|  |
| --- |
| **Deep Learning for Image Classification of Fruits Image Dataset** |

**Ashish Tiwari**

Department of Computer Science

Georgia State University, Atlanta, GA

[atiwari5@student.gsu.edu](mailto:atiwari5@student.gsu.edu)

**Swaroop Devaraju**

Department of Computer Science

Georgia State University, Atlanta, GA

[sdevaraju1@student.gsu.edu](mailto:sdevaraju1@student.gsu.edu)

**Abstract**

The aim of this project is to build a trainable deep learning model that helps in classifying images of various fruits based upon there visual appearances. This project will be helpful in classifying and training the neural net for predicting the type of fruit by using images which are photographed from all sides. By creating a trainable deep learning model, classification of fruits will be more clearly done by the machine.

1. **Introduction**

The main objective of this project is to train a deep neural network that can identify fruits from images. This is part of a more complex project that has the target of obtaining a neural network that can identify a much wider array of objects from images. First step in creating such application is to correctly identify the objects. Currently the identification is based on a deep neural network. Such a network would have numerous applications across multiple domains like autonomous navigation, modeling objects, controlling processes or human-robot interactions. The area we are most interested in is creating an autonomous robot that can perform more complex tasks than a regular industrial robot. An example of this is a robot that can perform inspections on the aisles of stores to identify out of place items or understocked shelves. Furthermore, this robot could be enhanced to be able to interact with the products so that it can solve the problems on its own.

At the start of this project we must choose the task of identifying fruits for several reasons. On one side, fruits have certain categories that are hard to differentiate, like the citrus genus, that contains oranges and grapefruits. Thus, we want to see how well an artificial intelligence can complete the task of classifying them. Another reason is that fruits are very often found in stores, so they serve as a good starting point for a new category of project.

In the area of image recognition and classification, the most successful results were obtained using artificial neural networks. This served as one of the reasons we chose to use a deep neural network in order to identify fruits from images. Deep neural networks have managed to outperform other machine learning algorithms. They also achieved the first superhuman pattern recognition in certain domains. This is further reinforced by the fact that deep learning is considered as an important step towards obtaining Strong AI. Secondly, deep neural networks - specifically convolutional neural networks - have been proved to obtain great results in the field of image recognition. We will present a few results on popular datasets and the used methods.

**2. Problem Statement**

Firstly, this is the topic that is highly neglected, there is no research going on classifying fruit dataset using neural networks. Secondly, the shape and size of the fruits varies a lot which will create some incorrect predictions. And at last, the size of the images is comparatively small in size, which will create some error in prediction of fruits.

**3**. **Data Description**

Data Source Link: <https://www.researchgate.net/publication/321475530_Fruits_360_data-set>

Data Format: **Images**

Data Labels:

A close up of a building

Description generated with high confidence

Table 1

Fruits were planted in the shaft of a low speed motor (3 rpm) and a short movie of 20 seconds was recorded. Behind the fruits we placed a white sheet of paper as background. All marked pixels are considered as being background (which is then ﬁlled with white) and the rest of pixels are considered as belonging to the object. The maximum value for the distance between 2 neighbor pixels is a parameter of the algorithm and is set (by trial and error) for each movie. Fruits were scaled to ﬁt a 100x100 pixels image. Other datasets (like MNIST) use 28x28 images, but we feel that small size is detrimental when you have too similar objects (a red cherry looks very similar to a red apple in small images). To diversify the image dataset, a program is created that ﬁlls the background of the images with uniform colors. A class of negative images is also introduced, so that the network can classify items that are not fruits. In this category images with uniform colors are there that match the backgrounds used in the train and validation images as well as non-uniform images to simulate various backgrounds on which fruits can be placed.

**4. Image Information and preprocessing**

* Total Number of Images: 38409
* Training set size: 28736 images.
* Validation set size: 9673 images.
* Number of classes: 60 (fruits).
* Image size: 100x100 pixels.

Images in the dataset are already preprocessed for use into training and validation datasets. Both the training and test datasets have 60 categories of fruits each have several images.

A close up of an apple

Description generated with high confidence

Figure 1: An image of Apple (category: red) from the dataset

**5. Experiment**

As mentioned above, the input is a standard RGB image of size 100 x 100 pixels. We applied convolutional neural network on the image data set with two convolutional layers. The first layer is a convolutional layer of shape 5 x 5 x 3 with 128 outputs. After this we applied max pooling with a 2 x 2 filter with stride 2. The second convolutional layer is of shape 5 x 5 x 128, this gives 64 outputs. After this we feed it to a fully connected layer with 64 inputs and 32 outputs. This is in turn fed to the activation function, SoftMax. The SoftMax loss layer has 32 inputs and the output is equal to the number of classes.

We used Keras library to implement the convolutional neural network. Keras library is user friendly and simple to understand. We designed a trainable deep model by using Sequential Model in Keras library and added layers with the help of “add” function. Conv2D, MaxPooling2D, Dense functions are some of the functions that we used to implement convolution, max pooling, fully connected and SoftMax layers. We trained our model and in the first run we got our accuracy around 82%. We are working on increasing the accuracy of our model.

To be able to detect fruits from images we used the previously described neural network which was trained over 1000 iterations with batches of 150 images selected at random from the train set. Every 50 steps we calculated the accuracy using cross-validation.

Deep residual network is also the one that we wanted to try out. This network was developed by researchers at Microsoft. They used 152 layers and 8 times deeper than VGG network. We planned to use resnet50 but because of the GPU limitations we used resnet18. Unfortunately, we could run for only one epoch which gave us an accuracy of ~90%.

**6. Final Results**

We later updated the model. We applied convolutional neural network on the image data set with four convolutional layers. First convolutional layer with 32 outputs and pooling. Second convolutional layer with 64 outputs and pooling. Third convolutional layer with 128 outputs and pooling. A final layer with 256 outputs, a dropout of 25% and pooling. After this we flattened the output and added two fully connected layers of dimension 128. This was fed to a softmax layer with 60 outputs. Table 2 gives the detail idea about the model.

|  |  |  |
| --- | --- | --- |
| **Layer Type** | **Dimensions** | **Output** |
| Convolutional | 5 x 5 x 3 | 32 |
| Max Pooling | 2 x 2 – Stride 2 | - |
| Convolutional | 5 x 5 x 32 | 64 |
| Max Pooling | 2 x 2 – Stride 2 | - |
| Convolutional | 5 x 5 x 64 | 128 |
| Max Pooling | 2 x 2 – Stride 2 | - |
| Convolutional | 5 x 5 x 128 | 256 |
| Dropout | 25 % | - |
| Max Pooling | 2 x 2 – Stride 2 | - |
| Flatten | - | - |
| Fully connected | 13 x 13 x 256 | 128 |
| Dropout | 25 % | - |
| Fully connected | 128 | 128 |
| Dropout | 25 % | - |
| SoftMax | 128 | 60 |

Table 2

We ran the above model for 20 epochs, which gave us validation accuracy of ~96 % and training accuracy of ~95 % (Figure 2). Also, we got training loss of ~25 % and validation loss of ~18 % (Figure 3). In the demo the above model helped us to predict ~90 % of the fruits and predicted an image obtained via internet.

A close up of a map

Description generated with very high confidence

Figure 2

A screenshot of a cell phone

Description generated with high confidence

Figure 3

We also experimented with different optimizers (Figure 4). Adam seems to the best one.

A close up of a map

Description generated with high confidence

Figure 4

Table 3 shows the test images and the predicted images.

|  |  |  |
| --- | --- | --- |
| **Name of the test image** | **Test image** | **Predicted image** |
| Banana |  |  |
| Coco |  |  |
| Golden Apple |  |  |
| Mango |  |  |
| Pitahaya Red/Dragon fruit |  |  |

Table 3

**8. Conclusion and Future work**

We conclude by saying that our research is valid and important to the deep learning community. Also, we foresee plenty of applications. One can be amazon grocery store, where customers can take as many fruits as they please and the fruits are predicted accurately and charged for the customer.

Recognizing fruits in a basket full of fruits image is our first goal. For this to happen each fruit must be cropped from the image and predicted. Secondly, recognizing a fruit in real time i.e. in a video. This can be done via using OpenCV to access the webcam and feed the data to the model to predict the fruit. Finally, deploying our model as a web app and/or a mobile app which detects multiple fruits.

**9. References**

* Horea Muresan, Mihai Oltean, Fruit recognition from images using deep learning, Technical Report, Babes-Bolyai University, 2017
* Ming Liang, Xiaolin Hu. Recurrent Convolutional Neural Network for Object Recognition IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Boston, pp. 3367-3375, 2015.
* Jurgen Schmidhuber. Deep learning in neural networks: An overview. Neural Networks vol. 61, 85-117, 2015