Effectiveness of Transfer Learning and Fine Tuning in Automated Fruit Image Classification

Raheel Siddiqi
Bahria University (Karachi Campus)
13 National Stadium Road, Karachi,
Pakistan
0092-3312118585
drraheel.bukc@bahria.edu.pk

ABSTRACT
Automated fruit image classification is a challenging problem. The study presented (in this paper) analyzes the effectiveness of transfer learning and fine tuning in improving classification accuracy for this problem. For this purpose, Inception v3 and VGG16 models are exploited. The dataset used in this study is the Fruits 360 dataset containing 72 classes and 48,249 images. The paper presents experiments that prove that transfer learning and fine tuning can significantly improve fruit image classification accuracy. Transfer learning using VGG16 model has been demonstrated to give the best classification accuracy of 99.27%. Experiments have also shown that fine tuning using VGG16 and transfer learning using Inception v3 also produce quite impressive fruit image classification accuracies. Not only is the effectiveness of transfer learning and fine tuning demonstrated through experiments, but a self-designed 14-layer convolutional neural net has also proven to be exceptionally good at the task with classification accuracy of 96.79%.

CCS Concepts
• Computing methodologies → Object recognition • Computer systems organization → Neural networks.

Keywords
Fruit image classification; Transfer Learning; Fine Tuning; Convolutional Neural Network; Fruits 360 dataset.

1. INTRODUCTION
Automated fruit classification is a challenging problem as fruits come in different shapes, color, size, texture etc. A robust and reliable solution to this problem can have variety of applications. A possible and much needed utility of such a solution can be supermarket price determination.

Fruit classification is an important task carried out by cashiers in supermarkets at the point of sale terminals. These cashiers need to recognize not only the fruit type (i.e. apple, watermelon, strawberry) but also its variety (i.e. Gala, Fuji, Granny Smith) for the purpose of price determination [1]. The use of barcodes has resolved this problem for packaged fruits but in most cases consumers want to buy fruits in loose, unpackaged form. This allows flexibility for consumers in their selection of fruits but this also means that (in such cases) fruits cannot be pre-packaged. In such situations, automated fruit classification system may be integrated with weighing machines to compute the price.

It is important to note that a lot of work in this area has focused on the external quality inspection of fruits [2, 3]. **This paper focuses on fruit type classification** rather than on classification of fruits based on external quality or visual defects.

Computer vision and image processing techniques have been extensively exploited to solve this problem [1, 4, 5, 6, 7]. The classifiers need to learn the discriminative features from the training samples. Their performance can then be accessed using test samples. But before the classifier is trained, the data need to be preprocessed.

In the past, expensive equipment such as near-infrared imaging [8], gas sensors [9], electronic nose [10] etc. have been used to scan or detect the fruits. These devices required professional operators and normally produced lower than 85% accuracy rate [5]. It should be remembered that supermarkets operate on low profit margin and therefore, hardware must be inexpensive [16]. In recent years, research has focused on taking input through less-expensive digital cameras and the accuracy results are much higher.

In this paper, the author tries to analyze the effectiveness of transfer learning and fine tuning in automated fruit image classification. For this purpose, the author uses *Inception v3* [19, 20] and *VGGNet* [21] models. The dataset used in this study is the *Fruits 360 dataset* [23] available from [22]. There are 72 classes in the dataset and in total 48,249 images (one fruit per image).

Section 2 presents literature review for image-based fruit type classification. Section 3 presents a brief overview of the Fruits 360 dataset. The dataset is used in all the experiments presented in this paper. In section 4, some state of the art computer vision techniques and models are briefly analyzed. These techniques and models are exploited in the experiments presented in the later sections. Section 5 presents experimental setups for all the four experiments. Section 6 presents results of the experiments and some analysis of the results. Section 7 concludes the paper by outlining key contributions and also presenting some directions for future research.

2. LITERATURE REVIEW
The author divides the history of image-based fruit type classification in to two eras: (i) pre-Convolutional Neural Network (CNN) era, and (ii) CNN era. The techniques and approaches of the two eras have been summarized in the following sub-sections.

ICDLT 2019, July 5–7, 2019, Xiamen, China

ACM ISBN 978-1-4503-7160-5/19/07…$15.00
https://doi.org/10.1145/3342999.3343002

2.1 Use of Hand-crafted Features and Machine Learning Techniques (i.e. Pre-CNN era)

VeggieVision [16] was the first serious attempt to develop a produce recognition system. The system consisted of an integrated scale and a digital camera. When item was placed on the scale, an image was taken by the camera. Features such as color, texture etc. was extracted and compared with stored features of various produce types. These stored features were learned during the training process. The classification accuracy was 82.6% for the top choice when the training and testing datasets were from the same store. The classification accuracy declined significantly when the training and testing datasets were from different stores [16].

Seng and Mirisaee [17] proposed another fruit recognition system that measured fruit features based on color, shape and size. Mean RGB value was used for color, measure of roundness was used for shape and area and perimeter values were used for size. These feature values were then exploited for the classification purpose using the k-nearest neighbor algorithm. Even though high accuracy rates were reported [17], the training and testing datasets were very limited.

Rocha et al. [1] proposed a new approach that relied on feature fusion and required very less number of training samples (e.g. up to 30 images) to attain high level of precision. The approach can combine many features and classifiers and is trained and tested on a dataset comprising of 15 produce types and 2633 images collected on-site. The proposed feature fusion technique has resulted in significant reduction in classification error [1].

Zhang and Wu [4] proposed another fruit classification method based on a multi-class Kernel Support Vector Machine (KSVM). The proposed method had four stages: first, the image was acquired and preprocessed using split-and-merge algorithm to remove background; second, feature space was composed by extracting features such as color, texture, shape; third, dimensions of the feature space were reduced using principal component analysis; finally, three different kinds of SVMs (i.e. Winner-Takes-All, Max-Wins-Voting and Directed-Acyclic Graph) were applied to solve the fruit classification problem. Each of these SVMs was experimented with three different kernels (i.e. linear kernel, Homogeneous Polynomial kernel and Gaussian Radial Basis kernel). SVMs using linear kernel had lowest classification accuracies. Winner-Takes-All SVM performed least efficiently and took a lot more time when compared with the other two SVMs. Max-Wins-Voting SVM with Gaussian Radial Basis kernel achieved best results with a classification accuracy of 88.2%. As for the classification speed, Directed-Acyclic Graph SVM performed most efficiently. The dataset consisted of 1,653 fruit images belonging to 18 different categories. Training was performed using stratified 5-fold cross validation. Training set size was 1,322 images and test set size was 331 images.

In an attempt to further improve accuracy, Zhang et al. [7] proposed a hybrid fruit-classification method. This method was based on Fitness-Scaled Chaotic Artificial Bee Colony (FSCABC) algorithm and Feedforward Neural Network (FNN). The FNN used had three layers: input layer, one hidden layer and the output layer. FSCABC algorithm is used for weight optimization. Traditional gradient-based optimization algorithms, such as back-propagation algorithms, were not used because they can easily get stuck in the local best. The experiments were conducted using the same dataset of [4] i.e. the dataset consisted of 1,653 fruit images belonging to 18 different categories. The performance of FSCABC-FNN based classifier on the test set was better than the KSVM classifier presented in [4]. Overall, classification accuracy for FSCABC-FNN was 89.1%.

In [6], two more machine learning based fruit classification methods were proposed. The methods were based on wavelet entropy, principal component analysis, feedforward neural network trained using FSCABC and biogeography-based optimization algorithms. Both methods produced classification accuracy of 89.5%. The methods were trained and tested on the dataset used in [4, 7]. The dataset preprocessing and feature extraction techniques were quite similar in [4, 6, 7]. The accuracy result of 89.5% was comparatively higher than the accuracy achieved using previous techniques [4, 7, 16], but still much lower than the results obtained in other application areas like medical classification, face classification etc. Better fruit image classification techniques were therefore needed.

2.2 Use of CNN and Data Augmentation for Fruit Image Classification

In [5], Zhang et al. proposed a 13-layer Convolutional Neural Network (CNN) for fruit classification. Their empirically validated approach was a decent jump from the previous efforts [1, 4, 6, 7, 16, 17] to solve the fruit classification problem. Earlier systems had two weaknesses: (1) they all used handcrafted features, and (2) the classifiers were of simpler structures and therefore lacked the ability to map the complicated features to the final classification result [5]. Zhang et al. [5] argues that CNN can help overcome these weaknesses. CNNs require very less data preprocessing (compared to other image classification algorithms) as they learn the features/filters themselves. In addition, CNNs have been successfully applied to solve various complex image classification problems such as the various medical image classification problems [11, 12, 13]. Zhang et al. [5] investigated the application of CNN on fruit image classification as it was not attempted before.

The fruit image dataset size was 3600. There were 18 fruit types and 200 images per fruit type. The images were either collected on-site through digital camera or were downloaded from the internet. Preprocessing of data involved moving the fruit to the center of the image, resizing the image to a 256x256 matrix, removing background and labeling each image manually to one of the 18 fruit types.

The training data was augmented by creating fake images [5]. This was done using five different image processing techniques: image rotation, gamma correction, noise injection, scale transform and affine transform. As a result of data augmentation, the size of the training data increased from 1800 images to 63,000 images i.e. 35 times the original.

The CNN constructed consisted of 13-layers. The combination of a convolution layer and pooling layer is referred to as 'combined layer'. Zhang et al. [5] experimented with various numbers of combined layers and they found that 4 combined layers produced the best results (in terms of accuracy). Overall accuracy was 94.94% which was at least 5 percentage points higher than the other state-of-the-art approaches [4, 6, 7, 14, 15].

Zhang et al. [5] also tested their CNN-based model on imperfect images (i.e. fruit images with complicated background, fruit images with camera not well focused, fruit images with decay and partially occluded fruit images). Overall accuracy decreased at most 5%. The performance of the model deteriorates most with complicated background images (i.e. 89.6% accuracy). On the other hand, accuracy over decay images was almost as high as the accuracy over the original clean data.

Another experiment was carried out to see the effect of data augmentation on accuracy rates [5]. Experimental results indicate that data augmentation has resulted in improvement of accuracy rates, especially when data is not clean (e.g. fruit images with occlusion or complicated background).

Wang and Chen [40] have improved the work presented in [5]. They applied an improved 8-layer deep CNN for the purpose of fruit category classification. Instead of using plain rectified linear units, parametric rectified linear units are used. Dropout layer is placed before each fully connected layer. Data augmentation was also used to help avoid overfitting. The test set classification accuracy is 95.67% which is better than all the previous techniques and experiments [4, 5, 6, 7, 14, 15].

3. FRUITS 360 DATASET

Fruits 360 dataset consists of 48, 249 fruit images (36,117 images in the training set and 12,132 images in the test set) [22, 23]. There are 72 different types of fruits in the dataset and each image contains only one fruit. For each fruit type, the training set and test set contain slightly varying number of images but in most cases around 490 training images and 164 test images are present for each fruit type. Size of each image is 100 X 100 pixels.

The images are obtained by making a short twenty seconds video of fruit while it is slowly rotated by a motor and then extracting frames/images from that video [23]. A white sheet of paper is placed as background. This process is repeated for every fruit type. Later, the background is removed from each fruit image through a dedicated algorithm. This is done because background is not uniform due to varying light conditions. Figure 1 depicts some images from Fruits 360 dataset.

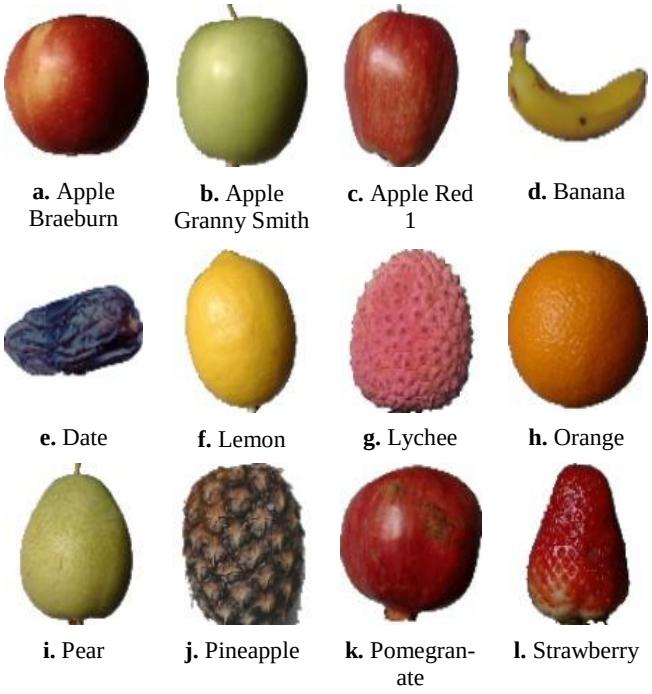

Figure 1. Some images from Fruits 360 dataset. Class label for each image is also given.

The following fruits are included in the dataset: 'Apple Braeburn', 'Apple Golden 1', 'Apple Golden 2', 'Apple Golden 3', 'Apple Granny Smith', 'Apple Red 1', 'Apple Red 2', 'Apple Red 3', 'Apple Red Delicious', 'Apple Red Yellow', 'Apricot', 'Avocado', 'Avocado ripe', 'Banana', 'Banana Red', 'Cactus fruit', 'Cantaloupe 1', 'Cantaloupe 2', 'Carambula', 'Cherry 1', 'Cherry 2', 'Cherry Rainier', 'Cherry Wax Yellow', 'Clementine', 'Cocos', 'Dates', 'Granadilla', 'Grape Pink', 'Grape White', 'Grape White 2', 'Grapefruit Pink', 'Grapefruit White', 'Guava', 'Huckleberry', 'Kaki', 'Kiwi', 'Kumquats', 'Lemon', 'Lemon Meyer', 'Limes', 'Lychee', 'Mandarine', 'Mango', 'Maracuja', 'Melon Piel de Sapo', 'Mulberry', 'Nectarine', 'Orange', 'Papaya', 'Passion Fruit', 'Peach', 'Peach Flat', 'Pear', 'Pear Abate', 'Pear Monster', 'Pear Williams', 'Pepino', 'Physalis', 'Physalis with Husk', 'Pineapple', 'Pineapple Mini', 'Pitahaya Red', 'Plum', 'Pomegranate', 'Quince', 'Rambutan', 'Raspberry', 'Salak', 'Strawberry', 'Strawberry Wedge', 'Tamarillo' and 'Tangelo'.

4. METHODS

In this section, we explore and evaluate some computer vision techniques and models based on CNN. First, a brief overview of CNN is given. Then, the processes of transfer learning and fine tuning are explained. Later, some details on pre-trained models like AlexNet, GoogleNet and VGGNet are presented. The techniques and models presented (in this section) forms the basis of the experiments presented later on in this paper. The aim of the author is to assess the effectiveness of these computer vision techniques and models in solving the automated fruit image classification problem.

4.1 Convolutional Neural Networks (CNNs)

CNNs are a kind of deep neural network [34] that are most commonly applied at solving computer vision problems such as image classification [30], object detection [35] etc. Although the applications of CNN date back to 1990s, they were not fully embraced by the computer vision community till the ImageNet competition in 2012 [36]. CNNs demonstrated spectacular success during the competition and this brought about a revolution in computer vision [36]. CNNs are now the dominant approach for almost all image classification and object detection tasks and approach human performance on some tasks [36].

CNNs are designed to recognize patterns in pixel based images with very little or no preprocessing [34]. They can recognize patterns with a very high level of variability and are also robust to distortions and simple geometric transformations. A CNN, typically, consists of an input layer, an output layer and multiple hidden layers. The hidden layers of a CNN typically consist of convolutional layers, pooling layers and fully connected layers. The author has exploited CNNs in his attempt to solve the fruit image classification problem.

4.2 Transfer Learning and Fine Tuning

Transfer learning is the process of reusing a pre-trained model trained on a large dataset, typically on a large-scale image classification task [18, 24]. Features learned by the pre-trained model can effectively act as a generic model of the visual world, and hence the model can be used for different computer vision problems. For instance, a model may be trained on ImageNet [25] (where classes are mostly animals and everyday objects) and then it is repurposed for a task like fruit image classification.

Convolutional Neural Networks (used for image classification tasks) comprise of two parts: (1) a series of convolution and pooling layers, and (2) a densely connected classifier. The first part is called the convolutional base of the model. During transfer learning, the convolutional base is retained but the trained classifier is removed and a new classifier is added which is randomly initialized. This new classifier is then trained on the output of the con-

volutional base so that it can identify new classes. Figure 2 depicts the process of transfer learning.

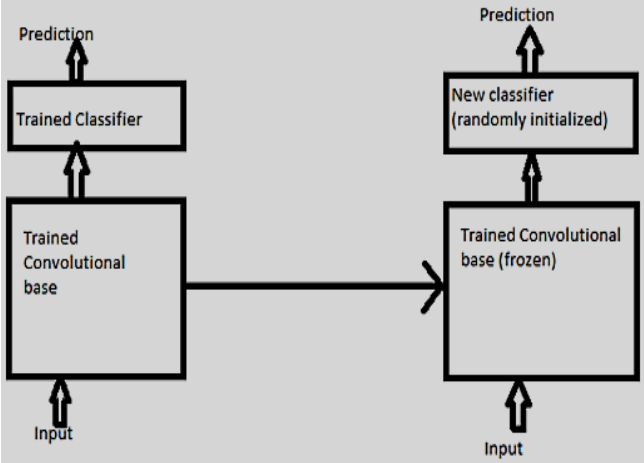

Figure 2. Changing classifier while keeping the same convolutional base.

Fine tuning is another widely used technique for model reuse. Fine-tuning consists of unfreezing a few of the top layers of a frozen model base and jointly training both the newly added classifier and the unfrozen layers of the model [24]. So, the main difference between fine tuning and transfer learning is that in transfer learning only the weights of the newly added classifier are optimized. On the other hand, in fine-tuning, we optimize both the weights of the classifier as well as the weights of some or all of the layers of the pre-trained model base [26]. Figure 3 depicts an example of fine tuning applied on the pre-trained VGG16 model [21].

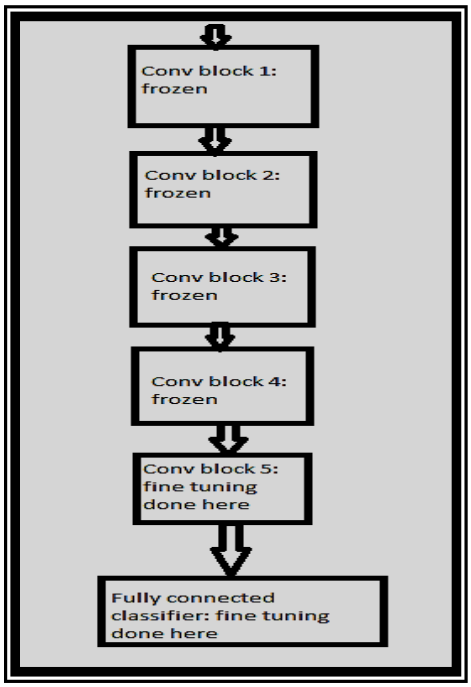

Figure 3. Fine tuning the last convolutional block and the fully connected classifier.

It must be remembered that fine-tuning the top layers of the pre-trained model is only possible once the classifier on top has already been trained. If the classifier is not already trained, the error signal propagated backwards during training will be too large and will destroy the abstract representations previously learned by the pre-trained models' top layers that are being fine-tuned [24]. Due to this reason, the recommended steps for fine-tuning are [24]:

1. Add a custom classifier on top of a pre-trained base network.
2. Freeze the weights of the base network.
3. Train the custom classifier.
4. Unfreeze some layers in the base network.
5. Jointly train both the unfrozen layers and the custom classifier.

The performances resulting from the use of transfer learning and fine tuning (when applied to solve the fruit image classification problem) have been presented in this paper.

4.3 AlexNet, GoogleNet and VGGNet models

Computer vision researchers have demonstrated progress in image classification tasks by validating their results against ImageNet [27]. The ImageNet Large Scale Visual Recognition Challenge (ILSVRC) has been running annually since 2010. The challenge is based on ImageNet dataset [25]. ImageNet is much larger in size and diversity compared with other standard image classification datasets (e.g. Caltech 101 [28] and Caltech 256 [29]). As reported in [27], ImageNet has 14,197,122 images from 21841 different categories**.** Only a subset of these images is used in ILSVRC competitions, where each participant is usually provided with 1.4 million images [24] belonging to 1000 different classes.

Over the years, a number of high performing models have been developed and evaluated by various research teams at ILSVRC. Examples of such models include AlexNet [30], VGGNet [21], Inception v3 [20] etc. 2012 ILSVRC winner AlexNet was the first large, deep CNN-based model to have performed so well in the competition. On the test data, AlexNet achieved top-5 test error rate of 15.3%, compared to 26.2% achieved by the second-best entry. AlexNet has 60 million parameters and 650,000 neurons and consists of five convolutional layers (some of which are followed by max-pooling layers) and three fully-connected layers [30]. AlexNet has been applied in various computer vision tasks such as object-detection [31], segmentation [32], video classification [33] etc.

GoogleNet (also known as Inception v1) [19] and VGGNet [21] were the winner and runner-up of the ILSVRC 2014 competition, respectively. GoogleNet is a 22 layer CNN with a top-5 error rate of 6.7%. This performance was very close to human-level performance. GoogleNet also has around nine times less number of parameters than AlexNet. This leads to more efficiency and less consumption of computational resources. GoogleNet was one of the first CNN models that strayed away from the traditional practice of stacking the convolutional and pooling layers in a sequential manner. The designers of the model have showed that creative structuring of the layers can lead to improved performance and computational efficiency [19]. The original model was called GoogleNet but subsequent versions were referred to as Inception v*N* where *N* refers to the version number released by Google. The experiments presented in this paper exploits Inception v3 model that is presented in [20]. The Inception v3 model has an even better ImageNet classification accuracy than earlier Inception models [20].

On the other hand, VGGNet is a 19 layer CNN that uses 3x3 filters with stride and pad of 1, along with 2x2 maxpooling layers with stride of 2. VGGNet achieved top-5 test error rate of 7.3% in

the ILSVRC 2014 competition. The model reinforced the idea that CNNs need to have a deep network of layers in order for the hierarchical representation of visual data to work [21]. In essence, VGGNet's main design paradigm is: "keep the network simple and deep". VGGNet also has a 16 layer variant which the author has used in the experiments presented in this paper. Although VGGNet has a simple architecture, it has three times more parameters than AlexNet resulting in high computational costs. This is the major drawback of VGGNet [20]. Table 1 summarizes some key differences between AlexNet, GoogleNet and VGGNet.

Table 1. Some key differences between AlexNet, GoogleNet and VGGNet

Year	CNN	Place in ILSVRC	Top-5 error rate	No. of parameters
2012	AlexNet	1st	15.3%	60 million
2014	GoogleNet	1st	6.67%	7 million
2014	VGGNet	2nd	7.3%	138 million

5. EXPERIMENTAL SETUPS

All experiments are carried out using TensorFlow version 1.10.0. All the code is written and executed on Jupyter notebooks. The hardware specification (for the experiments) is given in Table 2.

Table 2. Hardware used for the experiments.

CPU	Intel® Core™ i7 7700HQ
RAM	16 GB
GPU	NVIDIA's GeForce® GTX 1050 Ti

Four different experimental setups are created and all of them are described in detail in the following subsections:

5.1 Experiment #1: A 14-Layer Convolutional Neural Network

A self-designed 14-layer convolutional neural network applied on the fruit image classification problem. The network is trained on augmented data. This CNN is built and trained from scratch and no pre-trained model has been exploited in this case. The author's aim is to measure the performance of a self-designed CNN that does not use transfer learning and fine tuning. This will enable performance comparison with situations where transfer learning and fine tuning is used.

The CNN has four pairs of convolutional and pooling layers. The convolutional and pooling layers are followed by a flatten layer, a dropout layer and two dense layers. Details of the CNN structure are given in Table 3. The hyper parameters used during the CNN training are given in Table 4.

5.2 Experiment #2: Transfer learning using Inception v3

In the second experiment, transfer learning is performed by exploiting Inception v3 pre-trained model. The Inception model is very capable of extracting useful information from an image. The Inception model can be reused by merely replacing the layers that does the final classification.

Table 3. The 14-layer CNN structure

Layer Name	Details
Input Layer	Dimensions= (100×100×3)
Convolutional Layer 1	Number of Filters= 32, Kernel Size=(3,3), Activation=ReLU
Pooling Layer 1	Pool Size=(2,2)
Convolutional Layer 2	Number of Filters= 64, Kernel Size=(3,3), Activation=ReLU
Pooling Layer 2	Pool Size=(2,2)
Convolutional Layer 3	Number of Filters=128, Kernel Size=(3,3), Activation=ReLU
Pooling Layer 3	Pool Size=(2,2)
Convolutional Layer 4	Number of Filters=128, Kernel Size=(3,3), Activation=ReLU
Pooling Layer 4	Pool Size=(2,2)
Flatten Layer	Flattens the input tensor.
Dropout Layer	Dropout rate=0.5
Dense Layer 1	Units=512, Activation=ReLU
Dense Layer 2	Units=72, Activation=Softmax
Output Layer	Dimensions=(1×72)

Table 4. The hyper parameters for the 14-layer CNN training

Loss function	Categorical Cross Entropy
Optimizer	Adam
Learning Rate	1e-4
Number of epochs	100
Steps per epoch	100
Batch size	32

Transfer values are first computed for all the training and test images and saved in a cache file. These transfer values are then used to train a self-added classifier. Figure 4 depicts this process. The figure shows that first the input image is processed with the Inception model and then just prior to the final classification layer of the Inception model, the transfer values are saved to a cache-file. For each image, 2048 transfer values are computed.

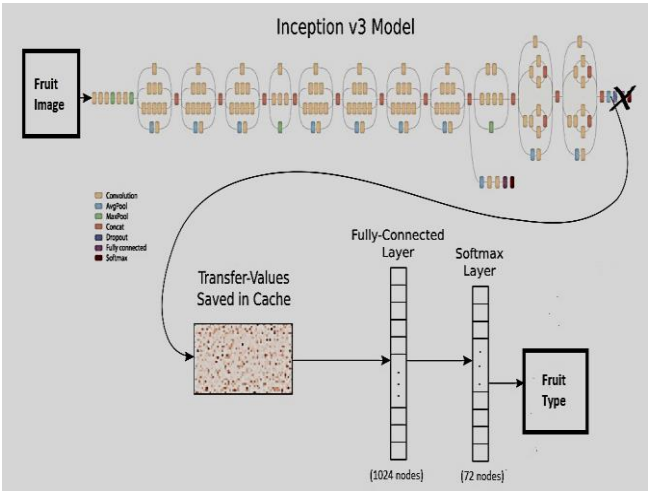

Figure 4. Transfer learning using Inception v3 model

Transfer values are analyzed using methods of dimensionality reduction called Principal Component Analysis (PCA) [37] and t-SNE [38, 39]. For each of the first 3000 training images, a combi-

nation of PCA and t-SNE is used to reduce transfer values from 2048 to 2 per image. Figure 5 depicts scatter plot based on the reduced transfer values of the 3000 images. From Figure 5, it can be easily inferred that the transfer-values from the Inception model appear to contain enough information to separate the Fruits-360 dataset images into classes. Unfortunately, there are some instances of overlap, so the separation is not 100% perfect.

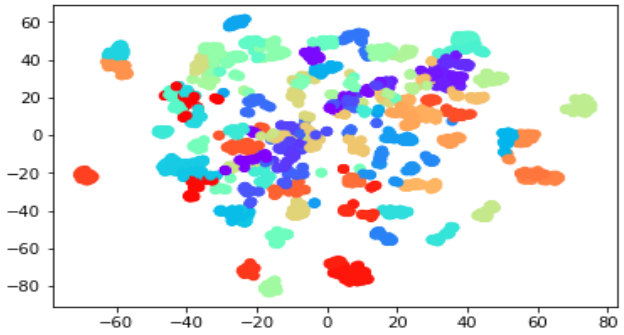

Figure 5. Scatter plot for the reduced transfer values of the 3000 training images

For training the classifier, cross entropy is used as the loss function. Optimization is done using Adam optimizer with a learning rate of 1e-4. Training batch size is 64 and 8000 iterations are performed during training. Training data for each batch is chosen randomly. No data augmentation is performed in this experiment.

5.3 Experiment #3: Transfer learning using VGG16

VGG16 model is utilized for transfer learning. The VGG16 model contains five convolutional blocks. Each convolutional block contains 2 or 3 convolutional layers and a pooling layer [24]. Training data is augmented and loaded using an image data generator. VGG16's convolutional base is retained but a self-designed classifier is added and trained. This process is depicted in Figure 2 and Figure 6. The structure of the new CNN is given in Table 5. All the hyper parameters used in this experiment are the same as given in Table 4.

Table 5. Structure of the CNN that performs transfer learning based on the VGG16 pre-trained model.

Layer Name	Details
Convolutional Base	This includes all the five convolutional blocks of the VGG16 model. The weights of these five convolutional blocks are kept frozen. The original VGG16 classifier layers are excluded.
Flatten Layer	Flattens the input tensor.
Dense Layer 1	Units=256, Activation=ReLU
Dense Layer 2	Units=72, Activation=Softmax

5.4 Experiment #4: Fine Tuning using VGG16

Fine tuning performed using VGG16 model. The weights of the first four convolutional blocks are kept frozen. All the convolutional and pooling layers of the fifth convolutional block are set as trainable. All the layers of the fifth convolutional block are trained along with a self-added classifier. The structure of the CNN is the same as given in Figure 6 and Table 5. As already stated, the only difference is that the layers of the fifth convolutional block are trainable. The hyper parameters used during training are the same as given in Table 4 except two changes. The optimizer used is RMSprop and the learning rate is 1e-5. The reason for choosing RMSprop is that it is giving better classification accuracy than the Adam optimizer. The learning rate is kept much lower because the author wants to limit the magnitude of the modifications that are made to the representations of the layers that are being fine-tuned [24].

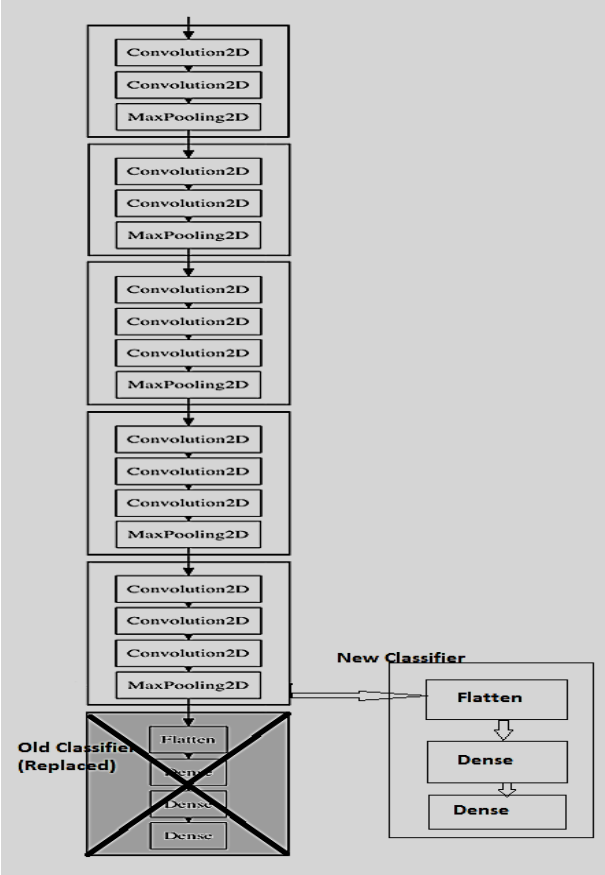

Figure 6. Transfer learning using VGG16

6. Results and Analysis

Table 6 presents the test set classification accuracies for the four experiments[1]. It can be easily observed that the highest classification accuracy is for experiment #3 (i.e. transfer learning performed using VGG16). It can also be seen that the use of transfer learning and fine tuning significantly improves the classification accuracy. The test set consists of 12,132 images and the classification accuracies represent the percentage of correctly classified test set images.

Figure 7 shows how training and validation accuracies as well as training and validation loss evolved with the number of epochs for experiment #1, 3 and 4. The validation loss fluctuate little bit but overall stays along with the training loss in all the three experiments. The validation loss never showed any persistent upward trend in all the three experiments. Overall, the validation accuracy also stays quite close to the training accuracy in all the three experiments. This indicates that there was no problem of overfitting.

[1] Github repository containing Jupyter notebooks for all the four experiments:
https://github.com/raheelsiddiqi2013/fruit_image_classification

It must be remembered that the validation data was completely different from the training data. No training sample was used for the validation purpose.

Table 6. Test set classification accuracies for the four experiments

Experiment	Test Set Classification Accuracy
Experiment #1: A 14-Layer Convolutional Neural Network	96.79%
Experiment #2: Transfer learning using Inception v3	98.1%
Experiment #3: Transfer learning using VGG16	99.27%
Experiment #4: Fine Tuning using VGG16	98.01%

All the previous research [4, 5, 6, 7, 14, 15, 40] in this area considered only 18 fruit types and between 1653 to 3600 images. The research presented in this paper is based on a much larger dataset and a far greater number of fruit types (i.e. 48, 249 fruit images and 72 different fruit types). For all the previous studies, the number of images per class is very low (as low as 61). For Fruits 360 dataset, there are at least 650 images per fruit type. This means that the results presented in this paper are statistically more significant because a much larger dataset is involved.

Up till now, the highest classification accuracy reported for automated fruit image classification is 95.67% [40]. It has been shown in this paper that 99.27% classification accuracy can be achieved through transfer learning using the pre-trained VGG16 model. This is certainly a remarkable improvement. Even the classification accuracies achieved in experiment #1, #2 and #4 are much higher than those achieved in all the past studies [4, 5, 6, 7, 14, 15, 40].

Unlike past approaches [4, 5, 6, 7, 14, 15, 40], the Fruits 360 dataset is not subjected to any pre-processing for the experiments. The images are neither cropped nor resized. The original size of

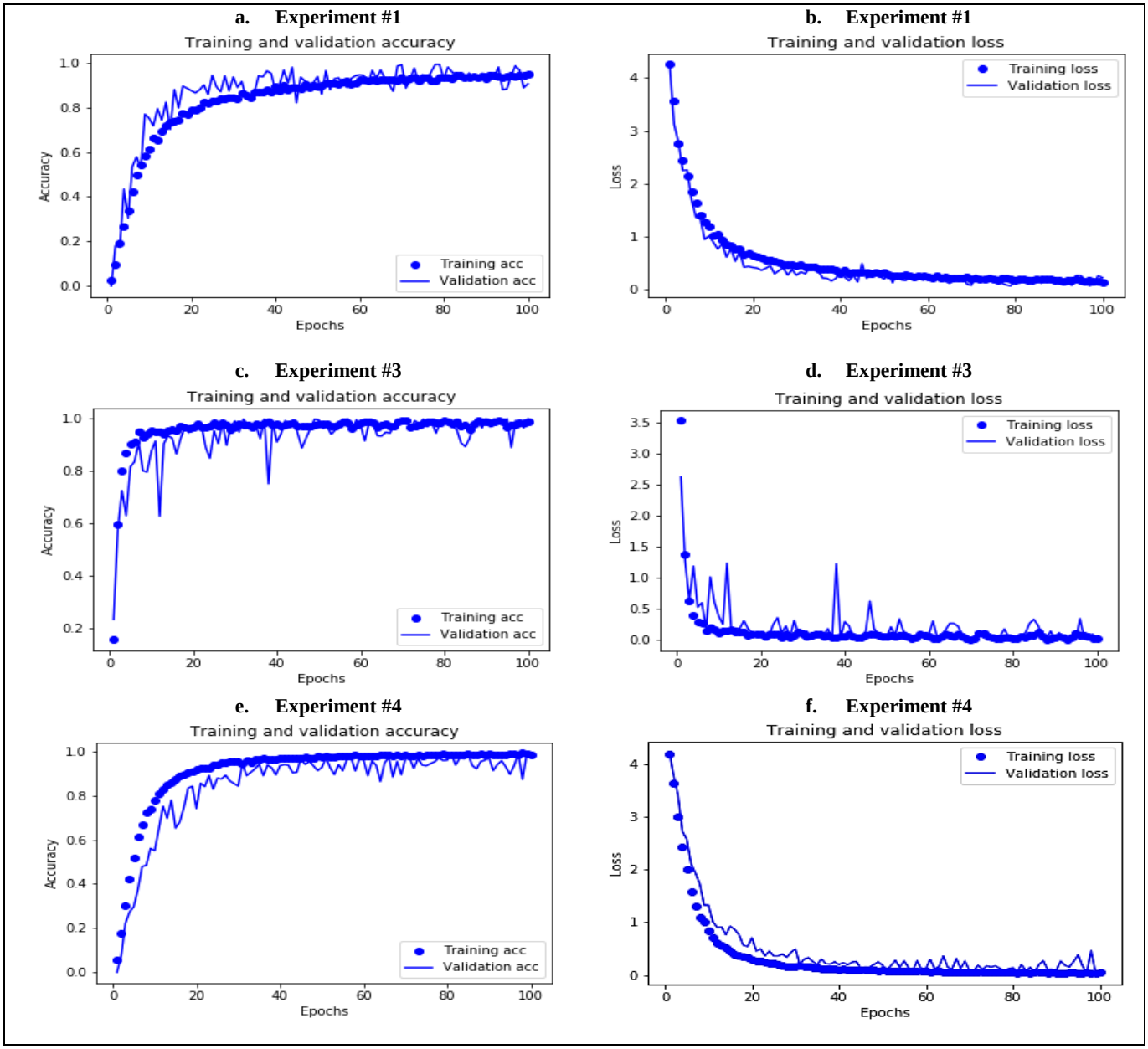

Figure 7. Graphs depicting how training and validation values changed with the number of epochs

100 × 100 pixels is used. The images of Fruits 360 dataset also come with clean, white background. So, no image background removal is done. This is unlike past approaches where the task of background removal was done by the researchers themselves. And also, the important task of feature extraction from images is handled by the CNNs themselves. This was not the case in many past approaches [4, 6, 7, 14, 15], where features like color, shape, texture etc. were extracted separately using various techniques chosen or designed by the researchers themselves.

Apart from the use of pre-trained models, experiment #1 has demonstrated that a carefully designed convolutional neural network (that does not exploit any pre-trained model) can also yield very high classification accuracy (i.e. 96.79%). This result is significant in the sense that the classification accuracy obtained is much higher than the previous two studies [5, 40] where CNN was also used.

7. CONCLUSION

In this paper, it has been shown through experiments that transfer learning and fine tuning can significantly improve fruit image classification accuracy. Transfer learning using VGG16 pre-trained model has been demonstrated to result in the best classification accuracy of 99.27%. Fine tuning using VGG16 has produced 98.01% classification accuracy while transfer learning using Inception v3 has produced 98.1% classification accuracy. These are much better results when compared with all the past research works [4, 5, 6, 7, 14, 15, 40] in the area of automated fruit image classification. The number of fruit types and the total number of images in the Fruits 360 dataset are also much greater compared to the datasets used in the past research.

Some of the short-comings in the presented work are:

i. All the Fruits 360 images are clean i.e. no imperfect image. Therefore, all experiments are carried out using clean images. This means that the trained models may not perform that well on imperfect images (e.g. images with complicated background or images where the fruit is not focused properly etc.).

ii. In addition to imperfect images, the research work may be extended to sliced, dried, canned and tinned fruits.

iii. More research is required to find the optimal number and size of dense layers used in experiment #2, 3 and 4.

iv. More research is required in the context of fine tuning using VGG16. Perhaps, the network may perform better if the number of epochs is increased and/or the learning rate is reduced further. Another possibility is that the convolutional block 4 of VGG16 is also set to trainable i.e. the weights of the 4th convolutional block are also unfrozen.

The above mentioned short-comings provide directions for future research.

8. REFERENCES

[1] Anderson Rocha, Daniel C. Hauagge, Jacques Wainer and Siome Goldenstein. 2010. Automatic fruit and vegetable classification from images. *Computers and Electronics in Agriculture* 70, 1 (Jan. 2010), 96-104. DOI: https://doi.org/10.1016/j.compag.2009.09.002

[2] Baohua Zhang, Wenqian Huang, Jiangbo Li, Chunjiang Zhao, Shuxiang Fan, Jitao Wu and Chengliang Liu. 2014. Principles, developments and applications of computer vision for external quality inspection of fruits and vegetables: A review. *Food Research Journal*, 62 (Aug. 2014), 326–343. DOI: https://doi.org/10.1016/j.foodres.2014.03.012

[3] Anuja Bhargava and Atul Bansal. 2018. Fruits and vegetables quality evaluation using computer vision: A review. *Journal of King Saud University – Computer and Information Sciences*. (Jun. 2018) DOI: https://doi.org/10.1016/j.jksuci.2018.06.002 [in Press]

[4] Yudong Zhang and Lenan Wu. 2012. Classification of Fruits Using Computer vision and a Multiclass Support Vector Machine. *Sensors* 12, 9 (Sept. 2012), 12489-12505. DOI: https://doi.org/10.3390/s120912489

[5] Yu-Dong Zhang, Zhengchao Dong, Xianqing Chen, Wenjuan Jia, Sidan Du, Khan Muhammad and Shui-Hua Wang. 2019. Image based Fruit Category Classification by 13-layer Deep Convolutional Neural Network and Data Augmentation. *Multimedia Tools and Applications* 78, 3 (Feb. 2019), 3613-3632. DOI: https://doi.org/10.1007/s11042-017-5243-3

[6] Shuihua Wang, Yudong Zhang, Genlin Ji, Jiquan Yang, Jianguo Wu and Ling Wei. 2015. Fruit Classification by Wavelet-Entropy and Feedforward Neural Network Trained by Fitness-Scaled Chaotic ABC and Biogeography-Based Optimization. *Entropy* 17, 8 (Aug. 2015), 5711-5728. DOI: https://doi.org/10.3390/e17085711

[7] Yudong Zhang, Shuihua Wang, Genlin Ji and Preetha Philips. 2014. Fruit Classification using Computer Vision and Feedforward Neural Network. *Journal of Food Engineering* 143 (Dec. 2014), 167-177. DOI: https://doi.org/10.1016/j.jfoodeng.2014.07.001

[8] Wenhao Shao, Yanjie Li, Songfeng Diao, Jingmin Jiang and Ruxiang Dong. 2017. Rapid classification of Chinese quince (Chaenomeles speciosa Nakai) fruit provenance by near-infrared spectroscopy and multivariate calibration. *Analytical and Bioanalytical Chemistry* 409, 1 (Jan. 2017) 115–120. DOI: https://doi.org/10.1007/s00216-016-9944-7

[9] Radi, S. Ciptohadijoyo, W.S. Litananda, M. Rivai and M.H. Purnomo. 2016. Electronic nose based on partition column integrated with gas sensor for fruit identification and classification. *Computers and Electronics in Agriculture* 121 (Feb. 2016), 429-435. DOI: https://doi.org/10.1016/j.compag.2015.11.013

[10] M. Fatih Adak and Nejat Yumusak. 2016. Classification of E-Nose Aroma Data of Four Fruit Types by ABC-Based Neural Network. *Sensors* 16, 3 (Feb. 2016). DOI: https://doi.org/10.3390/s16030304

[11] Andre Esteva, Brett Kuprel, Roberto A. Novoa, Justin Ko, Susan M. Swetter, Helen M. Blau and Sebastian Thrun. 2017. Dermatologist-level classification of skin cancer with deep neural networks. *Nature* 542 (Feb. 2017), 115–118. DOI: https://doi.org/10.1038/nature21056

[12] Paras Lakhani and Baskaran Sundaram. 2017. Deep Learning at Chest Radiography: Automated Classification of Pulmonary Tuberculosis by Using Convolutional Neural Networks. *Radiology* 284, 2 (Apr. 2017), 574-582. DOI: https://doi.org/10.1148/radiol.2017162326

[13] Teresa Araújo , Guilherme Aresta , Eduardo Castro , José Rouco, Paulo Aguiar, Catarina Eloy, António Polónia and Aurélio Campilho. 2017. Classification of breast cancer histology images using Convolutional Neural Networks. *PLoS ONE* 12, 6 (Jun. 2017), e0177544. DOI: https://doi.org/10.1371/journal.pone.0177544

[14] Zhihai Lu, Siyuan Lu, Shuihua Wang, Yujie Li, Yudong Zhang and Huimin Lu. 2017. A Fruit Sensing and Classification System by Fractional Fourier Entropy and Improved Hybrid Genetic Algorithm. In: *Proceedings of the 5th IIAE International Conference on Industrial Application Engineering 2017*, Kitakyushu, The Institute of Industrial Applications Engineers, Japan, 293-299. DOI: https://doi.org/10.12792/iciae2017.053

[15] Shuihua Wang, Zhihai Lu, Jiquan Yang, Yu-Dong Zhang, John Liu, Ling Wei, Shufang Chen, Preetha Phillips and Zhengchao Dong. 2016. Fractional Fourier Entropy increases the recognition rate of fruit type detection. *BMC Plant Biology*, 16, S2, Article 10 (Oct. 2016). DOI: https://doi.org/10.1186/s12870-016-0904-3

[16] R. M. Bolle, J. H. Connell, N. Haas, R. Mohan and G. Taubin. 1996. VeggieVision: a produce recognition system. In: *Proceedings of the Third IEEE Workshop on Applications of Computer Vision (WACV'96)*. Sarasota, FL, USA, 244-251. DOI: https://doi.org/10.1109/ACV.1996.572062

[17] W. Chaw Seng and S. Hadi Mirisaee. 2009. A new method for fruits recognition system. In: *Proceedings of the 2009 International Conference on Electrical Engineering and Informatics*, IEEE, Selangor, Malaysia. DOI: https://doi.org/10.1109/ICEEI.2009.5254804

[18] Hoo-Chang Shin, Holger R. Roth, Mingchen Gao, Le Lu, Ziyue Xu, Isabella Nogues, Jianhua Yao, Daniel Mollura and Ronald M. Summers. 2016. Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning. *IEEE Transactions on Medical Imaging* 35, 5 (May 2016) 1285–1298. DOI: https://doi.org/10.1109/TMI.2016.2528162

[19] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke and Andrew Rabinovich. 2014. Going Deeper With Convolutions. arXiv:1409.4842. Retrieved from https://arxiv.org/abs/1409.4842

[20] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens and Zbigniew Wojna. 2015. Rethinking the Inception Architecture for Computer Vision. arXiv:1512.00567. Retrieved from https://arxiv.org/abs/1512.00567

[21] Karen Simonyan and Andrew Zisserman. 2015. Very Deep Convolutional Networks for large-scale Image Recognition. arXiv:1409.1556. Retrieved from https://arxiv.org/abs/1409.1556

[22] Horea Muresan. 2018. Fruits-360: A dataset of images containing fruits. (July 2018). Retrieved July 15, 2018 from https://github.com/Horea94/Fruit-Images-Dataset

[23] Horea Mureşan and Mihai Oltean. 2018. Fruit recognition from images using deep learning. arXiv:1712.00580. Retrieved from https://arxiv.org/abs/1712.00580

[24] François Chollet. 2018. *Deep Learning with Python*. Manning Publications, New York, USA.

[25] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li and Li Fei-Fei. 2009. ImageNet: A large-scale hierarchical image database. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, Miami, USA. DOI: https://doi.org/10.1109/CVPR.2009.5206848

[26] Magnus Erik Hvass Pedersen. 2018. TensorFlow Tutorial: Fine-Tuning. Retrieved September 12, 2018 from https://github.com/Hvass-Labs/TensorFlow-Tutorials/blob/master/10_Fine-Tuning.ipynb

[27] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg and Li Fei-Fei. 2015. ImageNet Large Scale Visual Recognition Challenge. arXiv:1409.0575. Retrieved from https://arxiv.org/abs/1409.0575

[28] Li Fei-Fei, R. Fergus and P. Perona. 2004. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In: *IEEE CVPR Workshop of Generative Model Based Vision (WGMBV)*. IEEE, Washington, DC, USA. DOI: https://doi.org/10.1109/CVPR.2004.383

[29] Greg Griffin, Alex Holub and Pietro Perona. 2007. *Caltech-256 object category dataset*. Technical report 7694. Caltech.

[30] Alex Krizhevsky, Ilya Sutskever and Geoffrey E. Hinton. 2017. ImageNet Classification with Deep Convolutional Neural Networks. *Commun. ACM* 60, 6 (Jun. 2017), 84-90. DOI: https://doi.org/10.1145/3065386

[31] Ross Girshick, Jeff Donahue, Trevor Darrell and Jitendra Malik. 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. arXiv:1311.2524. Retrieved from https://arxiv.org/abs/1311.2524

[32] Jonathan Long, Evan Shelhamer and Trevor Darrell. 2015. Fully convolutional networks for semantic segmentation. arXiv:1411.4038. Retrieved from https://arxiv.org/abs/1411.4038

[33] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar and Li Fei-Fei. 2014. Large-scale video classification with convolutional neural networks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Columbus, OH, USA, 1725–1732. DOI: https://doi.org/10.1109/CVPR.2014.223

[34] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner. 1998. Gradient based learning applied to document recognition. In: *Proceedings of the IEEE*, 86, 11 (Nov 1998), 2278–2324. DOI: https://doi.org/10.1109/5.726791

[35] Pierre Sermanet, David Eigen, Xiang Zhang, Michael Mathieu, Rob Fergus and Yann LeCun. 2014. OverFeat: Integrated recognition, localization and detection using convolutional networks. arXiv:1312.6229. Retrieved from https://arxiv.org/abs/1312.6229

[36] Yann LeCun, Yoshua Bengio and Geoffrey Hinton. 2015. Deep Learning. *Nature* 521 (May 2015), 436–444. DOI: https://doi.org/10.1038/nature14539

[37] Hervé Abdi and Lynne J. Williams. 2010. Principal Component Analysis. *WIREs Computational Statistics*, 2, 4 (Jul. 2010), 433-459. DOI: https://doi.org/10.1002/wics.101

[38] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9 (Nov. 2008), 2579-2605.

[39] Martin Wattenberg, Fernanda Viégas and Ian Johnson. 2016. How to Use t-SNE Effectively. *Distill*. DOI: http://doi.org/10.23915/distill.00002

[40] Shui-Hua Wang and Yi Chen. 2018. Fruit category classification via an eight-layer convolutional neural network with parametric rectified linear unit and dropout technique. *Multimedia Tools and Applications*, (Sep. 2018), 1-17. DOI: https://doi.org/10.1007/s11042-018-6661-6