

Image Classification Using an Ensemble-Based Deep CNN



Aloysius Neena and M. Geetha

Abstract For the customary classification algorithms, performance depends on feature extraction methods. However, it is challenging to extract such unique features. With the advancement of Convolutional Neural Networks (CNN), which is the widely used Deep Learning Framework, there seems to be a substantial improvement in classification performance combined with implicit feature extraction process. But, training a CNN is an intensive process that often needs high computing machines (GPU) and may take hours or even days. This may confine its application in a few situations. Considering these factors, an ensemble architecture is modelled, that is trained on a subset of mutually exclusive classes, grouped by Hierarchical Agglomerative Clustering based on similarity. A new Probabilistic Ensemble-Based Classifier is designed for classifying an image. This new model is trained in comparatively lesser time with classification accuracy comparable to the traditional ensemble model. Also, GPUs are not necessary for training this model, even for large datasets.

Keywords Convolutional neural networks · Deep learning · Computer vision · Image classification

1 Introduction

Convolutional Neural Network is the widely used deep learning framework which was inspired by the visual cortex of animals [1]. Initially, it had been widely used for object recognition tasks but now it is being examined in other domains as well [2]. The neocognitron in 1980 [3] is considered as the predecessor of ConvNets. LeNet was the pioneering work in Convolutional Neural Networks by Jackel et al. [4]

A. Neena (✉) · M. Geetha

Department of Computer Science and Engineering, Amrita School of Engineering, Amritapuri, Amrita Vishwa Vidyapeetham, Amrita University, Coimbatore, India
e-mail: neenalloysius@gmail.com

M. Geetha

e-mail: geetham@am.amrita.edu

© Springer Nature Singapore Pte Ltd. 2018

P. K. Sa et al. (eds.), *Recent Findings in Intelligent Computing Techniques*,
Advances in Intelligent Systems and Computing 709,
https://doi.org/10.1007/978-981-10-8633-5_44

445

in 1990. It was specifically designed to classify handwritten digits and was successful in recognizing visual patterns directly from the input image without any preprocessing. But, due to lack of sufficient training data and computing power, this architecture failed to perform well in complex problems. Later in 2012, with the rise of GPU computing, Krizhevsky et al. [5] had come up with a CNN model that succeeded in drastically bringing down the error rate on ImageNet 2012 Large-Scale Visual Recognition Challenge (ILSVRC-2012) [6]. Over the years later, their work has become one of the most influential one in the field of computer vision and used by many for trying out variations in CNN architecture. But initially their results also daunted many in the area of computer vision due to the fact that the high-capacity classification of CNN is owed to huge labelled training dataset like ImageNet and it is obviously difficult in practice to have such large labelled datasets in different domains.

The aforementioned problem is addressed using Transfer Learning. The mid-level feature representations learned by a ConvNet on a large dataset are transferred to other object recognition tasks with limited training data. The main challenge while transferring knowledge is that it should produce positive learning in the target task. There is a high chance of negative transfer learning when the source and target tasks are less related. The datasets chosen as source is ImageNet and the target is Caltech101. Chances of negative transfer learning are less in our case since the source and target dataset are not totally unrelated.

The specific contributions of this work are as follows: we have trained a new ensemble model of convolutional neural network on Caltech101 and achieved the best results in terms of time complexity on this dataset, without compromise on accuracy. The subset of classes fed to each network of the ensemble (conveniently called pipeline) is grouped by Hierarchical Clustering algorithm based on single linkage to suit our need for grouping similar classes. The subsets chosen are mutually exclusive. Also, the concept of transfer learning is applied by retraining a trained AlexNet model which has significantly reduced the training time and also improved learning. Further to improve the learning process, visual saliency maps of all training images are generated to identify the salient portion of images and the network is trained using this. Much of the unnecessary background details are eliminated in the process.

2 Related Works

Even though Convolutional Neural Networks were introduced in 1990 by LeCun et al. [7], the architecture developed by Alex Krizhevsky et al. [5] is credited as the first work in CNN to popularize it in the field of computer vision. It has a total of 8 learned layers—5 convolution layers and 3 fully connected layers. The network was similar to LeNet but instead of alternating convolution layers and pooling layers, AlexNet had all the convolution layers stacked together. And compared to LeNet, this network is much bigger and deeper.

An improvement over AlexNet was the CNN architecture by Zeiler and Fergus [8]. They have presented a novel way to visualize the activity within the ConvNets using a multi-layered Deconvolutional Network (DeConvNet) [9]. A DeConvNet is also a ConvNet that operates in reverse direction, mapping features to input pixel space. So the visualization of a ConvNet is done by attaching a DeConvNet to each of its layers. This architecture can be used to observe the evolution of features during training and also to troubleshoot the network in case of any issues. They have used these tools to analyse the components of AlexNet and did a tweaking of the network by reducing the filter size and stride in first layer and expanded the size of the middle convolutional layers, resulting in an improved version over AlexNet.

Szegedy et al. [10] from Google, later in 2012 proposed an architecture called GoogleLeNet with a new module, Inception(v1), that gives more utilization of the computing resources in the network. GoogLeNet is a particular incarnation that has 22 layers of Inception module but with less parameters compared to AlexNet. This module has multiple convolution filters applied on the input image along with pooling and then combining the results. This leads to multi-level feature extraction from each input and also abstract features from different scales simultaneously.

Another famous architecture is VGGNet by K. Simonyan and A. Zisserman [11]. They have done a thorough analysis of the depth factor in a ConvNet, keeping all other parameters fixed. This try could have led to huge number of parameters in the network but it was efficiently controlled by using very small 3×3 convolution filters in all layers. This study has led to the development of a more accurate ConvNet architecture called, VGGNet.

Szegedy et al., in 2015 proposed an architecture [12], which is an improvement over GoogLeNet where the training of Inception modules (by Szegedy et al. [10]) are accelerated when trained with residual connections (introduced by He et al. [13]). The network has yielded state-of-the-art performance in the 2015 ILSVRC challenge and has won the contest.

A residual learning framework was presented by Kaiming et al. [13], where the layers learn residual functions with respect to the inputs received instead of learning unreferenced functions. The main drawback of this network is that it is much expensive to evaluate due to the huge number of parameters.

The work by Soman et al. [14] does grouping of misclassified characters together to improve accuracy which is in line with our work, but the performance is found to be dropping down when the number of classes exceeds the range of 40.

3 Proposed Ensemble Architecture

The proposed architecture can be viewed as a ConvNet which is replicated more than once (called as pipelines), each trained on a subset of class labels with different parameter settings. Here, subset of dataset refers to subset of classes or labels. This inherently means that the training subsets formed are mutually exclusive. The advantage of training on a subset of classes are analysed to be multifold, i.e. training

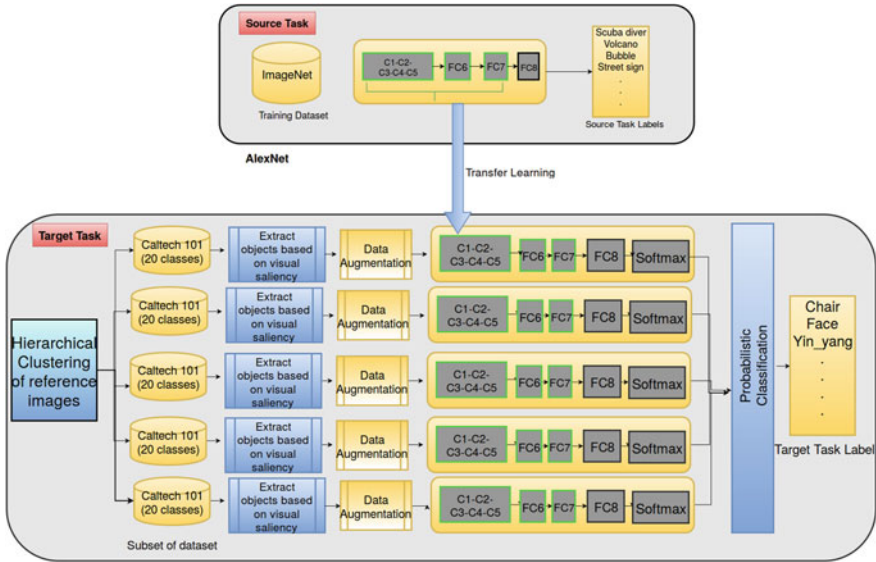


Fig. 1 Proposed Ensemble Architecture

time is expected to be reduced significantly; eliminate the need for high-end GPUs for the training of ConvNets on huge datasets.

Figure 1 clearly shows all the components involved, starting from the process of transfer learning whereby the new model gets initial weights from AlexNet trained on ImageNet. From the 101 classes, reference images are selected for each class and it is subjected to Hierarchical Agglomerative Clustering which results in group of similar images. Based on this grouping, mutually exclusive subsets are formed which is fed to network after preprocessing steps (training). The following sections detail all the steps involved.

3.1 Transfer Learning

The mid-level feature representations learned by AlexNet model on ImageNet are efficiently transferred for training the new network on Caltech101. Mid-level features or generalized features are captured in the first seven layers, i.e. from first conv layer to second fully connected layer (FC7). The learned weights of these layers are used in our model as well and these are kept constant and not updated during training. The final fully connected layer FC8 and classifier of source task are more specific to ImageNet hence we ignore them and add new FC8 and softmax classifier, which are retrained.

3.2 Clustering

Each pipeline in the new ensemble architecture is to be trained on images that belong to similar set of classes. And the grouping of similar classes is done by hierarchical clustering. Initially, reference images are selected for each class, the one with minimum noise. Based on the similarity matrix computed, Hierarchical Agglomerative Clustering (HAC) of the reference images is done. HAC follows a bottom-up approach, i.e. hierarchy of clusters are formed by recursively merging, starting from individual elements. Maximum similarity metric is considered for merging process, called as single-linkage clustering. Classes belonging to a cluster are considered for training a pipeline of the proposed ensemble, and thereby expecting a model that can be trained in lesser time without the need of GPUs, compared to the existing one.

3.3 Preprocessing Steps

The bottom-up method of computing saliency maps, Graph-Based Visual Saliency (GBVS), proposed by Koch et al. [15] is used to detect the objects (Fig. 2). The method is particularly useful when images have multiple objects and background. Based on the saliency maps, bounding box is drawn and objects are cropped from the original image, thereby removing much of the background information.

The customary procedure of random cropping is replaced by resizing the visual saliency-based detected object image to the standard input size required for AlexNet model. In addition to this, another data augmentation applied is horizontal flipping of the images. This is done based on the requirement that objects should be equally recognizable even if it is its mirror image. Applying more relevant transformations, the model is exposed to additional variations without the need of more labelled training images. Also, the problem of overfitting can be solved and thereby improving the model's ability to generalize.



Fig. 2 Starting from left, original image followed by the saliency map and original image with bounding box

3.4 Ensemble Training

Based on the results of hierarchical clustering, classes in each cluster is given as input to each pipeline of the ensemble model. ConvNets are usually trained on GPUs. But we have trained the new model without GPU, with parameter settings like 50 training data and 10 testing data, trained for a total of 15 epochs with batch size as 10 and 0.03 as learning rate. The activation function used is Rectified Linear Units (ReLU), i.e.

$$f(x) = \max(0, x) \quad (1)$$

If $f(x, y)$ is the input image and $w[s, t]$ is the filter, then the basic convolution operation is given by

$$g[x, y] = \sum_{s=-a}^a \sum_{t=-b}^b w[s, t] \cdot f[x + s, y + t] \quad (2)$$

Existing ensemble architecture, with equal number of pipelines, is also trained on full dataset by varying parameters to do a comparative study on performance. Top-1 and top-5 errors are computed in the process. Both the metrics decrease progressively over the training phase.

Algorithm 1 Proposed Ensemble

INPUT: Set of images img

OUTPUT: A trained network t

```

1: procedure PROPOSED-ENSEMBLE( $img$ )
2:   Initialize weights with that of pretrained AlexNet.
3:   Select reference image from each class.
4:   Compute similarity score matrix  $M$ .
5:   set of clusters  $C \leftarrow Hierarchical\_Clustering(M)$ 
6:   for each dataset  $i \in C(i)$  do
7:     Saliency_Extraction( $i$ );
8:      $t \leftarrow Train(i)$ .
9:   end for
10:  Return trained model  $t$ 
11: end procedure

```

3.5 Probabilistic Classifier

Testing of ensemble model involves feature computation and softmax classification (with scores) with each pipeline model (as given in Algorithm 2). The state-of-art ensemble networks does prediction by averaging the softmax classifier's score values. We have come up with a probabilistic classifier where we select the maximum score of softmax classifier from each pipeline and again a maxima of all the maximum scores. This is based on the presumption that given a test image, the pipeline

which has learned the features accurately will recognize it with a very high probability compared to other incorrect classification scores of other pipelines.

Algorithm 2 CNN Testing

INPUT: Any image from Caltech-101 or of similar data distribution

OUTPUT: Object label with predicted score

1: **procedure** CNN-TEST

2: Load the saved models of each pipeline.

3: Replace the last softmax loss layer with softmax classifier.

4: Each pipeline computes score for the given image using the same convolution and pooling operations done during training.

5: Find maximum of scores from each pipeline. Let it be $score(i)$, where i represents pipeline.

6: $final_score \leftarrow \max(score(i))$

7: Return associated label, $final_score$.

8: **end procedure**

4 Results and Performance Evaluation

The new ensemble model is trained on Caltech101 dataset using initialization weights from AlexNet trained on ImageNet.

4.1 Dataset

The source dataset for transfer learning of mid-level representations is chosen as ImageNet. The images have center-focused objects with less background clutter. The AlexNet model trained on ImageNet is chosen as the source task. The main advantage of selecting AlexNet as the source model over other models is that, since it is trained on the largest image database available, the mid-level representations learned will be more accurate and can be easily adapted to any other challenging datasets of different data distributions. The target dataset chosen for studying the impacts of transfer learning is Caltech101. It contains a total of 9,146 images distributed across 102 categories.

4.2 Testing

Testing is done on Caltech101 dataset by considering 8 classes, 25 classes, and full dataset. This incremental testing approach has ultimately proved useful in understanding the correlation between the number of classes, number of pipelines in the ensemble and classification accuracy.

Table 1 Caltech-101 classification accuracy for our ConvNet model trained on 8 classes, against the alternate approach

Models	Acc%	Train time
New ensemble	80	approx. 20 mins
Score-averaging ensemble	79	approx. 45 mins
Single-nonpipelined	78	approx. 20 mins

Table 2 Class-wise accuracy

Class	Acc% (new)	Acc% (existing)
Airplanes	70	60
Beaver	80	80
Car side	30	20
Dalmatian	100	100
Elephant	100	90
Helicopter	100	100
Kangaroo	90	100
Motorbikes	70	80

Table 3 Caltech-101 classification accuracy for our ConvNet model trained on 25 classes, against the alternate approach

Models	Acc%	Train time
New ensemble	84	approx. 40 mins
Score-averaging ensemble	83.66	approx. 1.5 h
Single-non-pipelined	83	approx. 40 mins

Case 1: 8 Classes and 2 Pipelines—We have trained an ensemble model of two pipelines, for a total of 8 classes, i.e. 4 classes per pipeline. Also, a score-averaging ensemble of comparable size (two pipelines), 8 classes per pipeline is also modelled. And the results are given in Tables 1 and 2.

Case 2: 25 Classes and 2 Pipelines—Next, the number of classes are increased and trained an ensemble model of two pipelines, for a total of 25 classes, 12 in one pipeline and 13 in the other. In this case as well a score-averaging ensemble of comparable size (two pipelines), 25 classes per pipeline is modelled. The test results are shown in Tables 3 and 4.

Case 3: 101 Classes and 5 Pipelines - Having seen the good results in above two scenarios, we have trained the ensemble on the whole Caltech-101 dataset, for 101 classes. Since we have more number of classes in this case, the ensemble is designed to have 5 pipelines with 20 classes per pipeline except one having 21 classes. The score averaging ensemble as well has 5 pipelines, each trained on full dataset.

Table 4 Class-wise accuracy

Class	Acc% (new)	Acc% (existing)
Airplanes	50	60
Beaver	80	80
Binocular	30	40
Bonsai	100	100
Brontosaurus	90	100
Camera	100	100
Cellphone	100	80
Chair	80	90
Dalmatian	100	70
Elephant	50	60
Ferry	100	100
Garfield	100	100
Gerenuk	100	90
Helicopter	90	90
Joshua tree	90	100
Kangaroo	100	90
Leopards	60	50
Llama	60	70
Okapi	80	90
Panda	100	90
Rhino	90	90
Stegosaurus	70	80
wheelchair	100	100
Wild cat	100	90
Windsor chair	80	70

Table 5 Caltech-101 classification accuracy for our ConvNet model, against the alternate approach

Models	Acc%	Train time
New ensemble	68	approx. 3
Score-averaging ensemble	78.48	approx. 15 h
Single-non-pipelined	78	approx. 3 h

Classification accuracies for the model as such as well as for per-class are detailed in Tables 5 and 6.

Figure 3 shows the top 3 classes with high classification accuracies and Fig. 4 shows top 3 classes with low classification accuracies, compared to the state-of-the-art model. Incorrectly classified are highlighted in red and those in green are correctly

Table 6 Class-wise accuracy

Class	Acc% (new)	Acc% (existing)
Airplanes	50	40
Beaver	70	80
Binocular	50	20
Bonsai	100	100
Brontosaurus	90	60
Camera	90	50
Cellphone	20	0
Chair	90	80
Dalmatian	70	60
Elephant	50	50
Ferry	100	100
Garfield	100	90
Gerenuk	90	80
Helicopter	90	100
Joshua tree	100	100
Kangaroo	80	60
Leopards	50	50
Llama	60	50
Okapi	80	80
Panda	90	90
Rhino	80	80
Stegosaurus	80	80
Wheelchair	90	100
Wild cat	80	50
Windsor chair	80	70

Fig. 3 Top 3 classes for which our method has performed well compared to alternate approach



Fig. 4 Classes for which our method has very low classification results compared to the alternate approach



Fig. 5 Sample predicted output



classified. Figure 5 shows a sample prediction, where the given image (airplanes) is predicted with the highest score of 1.

4.3 Evaluation

The result clearly shows improved accuracy with the new model when compared to the existing architecture, for less number of classes, with a reduction in training time. Thus, the new method is particularly useful when a new convolutional neural network is to be trained on large datasets like ImageNet where training complexity is a critical factor. However, performance drops as the number of training classes increases. Since we have increased the number of pipelines proportional to the number of classes, the probabilistic classification is severely impacted when maximal is chosen from the set of maximals. From an analysis of the score and class predictions it is found that in most of the cases, individual pipelines have correctly predicted the desired class. But with probabilistic classification we miss the desired result. This has led to a significant reduction in the classification accuracy of the new model. But the conventional requirement of high computing machines for training convnets on huge datasets can be eliminated with the proposed ensemble architecture.

5 Conclusion

Various aspects of CNN have been analysed, starting from transfer learning of feature representations from a pretrained model and the new model is actually found to be well adapted to the target dataset. With accuracies comparable to the existing model, we were able to bring about a decrease in the training time, thus reducing the time complexity of network. Our testing is limited to only one dataset in this work. We plan to have more rigorous testing of the model on challenging datasets like Caltech-256 and Pascal-VOC, in our future work.

References

1. Hubel, D.H., Wiesel, T.N.: Receptive fields and functional architecture of monkey striate cortex. *J. Physiol.* **195**(1), 215 (1968)
2. Nithin, D.K., Sivakumar, P.B.: Generic feature learning in computer vision. *Procedia Comput. Sci.* **58**, 202–209 (2015)
3. Fukushima, K.: Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biol. Cybern.* **36**(4), 193–202 (1980)
4. John, S., Henderson, D., Howard, R.E., Hubbard, W., Jackel, L.C., Boser, B., Lawrence, D.: Handwritten digit recognition with a back-propagation network. In: *Advances in Neural Information Processing Systems*. Citeseer (1990)
5. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: *Advances in Neural Information Processing Systems*, pp. 1097–1105 (2012)
6. Fei-Fei, L., Berg, A., Deng, J.: Large Scale Visual Recognition Challenge (2010)
7. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. *Proc. IEEE* **86**(11), 2278–2324 (1998)
8. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: *ECCV*, pp. 818–833. Springer (2014)
9. Matthew D Zeiler, Graham W Taylor, and Rob Fergus. Adaptive deconvolutional networks for mid and high level feature learning. In: *2011 ICCV*, pp. 2018–2025. IEEE (2011)
10. Szegedy, C., et al.: Going deeper with convolutions. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–9 (2015)
11. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition (2014). [arXiv:1409.1556](https://arxiv.org/abs/1409.1556)
12. Szegedy, C., Ioffe, S., Vanhoucke, V.: Inception-v4, inception-resnet and the impact of residual connections on learning (2016). [arXiv:1602.07261](https://arxiv.org/abs/1602.07261)
13. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition (2015). [arXiv:1512.03385](https://arxiv.org/abs/1512.03385)
14. Kumar, K., Sachin, S., Anil, R.M., Soman, K.P.: Convolutional neural networks for the recognition of malayalam characters. In: *FICTA 2014*, pp. 493–500. Springer (2015)
15. Harel, J., Koch, C., Perona, P.: Graph-based visual saliency. In: *Advances in Neural Information Processing Systems*, pp. 545–552 (2006)