Deep transfer learning approach for classification of grocery store dataset using data augmentation

Asmaa Mohamed[1], Sawsan Mohammed Aziz[1], Heba F.Eid[1]
[1]Department of Mathematics and Computer Science
Faculty of Science
Al-Azhar University
Egypt

Abstract—

There are approximately 285 million visually impaired people worldwide. People with visual impairments have difficult performing shopping activities. In this paper, we proposed an image classification method that uses a convolutional neural network (CNN), transfer learning, fine tuning, batch normalization, data augmentation, and dropout to develop a system to classify multiclass images of grocery stores. The proposed method is intended to help visually impaired people perform shopping tasks.Several deep learning algorithms have been developed to solve the problem of grocery store product identification and classification. Such techniques, however, have limitations. As a result, this paper utilizes deep learning in combination with the concept of data augmentation, which is based on three well-known Convolutional Neural Network (CNN) architectures (Visual Geometry Group (VGG19), MobileNet, and Extreme Inception (Xception),ImageNet was used to train these three models. The suggested approach's performance is evaluated using the Grocery store dataset. Compared to this research's current techniques, The proposed method was achieved using a hybrid model illustrated as follows; on the base model, the Xception proved to be the best model, with an F1-score of 98% on the testing set. The Xception network was the most effective for the fruits model, obtaining an F1-score of 98% on the testing set. For the vegetables model, MobileNet was identified as the best model, obtaining an F1-score of 94% on the testing set. For the packages model, the VGG19 network obtained the best results, demonstrating an F1-score of 97% on the testing set. The proposed model outperformed existing state-of-the-art models in terms of classification accuracy, precision, recall, and F1-score.

Keywords*—convolutional neural network; data augmentation; transfer learning; fine-tuning.*
DOI Number: 10.14704/nq.2022.20.13.NQ88036 **NeuroQuantology 2022; 20(13): 265-278**

I. INTRODUCTION

The World Health Organization [1]estimates that over 2.2 billion people worldwide are blind or have impaired vision. Assistive technologies attempt to help people with disabilities achieve greater independence and inclusion. Several computer vision and machine learning systems have been proposed to help visually impaired people with various tasks, e.g., navigation [2], text reading [3], identification currency(as in a system that helpsvisually impaired people identify the specific denomination/amount of physical coins and bills)[4], and face recognition[5]. In addition, continuous advancements in information technology (IT) are expanding the potential for IT-based mobile assistive solutions to help visually impaired persons achieve independence, safety, and an enhanced quality of life. People with visual impairments have difficult performing shopping tasks. Thus, developing technological innovations to help people perform these activities is important. This study attempts an approach for classifying grocery store dataset. In future, We can deploy this hybrid model on mobile phones and other handheld devices accessible via our haptic (touch) and audio sensory channels to help visually impaired in their daily shopping.

In this study, we focused on object recognition approaches for the visually impaired to recognize products typically found in grocery stores, supermarkets, refrigerators, and pantries. A system to recognize such objects would be beneficial for a visually impaired person because they could independently identify food products, e.g., the user could determine whether they had juice or milk in their refrigerator or pantry. However, object recognition has several challenges, e.g., uncontrolled environmental conditions, significant lighting variations, background noise, orientation, the camera's distance from the object, and partial occlusions. In addition, it is difficult for a visually impaired person to differentiate visually different objects that are identical to the touch, e.g., milk cartons and juice boxes.

In this paper, we proposed a method to classify grocery products to help visually impaired people. The proposed method is based on a convolutional neural network (CNN) that employs transfer learning[6], fine tuning of pre-trained models, data augmentation, batch normalization, and dropout techniques. We suggest a client-server architecture (this is genuinely a suggestion for future implementation of the proposed method), and the proposed method could be implemented on a wearable device(system), e.g., smartphones, smart refrigerators, and smart glasses. Thus, the system is expected to help visually impaired users identify items found in various environments, e.g., their homes, and help them purchase products independently. Mobile phones are becoming more accessible to those with visual impairments. In addition, innovative assistive applications for the visually impaired are implemented on common devices that can be used in motion (e.g., smartphones). In addition, such applications can be implemented in an environment that may be in motion (e.g., a public transportation system) or in an environment in which the user may be in motion (e.g., an intelligent home) [7]

This current paper's contribution enhances the accuracy and F1-score when classifying multiclass images on the grocery store dataset using hybrid transfer learning models. First, a base model is used to classify the input image into one of three classes (fruits, vegetables, or packages), and then the image is tested again to categorize it into an appropriate subclass based on this decision (e.g., the specific type of fruit or milk products). The new feature of the proposed method is that it obtained results superior to state-of-the-art grocery product classification methods, as we will explain in the method section.

The remainder of this article is organized as follows: Related publications offer the most frequent grocery store terms and approaches in Section II. Section III illustrates methods , Section IV illustrates the proposed and methodology, while Section V examines the proposed model's outcomes and compares them to other relevant models. Section V is the conclusion and future work.

II. RELATED WORK

A previous study[8] used deep learning to create a framework to classify fruit. This framework was built on two distinct deep learning architectures, a light model comprising six CNN layers and a 16-layer fine-tuned and pretrained deep learning model from the Visual Geometry Group (VGG16). This framework was evaluated using two color picture datasets, one of which is publicly accessible. The first dataset (dataset 1) contained images of easy-to-classify fruit, and the second dataset (dataset 2) contained difficult-to-classify fruit images. In dataset 1, the first and second models obtained classification accuracies of 99.49% and 99.75%, respectively, and the classification accuracies of the first and second models in dataset 2 were 85.43% and 96.75%, respectively.

Machado et al. [9] performed a systematic literature review to identify state-of-the-art of grocery product recognition methods. Here they described and analyzed five public datasets to classify grocery products. The most relevant papers are summarized in the following.

Rivera-Rubio et al. [10] proposed three approaches that combined different techniques: (1) the scale-invariant feature transform (SIFT), k-means, and a support vector Machine (SVM); (2) SIFT and locality-constrained linear coding (LLC); and (3) SIFT, principal component analysis, Fisher vector encoding, and an SVM. Rivera-Rubio et al. [10]also introduced the SHORT-100 dataset Accuracy evaluations were performed on two sets of images: (1) images captured by a smartphone and (2) images extracted from videos. The SIFT+K-Means+SVM approach obtained the best average accuracy with 77.51% on the first set, and the best approach was SIFT+LLC on the second set, which obtained an average accuracy of 69.41%.

Varol and Kuzu[11]considered the problem of recognizing a specific commodity (cigarette packages) in images of stocked shelves in different establishments. They also introduced the grocery dataset and proposed an approach in which the Viola-Jones detected objects of interest, trained with a histogram of oriented gradients characteristics. From the detected region, only 40% of the image was used for classification at the top of the package where the logo is located. Then, the logo image was represented with shape information (described by SIFT) and color information (described by thehue, saturation, value (HSV) model) using the BoW (Bag-of-Words) approach. In addition, visual vocabularies of the color formation were created, and their respective descriptors were clustered using the k-means clustering algorithm. The SVM classifier was then used to recognize the brand of the product. Experiments were performed to evaluate accuracy, and the following results were obtained: 85.9% accuracy using only SIFT descriptors, 60.5% accuracy using only HSV descriptors, and 92.3% accuracy using both descriptors.

Jund et al. [12] used transfer learning through the Caffe deep learning framework. They proposed a fine-tuning CNN (FT-CNN) and introduced the Freiburg Groceries dataset, which was used to evaluate accuracy. They found that their method obtained an average accuracy of 78.9%.

Klasson et al. [13] employed three different pretrained CNNs, i.e., the AlexNet, VGG16, and DenseNet-169 CNNs, as feature descriptors for an SVM classifier. Here, layers were extracted from the CNNs with and without fine tuning. They also proposed a method that only used FT-CNNs as descriptors and classifiers. The article introduced a new dataset for grocery store, and this basis allowed both fine-grained and coarse-grained classification of specific products. The FT-CNN DenseNet-169 approach with the SVM obtained an accuracy rate of 85.0% when classifying specific products. With the CNN DenseNet-169 approach without fine tuning, the SVM classifier obtained a better accuracy rate of 85.2% when classifying product classes.

III. METHODS

The proposed method uses different pretrained CNNs provided by the Keras library. In addition, we focus on a multiclass classification problem; thus, the SoftMax activation function is used in the final layer. Tests were performed to increase accuracy and reduce the loss function through transfer learning with fine tuning, data augmentation, batch normalization, and dropout layers. In addition, adjustments were made to the resolution of the images, and the domain-batch size value and photometric normalization functions from the Keras library were used for the proposed method.

A. *Convolutional Neural Networks*

CNNs are commonly used in image processing tasks. CNNs overcome the primary limitations of deep neural networks. In addition to image classification, they help with object identification, picture segmentation, GANs, and other tasks involving images. There are several approaches to constructing a CNN, and numerous pretrained models employ CNNs to perform various tasks. In this study, we extensively used CNNs. The core building blocks of a CNN are convolutions, filters, stridesand padding, and pooling. Several pretrained CNN models that perform very well in classification tasks are available, includingVGG[14],Inception[15], ResNet[16], AlexNet [17], GoogleNet[18], MobileNetV2[19],Xception[20] and DensNet[21].In this study, we used six pretrained models (i.e., AlexNet, VGG, ResNet, MobileNet, Xception, and Inception) to classify grocery store images. Here, each model was configured with different depths, numbers of parameters, and input sizes. However, they were all trained on ImageNet [22]. In addition, the multiclassclassification of fully connected layers in each model was updated. For the previously stated architectures, ImageNet weights were also used; thus, they were not learned from scratch. In addition, where images of grocery stores are resized to 224*244*3 pixelsfor fine-tuning with Keras.

B. *Deep Transfer Learning*

Transfer learning (TL) is a standard machine learning approach to creating a model from a pretrained model [6]. A pretrained model is trained to handle a particular task using an extensive dataset (e.g., ImageNet). Data-related features are learned in each layer of the deep neural network (DNN), and each layer is linked to a deeper layer by a set of trainable weights; thus, previously learned features serve as the input data for the next deeper layer[23,24]. In addition, the input data and learned weights are convolved to generate a new feature map, then sent feature map to an activation function [25].

Transfer learning aims to enhance target learners' performance on target domains by transferring information from separate but related source domains. In this method, the reliance on a large amount of target-domain data for creating target learners may be decreased. Transfer learning has become a prominent and promising field in machine learning due to its broad application prospects.

C. *Data Normalization*

Data normalization is defined as "the conversion of image data pixels to a predetermined range," i.e., [0, 1] or [−1, 1]. Most images have pixel values ranging from 0 to 255, and large integer values may interfere with or slow down the learning process in a DNN. Thus, image normalization is recommended to normalize the pixel values in the range [0, 1].

The images in the dataset were normalized (i.e., rescaled) using the ImageDataGenerator function in Python with the parameter rescale=1. /255.

D. *Data Augmentation*

A DL model must be very generalizable in order to be trained adequately. The diversity of training data and the model's ability to categorize blind test data accurately are generalizable. Generally, the model must include many data alternatives, e.g., different sample orientations or color scales. A DL model may misclassify an image because a different version of the image with a different color range, orientation, or location was not included in the training data. In addition, models with poor generalizability often overfit the training data, which indicates that they achieve high training accuracy but exhibit poor validation accuracy. To construct an effective DL model, the validation accuracy must increase with the training accuracy, and validation errors must be reduced in tandem with the training error. Data augmentation[25], [26]can facilitate the development of more comprehensive models that encompass as many possibilities as possible, and it is one of the most practical approaches to prevent model overfitting.

In this study, data augmentation was performed in Python using the ImageDataGenerator class from Keras, including standardization, rotation, shifts, changes in brightness,

The Keras ImageDataGenerator class is designed to provide real-time data augmentation, which is its primary benefit. The ImageDataGenerator class provides the model with image augmentation of the images in each epoch.

IV. Proposed Methodology

The proposed method aims to classify the input image to its subclass because this dataset has 83 classes. When we constructed a CNN architecture as the feature extractor and an SVM as the classifier, the accuracy was 66%. Then, we used a pretrained model overall 83 classesof the dataset; however, the accuracy was insufficient. The dataset is hierarchical; thus, we suggest a multi-model comprising a hybrid of pretrained models to classify the multiclass images in the dataset.

A. *Dataset Description*

The dataset comprises images taken from fruit and vegetable sections and refrigerated products, including dairy and juice products, in 18 different grocery shops. The dataset contains 5125 photos divided into 81 fine-grained groups, with 30 to 138 images in each class.

Our goal was to acquire natural images in the identical conditions that they would be in as part of a mobile phone assistive application. All images were captured using a 16-megapixel Android smartphone camera at various distances and perspectives. Other things in the background or items put on the incorrect shelf are sometimes shown in the images, in addition to the targeted item. Because these are common conditions in grocery store environments, image classifiers utilized for aiding devices must be capable of working effectively with such noise. The lighting in the images may also change based on where the things are in the store. Images are sometimes shot while the photographer holds the thing in his or her hand. This is especially true for refrigerated products, which are often stacked compactly in refrigerators. We deliberately varied the object's location in these images so that it is not always centered or present in its entirety.

Klasson et al. [13] provided a description and detailed analysis of the dataset used in the current study.

B. *Data Preprocessing*

After training the models on the dataset, we found that the dataset introduced by Klasson et al. included classes that were arranged incorrectly, i.e., there were classes in the test set that did not exist in the training or validation sets (and vice versa). In addition, the training, validation, and testing sets were not split logically at an effective ratio.

Thus, we split the dataset by combining all images and dividing it into the training, validation, and testing sets at a ratio of 70:10:20 to satisfy our objectives in terms of computational costs when training and evaluating the models, as well as the representativeness of the training and test sets. With this split, we found that accuracy improved. To increase the size of our training set, we applied different data augmentation techniques to create more images. As the target (i.e., a captured image of a product) may exist in a variety of conditions, e.g., different orientations, locations, scales, and brightness, we applied realistic transformations, e.g., image rotation with a maximum of 40 degrees, width and height scaling at a factor of 0.2, shearing with a small value of 0.2; to mimic motion blur, zoom with 0.2, and horizontal flip (to be able to identify the object from both sides). Fig. 1 shows samples of the images obtained applying these data augmentation techniques.

The data augmentation is exclusively applied to the training data in Table I

TABLE I. THE AUGMENTATION TECHNIQUES AND THE PARAMETER OF EACH TECHNIQUE

Argument	Parameter value
Rescale	1/255
Rotation range	40
Zoom range	0.2
Shear range	0.2
Horizontal shift range	0.2
Vertical shift range	0.2
Horizontal flip	True

Fig. 1. Samples of augmented images after applying (a) rotation, (b) width and height scaling, (c) shearing, (d) zoom, (e) horizontal flip, and (f) a combination of all techniques.

To apply data augmentation, we used the Keras ImageDataGenerator class, which is an online augmentation technique. This process is described as follows.

- Accept Input a batch of images used for training.
- Apply a predetermined series of transformations to each image in the batch.
- Replace the original batch with the new randomly transformed batch.
- Train the CNN on the randomly transformed batch.

C. Base Model

Our model creation approach was divided into two stages. First, we constructed and tuned a base CNN model, and then we applied TL utilizing several pretrained models: VGG[14],InceptionV2[28], ResNet[16], AlexNet [17], MobileNetV2[19]. In the model construction and tuning process, we discovered that increasing the number of epochs improved accuracy, early stopping helped reduce execution time, changing the batch size improved the execution time and memory requirements, and data augmentation helped avoid data overfitting.

The proposed model, which is based on a hybrid of the pretrained models, was designed to predict and classify the grocery store image dataset. The pretrained architecture utilized in the proposed train the dataset as we will train four models. First, constructed the base model to classify the input image into one of three classes (i.e., fruits, vegetables, or packages), and then, based on this initial classification result, we classified the image into its subclass (e.g., which type of fruit or milk product). We retrieved extracted features, and this architecture calculates the categorical cross-entropy loss function for classification problems. Then, the hybrid model was assessed using the test set. Fig. 2 shows a flow diagram for this framework.The proposed method isdivided into four phases, as shown in Fig. 3.Each phase of the proposed method is described in method and experiments and results section. Data-driven models that can be trained to learn essential features from raw Input are becoming increasingly popular, particularly when used in conjunction with feature learning, which is a core advantage of DL [4], [5].

A comparison of several TL models is presented in the following sections. DL models require a large dataset to function

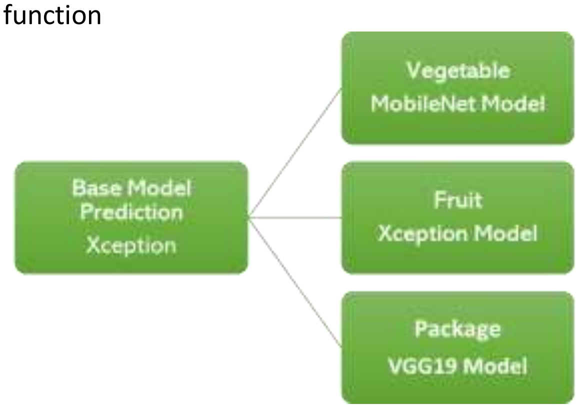


Fig. 2. Framework flow diagram.

effectively[29],and model performance can be improved by supplementing the available source images. This may also be accomplished by image preprocessing.The proposed model is based on a modified CNN that incorporates training weights from the MobileNet, VGG19, and Xception models. The learning scenario begins with the input layer getting images of grocery store products. Three pre-trained networks share the input layer. By freezing all of the layers of three models, they are reshaped. However, in order to add the required output layers, we dropped the top layer (output layer) from each model.

A Conv 1x1 layer with 1024 filters with padding-zero and stride-one was added to gather the most essential features and allow for reduced dimensionality, followed by a flattening layer to convert the matrix to a one-dimensional tensor (vector). As seen in Fig. 4,, and Fig. 5 shows a block diagram of the proposed method.

V. EXPERIMENTS AND RESULTS.

A. *Experimental Settings*

This study illustrates that transfer learning(using VGG19 Model, MobileNet, and Xception) achieves the maximum accuracy using a grocery store dataset Table III shows how the dataset is partitioned in a 7:2:1 ratio. All tests are carried out using Google's Colab [30], the Keras framework [31], which can operate on top of TensorFlow, and the Python programming language. All experiments were carried out on an NVIDIA Tesla K80 GPU (Graphical Processing Unit) with 12 GB of RAM.

B. Experimental Evaluation

In the testing phase equations provided in Table II, the performance of the proposed model is assessed using accuracy (1),F1-score (2), precision (3), recall (4), and confusion matrix. Furthermore, as illustrated in Fig. 9-12, loss, accuracy, validation loss, and validation accuracy are determined at various epochs throughout the training phase. In addition, we compared the performance of our proposed model (shown in Table IV-VI) with that of other models (e.g. [13]) that operate on the same dataset (shown in TableVIII) and with each architecture utilized in the proposed model separately.

The Adam optimizer was used to compile the proposed model, which is a stochastic gradient descent approach with a learning rate of 2e-5. The loss function is a categorical cross-entropy that is used to estimate the loss in the multi-class classification task.

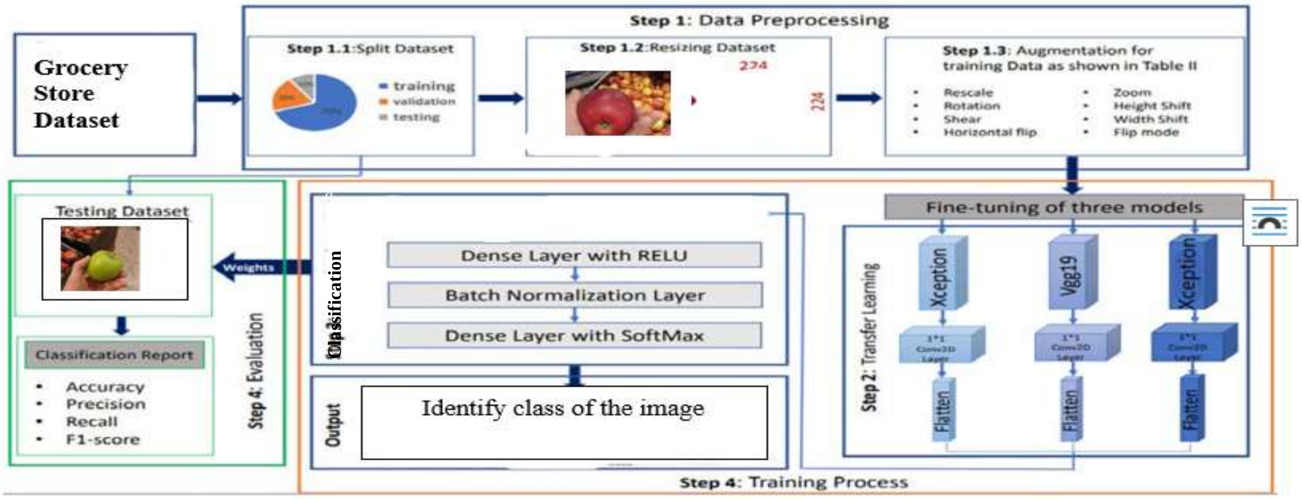


Fig. 3 The framework of the proposed model

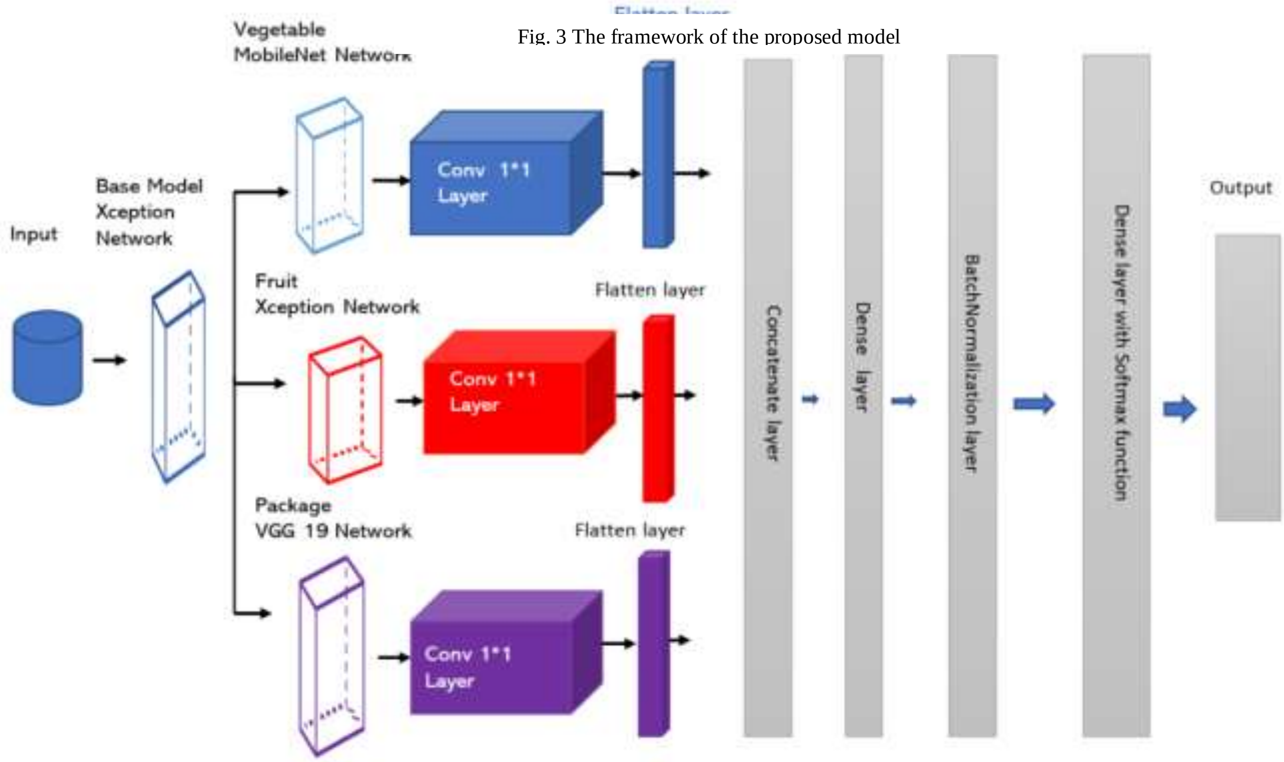


Fig. 4 The overall architecture of end-to-end model

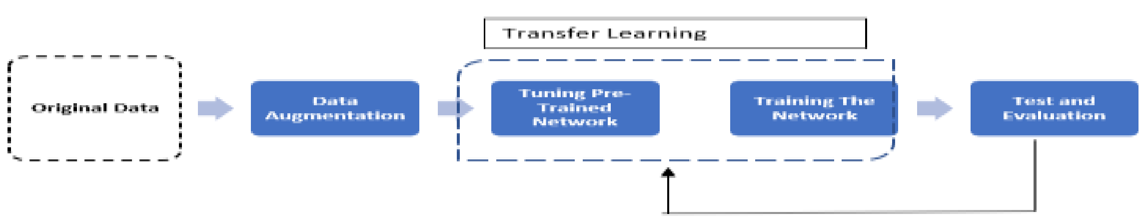


Fig.5 Block diagram of the proposed approach for the classification of different grocery dataset.

TABLE II. PERFORMANCE EQUATIONS SUMMARY

Assessments	Equation	Equ.No
Accuracy (Acc)	$\frac{TP + TN}{TP + TN + FP + FN}$	(1)
F1-Score (F)	$2 * \frac{P*R}{P+R}$	(2)
Precision (P)	$\frac{TP}{TP + FP}$	(3)
Recall (R)	$\frac{TP}{TP + FN}$	(4)

Where: True Positive (TP), the model properly predicts the positive class. The True Negative (TN) model classifies the negative class properly. The model predicts the positive class incorrectly in a false positive (FP). The model predicts the negative class incorrectly in false negative (FN).

C. Experimental Results and Analysis

Here, we present the findings of 6 experiments On the base model, and the results of a comprehensive experimental study of *i.* multiclass images of grocery stores using a pretrained CNN model and a hybrid model are reported. In addition, a comparative examination of various models is presented, and the findings gained from these models are compared to current state-of-the-art methods. Finally, the best-performing model is identified.

Unlike classical neural networks for image recognition tasks, where image features are defined *ii* manually, CNNs allow us to extract higher representations of the image content. However, there is a lack of sufficiently large datasets; thus, very few individuals train a full CNN from the start (with random initialization). Instead, it is common practice to pretrain a ConvNet on a large dataset (e.g., ImageNet, which contains 1.2 million images) and then utilize the pretrained ConvNet as initialization or a fixed feature extractor to solve the target problem, which is the concept behind TL. Given the vast resources required to *iii* train deep learning models or the extensive and complex datasets on which deep learning models are trained.

Accordingly, we pretrained several networks and compared each network's performance to identify the best model for our task. The broad framework for the TL phases we employed is as follows:

1. Load a pretrained CNN model trained on an extensive dataset.
2. Freeze the parameters of the model's lowest convolutional layers(i.e., the weights).
3. Add a custom classifier with numerous layers of trainable parameters to the model.
4. Train the classifier layers using the available training data.
5. Adjust the hyperparameters(Learning rate, Number of epochs) and unfreeze additional layers as required.

The following steps are involved in fine-tuning a pretrained model:

(i) Set up new fully connected and output layers.
(ii) Freeze the pretrained layers
(iii) Unfreeze and retrain the last few pretrained layers.

1) Base model networks

Starting with the base model, we evaluated several pretrained networks simultaneously trained on the ImageNet dataset to classify the dataset into three main classes (i.e., fruits, vegetables, and packages). 272

AlexNet

AlexNet[32] is a leading architect in pretrained models in the computer vision domain. However, it is not available in Keras; thus, we implemented the architecture as shown in Fig. 6. After several runs with different optimizers and epochs, training-validation accuracy, and loss alongside evaluation on the test set did not show good results with that dataset (see Fig. 7)

ResNet

ResNet (residual networks) first presented the skip connection concept to solve the vanishing gradient issue, also known as a residual block or identity block. We trained ResNet-50 and ResNet101-V2, Training-validation accuracy and loss over 30 epochs for ResNet50 with RMSprop optimizer, ResNet50 with SGD optimizer, and ResNet101-V2 with RMSprop optimizer. The results are summarized in Fig. 7. However, as can be seen, it does not fit the data well.

VGG

We found that the VGG models fit the data well. We demonstrated stable accuracy graphs (Fig. 7). Training-validation accuracy and loss with Adam optimizer [33], the learning rate of 0.001 over 50 epochs for VGG16 network, VGG19 network, we can also notice that the precision and recall metrics of the test set showed obvious model mistakes and misclassification, specifically in the vegetable class.

iv. Inception

We evaluated Inception-V2 and Inception-V3, and the results are shown in Fig. 7. We found that Inception-V2 obtainedbetter results than Inception-V3; however, misclassification problems with the vegetable class were observed. Training with RMSprop optimizer, the learning rate of 0.0001, over 30 epochs for Inception-V2 network, Inception-V3 network.

v. MobileNet

We evaluated two versions of MobileNet on the base model. Although the confusion matrices on the test set evaluation exhibited promising results with few mistakes, the results of the learning curves were very noisy, including many oscillations, which means that the model could not fit many batches of the validation set and could not be generalized. The results are presented in Fig. 7.

vi. Xception

Xception stands for Extreme version of Inception that relies on modified deep-wise separable convolutional layers[20]. Regardless of the training cost, Xception shows good improvements over the Inception network. We found that Xception was the best model to fit our dataset, demonstrating both stable learning curves and good evaluation results on the test set, as shown in Figs. 7 and 8. Thus, Xception was selected as the final base model.

2) Final model results

We observed that the data required rearranging. The dataset is divided into three folders training, validation, and test. We discovered that the dataset proposed by Klasson et al.[13] had some classes arranged incorrectly, i.e., some classes in the test set did not exist in the training or validation sets (and vice versa),and that the training, validation, and testing sets were not split logically with the correct percentages.

Thus, we integrated all images and split the dataset into training, validation, and testing sets at a ratio of 70:10:20, respectively, as shown in Table III. As a result, there were 3780 images in the training set, 508 images in the validation set, and 1156 images in the testing set. We started training the second-level models of the hierarchy (show Fig.2) that will subclassify each category into its subclasses. Here, we started training the same previous pre-trained models on the subclasses of each category; we identified the models that exhibited the best-fit reaching. In addition, for precision, we retrained the Xception base model again on the rearranged data.

TABLE III. DESCRIPTION OF THE UTILIZED DATASET AFTER AUGMENTATION

Category	**Training set**	**Validation set**	**Testing set**
fruit	1649	224	497
vegetables	913	122	281
packages	1218	162	378
Total	3780	508	1156

Following, we list the best-resulting model for each category along with its learning curves, confusion matrix, and evaluation metrics on the test set:

1. *Base model (Xception)*

Fig. 8 shows the learning curves of the xception network for our base model after retraining, Fig. 9 shows the confusion matrix of the test set, and Table IV summarizes the test evaluation metrics.

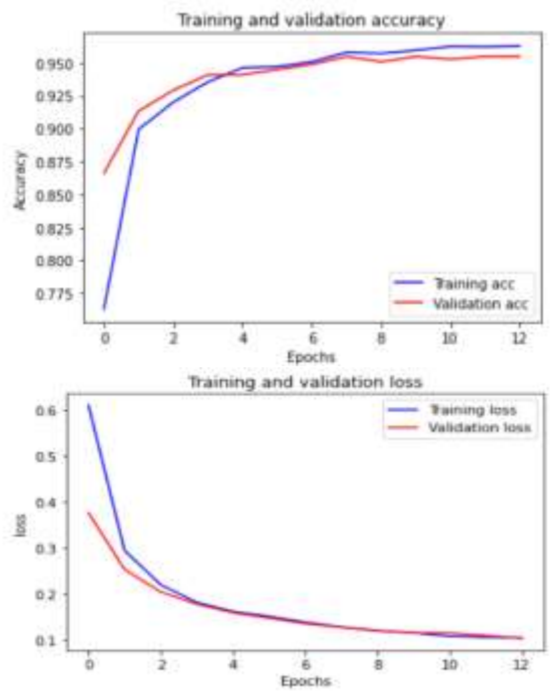


Fig. 8Training-validation accuracy and loss graphs of the Xception model over 50 epochs.

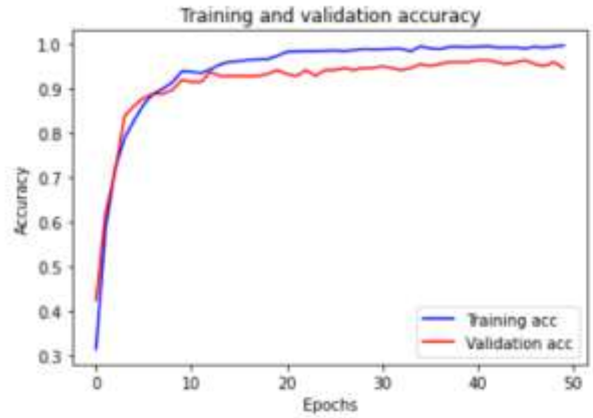
Fig.9Confusion matrix of the test set(Base Model).

TABLE IV. EVALUATION METRICS OF A BASE MODEL

Testing set	
	Base model Xception
Precision (weighted avg)	98.0
Recall (weighted avg)	98.0
F1-score (weighted avg)	98.0
Accuracy	98.0

2. *Fruits model (Xception)*

Xception network once again outstanding the fruits data and was by far the best model. We can notice this in the learning curves (Fig. 10) and the test set evaluation metrics (Table V).



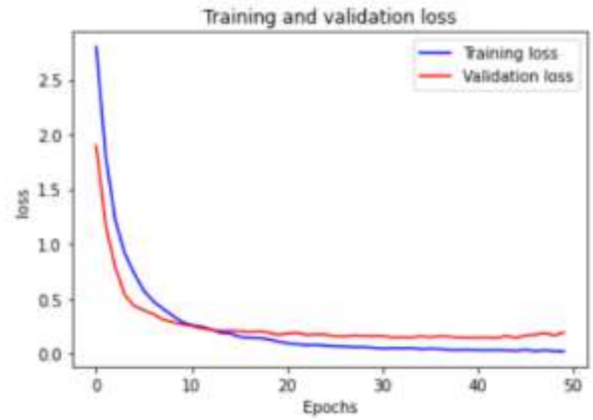


Fig.10Training-validation accuracy and loss graphs of fruits model for the Xception network

TABLE V. EVALUATION METRICS OF THE FRUITS MODEL

Testing set	
	Fruits Xception
Precision (weighted avg)	98.0
Recall (weighted avg)	98.0
F1-score (weighted avg)	98.0
Accuracy	98.0

3. *Vegetables model (MobileNet)*

Vegetable classes were very challenging, and we evaluated several networks with multiple fine-tuning and different parameters; at last, MobileNet was the best network for the vegetable data. Fig. 11 and Table VI show the train-validation accuracy, loss graphs, and evaluation metrics.

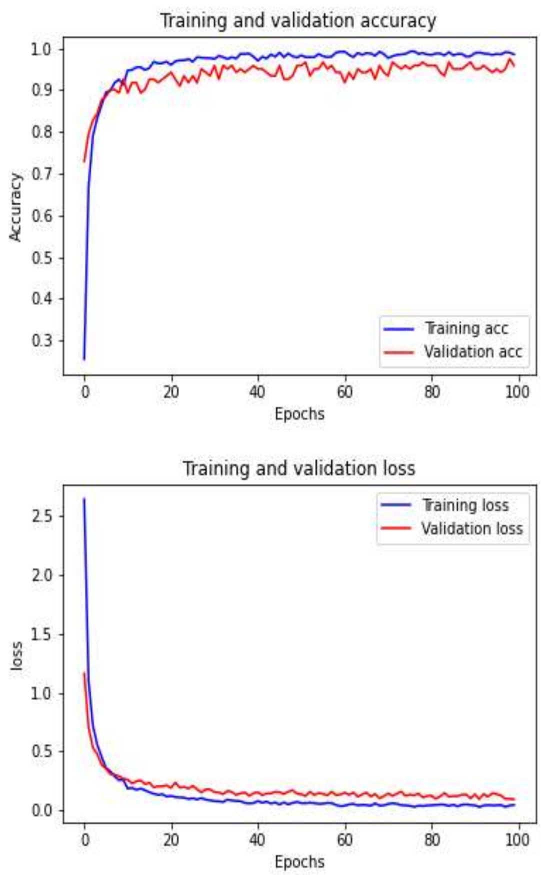


Fig.11Training-validation accuracy and loss graphs of vegetables model with MobileNet network

TABLE VI. EVALUATION METRICS OF THE VEGETABLE MODEL

Testing set	
	Vegetables MobileNet
Precision (weighted avg)	95.0
Recall (weighted avg)	95.0
F1-score (weighted avg)	94.0
Accuracy	95.0

4. Packages model (VGG19)

VGG-19 was the best network for package data, and the results are shown in Fig. 12 for learning curves and Table VII for evaluation metrics on the test set.

The proposed approach based on hybrid transfer learning models achieved superior accuracy on the approaches based on CNNs and on the approaches found in the literature (see Table VIII).

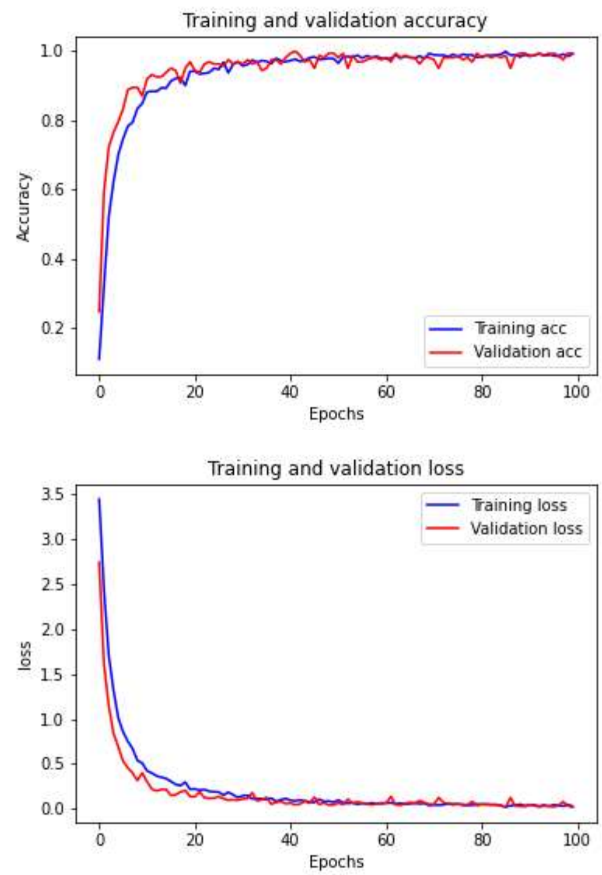


Fig.12Training-validation accuracy and loss graphs of packages model with VGG19 network

TABLE VII. EVALUATION METRICS OF THE PACKAGES MODEL

Testing set	
	Packages VGG19
Precision (weighted avg)	97.0
Recall (weighted avg)	96.0
F1-score (weighted avg)	97.0
Accuracy	97.0

Table VIII. Literature accuracy results versus the best approaches of this work

Database		Accuracy (in percentage)				
		State-of-Art	Proposed model			
			Base_model	Fruit model	Vegetable model	Package model
SHORT-100	ST-	77.51	________		________	

[9]	SG					
	VF-SG	69.41				
Grocery [10]		92.3	___________________		___________________	
Freiburg Groceries [11]		78.9	___________________		___________________	
Grocery Store[13]		85.0	98.0	98.0	94.0	97.0
		85.2				

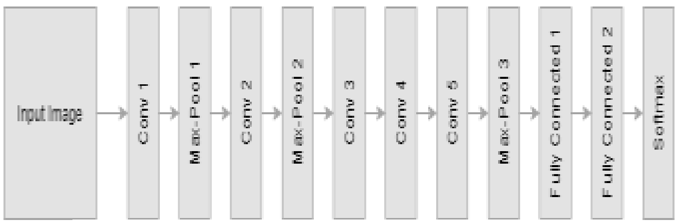


Fig.6 Alex architecture

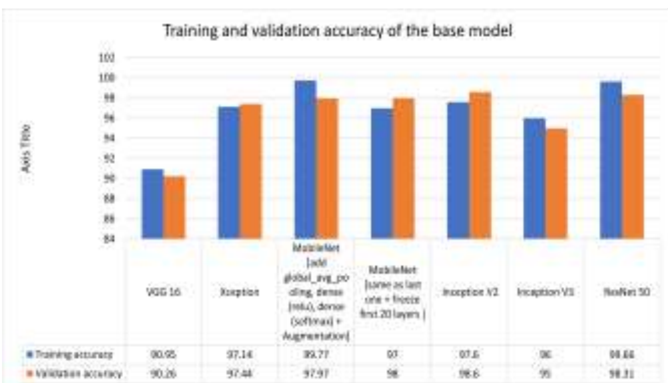


Fig.7 Training and validation accuracy of the base model

VI. CONCLUSION

To identify the grocery store dataset, we proposed a unique framework of transfer learning in pre-trained deep learning architectures based on VGG19, Xception, and MobileNetV2. We made use of a real-world dataset. Fine-tuning pre-trained models weighted on the ImageNet dataset was used in our proposed methodology. The experiment findings show that pre-trained Xception on the base model is effective, with an F1-score of 98% on the testing set. The Xception network performed the best for the fruits model, achieving an F1-score of 98% on the testing set. MobileNet was found as the top model for the vegetables model, with an F1-score of 94% on the testing set. The VGG19 network performed best for the packages model, achieving an F1-score of 97% on the testing set. The experiment results could be relevant in applying the model obtained in mobile phones. People can benefit from learning without restrictions. An essential element is that consumers are more knowledgeable of the many types of grocery store products, particularly for the visually impaired. In future work, we will research and test alternative CNN architectures as well as other well-known one-stage detectors such as a YOLO or Siamese neural network to identify an appropriate one for classifying grocery store species.

REFERENCES

[1] R.R.A. Bourne et al, Causes of blindness and vision impairment in 2020 and trends over 30 years, and prevalence of avoidable blindness in relation to VISION 2020: The Right to Sight: An analysis for the Global Burden of Disease Study, Lancet Glob Health. 9 (2021) e144–e160. https://doi.org/10.1016/S2214-109X(20)30489-7.

[2] D. Bal, M.M. Islam Tusher, M. Rahman, M.S. Rahman Saymon, NAVIX: A Wearable Navigation System for Visually Impaired Persons, 2020 2nd International Conference on Sustainable Technologies for Industry 4.0, STI 2020. (2020). https://doi.org/10.1109/STI50764.2020.9350480.

[3] G. Vaidya, K. Vaidya, K. Bhosale, Text recognition system for visually impaired using portable camera, 2020 International Conference on Convergence to Digital World - Quo Vadis, ICCDW 2020. (2020). https://doi.org/10.1109/ICCDW45521.2020.9318706.

[4] L.P. Sousa, R.M.S. Veras, L.H.S. Vogado, L.S. Britto Neto, R.R.V. Silva, F.H.D. Araujo, F.N.S. Medeiros, Banknote Identification Methodology for Visually Impaired People, International Conference on Systems, Signals, and Image Processing. 2020-July (2020) 261–266. https://doi.org/10.1109/IWSSIP48289.2020.9145294.

[5] L. de Sousa Britto Neto, V.R.M.L. Maike, F.L. Koch, M.C.C. Baranauskas, A. de Rezende Rocha, S.K. Goldenstein, A wearable face recognition system built into a smartwatch and the visually impaired user, ICEIS 2015 - 17th International Conference on Enterprise Information Systems,

Proceedings. 3 (2015) 5–12. https://doi.org/10.5220/0005370200050012.

[6] K. Weiss, T.M. Khoshgoftaar, D.D. Wang, A survey of transfer learning, J Big Data. 3 (2016) 1–40. https://doi.org/10.1186/S40537-016-0043-6/TABLES/6.

[7] L. Hakobyan, J. Lumsden, D. O'Sullivan, H. Bartlett, Mobile assistive technologies for the visually impaired, Surv Ophthalmol. 58 (2013) 513–528. https://doi.org/10.1016/J.SURVOPHTHAL.2012.10.004.

[8] M. Shamim Hossain, M. Al-Hammadi, G. Muhammad, Automatic Fruit Classification Using Deep Learning for Industrial Applications, IEEE Trans Industr Inform. 15 (2019) 1027–1034. https://doi.org/10.1109/TII.2018.2875149.

[9] A Systematic Review on Product Recognition for Aiding Visually Impaired People | IEEE Latin America Transactions, (n.d.). https://latamt.ieeer9.org/index.php/transactions/article/view/3760 (accessed April 11, 2022).

[10] J. Rivera-Rubio, S. Idrees, I. Alexiou, L. Hadjilucas, A.A. Bharath, Small Hand-held Object Recognition Test (SHORT), 2014 IEEE Winter Conference on Applications of Computer Vision, WACV 2014. (2014) 524–531. https://doi.org/10.1109/WACV.2014.6836057.

[11] G. Varol, R.S. Kuzu, Toward retail product recognition on grocery shelves, Sixth International Conference on Graphic and Image Processing (ICGIP 2014). 9443 (2015) 944309. https://doi.org/10.1117/12.2179127.

[12] P. Jund, N. Abdo, A. Eitel, W. Burgard, The Freiburg Groceries Dataset, (2016). https://doi.org/10.48550/arxiv.1611.05799.

[13] M. Klasson, C. Zhang, H. Kjellström, A Hierarchical Grocery Store Image Dataset with Visual and Semantic Labels, (2019). http://arxiv.org/abs/1901.00711 (accessed March 9, 2022).

[14] K. Simonyan, A. Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings. (2014). https://doi.org/10.48550/arxiv.1409.1556.

[15] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the Inception Architecture for Computer Vision, Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2016-December (2016) 2818–2826. https://doi.org/10.1109/CVPR.2016.308.

[16] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2016-December (2016) 770–778. https://doi.org/10.1109/CVPR.2016.90.

[17] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet Classification with Deep Convolutional Neural Networks, (n.d.). http://code.google.com/p/cuda-convnet/ (accessed June 6, 2022).

[18] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 07-12-June-2015 (2015) 1–9. https://doi.org/10.1109/CVPR.2015.7298594.

[19] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L.C. Chen, MobileNetV2: Inverted Residuals and Linear Bottlenecks, Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. (2018) 4510–4520. https://doi.org/10.1109/CVPR.2018.00474.

[20] F. Chollet, Xception: Deep learning with depthwise separable convolutions, Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017. 2017-January (2017) 1800–1807. https://doi.org/10.1109/CVPR.2017.195.

[21] G. Huang, Z. Liu, L. van der Maaten, K.Q. Weinberger, Densely connected convolutional networks, Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017. 2017-January (2017) 2261–2269. https://doi.org/10.1109/CVPR.2017.243.

[22] J. Deng, W. Dong, R. Socher, L.-J. Li, Kai Li, Li Fei-Fei, ImageNet: A large-scale hierarchical image database, (2010) 248–255. https://doi.org/10.1109/CVPR.2009.5206848.

[23] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, Q. He, A Comprehensive Survey on Transfer Learning, Proceedings of the IEEE. 109

(2021) 43–76. https://doi.org/10.1109/JPROC.2020.3004555.

[24] L. Alzubaidi, J. Zhang, A.J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M.A. Fadhel, M. Al-Amidie, L. Farhan, Review of deep learning: concepts, CNN architectures, challenges, applications, future directions, J Big Data. 8 (2021). https://doi.org/10.1186/s40537-021-00444-8.

[25] W. Rawat, Z. Wang, Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review, Neural Comput. 29 (2017) 2352–2449. https://doi.org/10.1162/NECO_A_00990.

[26] K. Maharana, S. Mondal, B. Nemade, A Review: Data Pre-Processing and Data Augmentation Techniques, Global Transitions Proceedings. (2022). https://doi.org/10.1016/J.GLTP.2022.04.020.

[27] A. Asperti, C. Mastronardo, The Effectiveness of Data Augmentation for Detection of Gastrointestinal Diseases from Endoscopical Images, (2017). http://arxiv.org/abs/1712.03689 (accessed June 7, 2022).

[28] S. Ioffe, C. Szegedy, Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift, (2015) 448–456. https://proceedings.mlr.press/v37/ioffe15.html (accessed September 24, 2022).

[29] Y. Lecun, Y. Bengio, G. Hinton, Deep learning, Nature 2015 521:7553. 521 (2015) 436–444. https://doi.org/10.1038/nature14539.

[30] E. Bisong, Building Machine Learning and Deep Learning Models on Google Cloud Platform, Building Machine Learning and Deep Learning Models on Google Cloud Platform. (2019). https://doi.org/10.1007/978-1-4842-4470-8.

[31] Keras Team, “Getting started,” Keras.io. [Online]. Available: https://keras.io/getting_started/. [Accessed: 24-Sep-2022].

[32] M.Z. Alom, T.M. Taha, C. Yakopcic, S. Westberg, P. Sidike, M.S. Nasrin, B.C. van Esesn, A.A.S. Awwal, V.K. Asari, The History Began from AlexNet: A Comprehensive Survey on Deep Learning Approaches, (2018). https://doi.org/10.48550/arxiv.1803.01164.

[33] D.P. Kingma, L.J. Ba, Adam: A Method for Stochastic Optimization, (2015).