Fruit Image Classification

Authors

Gayoung (Ellena) Ko, Sijia (Neveah) Zhan

Introduction

Our project investigates the performance of image classification models trained on relatively clean datasets when applied to real-world, noisy image data. Specifically, we utilized the Fruits-360 dataset available on Kaggle, which comprises images of fruits, vegetables, nuts, and seeds, with a predominant focus on fruits. The dataset includes several variations, and for this study, we employed both the 'fruits-360_100x100' and 'fruits-360_original-size' versions to train different models. The rationale behind this selection, along with the detailed methodology, is outlined in the following section.

Our goal was to classify everyday fruit images using our trained model to test model generalization. In order to make this project more interesting and to make our model work better, we planned multiple experiments : 1. Training data with and without data augmentation, 2. Training using three optimizers (adam, RMSprop, and SGD) for comparison, 3. Implementing a pretrained model (MobileNetV2) and comparing against our trained CNN architecture (with best optimizer). We took a trip to Target and Whole Foods grocery stores and took messy-real-world images from a variety of angles, lighting, and etc. , in order to make our project even more realistic. Our hope for the outcome of this project is that even with different lighting, plastic wrappings, and someone’s hands halfway in the frame, our model will be able to classify the fruits correctly.

Methodology

Our methodology is structured in two phases.

Method 1 utilized the ‘Fruits-360_100x100’ dataset, a clean and standardized version where all images are resized to 100×100 pixels. We performed image preprocessing steps including resizing, normalization, and various data augmentation techniques such as random flipping, rotation, zooming, translation, and noise injection. A validation set was created using a 20% split through TensorFlow.

We trained a custom Convolutional Neural Network (CNN) and evaluated its performance with and without data augmentation. Several optimizers were tested, including Adam, RMSprop, and SGD. We also compared the performance of our CNN model with MobileNetV2, a pre-trained model that leverages transfer learning. In this comparison, our CNN showed high training accuracy. However, when evaluated on the official Fruits-360 test set provided by Kaggle, the model’s accuracy dropped significantly, indicating poor generalization.

To address this, we experimented with alternative augmentation strategies but observed only marginal improvements. Further analysis revealed that the model was overfitting on the training data, as training accuracy remained high while test accuracy was considerably lower.

Method 2 involved switching to the ‘Fruits-360_original-size’ dataset, which presented more variability and noise, better simulating real-world conditions. Using the same preprocessing pipeline, we re-trained the CNN with RMSprop and again compared it with MobileNetV2. In this case, MobileNetV2 demonstrated better performance and achieved high accuracy on the original test set.

However, when applied to a separate, messier dataset we collected, MobileNetV2 consistently misclassified all images as 'apple'. This behavior was likely due to the overrepresentation of apples and their subcategories in the training data, which caused the model to overfit to that class.

To improve generalization, we fine-tuned the MobileNetV2 model by replacing its classifier layer to recognize eight real-world fruit categories. We used a stratified 80/20 train-test split per class in our custom dataset to ensure balanced evaluation, with the goal of enhancing real-world performance.

Code Acknowledgement (also mentioned in the ipynb file) : Portions of our code were inspired by DS340 homework 3 and some were adapted from ChatGPT and online resources.

Results

Final evaluation of our system

Model	Test Accuracy on Fruits-360	Test Accuracy on Fruits-360 original set	Test Accuracy on self-collected dataset
CNN	0.0093	0.9087	N/A

MobileNetV2	0.0083	0.9888	0.1957
Fine-tuned MobileNetV2	N/A	N/A	0.2500

Key findings :

- MobileNetV2 and CNN performed well on the original Fruits-360 original dataset but not on the self-collected dataset
- Our model did not perform well in real-world adaptation

Results of experiments proposed in our project proposal

Experiment 1 (training with and without data augmentation) results on Fruits-360 100x100 dataset

- no augmentation : 0.6139 - loss: 1.3894 - val_accuracy: 0.7295 - val_loss: 0.9398

- without augmentation : accuracy: 0.8977 - loss: 0.3036 - val_accuracy: 0.9655 - val_loss: 0.0894

We didn't compare no augmentation vs augmentation for Fruits-360 original set

Experiment 2 (training using 3 optimizers for comparison) results on Fruits-360 100x100 dataset

- Adam - accuracy: 0.9543 - loss: 0.1285 - val_accuracy: 0.9580 - val_loss: 0.1189
- RMSprop - accuracy: 0.9514 - loss: 0.1460 - val_accuracy: 0.9746 - val_loss: 0.0724
- SGD -accuracy: 0.9239 - loss: 0.2356 - val_accuracy: 0.9645 - val_loss: 0.1000

We didn't compare no augmentation vs augmentation for Fruits-360 original set

Experiment 3 (implement pretrained model and compare against our CNN) results on Fruits-360 100x100 dataset

- CNN Train/Validation : loss: 0.07241509109735489, accuracy: 0.9746243953704834
- MobileNetV2 Train/Validation : loss: 3.0320653915405273, accuracy: 0.41404953598976135
- CNN Test : accuracy: 0.0093 - loss: 160.9478
- MobileNetV2 Test : accuracy: 0.0083 - loss: 21.9655

Experiment 3 (implement pretrained model and compare against our CNN) results on Fruits-360 original size dataset

- CNN Test/Validation : accuracy: 0.7531 - loss: 0.7786 - val_accuracy: 0.7193 - val_loss: 1.4674
- MobileNetV2 Test/Validation : accuracy: 0.9944 - loss: 0.0257 - val_accuracy: 0.9864 - val_loss: 0.0450
- CNN Test : accuracy: 0.9087 - loss: 0.2845
- MobileNetV2 Test : accuracy: 0.9888 - loss: 0.0397

Conclusion

Our project demonstrates the challenge of applying image classification models trained on clean datasets. In conclusion, CNN and the pre-trained model performed well on the training, validation, and testing (for the Fruits-360 original dataset only); it did not work well with our dataset, meaning it failed to generalize to messy real-world fruit photos. This suggests the limitation of our model, which is trained on studio-like data. However, we were able to get a valuable outcome by fine-tuning our MobileNetV2 model and were able to improve our test accuracy. Through the three experiments, we learned the value of data augmentation, the use of different optimizations, and fine-tuning when training and testing our model as it determines the performance of the model. Further work could involve collecting and training on even messier training data to improve its performance and generalization to real-world images.