As a result of the ablation tests, we identified that there are two important considerations for a notable improvement in classification accuracy; a comprehensive refinement of target categories and maximizing the number of classifiers jointly optimized through the hierarchizing of the refined subcategories. Validation results of various configurations are shown in Table 3. When configurations are separated into two groups, the deep ensemble configurations converge more stably than the shallow ensemble configurations over the entire training process . In conclusion, the higher the number of classifiers, the higher will be the final classification accuracy achieved with a relatively small variation. This is as a result of many classifiers interacting to make the proposed network's training stable through overfitting suppression and underfitting refinement between them.

Inference Configuration		Top-1 Accuracy (%)		Model Capacity and Efficiency		
		Level 2	Level 5	Parameters	FLOPS	Time
Backbone	ResNet50	90.08	81.82	23.61 M	3.87 G	15.11 ms
All Classifiers	ResNet50$_{avg}$	91.38	**83.95**	259.71 M	42.57 G	26.24 ms
	ResNet50$_{voting}$	91.30	83.64	259.71 M	42.57 G	15.23 ms
	CEN$_{f1_avg}$	**92.86**	82.80	24.59 M	3.87 G	29.27 ms
	CEN$_{f1_voting}$	92.46	82.57	24.59 M	3.87 G	18.26 ms
	CEN$_{a2_avg}$	92.49	82.59	24.59 M	3.87 G	29.27 ms
	CEN$_{a2_voting}$	92.18	82.40	24.59 M	3.87 G	18.26 ms
Top 3 Classifiers	ResNet50$_{avg}$	91.04	**83.39**	24.59 M	3.87 G	29.27 ms
	ResNet50$_{voting}$	90.82	82.88	24.59 M	3.87 G	18.26 ms
	CEN$_{f1_avg}$	**92.89**	82.83	24.59 M	3.87 G	29.27 ms
	CEN$_{f1_voting}$	92.50	82.56	24.59 M	3.87 G	18.26 ms
	CEN$_{a2_avg}$	92.53	82.58	24.59 M	3.87 G	29.27 ms
	CEN$_{a2_voting}$	92.12	82.40	24.59 M	3.87 G	18.26 ms

Table 4: Comparison of ensemble configurations for inference. All ensemble methods surpass the classification accuracy of a backbone model for both of level 2 and level 5. If a detailed label hierarchy is constructed, the CEN architectures achieve higher accuracy than ensembles of backbone models. Also, the CEN architectures have much less parameters and FLOPS than ensemble models of back a little more parameters than ensembles of backbone models.

5.3 Ensemble Effect on Single Networks

Analyzing the results of Section 5.1 and 5.2, it can be observed that the simultaneous optimization of multiple hyperplanes of various grouping levels causes a meaningful improvement to a single deep network's classification accuracy. Finally, we compared the proposed CENs with model ensemble methods. In general, single networks trained on different conditions are combined using model ensemble techniques such as averaging and voting to improve the classification accuracy. We evaluated if a CEN architecture equipped with end-to-end training could substitute model ensembles. For levels 2 (LNG_2) and 5 (LNG_5), results of a base model ResNet50 and CENs are represented in Table 4. We selected 'Classifier 2$\sim$12' and 'Classifier 5$\sim$12' configurations (Table 3) for CEN models. To compare model ensembles, we trained ResNet50 models with different random initial weights and same hyperparameters. The number of ResNet50 models matches the number of CEN's classifiers. After the training, we considered two approaches for an inference; using all classifiers and only the top three classifiers. Next, for each approach, we compared two ensemble techniques; output averaging and majority voting. From the results, the following three observations were made: (1) CENs can be used as substitutes because the accuracy difference between CENs and model ensembles is insignificant, regardless of the level. Notably, CENs show higher accuracy when a lower hierarchy adequately defined (level 2). (2) CENs are not significantly affected by the number of classifiers used in the inference. (3) Averaging is better than majority voting for ensemble technique. Additionally, we compared model size, complexity, and processing time. CENs have considerably fewer parameters and low model complexity, and the difference of processing time is within 3ms. Even when compared with the original single ResNet, there are no significant differences. In conclusion, the proposed CEN architecture can substitute model ensemble methods as well as improve the classification accuracy of single deep network models.

6 Conclusion

We proposed the CEN architecture to improve the classification accuracy of a large scale image dataset. Simultaneously training hyperplanes with diverse discrimination perspectives reduces overfitting and improves final accuracy. Moreover, by simply averaging the inference results of all classifiers, the CEN boosts the top-1 accuracy by up to 2.81% Considering the number of parameters, FLOPS, processing time, as well as the accuracy, the proposed CEN is significantly effective when compared to model ensembles that combine multiple convolutional neural networks.

References

[1] B. Alsallakh, A. Jourabloo, M. Ye, X. Liu, and L. Ren. Do convolutional neural networks learn class hierarchy? *IEEE Transactions on Visualization and Computer Graphics*, 24(1):152–162, 2018.

[2] S. Bengio, J. Weston, and D. Grangier. Label embedding trees for large multi-class tasks. *In NIPS*, 2010.

[3] L. Breiman. Bagging predictors. *Journal Machine Learning*, 24(2):123–140, 1996.

[4] T. Chen and C. Guestrin. Xgboost: A scalable tree boosting system. *In KDD*, 2016.

[5] J. Deng, N. Ding, Y. Jia, A. Frome, K. Murphy, S. Bengio, Y. Li, H. Neven, and H. Adam. Large-scale object classification using label relation graphs. *In ECCV*, 2014.

[6] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. *In CVPR*, 2009.

[7] J. Deng, S. Satheesh, A. C. Berg, and F. Li. Fast and balanced: Efficient label tree learning for large scale object recognition. *In NIPS*, 2011.

[8] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

[9] T. Gao and D. Koller. Discriminative learning of relaxed hierarchy for large-scale visual recognition. *In ICCV*, 2011.

[10] W. Ge and Y. Yu. Borrowing treasures from the wealthy: Deep transfer learning through selective joint fine-tuning. *In CVPR*, 2017.

[11] P. Goyal, P. Dollar, R. Girshick, P. Noordhuis, L. Wesolowski, A. Kyrola, A. Tulloch, Y. Jia, and K. He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv:1706.02677*, 2017.

[12] G. Griffin and P. Perona. Learning and using taxonomies for fast visual categorization. *In CVPR*, 2008.

[13] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *In CVPR*, 2016.

[14] G. Huang, Z. Liu, L. V. D. Maaten, and K. Q. Weinberger. Densely connected convolutional networks. *In CVPR*, 2017.

[15] H. Inoue. Fast and accurate inference with adaptive ensemble prediction for deep networks. *arXiv:1702.08259*, 2017.

[16] Y. Jia, J. T. Abbott, J. Austerweil, T. Griffiths, and T. Darrell. Visual concept learning: Combining machine vision and bayesian generalization on concept hierarchies. *In NIPS*, 2013.

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *In NIPS*, 2012.

[18] G. Larsson, M. Maire, and G. Shakhnarovich. Fractalnet: Ultra-deep neural networks without residuals. *In ICLR*, 2017.

[19] L. J. Li, C. Wang, Y. Lim, D. M. Blei, and L. Fei-Fei. Building and using a semantivisual image hierarchy. *In CVPR*, 2010.

[20] S. Z. Li, Z. Zhang, H. Y. Shum, and H. Zhang. Floatboost learning for classification. *In NIPS*, 2003.

[21] B. Liu, F. Sadeghi, M. Tappen, O. Shamir, and C. Liu. Probabilistic label trees for efficient large scale image classifica- tion. *In CVPR*, 2013.

[22] R. F. M. D. Zeiler. Visualizing and understanding convolutional networks. *arXiv:1311.2901*, 2013.

[23] M. Marszalek and C. Schmid. Semantic hierarchies for visual object recognition. *In CVPR*, 2007.

[24] M. Marszałek and C. Schmid. Constructing category hierarchies for visual recognition. *In ECCV*, 2008.

[25] A. M. MartoAnez01 and A. C. Kak. Pca versus lda. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(2):228–233, 2001.

[26] L. Mason, J. Baxter, P. Bartlett, and M. Frean. Boosting algorithms as gradient descent. *In NIPS*, 2000.

[27] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[28] R. E. Schapire and Y. Freund. Boosting: Foundations and algorithms. *MIT Press*.

[29] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *In ICLR*, 2015.

[30] N. Srivastava and R. Salakhutdinov. Discriminative transfer learning with tree-based priors. *In NIPS*, 2013.

[31] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. *In AAAI*, 2017.

[32] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *In CVPR*, 2015.

[33] C. Szegedy, V. Vanhoucke, S. Ioffe, and J. Shlens. Rethinking the inception architecture for computer vision. *In CVPR*, 2016.

[34] S. Targ, D. Almeida, and K. Lyman. Resnet in resnet: Generalizing residual architectures. *In ICLRW*, 2016.

[35] Z. Yan, H. Zhang, R. Piramuthu, V. Jagadeesh, D. DeCoste, W. Di, and Y. Yu. Hd-cnn: Hierarchical deep convolutional neural networks for large scale visual recognition. *In ICCV*, 2015.

[36] S. Zagoruyko and N. Komodakis. Wide residual networks. *In BMVC*, 2016.

[37] X. Zhang, F. Zhou, Y. Lin, and S. Zhang. Embedding label structures for fine-grained feature representation. *In CVPR*, 2016.