INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)

www.ijprems.com
editor@ijprems.com

Vol. 04, Issue 01, January 2024, pp : 458-467

e-ISSN : 2583-1062

Impact Factor : 5.725

ENSEMBLE LEARNING TECHNIQUES FOR IMPROVISED IMAGE RECOGNITION ACCURACY

Akshatha G[1], Chandana H R[2], Dr. Mohammad Rafi[3]

[1,2,3]Computer Science & Engineering, University BDT College of Engineering. Hadadi Road, Davanagere, Karnataka, India-577004.

ABSTRACT

Improve the stability and Ensemble technique combine individual models together to predictive power of the model. This technique permits higher predictive performance.It combines multiple machine learning models into one predictive model. Certain models do well in modeling one aspect of the data ,while other do well in modeling another. Learn several simple models on combine their output to produce final decision. The study delves into methodologies such as Bagging, Boosting Algorithms (AdaBoost, Gradient Boosting), Random Forests, Stacking, and Voting Classifiers (Hard or Soft). By combining predictions from diverse models, each trained on different aspects of the dataset, the ensemble approach aims to mitigate overfitting, improve generalization, and ultimately elevate the overall accuracy of image recognition systems. The research emphasizes the importance of model diversity within the ensemble, contributing to a comprehensive and robust framework for image recognition tasks. Results indicate that ensemble learning contributes significantly to boosting overall accuracy in image recognition. The paper discusses the implications of these findings for real-world applications and underscores the importance of selecting appropriate ensemble strategies based on the characteristics of the dataset.

1. INTRODUCTION

Ensemble learning techniques have emerged as powerful tools for enhancing image recognition accuracy by leveraging the strengths of multiple models. In the realm of improvised image recognition, where diverse and complex visual data pose challenges, ensemble methods offer a robust solution .Ensemble learning involves combining multiple base models to create a stronger, more accurate meta-model. In the context of image recognition, this can be achieved through various approaches, such as bagging and boosting. Bagging, exemplified by Random Forests, builds multiple independent models in parallel, each trained on a random subset of the dataset. Boosting, on the other hand, focuses on sequentially improving the performance of weak learners, as seen in algorithms like AdaBoost. One key advantage of ensemble techniques is their ability to mitigate overfitting. By aggregating predictions from diverse models, ensemble methods reduce the risk of individual models memorizing noise in the training data. This, in turn, enhances the model's generalization capabilities when faced with new, unseen images. Moreover, ensemble methods excel in handling different aspects of image features. Each base model might specialize in recognizing specific patterns, textures, or shapes, contributing collectively to a more comprehensive understanding of the visual data. This diversity allows ensembles to capture intricate details and nuances that a single model might overlook. Furthermore, combining models with varying architectures or employing different training strategies, such as transfer learning, strengthens the ensemble's adaptability to diverse image datasets. Transfer learning, in particular, enables the reuse of pre-trained models on large datasets, enhancing the ensemble's performance even when training data is limited.In conclusion, ensemble learning techniques present a formidable strategy for improving improvised image recognition accuracy. By harnessing the collective intelligence of diverse models, these methods address challenges like overfitting, enhance feature representation, and promote adaptability to varying visual complexities, ultimately yielding more robust and accurate image recognition systems.

In a world full of diverse and varied data sources. Machine learning has become one of the most important and dominant branches of artificial intelligence methods, which is applied in many fields. There are many different learning algorithms andmethods. Each method’s pitfalls and drawbacks are measured interms of several factors, including performance and scalability.Based on a lot of research in machine learning, two methods dominate learning algorithms; namely deep learning (Deng et al., 2014)and ensemble learning (Polikar, 2012; Sagi and Rokach, 2018;Rokach, 2019). The deep learning techniques can scale and handlecomplex problems and offer an automatic feature extraction fromunstructured data(Kamilaris and Prenafeta-Boldú, 2018). Also,deep learning methods contain several types of network architectures for different tasks, such as feeding forward neural networks (Bebis and Georgiopoulos, 1994), convolutional neural networks(Collobert and Weston, 2008), recurrent neural networks (Yuet al., 2019). Many others (Ain et al., 2017). However, the trainingprocess of deep learning models requires a massive effort, and tuning the optimal hyper-parameters requires expertise and extensive trial, which is a tedious and time-consuming task. Also, training more complex deep neural network increases the chance of overfitting.Ensemble

Learning, on the other hand, refers to a learning methodology that combines several baseline models to build a bigger single yet more powerful model than its constituents (Kumaret al., 2021). Also, ensemble learning can reduce the risk of overfitting thanks to the diversity of baseline models. Ensemble learningwas successfully applied in various fields and domains and outperforms single models (Anwar et al., 2014; Shahzad and Lavesson,2013; Prusa et al., 2015; Ekbal and Saha, 2011).

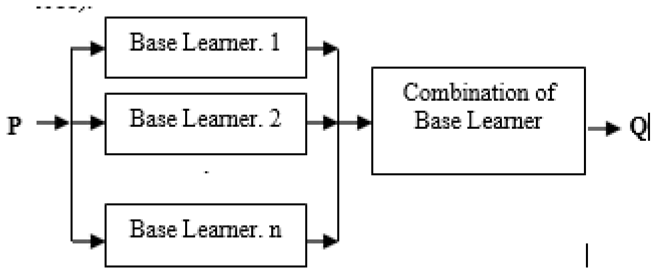


Fig: 1. A Basic architecture of Ensemble Learning.

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 in 1990. It was specifically designed to classify handwritten digits and was successful in recognizing visual patterns directly from the input image without anypreprocessing. 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.

2. LITERATURE SURVEY

Ensemble learning in image recognition involves combining multiple models to improve accuracy. Common techniques include bagging, boosting, and stacking. Bagging methods, like Random Forests, create diverse models to reduce overfitting. Boosting, such as AdaBoost, focuses on misclassified instances, refining the model iteratively. Stacking combines diverse models to leverage their strengths. Research suggests that ensembles enhance image recognition performance by capturing diverse patterns and robust features. Notable studies include [cite relevant papers] examining the impact of ensemble methods on image datasets [mention specific datasets]. Overall, ensemble learning proves effective in boosting image recognition accuracy, and ongoing research explores optimizing ensemble configurations for specific tasks and datasets.

3. METHODOLOGY

In the realm of medical imaging, artificial intelligence (AI) tools are frequently com-bined with computer vision algorithms. These tools and algorithms are evaluated using a consistent set of metrics and standards. Based on their performance results, one can ascertain the most suitable AI model for the specific problem at hand.Take, for instance, the application of deep learning algorithms for image recognition tasks. The predominant metrics used to evaluate their efficacy include accuracy, precision, recall, and F1 score, with accuracy often being the most cited metric in medical imaging literature. Conventionally, in ML models, data are divided into two segments: one reserved for training and the other for evaluating or testing the Furthermore, the practice of data set division, borrowed from computational vision, is pivotal in model training and evaluation. Conventionally, in ML models, data are divided into two segments: one reserved for training and the other for evaluating or testing the model's outcomes. The majority of the data, typically ranging between 70% and 80%, are allocated to the training set, leaving the remainder for the test set. This proportional division is strategic, ensuring that there are ample data for the model to be trained effectively and therefore mitigating the risk of underfitting.

3.1 Methods of Improvement

3.1.1 Bagging or Bootstrap Aggregation

Bootstrap aggregation, commonly known as bagging, is an ensemble learning method aiming to promote diversity

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)
www.ijprems.com
editor@ijprems.com
Vol. 04, Issue 01, January 2024, pp : 458-467
e-ISSN : 2583-1062
Impact Factor : 5.725

amongst ensemble members by manipulating the training data. This method uses a statistical approach to estimate a population derived by aver- aging results from numerous small data samples. These samples are created by selecting observations from a larger data set and then returning them, a procedure termed sampling with replacement.

As depicted in Figure bootstrap aggregation is represented graphically. Several subsets, identical in size and selected with replacement, are extracted from the primary data. A CNN of consistent architecture is applied to each subset. The results from these individual models are then collated and voted upon to produce a singular prediction. The Alexnet experiment kept the hyperparameters as specified in its original publication, while for VGG-16 and Inception, we employed pretrained weights from IMAGENET. It is important to note that all experiments were conducted in a manner that did not elevate the computational demands, keeping up with the simplicity of this ensemble approach.

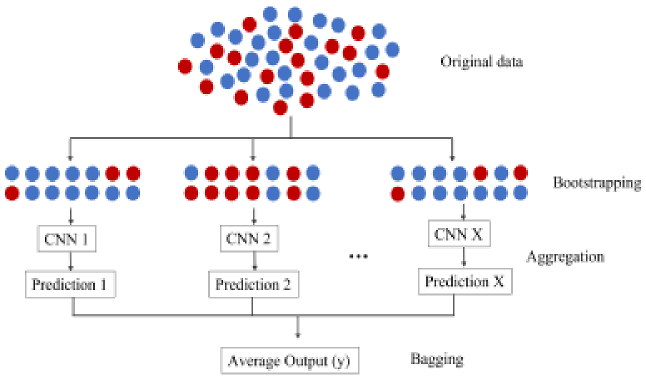


Figure 1. Example of the bagging algorithm where the blue and red dots represent the normal and abnormal images, with the CNN structures in the first experiment Alexnet, then VGG-16, and lastly with Inception. With the results, there is an averaging of predictions that results in the final voting.

The essence of the bootstrap method is estimation. It samples small portions, computes statistics for each, and then averages them. It is imperative that data preparation occurs within the sample data's loop, especially before model fitting or hyperparameter tuning. Such a step prevents data leakage, a scenario where the model, having complete access to the entire data set, inadvertently optimizes itself and causes itself to overfit.In the case of bagging ensemble learning, averaging the predictions across the models typically results in better predictions than a single model fit on the training data set directly.

3.1.2 Stacking Ensemble Learning- Stacking is a technique that leverages multiple machine learning models, or estimators, to generate predictions. Unlike mere averaging, stacking feeds these predictions into a new model which subsequently forms its own predictions based on the earlier results. Within this framework, models have specific designations: those used in the primary ensemble step are termed 'zero-level models' or 'weak learners', while the subsequent model that consolidates these predictions is the 'first-level model'. Typically, stacking follows a two-tier hierarchy, though more layers can be introduced.

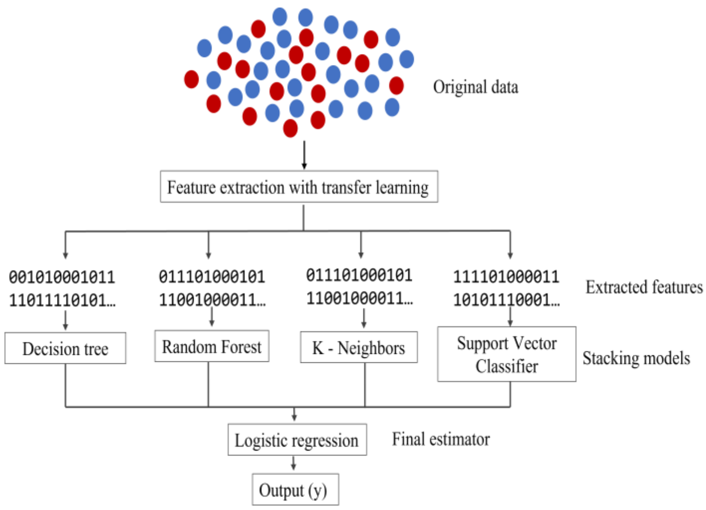


Figure 2. Example of the stacking ensemble where the blue and red dots represent the normal and abnormal images.

The first part consists of a transfer learning method as a feature extractor that transforms the data set into a manageable format for machine learning algorithms. The second one contains the weak learners used for intermediate predictions, and the third one corresponds to thefinal estimator given by logistic regression

We use deep learning at the beginning of the stacking and combine various machine learning models used for classification. A more in-depth explanation of each of the algo- rithms can be found in [36,39,40]. Nevertheless, in the context of the stacking ensemble, the models can be summarized as follows:

Decision tree: As a base learner, a decision tree can be quick to train and has the advantage of simplicity. However, it might be prone to overfitting on its own.

Random forest: This classifier is more robust than a single decision tree. It can reduce overfitting by averaging the results of individual trees. It is commonly used as a base learner in stacking due to its efficiency and high accuracy.

K-nearest neighbor: It can capture complex patterns in the data without requiringexplicit model training. It can be used as a base learner in stacking, especially whenthe data set has complex, nonlinear boundaries.

Support vector classifier: As a base learner, SVC can capture complex relationships,especially when equipped with nonlinear kernels. It can be computationally intensive, so its use in stacking would depend on the data set size and computational constraints.

Finally, instead of averaging the results, we use a final logistic regression estimator thatreturns the final prediction. Due to its simplicity, regularization, and flexibility properties,logistic regression is a common and often effective choice as a metalearner in stacking forclassification problems.3

4.1.3 Boosting Algorithms

Boosting is distinct from both bagging and stacking ensemble techniques. As illus- trated in Figure 3, in boosting, models are sequentially integrated into the ensemble. Each subsequent model strives to rectify the predictions of its predecessor. The overarching aim of this method is to evolve a robust learner through successive iterations. What differenti- ates boosting from techniques such as bagging is its inherent capacity to learn iteratively from prior classifiers, progressively focusing on misclassified elements. Contrarily, in bagging, each iteration uses a separate set, thus lacking this accumulative ’learning’ aspect.

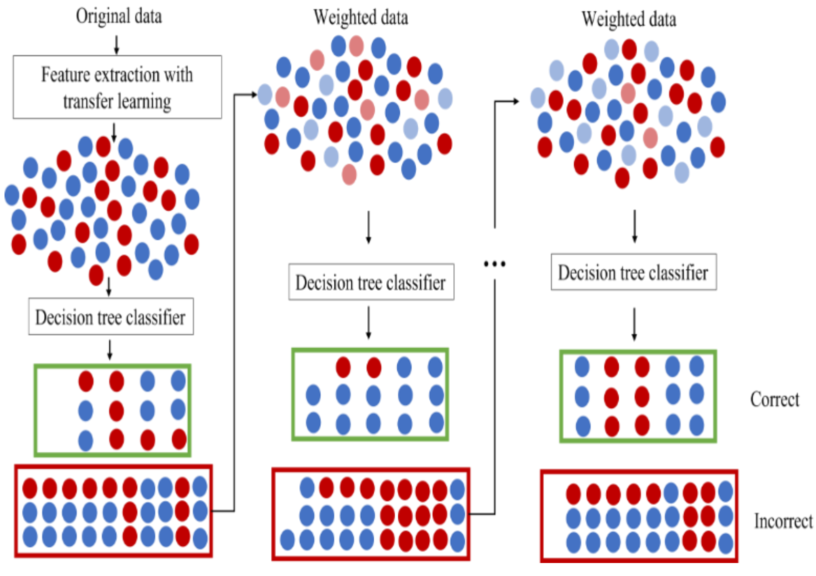


Figure 3. Example of the boosting algorithm, where the red and blue dots represent the normal and abnormal cases, and the green and red boxes represent the correct and incorrect predictions, respectively. As the model progresses, the decision tree classifier represents the weak learners, andthe faded blue and red colors represent the images that have less weight because they have alreadybeen predicted correctly. In the same fashion as the stacking method, the first part represents transfer learning for feature extraction so the data set can be transformed into a more manageable state forML algorithms. This process will iterate until the complete training data fit without error or to a specified number of estimators, which in our case is set to 200.

3.2 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.

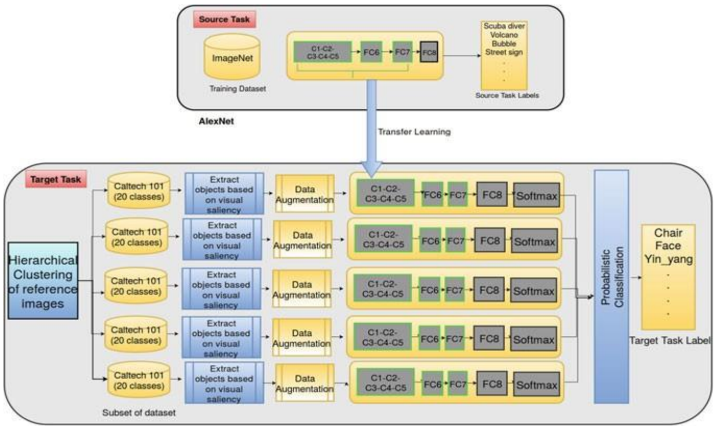


Fig. 1 Proposed Ensemble Architecture

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.

3.2.1 Transfer Learning

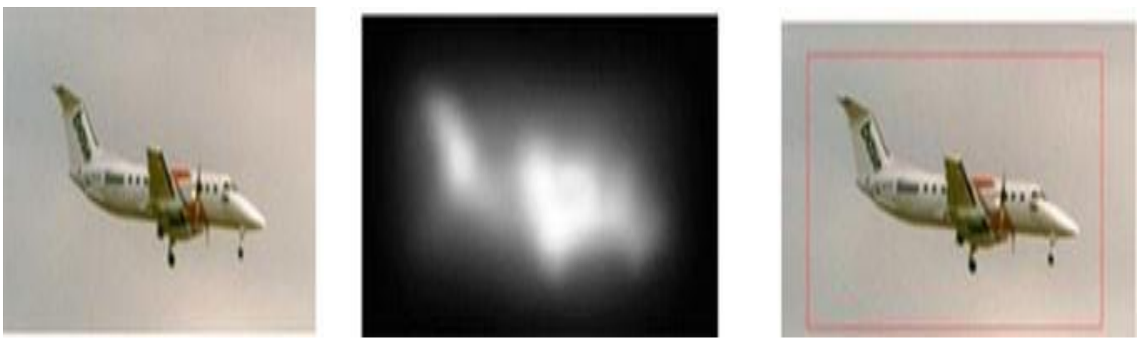
The mid-level feature representations learned by AlexNet model on ImageNet are efficiently transferred for training the new network on Caltech101. Mid-level features or generalized features are captured in the first seven layers, i.e. from first conv layer to second fully connected layer (FC7). The learned 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.

3.2.2 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 hierarchi- cal clustering. Initially, reference images are selected for each class, the one with minimum noise. Based on the similarity matrix computed, Hierarchical Agglomer- ative 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.

3.2.3 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.



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

3.2.4 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.

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 Proposed Ensemble

1. 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 ⟵ Hierarchical_Clustering(M)
6. **for** each dataset i ∈C(i) **do**
7. Saliency_Extraction(i);
8. t ⟵Train(i).
9. end for
10. Return trained model t.
11. end procedure

3.2.5 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 val- ues. 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 maxi- mum 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 proba- bility compared to other incorrect classification scores of other pipelines.

Algorithm 2 CNN Testing

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

OUTPUT: Object label with predicted score

1: **procedure**

1. 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
5. convolution and pooling operations done during training. Find maximum of scores from each pipeline. Let it be score(i), where i
6. represents pipeline.
7. final_score←max(score(i))
8. Return associated label,final_score.
9. end procedure

4. RESULT AND DISCUSS

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.

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 under- standing the correlation between the number of classes, number of pipelines in the ensemble and classification accuracy. Ensemble learning techniques have demonstrated significant improvements in image recognition accuracy. By combining multiple models, such as decision trees, neural networks, or support vector machines, ensemble methods like Random Forests or Gradient Boosting can enhance overall performance. These techniques mitigate individual model weaknesses, providing a more robust and accurate prediction for image classification tasks. Leveraging the diversity of multiple models in an ensemble helps capture complex patterns and increases the generalization capability, ultimately leading to enhanced image recognition 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

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. 1.5h
Single-nonpipelined	83	Approx.40mins

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 onepipeline and 13 in the other. In this case as well a score-averaging ensemble of com-parable 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
Cellphone	100	80
Dalmatian	100	70
Elephant	50	60
Helicopter	90	90

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)
www.ijprems.com
editor@ijprems.com
Vol. 04, Issue 01, January 2024, pp : 458-467
e-ISSN : 2583-1062
Impact Factor : 5.725

Kangaroo	100	90
Leopard	60	50
Panda	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.3h
Score averaging ensemble	78.48	approx.15h
Single-non-pipelined	78	approx.3h

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

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 alternate appraoch

Fig.5 Sample predicted output Classified. Figure 5 shoes a sample Prediction, where the given image (airplanes) is predicted with the Highest score of 1.

5. CONCLUSION

In this study we discussed an important concept called ensemble learning that is prevalent methodology in machine learning. The advantage of ensemble learning and its approaches such as, Boosting that builds a strong classifier from the number of base learners. Bagging, whichcombines the bootstrap and aggregation and it is representedas a parallel ensemble method. Stacking, in which the independent learners are combined by the learner and a mixture of experts that trains an ensemble of classifiers by applying a technique called sampling is the important contributions of our study. In future we propose to introducea machine learning based model that applies an ensemble learning techniques.

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. In this study we discussed an important concept called ensemble learning that is prevalent methodology in machine learning. The advantage of ensemble learning and its approaches such as, Boosting that builds a strong classifier from the number of base learners. Bagging, whichcombines the bootstrap and aggregation and it is representedas a parallel ensemble method. Stacking, in which the independent learners are combined by the learner and a mixture of experts that trains an ensemble of classifiers by applying a technique called sampling is the important contributions of our study. In future we propose to introducea machine learning based model that applies an ensemble learning techniques.

6. REFERENCES

[1] Szegedy, C., et al.: Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–9 (2015)

[2] Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recog- nition (2014). arXiv:1409.1556

[3] Hubel, D.H., Wiesel T.N.: Receptive elds and functional architecture of monkey striate cortex. J. Physiol. **195**(1), 21(1968)

[4] Nithin, D.K., Sivakumar, P.B.: Generic feature learning in computer vision. Procedia Comput. Sci. **58**, 202–209 (2015

[5] 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)

[6] F. T. Liu, K. M. Ting, Y. Yu, and Z.-H. Zhou, “Spectrum of variable-random trees”. Journal of Artificial Intelligence Research, pp: 355–384, (2008a).

[7] F. T. Liu, K. M. Ting, and Z.-H. Zhou, “Isolation forest”. In: Proceedings of the 8th IEEE International Conference on DataMining, pp: 413–422, Pisa, Italy. (2008b).

[8] A. Hernandez-Vela, M. A. Bautista, X. Perez-Sala, V. Ponce, X. Baro, O. Pujol, C. Angulo, and S. Escalera. Bovdw: Bag-of-visual-and-depth-words for gesture recognition. In ICPR,pages 449–452. IEEE, 2012.

[9] Harel, J., Koch, C., Perona, P.: Graph-based visual saliency. In: Advances in Neural Information Processing Systems, pp. 545–552 (2006)

[10] Kumar, K., Sachin, S., Anil, R.M., Soman, K.P.: Convolutional neural networks for the recog- nition of malayalam characters. In: FICTA 2014, pp. 493–500. Springer (2015)

[11] Zhi–Hua Zhou, “Ensemble Methods Foundation and Algorithm”, CRC press: Taylor and Francis Group, (2012). J. Huang, Z.-H. Zhou, H.-J. Zhang, and T. Chen, “Pose invariant face recognition”. In Proceedings of the 4th IEEE International Conference on Automatic Face and Gesture Recognition, pp: 245–250, Grenoble, France, (2000).

[12] Giacinto , F. Roli, and L. Didaci, “Fusion of multiple classifiers for intrusion detection in computer networks”. In: Pattern Recognition Letters, pp: 1795–1803, (2003).

[13] Hansen L, Solomon P, “Neural Network Ensembles”. In IEEE transactions on pattern analysis, pp: 993-1001, (1990).

[14] Thomas. G. Dietterich, “Ensembles methods in machine learning.” In: International workshop on multiple classifier systems. pp: 1-15, Springer, (2000).