Deep Learning for Plant Species Identification: An AI Research Project

[1] B. Naresh, [2] Bachala Swetha,

[1]Assistant Professor, Megha Institute of Engineering & Technology for Women, Ghatkesar.
[2] MCA Student, Megha Institute of Engineering & Technology for Women, Ghatkesar.

Abstract

These days, researchers are putting a lot of effort into artificial intelligence (AI) with the goal of making AI smarter. Particularly in the field of object identification, machine learning gained a lot of traction. The author built a CNN project with pycharm, anaconda, and kera to determine the best way to improve recognition program accuracy and recognition speed, with the goal of providing a faster and more accurate plant species recognition program. The project also introduced deep learning and convolution neural networks (CNNs). The author made an effort to discover the optimal solution by adjusting the learning period time and the capacity of the learning data collection. Analysis of the test results shows that increasing the number of learning epochs and the size of the training picture collection both contribute to faster and more accurate identification. Improving accuracy is the most noticeable result of increasing learning time, and expanding the size of the training set is the best way to decrease recognition time. The results of the experiment, the essay's shortcomings, and a prediction for the future of machine learning in the plant field were all included at the conclusion of the thesis.

Keywords:

AI, convolutional neural networks, deep learning, plants

I.INTRODUCTION:

Navigation, autonomous driving, and data analysis are just a few of the several fields that have benefited greatly from the fast advancements in artificial intelligence (AI) in recent years. The integration of machine learning with this technology has also led to its use in object identification, and associated applications have found their way into many electronic gadgets. For instance, once the user places an item on the camera sight, Alipay's function provides information about the thing [1]. Improving artificial intelligence via machine learning is a hot topic right now. However, a more impressive approach called CapsNet may take the role of convolutional neural networks due to their issues with value data loss in pool layers and high sample demand for backpropagation algorithms. The programmers discovered an efficient way to optimize the application since they cannot accept the poor recognition speed and low accuracy. In this paper, we will build a convolutional neural network algorithm that can identify various plant species, and we will look for ways to make it faster and more accurate. For instance, programmers may discover the most efficient strategy to develop the AI recognition software and attain optimum efficiency by

determining if it is advantageous to increase the training sample size. A CNN image recognition software would be built and tested using the following tools: pycharm, anaconda, keras, opencv-python, tensorflow, and the numpy library. The author would get the plant picture dataset from the Kaggle website. Program accuracy and speed might be enhanced by deciding whether to enlarge the training picture collection or increase the number of learning epochs. By the end of the piece, the author has compared the two approaches and settled on the one that will help programmers enhance their object recognition software the most.

II.MAINBODY

A. Introduction to Machine Learning

At its heart, AI is machine learning; under the old-school programming paradigm, humans fed computers data and algorithms, and the computers produced results. However, machine learning systems use data and results as input, search for the technique employed by older systems, and then train computers to remember and apply that approach to new situations. As time goes on, a single computer learns more and more algorithms, allowing it to tackle more and more problems.Machine learning includes the subfield known as deep learning. Jeff Dean and Andrew Ng's Deep Neural Network was released in 2012 as part of the Google Brain project. Multiple individual neurons make up a neural layer. While both regular and deep neural networks have three main components—an input layer, a hidden layer, and an output layer—deep neural networks vary in that they only have one hidden layer rather than two or more. Using weights, input is changed to an intermediate state (a neural unit). Then, the input is translated into output using various weights across layers. Using the same input data, various weight combinations will result in different outputs. Thus, deep learning's nonlinearity, generativity, and cross-modularity are its three main benefits. As a result, it has lately achieved remarkable advancements in several domains, including picture categorization and video comprehension, and has grown into a crucial instrument in numerous recognition applications [1]. The system's learning capability will be enhanced as the number of layers increases.

B. Convolutional Neural Networks: An Introduction

One method in deep learning that is used for picture identification is the convolutional neural network. [2] in This network uses convolutional calculations and is a kind of neural network. The convolutional neural network is more suited to handle data in two dimensions than the deep neural network. There are essentially five stages to the CNN: The input layer comes first. Both DNN and CNN use vector formats for its inputs, however CNN uses a three-dimensional pixel matrix. How long and width stand for the relative picture data, whereas depth is used to record the image's color. Next, we have the convolution layer. Consider the convolution layer as a filter. It iteratively removes the input image's node matrix before convoluting the kernel matrix to generate a fresh output matrix. The convolution kernel is involved in this stage, making it the most significant in CNN. Lastly, the pooling layer. Given that the output matrix from step 2 could be too big, the primary objective of this layer is to decrease the matrix size. Actually, it speeds up the computation speed by lowering the number of nodes in step 4, which is the same as reducing the neural network parameters. The full-connection layer is the fourth. The primary function of this layer is to sort the many pieces of detail data obtained in step 3 into their

respective categories. The next step is to activate it and begin training. At last, we reach the graphics derived section. Two visual representations of the output are accuracy and loss. A more accurate model is indicated by a smaller loss value. The author obtained over 10,000 visuals from Kaggle for the picture test. Lastly, there is the Softmax layer, which is used to get the probability distribution graphic. Compared to deep learning neural networks, convolutional networks were able to retain plane structural information and provide more accurate forecasts, according to the technique.

C. The region used by the convolutional neural network for plant identification firstly, procedures Computer developers have been eager to keep developing quicker and more accurate programs concerning face recognition and video comprehension, since image recognition software has recently been a popular topic due to the expansion of camera devices and the cumulative need for security. Due to its widespread usage, CNN became the optimal object [3]. Finding certain types of plants may be a real challenge, particularly when you're out in nature. Ordinary outdoor adventurers and botanists both depend on software that can identify plants. Because of the hazardous fruits and dangerous snakes, precision and promptness are of the utmost importance. In order to determine the optimal way to enhance CNN in practice, the author plans to conduct tests to see if increasing the number of learning epochs or the quantity of extended learning images may improve the speed and accuracy of machine learning tests and recognitions. For this experiment, the author relied on pycharm in a 32-bit environment that also included opencv-python, tensorflow, numpy, keras, and the os libraries. The following components made up the project: An epoch number, three parameters for the convolution layer, and the address of the learning set are mostly stored in the parameter section. The next step was loading the images, which included preparing an empty picture list and saving the array details. Next, we have the transformer, which is mostly responsible for transforming images into matrix arrays using cv2. The next step in constructing the network is to divide the whole data array into a train set and a test set using the train_test_split() function. Then, using the add() method, we add various building blocks to the convolution, pooling, full-connection, and softmax layers.

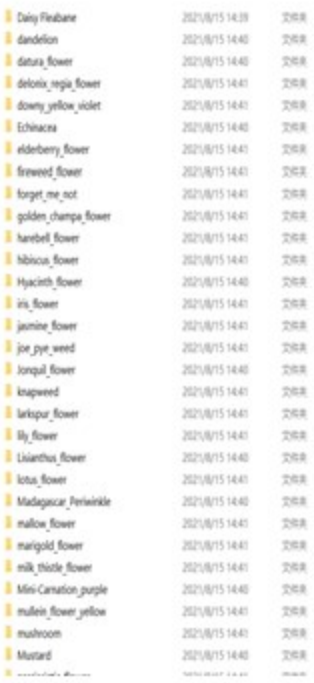

Figure 2.Large dataset

Since the majority of the photos are 256 × 256 pixels in size, the usual values for Isetwidth, length, and depth are 256 and 3, respectively. There are 32 batches. The convolution layer provides the option to utilize the sigmoid-tanh-ReLU activation function, although this is not recommended because of the drawbacks of the ReLu function.The author primarily used the ReLu function in the convolution layer to eliminate all negative values and set zero, which helped with the disappearance of gradients and reduced computation time [4]. 2) Do some testing Step one: lengthen the time spent learning According to Maram, extending the learning period involves raising the iterate time throughout the training process.

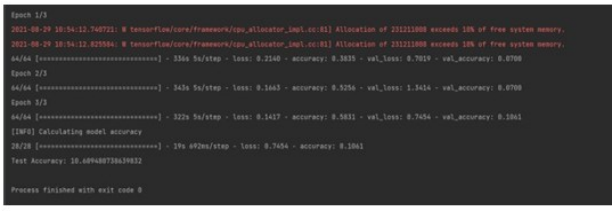

Figure3.Thepython console out put about all parameters for each epochs

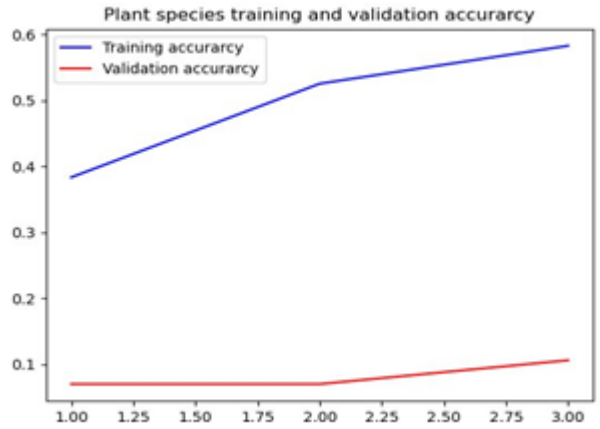

Figure4Plant species training and validation accuracy

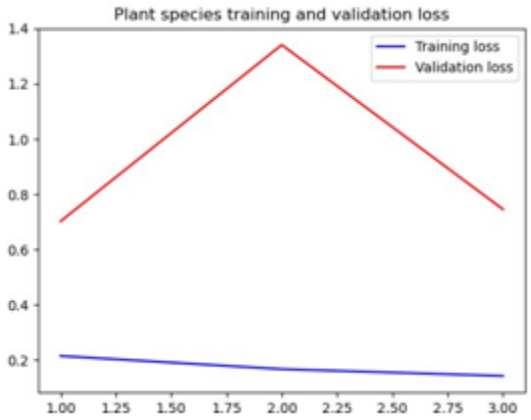

Figure5 Plant species training and validation loss

whenepoch=7,theresult is:

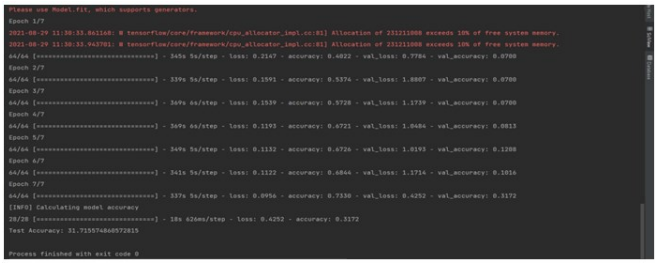

Figure 6 Thepython console out put about all parameters for eache pochs.

whenepoch=7,theresult is:

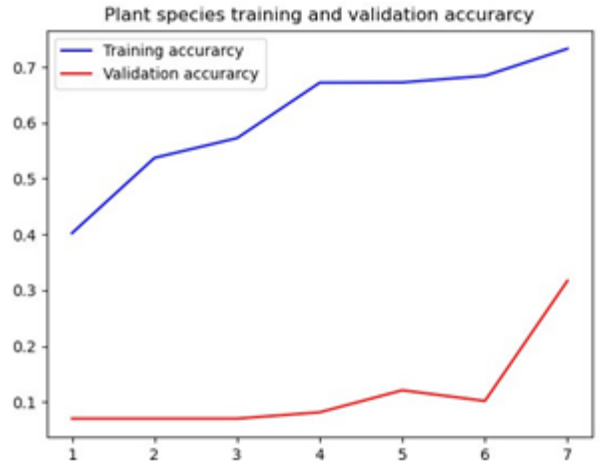

Figure7Plants pecies training and validation accuracy.

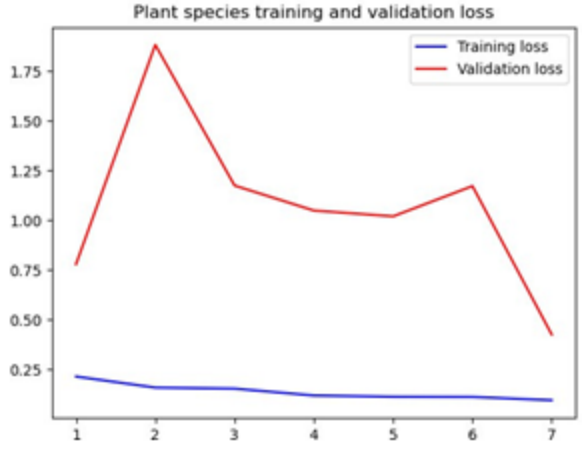

Figure. 8 Plants.peciestrainingandvalidationloss

whenepoch=15,theresult is:

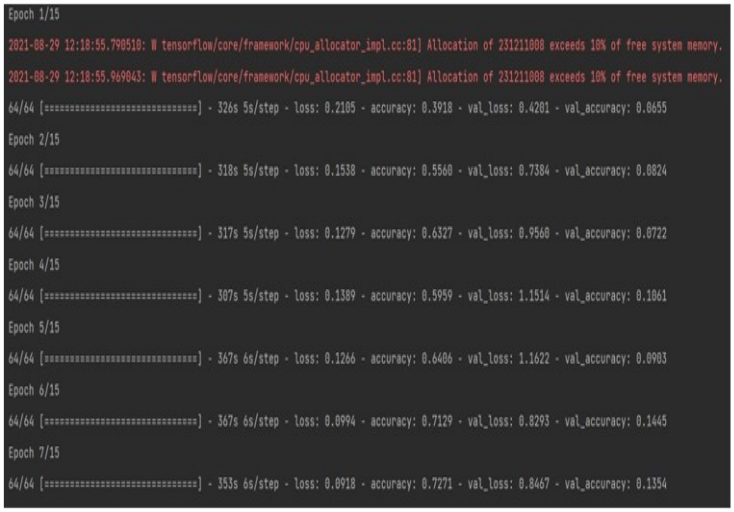

Figure 9 Thepython console output about all parameters for eache pochs.

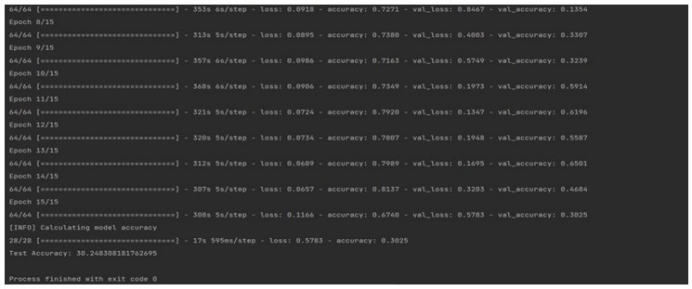

Figure 10 Thepython console out put about all parameters for eache pochs

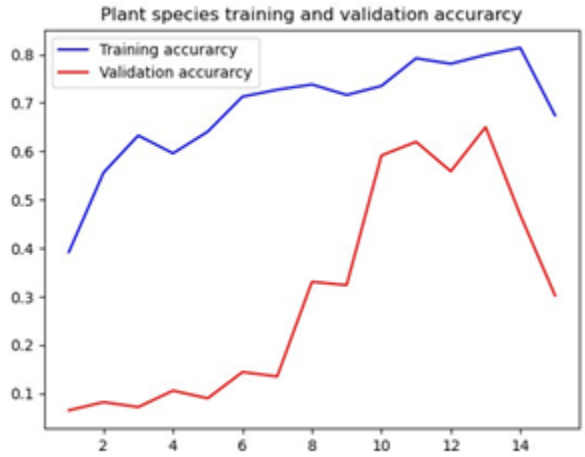

Figure11Plantspeciestrainingandvalidationaccuracy

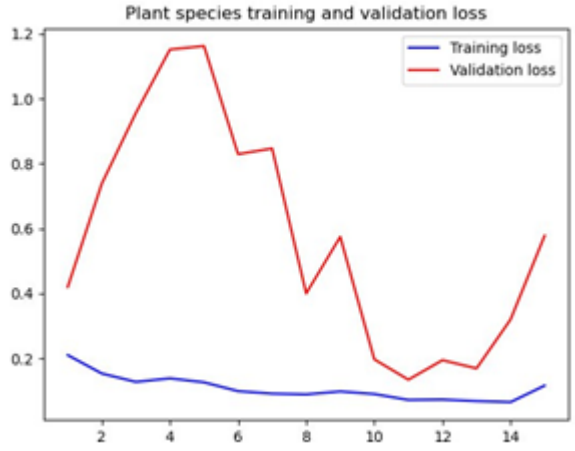

Figure12Plantspeciestrainingandvalidationloss

When epoch is less than 5, the validation accuracy ranged about 0.57, whereas the loss fluctuated around 1.1, assuming the odd epoch 15 result in c is ignored. At 13 epochs, however, validation accuracy reached 0.6501, a significant increase over epochs 5 through 5. Contrarily, in the eleventh epoch, the validation loss hit rock bottom at 0.1347.Three epochs took 19 seconds, seven epochs 18 seconds, and fifteen epochs seventeen seconds each test.

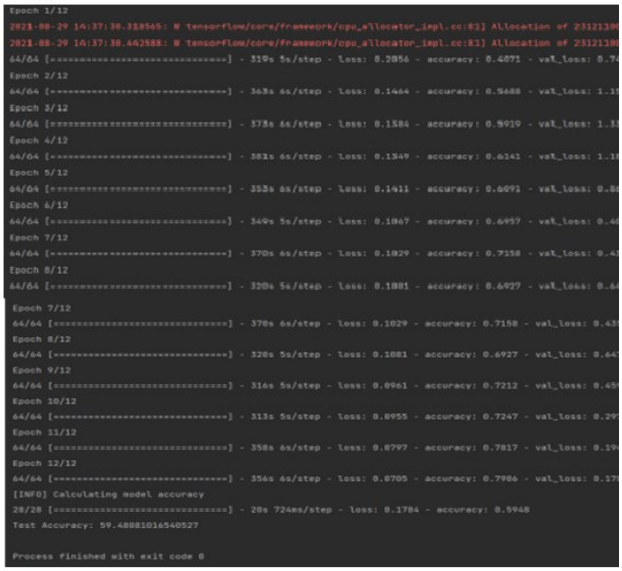

Figure13&14Thepythonconsoleoutputaboutallp arametersforeachepochs.

Finally, additional learning epochs will save more valid recognition time, but the system's recognition accuracy will decrease when the epoch count exceeds a critical threshold. c) The dataset for training To make the training picture set larger, you need to change the value of the "test_size" argument in the train_test_split() method. A bigger training set may be achieved with a smaller "test_size". The epoch is set to 12. at test_size=0.3, with trainingset:testset ratio of 7:3.

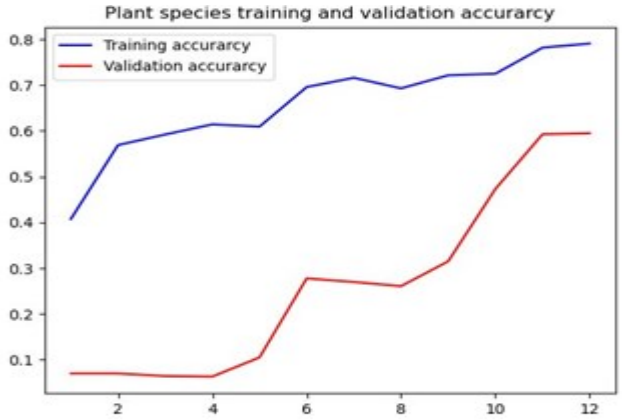

Figure15Plantspeciestrainingandvalidationaccuracy

whentest_size=0.5,trainingset:testset=1:1.

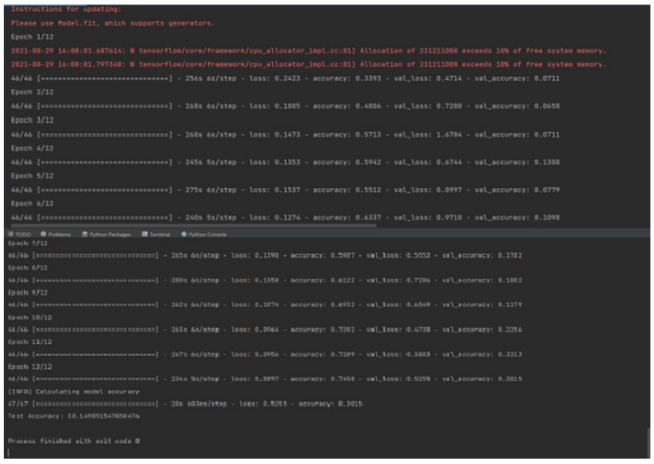

Figure16&17Thepythonconsole outputaboutallparametersforeachepochs.

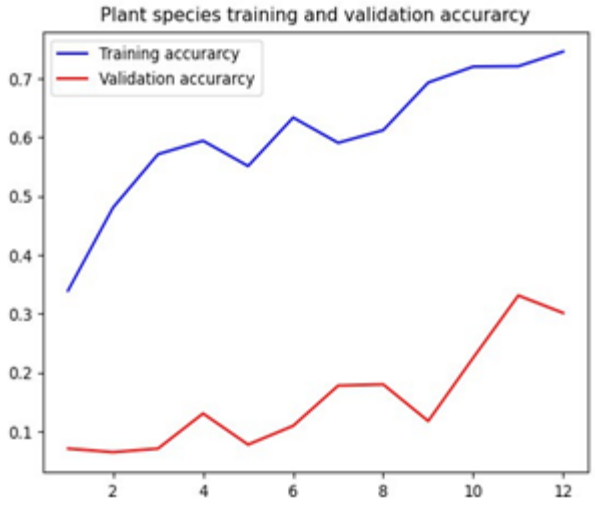

Figure18Plantspeciestrainingandvalidationaccuracy.

whentest_size=0.7,trainingset:testset=3:7.

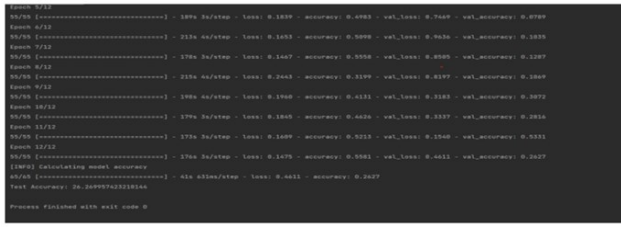

Figure19Thepythonconsoleoutputaboutallparametersforeachepochs.

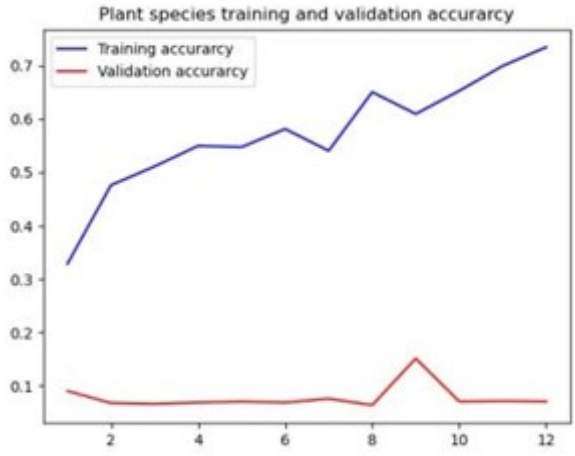

Figure20Plantspeciestrainingandvalidationaccuracy.

At test size=0.3, the author discovered that when comparing validation accuracy at 12 the pochinall plots,

When the test size is 0.5, the validation accuracy is 0.3015; when the test size is 0.7, the validation accuracy is 0.0586. The current validation accuracy is 0.5948. Reduced accuracy is a direct result of a smaller training set. The test in A cost 20 seconds, the test in B cost 28 seconds, and the test in C cost 41 seconds. on order to have the software spend more time on the recognition test, the author discovered that a smaller training set is required.

Result and analysis

Expanding the training picture collection and increasing learning epochs time were both shown to enhance identification speed and accuracy, according to the author's comparison of the two approaches' findings. Increasing the learning time has a more noticeable impact on boosting accuracy, while increasing the size of the training set is a better way to decrease recognition time. Another crucial aspect to consider is that in order to achieve the accuracy increase effect, the learning epochs need to surpass a critical threshold.

III.CONCLUSION:

In order to enhance CNNs, this article is confirming the utilization of increasing learning epochs time and expanding the training picture collection. The conclusion is that software programmers may enhance the learning time in advance, surpass the criteria, or grow the training picture collection to be

big enough to improve the program's identification accuracy and speed. Nevertheless, there is still a lot of room for improvement in the author's study. The author did not investigate the impact of altering batch size and other potential metrics since their computer's GPU was too limited. The author intends to go more deeply into this field and make improvements to more powerful facilities in the future in order to fix these issues. The author successfully ran the program in PyCharm and obtained the experimental results.

REFERENCES

[1] XiaoBai& XiangWang& Xianglong Liu & Qiang Liu & Jingkuan Song& NicuSebe & BeenKim(2021), *Explainable deep learning for efficientand robust pattern recognition: Asurvey of recent developments*, 108012,180

[2] Al-AniMustafa & MajidAhmed & JamalAlshaibi & EvgenyKostyuchenko& AlexanderShelupanov(2021), *A review of artificial intelligence basedmalware detection using deep learning,* ISSN 2214-7853.

[3] Xin Yin & Quansheng Liu & Xing Huang & Yucong Pan(2021),*Real-timeprediction of rockburst intensity using an integrated CNN-Adam-BOalgorithm based on microseismic data and its engineering application,*TunnellingandUndergroundSpaceTechnology,Volume117,104133, ISSN0886-7798

[4] Mohit Agarwal & Suneet Gupta & K.K. Biswas(2021),*A new Conv2Dmodel with modified ReLU activation function for identification ofdiseasetypeandseverityincucumberplant,*SustainableComputing: InformaticsandSystems,Volume30,2021,100473,ISSN2210-5379

[5] Maram Mahmoud A. Monshi & Josiah Poon & Vera Chung & FahadMahmoud Monshi(2021), *CovidXrayNet: Optimizing data augmentationand CNN hyperparameters for improved COVID-19 detection from CXR,*ComputersinBiologyandMedicine,Volume133,104375,ISSN0010-4825

[6] Anil Kumar & Govind Vashishtha & C.P. Gandhi & Hesheng Tang &Jiawei Xiang,*Tacho-less sparse CNN to detect defects in rotor-bearingsystems at varying speed*, Engineering Applications of ArtificialIntelligence, Volume 104, 104401,ISSN 0952-1976,

[7] D.N.V.S.L.S. Indira & Jyothi Goddu & Baisani Indraja & Vijaya MadhaviLakshmi Challa & Bezawada Manasa(2021), *A review on fruitrecognition and feature evaluation using CNN,* Materials Today:Proceedings, ISSN 2214-7853

[8] Yurii Kardovskyi & Sungwoo Moon(2021), *Artificial intelligence qualityinspection of steel bars installation by integrating mask R-CNN andstereo vision,*Automation in Construction, Volume 130, 103850, ISSN0926-5805