Heliyon 10 (2024) e29438

Contents lists available at ScienceDirect

Heliyon

journal homepage: www.cell.com/heliyon

Research article

Enhanced deep learning technique for sugarcane leaf disease classification and mobile application integration

Swapnil Dadabhau Daphal [a,*], Sanjay M. Koli [b]

[a] Department of E&TC Engineering, G. H. Raisoni College of Engineering & Management, Wagholi, Pune 412207, Maharashtra, India
[b] Department of E&TC Engineering, Ajeenkya DY Patil School of Engineering, Charholi Bk., Pune 412105, Maharashtra, India

A R T I C L E I N F O

Dataset link: https://data.mendeley.com/datasets/9424skmnrk/1

Keywords:
Deep learning
Disease classification
Agriculture
Sugarcane database

A B S T R A C T

With an emphasis on classifying diseases of sugarcane leaves, this research suggests an attention-based multilevel deep learning architecture for reliably classifying plant diseases. The suggested architecture comprises spatial and channel attention for saliency detection and blends features from lower to higher levels. On a self-created database, the model outperformed cutting-edge models like VGG19, ResNet50, XceptionNet, and EfficientNet_B7 with an accuracy of 86.53%. The findings show how essential all-level characteristics are for categorizing images and how they can improve efficiency even with tiny databases. The suggested architecture has the potential to support the early detection and diagnosis of plant diseases, enabling fast crop damage mitigation. Additionally, the implementation of the proposed AMRCNN model in the Android phone-based application gives an opportunity for the widespread use of mobile phones in the classification of sugarcane diseases.

1. Introduction

With the rapid developments in digital technologies the acquisition of scene images is more convenient. It has been the primary reason of genesis of real time system based on smart phones. It has almost covered all sectors including food, health, finance, manufacturing, production etc. Agriculture remains no different this development. Smart cultivation practices, plant growth monitoring, plant harvesting etc. are major areas where the influence of digitization can be clearly seen. However, Plant diseases have a deteriorating impact of agricultural yields and remain key threat to global food security. It is essential to detect and use control remedies to mitigate the harmful effect of the diseases. Early detection of the plant diseases can significantly help to bring these losses in control [1]. With pool of digital information, farmers can approach digital centers and seek the help of plant pathologist and experts to get the correct remedies. However, manual detection of disease is erroneous and time consuming. Many researchers have explored the way to automatically detect the plant disease based on image processing techniques combined with machine learning. Improvement in the classification accuracy and reduction in the latency in the diagnosis were key objectives in these techniques [2–4]. To ensure better diagnosis results image manipulation techniques [5], dimensionality reductions [6], and fuzzy system were employed. In recent years, deep learning techniques have expanded their reach to many applications. Plant disease diagnosis systems have been in discussion for a long and are considered a standard method for the plant disease detection and classification.

* Corresponding author.
E-mail address: daphalsd01@gmail.com (S.D. Daphal).

https://doi.org/10.1016/j.heliyon.2024.e29438
Received 20 January 2024; Received in revised form 8 April 2024; Accepted 8 April 2024

Available online 12 April 2024

The progress in deep learning has made great achievements in the classification of the images with its feature extraction and efficient learning ability [7,8]. It has been a trend to apply deep learning based convolutional neural networks to computer vision task. To handle more specific use cases SegNet [9], representative networks like feed-forward convolutional network (FCN) [10] and U-Net [11] have been suggested and studied. These deep learning based approaches have surpassed the traditional approaches in terms of end-to-end learning, reduce losses, saving manpower and material resources. It is also true that with the increase in the depth of the CNN are more prone to the well known problem of vanishing gradient. To address this issue residual network was discussed in [12], that utilized short connection to link input of the residual network to the output. This network offered the reduced complexity and overcome the problem of vanishing gradient. A large and well organized database is equally important for the functioning of deep learning architecture. Large open source data-set were used in experimentation for disease diagnosis of plants like grape, apple and paddy [13–15]. An accuracy of 85% was recorded in those experiments. A plant village data-set was used, which comprised of images taken in laboratory environment belonging to 26 different classes. As these images were taken in constrained environment discussion on its portability to real time scenario were mentioned in [16]. In some situations database collection is non-trivial task and training a network with small database becomes challenging, here transfer learning based methods were used effectively for disease classification [17–19].

Regardless of challenges, it is quest to build a model with improved classification performance and less overhead. In [20] architectural modifications using residual network leveraged to perform superior than convolutional network for scene classification. Useful features were obtained more accurately by combining channel attention with residual network. By adding convolutional layer in the output the weights of the feature map were adjusted to focus on important features of the images [21,22]. Also studies in [23,24] have further highlighted the importance of attention mechanism in deep learning techniques. (Pyramid level attention and feature extraction should be discussed.) To summarize, to make image classification more robust and invariant to noise, attention mechanism can significantly help. Additionally, real time applications largely demands specific data-sets and generating such data is again big challenges. The database created in controlled environment fare well in terms of laboratory statistics but fails majorly in real time use cases [25]. To address these challenges database having real time field images have been included in this study. Attempts have been made to maintain variety in terms of orientation, distance, illumination and other environmental condition while collecting the images. The major contributions of this paper are as follows,

1. A general approach called attention-based multi-level residual convolutional neural network (AMRCNN) using features from multiple levels of the network is proposed to generate a large amount of information for sugarcane disease recognition and keep the model less complex for real-time processing. Additionally, the suggested architecture's attention mechanism eliminates unwanted features.
2. The proposed framework aims to improve the classification of plant leaf diseases by using a residual structure with an integrated attention block.
3. An Android based mobile application is developed to integrate proposed model in real time test scenario.

The remaining paper is organized as follows: Section 2 gives the detailed information about proposed architectures and its various modules. Experimental results mainly focusing on database, evaluation metrics, hyper parameter and results are discussed in section 3. The conclusion and findings are discussed in section 4.

2. Proposed method

In this section, a concise overview of the network architecture of the proposed method is presented, prior to delving into its crucial components. The proposed method for extracting deep features and classifying sugarcane diseases is the AMRCNN. Each component of the proposed method will be meticulously described in subsequent subsections.

2.1. Dataset

The study gathered photos of sugarcane leaves in various cultivated fields with various weather conditions. Five classes, including four classes of sick sugarcane leaves and one class of healthy sugarcane leaf samples, made up the initial dataset for sugarcane leaf disease. There were roughly 500 photos in each class, which is a small number. The sugarcane database, which shows the class-wise distribution of the picture samples, is shown in Table 1. The original photos were downsized to 256x256 to aid computations. For improved performance, the dataset was split into training, validation, and test sets in a 70:15:15 ratio. Such splitting allowed for a sufficient number of images in each category. We created a set of images for training, testing, and validation, maintaining consistency throughout all experimental variations, which facilitated easier interpretation of results. The training and validation datasets were used during training, and the test dataset was used to assess the model. A sample from each class is shown in Fig. 1, and a summary of the database's statistics is given in Table 1. The database is made available freely to research community [26]. All experimentation in the subsequent section is carried out without any augmentation to establish the baseline for the model and database. Fig. 1(a) shows the sample image from rust class, Fig. 1(b) represents mosaic, Fig. 1(c) healthy, Fig. 1(d) red rot and Fig. 1(e) shows the image associated with yellow class.

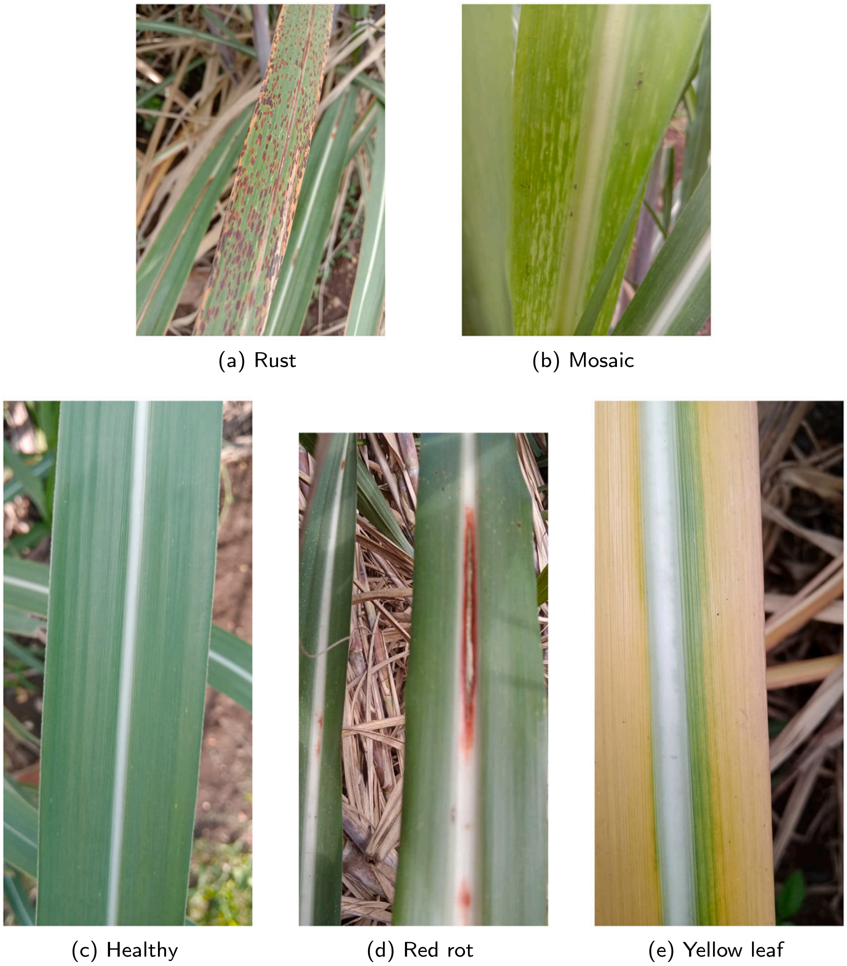

Fig. 1. Sample images from database for each class: (1a) Rust, (1b) Mosaic, (1c) Healthy, (1d) Red rot, (1e) Yellow leaf.

Table 1
Dataset statistics: Originally collected.

Class_Name	Total_Images	in %
Sugarcane_Healthy	520	20.2%
Sugarcane_Rust	514	20.0%
Sugarcane_Red rot	519	20.2%
Sugarcane_Yellow	505	19.7%
Sugarcane_Mosaic	511	19.9%
Total	2569	100%

2.2. Attention based multi-level residual convolutional neural network

By stacking convolutional layers, conventional convolutional neural networks (CNN) enhance the performance of image classification while adding more parameters and computational complexity. A feature extraction module made up of attention-based residual blocks from the standpoint of network width to enhance the performance of image classification has been proposed. The proposed method's architectural layout is shown in Fig. 2.

The proposed architecture, as depicted in Fig. 2, considers both mid-level features and high-level features when classifying data. Each level of the network's residual blocks extracts these feature maps. Because the first residual block contains simple filters for sugarcane disease classification, the information from that block is not immediately utilized. It is commonly known that the deep CNN's early layers can extract low-level features or the basic aspects of an image's structure. The networks at this stage can only learn and interpret simple textures, which are insufficient to convey the fundamental and significant characteristics of sugarcane disease. Low level features are captured by initial layers of the network and roughly include edges, corners and color patches. The mid-level features are patterns learned by the middle of network and have textures, object shapes. High level features helps in making more

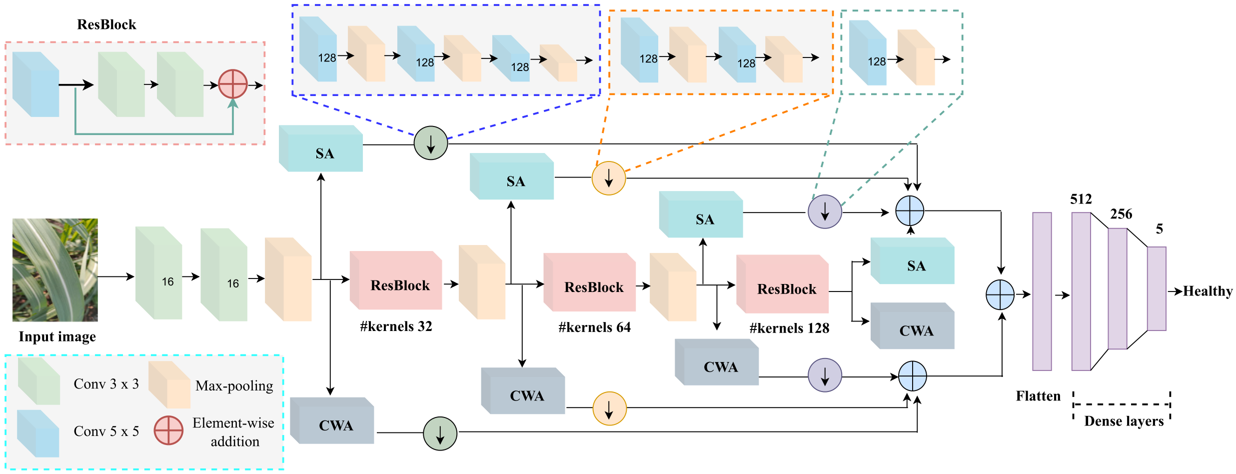

Fig. 2. Detailed description of proposed AMRCNN.

clear scenario present in the image. On the other hand, by merging learnt features from the preceding layers, midlevel and high-level features (from the second ResBlock to the third ResBlock) are made more complex.

Furthermore, the third ResBlock's mid-level layers produce local features, whilst the third ResBlock's high-level layers are crucial for identifying sugarcane disease through global features. Local features may include keypoints, descriptors and some inherent pattern similar to local binary patterns. Global features may include moments and activations. The global characteristics of a CNN, or the entire image characteristic, are typically used by other works for classification. This study's suggested AMRCNN recognizes sugarcane disease by carefully combining local and global features. This mechanism helps to focus on important parts of the input image. Unlike conventional CNNs, our networks provide a large number of essential properties that make the system robust to several sugarcane diseases and difficulties with image quality, respectively.

The incorporation of each block in the proposed method plays a vital role in reducing the loss of partial feature information related to plant diseases and increasing recognition rates, as shown in Fig. 2. To address issues of gradient disappearance and model optimization difficulties associated with a large number of convolution layers, a residual structure has been employed. The residual structure is advantageous because it can address the problem of increasing network depth degradation. The mapping relationship between the residual structure's convolutional layers is $H(x) = F(x) + x$, which can restore the network to its optimal state with a small update to some of the weight values.

Sugarcane disease spots possess high inter-class variations and low intra-class variances, leading to confusion when distinguishing different types of samples in a deep CNN. Therefore, we propose embedding the modified attention mechanism after each residual block of the network to accurately locate the disease spot's location in the image and improve the recognition accuracy of the model. By doing so, the impact of redundant information in the input image on the network classification performance can be reduced. The suggested method also incorporates a channel and spatial attention module, which can be easily integrated into the proposed network. The subsequent subsections provide an in-depth explanation of the attention mechanism.

Finally, combined the attentive features of three blocks using element-wise addition operation and given to fully-connected layers as input for further feature extraction and classification. To combine each level of deep attentive features, it is required to maintain the same size of feature maps. To maintain the consistent size of each block attentive features, a downsampling method is applied with the help of convolutional and max-pooling operation to extract more intuitive features relevant to sugarcane disease. The architecture of the downsampling method at three levels is shown in Fig. 2. Two fully-connected layers are employed in the proposed architecture, with the size of 512 and 256. The last layer of AMRCNN is softmax with 5 classes.

2.3. *Attention mechanism*

The ability to pay attention is an important feature of the human visual system (HVS). HVS does not process the entire scene all at once. Despite this, it focuses on key locations to acquire more specific feature information from the scene and generate helpful conclusions. In this work, channel attention and spatial attention used efficiently for saliency detection.

2.3.1. *Channel attention*

By taking into account the inter-channel relationship of feature space, a channel attention feature map is created. Every channel in the feature space is thought of as a feature detector, and the channel's attention determines "what" is significant in the scene. The feature space's spatial dimension must be squeezed for the effective calculation of the channel attention. Average pooling and maximum pooling are used in tandem to aggregate spatial information. By this method, networks' representational power is significantly increased [27]. Two different spatial descriptors $F_{max}^{channel}$ and $F_{avg}^{channel}$ are generated that represents features obtained by max pooling and average pooling, respectively. Those descriptors are passed to a common multi-layer perceptron network to

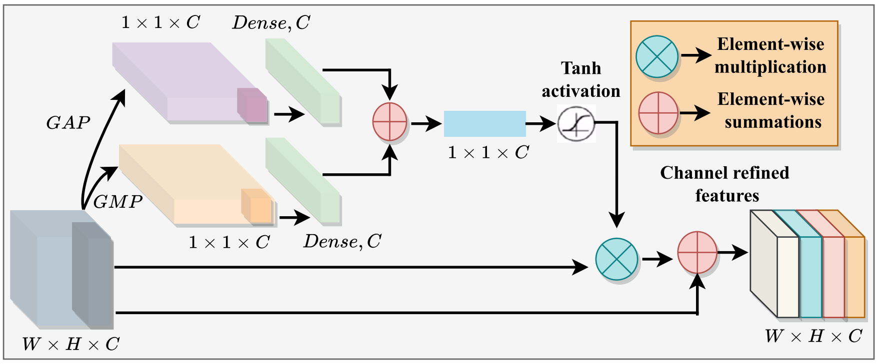

Fig. 3. Channel wise attention.

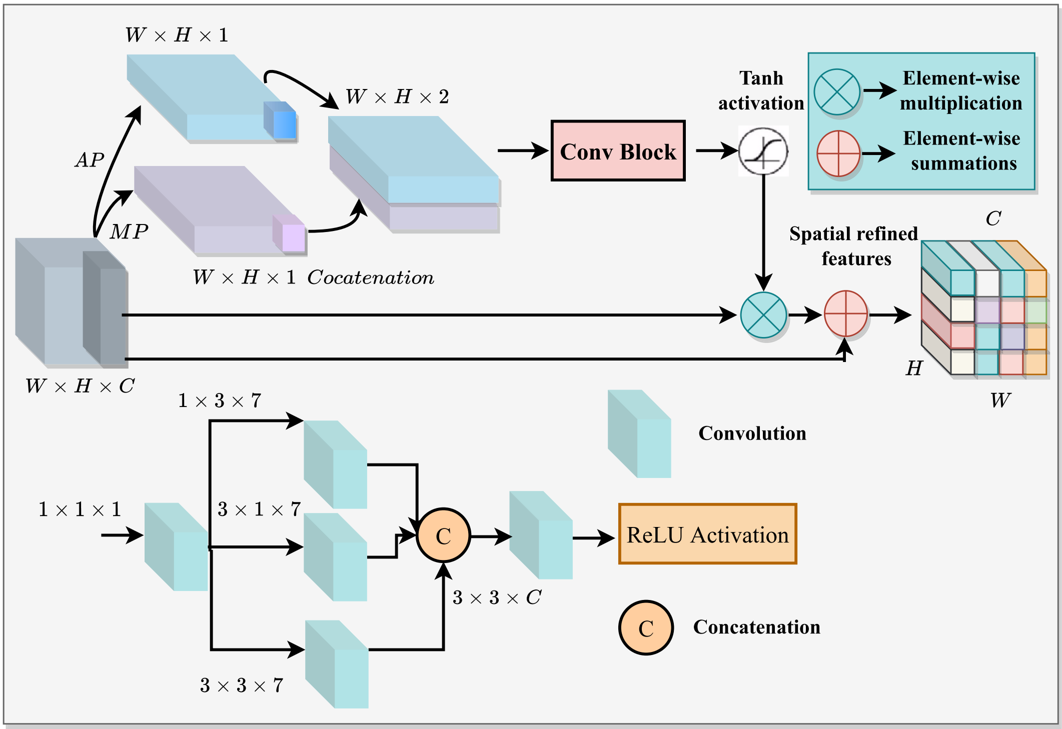

Fig. 4. Spatial attention.

produce channel attention map $Map_{channel} \in R^{C\times1\times1}$. Parameter overhead reduction is achieved by setting hidden activation size to $R^{\frac{C}{r}\times1\times1}$, r is reduction factor. The outcomes of Multi-layer perceptron network are merged to produce output features by using element wise summation. The mathematical calculation for the channel attention map is given in equation (1),

$$\begin{aligned} Map_c(F) &= sigmoid(MLP(AveragePool(F)) + MLP(MaxPool(F))) \\ &= sigmoid(Wt_1(Wt_0(F^{channel}_{\text{average}})) + Wt_1(Wt_0(F^{channel}_{\text{max}}))) \end{aligned} \tag{1}$$

where, $Wt_0 \in R^{\frac{C}{r}\times C}$ and $Wt_1 \in R^{C\times\frac{C}{r}}$, weights are shared by inputs and the ReLU activation and then Wt_0. Fig. 3, shows the channel wise attention process in details.

2.3.2. Spatial attention

Spatial attention, as opposed to channel attention, is focused on the informational elements of the image. To create an effective feature descriptor for the computation of spatial attention, first, perform average-pooling and max-pooling operations along the channel axis. It has been proven that focusing on interesting regions is possible by employing pooling techniques along the channel axis. To automatically differentiate the relative relevance of various regions of the feature map, the spatial attention module, as illustrated in Fig. 4, is built to adaptively boost the information in the features that contribute to the classification and suppress

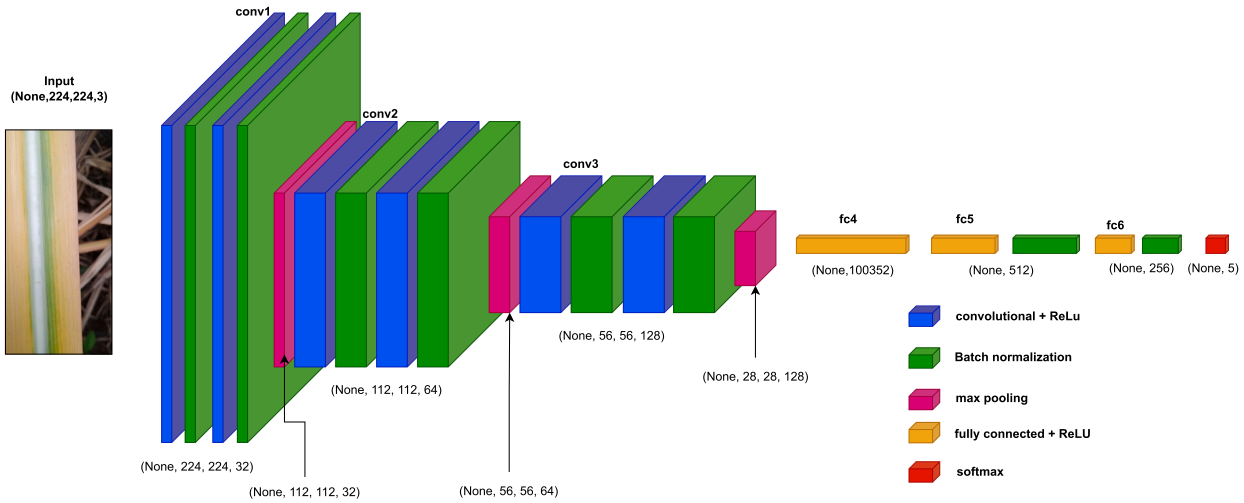

Fig. 5. When using Android Studio, this simple CNN was used for comparison purposes during the experiment.

the regions that are less important. The spatial attention module is made up of 4 convolutional layers and an activation function. Convolutional layers are capable of integrating data from several channels, normalizing it using the activation function to create a set of weights, and multiplying it proportionally with the input feature map to learn the meaning of each coordinate point in the feature space on-the-fly.

A convolution is applied on concatenated features to generate a spatial attention map $Map_s(F) \in R^{H\times W}$ which decides where to concentrate in scene. Two 2D maps, average pooled features ($F_{avg}^{spatial} \in R^{1\times H\times W}$) and spatially pooled features ($F_{max}^{spatial} \in R^{1\times H\times W}$) are generated that are applied on channel information of feature map. A convolution is applied on the concatenation of those features to produce 2D spatial attention map. In short, equation (2) explains the operation,

$$\begin{aligned} Map_s(F) &= sigmoid(f^{7\times 7}([AveragePool(F); MaxPool(F)])) \\ &= sigmoid(f^{7\times 7}([F_{\text{avg}}^{spatial}; F_{\text{max}}^{spatial}])) \end{aligned} \tag{2}$$

where, $f^{7\times 7}$ is a convolution with filter size of 7×7.

2.4. Parameters of AMRCNN

All important network architecture parameters were set to demonstrate that the network is ubiquitous, regardless of how individual traits will alter the network's ideal parameters.

- Adam optimization is chosen as the network's optimizer algorithm and the parameter is specified as the initial value.
- Categorical cross-entropy is chosen as a loss function for sugarcane classification problem.
- The learning rate is set as 0.0001.
- In each convolutional and fully-connected layer, LeakyReLU is employed as an activation function. Alpha value was set as 0.02.

2.5. Proposed real time system

A fast-expanding field of research uses machine learning and deep learning methods to produce mobile applications for the classification of crop diseases. TensorFlow Lite, a lightweight version of the noted machine learning framework TensorFlow, is one of the most widely used tools for this purpose. In Android Studio, developers can use TensorFlow Lite as a backend to build effective and precise models for classifying various sugarcane diseases. The TFLite model file (.tflite) is added into android project folder. Given model is then loaded into android application using TFLite interpreter and prediction are made on the input data captured by mobile camera. Convolutional neural networks (CNNs) and other supervised learning algorithms are used to train the model in this manner, which also entails preparing the input data and deploying the model on resource-constrained mobile devices. Quantization strategies, which assist reduce the model's size and increase its speed, are only two of the tools that TensorFlow Lite offers to help you optimize the model's performance. The end result is a mobile app that makes it possible for growers and agronomists to rapidly and precisely identify sugarcane diseases, which can improve crop management and result in better yields and lower costs. A simple CNN model used in the study for comparative analysis is shown in Fig. 5. The brief view of proposed real time system using anroid application is outlined in Fig. 6.

The detailed operation stages of the Android application are explained in Fig. 7.

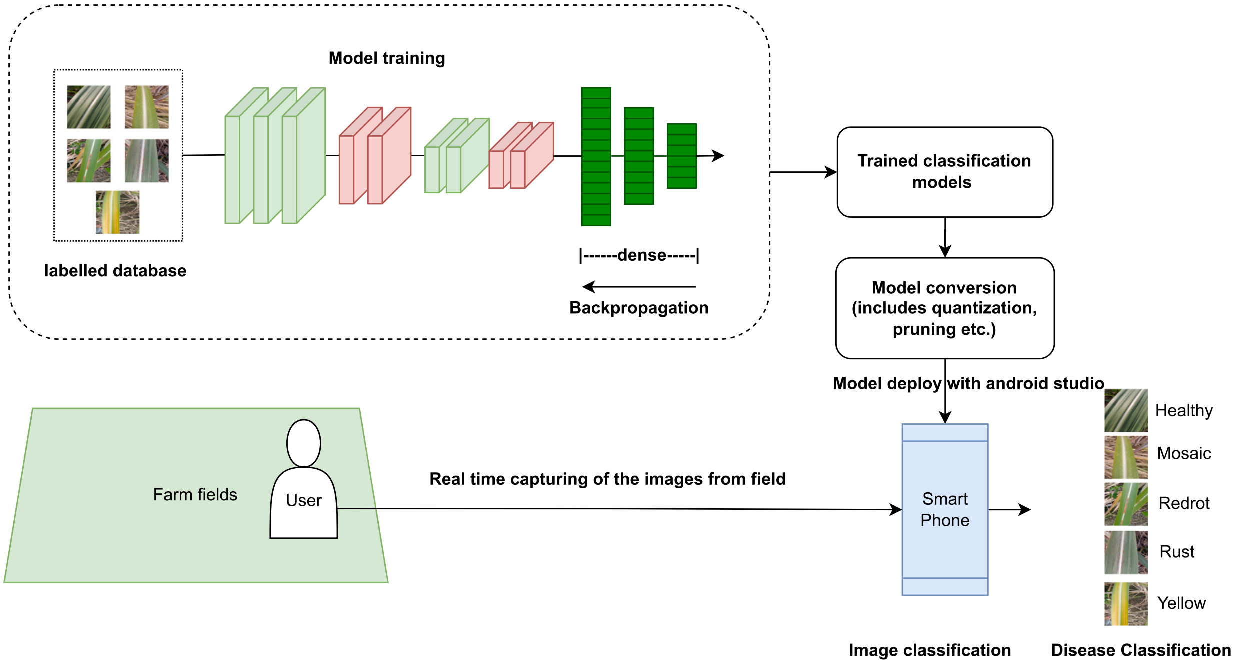

Fig. 6. Block diagram of the complete sugarcane plant disease diagnosis system.

3. Experiments

The experiments are performed with Google Colaboratory, Google Compute Engine, NVIDIA Tesla T4 and GDDR6 with 16 GB capacity. The proposed architecture is implemented using Tensorflow-Keras framework.

3.1. Evaluation metrics

The following major terms are determined from the obtained confusion matrix: The predicted value of the true positives (TP)-model matches the actual value; False positives (FP or type 1 error) occur when the actual value is negative while the model predicts a positive value; True negative (TN)-the actual negative value matches the predicted negative value of the model; False negative (FN) or Type-2 error - The model predicted a negative result while the actual value is positive. Mathematically, accuracy, precision, recall and F1 score are given by equation (3), (4), (5), and (6), respectively.

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \tag{3}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{4}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{5}$$

$$\text{F-1 Score} = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{6}$$

3.2. Hyperparameter selection & other details

The overall sugarcane leaf images are divided into 5 classes while training the network. All models were trained with following details.

- **Framework:** Tensorflow 2.0
- **Training stop criteria:** EarlyStopping (Patience = 20)
- **Model Checkpoint:** Save Best only
- **Other callbacks:** CSVLogger
- **Optimizer:** Adam
- **Epochs:** 30
- **Input image size:** 256x256
- **batch size:** 8

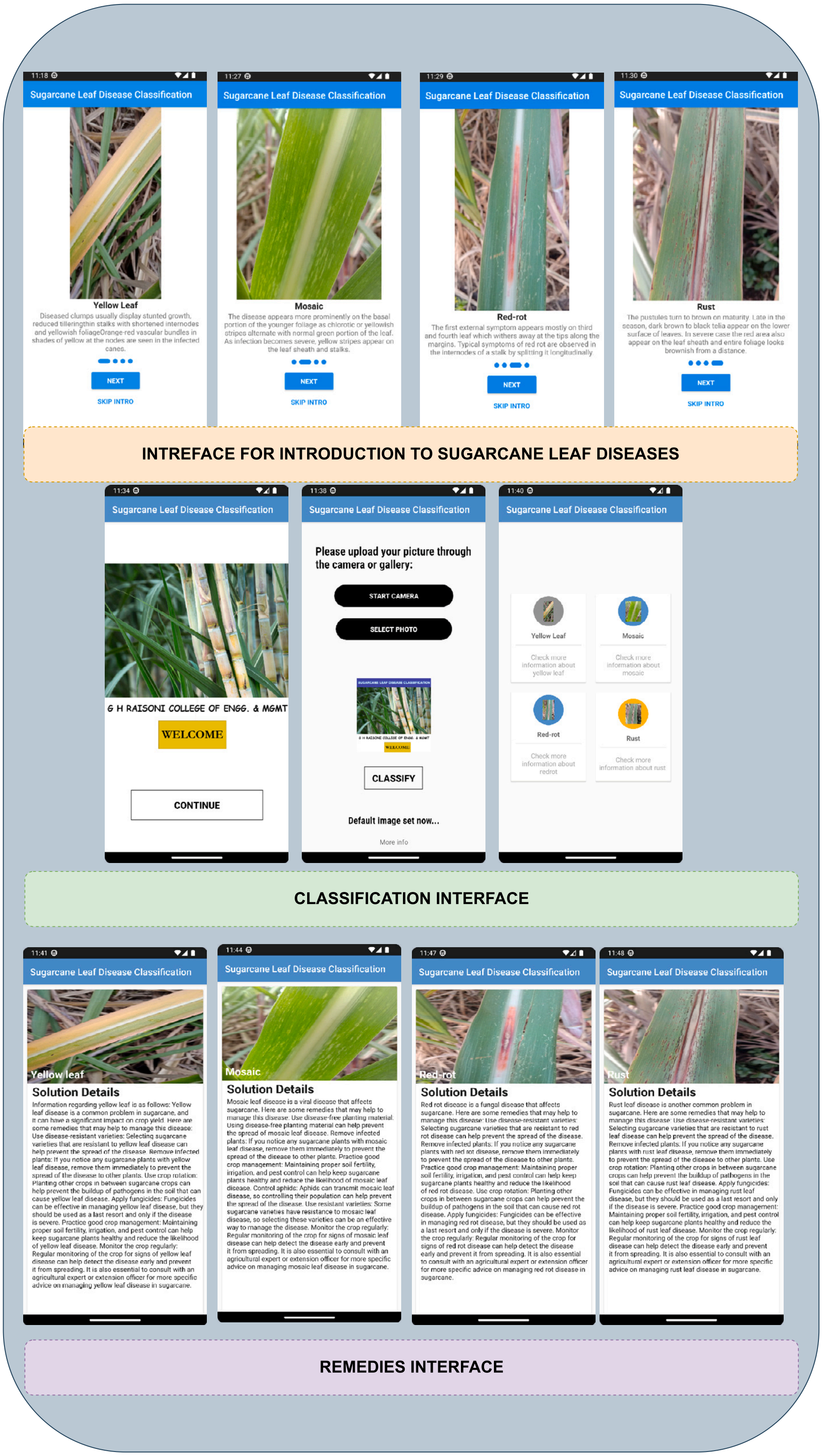

Fig. 7. Overall Operation of application.

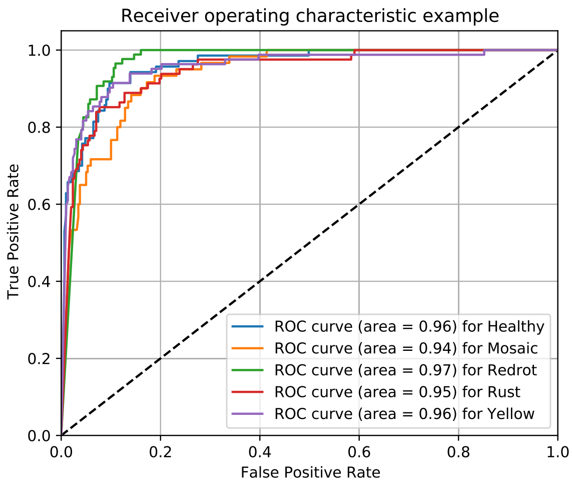

Fig. 8. ROC curves with AMRCNN.

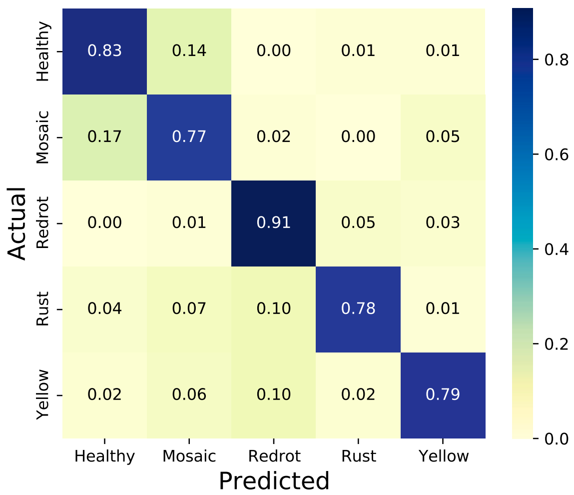

Fig. 9. Confusion matrix for AMRCNN.

- **Learning rate:** 0.0001
- **learning rate decay:** $1 \times e^{-6}$

3.3. Evaluation results and discussion

The proposed AMRCNN model is compared with several existing benchmark deep learning architectures. Four well-known deep-learning architectures were used for the comparison, including VGGNet, ResNet50-v2, XceptionNet, and EfficientNet-B7. A transfer learning-based approach was used to train the initial layers of these architectures and later fine-tuned by a custom sugarcane leaf disease database. The performance of these models is given in Table 2. ResNet achieves the best performance compared to the other deep learning architectures due to its deep interpretations and effective handling of vanishing gradients. Despite this, the proposed AMRCNN architecture performs better than the baseline architectures. Fig. 8, shows the receiver operating curve which is commonly used in classification. It plots the true positive rate versus the false positive rate of the classifier. It is known that a perfect classifier will have ROC-area under a curve equal to 1, whereas a random classifier will have AUC equal to 0.5. It is evident from Fig. 8, AUC values are near 1, indicating AMRCNN is a good classifier. A confusion matrix is given in Fig. 9 for the AMRCNN classifier. Most of the images are on the principal diagonal which shows classification with the model is fairly good. This finding suggests that a pyramid-based network that integrates channel and spatial attention pays more attention to the salient part of the images and neglects the noise component. This network fares comparatively well even though the network is merely trained on small databases. It is worth noting that, to a certain extent required performances can be achieved with architectural modifications even if the database is small. Fig. 10, gives a pictorial comparison of all methods used in the experimentation and also highlights the slight improvement observed with the proposed architecture.

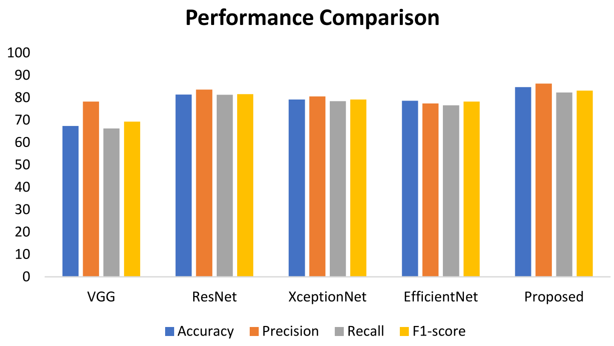

Fig. 10. Performance comparison of AMRCNN framework with various architectures.

Table 2
Comparison of AMRCNN framework results with different benchmark methodologies.

Model	Method	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
VGG [28]	**VGG19**	67.33	78.23	66.27	69.34
ResNet [29]	**ResNet50_v2**	81.42	83.64	81.33	81.61
XceptionNet [30]	**XceptionNet**	79.21	80.6	78.46	79.15
EfficientNet [31]	**EfficientNet_B7**	78.66	77.43	76.56	78.31
Proposed	**AMRCNN**	86.53	86.34	82.32	83.15

Table 3
Model quantization results and real time performance of proposed model against simple CNN.

model name	Converted to	size	accuracy	inference time in ms
simple CNN	Original	84 MB	78.22%	865
	float 32	46.59 MB	74.23%	655
	float 16	25.64 MB	72.44%	435
	int 8	6.34 MB	70.23%	375
Proposed model: AMRCNN	Original	153 MB	86.53%	920
	float 32	78.68 MB	84.65%	700
	float 16	38.67 MB	82.87%	524
	int 8	9.64 MB	81.22%	450

3.4. Model quantization and its performances

The general approach for quantization of models used in the work is as follows:

1. Models were trained using TensorFlow 2.0 API with the default float32 datatype.
2. Later, models were quantized to float16 using TensorFlow's API to explicitly cast tensors to float16.
3. TensorFlow's API was used to cast tensors to int8 to reduce the size further.

In the above process, the entire model was quantized with the aforementioned options instead of selective layers, and the quantization strategy was uniform across all layers. This process reduces the model size and may impact the performance of the model. A trade-off must be made between the desired performance and the required model size. Please refer to the TensorFlow documentation for further information [32].

Table 3 compares the real-time functionality and quantization outcomes of a simple CNN model, shown in Fig. 5, with a suggested model known as AMRCNN. The models' original sizes, accuracy, and inference times are shown in the table, along with their quantization sizes, accuracy, and inference times for float 32, float 16, and int 8 data types. The table demonstrates that at all quantization levels, the proposed AMRCNN model generally outperforms the simple CNN model in terms of accuracy and inference time. However, using data types with lower bit widths for quantization results in a smaller model and faster inference time but at the expense of accuracy.

3.5. Field test

The proposed system was tested in the field, where fresh images, not part of the already existing dataset, were captured with the help of a mobile camera. It is worth noting that during the database creation, the labels were assigned by a team of 5 people, including 3 expert farmers with over 10 years of experience in sugarcane cultivation, and 2 agriculture experts holding certified degrees in

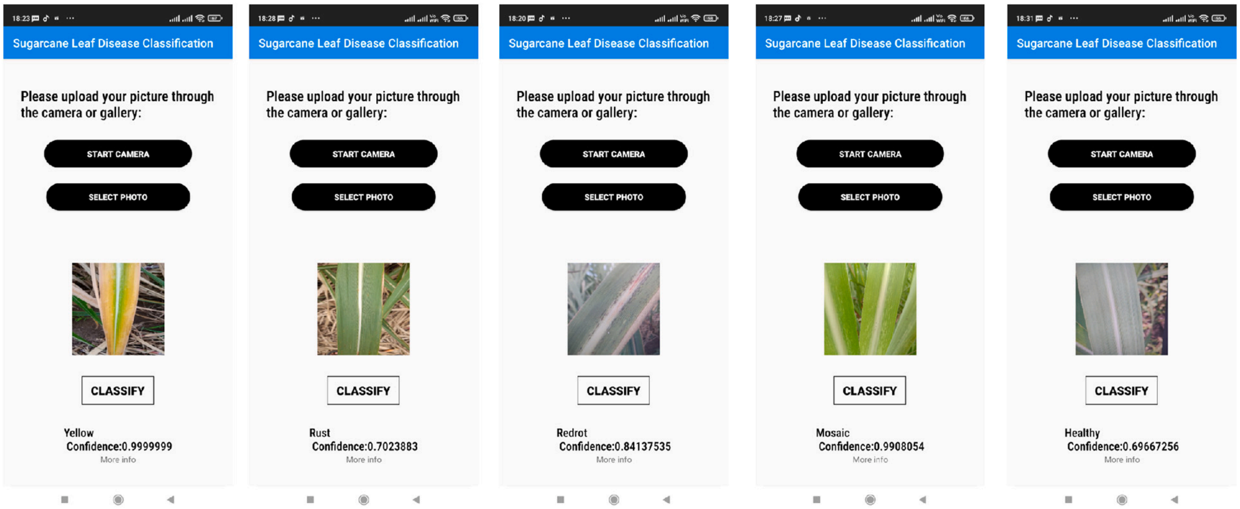

Fig. 11. Classification results of in-field testing of android application.

agricultural sciences plus having over 5 years of experience in plant pathology and relevant areas. In-field results were validated by each of the aforementioned groups to ensure error-free analysis. The Fig. 11 shows five screenshots, each of which represents a distinct sugarcane disease type, along with the score of classification accuracy achieved by the disease using an Android app. The disease type is identified beneath every image, which provides a clear idea of the sugarcane leaves that have been infected. The application's success in correctly recognizing the particular condition is demonstrated by the accuracy percentage that is displayed alongside each disease type. Together, the photos and accuracy percentages show how well the application can identify various sugarcane disease types and give farmers useful assistance in making wise crop management decisions.

3.6. Limitations

There are certain limitations associated with the studies made so far in above sections.

1. **Dataset size & diversity:**
   Small datasets like the one used in the study pose a significant risk that the model may learn peculiar patterns and sometimes fail to give correct predictions on new data. In this study, to mitigate this risk to a certain extent, samples were collected with a preferable degree of variety in terms of rotation, varying light conditions, and even with different capturing devices. However, an increase in the database size would certainly add more robustness to the model and lead to better generalization.
2. **Complexity of model vs. dataset size:**
   Complex models and small datasets often lead to overfitting, while simple models may struggle with underfitting and fail to generalize effectively. In this study, the AMRCNN model is relatively complex compared to the available dataset. In AMRCNN, the hierarchy level is restricted to only four levels to achieve the desired level of performance. However, caution must be exercised when further increasing complexity.
3. **Single dataset evaluation:** The proposed model and other well known architectures are tested with single self-created database to establish the baseline. It may sometimes impact the reproducibility of the results.

4. Conclusion

This study proposes a method for real-time field picture categorization utilizing the attention-based multilevel residual convolutional neural network (AMRCNN). A data augmentation strategy can be further used to address the issue of overfitting. This technique has increased the number of samples in the database, which has helped the network generalize more effectively. During the training phase, the classification performance was considerably enhanced by the saliency-detecting capacity caused by the attention modules with features extracted from multi-levels. According to experimental findings, the suggested AMRCNN outperformed state-of-the VGGNet, ResNet, XceptionNet, and EfficientNet models with a classification accuracy of 86.53%. However, there is still room for advancement, and it is anticipated that future work will explore architectural advancements, fine-tune the hyperparameters, and expand the dataset size in order to further improve the network's performance. Overall, our results point to the AMRCNN as a good contender for precise and effective classification of sugarcane leaf disease in real time. Additionally, by using tensorflow lite tools for quantization, the AMRCNN's running time is reduced significantly, making it suitable for deployment in a mobile-based system. The created application gives the user a way to interact with the model and utilize it to address the practical classification issue of diseases affecting sugarcane crops. As a result, the farmers are able to identify diseases in their early stages and spare their sugarcane crops from being destroyed.

Funding

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

CRediT authorship contribution statement

Swapnil Dadabhau Daphal: Writing – original draft, Visualization, Validation, Software, Resources, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization. **Sanjay M. Koli:** Writing – review & editing, Supervision, Project administration, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Dataset details: Daphal, Swapnil; Koli, Sanjay (2022), "Sugarcane Leaf Disease Dataset", Mendeley Data, V1, doi: https://doi.org/10.17632/9424skmnrk.1

Here is the URL: https://data.mendeley.com/datasets/9424skmnrk/1

References

[1] Amos P.K. Tai, Maria Val Martin, Colette L. Heald, Threat to future global food security from climate change and ozone air pollution, Nat. Clim. Change 4 (9) (2014) 817–821.
[2] Vijai Singh, Namita Sharma, Shikha Singh, A review of imaging techniques for plant disease detection, Artif. Intell. Agric. 4 (2020) 229–242.
[3] Xin Yang, Tingwei Guo, Machine learning in plant disease research, Eur. J. BioMed. Res. 3 (2017) 6.
[4] Javaid Ahmad Wani, Sparsh Sharma, Malik Muzamil, Suhaib Ahmed, Surbhi Sharma, Saurabh Singh, Machine learning and deep learning based computational techniques in automatic agricultural diseases detection: methodologies, applications, and challenges, Arch. Comput. Methods Eng. 29 (1) (2022) 641–677.
[5] Clive Bock, Gavin Poole, P.E. Parker, Timothy Gottwald, Plant disease severity estimated visually, by digital photography and image analysis, and by hyperspectral imaging, Crit. Rev. Plant Sci. 29 (2010).
[6] Miguel Pagola, Rubén Ortiz, Ignacio Irigoyen, Humberto Bustince, Edurne Barrenechea, Pedro Aparicio-Tejo, Carmen Lamsfus, Berta Lasa, New method to assess barley nitrogen nutrition status based on image colour analysis: comparison with spad-502, Comput. Electron. Agric. 65 (2) (2009) 213–218.
[7] Yansheng Li, Yongjun Zhang, Zhihui Zhu, Error-tolerant deep learning for remote sensing image scene classification, IEEE Trans. Cybern. 51 (4) (2021) 1756–1768.
[8] Xiao Xiang Zhu, Devis Tuia, Lichao Mou, Gui-Song Xia, Liangpei Zhang, Feng Xu, Friedrich Fraundorfer, Deep learning in remote sensing: a comprehensive review and list of resources, IEEE Geosci. Remote Sens. Mag. 5 (4) (2017) 8–36.
[9] Vijay Badrinarayanan, Alex Kendall, Roberto Cipolla, Segnet: a deep convolutional Encoder-Decoder architecture for image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 39 (12) (2017) 2481–2495.
[10] Evan Shelhamer, Jonathan Long, Trevor Darrell, Fully convolutional networks for semantic segmentation, in: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 3431–3440.
[11] Olaf Ronneberger, Philipp Fischer, Thomas Brox, U-net: Convolutional networks for biomedical image segmentation, CoRR, arXiv:1505.04597 [abs], 2015.
[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, Deep residual learning for image recognition, CoRR, arXiv:1512.03385 [abs], 2015.
[13] Yibin Wang, Haifeng Wang, Zhaohua Peng, Rice diseases detection and classification using attention based neural network and Bayesian optimization, Expert Syst. Appl. 178 (2021) 114770.
[14] Henan Sun, Haowei Xu, Liu Bin, Dongjian He, Jinrong He, Haixi Zhang, Nan Geng, Mean-ssd: a novel real-time detector for apple leaf diseases using improved light-weight convolutional neural networks, Comput. Electron. Agric. 189 (2021) 106379.
[15] Xiaoyue Xie, Yuan Ma, Liu Bin, Jinrong He, Shuqin Li, Hongyan Wang, A deep-learning-based real-time detector for grape leaf diseases using improved convolutional neural networks, Front. Plant Sci. 11 (2020).
[16] Swapnil Dadabhau Daphal, Sanjay M. Koli, Transfer learning approach to sugarcane foliar disease classification with state-of-the-art sugarcane database, in: 2021 International Conference on Computational Intelligence and Computing Applications (ICCICA), 2021, pp. 1–4.
[17] Solemane Coulibaly, Bernard Kamsu-Foguem, Dantouma Kamissoko, Daouda Traore, Deep neural networks with transfer learning in millet crop images, Comput. Ind. 108 (2019) 115–120.
[18] Michael Gomez Selvaraj, Alejandro Vergara, Henry Ruiz, Nancy Safari, Sivalingam Elayabalan, Walter Ocimati, Guy Blomme, Ai-powered banana diseases and pest detection, Plant Methods 15 (1) (2019) 92.
[19] Amreen Abbas, Sweta Jain, Mahesh Gour, Swetha Vankudothu, Tomato plant disease detection using transfer learning with c-gan synthetic images, Comput. Electron. Agric. 187 (2021).
[20] Gong Cheng, Xingxing Xie, Junwei Han, Lei Guo, Gui-Song Xia, Remote sensing image scene classification meets deep learning: challenges, methods, benchmarks, and opportunities, IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 13 (2020) 3735–3756.
[21] Jakob Engel, Vladlen Koltun, Daniel Cremers, Direct sparse odometry, CoRR, arXiv:1607.02565 [abs], 2016.
[22] Gong Cheng, Xingxing Xie, Junwei Han, Lei Guo, Gui-Song Xia, Remote sensing image scene classification meets deep learning: challenges, methods, benchmarks, and opportunities, IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 13 (2020) 3735–3756.
[23] Wenming Cao, Zhiyue Yan, Zhiquan He, Zhihai He, A comprehensive survey on geometric deep learning, IEEE Access 8 (2020) 35929–35949.
[24] Qifan Wang, Man Cheng, Xuepeng Xiao, Hongbo Yuan, Jiajun Zhu, Caihu Fan, Jinlin Zhang, An image segmentation method based on deep learning for damage assessment of the invasive weed *Solanum rostratum* Dunal, Comput. Electron. Agric. 188 (2021) 106320.
[25] Sharada Mohanty, David Hughes, Marcel Salathe, Using deep learning for image-based plant disease detection, Front. Plant Sci. 7 (2016).
[26] Swapnil Daphal, Sanjay Koli, Sugarcane leaf disease dataset, https://data.mendeley.com/datasets/9424skmnrk/1, 2022.
[27] Sanghyun Woo, Jongchan Park, Joon-Young Lee, In So Kweon, Cbam: Convolutional Block Attention Module, 2018.

[28] Karen Simonyan, Andrew Zisserman, Very deep convolutional networks for large-scale image recognition, CoRR, arXiv:1409.1556 [abs], 2014.
[29] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, Deep residual learning for image recognition, CoRR, arXiv:1512.03385 [abs], 2015.
[30] François Chollet, Xception: deep learning with depthwise separable convolutions, CoRR, arXiv:1610.02357 [abs], 2016.
[31] Mingxing Tan, Quoc V. Le, Efficientnet: rethinking model scaling for convolutional neural networks, CoRR, arXiv:1905.11946 [abs], 2019.
[32] TensorFlow Lite, https://www.tensorflow.org/lite/models/convert/, 2024.