Fruit and Quality Detection System to Assist People with Down Syndrome

Leenakshi Poudwal[1], Prof. Nilesh Rathod[2], Prathamesh Kagane[3], Niraj Shirkar[4]

[1]*Manjara Charitable Trust's Rajiv Gandhi Institute of Technology, Mumbai, India*
Department of Information Technology, Mumbai University, India

Abstract: Fruit classification using a Deep Convolutional Neural Network (CNN) is a promising application of computer Vision in this AI-driven decade. Fruit Recognition has been gaining attraction in recent years due to its application in agriculture and food industry. In this project, we propose a multi-fruit classification scheme based on Deep Convolutional Neural Network and we aim to build a Deep Neural Network to correctly identify the fruits and classify them according to their quality that is- into fresh and rotten. A fruit recognition process usually comprises of three steps: 1. Image acquisition 2. Pre-processing 3. Image analysis.
Keywords: Fruit Classification, Convolution neural networks, Deep neural networks, Multi-fruit classification, Computer Vision

I. INTRODUCTION

Convolutional Neural Networks (CNNs) are basically fully connected feed forward neural networks. They are quite effective at reducing the amount of parameters without sacrificing model quality. Images have a high dimensionality (each pixel is considered a feature), which suits the capabilities of CNNs outlined above. It was created in order to improve feature extraction and keep an image's shape information in the output. An approach is used for training a CNN model to classify an image according to its category, so that when fresh data is added as an input, the model can appropriately classify with respect to the input dataset.

CNN is a Deep Learning system that can take an image as an input, assign priority to various aspects/objects in the image, and distinguish between them. In ANN, a single layer contains multiple neurons. The weighted sum of all the neurons in a layer, which becomes the input of a neuron in the following layer, with a bias value added. They're linked to the Receptive Field in the area. A robust network is trained using the cost function that was generated.

In the domains of Robotic Harvesting, Yield Mapping for Agriculture, and Assisting people with Down Syndrome, a high-performing model for Fruit Recognition is critical. Only when a fruit identification system is trained on a readily available, credible dataset, shows real-time predictions, and properly classifies diverse types of fruits can it be considered accurate. We used Convolutional Neural Networks to develop a Multi-Fruit Classifier for our project. The image's input size is set at 35 by 35 pixels (RGB images). We played with the NN Architectural layers, filters, and hyperparameter tuning. The latest model's evaluation reveals a significantly higher rate of fruit recognition. This model was created in Python using Keras.

By far the most widely utilised method in multi-fruit recognition is Computer Vision (CV). DNN outperform all other machine learning techniques. Advancements in Computer Vision using Deep Learning have been built and developed over time, particularly through the use of one specific algorithm – the Convolutional Neural Network. Deep Learning uses CNN in Visual Identification Tasks, which span a wide range of domains, including video and image recognition, face recognition, handwritten digit recognition, and fruit recognition. Fruit Recognition using CNN, for example, has achieved near-human-level excellence in these categories. The biological model of mammalian visual systems inspired the development of the CNN. D. H. Hubel et colleagues. discovered in 1062 that cells in the cat's visual brain are refined to a small portion of the visual field known as the receptive field. The pattern recognition approach proposed by Fukushima in 1980 was motivated by the well-elucidated studies of D. H. Hubel et al. A typical CNN has a neural network architecture that includes convolution and pooling layers for extracting and combining high-level information from 2D input.

II. RELATED WORK

Fruit Recognition and Classification tasks are hampered by illuminance fluctuations, object obstacles, sharpened edges, texturing, reflectivity, and a variety of other issues. To address these issues, research papers are available. Fruit recognition should be viewed as an issue of Image Segmentation. Problem of Fruit Recognition as an Image Segmentation challenge is addressed in Computer Vision literature. Wang et al. [5] developed an Apple detection system based on apple's colour. They looked into the issue of apple detection for forecasting yield.

Hung et al. [6] proposed a conditional random field-based five-class segmentation approach for almond segmentation. Using a Sparse Autoencoder, this approach learned characteristics (SAE). The CRF framework was used to implement these features, which resulted in a phenomenal segmentation performance. In a well-written paper [7], the author uses transfer learning to modify a Faster R- CNN. The early and late fusion approaches are discovered by combining RGB and NIR. Fruit recognition using the deep fruit system [7] was a big step forward in the development of DL techniques for fruit detection, gaining popularity in the CV area as a result. A different study [8] used two back propagation techniques to train neural networks on photos of Apple Gala variety trees in order to compute yield for the next seasons. Furthermore, detection in relation to camera angle [9], Scale Invariant Feature Transform (SIFT), and an upgraded ChanVese level-set model [10] are also being investigated.

III. PROPOSED WORK

A. Methodology

A set of Convolution layers and Fully Connected layers make up Convolutional Neural Networks. Feature maps (mappings of activations of different parts of an image) are created by the convolution layers, which are then pooled, flattened out, and passed on to the fully connected layers. The Convolutional Layer 1 is the first hidden layer, with 64 filters, a kernel of size (3x3), striding 1, and padding 'same' to keep the output convolution's spatial dimensions the same as the input. The Rectified Linear Units (ReLU) as an activation function and Batch normalization as a strategy for training very deep neural networks to standardize the inputs to a layer for each mini-batch are then followed. The first Convolutional Layer is traditionally responsible for capturing Low-Level characteristics like edges, colour, gradient direction, and so on. The architecture adjusts to the High-Level characteristics as well as the new layers, resulting in a complete knowledge of the images in the dataset. Spatial Dropout (Dropout for 2D Feature Maps) and L2 Kernel Regularization are two techniques for reducing overfitting and improving the network's performance by making it more robust. Following another Batch Normalization, the second Convolutional Layer comprises 128 filters and a kernel size of 3x3. While training deep neural networks, we employed Leaky ReLU as the second activation to overcome the dying ReLU problem caused by vanishing gradient. Following the activation, a Max pooling 2D layer with a pool size of (2x2) is added. Pooling is employed in Convolutional Neural Networks in addition to down sampling to make the identification of specific features in the input invariant to scale and orientation changes. As a result of the tactics used, the network is able to generalize better and is less likely to overfit the training data.

$$\sigma(z) = (1 + \exp(-z)) - 1 \qquad (1)$$

Before accessing the fully linked layers, a Flatten layer turns the 2D filter matrix into a 1D feature vector. 250 units and a ReLU activation make up the Fully Connected layer 1. To avoid overfitting, a Dropout with a probability of 0.5 is employed between the last hidden layer and the output layer, which randomly switches off 50% of the nodes per epoch step. Finally, the Fully Connected Layer 2 or Output Layer, where Softmax Classifier activation is used to forecast the model's output, with the number of units equal to the number of categories.

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{k} e^{z_k}} \quad for\ j = 1, \dots k \qquad (2)$$

The optimizer in Deep Neural Networks is critical for calculating the model's error function. We utilised the Adam [15] optimizer for training our model. Adam represents Adaptive Moment Estimation, which calculates hyper-parameter adaptive learning rates. Adam combines the finest features of the AdaGrad and RMSProp methods to create an optimization technique for noisy issues with sparse gradients. Adam is simple to set up, and the default configuration parameters work well for the majority of problems. From a learning rate of 3e-4, 1 as 0.999, 2 as 0.999, and Epsilon 1e-8, we trained our CNN Model using Adam optimizer. A low learning rate is required to reduce the training error of the model. Adam employs a method for estimating the moments.

$$n_t = \beta_1 * n_{t-1} + (1 - \beta_1) * g_t \qquad (3)$$
$$s_t = \beta_2 * s_{t-1} + (1 - \beta_2) * g_t^2 \qquad (4)$$

The First and the Second moments are denoted by nt and st, the gradient is denoted by gt, and hyper-parameters are denoted by 1 and 2. The latest on weight:

$$w_t = w_{t-1} - n * \frac{\hat{n}}{\sqrt{\hat{s}} + \epsilon} \qquad (5)$$

In order to compute the model's performance, the error rate must be estimated. For multi-class classification issues, the categorical cross entropy cost function has been used as a loss function. [12] is the definition of the cost function:

$$C(w,b) = \frac{1}{2n} \sum_x [y(x) - a^2]^2 \qquad (6)$$

Where w stands for the network's weight accumulation, b for bias, n for total number of training inputs, a for actual output. The [11] training method aims to find a set of bias and weight that results in a low cost, in order to reduce cost C(w,b) as a function of bias and weight to a smaller degree. The optimizer can now match the global faster thanks to an artificial learning rate reduction technique [13].

B. System Architecture

The CNN architecture consists of three convolutional layers, each followed by pooling layers and two fully linked layers. Every convolutional and fully connected layer's output is subjected to the Relu non linearity. The 32*32*3 input image is filtered using 32 kernels of size 5*5*3 in the first convolutional layer. Second convolutional layer comprises 32 5*5*32 kernels. Third convolutional layer includes 64 5*5*32 kernels. With a stride of 2, all pooling layers pool over 3*3 regions. Each of the completely connected layers contains 64 neurons. The softmax classifier is then applied to the final layer.

C. Process Flow

The aspects of the procedure include picture samples, data augmentation, vgg16 pretrained model, testing set, training set, validation set, transfer learning, training, and recognition, as shown in Figure 1.
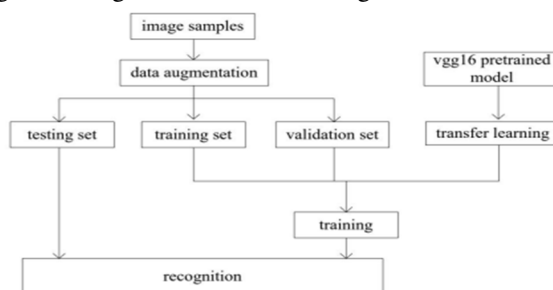

Fig.1. Process flow chart

D. Data Set

A total of 1512 photos of three different types of fruits are included in the data collection.

We used photos of three different kinds of fruits: apple, orange, and banana. We'll also categorise them as rotting or fresh. As a result, we've captured photographs of three various types of fruits, including both rotten and fresh fruits.

We used 1212 photos to train the data set and 300 images to test the data set out of the total images.

Ever image class will tend to have its own folder in a multi-category classifier like this one — "Apple," "Oranges," and so on. flow from directory()

ImageDataGenerator has a property called Flow from directory that provides a memory-efficient iterator object. Fruit images were created by recording the fruits as they rotated around a motor and then making frames. As a backdrop, white paper is placed behind the fruits. Due of the mismatch in lighting, a flood fill type method was utilised to remove the fruit from their background.

IV. EXPERIMENTAL ANALYSIS

A. Model Implementation

The regions that extract from the source pictures train the CNN. The original image's type is determined by combining the findings of each region classification. The final findings of the experiment demonstrate that the fruit recognition rate has much increased, and the proposed method may now be used to identify several fruit varieties in a photo in future study.
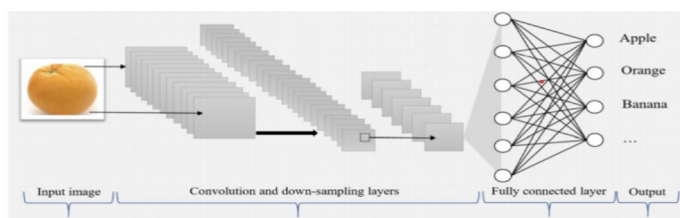

Fig.2. CNN model

Separate training, testing and validation data are created to determine the model's performance. The model is trained using the training dataset. During the training process, a validation set is used to check the model's performance, which aids in tuning the model's hyper-parameters. The test data is utilised to determine the final model's performance. There are 1512 fruit photos in total in the dataset. 1212 images are utilised for validation and 300 images are used to train the model out of the total. Following the training, random photos of three fruits (apple, orange, and banana) are used. The validation data was made up of a variety of fruit images. Below are some examples of test images:

Fig.3. Fruit images

The photos of validation data are utilised here to check the system validity. During this test, the model predicts which class the data belongs to. All validation data photos must be cropped in 256x256 sizes throughout this type of testing.

B. *Model Output*

Table.1. Epochs vs loss and accuracy.

Epochs	Loss	Val_ loss	Accuracy	Val_ accuracy
20	0.2843	0.3824	0.8969	0.8567
25	0.1407	0.5500	0.9530	0.8400
30	0.0823	0.9752	0.9752	0.8533

This CNN model has also been deployed to a Flask application. We'd have to submit a picture of the fruit, and it'd tell us what kind of fruit it is and if it's rotting or not. We have used HTML, CSS and JSON for frontend for our Flask application Website. The IDE used is VS-code.
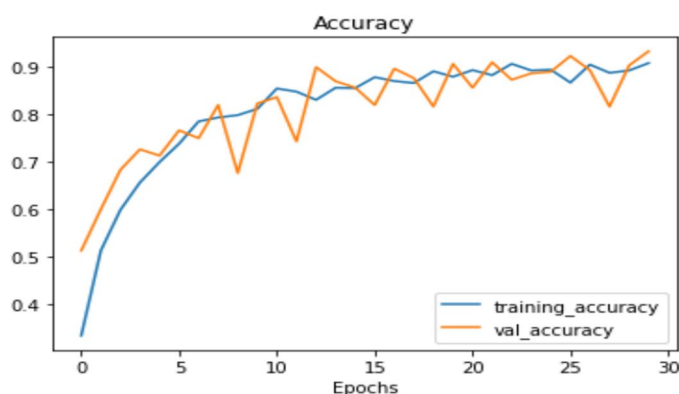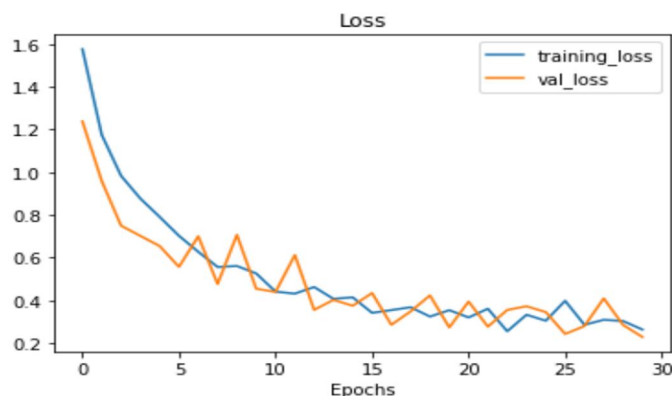

Fig.4. Epochs vs accuracy graph

Fig.5. Epochs vs Loss graph

The following graphs gives insights on the loss and accuracy of the model. The model has achieved highest accuracy of 0.9752 at epoch 30. At this accuracy, we will also be able to predict the quality of fruits- if its fresh or rotten.

C. Model Comparison

In this section, we discuss about the results that we obtained after implementing our model and comparison of our implemented model with different classification models based on different performance metrics like precision, recall and f1-score.

Table No. 2 Comparison of performance metrics of different models

Classification model	Accuracy	Precision	Recall	F1-Score
CNN + Adam optimizer	0.95	0.94	0.95	0.95
Random Forest	0.84	0.86	0.84	0.84
KNN	0.70	0.75	0.70	0.71
Decision tree	0.66	0.66	0.66	0.66
Gaussian Naive Bayes	0.73	0.75	0.73	0.73

Thus, from the confusion matrix, and comparison table, we can clearly observe that CNN model with Adam optimizer gives the best results having highest accuracy of 95%. The other models had decent accuracies but didn't give best results as compared to our model. Also, its TP, TN, FP, FN were the best among the rest of the models. We have obtained highest F1-Score of 0.95 for our CNN model which suggests high values for both precision and recall. Hence, after analysis of the results and values, we can say that our fruit classification system using CNN model proves to be the best among other models.

V. CONCLUSION

The goal of this research is to employ a Convolutional Neural Network (CNN) to classify and detect Mixed Fruits with the highest level of accuracy possible in order to help people with Down syndrome. Transfer learning technology will be used to minimise training parameters and increase training speed, based on the vgg 16 pre-training model. This research aims to produce test results that are very accurate in fruit recognition and can be used to a wide range of fruits and vegetables. As a result, it is beneficial to the smart realisation of the fruit and vegetable sales market.

This project aims to kick-start an area that is currently under-explored. During this study, we had the opportunity to look into a fraction of the profound learning algorithms and identify strengths and weaknesses. We learned about deep learning and developed a product that can recognise fruits from photographs. We believe that the findings and tactics presented in this paper can be applied to a larger project. One of the main objectives, in our opinion, is to improve the neural system's precision. This includes delving further into various aspects of the system's structure.

REFERENCES

[1] T. Schaul, S. Zhang, and Y. LeCun, "No more pesky learning rates," in International Conference on Machine Learning, 2013, pp. 343-351.

[2] Rathod, Nilesh, and Sunil Wankhade. "Review of Optimization in Improving Extreme Learning Machine." EAI Endorsed Transactions on Industrial Networks and Intelligent Systems 8.28 (2021): e2

[3] Rathod, N., & Wankhade, S. (2021). An Enhanced Extreme Learning Machine Model for Improving Accuracy. In Proceedings of Integrated Intelligence Enable Networks and Computing (pp. 613-621). Springer, Singapore.

[4] Gaikwad, Sharmila. "Study on Artificial Intelligence in Healthcare." 2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS). Vol. 1. IEEE, 2021.

[5] Q. Wang, S. Nuske, M. Bergerman, and S. Singh, "Automated crop yield estimation for apple orchards," in Experimental robotics, 2013, pp. 745-758.

[6] C. Hung, J. Nieto, Z. Taylor, J. Underwood, and S. Sukkarieh, "Orchard fruit segmentation using multi-spectral feature learning," in 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2013, pp. 5314-5320.

[7] I. Sa, Z. Ge, F. Dayoub, B. Upcroft, T. Perez, and C. McCool, "Deepfruits: A fruit detection system using deep neural networks," Sensors, vol. 16, p. 1222, 2016.

[8] H. Cheng, L. Damerow, Y. Sun, and M. Blanke, "Early yield prediction using image analysis of apple fruit and tree canopy features with neural networks," Journal of Imaging, vol. 3, p. 6, 2017.

[9] T. F. Chan and L. A. Vese, "Active contours without edges," IEEE Transactions on image processing, vol. 10, pp. 266-277, 2001.

[10] J. Hemming, J. Ruizendaal, J. Hofstee, and E. van Henten, "Fruit detectability analysis for different camera positions in sweetpepper," Sensors, vol. 14, pp. 6032-6044, 2014.

[11] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980, 2014.

[12] M. A. Nielsen, Neural networks and deep learning vol. 25: Determination press USA, 2015.

[13] T. Schaul, S. Zhang, and Y. LeCun, "No more pesky learning rates," in International Conference on Machine Learning, 2013, pp. 343-351.

[14] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, et al., "Backpropagation applied to handwritten zip code recognition," Neural computation, vol. 1, pp. 541-551, 1989.

[15] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, 2014, pp. 1725-1732.

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in Advances in neural information processing systems, 2012, pp. 1097-1105.

INTERNATIONAL JOURNAL
FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)