Identifying Fruits and Vegetables Using CNN and MobilenetV2

UNIVERSITY OF HOUSTON

Maryum Mateen, Sarah McReynolds, Nouhad Rizk, Shadmun Talukder Shahed, Aaron Wright

Department of Computer Science, University of Houston

Abstract

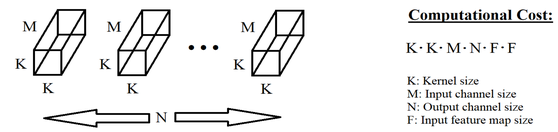
We combined machine learning and agriculture to improve crop identification and speed up agricultural sorting. We performed a comparison of Convolution Neural Network (CNN) and MobileNetV2 to see if MobileNetV2 is more effective than the widely used CNN. We used an image recognition dataset that contained folders of images from 36 different types of fruits and vegetables. There are 3825 distinct images in the dataset. We had to figure out the kernel size and stride length before we could use the CNN. Additionally, we also had to determine the best pooling method to use on the images. As a result, max pooling was chosen as the most appropriate technique for this study. We will use the Keras MobileNetV2 model to work with MobileNetV2, which has built-in convolution and dense layers. Rather than processing entire batches of input sequences at once, the model performs depth wise separable convolution with point and depth wise layers, followed by batch normalization. Our research demonstrates that utilizing MobileNetV2 is more beneficial in this process compared to using CNN.

Background

Our main goal is to optimize the fruit sorting problem most commonly encountered in agriculture [3]. Food sorting is a tedious process usually done by hand but optimizing this problem by using a machine learning algorithm not only makes it more efficient but it also helps ensure the quality in food products is preserved. Previously, CNN is the most commonly used algorithm to optimize this, however, we hypothesize that using MobileNetV2 could prove to be a better alternative because of its low latency, low computational cost (compared to standard convolution), and high accuracy [4]. In general CNN is very easy to use and implement thus making it a popular tool for image recognition problems but MobileNetV2 has also been used with comparable results.

Methods

Starting with the CNN, we built it with three convolution layers, with each layer followed by a pooling layer (using max pooling), and fully connected and dense layer at the end. For the convolutional and fully connected layer we used the ReLU activation and for the dense layer we used the softmax activation. At the compiling step we used the Adam optimizer and implemented categorical cross entropy since this accounts for a multiclass problem. This is the very standard architect for a CNN model with the parameters optimized so the results we obtained were almost exactly what we expected to obtain. As for the MobileNetV2, we will be using the Keras MobileNetV2 which has built-in convolution and dense layers. This model performs depthwise separable convolutions with two distinct stages: A filtering stage featuring depthwise convolution, and a combination stage using pointwise convolution. This is in contrast with the convolution neural network (CNN) used in this research, which processes entire batches of input sequences at once. Because MobileNetV2 adds the number of depthwise convolutions separately, this leads to a lower computational cost since multiplication in each algorithm is a more expensive operation compared to addition.

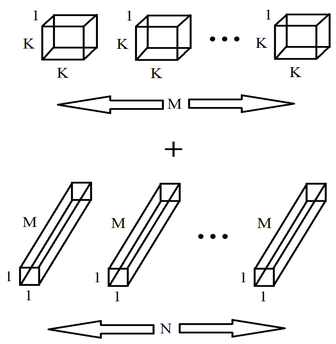


Computational Cost:

$K \cdot K \cdot M \cdot N \cdot F \cdot F$

K: Kernel size
M: Input channel size
N: Output channel size
F: Input feature map size

(a) Standard Convolution Filters in Convolution Neural Networks (CNNs)



Computational Cost:

$K \cdot K \cdot M \cdot F \cdot F + M \cdot N \cdot F \cdot F$

K: Kernel size
M: Input channel size
N: Output channel size
F: Input feature map size

Note: Depthwise seperable convolutions break the interaction between the number of output channels (N) and size of the kernel (K)

(b) Depthwise Seperable Convolutions in MobileNetV2

Results

In accordance with the data, and the performances between both the CNN and MobileNetV2, both performed well (89% vs 94% testing accuracy). However, the results from CNN carry huge caveats. First, despite using the same methodology for extracting and storing the data (using a pandas dataframe with tensorflow preprocessing for scaling) the CNN has significantly higher performance costs, using more than sixteen gigabytes of memory. To circumvent this, data was trained in incredibly smaller batches (batch size 10 compared to 32) with a smaller training set, which still had significant performance issues with this dataset(approx 994 seconds to train with GPU). Second, despite the fact CNN used more resources, the results of the training were slightly less accurate than that of MobileNetV2, all the while MobileNetV2 ran in less time, with less resources while producing better results with this dataset.

Conclusion

When compared side-by-side, MobileNetV2 is the better option compared to a CNN. With it comes better performance, optimized resource usage, and better handling of large datasets. It is clear from the experimentation and results that there would be a great benefit in implementing this algorithm with food sorting in the agriculture sector.

Future Direction

Moving ahead, there are several ways in which this technology might be utilized and developed. We concentrated primarily on being able to recognize various types of vegetables in our article. This algorithm, however, may be enhanced to detect distinct types of each specific fruit and vegetable. Furthermore, our results hopefully can lead into more research of using MobileNetV2 as an alternative to CNN especially with how well it was able to handle the amount and variety of data we had.

Acknowledgments

Thanks to the guidance and support of Dr. Rizk for giving us this opportunity and to the Hewlett Packard Enterprise Data Science Institute for having us in this showcase.