Fruit and Vegetable Image Classification Using Convolutional Neural Networks

Adam T. Nur
ITCS-5156, Spring 2024
Dr. Minwoo Lee
April 29, 2024

Abstract

This written report focuses on Convolutional Neural Networks (CNNs) and their ability to classify images. The objective is to explore the effectiveness of different CNN architectures when it comes to classification accuracy. I had initially started a standard CNN online to start off the project and I had modified the started CNN like adding more dropout and batch normalization to prevent issues like overfitting. We evaluated our models using a large dataset filled with images of fruit. This study will go in depth into the factors that increase the performance of CNNs for image recognition.

Introduction

1.1 Problem Statement

When working with Traditional Convolutional Neural Networks (CNNs) for image recognition, certain factors like color, size, shape, and texture could cause challenges. Many different image recognition models struggle with datasets like this. They tend to overfit and work well on training images but are good with new ones. Also training these models is computationally expensive and wastes a lot of resources which is impractical.

1.2 Motivation

Identifying fruits can be useful in many different situations, a couple of examples could be in fields like retail as well as agriculture. It can also be used for other kinds of pro. Making CNNs that are better at recognizing fruit images but also help us advance how we can improve deep learning for more complex tasks.

1.3 Open Questions

This study will cover multiple different open questions in the domain of image classification:

- How can we improve CNNs to handle the wide range of natural image variations?
- What changes can we make to reduce overfitting and help CNNs work better with new data?
- Can these improved models maintain high accuracy in different environments without needing lots of extra training?

1.4 Approach Overview

To face these challenges, we have modified standard CNN architectures to include higher rates of dropout and batch normalization—techniques proven to prevent overfitting. These modified networks are tested on a large dataset of fruit images, examining how these changes affect the models' ability to adapt across different situations. We'll judge each model based on how accurately it classifies images and how consistent it is across different test scenarios.

Backgrounds

2.1 Summary of Related Research

There were many different resources I have used when exploring different approaches to improving the performance of CNNs in image classification tasks, especially in the context of image recognition. The main two are:

1. **Very Deep Convolutional Networks for Large-Scale Image Recognition:** This paper deals with how the depth of convolutional networks influences their accuracy during to the identification of large objects. When tested on large Which make use of small (3x3) convolution filters at different depths. The study observed that networks with 16 to 19 layers deep into the network performed better than previous models.
2. **Deep Residual Learning for Image Recognition:** This research introduces a residual learning framework to train deeper Neural network models to be more effective. This paper presents how residual networks have higher optimization easiness and can achieve higher greater amount of depth

2.2 Pros and Cons

Transfer learning and data augmentation have many advantages when it comes to improving CNN performance but it also comes with some drawbacks as well

Pros:

- Transfer learning allows the use of prior knowledge using pre-trained models which can save resources
- Data augmentation increases the diversity of the training data, increasing the model's generalizability

Cons:

- Transfer learning can lead to bias which would result in lower performance.
- Data augmentation techniques may generate unrealistic images and limit the model's ability to adapt to real-world scenarios.

2.3 Relation to the Main Method

Data augmentation techniques will play an important role in the main proposal of this study as it adds variety to the training dataset with different variations of fruit images. I am planning on using this technique to improve the performance of CNNs for fruit image recognition

Methods

3.1 Model Architecture

- **Standard CNN with Dropout and Batch Normalization**:

  Sequential model made up of convolutional layers followed by ReLU activation, max pooling, dropout, and batch normalization layers. Additional dropout layers (dropout rates of 0.3 and 0.4) and batch normalization layers were added to prevent overfitting.

- **Improved CNN with Dropout, Batch Normalization, and Data Augmentation**:

  Enhanced model architecture featuring increased dropout rates (dropout rate of 0.4) and batch normalization layers. Added Data augmentation techniques (shear, zoom, horizontal flip) applied using the 'ImageDataGenerator' in Keras.

3.2 Data Preprocessing

- **Rescaling**: Images rescaled to the range [0, 1].
- **Image Augmentation**: Applies multiple techniques like shear, zoom, and horizontal flip to increase dataset size and improve the model's generalization.
- **Image Loading and Conversion**: Images loaded, resized to a fixed size (100x100), and converted to arrays for model input.

3.3 Training:

Layer (type)	Output Shape	Param #
conv2d_3 (Conv2D)	(None, 98, 98, 32)	896
activation_5 (Activation)	(None, 98, 98, 32)	0
max_pooling2d_3 (MaxPooling2D)	(None, 49, 49, 32)	0
conv2d_4 (Conv2D)	(None, 47, 47, 32)	9,248
activation_6 (Activation)	(None, 47, 47, 32)	0
max_pooling2d_4 (MaxPooling2D)	(None, 23, 23, 32)	0
conv2d_5 (Conv2D)	(None, 21, 21, 64)	18,496
activation_7 (Activation)	(None, 21, 21, 64)	0
max_pooling2d_5 (MaxPooling2D)	(None, 10, 10, 64)	0
flatten_1 (Flatten)	(None, 6400)	0
dense_2 (Dense)	(None, 1024)	6,554,624
activation_8 (Activation)	(None, 1024)	0
dropout_1 (Dropout)	(None, 1024)	0
dense_3 (Dense)	(None, 131)	134,275
activation_9 (Activation)	(None, 131)	0

Figure 1: Standard CNN Model Summary

- **Base Model**:
  - Batch size: 64
  - Epochs: 100
- **Batch Normalization Model** and **Learning Rate and Dropout Model**:
  - Batch size: 64
  - Epochs: 30
- **Rest of the Models**:
  - Batch size: 32
  - Epochs: 30

Training was conducted using the `fit` function, specifying the number of steps per epoch and validation steps.

3.4 Evaluation:

Experiments

4.1 Experimental Setup

Distinct CNN models were developed and evaluated:

Table 1: Test Loss and Test Accuracy for All Models

Model	Test Loss	Test Accuracy
Standard CNN	0.14	0.96
Batch Normalization CNN	9.35	0.0065
Dropout CNN	0.32	0.91
Improved CNN	0.53	0.84
Custom CNN	0.24	0.98

- **Basic CNN Model:** Uses conventional convolutional and pooling layers followed by dense layers.
- **Batch Normalization Model:** Adds batch normalization after each convolutional layer to improve training stability.
- **Adjusted Learning Rate and Dropout Model:** Added increased dropout and a tuned learning rate for better generalization.
- **Improved Model:** Added deeper layers, augmented with dropout and different data augmentation techniques like zooming, flipping, and shearing.
- **TensorFlow-based Model:** Used similar architecture to the Improved Model, but specifically fine-tuned for performance.

Each model was compiled using categorical cross-entropy as the loss function and accuracy as the evaluation metric, with RMSprop as the optimizer.

Test Results of the Proposed Method

- The **Basic CNN Model** had moderate success, which showed the potential need for enhanced regularization strategies in order to prevent overfitting.
- The **Batch Normalization Model** showed improved training dynamics but suffered on validation. Suffered from overfitting even though the normalization.
- The **Adjusted Learning Rate and Dropout Model** had really good accuracy and generalization.
- The **Improved Model** provided high accuracy and low test loss, affirming its robustness and the positive impact of dropout and data augmentation.
- The **TensorFlow-based Model** had the highest accuracy.

Deep Analysis/Discussion About the Results

The performance in each of the models was all very different, which shows us that the design of the architecture and training strategies are important in deep learning. All the models relatively the same but changes in dropout rates, batch normalization, and data augmentation all changed their effectiveness. The best results came from models that had a good balance between complexity and regularization and applied training enhancements.

Conclusion

In this project, the classification effectiveness of different architectures of CNN on fruit images was explored based on design and training strategies. In the process, I applied common CNN models with the augmentation techniques of dropout, batch normalization, and data augmentation, which not only reduced overfitting but also did so in a way that kept computational efficiency. This project greatly deepened my understanding of how one can optimize neural networks.

References

- Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, "Deep Residual Learning for Image Recognition," DOI: 10.48550/arXiv.1512.03385, arXiv:1512.03385 [cs.CV], December 2015.
- Karen Simonyan, Andrew Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," DOI: 10.48550/arXiv.1409.1556, arXiv:1409.1556 [cs.CV], April 2015.
- "Fruit Image Classification Using CNN on Google Colab," Medium, Hacker Dawn, Accessed on: April 22, 2024.