Optimizing CNN-Based Image Classification with Ensemble Models and Transfer Learning

Muddasar Yasin
School of Electrical Engineering & Computer Science
National University of Science & Technology
Islamabad, Pakistan
myasin.msee21seecs@seecs.edu.pk

Dr. Ahmad Salman
School of Electrical Engineering & Computer Science
National University of Science & Technology
Islamabad, Pakistan
ahmad.salman@seecs.edu.pk

Dr. Haris Masood
Department of Software & Computer Engineering
University of Wah
Taxila, Pakistan
haris.masood@wecuw.edu.pk

Anam Maqbool
Department of Electrical Engineering
Air University
Islamabad, Pakistan
anam.maqbool@mail.au.edu.pk

***Abstract*—This research explores the application of Boosting Ensemble methodologies to improve the accuracy of Convolutional Neural Networks (CNNs) on diverse image datasets, such as CIFAR-10, CIFAR-100, and Fashion MNIST. The central focus is on employing AdaBoost, a widely used boosting algorithm, to enhance the performance of the underlying CNN models. Additionally, the study explores AdaBoost's effectiveness in handling imbalanced datasets, providing insights into its potential to address class imbalances.**

The empirical results highlight AdaBoost as an effective complementary strategy for enhancing CNN accuracy, particularly in scenarios with imbalanced class distributions. Noteworthy is the fact that our ensemble model achieved a 6% higher test accuracy compared to the baseline CNN. These outcomes make substantial contributions to the ongoing research in ensemble learning and offer valuable insights for practitioners involved in image classification tasks.

***Index Terms*—Convolution Neural Network, Ensemble Learning, AdaBoost, Imbalanced Data, Transfer Learning**

I. Introduction

Convolutional Neural Networks (CNNs) have emerged as highly effective tools for Computer Vision(CV) tasks, excelling in image recognition and classification [1] and many other fundamental applications in CV. By utilizing convolutional layers to automatically learn intricate features from raw pixel data, CNNs showcase a unique ability to capture spatial hierarchies. This characteristic makes them pivotal in various applications related to image processing and analysis.

In parallel, *Ensemble Learning* has gained recognition as a powerful strategy to enhance the performance of machine learning models [2]. Ensemble methods operate on the principle of making collective decisions [3]. This involves a group of individual classifiers working together to determine the most appropriate output. The decision-making process can be achieved through voting or averaging probabilities. In the case of voting, each classifier predicts a class, and the final class is determined through a voting mechanism among them. To avoid tie situations, it is recommended to use an odd number of classifiers.

Alternatively, individual classifiers can predict the probability for a class, and the final class is determined by averaging these probabilities. The former approach is termed as hard voting, while the latter is referred to as soft voting. [4] Ensemble methods enhance performance by reducing the variance in prediction errors made by the individual classifiers. We encounter ensemble learning in our daily lives, such as when deciding to watch a movie based on review ratings, which essentially represents a collective decision.

The foundation of ensemble learning lies in the concept of the wisdom of the crowd. This theory suggests that combining knowledge from multiple sources often leads to decisions that are superior to those made by a single entity. In 1990, Schapire [5] introduced a novel approach known as the Adaboost algorithm, which combines several weak learners to function collectively as a strong learner.

Since 2008, researchers have been utilizing ensemble learning approaches to address real-life challenges in various domains, including petrochemicals, bioinformatics, medicine, remote sensing, education, and software bug detection. An ensemble model involves the collaboration of multiple classifiers that train on the same dataset, and their outputs are combined using methods such as weighted averaging, simple averaging, voting, or probability. Ensemble methods leverage this concept in addressing machine learning (ML) problems, working towards predicting the most accurate output compared to relying on a single method.

The challenge of handling class imbalance in classification scenarios, where certain classes are underrepresented, has prompted exploration into Ensemble Learning techniques such as AdaBoost [6]. AdaBoost, or Adaptive Boosting, sequentially trains weak learners, assigning higher weights to misclassified instances to iteratively improve accuracy. This study investigates the synergy between CNNs, Ensemble Learning, and AdaBoost, seeking to leverage the strengths of both approaches to address the complexities associated with imbalanced datasets.

In brief, this study presents several significant findings 1)- Demonstrating the effectiveness of ensemble learning in enhancing the generalization and accuracy of convolutional neural networks. 2)- Illustrating the utility of AdaBoost in addressing imbalanced datasets, resulting in improved outcomes. 3)- Comparing the performance of ensemble models with Transfer Learning algorithms and providing a detailed comparison in Table VI.

Upcoming sections will delve into a detailed exploration of literature and methodology, shedding light on the investigation's underpinning methodology, and contributing to the ongoing dialogue on neural network optimization in the era of artificial intelligence.

II. Literature Review

This exploration encompasses scrutinizing research studies, methodologies, and progressions intended to utilize ensemble strategies, notably AdaBoost, for augmenting CNN model's performance. CNNs have emerged as powerful tools for Image recognition [7], pattern analysis [8], and feature extraction [9]. As the pursuit of enhanced performance continues, researchers have turned their attention toward leveraging the strength of ensemble learning techniques to further boost the capabilities of convolution neural networks. Gowthami S and Harikumar R [10] focus on the performance analysis of boosting-based transfer learning in deep CNN for image classification, addressing the challenges of imbalanced datasets and improving classifier performance. The experiments conducted on benign and malignant melanoma images from the International Skin Imaging Collaboration(ISIC) database demonstrate the effectiveness of the approach, achieving an accuracy of 99.19 % and a sensitivity of 98.46%. Neelesh Mungoli [11] has proposed an Adaptive Ensemble Learning framework that combines ensemble learning strategies with deep learning architectures to enhance the performance of deep neural networks. By leveraging intelligent feature fusion methods, the framework generates more discriminative and effective feature representations, leading to improved model performance and generalization capabilities.

Tsehay Admassu Assegie [12] proposes a breast cancer prediction model using decision tree and adaptive boosting (Adaboost) algorithms. The model is evaluated using an extensive experimental analysis on a breast cancer dataset from the Kaggle data repository. The dataset consists of 569 observations, with 37.25% being benign and 62.74% being malignant. The class distribution of the dataset is highly imbalanced, leading to poor performance of the decision tree algorithm in predicting malignant observations. To address this, the adaptive boosting algorithm is employed to improve the performance of the decision tree on malignant observations. The analysis shows that the adaptive boosting algorithm achieves an accuracy of 92.53%, while the decision tree algorithm achieves an accuracy of 88.80%. Haoyu Zhang, Yushi Chen, and Xin He [13] proposed a method called Boosting-CNN that combines deep convolutional neural networks (CNNs) and ensemble learning for hyperspectral image (HSI) classification. It uses multiple well-designed CNNs and adaptive boosting to improve classification accuracy. The final classification result is obtained through weighted voting of the CNNs. To address the issue of imbalanced training samples in HSI classification, the paper introduces a soft class balanced loss to mitigate the influence of imbalance. Experimental results on two popular hyperspectral datasets (Salinas and Pavia University) demonstrate that the proposed method achieves better classification accuracy compared to other methods.

Aboozar Taherkhani [14] proposed AdaBoost-CNN, an Adaptive Boosting algorithm that enhances the classification performance of traditional CNN models for multi-class imbalanced datasets using transfer learning techniques. The algorithm achieves improved accuracy, precision, and recall compared to traditional CNN models and outperforms other state-of-the-art algorithms, such as Random Forest and Support Vector Machines, in terms of classification accuracy and F1-score. Shin-Jye Lee [15] presented the usage of a trained deep convolutional neural network model to extract image features and the AdaBoost algorithm to assemble Softmax classifiers, resulting in improved accuracy and reduced retraining time cost. Ricardo Fuentes [16] proposed Adaptive Robust Transfer Learning (ART), a flexible pipeline for transfer learning with machine learning algorithms, providing theoretical guarantees for adaptive transfer and preventing negative transfer demonstrating the promising performance of ART through empirical studies on regression, classification, sparse learning, and a real-data analysis for a mortality study. Ke Zhao, Feng Jia and Haidong Shao [17] proposed a method called transfer adaptive boosting with Squeeze-and-Excitation Attention Convolutional Neural Network (SEACNN) to address the issue of unbalanced fault diagnosis in rolling bearings. The method combines an SEACNN model for feature extraction and identification, with an AdaBoost algorithm for handling unbalanced fault datasets. Transfer learning is also employed to transfer knowledge from one SEACNN estimator to the next, improving the identification performance. The proposed method is evaluated through extensive experiments, demonstrating its effectiveness in accurately classifying unbalanced datasets in fault diagnosis of rolling bearings. Yuki Kawana, Norimichi Ukita, Jia-Bin Huang, and Ming-Hsuan Yang [18] introduced an approach employing a CNN to capture intricate interdependencies. This network utilizes deep convolution and deconvolution layers to achieve comprehensive representations, resulting in resilient and precise pose estimation. The effectiveness of the proposed ensemble model [18] is assessed on publicly available datasets, showcasing favorable performance in comparison to baseline models and state-of-the-art methods.

III. Proposed Methodology

The proposed methodology encompasses a dual-phase approach. Initially, a foundational Convolutional Neural Network undergoes training on a specified dataset. Subsequently, employing the fundamental tenets of Transfer Learning, the learned weights from the CNN are harnessed to train the ensemble model, thereby augmenting the overall accuracy of the CNN.

A. Training a CNN

A basic Convolutional Neural Network (CNN) undergoes training through a structured sequence of layers: convolutional layers, pooling layers, and finally fully connected layers [19]. Think of it like building blocks, where the lower layers discern simple things, and the higher layers understand more complex stuff. The initial convolutional layers focus on extracting local details from the input, generating distinct "feature maps" for different aspects. They use shared weights known as "kernel" to map the input to these feature maps. Then, a non-linear function like ReLU or sigmoid is used to improve the results.

After each convolutional layer, a max-pooling layer picks the most important information, reducing the data size and making things easier to handle. Following the convolutional layers, there are fully connected hidden layers that get the important features in a rearranged way. To make this work, the outputs from the convolutional layers are flattened into a single vector. These layers use non-linear functions to add complexity to the decision-making process.

At the top of this setup, there's a logistic regression model that uses the knowledge gathered from the previous layers. Its job is to create a final output, sorting the input into different categories. To do this, it uses the SoftMax function, which turns the output into a probability distribution, showing how likely each category is. This process helps make well-informed decisions.

B. Ensemble Configuration

After the basic CNN learns some things, an Ensemble of models is created, which is a collection of multiple Convolutional Neural Network (CNN) models. Each CNN model in the ensemble is considered a "weak learner" because it may not be individually highly accurate, but the ensemble aims to combine their strengths for better overall performance. Each CNN model in the ensemble is trained on the entire dataset (both features and labels). During training, the model learns to recognize patterns and relationships within the data that allow it to make predictions. After training each model, its performance is evaluated on the training set. The evaluation involves making predictions on the training set and comparing them to the actual labels.

The error is calculated by measuring the disagreement between the predicted labels and the actual labels as shown in equation 1.

$$\epsilon = \frac{\Sigma_{i=1}^{N} \delta(y^*, i \neq y, i)}{N} \tag{1}$$

where ϵ= Error rate, N = Number of weak Estimators, y^* is the predicted label, y is the true lable for sample i and δ is the kronecker delta function which returns 1 if the condition inside is true and 0 otherwise. It indicates whether the predicted label is not equal to the true label.

The error indicates how well or poorly the model is performing on the training set. The weights assigned to each model are calculated using the AdaBoost algorithm. AdaBoost assigns higher weights to models that perform well (have lower error) and lower weights to models that perform poorly. Once the models are trained and assigned weights, they are used to make predictions on a test set. For each model, predictions are made, and these predictions are weighted based on the previously assigned weights.

The final predictions for the ensemble are obtained by combining the weighted predictions of each individual model. The mathematical equation for combining weights are shown in equation 2.

$$w = \eta * \ln \frac{(1-\epsilon)}{\epsilon} \tag{2}$$

where w = model weight, η = learning rate, hyperparameter that controls the step size of weight update, ϵ is the error rate calculated from the models prediction. The performance of the model depends upon the error rate. Higher the error rate, the weight will be adjusted more. The logarithmic function helps to scale the adjustment of weight.

The models with higher weights contribute more to the final prediction, while those with lower weights contribute less. The rationale behind using an ensemble is that even if individual models are not highly accurate, their diverse perspectives and strengths may complement each other. By combining the predictions of multiple weak learners with different focuses, the ensemble aims to achieve a more robust and accurate prediction on the test set. Flowchart for ensemble model can be seen in figure 1.

IV. Dataset Description

Ensemble model performance is evaluated on three datasets i.e. CIFAR10 [20], CIFAR100 [20] and Fashion-MNIST [21] in this section.

A. CIFAR10 Dataset

The Canadian Institute For Advanced Research(CIFAR) [20] dataset is a popular benchmark dataset in the field of machine learning and computer vision. The "10" in CIFAR-10 [20] represents the number of different classes or categories present in the dataset. CIFAR-10 consists of color images, each of size 32x32 pixels. The dataset is divided into ten

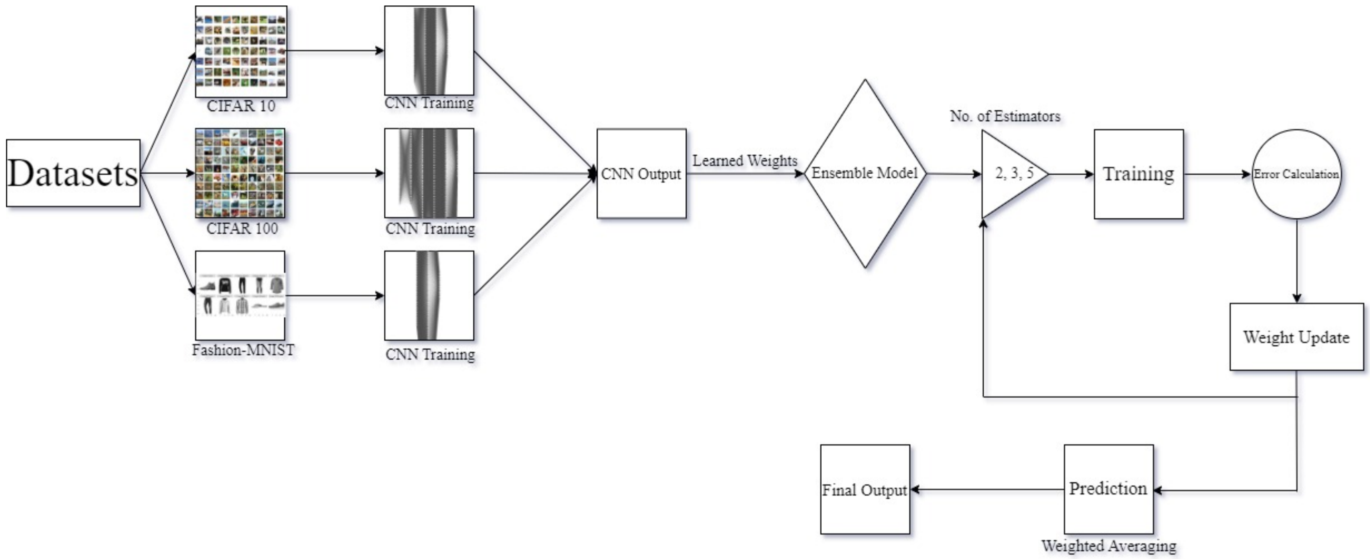

Fig. 1. Block Diagram of Ensemble Learning

classes such as airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck. The dataset contains a total of 60,000 images. The images are split into 50,000 for training and 10,000 for testing, providing a standard split for evaluating model performance.

B. CIFAR100 Dataset

CIFAR-100 [20] is a widely used dataset in the field of machine learning and computer vision. It is an extension of the CIFAR-10 dataset and consists of 60,000 32x32 color images in 100 different classes, with each class containing 600 images. The dataset is divided into 50,000 training images and 10,000 testing images. Each image in CIFAR-100 belongs to one of the 100 classes, and these classes are further grouped into 20 super classes. The dataset is designed to be challenging, covering a diverse range of object categories. Some examples of classes in CIFAR-100 include ”apple,” ”beaver,” ”clock,” ”forest,” ”man,” and ”woman.”

C. Fashion MNIST Dataset

Fashion-MNIST [21] is a dataset of Zalando’s article images—consisting of a training set of 60,000 examples and a test set of 10,000 examples. Each example is a 28x28 grayscale image, associated with a label from 10 classes. Each training and test example is assigned to one of the labels such as T-shirt/top, Trouser, Pullover, Dress, Coat, Sandal, Shirt, Sneaker, Bag and Ankle boot.

V. Experimental Results

In this section, the experimental test results on the proposed model is explained. Performance of ensemble model is compared with Transfer Learning Algorithm and benchmark CNN using CIFAR-10, CIFAR100 and Fashion MNIST Dataset.

A. Experimental Results of AdaBoost with Decision Tree

Y Freundand RE Schapire [5] proposed AdaBoost demonstrating significant efficacy in tasks involving binary classification with decision tree as weak classifier, where the primary goal is to distinguish between two distinct classes. The algorithm’s adaptability and its focus on misclassified instances during training make it particularly adept at addressing class imbalances. Its design is tailored to enhance the performance of weak classifiers, facilitating the amalgamation of their predictions to construct a robust classifier.

Moreover, AdaBoost showcases adaptability to the underlying data distribution by dynamically adjusting instance weights during the training phase. This flexibility proves beneficial, especially in scenarios where one class is underrepresented, contributing to the algorithm’s success in handling imbalanced datasets.

On the contrary, training convolutional neural networks (CNNs) on imbalanced datasets poses challenges, potentially leading to suboptimal model performance. CNNs’ inherent bias towards the majority class in the presence of imbalances may result in prioritizing the dominant class, potentially neglecting the minority class and yielding subpar generalization. It’s crucial to note that accuracy can be a deceptive metric in imbalanced settings, as high accuracy may be achieved by predominantly predicting the majority class, even if performance on minority classes is inadequate.

While AdaBoost excels in binary and imbalanced data scenarios, utilizing a basic decision tree as a weak classifier may pose limitations in handling complex datasets or multiclass scenarios like CIFAR-10 and MNIST. Decision trees’ simplicity may hinder their ability to capture intricate relationships within data, particularly in the presence of diverse classes.

To validate these points, we generated a binary class imbalanced dataset of cat and dog images. Initially balanced, the dataset comprised 279 training images for each class and 70 test images per class, totaling 558 training images and 140 test images. For experimental purposes, intentional efforts were made to create an imbalanced dataset. In this modified version, the training set retained 279 images for the dogs class while intentionally reducing the number of cat images to 71. This deliberate imbalance was introduced to explore and assess the performance of both ensemble models and CNNs under such conditions.

Results of AdaBoost on imbalance binary dataset and CIFAR-10 and MNIST [22] dataset are shown in the following table with varrying number of estimators.

TABLE I
RESULTS USING ADABOOST WITH DECISION TREE AS WEAK CLASSIFIER

Dataset	Estimators	AdaBoost Accuracy	CNN Accuracy
Cats & Dogs	20	90.00 %	52.14 %
Cats & Dogs	40	96.54 %	52.14 %
MNIST	20	69.67 %	97.22 %
MNIST	40	73.51 %	97.22 %
CIFAR10	20	28.75 %	69.29 %
CIFAR10	20	30.37 %	69.29 %

B. Results for Using Past Knowledge for Better Accuracy:

The approach involves transferring what the first CNN has come to know to a second CNN that is frequently designed for a similar dataset. The aim is to leverage the useful pieces of information that the first CNN revealed concerning similar and unrelated problems. Since using old weights to improve the accuracy increases the speed with which later CNNs can learn, especially when the new job is somewhat similar to the first job. This helps the model to have prior knowledge and hence improve performance even when there is little labeled data. While some old weights can be of value, they might not always be beneficial, even more, so if the jobs are too dissimilar.
Rather than using a pre-trained model, we will first train a CNN and use its weights for the next CNN. We will freeze the convolution layers of the second model and change the fully connected layers. After the evaluation, there was almost 2.91% increase in the accuracy of the 2nd model as compared to the first model. An accuracy comparison for CIFAR10, CIFAR100 and Fashion MNIST is given in Table II.

TABLE II
ACCURACY BY USING WEIGHTS OF PREVIOUS CNN

Dataset	CNN Accuracy	Accuracy Using Previous Weights
CIFAR10	70.13%	73.04%
CIFAR100	43.22%	46.73%
Fashion Mnist	91.61%	91.64%

The CNN architectures are tailored differently for each dataset to optimize accuracy. For instance, in Fashion MNIST, a simpler CNN design is effective, leveraging the grayscale nature of the dataset to achieve accuracies reaching 90%. On the other hand, CIFAR-100, characterized by more intricate images, necessitates deeper architectures with increased layer complexity to enhance accuracy. Architecture used for training CIFAR10, CIFAR100 and Fashion MNIST is given in the following table III.

TABLE III
CNN ARCHITECTURE USED FOR CIFAR10, CIFAR100 & FASHIONMNIST

Dataset	Classes	Layers	Filters	Activation
		Conv2D	32	ReLU
		Max-Pooling	NA	NA
		Conv2D	64	ReLU
		Max-Pooling	NA	NA
		Dense	128	ReLU
CIFAR10	10	Dense	10	Softmax
		Conv2D	32	
		Conv2D	32	ReLU
		Max-Pooling	NA	ReLU
		Conv2D	64	NA
		Conv2D	64	ReLU
		Max-Pooling	NA	ReLU
		Dropout	50%	NA
		Dense	512	ReLU
		Dropout	50%	NA
CIFAR100	100	Dense	10	Softmax
		Conv2D	32	ReLU
		Max-Pooling	NA	NA
		Dense	128	ReLU
FashionMNIST	10	Dense	10	Softmax

C. Result of Ensemble Model on CIFAR10, CIFAR100 & FashionMNIST

Experimental results for CIFAR10, CIFAR100 & FashionMNIST using the ensemble model are discussed in this section. Ensemble model is tested for different number of estimators and all estimators are tested for different number of eppochs. For five estimators and each estimator tested for 15 training epochs ensemble model gave 76.27% accuracy for CIFAR10 dataset respectively. CNN was trained for 15 epochs. Results for different number of estimators and epochs are given in the table IV:

TABLE IV
ACCURACY USING ENCEMBLE MODEL

Dataset	Estimators	Ensemble Accuracy	CNN Accuracy
CIFAR10	02	76.25% %	70.13 %
CIFAR10	05	76.03% %	70.13 %
CIFAR100	02	48.00% %	43.22 %
CIFAR100	05	50.12%	43.22 %
FashionMNIST	02	92.7% %	91.60 %
FashionMNIST	05	92.5% %	91.60 %

When number of epochs for CNN were changed to 20 from 15 it resulted in the change of accuracy of CNN. So, number of epochs for ensemble model were kept to 15 to check the effect of change in accuracy of ensemble model compariosn in CNN. Results for accuracy of CNN for 20 epochs and Ensemble model is in table V:

TABLE V
RESULTS WITH INCREASED CNN EPOCHS

Dataset	Ensemble Accuracy	Epochs for CNN	CNN Accuracy
CIFAR10	76.47%	20	68.52%
CIFAR100	747.88%	20	45.62%

VI. COMPARISON OF TRANSFER LEARNING & ENSEMBLE MODEL

Transfer learning seeks to improve the performance of target learners in specific domains by leveraging knowledge from different yet related source domains [23]. The goal is to enhance a learner in one domain by transferring valuable information from a related domain. In cases where obtaining training data is expensive or challenging, there is a need to develop high-performance learners trained with readily available data from diverse domains, commonly referred to as transfer learning [24].

In the realm of traditional machine learning, both training and testing data typically share the same input feature space and data distribution. Discrepancies in data distribution between the two sets can result in a degradation of the predictive learner's performance. The necessity for transfer learning arises when there is a limited supply of target training data, attributed to factors such as data rarity, high costs associated with data collection and labeling, or the inaccessibility of the data. In our experiments, we applied transfer learning to train models on the CIFAR-10 and CIFAR-100 dataset using ResNet50 and AlexNet [19] as pre-trained models [25]. The accuracy comparison of transfer learning using various models, along with an ensemble model and a simple CNN, is presented in the table below.

TABLE VI
COMPARISON OF TRANSFER LEARNING & ENSEMBLE MODEL

Dataset	AlexNet	ResNET50	Ensemble Accuracy
CIFAR10	36.12%	32.08%	76.47%
CIFAR100	44.10%	48.82%	50.12%

Utilizing transfer learning is a beneficial strategy to harness knowledge from related domains, yet it presents challenges related to adaptability and domain mismatch. The reliance on pre-trained models in transfer learning may hinder adaptability to the unique characteristics of the target dataset. The knowledge transferred from the source domain may not seamlessly align with the nuances of the target domain. This methodology assumes a shared set of features between the source and target domains. However, if there is substantial dissimilarity between the domains, the transferred knowledge may not effectively contribute to the target task. Neha Sharma,Vibhor Jain and Anju Mishra [25] concluded in their results that higher number of layers are required to get higher accuracy. The findings indicated that networks trained through transfer learning performed better than existing ones, demonstrating elevated accuracy rates. Specific objects such as ”chair,” ”train,” and ”wardrobe” achieved flawless recognition with 147-layered networks, while objects like ”cars” exhibited perfect recognition with 177-layered networks [25]. Additionally, the implementation of transfer learning often entails the use of pre-trained models, which can exhibit complex architectures.

VII. CONCLUSION

In this paper, the use of an ensemble model, particularly incorporating AdaBoost, has emerged as an effective strategy for enhancing the accuracy of Convolutional Neural Networks (CNNs). The primary objective of this research was to boost the performance of CNNs by leveraging the strengths of diverse models through ensemble learning. The results obtained have showcased significant advancements compared to standalone CNNs.

A comparative analysis between ensemble models and alternative techniques, such as transfer learning, indicated that the ensemble approach not only surpassed in terms of accuracy but also demonstrated a reduction in the number of parameters. This reduction is particularly noteworthy as it directly translates into a decrease in computational costs, rendering the ensemble model more resource-efficient and practical for real-world applications.

A notable aspect of this study is the successful training of the AdaBoost on an imbalanced dataset. The AdaBoost approach exhibited superior results in addressing class imbalances compared to the standalone CNN. This implies that AdaBoost, as a component of the ensemble, contributes to the model's robustness in scenarios where class distribution is uneven.

VIII. FUTURE WORK

Moving forward, potential research directions in this domain could explore diverse avenues for further improvement. Firstly, investigating alternative ensemble techniques beyond AdaBoost, such as bagging or stacking, could yield additional insights into optimal model combinations for enhancing CNN performance. Additionally, exploring the impact of varying ensemble sizes and incorporating different base models within the ensemble may lead to the identification of more effective configurations.

Furthermore, addressing the interpretability of ensemble models remains a crucial aspect for broader adoption in real-world applications. Developing methodologies to interpret and explain the decisions made by the ensemble could enhance the model's trustworthiness and applicability in sensitive domains.

Finally, with the continuous evolution of technology, integrating ensemble models with emerging techniques like neural architecture search (NAS) or automated machine learning (AutoML) could pave the way for more efficient and adaptive models. These approaches have the potential to automate the process of selecting optimal architectures and hyperparameters, thereby reducing the burden on practitioners.

During the preparation of this work we used ChatGPT, Grammarly in order to improve the clarity and coherence of the writing. After using these tools, the work is totally reviewed and edited the content as needed and we take full responsibility for the content of the publication.

REFERENCES

[1] Rafał Scherer. *Computer vision methods for fast image classification and retrieval*. Springer, 2020.

[2] TG Dietterich et al. Ensemble learning. the handbook of brain theory and neural networks. *Arbib MA*, 2(1):110–125, 2002.

[3] Robi Polikar. Ensemble based systems in decision making. *IEEE Circuits and systems magazine*, 6(3):21–45, 2006.

[4] Suyash Kumar, Prabhjot Kaur, and Anjana Gosain. A comprehensive survey on ensemble methods. In *2022 IEEE 7th International conference for Convergence in Technology (I2CT)*, pages 1–7. IEEE, 2022.

[5] Yoav Freund and Robert E Schapire. A desicion-theoretic generalization of on-line learning and an application to boosting. In *European conference on computational learning theory*, pages 23–37. Springer, 1995.

[6] Wonji Lee, Chi-Hyuck Jun, and Jong-Seok Lee. Instance categorization by support vector machines to adjust weights in adaboost for imbalanced data classification. *Information Sciences*, 381:92–103, 2017.

[7] Rahul Chauhan, Kamal Kumar Ghanshala, and RC Joshi. Convolutional neural network (cnn) for image detection and recognition. In *2018 first international conference on secure cyber computing and communication (ICSCCC)*, pages 278–282. IEEE, 2018.

[8] Louis Lettry, Michal Perdoch, Kenneth Vanhoey, and Luc Van Gool. Repeated pattern detection using cnn activations. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 47–55. IEEE, 2017.

[9] Manjunath Jogin, MS Madhulika, GD Divya, RK Meghana, S Apoorva, et al. Feature extraction using convolution neural networks (cnn) and deep learning. In *2018 3rd IEEE international conference on recent trends in electronics, information & communication technology (RTEICT)*, pages 2319–2323. IEEE, 2018.

[10] S Gowthami and R Harikumar. Performance analysis of incremental boosting based transfer learning in deep cnn. In *2022 3rd International Conference on Communication, Computing and Industry 4.0 (C2I4)*, pages 1–6. IEEE, 2022.

[11] Neelesh Mungoli. Adaptive ensemble learning: Boosting model performance through intelligent feature fusion in deep neural networks. *arXiv preprint arXiv:2304.02653*, 2023.

[12] Tsehay Admassu Assegie, R Lakshmi Tulasi, and N Komal Kumar. Breast cancer prediction model with decision tree and adaptive boosting. *IAES International Journal of Artificial Intelligence*, 10(1):184, 2021.

[13] Haoyu Zhang, Yushi Chen, Xin He, and Xingliang Shen. Boosting cnn for hyperspectral image classification. In *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*, pages 3673–3676. IEEE, 2021.

[14] Aboozar Taherkhani, Georgina Cosma, and T Martin McGinnity. Adaboost-cnn: An adaptive boosting algorithm for convolutional neural networks to classify multi-class imbalanced datasets using transfer learning. *Neurocomputing*, 404:351–366, 2020.

[15] Shin-Jye Lee, Tonglin Chen, Lun Yu, and Chin-Hui Lai. Image classification based on the boost convolutional neural network. *IEEE Access*, 6:12755–12768, 2018.

[16] Boxiang Wang, Yunan Wu, and Chenglong Ye. The art of transfer learning: An adaptive and robust pipeline. *Stat*, page e582, 2023.

[17] Ke Zhao, Feng Jia, and Haidong Shao. Unbalanced fault diagnosis of rolling bearings using transfer adaptive boosting with squeeze-and-excitation attention convolutional neural network. *Measurement Science and Technology*, 34(4):044006, 2023.

[18] Yuki Kawana, Norimichi Ukita, Jia-Bin Huang, and Ming-Hsuan Yang. Ensemble convolutional neural networks for pose estimation. *Computer Vision and Image Understanding*, 169:62–74, 2018.

[19] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[20] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[21] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[22] Li Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE signal processing magazine*, 29(6):141–142, 2012.

[23] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1):43–76, 2021.

[24] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. A survey of transfer learning. *Journal of Big data*, 3(1):1–40, 2016.

[25] Neha Sharma, Vibhor Jain, and Anju Mishra. An analysis of convolutional neural networks for image classification. *Procedia computer science*, 132:377–384, 2018.