

A Modified Channel Attention Mechanism For Detecting Diseases In Tea Leaves

Anuj Kumar Das¹, Syed Sazzad Ahmed^{2*} and Kshirod Sarmah^{3*}

^{1,2}Department of Computer Science and Engineering, Assam Don Bosco University, Guwahati, Assam, India, ad6428989@gmail.com, sazzad@dbuniversity.ac.in

³Department of Computer Science, Pandit Deendayal Upadhyaya Adarsha Mahavidyalaya (A Govt. Model College), Goalpara, Assam, India, kshirodsarmah@gmail.com ORCID: (0009-0009-3398-3684)¹, (0000-0001-7476-2604)², (0009-0006-8780-5689)³

*Corresponding Authors: Kshirod Sarmah, Syed Sazzad Ahmed

Abstract

Detection of diseases in agricultural crop is crucial for improving the quality and yield of agricultural produce. Manual disease detection and identification requires expert guidance and is a costly process. Computing technologies can be highly effective in this endeavour. Convolutional Neural Networks (CNN) are computer vision algorithms that can be used for this purpose. Integration of attention modules with convolutional neural networks can improve the performance of the network. In our study we have integrated channel attention module with MobileNetV2 for detecting diseases in tea leaves. Furthermore, we introduced several modifications to the channel attention mechanism and evaluated their effectiveness in enhancing disease detection performance.

Keywords: CNN, Channel Attention, MobileNetV2, Transfer Learning.

INTRODUCTION

Tea (*Camellia sinensis*) is one of the world's most consumed beverages, cherished not only for its taste but also for its versatile applications across health, beauty, culinary arts, and industry. The global tea market is projected to grow from USD 69.26 billion in 2024 to USD 122.59 billion by 2033 [1], registering a compound annual growth rate (CAGR) of 6.55% between 2025 and 2033. Global tea production is dominated by countries like China, India, Kenya, and Sri Lanka [2]. Tea cultivation provides livelihood to people in these regions. Hence, it is imperative to ensure that both the quality and quantity of tea production remain high. However, the prevalence of disease remains a significant impediment to maintaining excellence in both the quality and quantity of tea production. The occurrence of diseases in tea plants is attributed to a range of factors, broadly categorized as biotic and abiotic. Biotic factors include diseases resulting from living pathogens such as fungi, bacteria, viruses, nematodes, algae etc. While abiotic factors include diseases resulting from environmental conditions and agricultural practices [3]. The effectiveness of prevention and cure is subject to precise and timely detection and identification of these diseases [4]. Traditional way involves detecting and identifying diseases with the naked eye. This process requires expert guidance, which is costly and time consuming.

Convolutional Neural Network (CNN) has revolutionized the field of plant disease detection [5,6]. CNNs are deep neural network that has a large number of hidden layers. Most of the layers are convolution layers that has the ability to extract features automatically through a convolution operation [7]. In convolution operation a dot product is performed between the values in the input feature map and that of a smaller sized matrix known as kernel or filter [8]. CNNs consists of other layers as well. Pooling layers are used for dimension reduction. Batch Normalization layers are used to normalize the inputs to layers. This accelerates the training process and regularizes the model [9]. The final layers of a CNN are the fully connected layers that performs the classification activity. This ability of CNN to automate the feature extraction and classification process has made it one of the most sought-after algorithms for computer vision task [10,11]. However, training a CNN effectively requires a large and diverse set of training samples.

To improve the performance of CNN, attention modules can be integrated with the network. Attention modules mimic the human visual system where attention is given to only the relevant features of an image [12,13]. Different types of attention modules have been proposed by researchers based on the type of attention required on the datasets. Channel attention [14], spatial attention [15], self-attention [16] are some of the

attention modules that are used to enhance model's performance. Hu et al. [17] proposed a channel attention module which employs global average pooling operation to squeeze the input and then uses multi-layer perceptron to generate weights for the channels. The weights were then assigned to the feature maps using multiplication operation. Woo et al. [18] modified the channel attention method proposed by Hu et al. [17] by implementing a global max pooling layer parallel to the global average pooling layer. Woo et al. [18] also adds a spatial attention module. This resulted in a better efficiency.

In this paper we have performed a study of the channel attention module for detecting diseases in tea leaves. The channel attention module proposed by Woo et al. [18] has been considered for the study. Also, we have proposed a modification of the channel attention module and examined its performance.

RELATED WORK

Malik et al. [19] proposed a customized CNN called TreeNet for disease detection in corn leaf. Features were extracted using TreeNet, DarkNet-53 and DenseNet201. Extracted features were fused using serial-based fusion method. Classification feature set is optimized using Entropy-coded Sine Cosine Algorithm. Support Vector Machine (SVM) and K-Nearest Neighbor (KNN) were used for classification. The model showed improved performance.

Chakrabarty et al. [20] used a model based on optimized Bidirectional Encoder representation from Image Transformers (BEiT) model and pretrained CNNs for detecting diseases in rice leaves. Local Interpretable Model-agnostic Explanations (LIME) and Simple Linear Iterative Clustering (SLIC) segmentation were used to interpret the decisions of the model.

Madeira et al. [21] proposed a customized CNN for vineyard disease detection. The model was tested against ResNet50 and MobileNetV2. MobileNetV2 demonstrated the best performance followed by the custom CNN.

Dharwadkar et al. [22] proposed a hybrid model for prediction of crop yields for rice and wheat. The proposed model integrates a 1D CNN along with a Long Short-Term Memory (LSTM) network and an attention layer. The model performed well when compared to conventional methods like SVM, Decision Tree Regressor and Random Forest Regressor.

Benfenati et al. [23] experimented with two different approaches for disease detection in cucumber leaves using multispectral images. The first approach consists of a convolutional autoencoder to extract compressed features followed by K-means clustering to group leaves based on disease presence and severity. While the second approach uses a deep CNN with residual connections to detect anomalies by training only on healthy leaf images. It then calculates an anomaly score based on reconstruction error. This helps to distinguish diseased leaves by the deviation from the learned healthy patterns.

Yao et al. [24] proposed a novel method for tea leaf blight disease detection. The proposed model is based on YOLOV7-tiny with three major enhancements. The novel method introduces a decoupled detection head to separate feature processing for classification and localization. Then a triple attention mechanism is added in the backbone to focus on critical spatial and channel information. Wasserstein Distance-Based Loss Function was used to improve robustness of bounding box regression.

Datta and Gupta [25] proposed a customized CNN with 16 layers, having 6 convolutional, 6 pooling and 2 fully connected layers for disease detection in tea leaves. Adam is used as the optimizer and Sparse Categorical Cross-Entropy is used as the loss function. An API backend was also developed to enable integration with IoT devices, mobile and web applications for real time detection of diseases.

METHODOLOGY

In our study we have used the CNN architecture MobileNetV2 [26] for feature extraction. Then the channel attention module was used to assign weights to the extracted features. Finally, fully connected layers were used for classification.

MOBILENETV2

MobileNetV2 is the enhanced version of the MobileNet [27] architecture. The MobileNet architecture employs depth-wise separable convolutions, which break down a standard convolution into depth-wise and point-wise convolutions. This approach reduces computational overhead by a factor of 8 to 9 compared to standard convolutions. To lower the computational cost even further, MobileNet uses two hyperparameters: the width multiplier and the resolution multiplier. MobileNetV2 enhances the MobileNet architecture by introducing the concepts of inverted residuals and linear bottleneck layers, and it utilizes ReLU6 as the activation function.

CHANNEL ATTENTION

Within the channel attention module [18], the input features $F \in \mathbb{R}^{H \times W}$ were subjected to concurrent processing via global average pooling [28] and global max pooling [29] mechanisms, facilitating the abstraction of salient feature representations. Here, $H \in \mathbb{Z}^+$ and $W \in \mathbb{Z}^+$ represents the height and width of the input feature map. As a result, the refined feature maps, F_{GA} and F_{GM} , emerge from the respective global average and max pooling transformations and is given by,

$$F_{GA} = \frac{1}{H \cdot W} \sum_{i=0}^{H-1} \sum_{j=0}^{W-1} F_{i,j} \quad (1)$$

$$F_{GM} = \max_{0 \leq i < H, 0 \leq j < W} F_{i,j} \quad (2)$$

$F_{i,j}$ represents the value at position (i, j) in the input feature map.

Then a shared Multi-Layer Perceptron (MLP) is used to process these feature maps individually. The MLP has a hidden layer of $\frac{c}{r}$ neurons, where 'c' is the number of input vectors and 'r' is the reduction ratio. The reduction ratio reduces the number of neurons by 'r' times. ReLU activation is used in the hidden layer to obtain the outputs F_{GAH} and F_{GMH} given by:

$$(F_{GAH})_j = \text{ReLU} \left(\sum_{i=1}^n W_{j,i} \cdot (F_{GA})_i + b_j \right) \quad (3)$$

$$(F_{GMH})_j = \text{ReLU} \left(\sum_{i=1}^n W_{j,i} \cdot (F_{GM})_i + b_j \right) \quad (4)$$

The final layer of the MLP contains 'c' neurons and generate the outputs F_{GAF} and F_{GMF} and is given by:

$$(F_{GAF})_k = \sum_{j=1}^n W_{k,j} \cdot (F_{GAH})_j + b_k \quad (5)$$

$$(F_{GMF})_k = \sum_{j=1}^n W_{k,j} \cdot (F_{GMH})_j + b_k \quad (6)$$

The outputs F_{AF} and F_{MF} are added element wise and then sigmoid is used as activation function to obtain F_{OP} and is given by:

$$F_{OP} = \sigma(F_{GAF} + F_{GMF}) \quad (7)$$

The feature map 'F' is then re-scaled with the output F_{OP} from the channel attention module. Re-scaling is done through channel-wise multiplication as:

$$F_{\text{new}} = F_{\text{Op}} \cdot F \quad (8)$$

where F_{new} is the re-scaled feature map that can be passed to the next layer as input. The block diagram in figure 1 visually illustrates the layers used in the channel attention module [18].

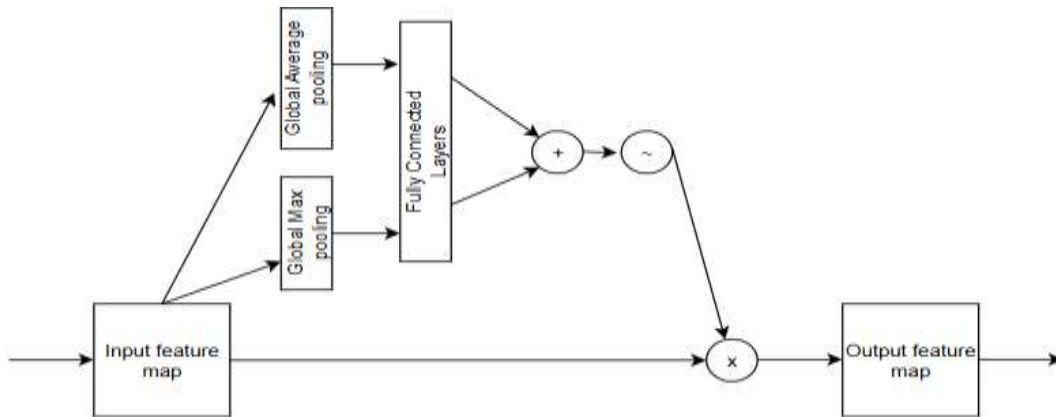


Figure1: Block diagram of channel attention module.

PROPOSED CHANNEL ATTENTION

We modified the channel attention mechanism in three different ways and evaluated how each variation affected its performance.

MODIFIED CHANNEL ATTENTION 1

The attention module discussed above employs global average pooling and global max pooling to squeeze the input. But this may lead to important information loss as the entire feature map is reduced to a single value. Missing out on important information while calculating the weights can lead to degraded performance [30,31]. So, in the proposed module global average pooling and global max pooling layers were replaced by average pooling and max pooling layers, as shown in figure 2, to obtain the outputs $F_A \in \mathbb{R}^{C \times H \times W}$ and $F_M \in \mathbb{R}^{C \times H \times W}$ respectively from the input $F \in \mathbb{R}^{C \times H \times W}$ and is given by:

$$(F_A)_{i,j} = \frac{1}{|w|} \sum_{(m,n) \in w} F_{(s,i+m,s,j+n)} \quad (9)$$

$$(F_M)_{i,j} = \max_{(m,n) \in w} F_{(s,i+m,s,j+n)} \quad (10)$$

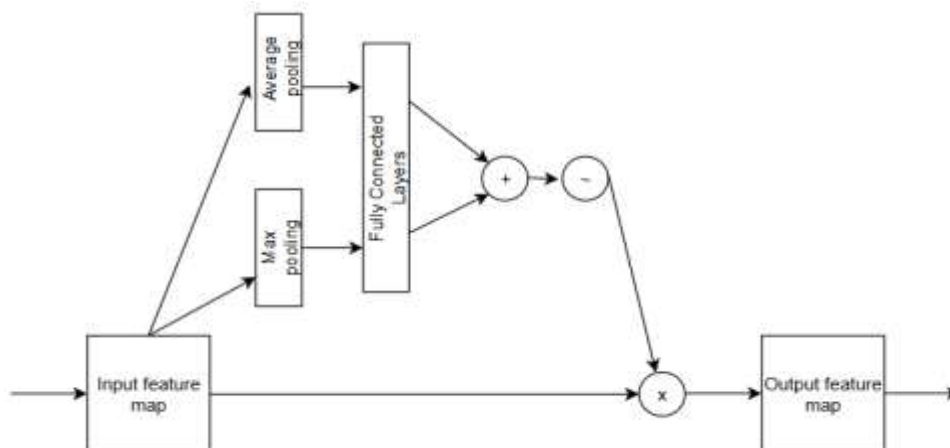


Figure 2: Block diagram of Modified Channel Attention 1.

MODIFIED CHANNEL ATTENTION 2

Here we have added global max pooling after average pooling and global average pooling after max pooling as shown in figure 3. The idea behind this sequence is that in average pooling followed by global max pooling, the noise is removed first and then the most prominent features were selected. While in max pooling followed by global average pooling, the most prominent features were selected first and then smooths them out [32].

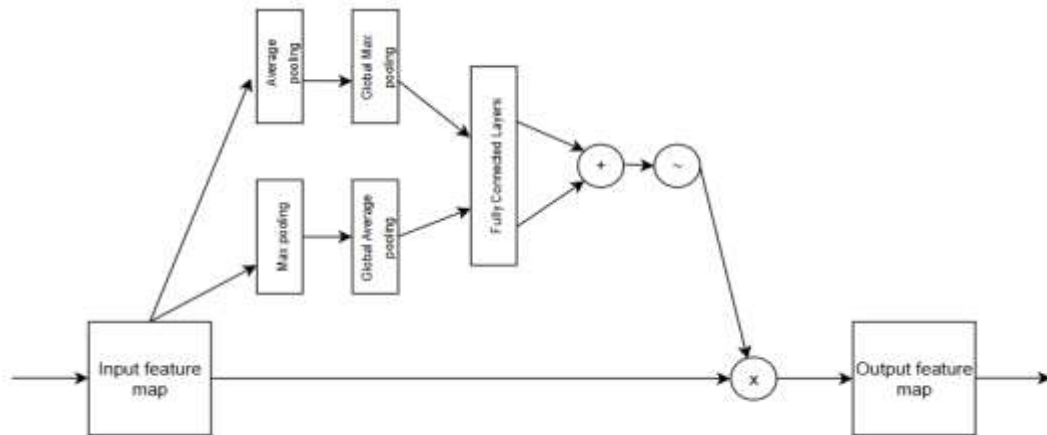


Figure 3: Block diagram of Modified Channel Attention 2.

MODIFIED CHANNEL ATTENTION 3

We have also performed a study by adding a global convolutional layer [33] instead of pooling layer as shown in figure 4. The idea behind using convolution layer is that it has the ability to capture meaningful information compared to pooling layers [34]. Global convolutional layer has the ability to create a compact channel descriptor.

$$(F_{\text{conv}})_{c_{\text{out}}} = \sum_{c_{\text{in}}=0}^{c_{\text{in}}-1} \sum_{i=0}^{H-1} \sum_{j=0}^{W-1} (F_A)_{c_{\text{in}},i,j} \cdot K_{c_{\text{out}},c_{\text{in}},i,j} \quad (11)$$

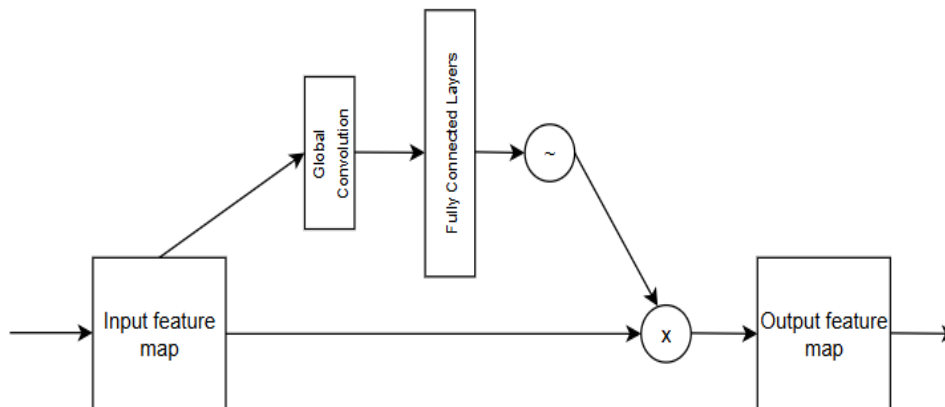


Figure 4: Block diagram of Modified Channel Attention 3.

EXPERIMENTAL SETUP

DATASET DESCRIPTION

To conduct the experiments, images of diseased tea leaves were sourced from online repositories¹. The dataset contains images of both healthy and diseased tea leaves, featuring conditions such as red spot, gray blight, brown blight, helopeltis, and algal spot. The dataset contains 1000 images of red spot, gray blight, helopeltis, algal spot, and healthy tea leaves and 867 images of brown blight tea leaves.

¹ <https://www.kaggle.com/datasets/saikatdatta1994/tea-leaf-disease>

IMPLEMENTATION

We used the Keras API to implement our work in Python. The Google Colaboratory (commonly known as Google Colab) environment, equipped with L4 GPU, was used to perform the experiments. The dataset was grouped into three categories: training, validation and testing. The models were trained using the training set. The validation and testing sets were used to validate and test the trained models. The training set contained 800 images of each for red spot, gray blight, helopeltis, algal spot and healthy tea leaves and 667 images of brown blight tea leaves. The validation and testing sets contained 100 image samples each. The input images were scaled to (224,224,3). Each model was trained for 100 epoch. When the model's validation accuracy was at its peak during training, model checkpoints were employed to save the weights. The training vs validation loss for each epoch was plotted. The transfer learning model of MobileNetV2 was used to mitigate the problem of insufficient samples for training deep neural networks. The fully connected layers contained a layer of 1000 neurons with ReLU as the activation function followed by an output layer of 6 neurons with SoftMax as the activation function. Image samples can be categorized by this output layer into six different classes: helopeltis, algal spot, brown blight, red spot, gray blight, and healthy tea leaves. The CNN models utilize Categorical Cross Entropy as their loss function. The optimizer used is Adam. To expand the training dataset, image augmentation techniques were employed to generate additional sample images. Four models were trained and tested. Each model had a different attention module. To evaluate the performance of the models, standard classification metrics such as accuracy, precision, recall, and F1-score were used [35]. The block diagram in figure 5 shows the process flow of the experiments performed.

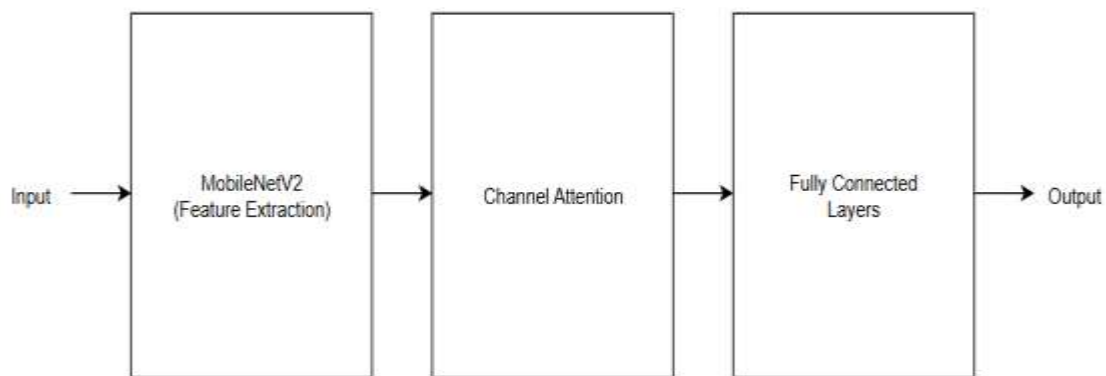


Figure 5: Block diagram representing the process flow of the experiments performed.

RESULTS

Figure 6, 7 and 8 present the performance metric for training, validation and testing phases, respectively. The model with Modified Channel Attention 3 had the best performance of all with testing precision and testing recall value of 95.46% and 94.83% respectively. This performance can be attributed to the global convolutional layer. The global convolutional layer can generate a single channel descriptor with more useful information in comparison to a global pooling layer. It has the ability to capture spatial information which is lacked by pooling layers. So, replacing pooling layers with global convolution layers has made the channel attention module more effective in generating channel-wise weights for the input feature map. Figure 9, 10, 11 and 12 presents the training and validation loss curve for the experimented models.

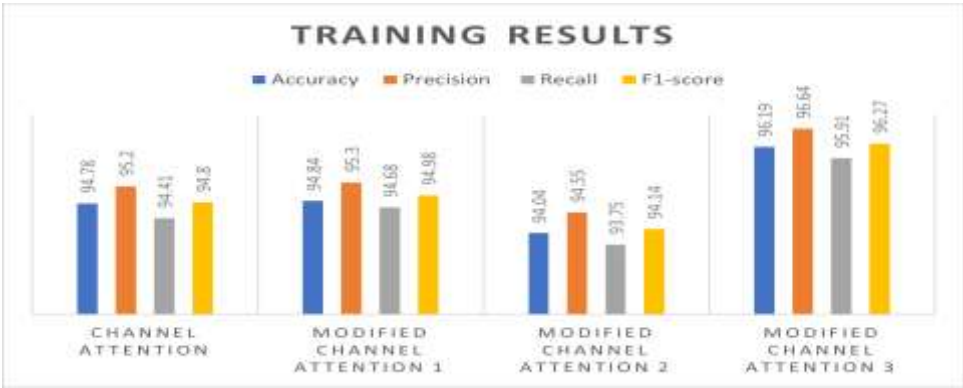


Figure 6: Training results of the models. All performance metrics are given in percentage (%).

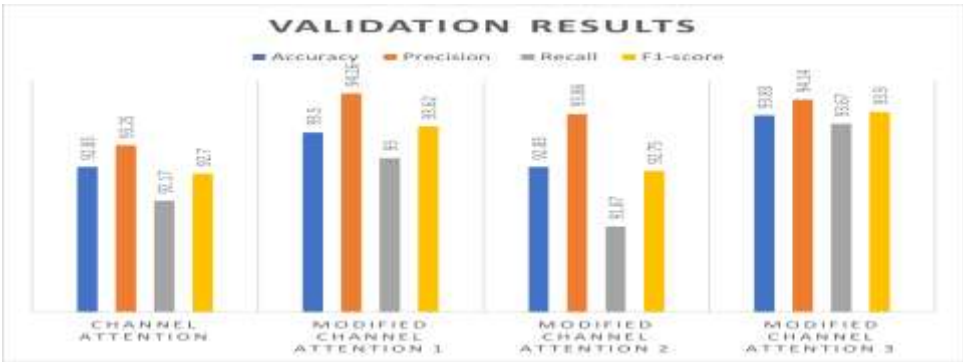


Figure 7: Validation results of the models. All performance metrics are given in percentage (%).

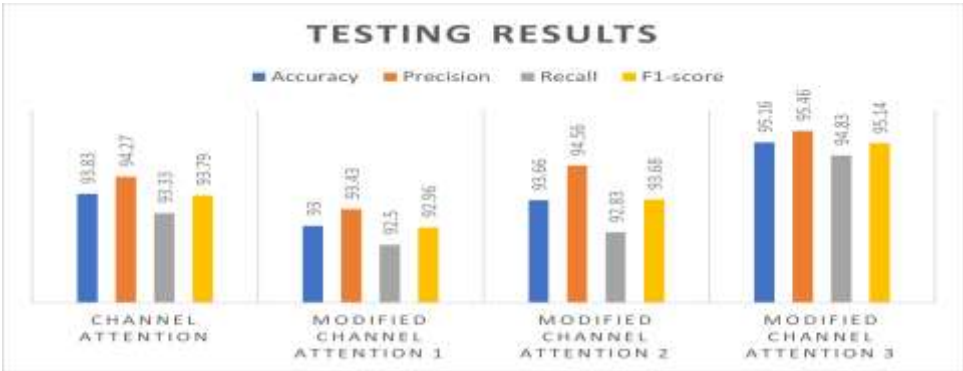


Figure 8: Testing results of the models. All performance metrics are given in percentage (%).

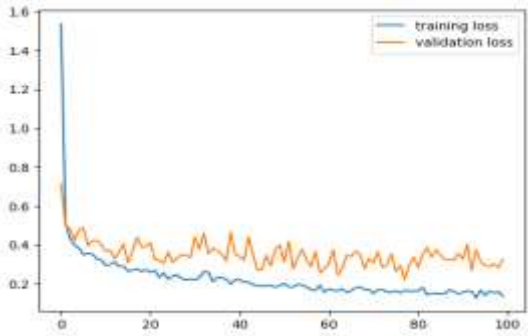


Figure 9: Training and validation loss curve for channel attention.

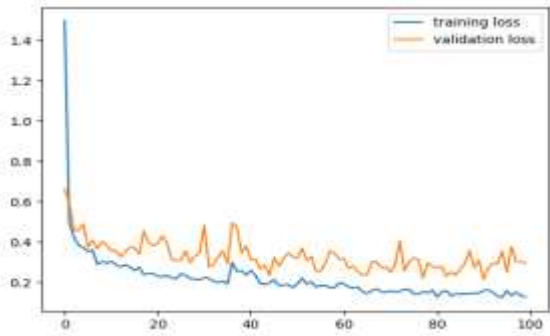


Figure 10: Training and validation loss curve for modified channel attention 1.

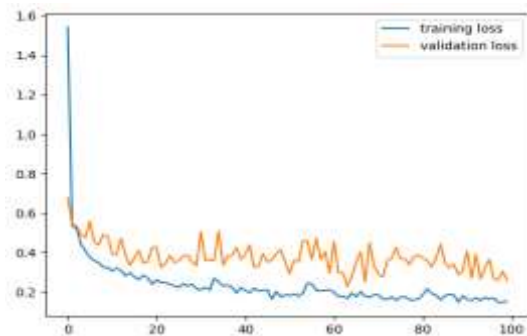


Figure 11: Training and validation loss curve for modified channel attention 2.

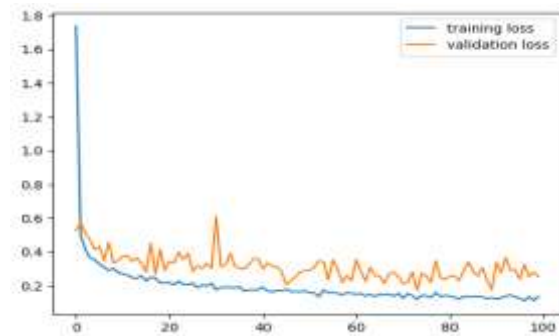


Figure 12: Training and validation loss curve for modified channel attention 3.

CONCLUSION

Attention module's ability to focus on the most relevant information helps to improve the performance of deep neural networks. This study focused on improving the channel attention mechanism for computer vision tasks. The proposed channel attention module demonstrated improved network performance compared to the channel attention proposed by Woo et al. [18]. The performance was validated empirically. As a future study, our proposed module can be tested with other network architectures configured in different patterns. Exploring these setups will demonstrate the effectiveness and reliability of the proposed architecture.

REFERENCES

1. Research and Markets, 2024, Tea Market Size and Share Analysis – Growth Trends and Forecast Report 2025–2033, <https://www.researchandmarkets.com/reports/5806952/tea-market-report>.
2. Kumarihami, P. C., H. H. M., and Song, K. J., 2018, "Review on Challenges and Opportunities in Global Tea Industry," *The Korean Tea Society*, 24(3), pp. 79–87.
3. Borah, A., Hazarika, S. N., and Thakur, D., 2022, "Potentiality of Actinobacteria to Combat Against Biotic and Abiotic Stresses in Tea [*Camellia sinensis* (L) O. Kuntze]," *Journal of Applied Microbiology*, 133(4), pp. 2314–2330.
4. Mahlein, A. K., 2016, "Plant Disease Detection by Imaging Sensors – Parallels and Specific Demands for Precision Agriculture and Plant Phenotyping," *Plant Disease*, 100(2), pp. 241–251.
5. Abade, A., Ferreira, P. A., and de Barros Vidal, F., 2021, "Plant Diseases Recognition on Images Using Convolutional Neural Networks: A Systematic Review," *Computers and Electronics in Agriculture*, 185, p. 106125.
6. Mohanty, S. P., Hughes, D. P., and Salathé, M., 2016, "Using Deep Learning for Image Based Plant Disease Detection," *Frontiers in Plant Science*, 7, Article 1419.
7. Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., ... and Farhan, L., 2021, "Review of Deep Learning: Concepts, CNN Architectures, Challenges, Applications, Future Directions," *Journal of Big Data*, 8(1), p. 53.
8. Krichen, M., 2023, "Convolutional Neural Networks: A Survey," *Computers*, 12(8), p. 151.
9. Ioffe, S., and Szegedy, C., 2015, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," *Proc. Int. Conf. Machine Learning (ICML)*, pp. 448–456, PMLR.
10. Li, Z., Liu, F., Yang, W., Peng, S., and Zhou, J., 2021, "A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects," *IEEE Trans. Neural Netw. Learn. Syst.*, 33(12), pp. 6999–7019.
11. Srinivas, S., Sarvadevabhatla, R. K., Mopuri, K. R., Prabhu, N., Kruthiventi, S. S., and Babu, R. V., 2016, "A Taxonomy of Deep Convolutional Neural Nets for Computer Vision," *Frontiers in Robotics and AI*, 2, Article 172022.
12. Guo, M. H., Xu, T. X., Liu, J. J., Liu, Z. N., Jiang, P. T., Mu, T. J., ... and Hu, S. M., 2022, "Attention Mechanisms in Computer Vision: A Survey," *Computational Visual Media*, 8(3), pp. 331–368.
13. Niu, Z., Zhong, G., and Yu, H., 2021, "A Review on the Attention Mechanism of Deep Learning," *Neurocomputing*, 452, pp. 48–62.

14. Wan, D., Lu, R., Shen, S., Xu, T., Lang, X., and Ren, Z., 2023, "Mixed Local Channel Attention for Object Detection," *Engineering Applications of Artificial Intelligence*, 123, p. 106442.
15. Awan, M. J., Masood, O. A., Mohammed, M. A., Yasin, A., Zain, A. M., Damaševičius, R., and Abdulkareem, K. H., 2021, "Image-Based Malware Classification Using VGG19 Network and Spatial Convolutional Attention," *Electronics*, 10(19), p. 2444.
16. Li, K., Wang, Y., Zhang, J., Gao, P., Song, G., Liu, Y., ... and Qiao, Y., 2023, "Uniformer: Unifying Convolution and Self-Attention for Visual Recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(10), pp. 12581–12600.
17. Hu, J., Shen, L., and Sun, G., 2018, "Squeeze-and-Excitation Networks," *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 7132–7141.
18. Woo, S., Park, J., Lee, J. Y., and Kweon, I. S., 2018, "CBAM: Convolutional Block Attention Module," *Proc. Eur. Conf. Comput. Vis. (ECCV)*, pp. 3–19.
19. Malik, M. M., Fayyaz, A. M., Yasmin, M., Abdulkadir, S. J., Al-Selwi, S. M., Raza, M., and Waheed, S., 2024, "A Novel Deep CNN Model With Entropy Coded Sine Cosine for Corn Disease Classification," *J. King Saud Univ. Comput. Inf. Sci.*, 36(7), p. 102126.
20. Chakrabarty, A., Ahmed, S. T., Islam, M. F. U., Aziz, S. M., and Maidin, S. S., 2024, "An Interpretable Fusion Model Integrating Lightweight CNN and Transformer Architectures for Rice Leaf Disease Identification," *Ecol. Inform.*, 82, p. 102718.
21. Madeira, M., Porfirio, R. P., Santos, P. A., and Madeira, R. N., 2024, "AI-Powered Solution for Plant Disease Detection in Viticulture," *Procedia Comput. Sci.*, 238, pp. 468–475.
22. Dharwadkar, N. V., Kalmani, V. H., and Thapa, V., 2023, "Crop Yield Prediction Using Deep Learning Algorithm Based on CNN-LSTM With Attention Layer and Skip Connection," unpublished.
23. Benfenati, A., Causin, P., Oberti, R., and Stefanello, G., 2023, "Unsupervised Deep Learning Techniques for Automatic Detection of Plant Diseases: Reducing the Need of Manual Labelling of Plant Images," *J. Math. Ind.*, 13(1), p. 5.
24. Yao, X., Lin, H., Bai, D., and Zhou, H., 2024, "A Small Target Tea Leaf Disease Detection Model Combined With Transfer Learning," *Forests*, 15(4), p. 591.
25. Datta, S., and Gupta, N., 2023, "A Novel Approach for the Detection of Tea Leaf Disease Using Deep Neural Network," *Procedia Comput. Sci.*, 218, pp. 2273–2286.
26. Sandler, M., et al., 2018, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 4510–4520.
27. Howard, A. G., et al., 2017, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," *arXiv preprint, arXiv:1704.04861*.
28. Chen, Q., and Yan, S., 2014, "Network in Network," *Proc. Int. Conf. Learn. Represent. (ICLR)*.
29. Oquab, M., Bottou, L., Laptev, I., and Sivic, J., 2015, "Is Object Localization for Free?—Weakly-Supervised Learning With Convolutional Neural Networks," *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 685–694.
30. Baozhou, Z., Hofstee, P., Lee, J., and Al-Ars, Z., 2021, "An Attention Module for Convolutional Neural Networks," *arXiv preprint, arXiv:2108.08205*.
31. Roy, A. G., Navab, N., and Wachinger, C., 2018, "Concurrent Spatial and Channel 'Squeeze & Excitation' in Fully Convolutional Networks," *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Interv.*, pp. 421–429, Springer, Cham.
32. Zafar, A., Aamir, M., Mohd Naw, N., Arshad, A., Riaz, S., Alruban, A., ... and Almotairi, S., 2022, "A Comparison of Pooling Methods for Convolutional Neural Networks," *Appl. Sci.*, 12(17), p. 8643.
33. Peng, C., Zhang, X., Yu, G., Luo, G., and Sun, J., 2017, "Large Kernel Matters—Improve Semantic Segmentation by Global Convolutional Network," *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 4353–4361.
34. Zhao, L., and Zhang, Z., 2024, "An Improved Pooling Method for Convolutional Neural Networks," *Sci. Rep.*, 14(1), p. 1589.
35. Grandini, M., Bagli, E., and Visani, G., 2020, "Metrics for Multi-Class Classification: An Overview," *arXiv preprint, arXiv:2008.05756*.