Convolutional Neural Network Ensemble Fine-Tuning for Extended Transfer Learning

Oxana Korzh, Mikel Joaristi, and Edoardo Serra(✉)

Computer Science Department, Boise State University, Boise, USA
{oxanakorzh,mikeljoaristi}@u.boisestate.edu,
edoardoserra@boisestate.edu

Abstract. Nowadays, image classification is a core task for many high impact applications such as object recognition, self-driving cars, national security (border monitoring, assault detection), safety (fire detection, distracted driving), geo-monitoring (cloud, rock and crop-disease detection). Convolutional Neural Networks(CNNs) are effective for those applications. However, they need to be trained with a huge number of examples and a consequently huge training time. Unfortunately, when the training set is not big enough and when re-train the model several times is needed, a common approach is to adopt a transfer learning procedure. Transfer learning procedures use networks already pretrained in other context and extract features from them or retrain them with a small dataset related to the specific application (fine-tuning). We propose to fine-tuning an ensemble of models combined together from multiple pretrained CNNs (AlexNet, VGG19 and GoogleNet). We test our approach on three different benchmark datasets: Yahoo! Shopping Shoe Image Content, UC Merced Land Use Dataset, and Caltech-UCSD Birds-200-2011 Dataset. Each one represents a different application. Our suggested approach always improves accuracy over the state of the art solutions and accuracy obtained by the returning of a single CNN. In the best case, we moved from accuracy of 70.5% to 93.14%.

Keywords: Image classification · CNN · Deep learning
Transfer learning

1 Introduction

One of the most promising technologies in machine learning is the concept of transfer learning [31]. Currently, deep learning models require large scale data for training. With transfer learning revolution we can use a relatively small data set for training a deep learning model for a particular application while simultaneously keeping the same performance and reducing the execution time of the training procedure. This method is based on the assumption that current deep learning methods can train the model on a very large and general data set that

F. Y. L. Chin et al. (Eds.): BIGDATA 2018, LNCS 10968, pp. 110–123, 2018.
https://doi.org/10.1007/978-3-319-94301-5_9

includes patterns from different application areas. For a particular application, you do not need to retrain this large model from scratch. You can modify the existing model to be specialized for a particular application while still presenting general knowledge that came from the pretrained model.

There are two main options for using pretrained models for transfer learning [11]. The first one is fine-tuning the model: short-term additional training is applied to the original model to add a particular training set to the model's knowledge base. The second one is to use of pretrained Convolutional Neural Networks (CNN) as a feature extractor to transform images into feature vectors for classification.

For transfer learning for convolutional neural networks [14] it is very popular to use general pretrained networks such as AlexNet [14], GoogleNet [26] and VGG [25] for solving the image classification task for a particular application. In this paper, we propose a method of increasing image classification accuracy by using transfer learning of pretrained CNNs combined into an ensemble. We implement transfer learning using the fine-tuning method [11].

The main advantage of our method is fine-tuning of CNN ensemble when general features from different pretrained networks are shared and applied for a particular application. Our method includes the following steps: (1) each pretrained CNN is fine-tuned independently for a particular application; (2) weights and biases from fine-tuned networks are used for initialization of the ensemble model; (3) CNN ensemble model is fine-tuned for a particular application; (4) fine-tuned ensemble model can be used as a classifier by itself or as a feature extractor for an external classifier.

The contributions of this paper are:

1. A method of applying fine-tuning procedure to ensemble approach;
2. huge image classification accuracy improvement in three different benchmarks: Yahoo! Shopping Shoes Image Content (93.14% vs best known 70.5% [12]), UC Merced Land Use Dataset (99.76% vs best known 96.90% [11]) and The Caltech-UCSD Birds-200-2011 Dataset (81.91% vs best known 75% [4]);
3. investigation of four different CNN ensembles and summary of their main features with recommendations for improving classification accuracy in particular applications;
4. investigation of fine-tuned CNN ensemble in the task of feature extraction for image classification with external classifiers (SVM, ExtraTrees, Logistic regression).

The remainder of this paper is organized as follows. Section 2 describes related work in the field of neural network ensemble processing. In Sect. 3 we describe data sets used in experiments. In Sect. 4, we propose our approach of CNN ensemble fine-tuning. Section 5 includes details, experiments, result's analysis and discussion. Finally, we have conclusions for this paper with some remarks.

2 Related Work

The ensemble of classifiers is a well known technique to increase classification accuracy [27]. Different approaches for combining classifiers into ensembles already exist. In [19] the approach trains ensembles that directly construct diverse hypotheses using additional artificially-constructed training examples. In the paper [18] general questions of joint loss functions are discussed. In terms of the image classification task, ensemble approach is most commonly used for solving the multi classification task. For example in [16] multiple outputs are extracted as a learning problem over an ensemble of deep networks using a stochastic gradient descent based approach to minimize the loss with respect to an oracle. In [21] authors investigate the problem of pedestrian detection with an ensemble method using histograms of oriented gradients and local receptive fields, which are provided by a convolutional neural network and classified by multi layer perceptrons and support vector machines. The final choice is done by using majority vote and fuzzy integral.Pretrained convolutional neural network fine-tuning technique is successfully used in different applications. Recent research shows that pretraining on general data followed by application-specific fine-tuning yields significant performance improvement in the image classification task. In [10] authors analyze the performance of different fine-tuned CNNs for classification of paintings into art epochs. Paper [23] describes fine-tuning strategy to transfer recognition capabilities from general domains to the specific challenge of plant identification. Authors in [24] used fine-tuning process for CNN models pretrained on natural image dataset to solve medical image processing tasks. Many approaches are also base on synthetic data generation for improve the robustness of the classifier. [8,9] are two works specialized on synthetic data generation In our previous paper [13] we described a stacking approach for improving deep CNN transfer learning for processing low quality remote sensing images. CNN ensemble is used to produce a combination of features, extracted from different CNNs and to combine them in a feature vector for further classification with an external classifier. Paper [15] proposes an ensemble of fine-tuned convolutional neural networks for medical image classification. It describes a method for classifying the modality of medical images using an ensemble of different CNN architectures. The various CNNs in the ensemble allow extracting image features at different semantic levels, thereby enabling the characterization of the varying distinct and subtle differences among modalities. The ensemble of fine-tuned CNNs allows adapting the generic features learned from natural images to be more specific for different medical imaging modalities. In [4], when given a test image, authors use groups of detected keypoints to compute multiple warped image regions that are aligned with prototypical models. Each region is fed through a deep convolutional network, and features are extracted from multiple layers. Then features are concatenated and used as a feature vector for classification. One more paper that we want to mention is [6] where authors use Trunk-Branch Ensemble Convolutional Neural Networks (TBE-CNN) for video-based face recognition. TBE-CNN is composed of one trunk network that learns representations for holistic face images and two branch networks that

learn representations for image patches cropped around facial components. The output feature maps of the trunk network and branch networks are fused by concatenation and then last fully connected layer is applied for classification.

The main contribution of out method proposed in this paper in comparison with existing approaches is a fine-tuning procedure for pretrained model ensemble based on the joint loss function. Each ensemble member is a pretrained CNN(AlexNet, VGG19 or GoogleNet) that is prior independently fine-tuned for the specific application domain. Fine-tuned ensemble can be used as image classifier or as feature extractor for further image processing.

3 Data Sets Used in Experiments

We selected three different known benchmark data sets for testing our solution. Datasets belong to different application areas and contain images of different quality and resolution. The first data set is **Yahoo! Shopping Shoes Image Content** [2]. This data set provides a new benchmark for the problem of fine grained object recognition using shoes as an example and contains a diverse collection of types of shoe photos. This dataset contains 107 classes, each corresponding to a type and brand of shoe. Images are in RGB format stored in JPEG format, each of three channels contains 8bit information. Image resolution is 640×480 pixels. The total number of images is 5250. Examples from this data set are shown on the Fig. 1. Paper [12] describes an approach for classification this data set using two-flow model based on usage of pretrained deep neural network for feature extraction. Also, authors extract features directly from the data set using dimensionality reduction. Features from both sources are combined and used in nonlinear classifier to get the final result. In the experiment, 90% of the data is used as train and 10% as test. Achieved classification accuracy is 70.5%. In the paper [3] an approach is proposed for constructing mid-level visual features for image classification. The image is transformed using the outputs of a collection of binary classifiers. These binary classifiers are trained to differentiate pairs of object classes in an object hierarchy. Using this approach authors received 64.7% classification accuracy on random 90/10 split of the Yahoo data set.

The second benchmark used in this paper is well known landscape dataset **UC Merced Land Use Dataset(UCM)** [30]. The images were extracted from the USGS National Map Urban Area Imagery [1] collection for various urban areas around the country. Dataset contains 21 classes and 100 images per each class (2100 images in total). The resolution of this imagery is 1 foot per

Fig. 1. Yahoo! Shopping Shoes Image Content Dataset.

Fig. 2. UC Merced Land Use Dataset.

Fig. 3. The Caltech-UCSD Birds-200-2011 Dataset.

pixel. Each image is 256×256 pixels. Images are in TIF format and contain 8bit three channel (RGB) information. Examples from UCM dataset are shown on the Fig. 2. UCM data set is a widely used benchmark for testing landscape imagery processing methods. Paper [11] describes combination of feature extraction methods using standard image processing methods (such as BOW [17], IFK [22], LLC [29]) and convolutional neural networks. Then feature combination is used for final classification. Best achieved classification accuracy is 96.90%. In the paper [20] fine-tuning process is applied to CNN to achieve better classification accuracy. Fine-tuned CNN is used as a feature extractor for further classification using linear SVM [5]. Achieved accuracy in 5-fold cross-validation process is 96.47%. The most accurate model is based on using fine-tuned GoogleNet [26] as feature extractor in combination with linear SVM for final classification.

The third data set is called **The Caltech-UCSD Birds-200-2011 Dataset** [28]. The dataset contains 11,788 images of 200 bird species. Images are in JPEG format(8bits per channel, RGB). This benchmark data set is used for testing different image processing algorithm: bird species categorization, detection, and part localization. Examples from Birds dataset are shown in Fig. 3. In [7] authors propose a nonparametric approach for part detection which is based on transferring part annotations from related training images to an unseen test image. Feature extraction step is focused on those parts of images where discriminative features are likely to be located. This approach achieves 57.8% classification accuracy. Paper [4] proposed classification methods based on estimating of the object's pose. The features are computed by applying deep convolutional nets to image patches that are located and normalized by the pose. Authors used deep convolutional feature implementations and fine-tuning feature learning for fine-grained classification. Achieved classification accuracy rate is 75%.

All of the three data sets are challenging. Data has large inner class variability and the image resolution is small for traditional feature extraction and fine-grain classification methods. All data sets are widely used for testing convolutional networks approaches for image classification task.

4 Methodology

In this paper, we proposed a fine-tuning procedure for pretrained model ensemble based on the joint loss function. This approach combines the power of different pretrained networks and yields to image classification accuracy increasing. The proposed method includes four main steps. First, each pretrained CNN is fine-tuned independently for a particular application. We include three different networks to test our approach: AlexNet, GoogleNet and VGG19. The second step is ensemble model initialization using weights and biases from single fine-tuned networks. In the third step, CNN ensemble model is fine-tuned for a particular application using joint loss function. The fourth step is final image classification: proposed fine-tuned ensemble model can be used as a classifier by itself or as a feature extractor for an external classifier. For fine-tuning process in each pretrained CNN we replace the last fully connected layer with a new fully connected layer with the number of perceptrons equal to the number of classes in

the dataset. The new layer is randomly initialized. After that standard training procedure with law, learning rate is started. After fine-tuning each CNN produces a feature vector with the number of elements equal to the number of classes in output data set. This vector can be processed with some function (for example softmax) to obtain probabilities for test image to be in the appropriate class or this vector can be used as an input of external classifier to obtain test pattern class. In this paper, we investigate four different ensemble models. The first model is AVnet and it is shown in Fig. 4. This model is a combination of AlexNet and VGG19 net. To initialize this ensemble we use weights and biases form single pretrained networks up to fc7 layer. Then a new fully connected layer is added after concatenation. This layer is randomly initialized before starting fine-tuning process. Next model is called AGnet (Fig. 5) and is an ensemble of AlexNet and GoogleNet. In case if GoogleNet is participating in the ensemble we use its last fully connected layer for concatenation with fc7 layer from AlexNet or VGG19. Figure 6 shows VGnet that is a combination of VGG19 and GoogleNet. And finally, we combine all three networks into AVGnet (Fig. 7). The joint loss function is a cross entropy loss function, which is defined as follows:

$$L = -\sum_{j=1}^{C} y_j \log p_j$$

where C is the number of target classes, y_j is the $j-th$ value of the ground truth probability (0 or 1 in our case), p_j is the $j-th$ output value of softmax applied after joint fully connected layer of ensemble network. After ensemble model fine-tuning process is finalized the model can be used as a classifier by itself or features from different layers can extracted and used as an input of external classifier.

Fig. 4. AlexNet-VGG19 ensemble

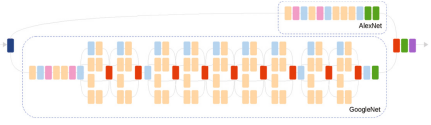

Fig. 5.

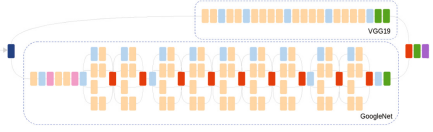

Fig. 6. VGG19-GoogleNet ensemble (VGnet).

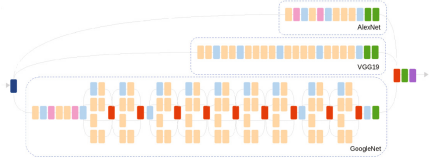

Fig. 7. AlexNet-VGG19-GoogleNet ensemble (AVGnet).

Table 1. Single network fine-tuning test classification accuracy

	Yahoo	UCM	Birds
AlexNet	76.56	96.90	65.14
VGG19	87.99	97.85	74.75
Google	83.05	97.85	77.68

Table 2. Ensemble network fine-tuning test classification accuracy

	Yahoo	UCM	Birbs
AVnet	90.45	98.81	79.71
AGnet	88.43	99.76	79.96
VGnet	89.66	99.76	79.88
AVGnet	90.35	99.76	80.56

Table 3. 10 cross fold validation test for CNN fine-tuning on Yahoo data set

	Alexnet	VGG	AVnet
Mean	73.92	84.82	87.83
ST Dev	2.69	1.90	1.92

5 Experiments

In our experiments we implement CNN ensemble using caffe [14]. We use three data sets mentioned in the part 3 for testing our methods: Yahoo! Shopping Shoes Image Content (Yahoo), UC Merced Land Use Dataset (UCM) and The Caltech-UCSD Birds-200-2011 Dataset (Birds). For each data set at first we fine-tune single networks (AlexNet, VGG19 and GoogleNet). Then weights from fine-tuned networks are used to initialize ensemble models for fine-tuning. We process fine-tuning of four ensembles for each data set. Then we compare the classification accuracy of fine-tuned networks and usage of fine-tuned networks as feature extractors in combination with external classifiers. In addition for Yahoo data set we provide the results of ten cross fold validation process for AlexNet, VGG19 and AVnet fine-tuning to estimate the stability of the model.

5.1 Yahoo! Shopping Shoes Image Content

In the Yahoo dataset the total number of images is 5250. We a make random 90/10 split to make our results comparable with experiments in [3,12].

Single Network Fine-Tuning. For single network fine-tuning we transfer images into lmdb database format for faster access. Also, we use fixed batch size in the fine-tuning process for all networks: 32 for train mode and 16 for the test. For AlexNet and GoogleNet we use 10000 iterations and for VGG19 - 20000 iterations. Learning rate starts from 0.0001 and decreases 10 times every 10000 steps. For the last, fully connected layer learning rate is ten times higher than for the other layers in the network. The best classification accuracy result with a single network for this data set was achieved for VGG19 network (87.99%). Even AlexNet shows 76.56% accuracy that is more than the best known accuracy achieved with non neural network fine-tuning methods. Fine-tuned GoogleNet classification accuracy is 83.05%. Classification accuracy for fine-tuned networks for all datasets is summarized in Table 1.

Ensemble Network Training. We are fine-tuning four ensembles: AVnet, AGnet, VGnet, AVGnet. Weight and bias initialization is computed from single fine-tuned networks using the layer by layer copy. Last fully connected layer is initialized randomly. We use batch size 40 in training mode for the first three

Table 4. Classification accuracy when fine-tuned CNN is used for feature extraction for further input to external classifier

Yahoo data set			
Feature extractor	SVM	ExtraTrees	LogReg
AlexNet fc6	74.09%	77.90%	77.33%
AlexNet fc7	75.23%	77.90%	76.19%
AlexNet fc8	76.95%	78.09%	77.52%
VGG19 cf6	88.76%	88.19%	89.14%
VGG19 cf7	88.95%	88.95%	89.71%
VGG19 cf8	89.14%	88.76%	89.90%
GoogleNet	84.95%	85.33%	85.33%
AlexNet + GoogleNet	85.14%	85.71%	86.28%
AlexNet + VGG19	87.42%	86.85%	87.80%
GoogleNet + VGG19	87.23%	88.57%	87.42%
AlexNet + GoogleNet + VGG19	86.28%	87.23%	87.42%
AVnet	90.66%	91.23%	90.85%
AGnet	89.90%	90.66%	91.42%
VGnet	90.47%	91.80%	92.00%
AVGnet	**92.00**%	**93.14**%	**92.57**%
UCM data set			
Feature extractor	SVM	ExtraTrees	LogReg
AlexNet fc6	96.90%	97.14%	97.62%
AlexNet fc7	96.90%	97.38%	97.85%
AlexNet fc8	97.14%	97.62%	98.09%
VGG19 cf6	97.85%	98.33%	98.33%
VGG19 cf7	98.33%	98.81%	98.57%
VGG19 cf8	97.85%	98.33%	98.33%
GoogleNet	98.33%	98.81%	98.81%
AlexNet + GoogleNet	99.52%	99.52%	99.52%
AlexNet + VGG19	99.28%	99.28%	99.28%
GoogleNet + VGG19	99.52%	99.52%	99.52%
AlexNet + GoogleNet + VGG19	99.05%	99.05%	99.05%
AVnet	99.52%	99.52%	99.28%
AGnet	**100.00**%	**100.00**%	**100.00**%
VGnet	99.76%	99.76%	99.76%
AVGnet	99.76%	99.76%	99.76%
Birds data set			
Feature extractor	SVM	ExtraTrees	LogReg
AlexNet fc6	67.14%	67.82%	67.99%
AlexNet fc7	68.67%	68.42%	68.25%
AlexNet fc8	67.99%	67.82%	67.91%
VGG19 cf6	76.40%	76.57%	76.91%
VGG19 cf7	78.18%	77.92%	78.09%
VGG19 cf8	77.75%	77.24%	77.07%
GoogleNet	78.26%	78.52%	78.35%
AlexNet + GoogleNet	78.69%	78.86%	78.94%
AlexNet + VGG19	78.13%	78.35%	78.69%
GoogleNet + VGG19	79.88%	80.05%	79.79%
AlexNet + GoogleNet + VGG19	79.28%	79.45%	79.45%
AVnet	80.64%	80.64%	80.73%
AGnet	80.98%	81.06%	80.98%
VGnet	81.23%	81.15%	80.89%
AVGnet	**81.57**%	**81.91**%	**81.74**%

networks and 32 for AVGnet. In test mode batch size is 16 for all networks. We use the same learning rate decreasing strategy as for single network fine-tuning: learning rate starts with 0.0001 and decreases every 10000 steps, for the last fully connected layer the value is 10 times bigger. Each network is trained with 30000 iterations. Best classification accuracy (90.45%) was achieved with AVnet - ensemble based on the combination of AlexNet and VGG19. Also, AVGnet ensemble has good classification accuracy (90.35%). Final classification accuracy for ensemble fine-tuning for all three data sets is shown in the Table 2.

Fine-Tuned CNN as a Feature Extractor. In this set of experiments, we use fine-tuned CNNs as feature extractors for further classification with an external classifier. For classification we use three classifiers: linear SVM (SVM), ExtraTrees classifier (ExtraTrees) and Logistic Regression (LogReg). Classifier implementation is based on sklearn library. For AlexNet and VGG19 we use feature vectors from three last fully connected layers: fc6, fc7 and fc8. In GoogleNet experiment we use features from last fully connected layer with ReLu and softmax transformations. In experiment AlexNet+GoogleNet we combine features from fc7 layer of AlexNet and last fully connected layer from GoogleNet in one feature vector. AlexNet+VGG19 combines fc7 from AlexNet and fc7 from VGG19. Similarly we do for GoogleNet + VGG19 and AlexNet + GoogleNet + VGG19 experiments. For AVnet, AGnet, VGnet and AVGnet we use a vector from concatenation layer for classification. Classification accuracy results are summarized in Table 4. In most of the experiments the best accuracy is achieved using ExtraTrees classifier. The best classification accuracy is 93.14% for AVGnet ensemble features in combination with ExtreTrees classifier. Also, it is interesting to compare classification results for the combination of feature vectors from different networks and features from ensemble model. Ensemble model gives in average 5% improvement in comparison with the combination of feature vectors (Table 3).

Ten Cross Fold Validation. We have selected three models for 10 folds cross validation process. It is a time consuming procedure because ten different models should be fine-tuned for each evaluation. We used stratified random 10 folder split for this experiment. AlexNet and VGG19 are fine-tuned independently for each of 10 splits. Then fine-tuned networks are used for initialization of appropriate AVnet. Ten cross-fold validation classification accuracy result for proposed ensemble model AVnet is 87.83% and the standard deviation is 1.92%. It means that model is stable and we are obtaining close results for any split. Figures 8, 9, 10, 11, 12 and 13 shows fine-tuning process test accuracy and test loss for AlexNet, VGG19 and AVnet fine-tuning.

5.2 UC Merced Land Use Dataset

Landscape imagery dataset UC Merced Land Use Dataset is the smallest one in our paper. It contains 2100 images for test and train split. We use 80/20 randomly stratified split where 80% of images go to training part and 20% are testing part (420 images for the test in total).

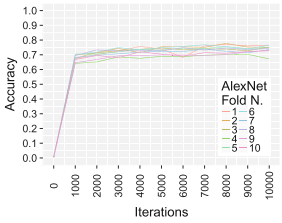

Fig. 8. Alexnet ten cross-fold validation fine-tuning test accuracy for Yahoo data set.

Fig. 9. VGG19 ten cross-fold validation fine-tuning test accuracy for Yahoo data set.

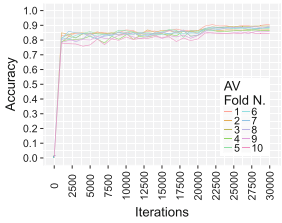

Fig. 10. AV net ten cross-fold validation fine-tuning test accuracy for Yahoo data set.

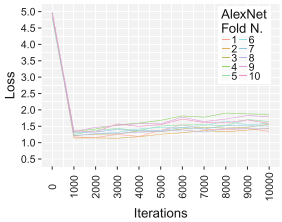

Fig. 11. Alexnet ten cross-fold validation fine-tuning test loss for Yahoo data set.

Fig. 12. VGG19 ten cross-fold validation fine-tuning test loss for Yahoo data set.

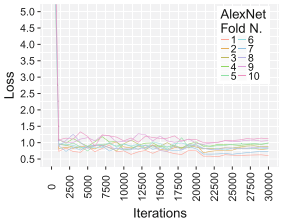

Fig. 13. AV net ten cross-fold validation fine-tuning test loss for Yahoo data set.

Single Network Fine-Tuning. In single CNN fine-tuning experiment we use the same methodology as declared in part 5.1 for Yahoo data set. Fine-tuned AlexNet shows 96.90% accuracy that is equal to best known result for this data set. Fine-tuned VGG19 and GoogleNet improve classification accuracy to 97.85%. Classification accuracies for fine-tuned networks are shown in Table 1.

Ensemble Network Training. Ensemble fine-tuning protocol for UCM data set is the same as for Yahoo data set. We fine-tune four ensembles using 40 batch size for train mode and 16 for test mode. As test and train part are not of a big size we use just 10000 iterations for fine-tuning without changing learning rate. After fine-tuning, most of the model shows 99.76% classification accuracy that means for this particular data set just one wrong classified image per 420 images in the test set.

Fine-Tuned CNN as a Feature Extractor. For this data set usage of external classifier after feature extraction is not reasonable because most of the model gives classification accuracy more than 99% without additional processing. The improvement is in half percent range. We achieve 100% classification accu-

racy for this data split using AGnet model in combination with ExtraTrees classifier. Classification accuracy results are summarized in Table 4.

5.3 Caltech-UCSD Birds-200-2011 Dataset

The Caltech-UCSD Birds-200-2011 Dataset is the most challenging of three datasets represented in this paper. Traditionally some kind of object detection method is applied for this kind of images to improve classification rate. But for us in this paper, the purpose was to show the benefits of ensemble model fine-tuning in comparison with single network fine-tuning. So we use image resampling instead of object detection. Images are downsampled to 256 pixels in the smallest dimension.

Single Network Fine-Tuning. Dataset is randomly split into 90/10 ratio of train and test part. Test set contains 1178 images. The fine-tuning protocol is the same as in part 5.1 for Yahoo data set. For single network fine-tuning the best classification accuracy rate (77.68%) is achieved using GoogleNet. This result is 7% more than known classification accuracy rate for this data set.

Ensemble Network Training. Best classification result with ensemble fine-tuning is achieved after fine-tuning AVGnet and is equal to 80.50%. During the fine-tuning process, we use learning rate decreasing strategy similar to the one used for Yahoo data set. Maximum iteration number is 30000 for all four models.

Fine-Tuned CNN as a Feature Extractor. Classification accuracy results are summarized in Table 4. We have in average 3% improvement for ensemble training in comparison with classification of combinations of feature vectors. Best image classification accuracy rate (81.91%) is shown by AVGnet model in combination with ExtraTrees classifier. We summarize best classification accuracy results in Fig. 14. Base is the best known classification accuracy result for each data set (70% for Yahoo, 96.90% for UCM, 75% for Birds). Single CNN is the best accuracy achieved with single network fine-tuning (87.99%, 97.85%, 77.68% accordingly). Ensemble shows the best result of ensemble CNN fine-tuning (90.45%, 99.76%, 80.56% accordingly). The best results in classification are achieved using fine-tuned ensemble for feature extraction and classification using ExtraTrees classifier (93.14%, 100%, 81.91% accordingly).

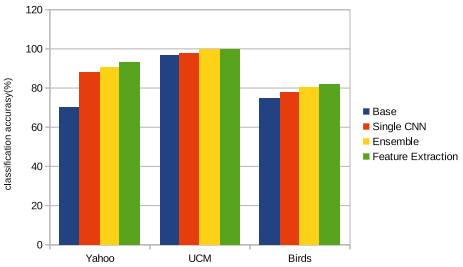

Fig. 14. Best classification accuracy results comparision

6 Conclusions

In this paper, we proposed a transfer learning approach based on the returning of CNN ensemble models combining multiple already pre-trained convolutional neural networks. We tested this approach on three benchmark datasets: Yahoo! Shopping Shoe Image Content, UC Merced Land Use Dataset and The Caltech-UCSD Birds-200-2011 Dataset. We observed that in all of the experiments our approach is able to classify better than the method present in literature and better than the standard transfer learning approach that fine-tunes only a single network at a time. In addition, we show that in terms of accuracy our approach works even better if it is used as a feature extractor. We obtained a maximum accuracy among all the ensemble models of 93.14% for Yahoo! Shopping Shoe Image Content, 100% for UC Merced Land Use Dataset and 81.91% for The Caltech-UCSD Birds-200-2011 Dataset. Our approach always improves the accuracy of all the state of art with 23% of classification accuracy improvement in the best case.

References

1. Usgs national map urban area imagery: 2017. https://nationalmap.gov/ortho.html. Accessed 14 Nov 2017
2. Yahoo webscope datasets: 2016. http://webscope.sandbox.yahoo.com/. Accessed 14 Nov 2017
3. Albaradei, S., Wang, Y., Cao, L., Li, L.-J.: Learning mid-level features from object hierarchy for image classification. In: Proceedings of 2014 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 235–240, November 2014
4. Branson, S., Van Horn, G., Belongie, S.J., Perona, P.: Bird species categorization using pose normalized deep convolutional nets. CoRR, abs/1406.2952 (2014)
5. Cortes, C., Vapnik, V.: Support-vector networks. Mach. Learn. **20**(3), 273–297 (1995)
6. Ding, C., Tao, D.: Trunk-branch ensemble convolutional neural networks for video-based face recognition. CoRR, abs/1607.05427 (2016)
7. Göring, C., Rodner, E., Freytag, A., Denzler, J.: Nonparametric part transfer for fine-grained recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2489–2496, June 2014
8. Guzzo, A., Moccia, L., Saccà, D., Serra, E.: Solving inverse frequent itemset mining with infrequency constraints via large-scale linear programs. ACM Trans. Knowl. Discov. Data (TKDD) **7**(4), 18 (2013)
9. Guzzo, A. Saccà, D., Serra, E.: An effective approach to inverse frequent set mining. In: Ninth IEEE International Conference on Data Mining, ICDM 2009, pp. 806–811. IEEE (2009)
10. Hentschel, C., Wiradarma, T.P., Sack, H.: Fine tuning cnns with scarce training data - adapting imagenet to art epoch classification. In: 2016 IEEE International Conference on Image Processing (ICIP), pp. 2381–8549, September 2016
11. Fan, H., Xia, G.-S., Jingwen, H., Zhang, L.: Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery. Remote Sens. **7**(11), 14680–14707 (2015)

12. Ilyukovich-Strakovskaya, A., Dral, A., Dral, E.: Using pre-trained models for fine-grained image classification in fashion field. In: Proceedings of the First International Workshop on Fashion and KDD, KDD 2016, August 2016
13. Korzh, O., Cook, G., Andersen, T., Serra, E.: Stacking approach for cnn transfer learning ensemble for remote sensing imager. In: Proceedings of Intelligent Systems Conference 2017, September 2017
14. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Proceedings of the NIPS 2012: Neural Information Processing Systems (2012)
15. Kumar, A., Kim, J., Lyndon, D., Fulham, M., Feng, D.: An ensemble of fine-tuned convolutional neural networks for medical image classification. IEEE J. Biomed. Health Inform. **21**(1), 31–40 (2017)
16. Lee, S., Purushwalkam, S., Cogswell, M., Ranjan, V., Crandall, D.J., Batra, D.: Stochastic multiple choice learning for training diverse deep ensembles. CoRR, abs/1606.07839 (2016)
17. Li, F.-F., Perona, P.: A bayesian hierarchical model for learning natural scene categories. In: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), pp. 524–531, June 2005
18. Liu, Y., Yao, X.: Ensemble learning via negative correlation. Neural Netw. **12**(10), 1399–1404 (1999)
19. Melville, P., Mooney, R.J.: Creating diversity in ensembles using artificial data. Inf. Fus. **6**(1), 99–111 (2005)
20. Nogueira, K., Penatti, O.A.B., dos Santos, J.A.: Towards better exploiting convolutional neural networks for remote sensing scene classification. In CoRR, volume abs/1602.01517 (2016)
21. Oliveira, L., Nunes, U., Peixoto, P.: On exploration of classifier ensemble synergism in pedestrian detection. IEEE Trans. Intell. Transp. Syst. **11**(1), 16–27 (2010)
22. Perronnin, F., Sánchez, J., Mensink, T.: Improving the fisher kernel for large-scale image classification. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part IV. LNCS, vol. 6314, pp. 143–156. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15561-1_11
23. Reyes, A.K., Caicedo, J.C., Camargo, J.E.: Fine-tuning deep convolutional networks for plant recognition. In: Working Notes of CLEF 2015, September 2015
24. Shin, H., Roth, H., Chen, M., Lu, L., Xu, Z., Nogues, I., Yao, J., Mollura, D.: Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning. IEEE Trans. Med. Imaging **35**, 1285–1298 (2016)
25. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. CoRR, abs/1409.1556 (2014)
26. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. volume abs/1409.4842 (2014)
27. Tumer, K., Ghosh, J.: Error correlation and error reduction in ensemble classifiers. Connect. Sci. **8**(3–4), 385–404 (1996)
28. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The caltech-ucsd birds-200-2011 dataset. California Institute of Technology, (CNS TR 2011 001) (2011)
29. Wang, J., Yang, J., Yu, K., Lv, F., Huang, T., Gong, Y.: Locality-constrained linear coding for image classification. In: Proceedings of the 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3360–3367, June 2010

30. Yang, Y., Newsam, S.: Bag-of-visual-words and spatial extensions for land-use classification. In: Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems, pp. 270–279, March 2010
31. Yosinski, J., Clune, J., Bengio, Y., Lipson, H.: How transferable are features in deep neural networks. In: Proceedings of Neural Information Processing Systems (NIPS 2014), November 2014