Fruit Recognition Using Deep Learning

12112004 12111409 12110419 12110727 12112315 12112009

***Abstract*—In this study, we utilized a high-quality image dataset comprising a wide variety of fruits and vegetables. Leveraging the power of machine learning, specifically Convolutional Neural Network (CNN) algorithms, we trained a neural network model to accurately distinguish between different types of fruits. The dataset played a crucial role in providing diverse and representative examples of fruits and vegetables, facilitating the training process. By employing CNN algorithms, which are specifically designed for image classification tasks, we trained the neural network to learn and extract relevant features from the fruit images. The network's architecture and parameters were optimized to achieve high accuracy in fruit classification. Through an iterative process of training, validation, and fine-tuning, the neural network model gradually improved its ability to distinguish between various fruit types. Finally, we discuss the future work direction of this method.**

I. Introduction

This experimental project aims to leverage Convolutional Neural Network (CNN) models implemented in TensorFlow and Python to enable a robot arm to recognize and classify fruits through a camera. The primary objective of this project is to explore the feasibility of fruit recognition and evaluate the effectiveness of CNN-based approaches. The machine learning process involves training the neural network model using a high-quality image dataset comprising a diverse range of fruits. By learning from this dataset, the machine's neural network develops the ability to accurately classify fruits based on visual features.

To enhance the accuracy and performance of the fruit recognition system, several solutions are explored. Data enhancement techniques, such as image augmentation, are employed to increase the variety and quantity of training samples. Besides, the experiment involves adjusting training parameters, including learning rate, batch size, and optimizer, to optimize the neural network's performance. By fine-tuning these parameters, the model can achieve better convergence and improve its ability to distinguish between different fruit types.

In the future work, modifications to the network structure will be explored to enhance the fruit recognition system. This includes experimenting with the addition or removal of layers, exploring different architectures, and considering pre-trained models. These modifications aim to improve the model's capability to extract relevant features from fruit images, leading to more accurate classification results.

Group 9 of ME336
April, 2024

II. Section

A. Experimental Framework

1) Data Set: Based on the existing data set, the machine learns the characteristics of different fruits and vegetables on the data set and classifies them based on what it learns. The entire dataset includes more than 90,000 photos of vegetables and fruits.

Details are as below: Total number of images: 90483. Training set size: 67692 images (one fruit or vegetable per image). Test set size: 22688 images (one fruit or vegetable per image). Multi-fruits set size: 103 images (more than one fruit (or fruit class) per image) Number of classes: 131 (fruits and vegetables). Image size: 100x100 pixels.

Different varieties of the same fruit (apple for instance) are stored as belonging to different classes.

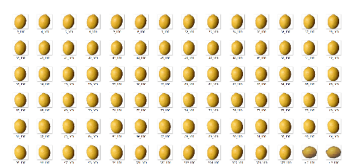

Fig. 1. Data Set

2) Experimental Data of the Reference Paper: For each scenario, the reference paper used the neural network which was trained for 25 epochs with batches of 50 images selected at random from the training set. For every epoch they calculated the accuracy using cross-validation. For testing, the paper ran the trained network on the test set. All models achieved very high accuracy on the training data set. The model trained with only RGB images obtained the best performance on the test set. A potential explanation for why the model trained on augmented data performed worse than the RGB one is that the training and test images were taken into identical lighting conditions and contained the same fruit. Thus, by augmenting the images, they are introducing variation in the training set that is not found in the test set. Conversely, training on the grayscale images produces a worse result because the conversion loses all features related to color. they further studied this problem by training and testing on just the Apple classes of images. The results were similar, with high accuracy on the train data, but low accuracy on the test data.

3) Code of CNN: Convolutional neural networks Convolutional neural networks (CNN) are part of deep learning models. Such a network can be composed of convolutional layers,

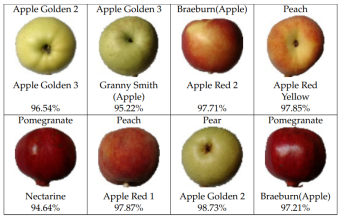

Fig. 2. Result

pooling layers, Re LU layers, fully connected layers, and loss layers. In a typical CNN architecture, each convolutional layer is followed by a Rectified Linear Unit(Re LU) layer, then a Pooling layer then one or more convolutional layers, and finally one or more fully connected layers. A characteristic that sets apart the CNN from a regular neural network is taking into account the structure of the images while processing them. Note that a regular neural network converts the input into a one-dimensional array which makes the trained classifier less sensitive to positional changes. They use multiple maps per layer with many layers of non-linear neurons. Even if the complexity of such networks makes them harder to train, by using graphical processors and special code written for them. The structure of the network uses winner-take-all neurons with max pooling that determine the winner neurons.

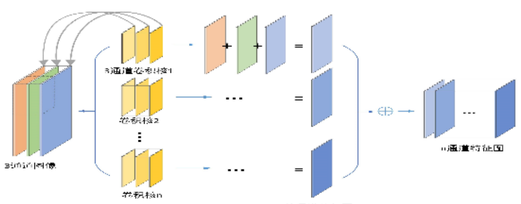

Fig. 3. Convolutional Neural Network

B. Results of Our Experiment

We first tested the integrity of the code by deleting files from the rest of the dataset, leaving only the folder ”Apple Crimson Snow”, the result is in line with the expected result.

```
Train: accuracy = 1.000000  ;  loss_v = 0.000000
Test: accuracy = 1.000000  ;  loss_v = 0.000000
```

Fig. 4. First Result

```
Total params: 5286323 (20.17 MB)
Trainable params: 5286323 (20.17 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________
None
Found 67692 images belonging to 131 classes.
Found 22688 images belonging to 131 classes.
```

Fig. 5. Data Used

Then we run the first test of the whole package of the dataset using our algorithm of CNN.

The loss rate decreases with the period and the accuracy increases with the period.

```
Epoch 2/25
1354/1354 [==============================] - ETA: 0s - loss: 4.9076 - acc: 0.0225
Epoch 2: loss improved from 9.81922 to 4.90758, saving model to output_files/fruit-360 model/model.h5
1354/1354 [==============================] - 222s 164ms/step - loss: 4.9076 - acc: 0.0225 - lr: 0.0010
Epoch 3/25
1354/1354 [==============================] - ETA: 0s - loss: 4.6877 - acc: 0.0402
Epoch 3: loss improved from 4.90758 to 4.68765, saving model to output_files/fruit-360 model/model.h5
1354/1354 [==============================] - 203s 150ms/step - loss: 4.6877 - acc: 0.0402 - lr: 0.0010
Epoch 4/25
1354/1354 [==============================] - ETA: 0s - loss: 4.4918 - acc: 0.0616
Epoch 4: loss improved from 4.68765 to 4.49177, saving model to output_files/fruit-360 model/model.h5
1354/1354 [==============================] - 204s 150ms/step - loss: 4.4918 - acc: 0.0616 - lr: 0.0010
Epoch 5/25
1354/1354 [==============================] - ETA: 0s - loss: 4.2899 - acc: 0.0850
Epoch 5: loss improved from 4.49177 to 4.28987, saving model to output_files/fruit-360 model/model.h5
1354/1354 [==============================] - 200s 148ms/step - loss: 4.2899 - acc: 0.0850 - lr: 0.0010
Epoch 6/25
1354/1354 [==============================] - ETA: 0s - loss: 4.0913 - acc: 0.1104
Epoch 6: loss improved from 4.28987 to 4.09128, saving model to output_files/fruit-360 model/model.h5
1354/1354 [==============================] - 214s 158ms/step - loss: 4.0913 - acc: 0.1104 - lr: 0.0010
Epoch 7/25
1354/1354 [==============================] - ETA: 0s - loss: 3.9078 - acc: 0.1367
Epoch 7: loss improved from 4.09128 to 3.90779, saving model to output_files/fruit-360 model/model.h5
1354/1354 [==============================] - 212s 157ms/step - loss: 3.9078 - acc: 0.1367 - lr: 0.0010
Epoch 8/25
1354/1354 [==============================] - ETA: 0s - loss: 3.7282 - acc: 0.1643
Epoch 8: loss improved from 3.90779 to 3.72817, saving model to output_files/fruit-360 model/model.h5
1354/1354 [==============================] - 213s 157ms/step - loss: 3.7282 - acc: 0.1643 - lr: 0.0010
Epoch 9/25
1354/1354 [==============================] - ETA: 0s - loss: 3.5499 - acc: 0.1896
Epoch 9: loss improved from 3.72817 to 3.54990, saving model to output_files/fruit-360 model/model.h5
1354/1354 [==============================] - 198s 146ms/step - loss: 3.5499 - acc: 0.1896 - lr: 0.0010
Epoch 10/25
1354/1354 [==============================] - ETA: 0s - loss: 3.3850 - acc: 0.2163
Epoch 10: loss improved from 3.54990 to 3.38500, saving model to output_files/fruit-360 model/model.h5
1354/1354 [==============================] - 197s 145ms/step - loss: 3.3850 - acc: 0.2163 - lr: 0.0010
Epoch 11/25
1354/1354 [==============================] - ETA: 0s - loss: 3.2379 - acc: 0.2425
Epoch 11: loss improved from 3.38500 to 3.23789, saving model to output_files/fruit-360 model/model.h5
1354/1354 [==============================] - 196s 145ms/step - loss: 3.2379 - acc: 0.2425 - lr: 0.0010
Epoch 12/25
1354/1354 [==============================] - ETA: 0s - loss: 3.1007 - acc: 0.2660
Epoch 12: loss improved from 3.23789 to 3.10073, saving model to output_files/fruit-360 model/model.h5
1354/1354 [==============================] - 198s 146ms/step - loss: 3.1007 - acc: 0.2660 - lr: 0.0010
```

Fig. 6. Evolution of Deep Learning

```
Train: accuracy = 0.754978 ; loss_v = 1.145999
Test: accuracy = 0.641617 ; loss_v = 1.656569
```

Fig. 7. Results of Deep Learning

The results we get are as below:

– a graph showing the evolution of the accuracy and loss during training

– the confusion matrix for the model

– a classification report file detailing the precision, recall and F1-score per class

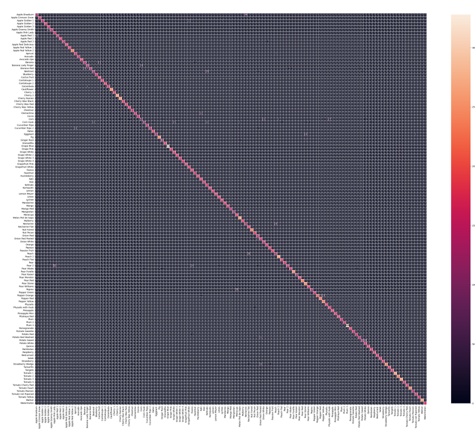

Fig. 8. Confusion Matrix

Based on the results obtained from the initial CNN test, it is apparent that there is room for improvement in the performance of the fruit recognition system. Therefore, our future work will focus on refining the algorithm to enhance its accuracy and robustness.

III. Future Work

A. Resize the Training Set

In the reference article, 67692 photos were used for training, and 100 percent of the training accuracy was obtained. We suspected the possibility of overfitting. We will try to train with fewer photos, hoping to achieve a training accuracy of 70 to 80, but a detection accuracy of 90, which will be more in line with our general rules of training and detection

Nr.	Configuration			Accuracy on training set	Accuracy on test set
1	Convolutional	5 x 5	16	100%	98.66%
	Convolutional	5 x 5	32		
	Convolutional	5 x 5	64		
	Convolutional	5 x 5	128		
	Fully connected	-	1024		
	Fully connected	-	256		

Fig. 9. One of the original training results

B. Adjust the network structure

Increasing the depth (more convolutional layers) or width (more neurons or convolutional cores) of the network can help the network learn more complex features.

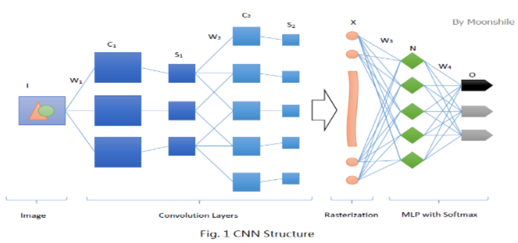

Fig. 10. Network structure

C. Adjust the training parameters

Different optimizers may perform better on specific tasks. Adjust the learning rate and learning rate decay strategy, we will also try using learning rate decay or cyclic learning rate. Allowing the model to train longer will help improve accuracy.

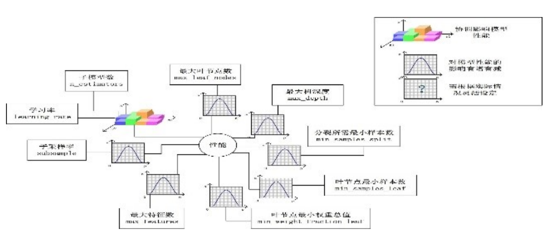

Fig. 11. Training parameters

IV. Conclusion

In our attempt to replicate an image classification experiment using TensorFlow for fruit classification, we encountered challenges in achieving the same level of accuracy as reported in the reference article. To address this issue, we conducted further investigation, including individual fruit testing and training with a smaller dataset.

Through these efforts, we gained insights into the complexities of image classification tasks and the factors that can impact replication results. While our replication results did not match the accuracy reported in the reference article, our investigation allowed us to identify potential issues and take necessary measures to address them.

These findings highlight the importance of meticulous experimental design, code validation, and careful consideration of dataset characteristics in image classification research. Replication challenges serve as a reminder of the need for transparency, reproducibility, and continuous improvement in research practices.

Moving forward, it is essential to learn from these experiences and continue refining our understanding of image classification experiments to ensure more accurate and reproducible results.

RSS Citations

Image datasets of fruits can be found from this article- Oltean, M., and Muresan, H. Fruits 360 dataset on github.

Target recognition is based on deep neural networks- Wikipedia. Google lens on wikipedia.

The framework in the project can be found in this article- TensorFlow

The following article introduces some methods of fruit image recognition- Sa, I., Ge, Z., Dayoub, F., Upcroft, B., Perez, T., and McCool, C. Deepfruits: A fruit detection system using deep neural networks. Sensors16, 8 (2016).

Rahnemoonfar, M., and Sheppard, C. Deep count: Fruit countingbased on deep simulated learning. Sensors 17, 4 (2017).

These papers introduce the deep learning of robot image recognition- Cires¸an, D. C., Meier, U., Masci, J., Gambardella, L. M., and Schmidhuber, J. Flexible, high performance convolutional neural networks forimage classification. In Proceedings of the Twenty-Second InternationalJoint Conference on Artificial Intelligence - Volume Volume Two (2011),IJCAI'11, AAAI Press, pp. 1237–1242 Schmidhuber, J. Deep learning in neural networks: An overview. CoRRabs/1404.7828 (2014)

Acknowledgment

References

[1] Bargoti, S., and Underwood, J. Deep fruit detection in orchards. In2017 IEEE International Conference on Robotics and Automation (ICRA)(May 2017)

[2] Cheng, H., Damerow, L., Sun, Y., and Blanke, M. Early yield prediction using image analysis of apple fruit and tree canopy features with neural networks. Journal of Imaging 3, 1 (2017).

[3] Li, D., Zhao, H., Zhao, X., Gao, Q., and Xu, L. Cucumber detection based on texture and color in greenhouse. International Journal of Pattern Recognition and Artificial Intelligence 31 (01 2017).

[4] Rahnemoonfar, M., and Sheppard, C. Deep count: Fruit counting based on deep simulated learning. Sensors 17, 4 (2017).

[5] Xiong, J., Liu, Z., Lin, R., Bu, R., He, Z., Yang, Z., and Liang, C. Green grape detection and picking-point calculation in a night-time natural environment using a charge-coupled device (ccd) vision sensor with artificial illumination. Sensors 18, 4 (2018).