CNNs For Image Classification

RUYU LIU
STUDENT ID: A1882974

Abstract

Image classification is an important component of computer vision and a trendy research field today. Convolutional Neural Networks (CNNs) are significant methods for addressing this task. In this paper, I proposed a basic CNN to classify 131 different classes of fruits based on the Fruits360 dataset. In addition, transfer learning techniques have also been used in the project. Two pre-trained deep CNN architectures, VGG and ResNet have been employed for the same fruit images. Besides, Stochastic Gradient Descent (SGD) has been uniformly employed as the optimization method, and Cross-Entropy is used as the loss function. After initial training and further training the models, I evaluated and compared the performance of the basic CNN, VGG16 and ResNet34 models. The best model is the VGG16, which achieved accuracy 98.98%, which is higher than the basic CNN model with accuracy 98.86% and ResNet34 model with accuracy 98.54%. Moreover, I compared my model result with those produced by other authors using other methods. Finally, I examined my proposed model by randomly testing 100 images, visualizing its classification results, and analyzing the wrong classification cases.

1. Introduction

Fruits are an essential part of our daily diet and the main source of nutrients we consume. From harvest to our consumption, fruits are identified and classified many times. For instance, grocery staff classify fruits and calculate their price [15]. Moreover, by recognizing different types of fruits, we can assess nutritional value and help consumers in choosing the best suit of fruits [8]. Therefore, fruit classification and recognition play a significant role in various practical industries.

Image classification, as a rapidly developing field of computer vision, has many mature applications in the real world such as facial recognition, image search, and medical diagnosis. Therefore, in recent years, many scholars have applied computer vision and machine learning technology to the field of fruit classification and recognition [11].

In addition to the basic CNN architecture, there are many well-trained efficient models for image classification, and we can use them by transfer learning(Fig. 1) to solve other problems [1]. These models are always built from scratch and trained on huge datasets and are able to detect edges and other features based on knowledge. Popular models for image classification tasks include VGG, ResNet, and MobileNet.

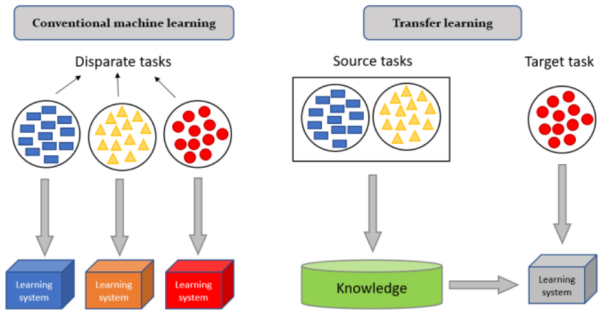


Figure 1. Transfer Learning

In this project, I will first build the basic CNN architecture model and then implement the VGG and the ResNet CNN models to classify a dataset of fruit images and compare their performance.

2. Proposed Methodology

In this section, I will briefly introduce the methods I proposed. I build a basic CNN on the Fruits360 dataset first and then transfer learning three popular image classification models VGG16, ResNet34 and ResNet50 to the dataset.

2.1. Basic CNN

CNN (Convolutional Neural Network) is a widely used deep learning architecture for image classification, both in various datasets such as MNIST [6] and CIFAR [5], as well as real-world applications like facial recognition [14] and autonomous driving [10].

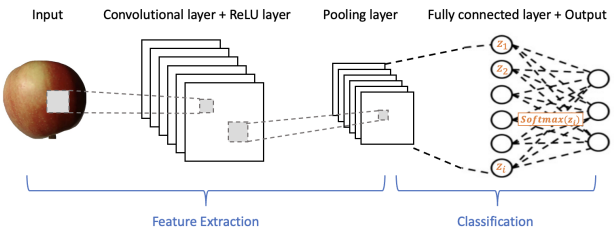


Figure 2. The minimal architecture of a basic CNN

The convolutional layer, pooling layer, ReLU (Rectified Linear Units) layer and fully connected layer are four main components of the CNN architecture. (Fig. 2)

- **Convolutional layer:** In this layer, the original input image features are passed through some small squares image filters to the next layer, so preserving both the information of each pixel and the spatial relationships between pixels. The number of filters, the size of the filters and the step size of filter movement (filter stride) are all important parameters that need to be tuned to improve the model performance.

- **ReLU layer:** It is immediately after the convolutional layer, as an activation function layer, determining how input values from the previous layer are computed in the next layer. Hence, the ReLU layer introduces non-linearity into the CNN. Other common activation functions include Sigmoid and Tanh [4].

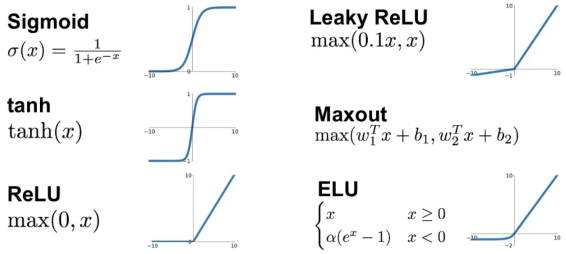


Figure 3. Different Activation Functions and their Graphs

- **Pooling layer:** The main role of this layer is to reduce the spatial dimensions of the outputted representation by the convolutional layer and ReLU layer, which can greatly reduce the amount of calculation. Pooling layers also act as image filters. The common pooling layer size is 2 by 2 with stride 2. For example, if the operation is max pooling, the image will jump 2 pixels over the whole input image feature map and retrieve the maximum value in each 2 by 2 small square. (Fig. 4)

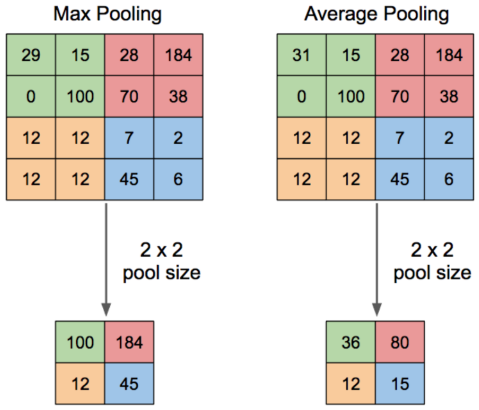


Figure 4. Two common types of pooling

- **Fully connected layer:** This is the final part of the CNN architecture. Each neuron is linked to each output of the previous layer. In multi-class neural networks, the Softmax activation function is commonly used after the fully connected layer.

$$Softmax(Z_i) = \frac{exp(Z_i)}{\sum_j exp(Z_j)} \quad (1)$$

where, Z represents the values from the neurons of the output layer.

I propose a basic CNN with 3 convolutional layers using the ReLU activation function. In order to reduce the amount of overfitting, batch normalisation layers were added after each convolutional layer. According to the Fruits360 dataset I use, the input are standard RGB images of size 100 by 100 pixels, and the output are 131 fruit classes. The structure of the basic CNN is detailed in Tab. 1.

2.2. VGG

One of the most popular CNN architecture for image classification is VGG (Visual Geometry Group). VGG, proposed by a group of Oxford researchers [13], was trained for multiple weeks on the ImageNet Dataset, which consists of 1000 classes and over 14 million 224 by 224 pixels images. The main idea of VGG is as the number of layers increases, the performance of the model increases. The structure of VGG16 in Fig. 5 shows that the layers are simple and consistent. All the convolutional layers are 3 x 3 with stride 1, all the pooling layers are 2 x 2 with stride 2. The activation functions used throughout are ReLU and a Softmax function for the output.

In this project, I employ the VGG16 model library with pre-trained weights. Additionally, I unfreeze the last layer and add the new classifier to classify the images into 131

classes.

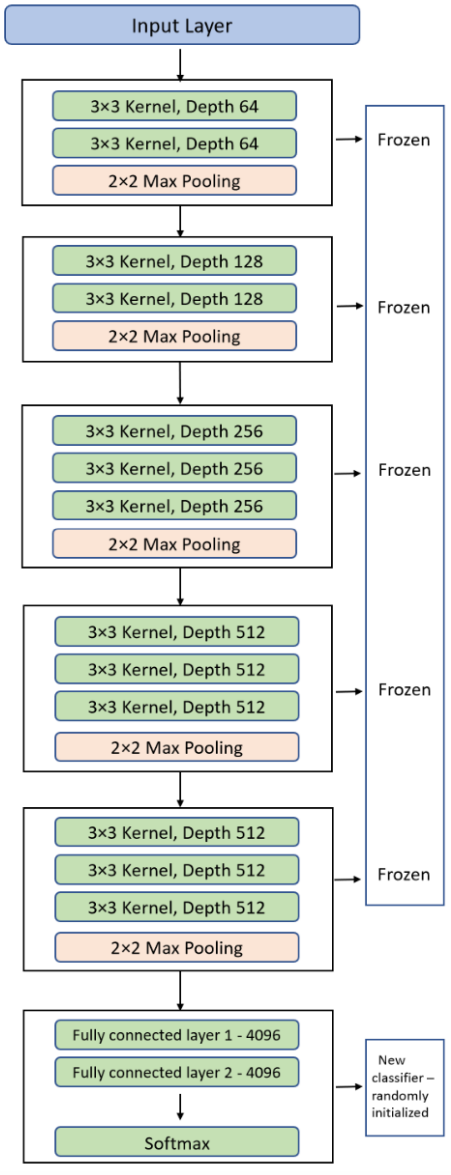


Figure 5. The structure of VGG16

2.3. ResNet

Another well-known architecture is ResNet (Residual Networks), which was first introduced in a computer vision research paper in 2015 [2]. The plain structure was inspired by the VGG architecture, but the Residual Networks proposed a new skip layer (Fig. 8a) to avoid degradation issues while stacking deeper layers. The skip layer improves gradient descent as it can locate and reference the previous layer to fine-tune the accuracy more and adjust accordingly if the accuracy begins to decrease or slow its pace. In this paper, I use the pre-trained ResNet34 and ResNet50 architecture models for the Fruits360 dataset. Both of them are unfrozen in the last layer to classify images into 131 categories.

2.4. Optimizer and Loss function

During training models, I use the Cross Entropy Loss function in back propagation algorithm to measure the error. And I apply Stochastic Gradient Descent [6] as the optimizer to minimum the loss function.

Multi-class cross-entropy loss formula:

$$H(p,q) = -\sum_{x \in classes} p(x) log\ q(x) \tag{2}$$

where $p(x)$ is the true probability distribution (one-hot) and $q(x)$ is the predicted probability distribution.

2.5. Performance Analysis Techniques

Each model is evaluated by the Accuracy on the test dataset.

Accuracy = $\frac{TP+TN}{TP+FP+FN+TN}$

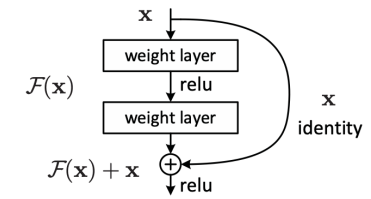
	True value Positive	Negative
Predicted Value	True Positive (TP)	False Positive (FP)
	False Negative (FN)	True Negative (TN)

Table 2. Confusion Matrix

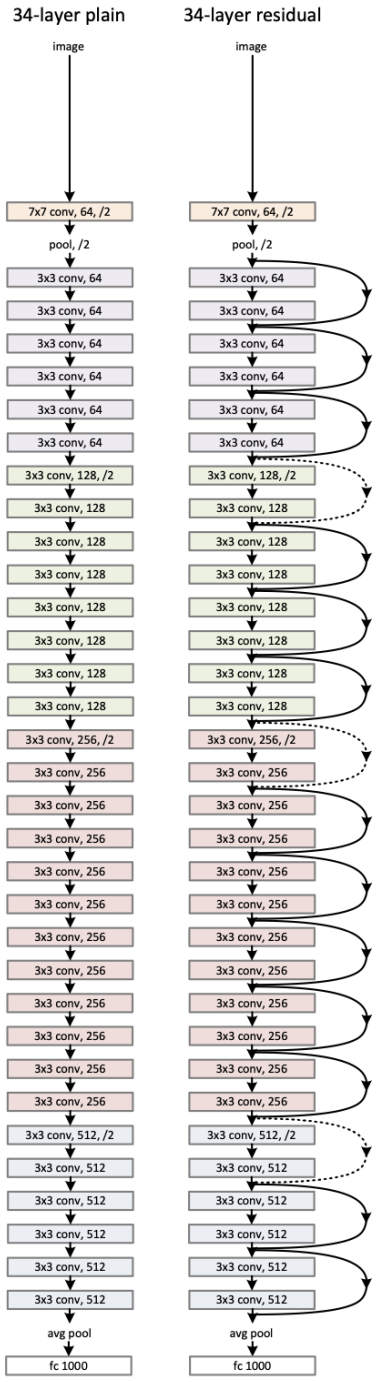
3. Experimental Result

3.1. Data

In this project, the dataset used is the Fruits360 [7], which contains a total of 90,483 images. The dataset contains 131 different types of fruits, about 70% of which are used for training (67,692 images) and 30% for testing (22,688 images). To create this dataset, the authors captured short videos of each fruit rotating continuously for 20 seconds. Frames from the video are then extracted as images, and a specific algorithm is used to separate the fruits in each image and replace the background with white. Finally, each image in the dataset is 100 x 100 pixels. A few of the images are shown below in Fig. 7.



(a) A single residual block with skip connection



(b) The demonstration of ResNet34

Figure 6. The structure of ResNet

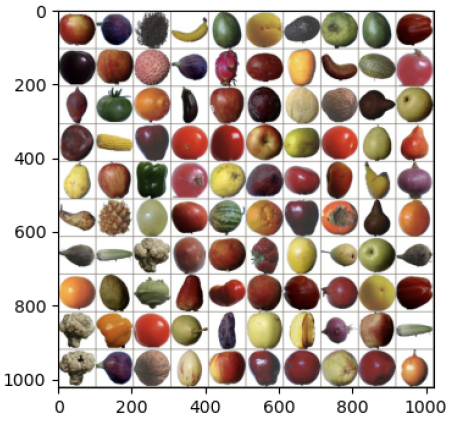


Figure 7. Images from the Fruits360 dataset

3.2. Model Performance

Initial Training

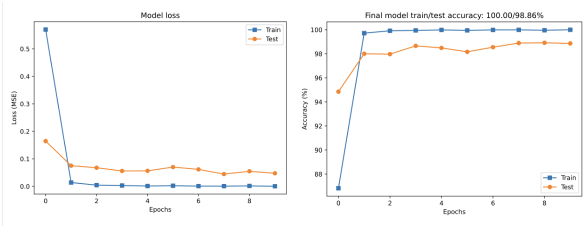
Due to computational constraints, I initially set the batch size to 100 and the learning rate to 0.01. I trained the basic CNN, VGG16, ResNet34, and ResNet50 models for 10 epochs each. For VGG and ResNet models, I only unfrozen the last layer of each model at first. The performance of these models is shown in Fig. 8.

From Tab. 3, the basic CNN performs well and achieves a test accuracy of 98.86%. The number of parameters being trained in the model is almost 7 million. The depth of the model is 6, which is the smallest of four models, so the training time of the basic CNN is the shortest. As shown in Fig. 8a, the basic CNN model does not over-fit, but the training loss has stabilized and is no longer decreasing, the test accuracy is no longer increasing, indicating that further epochs are not necessary. To further optimize the basic CNN model, I considered trying different batch sizes and adding image augmentation such as flipping and color changing the images.

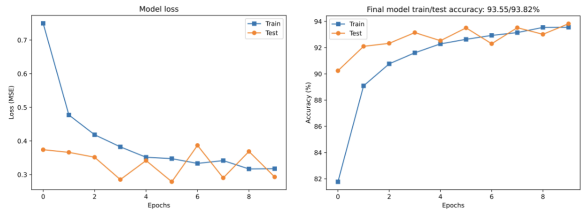
Figure Fig. 8b shows that the performance of the VGG16 model is not very good, with a test accuracy of only 93.82% after 10 epochs. This may be due to freezing all layers during training, we only trained 536707 parameters, some information has been lost. Therefore, to enhance the VGG16 model, I considered unfreezing all layers based on the current model, while reducing the learning rate to 0.001, and continuing to train for additional epochs.

The two ResNet models perform significantly better than VGG16, especially the ResNet34 model achieves a test accuracy of 97.45%. In the same way as VGG, I considered unfreezing the previously frozen layers of ResNet for further optimization.

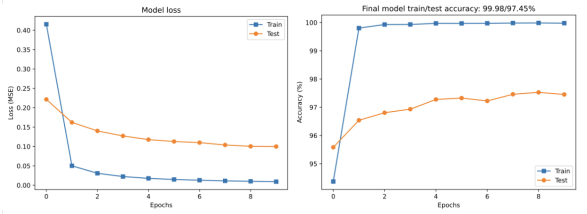
Due to computational limitations again, I only selected to further train VGG16 and ResNet34 model for additional 3 epochs.



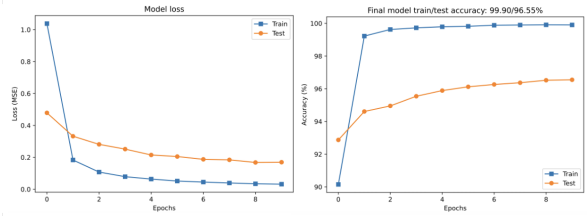
(a) Basic CNN



(b) Pre-trained VGG16



(c) Pre-trained ResNet34



(d) Pre-trained ResNet50

Figure 8. The train/test loss and accuracy of four models

Model	Accuracy	Training time(s)	Trainable Parameters	Depth
Basic CNN	98.86%	5503	7175523	6
VGG16	93.82%	32494	536707	16
ResNet34	97.45%	43001	67203	34
ResNet50	96.55%	72954	268419	50

Table 3. The Models performance after Initial Training

Further Training

After further optimising three models, the results are displayed in Tab. 4. In the basic CNN model, the batch size does not significantly affect the model's performance. Additionally, after adding data augmentation, there was no improvement in test accuracy.

After training with all layers unfrozen for 3 additional epochs, the test accuracy of the VGG16 model was significantly improved to 98.98%, while the test accuracy of ResNet34 was improved to 98.54%.

Model	Accuracy	Total Training time(h)	Trainable Parameters
Original Basic CNN	98.86%	1.5	7175523
Basic CNN (batchsize 32)	98.69%	1.8	7175523
Basic CNN (data augmentation)	96.14%	1.9	7175523
Improved VGG16	98.98%	14.2	134797251
Improved ResNet34	98.54%	15.7	21351875

Table 4. The Models performance after Further Training

I think that VGG and ResNet models, taking ten times longer to train than the 6-layer basic CNN model, did not significantly outperform the basic CNN due to the following reasons. Firstly, the Fruits360 dataset consists of fruit images with a pure white background, which is different from the noisy background in their original training datasets. In addition, the images are all fruits and with pure backgrounds which are relatively simple and may not require very deep models.

After further training the models, based on the results of all the models I tried, I proposed the final model as the VGG16 model. The test accuracy of the model is 98.98%. Many researchers have previously worked various models on the Fruits360 dataset with different numbers of the fruits classes. Tab. 5 compares some of their work results with the performance of my proposed model. It can be seen that my model has a good accuracy.

Authors	Method	Classes of the Dataset	Accuracy
[9]	CNN with Stochastic Gradient Descent with Momentum	48	98.08%
[12]	Customized VGG16	72	99.27%
[3]	Customized MobileNet	81	98.06%
[11]	Cascaded-ANFIS	131	98.36%
My model	Pre-trained VGG16	131	98.98%

Table 5. Comparative study of related research works with the Fruits360 data set

Examine the misses

Finally, I randomly tested 100 images from the Fruits360 test dataset using the model I proposed and visualized the classification results, as shown in Fig. 9. The gray images

represent correct classifications. However, the model incorrectly classified one image of Corn as Physalis with Husk and another image of Pear 2 as Onion White. Further analysis in Fig. 10 found that the image of Physalis with Husk is indeed very similar to Corn, and there are also images in Onion White that are very similar to the Pear 2 image. This observation shows that when classifying highly similar fruits, my model may occasionally make wrong classifications.

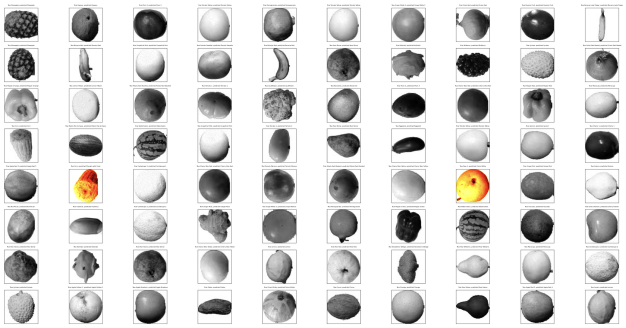


Figure 9. Random test 100 images

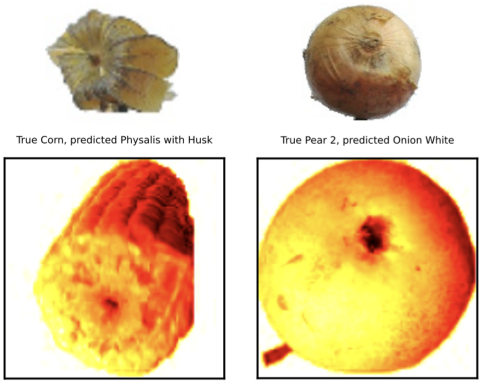


Figure 10. Wrong Classification Cases

4. Future Work

This project applies CNN to the Fruits360 dataset. I initially constructed a basic CNN model on the training set, along with transfer learning three famous pre-trained CNN architecture models VGG16, ResNet34, and ResNet50. I proposed the basic CNN model with 3 convolutional layers followed by 3 fully connected layers. Each convolutional layer is followed by a ReLU layer, a batch normalization layer and a 2x2 with stride 2 max-pooling layer. After training these four models for 10 epochs, the test accuracy of the basic CNN model reached 98.86%, while VGG and ResNet34 achieved 93.53% and 97.45%. Following the results and analysis, I decided to further train the basic CNN, VGG16, and ResNet34 models.

For the basic CNN, I tried to decrease the batch size and add data augmentation. For the VGG and ResNet, I unfroze all layers and reduced the learning rate, continuing training for an additional 3 epochs. After further training, the test accuracy of the VGG16 model and ResNet34 model improved to 98.98% and 98.54% respectively. However, reducing the batch size and adding data augmentation was not useful to the basic CNN model. In the end, I proposed the VGG16 model as our final model.

It's worth noting that, when compared to the results of other researchers, my model's accuracy is not the highest. There are several aspects in which my project can be further improved. Firstly, due to computational limitations, I had to limit the number of training epochs. For example, although the training loss of ResNet34 was still decreasing, I had to stop training. Future work could involve training the models for the enough number of epochs. Secondly, adjusting the parameters of the basic CNN model may improve the performance of the model, such as increasing the depth and tuning the number of filters in the convolutional layers. Lastly, further exploration of transfer learning with more alternative models for comparison may be helpful, such as MobileNet and InceptionV3.

5. Code

Here is my code link for this paper: GitHub link `https://github.com/ChrisLRY/Deep-Learning/blob/62540e7c9e3d717f0d065caa5a574216a3cf2e14/Assignment2_Demo.ipynb`.

References

[1] Henry C. (Henry Carlton) Ellis. The transfer of learning / henry c. ellis. In *The transfer of learning*, 1967 - 1965. 1

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 3

[3] Ziliang Huang, Yan Cao, and Tianbao Wang. Transfer learning with efficient convolutional neural networks for fruit recognition. In *2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, pages 358–362, 2019. 5

[4] Shruti Jadon. Introduction to different activation functions for deep learning. *Medium, Augmenting Humanity*, 16, 2018. 2

[5] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research). 1

[6] Yann LeCun and Corinna Cortes. The mnist database of handwritten digits. Accessed August 2014. 1, 3

[7] Horea Mureşan and Mihai Oltean. Fruit recognition from images using deep learning. *Acta Universitatis Sapientiae. Informatica*, 10(1):26–42, 2018. 3

[8] Jean A.T. Pennington and Rachel A. Fisher. Classification of fruits and vegetables. *Journal of Food Composition and Analysis*, 22:S23–S31, 2009. 1

[9] Seda Postalcıoğlu. Performance analysis of different optimizers for deep learning-based image recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, 34(02):2051003, 2020. 5

[10] Jennifer S Raj, Khaled Kamel, and Pavel Lafata. Convolutional neural network based on self-driving autonomous vehicle (cnn). In *Innovative Data Communication Technologies and Application*, volume 96, pages 929–943, 2022. 1

[11] Namal Rathnayake, Upaka Rathnayake, Tuan Linh Dang, and Yukinobu Hoshino. An efficient automatic fruit-360 image identification and recognition using a novel modified cascaded-anfis algorithm. *Sensors*, 22(12), 2022. 1, 5

[12] Raheel Siddiqi. Effectiveness of transfer learning and fine tuning in automated fruit image classification. pages 91–100, 2019. 5

[13] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. 2

[14] Qiyu Sun and Alexander Redei. Knock knock, who's there: Facial recognition using cnn-based classifiers. *International journal of advanced computer science applications*, 13(1), 2022. 1

[15] Baohua Zhang, Wenqian Huang, Jiangbo Li, Chunjiang Zhao, Shuxiang Fan, Jitao Wu, and Chengliang Liu. Principles, developments and applications of computer vision for external quality inspection of fruits and vegetables: A review. *Food Research International*, 62:326–343, 2014. 1

Layer Type	Number of filters	Size of Feature Map	Size of Kernel	Number of Stride	Number of Padding
Image input layer		100×100×3(Channels)			
1st convolutional layer	32	98×98×32	3×3	1	0
ReLU layer		98×98×32			
Normalization layer		98×98×32			
Max pooling	1	49×49×32	2x2	2	0
2nd convolutional layer	32	49×49×32	3×3	1	0
ReLU layer		47×47×32			
Normalization layer		47×47×32			
Max pooling	1	23×23×32	2x2	2	0
3rd convolutional layer	64	21×21×64	3×3	1	0
ReLU layer		21×21×64			
Normalization layer		21×21×64			
Max pooling	1	10×10×64	2x2	2	0
1st fully connected layer	10×10×64	1024			
2nd fully connected layer	1024	512			
Softmax layer	512	131			

Table 1. The structure of basic CNN