Rice Disease Classification

Mohamed Adel , Mohamed Atef, Omar Tarek, Samaa Maged

Dr. Ghada Khoriba

***Abstract*—in this paper, we propose a classification model for identifying different types of rice leaves based on their health condition, explicitly categorizing them as healthy, affected by brown spots, or affected by leaf blast.**

Through our proposed approach, we aim to achieve accurate classification of rice leaf images, which can contribute to early detection and intervention in plant health management systems.

Keywords—computer vision, agriculture, rice disease

I. INTRODUCTION

One of the most important agricultural products that people rely on for food is rice. Especially in our country, Egypt, it has importance in the food and economic security of the state. But there are many problems that threaten the production, including diseases that affect the rice leaf, such as (blast and brown disease), these two diseases will have an important role in this research.

Today, artificial intelligence has become one of the most important methods used to review and monitor all agricultural crops and early detection of diseases around the world. This is due to the high efficiency and continuous development of deep learning and machine learning algorithms.

this paper will discuss the classification problem our problem is to detect rice plant diseases. we had joined a competition on the Zindi website. this competition is closely connected to our research. This competition aims to detect rice planet diseases the data set was taken in Egypt by Microsoft. The data have 3 main classes 2 diseases (Blast, and Brown diseases), and 1 Healthy.

our approach in this paper is to build a neural network architecture that can be more effective than the existing approaches. after researching and skimming some related work, we found that most of it used pre-train models like (ResNet14, ResNet34, and ResNet50) with adding self-attention layers to try to accelerate the and develop performance of the approaches that preceded it like CNN (AlexNet, MobileNet, VGG, VGG16, and others). Finally, we decided to merge two architectures in two different papers first one approaches ResNet34 by putting the self-attention layers in the middle of the ResNet34 layers[reference]. the second the researcher used VGG16 but they put self-attention layers outside the model[reference]. By merging those two approaches our approach will be adding self-attention layers outside the ResNet34 and then connecting both of them with a Fully connected layer at the end.

II. METHODOLGY

A. Dataset description

The dataset includes 3815 images of rice leaves that have been classed as Healthy, Brown spot, or Leaf blast. The dataset was gathered using Zindi. The images gathered are regarded as data samples. Figure 1 shows the different types of rice leaves. From the collected dataset, 2670 images are used for training and validation, and 1145 images are used for testing. To perform augmentation, all the images are resized to 224 * 224 pixels. Vertical and horizontal flips were applied randomly on the images. For training and validation, the obtained samples were randomly divided into 70:30 proportions. The training and validation samples are chosen at random for each execution.

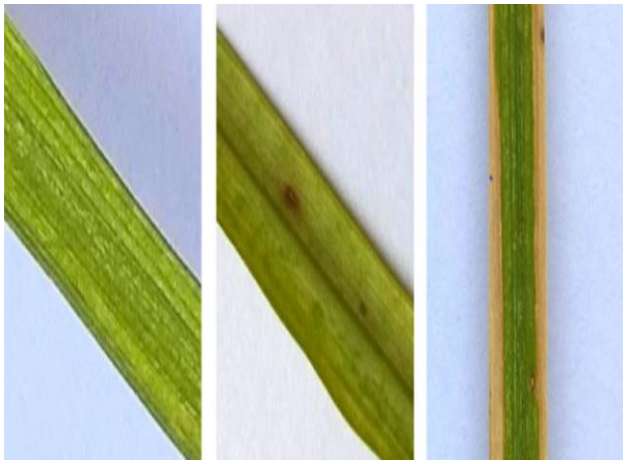

Fig. 1 Sample rice leaf images from the dataset. a) Healthy. b) Brown spot. c) Leaf blast.

B. Proposed approach

In this paper, we will apply pre-trained ResNet-34 with self-attention to classify the rice leaf images into healthy, brown, or blast. This approach is considered a merge between the two approaches mentioned in the literature [2], where we applied ResNet-34 like the first paper's approach, however, we applied self-attention outside of the model architecture as like the second paper's approach. The proposed model architecture is shown in figure 2. Additionally, we trained the model using mini-batches for 30 epochs, where each batch consists of 150 images for the training and the validation datasets. During each epoch, the model is trained on all batches (the whole dataset) and is verified using the validation batches. After that, the epoch with the minimum loss on both training and validation was picked for testing. Categorical entropy loss was selected to calculate the model loss, and Adam optimizer was chosen to optimize the model loss with a learning rate of 0.001.

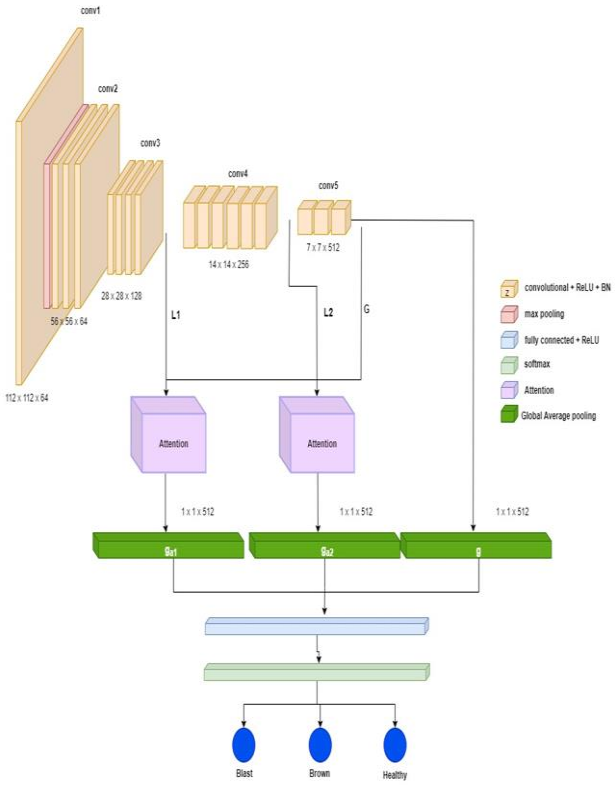

Fig. 2 The proposed ResNet-34 with self-attention architecture.

C. *Pre-trained ResNet-34*

ResNet34 is a 34-layer architecture that can be utilized to perform cutting-edge image classification. The number of parameters used in this model is 21.28 M. Figure 2 shows rice leaf disease identification by the ResNet34 model, it consists of one convolution and pooling step which are considered as the preliminary layer followed by 4 layers which have a similar pattern. Floating-point operations per second of ResNet34 are 3.6*109. It performs 3*3 convolutions with a fixed feature map dimension (F) of two, passing the input through every two convolutions (64, 128, 256, 512). All of the layers have the same width (W) and height (H) parameters. The first convolutional block (Conv1) of ResNet34 performs 3 operations. The first one is the Batch Normalization (BN) technique with mini-batches instead of the entire data set using Eq. (2), which helps in speeding up and making learning easier. The second is RELU (Rectified Linear Unit) which is a nonlinear activation function, and it means all neurons are not activated at the same time using Eq. (1). The RELU layer gives probability to the pixels with a positive value in the image and converts the negative value to zero.

$$f(a) = \max(0, a) \qquad (1)$$

where a is the positive value. Batch normalization is mainly applied in the layers of the neural network for optimization instead of applying to the actual data itself. The complete input dataset is processed in mini-batches instead of processing the data as a whole. This step mainly speeds up the training of the ResNet architecture and helps to use higher learning rates.

$$z^n = \frac{z - m_z}{s_z} \qquad (2)$$

Where z^n is the output of the batch normalization, z is the output neuron, m_z is the mean value of the neuron output, and s_z is the standard deviation of the neuron output. The output from the ReLU layer is given as the input to the pooling layer, and it is mainly applied in scenarios where the image parameters are too large to process. For max-pooling, we take the largest element obtained from the rectified feature map and the average pooling is obtained by computing the overall average value of every element present in the feature map.

$$PO = B^{x,x}_{max-pool}(R) \qquad (3)$$

$$PO = B^{x,x}_{avg-pool}(R) \qquad (4)$$

Where PO is the pooling output, $B^{x,x}_{max-pool}$ is the max pooling value of the x ×x input block, R is the input image, and $B^{x,x}_{avg-pool}$ is the average pooling value of the x ×x input block. The max-pooling function is last, and it is computed using Eq. (3). The 4 layers which follow block 1 perform the same operation as BN and RELU. The final 512 channel of the convolution layer is converted into linear layer 1*1 convolutions by taking average pooling (Eq. 4). Finally, it the fully connected to the SoftMax activation function with 4 Neurons.

This is the normal pre-trained ResNet-34. However, in our approach as seen in figure 2, we cancelled the fully connected layer at the end of ResNet-34 and added two self-attention layers between conv3 and conv4. These two layers help in a more detailed feature extraction that the regular ResNet-34, and thus result in an accurate classification of the rice leaf images. The outputs of these attention layers are then introduced for a fully connected layer with a SoftMax for classification. The details of the attention layer are discussed in the following section.

D. *Self-attention layer*

Before we introduce how the self-attention layer works, we will introduce some notation which we adopted from [1]. This notation is shown in figure 3.

Basic Notation

- s = a number between 1 and 34 indicating the conv layer (since there are 34 conv layers in this model);
- i = a number between 1 and n indicating the spatial location within a conv layer;
- g = the "global image descriptor" or "global feature vector" that is produced as the output of layer 34 . You can think of this as a vecto that describes the entire input image. This g is not to be confused with g_a, which is an output of an attention estimator.

More Notation

- l_i^s = the vector of output activations for conv layer s and spatial location i;
- $\mathcal{L}^s = \{l_1^s, l_2^s, \ldots, l_n^s\}$ = the set of feature vectors extracted at a given conv layer s;
- c_i^s = the "compatibility score" for conv layer s and spatial location which is calculated from l_i^s and the global feature vector g. The compatibility score is basically an intermediate step in the attentio calculation;
- a_i^s = the "attention weight" for conv layer s and spatial location i, which is calculated from c_i^s;
- g_a^s = the final output of the attention mechanism for conv layer s, which is calculated from a_i^s and l_i^s;
  - The subscript "a" here is not indexing into anything; it just indicates "attention output" and likely is intended to distingui g_a from the global feature vector g;

Fig. 3 Basic notation.

There are 3 steps in the attention layer mechanism:

1. **Calculation of the compatibility scores**.

The compatibility scores are calculated using local features l and the global feature map g. The scores rend to have a high value when the image patch described by the local features contains parts of the dominant image category. In the proposed architecture, the local features are taken after conv3 and conv4. The global feature map is taken after conv5. The proposed formula for calculating the compatibility scores is shown on Eq.5.

$$c_i^s = \langle u | l_i^s + g \rangle, i \in \{1 \ldots . n\} \qquad (5)$$

where u is a learned vector. We apply Eq.5 for the two local features $l_i^3 after\ conv3$ and $l_i^4 after\ conv4$. If g and l are not the same size, we use a convolution block of kernel size 1 to project l in order to have the same channels as g.

2. **Calculation of attention weights.**

It is the SoftMax of compatibility scores as seen in Eq.6.

$$a_i^s = \frac{exp(c_i^s)}{\sum_j^n exp(c_j^s)} \qquad (6)$$

3. **Calculation of the final output of the attention layer.**

We calculate the final output of the attention mechanism g_a^s for a particular layer s by taking a weighted combination of the l for that layer (recall that the l are just the outputs of that layer.) as seen in Eq.7. The weights we use are the attention weights that we just calculated.

$$g_a^s = \sum_{i=1}^{n} a_i^s \times l_i^s \qquad (7)$$

The outputs for the proposed model are $g_a^3\ (after\ conv3)$ and $g_a^4\ (after\ conv4)$.

4. **Make a classification prediction using the attention layers' outputs.**

The authors will use the attention layers outputs g_a^3 and g_a^4 along with the global feature map g to classify the rice leaf images. A concatenation operation will be applied to concatenate the three feature maps and then will be fed to a fully connected layer for classification.

III. RESULTS

In the model we randomly divide 2,670 images into a 70:30 proportions for training and validation. As shown in figure 4 we used Sparse Categorical Crossentropy as it is the method used in Zindi.

But when the model tested on Zindi private test set which is not available to us the score was 1.9.

\

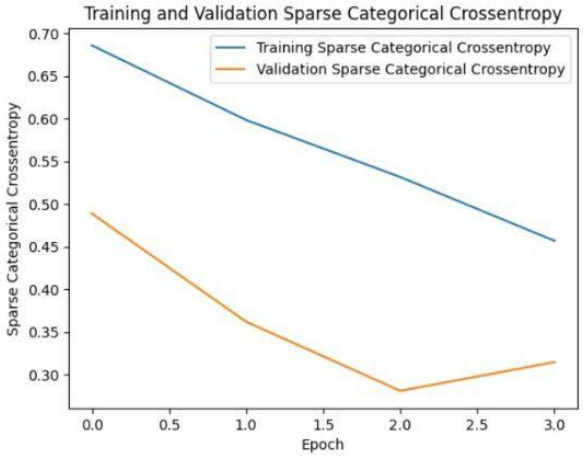

Fig. 4 Training and validation loss in our dataset

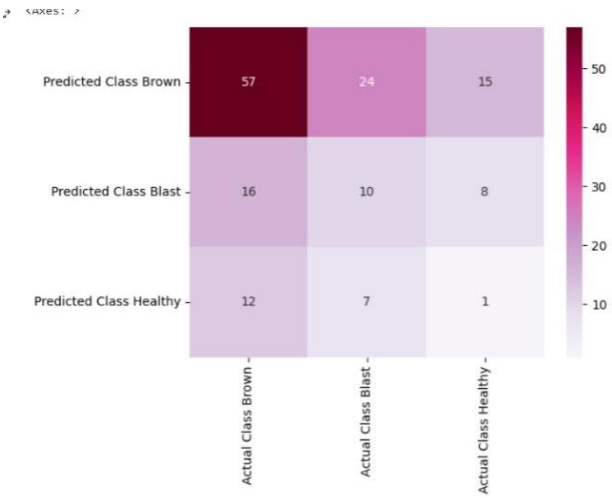

Fig. 5 Confusion Matrix

IV. CONCLUSION

V. In this paper, the rice leaf diseases are identified from the leaf images effectively using 2 pre-trained CNN models using transfer learning in ResNet50. Since the expected score is not met, the Self-Attention layer was added along with ResNet34 to get a better score. When the two models are compared, the ResNet34 with Self-Attention produces the better score. In our input dataset, 32% of the false leaf images were misclassified, while 68% of the leaf images were correctly classified. When all images were considered, Blast spot was predicted with an accuracy of 29% and a loss of 71%, Brown spot was predicted with an accuracy of 59.4% and a loss of 41.6%, and Healthy images were predicted with an accuracy of 5% and a loss of 95%. In future, the score can be improved by adding additional datasets and tuning the parameter of the CNN model. The same classification algorithm can be used to identify diseases in another crop.

REFRENCES

.

[1] Saumya Jetley, Nicholas A. Lord , Namhoon Lee & Philip H. S. Torr. LEARN TO PAY ATTENTION at ICLR (2018)

[2] Stephen, A., Punitha, A. & Chandrasekar, A. Designing self attention-based ResNet architecture for rice leaf disease classification. Neural Comput & Applic 35, 6737–6751 (2023). https://doi.org/10.1007/s00521-022-07793-2