Project Presentation

Fruit Image Classification using Transfer Learning

Kusum Panchal
Arun Kumar Ramakrishnan

Overview of the Project

- This project proposes a Deep Learning based system for fruit identification.
- Fruit-classification is a special use-case of Object Classification [1] that can save tremendous amount of time while handling these highly perishable goods in large fruit markets [8].
- The proposed system is capable to identify the category of fruits (apple, banana and orange) as well as its state (fresh/rotten).
- We have used a combination of Transfer Learning [2,3], Data Augmentation [4] and Fine Tuning [5] in this project.

Problem Formulation

- Given an image ***I*** of a fruit object, to classify ***I*** in class *c* while **c** ∈ {0,1,2,3,4,5} where
  - 0: fresh-apple
  - 1: fresh-banana
  - 2: fresh-orange
  - 3: rotten-apple
  - 4: rotten-banana
  - 5: rotten-orange
- To evaluate the performance of the proposed system using Quantitative & Qualitative measures.

Proposed Approach

The proposed approach involves the following steps. Its architecture has been represented in **Fig. 1** and **Fig. 2**.

1. **Load the base-model**: start with a model pre-trained on ImageNet; load the model with the correct weights, set an input shape, and choose to remove the last layers of the model.
2. **Freeze the base-model**: freeze the base model, so that all the learning from the ImageNet dataset does not get destroyed in the initial training.
3. **Add Custom Layers to the model**: add layers to the pretrained model. We need to pay close attention to the last dense layer and make sure it has the correct number of neurons to classify the different types of fruit.
4. **Compile the Model**: compile the model with loss and metrics options. We're training on a number of different categories, rather than a binary classification problem

Proposed Approach

Continued:

5. **Data Augmentation**: augment the data to improve the dataset using Keras 'ImageDataGenerator'.
6. **Load the Dataset**: load the train and validation datasets.
7. **Train the model**: pass the train and valid iterators into the fit function and train the model by running it.
8. **Fine-tune the model**: unfreeze the model and fine-tune learning-rate, loss function and optimizer.
9. **Evaluate the model**: evaluate the performance of the proposed model using qualitative and quantitative measures.

Proposed System

- The architecture of the proposed model is shown in **Fig. 1**. It contains an input layer, a pre-trained VGG Network [6], followed by a Pooling layer and Dense layer.

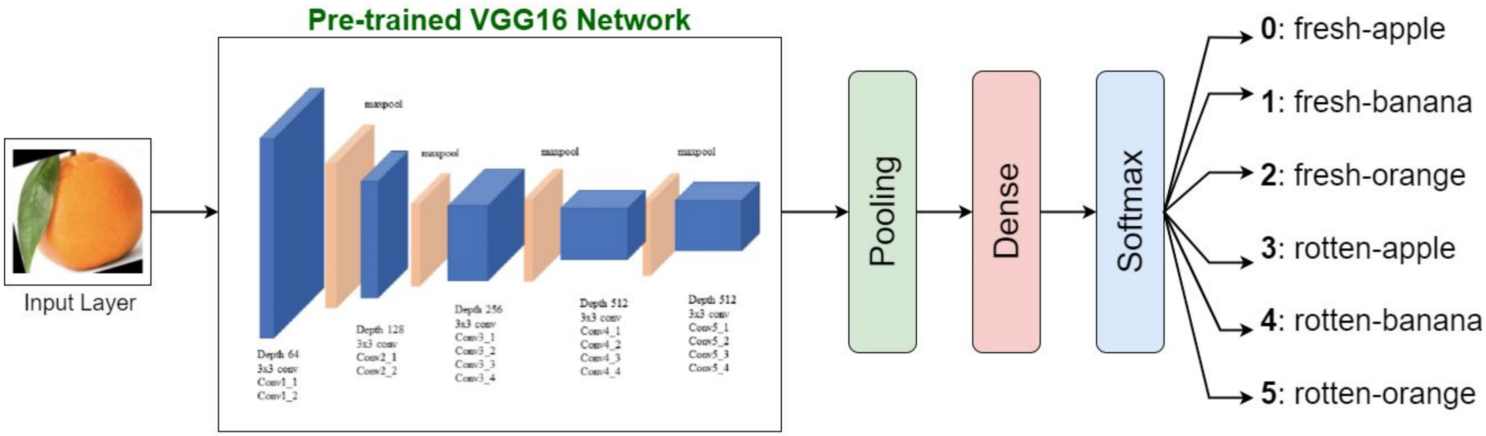

Fig. 1: Schematic Diagram of the Proposed System

Details of Proposed System's Layers

- The details of the aforementioned layers are listed below and shown in **Fig. 2**.

  -> **Input Layer**:
  Dimensions -- (None, 224, 224, 3)

  -> **Pre-trained VGG16 Network**:
  Dimensions -- (None, 7, 7, 512);
  Parameters: 14714688

  -> **Pooling Layer**:
  Dimensions -- (None, 512);
  Type -- Global Average Pool

  -> **Dense Layer**:
  Dimensions -- (None,1); Parameters: 513

```
model.summary()
```

```
Model: "model"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_2 (InputLayer)         [(None, 224, 224, 3)]     0
_________________________________________________________________
vgg16 (Model)                (None, 7, 7, 512)         14714688
_________________________________________________________________
global_average_pooling2d (Gl (None, 512)               0
_________________________________________________________________
dense (Dense)                (None, 6)                 513
=================================================================
Total params: 14,715,201
Trainable params: 513
Non-trainable params: 14,714,688
```

Fig. 2: Details of the Proposed System's Layers

Experimental Setup

- **Dataset**: The Fruits' dataset [7] containing 13,599 images of fruits of size (224, 224, 3) and 6 categories (fresh apples, fresh oranges, fresh bananas, stale apples, stale oranges and stale bananas) has been taken from Kaggle.

- **Training Strategy**:
  - 80%-20% train-test split has been used to train the proposed model.
  - The model is compiled with 'categorical_crossentropy' loss function as we have six categories.
  - We ran the proposed model five times, observed the accuracies and reported the average accuracy value.

Results: Quantitative Results

- An average accuracy of **97.37%** has been observed by the proposed system for fruit image classification.
- Confusion Matrix is shown in **Fig. 3**.

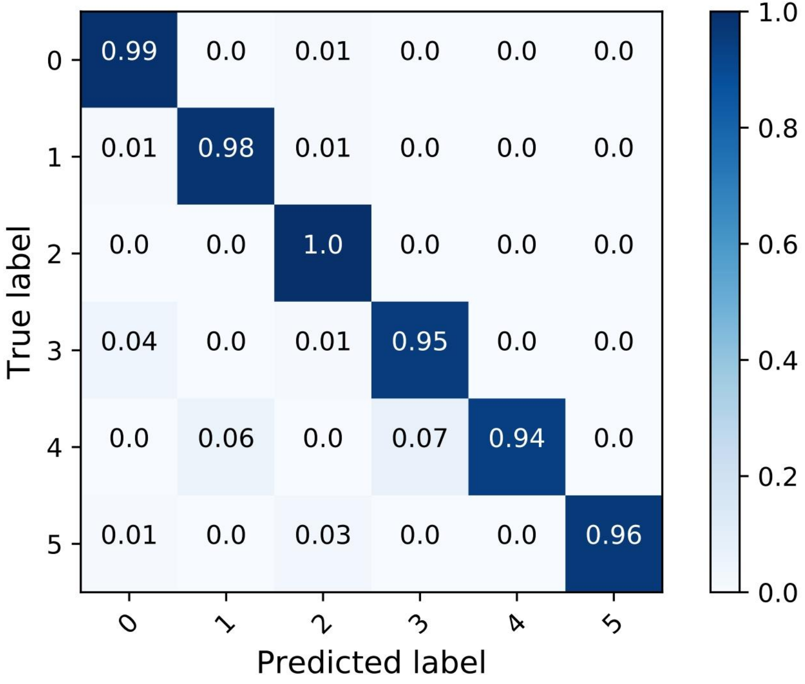

Fig. 3: Confusion Matrix

Results: Qualitative Results

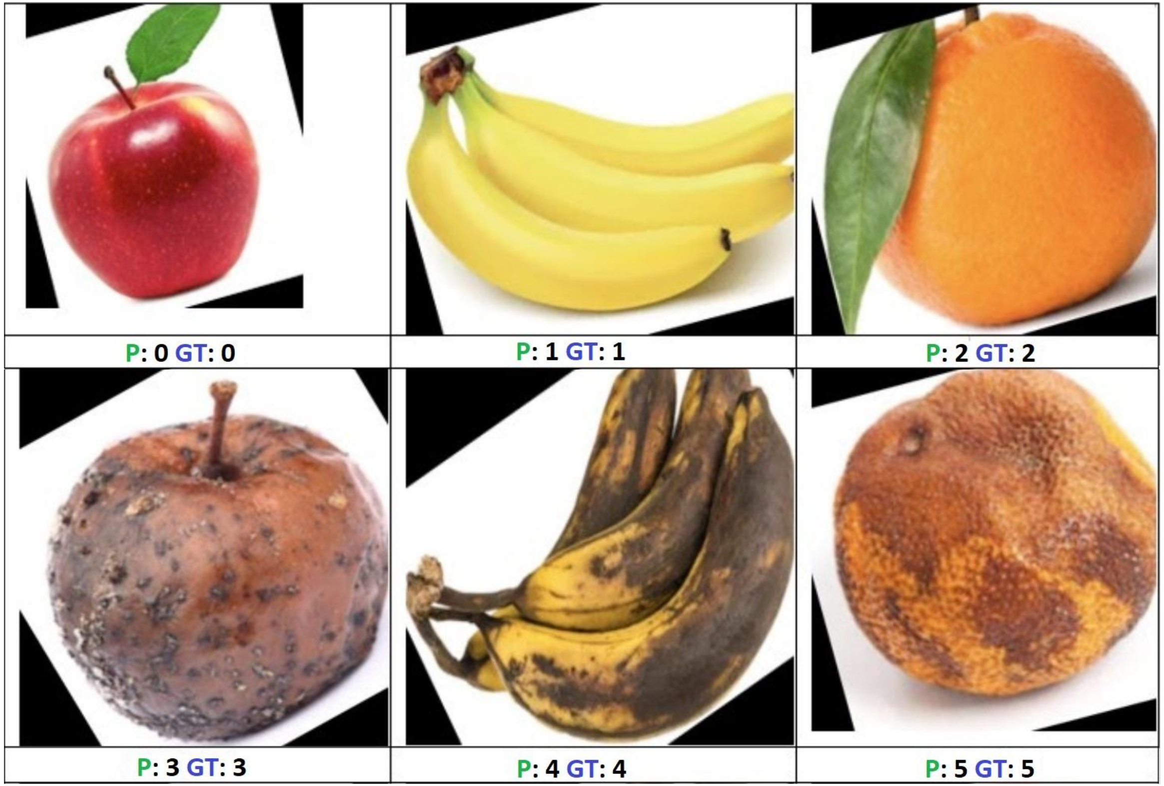

Fig. 4: Sample Results; Here '**P**' & '**GT**' denote Predicted & Ground-truth classes while 0, 1, 2, 3, 4, 5 and 6 denote the classes: 'fresh-apples', 'fresh-banana', 'fresh-oranges', 'rotten-apples', 'rotten-banana', and 'rotten-oranges'

Inference

- Overall, orange images have been identified with highest accuracy while banana images resulted in the least accuracy.
- Better results have been observed for the input images belonging to 'fresh' categories (fresh-apple, fresh-banana and fresh-orange) as compared to the images belonging to 'rotten' categories (rotten-apple, rotten-banana and rotten-orange).
- The images belonging to 'fresh' categories are never mis-classified as a 'rotten' category.
- The images belonging to 'rotten' categories are sometimes classified as a 'fresh' category. It infers that the proposed model more confidently classifies the input images belonging to 'fresh' categories; however, it sometimes identifies the rotten fruit image as a fresh one.
- The proposed system performed best with the hyper-parameter choices as 'RMSProp' for optimizer with learning rate = 0.00001 and 'CategoricalCrossentropy' for loss function.

References

[**1**] Yin Zhu† and 6 others. Heterogeneous Transfer Learning for Image Classification. Twenty-Fifth AAAI Conference on AI, held virtually February 2-9, 2021.

[**2**] Sajja Tulasi Krishna, Hemantha Kumar Kalluri . DL and Transfer Learning Approaches for Image Classification. (IJRTE), Issue-5S4, February 2019.

[**3**] Mohammad Rostami and 3 others. Deep Transfer Learning for Few-Shot SAR Image Classification. Published: 8 June 2019 by MDPI.

[**4**] Zhong, Zhun, et al. Random erasing data augmentation. Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 34. No. 07. 2020.

[**5**] Kumar, Puneet, Shalini Batra, and Balasubramanian Raman. Deep neural network hyper-parameter tuning through twofold genetic approach. Soft Computing 25.13 (2021): 8747-8771.

[**6**] Simonyan, Karen, and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv:1409.1556 (2014).

[**7**] https://www.kaggle.com/sriramr/fruits-fresh-and-rotten-for-classification

[**8**] Abdul Vahab and 3 others. Applications of Object Detection System. (IRJET), Issue-4, April 2019.

[**9**] Hands-on ML with Scikit-Learn and TensorFlow. Oreilly. 2017

[**10**] https://machinelearningmastery.com/object-recognition-with-deep-learning/

[**11**] https://www.irjet.net/archives/V6/i4/IRJET-V6I4920.pdf

[**12**] https://machinelearningmastery.com/object-recognition-with-deep-learning/

[**13**] https://www.tensorflow.org/tutorials/images/classification

Thank you

Kusum Panchal &
Arun Kumar Ramakrishnan