Sean Fabrega
Isabel Silverman
13 May 2022

AIT-Deep Learning Final Project: Fruit Image Classification

I. Introduction

We trained a convolutional neural network to recognize different fruits from images. We used two models and two datasets to train each of the models. There are many previous examples of fruit classification that we used as a base.

Grocery stores could use this CNN to identify different fruits during the checkout process. Instead of a barcode, cameras in the checkout machine could identify the type of fruit, weigh it, and then charge the customer accordingly. This would speed up the process of checkout and reduce the amount of wasteful packaging needed for produce. It could be useful to assist people with visual impairments to identify fruit using a mobile application on a smartphone or similar device.

II. Datasets

The first dataset contains 44406 images of 15 different types of fruit with a resolution of 320 x 258 pixels. During the collection of this dataset, the creators note that they introduced certain elements such as, “light, shadow, sunshine, pose variation” to simulate conditions in a supermarket or produce stall. Some of the images include hands partially covering the fruit. This dataset is from Kaggle, with a couple of notebooks training a deep learning model with the dataset.

The second dataset contains 90483 fruit and vegetable 100 x 100-pixel images. This dataset covers 131 different types of fruit/vegetables. The images were obtained by placing the fruits and vegetables into the shaft of a low-speed motor with a sheet of white paper behind it and recording a 20-second video using a Logitech C920 camera. There were still differences in backgrounds because of variations in the image lighting, so the creators designed an algorithm to replace all background pixels with white.

III. Previous Solutions

The most downloaded example using the Fruit Recognition dataset is “Classify 15 Fruits with TensorFlow” by Datalira (Databeru). She compares 27 pre-trained models. She finds that DenseNet201 from Keras gives the most accurate result with an accuracy of 0.9448 and val_accuracy of 0.9504. MobileNetV2 and MobileNet were not far behind with accuracies of 0.9405 and 0.9150, respectively. The rest of the tested models and their accuracies can be seen in Figure 1.

Sean Fabrega
Isabel Silverman
13 May 2022

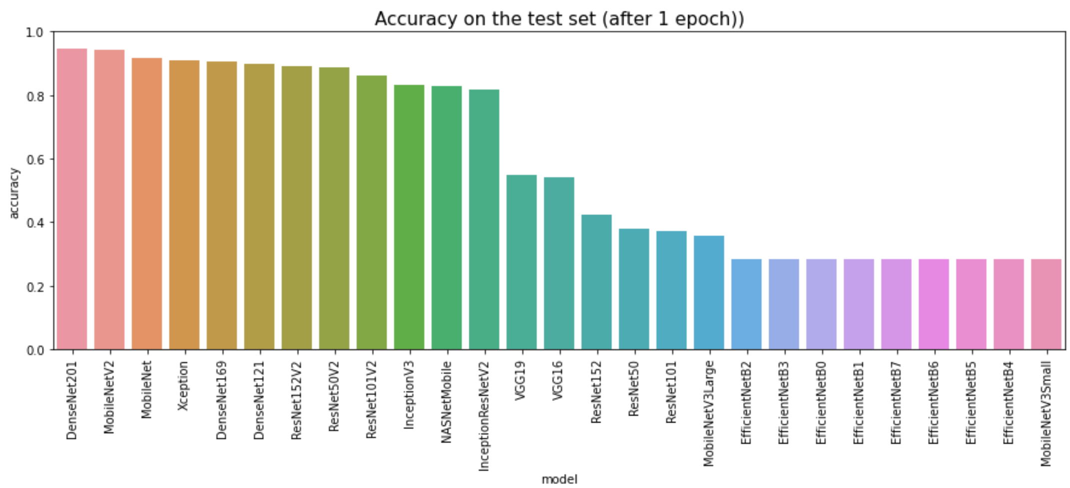

Figure 1

In June 2018, Horea Muresan and Mihai Oltean published a paper titled 'Fruit Recognition from images using deep learning' in Acta Universitatis Sapientiae, Informatica (Mureșan, Horea & Oltean, Mihai). They created a convolutional neural network with 11 layers using TensorFlow and Keras, seen in Table 2. This model was trained for 25 epochs using batches of 50 images. The model was trained using 5 different image treatments. The image treatments and their respective accuracies can be seen in Table 3.

Table 2: The structure of the neural network used in this paper.

Layer type	**Dimensions**	**Output**
Convolutional	5 x 5 x 4	16
Max pooling	2 x 2 — Stride: 2	-
Convolutional	5 x 5 x 16	32
Max pooling	2 x 2 — Stride: 2	-
Convolutional	5 x 5 x 32	64
Max pooling	2 x 2 — Stride: 2	-
Convolutional	5 x 5 x 64	128
Max pooling	2 x 2 — Stride: 2	-
Fully connected	5 x 5 x 128	1024
Fully connected	1024	256
Softmax	256	131

Table 3: Results of training the neural network on the fruits-360 dataset.

Scenario	**Accuracy on training set**	**Accuracy on test set**
Grayscale	100%	95.25%
RGB	**100%**	**98.66%**
HSV	99.99%	96.09%
HSV + Grayscale	99.99%	96.68%
HSV + Grayscale + hue/saturation change + flips	99.98%	96.44%

Sean Fabrega
Isabel Silverman
13 May 2022

IV. Proposed Method

We used an ImageDataGenerator to organize our datasets for training. For 'Fruit-Recognition' our process was pretty straightforward, accessing directories where the data was stored and generating datagens. We altered the size of the images to be 150 x 150 pixels and activities horizontal, vertical, and rotation data augmentation to increase the size of our dataset.

'Fruit-360' required a little more manual directory organization. We initially tried to train our model on the entire dataset, with 131 classes and almost 100,000 images. However, our model was overfitting and took a very long time to train. Looking more closely at the dataset, we decided to combine some classes and select only the fruit classes, since this dataset also contained images of vegetables. After this process, we had 25,157 images belonging to 34 classes. Then, we used the ImageDataGenerator as described above. For both datasets, we used an 80%, 10%, 10% train, validation, and test split.

We tested two base models, DenseNet201 and Inception V3. We chose DenseNet201 because it had the highest accuracy out of the 27 pre-trained models tested in "Classify 15 Fruits with TensorFlow" (Databeru). We chose Inception V3 because it worked well for the example we did in class and we wanted to see how it would work for our datasets. Using each of these as a base model, we trained two image classification deep neural networks using each of the two datasets described above. We added a Global Average Pooling layer to the base model and used an output Dense layer with nodes equal to the number of classes and softmax activation. In total, the model using InceptionV3 as the base and trained with the 'Fruit Recognition' dataset had 30,735 trainable parameters and 21,833,519 total parameters. The model using InceptionV3 as the base and trained with the 'Fruit-360' dataset had 69,666 trainable parameters and 21,872,450 total parameters. The model using DenseNet201 as the base and trained with 'Fruit Recognition' had 28,815 trainable parameters and 18,350,799 total parameters. The model using DenseNet201 as the base and trained with 'Fruit-360" had *insert* trainable parameters and *insert* total parameters. For both models, we used Adam as our optimizer and calculated loss using categorical cross-entropy. We also used early stopping to avoid overfitting.

V. Evaluation Method

First, we calculated the model's accuracy on the test data using accuracy_score from sklearn.metrics. Then we displayed a confusion matrix of the predicted values and actual labels. Finally, we tested our models on unseen images from the dataset we did not train it on. Note that the images in these datasets are quite different, so we wanted to see how accurate each model would be on unseen fruit in a different setting.

Sean Fabrega
Isabel Silverman
13 May 2022

VI. Results and Discussion

Overall, the DenseNet201 model produced a higher accuracy score and a better confusion matrix for both datasets compared to the Inception V3 model, which supports the findings of the previous solution. Using the “Fruit-Recognition” dataset, the DenseNet201 model has an accuracy score of 96.14% and the Inception V3 has a score of 89.3% on the test dataset. Training with the “Fruit-360” dataset produced an accuracy score of 96.83% with the DenseNet201 model and a score of 95.19% using the Inception V3 model on the test set. See Figure 2 for the confusion matrices for those four models.

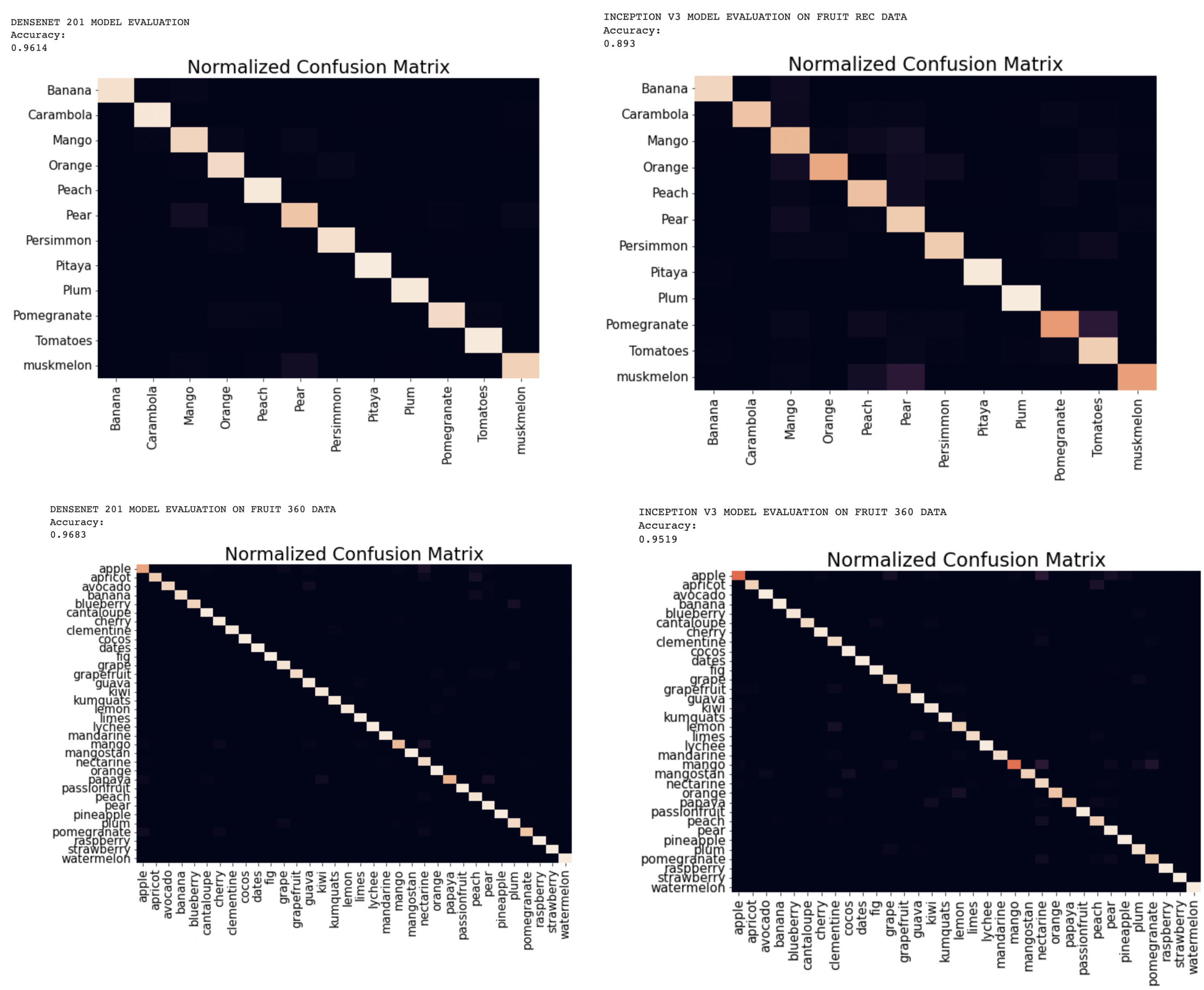

Figure 2

Sean Fabrega
Isabel Silverman
13 May 2022

We theorize that models had an easier time training and predicting with the “Fruit-360” dataset since the images have less noise. These images contain only one piece of fruit and a white background while the images in the “Fruit Recognition” dataset can contain multiple pieces of fruit and are photographed in a metallic tray. In addition, we suspect DenseNet201 has a higher accuracy score compared to Inception V3 because of either overfitting (Inception V3 has more trainable parameters) or DenseNet201 is simply better suited to fruit classification.

To further improve our accuracy, we propose adding more parameters to help the model classify fruits, specifically mass. A typical grocery store scanner already has to take the mass of produce when deciding how much an item costs, and it would also greatly assist the computer's classification of fruit. For example, oranges and mandarins look very similar, their size is the main differentiating factor. So, providing mass as another piece of information for the computer to use would improve accuracy. To expand the capabilities of our model, we propose training a model that can identify and quantify the amount of fruit in an image. This way, a grocery store scanner could also price the fruit based on quantity.

Sean Fabrega
Isabel Silverman
13 May 2022

Works Cited

Databeru. “Classify 15 Fruits with TensorFlow (Acc: 99,6%).” *Kaggle*, Kaggle, 10 Aug. 2021, www.kaggle.com/databeru/classify-15-fruits-with-tensorflow-acc-99-6.

Mureșan, Horea & Oltean, Mihai. (2018). Fruit recognition from images using deep learning. Acta Universitatis Sapientiae, Informatica. 10. 26-42. 10.2478/ausi-2018-0002.