Advances in Intelligent Systems and Computing 1270

Milan Tuba
Shyam Akashe
Amit Joshi *Editors*

ICT Systems and Sustainability

Proceedings of ICT4SD 2020, Volume 1

Springer

Advances in Intelligent Systems and Computing

Volume 1270

The series “Advances in Intelligent Systems and Computing” contains publications on theory, applications, and design methods of Intelligent Systems and Intelligent Computing. Virtually all disciplines such as engineering, natural sciences, computer and information science, ICT, economics, business, e-commerce, environment, healthcare, life science are covered. The list of topics spans all the areas of modern intelligent systems and computing such as: computational intelligence, soft computing including neural networks, fuzzy systems, evolutionary computing and the fusion of these paradigms, social intelligence, ambient intelligence, computational neuroscience, artificial life, virtual worlds and society, cognitive science and systems, Perception and Vision, DNA and immune based systems, self-organizing and adaptive systems, e-Learning and teaching, human-centered and human-centric computing, recommender systems, intelligent control, robotics and mechatronics including human-machine teaming, knowledge-based paradigms, learning paradigms, machine ethics, intelligent data analysis, knowledge management, intelligent agents, intelligent decision making and support, intelligent network security, trust management, interactive entertainment, Web intelligence and multimedia.

The publications within “Advances in Intelligent Systems and Computing” are primarily proceedings of important conferences, symposia and congresses. They cover significant recent developments in the field, both of a foundational and applicable character. An important characteristic feature of the series is the short publication time and world-wide distribution. This permits a rapid and broad dissemination of research results.

Indexed by SCOPUS, DBLP, EI Compendex, INSPEC, WTI Frankfurt eG, zbMATH, Japanese Science and Technology Agency (JST), SCImago.

All books published in the series are submitted for consideration in Web of Science.

More information about this series at http://www.springer.com/series/11156

Milan Tuba · Shyam Akashe · Amit Joshi
Editors

ICT Systems and Sustainability

Proceedings of ICT4SD 2020, Volume 1

Springer

Editors
Milan Tuba
Singidunum University
Belgrade, Serbia

Shyam Akashe
ITM University
Gwalior, Madhya Pradesh, India

Amit Joshi
Global Knowledge Research Foundation
Ahmedabad, Gujarat, India

ISSN 2194-5357 ISSN 2194-5365 (electronic)
Advances in Intelligent Systems and Computing
ISBN 978-981-15-8288-2 ISBN 978-981-15-8289-9 (eBook)
https://doi.org/10.1007/978-981-15-8289-9

This Springer imprint is published by the registered company Springer Nature Singapore Pte Ltd.
The registered company address is: 152 Beach Road, #21-01/04 Gateway East, Singapore 189721, Singapore

Preface

The Fifth International Conference on ICT for Sustainable Development (ICT4SD 2020) targets theory, development, applications, experiences and evaluation of interaction sciences with fellow students, researchers and practitioners. The conference is devoted to increasing the understanding role of technology issues, how engineering has day by day evolved to prepare human-friendly technology. The conference provided a platform for bringing forth significant research and literature across the field of ICT for Sustainable Development and provided an overview of the technologies awaiting unveiling. This interaction will be the focal point for leading experts to share their insights, provide guidance and address participant's questions and concerns. The conference was to be held during July 23–24, 2020, at Hotel Vivanta by Taj, Panaji, Goa, India, but due to the pandemic, this year it was organized through digital mode. The conference was organized by Global Knowledge Research Foundation, Supporting Partner InterYIT, International Federation for Information Processing, State Chamber Partner Goa Chamber of Commerce & Industry, National Chamber Partner as Knowledge Chamber of Commerce & Industry. Research submissions in various advanced technology areas were received after a rigorous peer review process with the help of program committee members and 187 external reviewers for 1000+ papers from 19 different countries including Algeria, USA, United Arab Emirates, Serbia, Qatar, Mauritius, Egypt, Saudi Arabia, Ethiopia, Oman out of which 160 were accepted with an acceptance ratio of 0.15. Technology is the driving force of progress in this era of globalization. Information and communication technology (ICT) has become a functional requirement for the socioeconomic growth and sustained development of any country. The influence of information communications technology (ICT) in shaping the process of globalization, particularly in productivity, commercial and financial spheres, is widely recognized. The ICT sector is undergoing a revolution that has momentous implications for the current and future social and economic situation of all the countries in the world. ICT plays a pivotal role in empowering people for self-efficacy and how it can facilitate this mission to reach out to grassroots level. Finally, it is concluded that ICT is a significant contributor to the success of the ongoing initiative of Startup India. In order to recognize and reward the extraordinary performance and achievements by ICT and allied sectors and promote universities, researchers and students through their research work adapting new scientific technologies and innovations. The two-day conference had presentations from the researchers, scientists, academia and students on the research work carried out by them in different sectors. ICT4SD Summit is a flagship event of the G R Foundation. This is the fourth edition. The summit was addressed by eminent dignitaries including Shri Manguirsh Pai Raikar, Chairperson, ASSOCHAM MSME National Council; Shri. Prajyot Mainkar, Chairman, IT Committee of Goa Chamber of Commerce and Industry; Mike Hinchey, President, IFIP and Chair IEEE, UK and Ireland; Milan Tuba, Vice-Rector for International Relations, Singidunum University, Serbia; Prof. Lance Fung, Australia; Prof. Jagdish Bansal, India; Mr. Aninda Bose, Springer; Dr. Amit Joshi, Director, G R Foundation. The overall conference had one inaugural session, one keynote session and 18 technical sessions during two days.

Belgrade, Serbia Milan Tuba
Gwalior, India Shyam Akashe
Ahmedabad, India Amit Joshi

Contents

About the Editors

Milan Tuba is the Vice Rector for International Relations, Singidunum University, Belgrade, Serbia and was the Head of the Department for Mathematical Sciences at State University of Novi Pazar and the Dean of the Graduate School of Computer Science at John Naisbitt University. He received B.S. in Mathematics, M.S. in Mathematics, M.S. in Computer Science, M.Ph. in Computer Science, Ph.D. in Computer Science from University of Belgrade and New York University. From 1983 to 1994 he was in the U.S.A. first at Vanderbilt University in Nashville and Courant Institute of Mathematical Sciences, New York University and later as Assistant Professor of Electrical Engineering at Cooper Union School of Engineering, New York. During that time he was the founder and director of Microprocessor Lab and VLSI Lab, leader of scientific projects and theses supervisor. From 1994 he was Assistant Professor of Computer Science and Director of Computer Center at University of Belgrade, from 2001 Associate Professor, Faculty of Mathematics, University of Belgrade, from 2004 also a Professor of Computer Science and Dean of the College of Computer Science, Megatrend University Belgrade. He was teaching more than 20 graduate and undergraduate courses, from VLSI Design and Computer Architecture to Computer Networks, Operating Systems, Image Processing, Calculus and Queuing Theory. His research interest includes nature-inspired optimizations applied to computer networks, image processing and combinatorial problems. Prof. Tuba is the author or coauthor of more than 200 scientific papers and co editor or member of the editorial board or scientific committee of a number of scientific journals and conferences. He was invited and delivered around 60 keynote and plenary lectures at international conferences. Member of the ACM, IEEE, AMS, SIAM, IFNA.

Shyam Akashe is a Professor at ITM University, Gwalior, Madhya Pradesh, India. He completed his Ph.D. at Thapar University, Punjab, and his M.Tech. in Electronics and Communication Engineering at the Institute of Technology & Management, Gwalior. He has authored 190 publications, including more than 50 papers in SCI-indexed journals. His main research focus is low-power system on chip (SoC) applications in which static random access memories (SRAMs) are omnipresent. He has authored two books entitled Moore's Law Alive: Gate-All-Around (GAA) Next Generation Transistor published by LAMBERT Academic Publishing, Germany, and Low Power High Speed CMOS Multiplexer Design published by Nova Science Publishers, Inc., New York, USA. He has also published over 120 papers in leading national and international refereed journals of repute. Dr. Akashe has participated in numerous national and international conferences and presented over 100 papers.

Amit Joshi is currently the Director of Global Knowledge Research Foundation, also an Entrepreneur & Researcher who has completed his Masters and research in the areas of cloud computing and cryptography in medical imaging. Dr. Joshi has an experience of around 10 years in academic and industry in prestigious organizations. Dr. Joshi is an active member of ACM, IEEE, CSI, AMIE, IACSIT, Singapore, IDES, ACEEE, NPA, and many other professional societies. Currently, Dr. Joshi is the International Chair of InterYIT at International Federation of Information Processing (IFIP, Austria). He has presented and published more than 50 papers in national and international journals/conferences of IEEE and ACM. Dr. Joshi has also edited more than 40 books which are published by Springer, ACM, and other reputed publishers. Dr. Joshi has also organized more than 50 national and international conferences and programs in association with ACM, Springer, and IEEE to name a few across different countries including India, UK, Europe, USA, Canada, Thailand, Egypt, and many more.

Classification of Leaves Using Convolutional Neural Network and Logistic Regression

Sapan Naik and Hinal Shah

Abstract The dissemination of this research paper proffers the method of classifying and recognizing the different leaves of plants from their image. The leaves of plants contain different features which vary from each other considering from its shape, size, texture, and color. The image of the leaf is captured from each curve, and the dataset of 50,000 images has been prepared for 50 plants (1000 images/plant) of South Gujarat. Pre-trained convolutional neural network (CNN) is utilized as a feature extractor, and logistic regression (LR) is used as a classifier for leaf classification. CNN is visualized using deconvolutional network (DN) to get an insight into extracted leaf's features. The process of leaf classification includes (1) dataset preparation and (2) features are extracted using pre-trained CNN models. This paper outlines six CNN models which are experimented, namely as Inception v4, Xception, ResNet, InceptionResNetV2, DenseNet, and MobileNet. (3) Stated on the extracted features, LR is trained and used for classification. (4) Finally used DN for visualizing features, which provides the insight into classification results. In experiments, Xception confers the highest Rank-1 accuracy of 93.4% and ResNet withholds with poor performance with 36.48% Rank-5 accuracy. MobileNet nearly takes 2.17 min for training and classification which is the fastest. DenseNet is the slowest and takes 30.48 min for leaf classification. MobileNet is accurate and is relatively close to Xception model and pursue of exceptionally fast execution. In our experiments, DN claims the shape and texture of a leaf (diverse orders of venation) are significant features for classification.

Keywords Convolutional neural network · Deconvolutional network · Feature extractions · Leaves classification · Logistic regression

S. Naik (✉) · H. Shah
Babu Madhav Institute of Information Technology, Uka Tarsadia, Surat, Gujrat 394350, India
e-mail: Sapan2307@gmail.com

M. Tuba et al. (eds.), *ICT Systems and Sustainability*, Advances in Intelligent Systems and Computing 1270, https://doi.org/10.1007/978-981-15-8289-9_7

1 Introduction

We live in a diverse world where there are indistinct trees. Trees and plants are the residual of the ecosystem and which are the necessity of our existence and sustainability. There are several plants which are not categorized in urban and rural streets. Approximately, 20% of the world population can differentiate them and the rest of us are not aware of the species of plants [1]. Hence, it is a laborious task to distinguish between them; we are going to bifurcate their leaves using computer vision technology.

Nonetheless, we know some of the species of trees and plants; still, it is quite challenging to differentiate similar kinds of leaves with precision. To maintain the perfection of these leaves, all the direct, as well as indirect details relevant to the leaves, should be acknowledged. All the minor details like the shape of the leaf, the sub-branch of the leaf, its texture, size, color are important for the identification of leaves [2].

Latterly, deep learning and CNN have availed a lot of popularity. Deep learning extracts the features of the image and drastically reduces the error in image recognition. CNN has been dominating in the field of image classification since long [3]. The influence of deep learning has been observed, when the team of Hinton won the ImageNet in image classification competition [4]. Plenty of work was done on the initial CNN model, and at the present moment, there are many modern CNN architecture models available which have outstanding performance. Some of them are Inception, ResNet, Xception, and MobileNet [5].

2 Related Work

By using Caffe model of deep convolutional network for implementation, an average accuracy of 96.3% was accomplished. Resulting to this, 44 different species of plants of Royal Botanic Gardens, Kew, England, had been classified using CNN [2] and 13 different types of plant diseases were identified out of healthy leaves [6], and visualization technique based on the deconvolutional networks was utilized for visualizing features. LeafNet—a plant identification system was developed [7] based on CNNs. LeafSnap, Flavia, and Foliage datasets were used for training and evaluating CNN. Results showed better performance compared to handcrafted customized systems.

For classification of the leaf, Life CLEF 2015 dataset was used [8]. CNN was used with three architectural models, namely GoogLeNet, AlexNet, and VGGNet. It attained an overall accuracy of 80% and an inverse rank score of 0.752. With the usage of CNN, leaf classification was implemented, where they had clinched that the shape is not a dominant feature for leaf, and it was more preferable to opt for different orders of venation. Deconvolutional network was used to get insight into chosen features. Plant identification from leaf vein patterns using CNN was proposed in [9] three different legume species: white bean, red bean, and soybean had been classified for the experiment. Fine-grained plant classification system submitted to LifeCLEF 2014 was presented in [10] where CNN was used, and they had achieved the score of 0.249. Forming of the complication in plant classification, digital morphometric techniques were reviewed in [11]. LeafSnap—a mobile application for plant species identification was developed using computer vision algorithms in [12]. It used the traditional method of image capturing, segmentation, feature extraction, and classification using available dataset. Dataset contains samples of 184 trees in the Northeastern USA.

Dataset was assembled in a native environment for the classification of plants where 10,000 images of 100 ornamental plant species were captured using the mobile phone [13] in Beijing Forestry University campus. Deep learning approach was used and achieved a recognition rate of 91.78% on the BJFU100 dataset. A tree classification system treelogy was established [14], where CNN and SVM were used to classify trees based on their leaves. Dataset of 57 trees with 5408 leaf images was prepared. Diseased leaf segmentation and recognition method were proposed in [15] using the fusion of superpixel, K-means, and pyramid of histograms of orientation gradients (PHOG) algorithm.

To increase the dataset size, six augmentation techniques were applied on three leaf datasets, namely Folio, AgrilPlant, and the Swedish in [16]. CNN was used to implement leaf classification task using AlexNet and GoogleNet. Seven new invariants (translation, rotation, and scaling invariants) for multi-component shapes were proposed in [17] for leaf classification issue. New invariants were robust to noise and mild deformations.

2.1 Contributions

A dataset with 50,000 images of leaves has been created for 50 different plants of South Gujarat, and the approach is presented for classification of leaves. CNN is used for feature extraction process with six architecture models, namely Inception v4, Xception, InceptionResNetV2, DenseNet, ResNet50, and MobileNet and to get its insight and extracted features (based on which leaves are categorized); DN is used. LR classifier is utilized for training and classification purpose. All the outcomes of the experiment have been compared with available work.

3 Materials and Methods

The preparation of dataset is discussed with the data augmentation method used. Here in this division, how to train, tune, and implement the CNN model is briefed. Later, short details of CNN architecture models are provided. At the end of this division, LR and DN are briefly discussed.

3.1 Dataset Preparation and Data Augmentation

The dataset of 50,000 images was befallen to categorize the leaves. For preparing the database, fifty different plants of South Gujarat were selected. Indiscriminating leaves, 100 s of them were gathered concerning each plant. Each of them is determined and verified by the native citizens (aged above 48 years) living in the vicinity of a particular plant. Total 1000 images of leaves are captured from 100 leaves (ten images of each leaf with different positions). For dataset preparation, iPhone 6 s is used to capture images. All the images are captured from top view (putting leaf beneath the camera) in daylight. White paper is used as background for capturing all images to reduce segmentation issue. The size (resolution) of captured images is 2448×2448 pixels. Images are resized to 224×224 and 299×299 based on CNN model. The two beneficial reasons for resizing are 1. CNN needs fixed-size input images and 2. to reduce computational time.

For reducing overfitting problem, dataset's size is increased by applying position shifting, rotation, and scaling. Even brightness and contrast corrections are performed, for increasing size of dataset. As CNN does not require any preprocessing on images except resizing, no other modifications have been performed on the dataset. We have applied data augmentation to 200 randomly selected images of each plant and created 400 new images out of it for each plant. In this way, we have enlarged the dataset to 70,000 images.

More detail of dataset (i.e., name of each plant whose leaf is considered for preparing dataset with one sample leaf image) is available at https://drive.google.com/open?id=1iN4QLGFYD2AHLMWx-FlcPoLczkp__lqw.

Dataset samples (100 images of each leaf) are available at https://drive.google.com/open?id=1No7P5M7gwaCQC–yEZuph0xpKDXazMua.

3.2 Overview of Convolution Neural Network

Basic CNN architecture contains four layers, namely convolution, nonlinearity, pooling, and fully connected. Each layer may occur multiple times in architecture. The number of each layer depends on the structure/model under use.

3.2.1 Convolution Layer

It is the uppermost layer of CNN which inputs an image. In this layer, filter (kernel or neuron) is applied to the image. Normally, filter size is $3 \times 3 \times$depth, where the depth of input image and filter is the same. Values of filter are multiplied with the pixel value of the image, and new array is created as output. This operation is called convolving, and output of convolving is two-dimensional array which is called feature map or activation map [18].

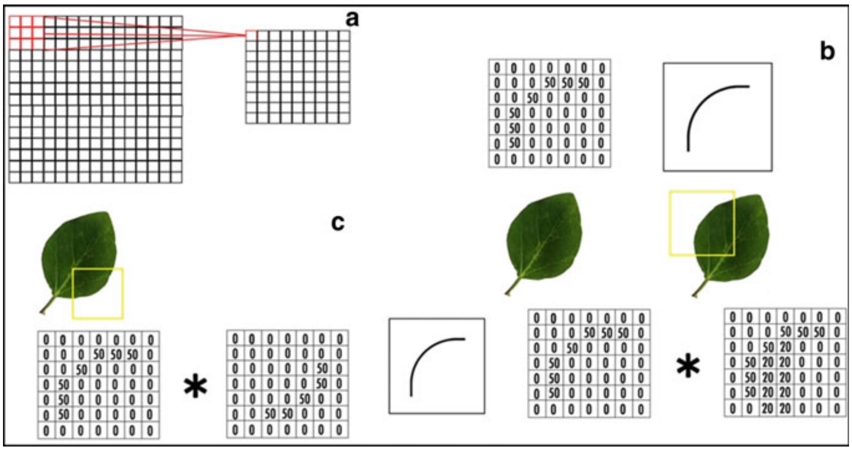

Fig. 1 Operation in convolution layer **a** 3 * 3 filter representation **b** convolving operation when feature is available **c** convolving operation when feature is not available

Various filters are practiced over convolution layer where individual filter represents various features of images like color, curve, or intensity. To spectacle the filters, convolving operation and shape detector filter are shown in Fig. 1.

3.2.2 Nonlinearity Layer

Different activation functions are applied to this layer. Some of these activation functions are Relu, sigmoid, and tanh. Relu is more preferable because it speeds up the training process.

3.2.3 Pooling Layer

Pooling layer is used for reducing the size of the activation map. Pooling reduces the chances of overfitting due to less parameter. Commonly pooling is performed with the size of 2 $\times$ 2. Different forms of pooling are max, mean, and median but max pooling is more popular. The output of pooling layer is given for flattening step where the two-dimensional array is converted into a single vector because this vector goes as an input to an artificial neural network.

3.2.4 Fully Connected Layer

This layer comprises neural network where each of the neurons is connected with the neighbor layer's neuron. Here, this layer sets the weight and bias.

3.3 Tuning the CNN Model

Tuning normally contains three phases, i.e., training, validating, and testing [3]. In the training phase, model is trained using dataset. Later on, this model gets validated, and finally, it gets tested in the testing phase using new input for the same dataset.

Tuning is also associated with choosing the right architecture for CNN or designing the new one [3]. Various architectures are available nowadays like AlexNet, GoogleNet, InceptionResNet, VGG, etc. It is preferable to utilize standard network architecture in the beginning. In tuning CNN, one needs to decide different parameters of the model like weight, biases, epoch, learning rate, validation, and testing percentage. Normally, backward propagation method is used to set these parameters. Once training gets completed, all the parameters are saved in a binary file known as a model.

When new image is under consideration, the saved model is loaded in the same network architecture and prospect of the new image is calculated. The process is termed as inference or prediction. During training, we split the dataset into batches, and we give training in rounds or iterations, which is called epoch.

3.4 Implementation of CNN

There is a technique called transfer learning, which reduces lot of work by using a fully trained model, already trained on dataset like ImageNet. The existing weights for new classes will be retrained. This can be run in less time on the laptop, without requiring a GPU. Time taken depends on the CNN model under consideration. Though it is not as good as full training but is unexpectedly effective.

3.5 CNN Architecture Models

Many architecture models are available for implementing CNN. Here, we have briefly discussed VGGNet, Inception, Xception, ResNet50, DenseNet, and MobileNet.

3.5.1 VGGNet

It was introduced in 2014 by Simonyan and Zisserman [19]. There are two versions of VGGNet, namely VGG16 and VGG19. Here 16 and 19 represent weight layers in the network. The networks are stagnant for training purpose, and in the context of memory, the weight is wider [20]. It has ImageNet top 5 errors as 8.0%, and it considers 138 million parameters [21].

3.5.2 Inception

It is micro-architecture and was first proposed by Szegedy et al. in [22] and later in [23]. Initially, it was termed GoogLeNet. The weight of Inception is smaller compared to VGGNet and requires 96 MB of memory space [20]. Inception uses batch normalization, image distribution, and RMSProp methods. It has ImageNet top 5 errors as 5.6%, and it uses 25 million parameters [24]. The latest version of this model is Inception v4.

3.5.3 ResNet

It is exotic architecture which depends on micro-architecture modules [20]. It was first proposed by He et al. in [25]. It introduces residual connections, and the model size is 102 MB for ResNet50. ResNet has ImageNet top 5 errors as 4.5% and uses 60 million parameters [24]. It uses max as well as average pooling.

3.5.4 Xception

It was presented by [26]. Weight for this model is 91 MB. In experiments, it slightly outperforms the Inception v3 model.

3.5.5 DenseNet

It was introduced by [27] which connects different layers in a feed-forward fashion. The advantages of DenseNet are that it strengthens feature propagation, alleviate the vanishing gradient problem, encourage feature reuse, and substantially reduce the number of parameters.

3.5.6 MobileNet

As the name suggests, it is mainly used for mobile and embedded vision applications. It was proposed by [28]. It is faster in training as well as in execution.

3.6 Logistic Regression

In simple linear regression (Eq. 1), y is replaced with sigmoid function (Eq. 2), so LR takes the form of Eq. 3.

$$y = b0 + b1x \tag{1}$$

$$p = 1/(1 + e^\wedge - y) \tag{2}$$

$$\ln(p/1 - p) = b0 + b1x \tag{3}$$

The reason behind choosing LR is implementation is easy and the availability of different tools. It provides probability scores for observations which are needed in our case [5].

3.7 Deconvolutional Network

DN is used to understand the effect of each filter applied during convolution layer. It facilitates us to visualize the internal working of CNN and how and based on which features it actually categorizes the leaves. The multilayered deconvolutional network was introduced by [29]. This method helps us to understand the function of an individual neuron. Here, feature maps are projected back to its original pixels. All the layers from starting are deconvolved and unpooled for full input pixel space.

4 Results and Discussion

Experiments are performed on the MacBook Pro (13-inch, mid-2012) machine. The machine has 2.5 GHz Intel Core i5 processor, 10 GB 1333 MHz DDR3 memory, and Intel HD Graphics 4000 1536 MB graphics card running on macOS High Sierra (version 10.13.6). Keras and TensorFlow libraries are used for the implementation of CNN, LR, and DN. Implementation needs following simple steps.

Step 1. The training image dataset is prepared with respective labels.
Step 2. Parameters are set in the configuration file.
Step 3. Features are extracted from final fully connected layers of pre-trained CNN and are stored. Using the transfer learning technique, the top layer of the CNN model is retrained [3]. Our dataset of leaves is used for retraining purpose.
Step 4. Machine learning model is trained for extracted features and labels of step 3.
Step 5. The trained model is evaluated and optimized if needed.
Step 6. Visualization of CNN is performed using DN.

Fifty folders (one for each leaf category) are created and labeled with leaf's name for extracting features through CNN (CNN takes each category folder as input). Weights of ImageNet dataset are used for pre-trained CNN model (as transfer learning is used).

To analyze, we randomly selected 200 images of leaves (total 10,000) from each category to extract their features, and 20 images of each category (total 1000) are taken into consideration to provide training to the classifier. For experiments, epoch value is set to 1000, the learning rate is 0.01, training batch size is 100, and validation percentage is 10.

Six CNN architecture models, namely Inception v4, Xception, DenseNet, ResNet50, InceptionResNetV2, and MobileNet are tested. The motive behind choosing these six models is good accuracy is achieved by these models and for CNN performance comparison purpose [5].

After configuration, features are extracted and stored as HDF5 format locally. It is important to check image size while extracting features. Image size of 224 × 224 pixels is chosen for ResNet50 and MobileNet models, and 299 × 299 pixels are chosen for Inception v4, Xception, and InceptionResNetV2 models. Different models require different time for feature extraction process. Table 1 shows the time required by each model to extract features.

After the feature extraction process is completed, features and labels are loaded. For training these features and labels, we have implemented logistic regression and Naïve Bayes classification models. We have compared the classification results of LR and Naïve Bayes with Inception v4 model, where LR performs better compared to Naïve Bayes. Due to this observation, LR is used as a classifier in all experiments.

Experiments results are shown in Table 2. It represents the accuracy achieved by each CNN model as Rank-1 and Rank-5. It also contains execution time required to train 20 images of each leaf (total 1000 images). If the image under consideration is classified with a perfect label with the highest probability, it is called Rank-1 accuracy; and if it is classified within top five probability (may not be first but within first five predicted labels), then it is called Rank-5 accuracy.

Table 1 Feature extraction time by CNN models

Model	Time (minutes)	Model	Time (minutes)
Inception v4	242.55	ResNet50	170.89
Xception	259.09	InceptionResNetV2	341.74
DenseNet	496.14	MobileNet	99.22

Table 2 Rank-1 and Rank-5 accuracy with training execution time for CNN models

	Rank-1 accuracy (%)	Rank-5 accuracy (%)	Time (minutes)
Inception v4	91.91	97.16	5.28
Xception	93.40	97.16	6.08
ResNet50	9.35	36.48	4.51
InceptionResNetV2	88.24	93.35	19.10
DenseNet	88.48	93.35	30.48
MobileNet	91.91	97.16	2.17

Table 3 Misclassification of leaves for CNN models

	Inceptionv4	Xception	ResNet50	InceptionResNetV2	DenseNet	MobileNet
Hibiscus		✓	✓			
Khatumada	✓	✓	✓	✓	✓	
Tecoma	✓	✓	✓	✓	✓	✓
Amli	✓	✓	✓		✓	
Jackfruit			✓			✓
Taggar			✓			✓
Bamboo	✓		✓			
Undirkani	✓		✓	✓	✓	

In Table 3, misclassified leaves are listed with the CNN model which was unable to predict them correctly. Tick mark in the box represents that given model has predicted given leaf incorrectly. Results are shown for all CNN architecture models.

By observing Table 3 and confusion matrix of all models, we have concluded that misclassification majorly happens with tecoma, amli, khatumada, and undirkani category of leaves (shown in Fig. 2).

We have implemented DN as described in [29]. For implementing DN, TensorFlow library is used in the back end and output images are generated in TensorBoard [30]. The visualization results are shown in Fig. 3. Considering the visualization, we can state that contour of the leaf is important feature for classification (Fig. 3b and 3c, the feature map is highlighted with yellow color). At the same time, venation of leaves is also playing an important role in the classification process (Fig. 3b, c), the feature map is highlighted with green color).

There are some standard datasets available to identify leaf classification, but we have not utilized these datasets. We have mainly focused on leaves of South Gujarat and created our own dataset. We have compared our leaves classification results with [7]. Table 4 represents the comparison results.

Fig. 2 Sample of leaf categories which are misclassified

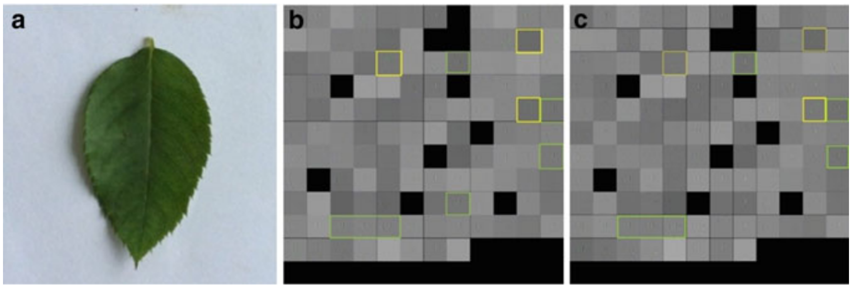

Fig. 3 Visualization at CNN (**a**); sample input image (**b**); visualization at Conv5 layer (**c**); visualization after maxpool 4 layer

Table 4 Comparisons of leaves classification methods

Method	Dataset	Top-1 accuracy (%)
Pierre Barre et al. (2017)	LeafSnap, Flavia, Foliage	86.3, 97.9, 95.8
Kadir (2014)	Flavia, Foliage	97.2, 95.0
Kumar et al. (2012)	LeafSnap	73.0
Wu et al. (2007)	Flavia	90.3
Our approach	New for South Gujarat	93.40

5 Conclusion

From experiments of leaves classification, we have concluded that highest Rank-1 accuracy achieved is 93.4% by Xception model. The fastest model is MobileNet which gives classification output (training and classification) in 2.17 min. The performance of ResNet model is dramatically poor, and DenseNet is found the slowest in our experiments. Performance of MobileNet is very close to Xception model in terms of accuracy, and execution is exceptionally fast. Even the Rank-5 accuracy of MobileNet is the highest with Xception and Inception v4 models. Our experiments show that the majority of the misclassifications occur in tecoma, amli, khatumada, and undirkani leaves. We have observed that shape (counter) is one of the most important features for leaves classification, while venation also plays very vital role for classification.

Acknowledgements Special thanks to 2013–2018 batch students of Babu Madhav Institute of Information Technology for helping in dataset preparation. Help of Yash Rana, Purva Desai, and Vidhi Choksi in implementation and content writing work is acknowledged by authors.

Conflict of Interest
The authors declare that there are no conflicts of interest.

References

1. W.S. Jeon, S.Y. Rhee, Plant leaf recognition using a convolution neural network. Int. J. Fuzzy Logic Intell. Syst. **17**, 26–34 (2017)
2. S.H. Lee, C.S. Chan, P. Wilkin, P. Wilkin, Deep-plant: plant identification with convolutional neural networks, in 2015 IEEE International Conference on Image Processing (ICIP) (2015, September), pp. 452–456
3. A. Sachan, Tensor flow Tutorial 2: Image Classifier Using Convolutional Neural Network [online]. Available cv-tricks.com/tensorflow-tutorial/training-convolutional-neural-network-for-image-classification/. Accessed on 17 June 2018
4. A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in Advances in Neural Information Processing Systems (2012), pp. 1097–1105
5. G. Ilango, Using Keras Pre-trained Deep Learning models for your own dataset. [Online]. Available https://gogul09.github.io/software/flower-recognition-deep-learning. Accessed on 15 July 2018
6. C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, in Proceedings of the IEEE conference on computer vision and pattern recognition (2016), pp. 2818–2826
7. P. Barré, B.C. Stöver, K.F. Müller, V. Steinhage, LeafNet: a computer vision system for automatic plant species identification. Ecological Inform. **40**, 50–56 (2017)
8. M.M. Ghazi, B. Yanikoglu, E. Aptoula, Plant identification using deep neural networks via optimization of transfer learning parameters. Neurocomputing **235**, 228–235 (2017)
9. L.C. Uzal, G.L. Grinblat, R. Namías, M.G. Larese, J.S. Bianchi, E.N. Morandi, P.M. Granitto, Seed-per-pod estimation for plant breeding using deep learning. Comput. Electron. Agric. **150**, 196–204 (2018)
10. N. Sünderhauf, C. McCool, B. Upcroft, T. Perez, Fine-grained plant classification using convolutional neural networks for feature extraction, in CLEF (2014), pp. 756–762
11. J.S. Cope, D. Corney, J.Y. Clark, P. Remagnino, P. Wilkin, Plant species identification using digital morphometrics: a review. Expert Syst. Appl. **39**(8), 7562–7573 (2012)
12. N. Kumar, P.N. Belhumeur, A. Biswas, D.W. Jacobs, W.J. Kress, I.C. Lopez, J.V. Soares, Leafsnap: a computer vision system for automatic plant species identification, in Computer vision–ECCV 2012 (2012), pp. 502–516
13. Y. Sun, Y. Liu, G. Wang, H. Wang, Deep learning for plant identification in natural environment. Comput. Intell. Neurosci. (2017)
14. İ. Çuğu, E. Şener, Ç. Erciyes, B. Balcı, E. Akın, I. Önal, A.O. Akyüz, Treelogy: A Novel Tree Classifier Utilizing Deep and Hand-crafted Representations (2017). arXiv:1701.08291
15. S. Zhang, H. Wang, W. Huang, Z. You, Plant diseased leaf segmentation and recognition by fusion of superpixel, K-means and PHOG. Optik-Int. J. Light Electron Opt. **157**, 866–872 (2018)
16. P. Pawara, E. Okafor, L. Schomaker, M. Wiering, Data augmentation for plant classification, in International Conference on Advanced Concepts for Intelligent Vision Systems (2017, September), pp. 615–626
17. M.B.H. Rhouma, J. Žunić, M.C. Younis, Moment invariants for multi-component shapes with applications to leaf classification. Comput. Electron. Agric. **142**, 326–337 (2017)
18. M.A. Nielsen, Neural networks and deep learning (2015), [Online]. Available http://neuralnetworksanddeeplearning.Com (2016)
19. K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition (2014). arXiv preprint arXiv:1409.1556
20. A. Rosebrock, ImageNet: VGGNet, ResNet, Inception, and Xception with Keras [online]. Available https://www.pyimagesearch.com/2017/03/20/imagenet-vggnet-resnet-inception-xception-keras/. Accessed on 18 Aug 2018
21. Coursera, Overview of modern CNN architectures [online]. Available https://www.coursera.org/lecture/intro-to-deep-learning/overview-of-modern-cnn-architectures-U0qGv. Accessed on 18 Aug 2018

22. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2015), pp. 1–9
23. C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2016), pp. 2818–2826
24. S.H. Lee, C.S. Chan, S.J. Mayo, P. Remagnino, How deep learning extracts and learns leaf features for plant classification. Pattern Recogn. **71**, 1–13 (2017)
25. K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2016), pp. 770–778
26. F. Chollet, Xception: Deep learning with depthwise separable convolutions (2017). arXiv preprint, 1610 – 02357
27. G. Huang, Z. Liu, L. Van Der Maaten, K.Q. Weinberger, Densely connected convolutional networks, in CVPR (2017, July), pp. 1, 3
28. A.G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam, Mobilenets: efficient convolutional neural networks for mobile vision applications (2017). arXiv preprint arXiv:1704.04861
29. M.D. Zeiler, R. Fergus, Visualizing and understanding convolutional networks, in European Conference on Computer Vision (2014, September), 818–833
30. B. Vikani, F. Shah, CNN Visualization, tf_cnnvis, 2017, [online]. Available at https://github.com/InFoCusp/tf_cnnvis. Accessed on 27 Aug 2018

Author Index

M. Tuba et al. (eds.), *ICT Systems and Sustainability*, Advances in Intelligent Systems
and Computing 1270, https://doi.org/10.1007/978-981-15-8289-9