International Journal of Academic Pedagogical Research (IJAPR)
ISSN: 2643-9603
Vol. 3 Issue 12, December – 2019, Pages: 16-20

Lemon Classification Using Deep Learning

Jawad Yousif AlZamily, Samy Salim Abu Naser

Department of Information Technology, Al-Azhar University, Gaza, Palestine

Abstract : Background: *Vegetable agriculture is very important to human continued existence and remains a key driver of many economies worldwide, especially in underdeveloped and developing economies.* ***Objectives:*** *There is an increasing demand for food and cash crops, due to the increasing in world population and the challenges enforced by climate modifications, there is an urgent need to increase plant production while reducing costs.* ***Methods:*** *In this paper, Lemon classification approach is presented with a dataset that contains approximately 2,000 images belong to 3 species at a few developing phases. Convolutional Neural Network (CNN) algorithms, a deep learning technique extensively applied to image recognition was used, for this task.* ***The results****: found that CNN-driven lemon classification applications when used in farming automation have the latent to enhance crop harvest and improve output and productivity when designed properly. The trained model achieved an accuracy of 99.48% on a held-out test set, demonstrating the feasibility of this approach.*

Keywords: Lemon, Classification, Deep Learning

1. INTRODUCTION:

Since the dawn of time, humans were depending on vegetable agriculture to survive, our ancestors would travel for long distances searching for food, no surprise that the first human civilizations began after the invention of agriculture, without crops will be impossible for humanity to survive. Modern technologies have given human society the ability to produce enough food to meet the demand of more than 7 billion people. However, food security still threated by many factors including plant diseases, see (Strange and Scott, 2005) .

But recent development in smartphones and computer vision would make their advanced HD cameras very interesting tool to lemon classification.

It is widely estimated that there will be between 5 and 6 billion smartphones on the globe by 2020. At the end of 2015, already 69% of the world's population had access to mobile broadband coverage. Significant impacts in image recognition were felt from 2011 to 2012. Although CNNs trained by back propagation had been around for decades, and GPU implementations of NNs for years, including CNNs, fast implementations of CNNs with max-pooling on GPUs in the style of Ciresan and colleagues were needed to progress on computer vision.(1-3)

In 2011, this approach achieved for the first time superhuman performance in a visual pattern recognition contest. Also in 2011, it won the ICDAR Chinese handwriting contest, and in May 2012, it won the ISBI image segmentation contest. Until 2011, CNNs did not play a major role at computer vision conferences, but in June 2012, a paper by Ciresan et al. at the leading conference CVPR showed how max-pooling CNNs on GPU can dramatically improve many vision benchmark records. on the same topic. In 2013 and 2014, the error rate on the ImageNet task using deep learning was further reduced, following a similar trend in large-scale speech recognition. The Wolfram Image Identification project publicized these improvements.

Some researchers assess that the October 2012 ImageNet victory anchored the start of a "deep learning revolution" that has transformed the AI industry

In this work, we show that a Deep Convolutional Neural Network (CNN) does well in classifying Lemon . In computer vision, CNNs have been known to be powerful visual models that yield hierarchies of features enabling accurate segmentation. They are also known to perform predictions relatively faster than other algorithms while maintaining competitive performance at the same time [4].

Deep Learning is an Artificial Intelligence (AI) subfield that imitates the works of a human brain in processing data and producing patterns for use in decision making. Deep learning is a subset of machine learning in artificial intelligence that has networks the skills of learning from data that is unlabeled or unstructured.

Deep Learning has grown hand-in-hand with the digital era, which has conveyed about an explosion of data in all forms and from every area of the world. This data, recognized as Big Data, is pinched from sources like social media, search engines, e-commerce platforms and more. This huge amount of data is freely accessible and can be shared through fintech applications like cloud computing. Though, the data, which normally is unstructured, is so massive that it could take years for humans to understand it and extract pertinent information. Companies understand the unbelievable potential that can result from disentanglement this wealth of information, and are progressively adapting to Artificial Intelligence systems for automated support [05-15].

Practical application of Deep Learning is fraud detection system. Using the fraud detection system mentioned above with machine learning, we can create a deep learning example. If the machine learning system created a model with parameters built around the amount of dollars a user sends or receives, the deep learning method can start building on the results offered

International Journal of Academic Pedagogical Research (IJAPR)
ISSN: 2643-9603
Vol. 3 Issue 12, December – 2019, Pages: 16-20

by machine learning. Each layer of its neural network builds on its previous layer with added data like retailer, sender, user, social media event, credit score, IP address and a host of other features that may take years to connect together if processed by a human being. Deep learning algorithms are trained to not just create patterns from all transactions, but to also know when a pattern is signaling the need for a fraudulent investigation. The final layer relays a signal to an analyst who may freeze the user's account until all pending investigations are finalized [16] .
Deep learning is used across all industries for a number of different tasks. Commercial apps that use image recognition, open source platforms with consumer recommendation apps and medical research tools that explore the possibility of reusing drugs for new ailments are a few of the examples of deep learning incorporation .

2. RELATED WORK

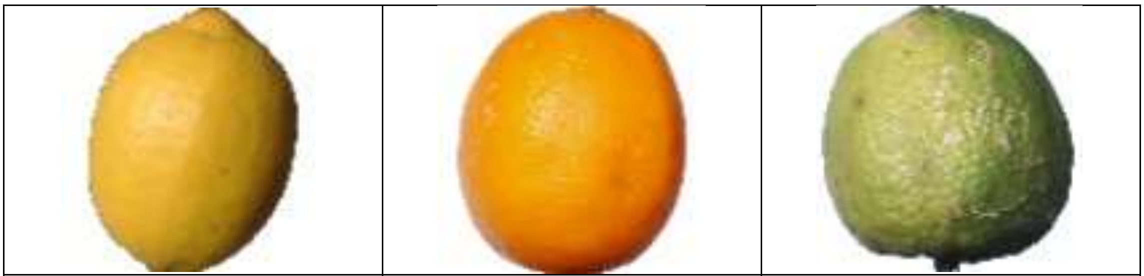
The Authors in [17] used deep learning to detect three lemon classification . They achieved a high accuracy in detecting the lemon classification.
So here, using state of the art deep learning techniques, we demonstrated the feasibility of our approach by using a public dataset of 2000 images for lemon , to produce a model that can be used in smartphones applications to identify 3 types of lemon, with an accuracy of 99.84% on a held-out test set.

3. METHODOLOGY

In this section we describe the proposed solution as selected convolutional network (ConvNet) architecture and discuss associated design choices and implementation aspects.

3.1 Dataset

The dataset used contains a set of 1976 images of approximately 1472 lemon image belonging to 3 classification . See Fig. 1 lemon classification dataset.



1. lemon	2. Lemon Meyer	3. Limes

Figure 1: Lemon Classification Dataset

We extracted our dataset from the well-known lemon dataset, which contains nearly 2,000 images . We choose to work with 1976 images Lemon ; our dataset contains samples for 3 types of lemon classification , 3 classes in total as follow :
- class (1): Lemon .
- class (2): Lemon Meyer .
- class (3): Limes .

The images were resized into 150×150 for faster computations but without compromising the quality of the data.

3.2 THE ARTIFICIAL CONVOLUTIONAL NEURAL NETWORKS: AN INTRODUCTION

In machine learning, a Convolutional Neural Network (CNN, or ConvNet) is a class of deep, feed-forward artificial neural networks, most commonly applied to analyzing visual imagery.
CNNs use a variation of multilayer perceptrons designed to require minimal preprocessing. They are also known as shift invariant or Space Invariant Artificial Neural Networks (SIANN), based on their shared-weights architecture and translation invariance characteristics.
Convolutional networks were inspired by biological processes in that the connectivity pattern between neurons resembles the organization of the animal visual cortex. Individual cortical neurons respond to stimuli only in a restricted region of the visual field known as the receptive field. The receptive fields of different neurons partially overlap such that they cover the entire visual field.
CNNs use relatively little pre-processing compared to other image classification algorithms. This means that the network learns the filters that in traditional algorithms were hand engineered. This independence from prior knowledge and human effort in feature design is a major advantage.
They have applications in image and video recognition, recommender systems and natural language processing [18] .

3.3 Building our network

International Journal of Academic Pedagogical Research (IJAPR)
ISSN: 2643-9603
Vol. 3 Issue 12, December – 2019, Pages: 16-20

So we have indeed 1482 training images, and then 432 validation images and 820 test images. In each split, there is the same number of samples from each class: this is a balanced multi classification problem, which means that classification accuracy will be an appropriate measure of success.
We our convnet will be a stack of alternated Conv2D (with relu activation) and MaxPooling2D layers.
However, since we are dealing with bigger images and a more complex problem, we will make our network accordingly larger: it will have one more Conv2D + MaxPooling2D stage. This serves both to augment the capacity of the network, and to further reduce the size of the feature maps, so that they aren't overly large when we reach the Flatten layer. Here, since we start from inputs of size 150x150 (a somewhat arbitrary choice), we end up with feature maps of size 7x7 right before the Flatten layer.
Note that the depth of the feature maps is progressively increasing in the network (from 32 to 128), while the size of the feature maps is decreasing (from 148x148 to 7x7). This is a pattern that you will see in almost all convnets.
Since we are attacking a multi classification problem, we are ending the network with a single unit (a Dense layer of size 1) and a sigmoid activation. This unit will encode the probability that the network is looking at one class or the other. See Fig.2 Sequential Model Summary .

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_5 (Conv2D)            (None, 148, 148, 32)      896
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 74, 74, 32)        0
_________________________________________________________________
conv2d_6 (Conv2D)            (None, 72, 72, 64)        18496
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 36, 36, 64)        0
_________________________________________________________________
conv2d_7 (Conv2D)            (None, 34, 34, 128)       73856
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 17, 17, 128)       0
_________________________________________________________________
conv2d_8 (Conv2D)            (None, 15, 15, 128)       147584
_________________________________________________________________
max_pooling2d_4 (MaxPooling2 (None, 7, 7, 128)         0
_________________________________________________________________
flatten_1 (Flatten)          (None, 6272)              0
_________________________________________________________________
dropout_1 (Dropout)          (None, 6272)              0
_________________________________________________________________
dense_1 (Dense)              (None, 512)               3211776
_________________________________________________________________
dense_2 (Dense)              (None, 3)                 1539
=================================================================
Total params: 3,454,147
Trainable params: 3,454,147
Non-trainable params: 0
```

Figure 2 : Sequential Model Summary.

For our compilation step, we'll go with the RMSprop optimizer as usual. Since we ended our network with a single sigmoid unit, we will use categorical crossentropy as our loss [19-20].

3.4 Data preprocessing

As you already know by now, data should be formatted into appropriately pre-processed floating point tensors before being fed into our network. Currently, our data sits on a drive as JPEG files, so the steps for getting it into our network are roughly:
1. Read the picture files.
2. Decode the JPEG content to RBG grids of pixels.
3. Convert these into floating point tensors.
4. Rescale the pixel values (between 0 and 255) to the [0, 1] interval (as you know, neural networks prefer to deal with small input values).

International Journal of Academic Pedagogical Research (IJAPR)
ISSN: 2643-9603
Vol. 3 Issue 12, December – 2019, Pages: 16-20

- It may seem a bit daunting, but thankfully Keras has utilities to take care of these steps automatically. Keras has a module with image processing helper tools, located at keras.preprocessing.image. In particular, it contains the class ImageDataGenerator which allows to quickly set up Python generators that can automatically turn image files on disk into batches of pre-processed tensors. This is what we will use here.
- It yields batches of 150x150 RGB images (shape (20, 150, 150, 3)) and binary labels (shape (20,)). 20 is the number of samples in each batch (the batch size). Note that the generator yields these batches indefinitely: it just loops endlessly over the images present in the target folder. For this reason, we need to break the iteration loop at some point .

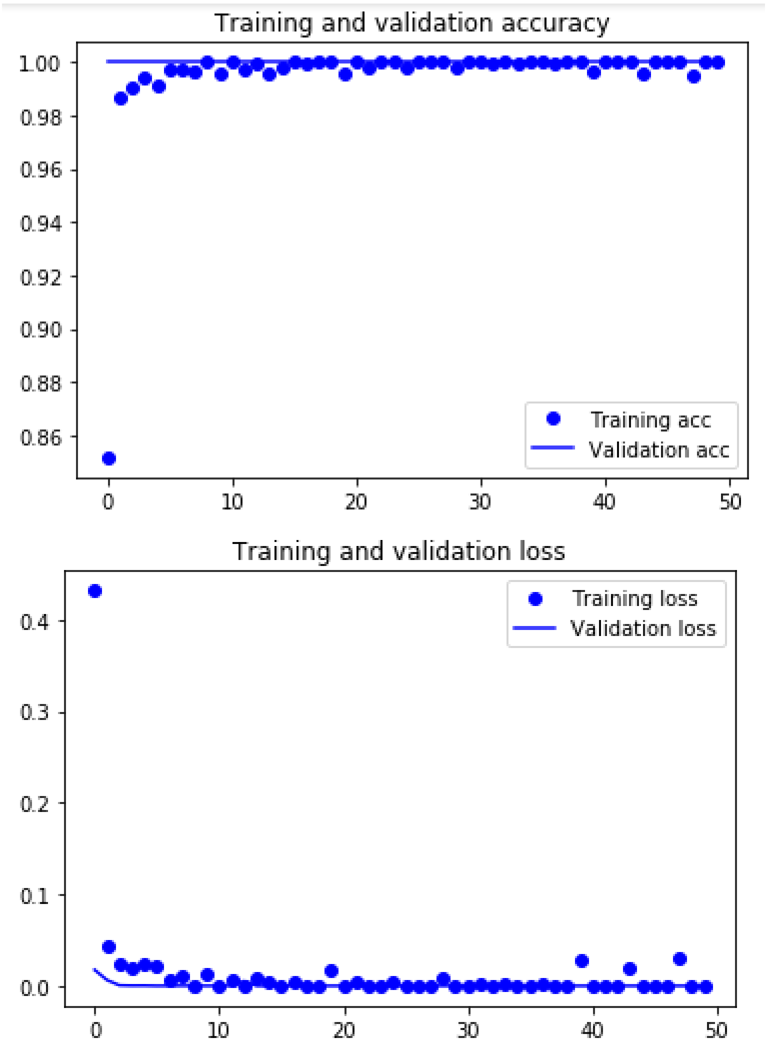


Figure 3 : The loss and Accuracy of the model.

4. EXPERIMENTS AND DISCUSSIONS

We have done experiment with the ConvNet from scratch . We used the Lemons image dataset that consists of 2000 images after resizing the images to 150x150 pixels. We divided the data into training (90%), validation (10%). The training accuracy was 99.8% and the validation accuracy was 100% as described above .

```
test_loss, test_acc = model.evaluate_generator(test_generator, steps=50)
print('test acc:', test_acc * 100)
```

```
test acc: 100.0
```

Figure 4 : Accuracy on the test data.

```
loss: 0.0047 - acc: 0.9983 - val_loss: 1.1921e-07 - val_acc: 1.0000
```

Figure 5 : Accuracy on training data.

5. CONCLUSION

More specifically, we have designed and implemented a three-class classifier that takes lemons images with a model using deep learning convolutional neural networks, and uses this model to predict the type of images of lemons.

The proposed approach achieves promising results – most notably, validation accuracy of 100% . (as seen in Figure 4)

We think that the more images we have the better the results will be.

REFERENCES

1. A Survey of FPGA-based Accelerators for Convolutional Neural Networks", NCAA, 2018 .
2. Hussain, Mahbub; Bird, Jordan J.; Faria, Diego R. (September 2018). Advances in Computational Intelligence Systems (1st ed.). Nottingham, UK.: Springer. ISBN 978-3-319-97982-3. Retrieved 3 December 2018.
3. "CS231n Convolutional Neural Networks for Visual Recognition". cs231n.github.io. Retrieved 2017-04-25.
4. Werbos, P.J. (1975). Beyond Regression: NewTools for Prediction and Analysis in the Behavioral Sciences .
5. Max A. Little, Patrick E. McSharry, Eric J. Hunter, Lorraine O. Ramig (2008), 'Suitability of dysphonia measurements for telemonitoring of Parkinson's disease', IEEE Transactions on Biomedical Engineering .
6. "NIPS 2017". Interpretable ML Symposium. 2017-10- 20. Retrieved 2018-09 12.
7. Zang, Jinliang; Wang, Le; Liu, Ziyi; Zhang, Qilin; Hua, Gang; Zheng, Nanning (2018). "Attention-Based Temporal Weighted Convolutional Neural Network for Action Recognition". IFIP Advances in Information and Communication Technology (PDF). Cham: Springer International Publishing. pp. 97–108. doi:10.1007/978-3- 319-92007-8_9. ISBN 978-3-319-92006-1. ISSN 1868- 4238.
8. Wang, Le; Zang, Jinliang; Zhang, Qilin; Niu, Zhenxing; Gang; Zheng, Nanning (2018-06-21). "Action Recognition by an Attention-Aware Temporal Weighted Convolutional Neural Network" (PDF). Sensors. MDPI
9. AG. 18 (7): 1979. doi:10.3390/s18071979. ISSN 1424- 8220