Journal of Artificial Intelligence and Data Science (JAIDA) Vol. 1, No. 1, pp. 45-52, 2021
Research Article
https://jaida.ikcu.edu.tr/

A Flower Status Tracker and Self Irrigation System (FloTIS)

Rumeysa KESKİN, Furkan GÜNEY, M. Erdal ÖZBEK*

İzmir Kâtip Çelebi University, Faculty of Engineering and Architecture,
Department of Electrical and Electronics Engineering, İzmir, Turkey

Abstract
The Internet of Things (IoT) provides solutions to many daily life problems. Smartphones with user-friendly applications make use of artificial intelligence solutions offered by deep learning techniques. In this work, we provide a sustainable solution to automatically monitor and control the irrigation process for detected flowers by combining deep learning and IoT techniques. The proposed flower status tracker and self-irrigation system (FloTIS) is implemented using a cloud-based server and an Android-based application to control the status of the flower which is being monitored by the local sensor devices. The system detects changes in the moisture of the soil and provides necessary irrigation for the flower. In order to optimize the water consumption, different classification algorithms are tested. The performance comparisons of similar works for example flower case denoted higher accuracy scores. Then the best generated deep learning model is deployed into the smartphone application that detects the flower type in order to determine the amount of water required for the daily irrigation for each type of flower. In this way, the system monitors water content in the soil and performs smart utilization of water while acknowledging the user.
Keywords: Automatic irrigation system; deep learning; IoT.

1. Introduction

There are many types of plants that are very susceptible to water, temperature, soil pH, etc., for their good growth health and overall development. One of the factors to be considered essential to the health of the plant is water and each plant grows healthy in different ranges of this factor. The lack of water or excessive water can make plants dry or rotten which results in insufficient growth or death. It is hardly feasible that everyone knows the appropriate information about the growth of plants for their needs. Even with just the correct execution of the irrigation, the factors of time, water, and a lot of money can be saved [1]. Therefore, the automatic irrigation system has become very convenient due to insufficient water resources and time requirements.

The efficiency of the systems can be increased through the use of machines where every object or “thing” can be connected by the Internet of Things (IoT) concept. The IoT-based network enables the communication between machine-to-machine (M2M), that is the network of the embedded sensors, actuators, and electronic devices for the purpose of collecting and exchanging data with other devices and systems over the Internet [2]. There are several related pieces of research that have been discussed regarding automatic plant irrigation systems developed and implemented with IoT technology. For example, an IoT-based irrigation system uses a regression algorithm to optimize and control the process of water consumption [3]. Another research has implemented a Web-based plant monitoring application that helps users to monitor the plant status of temperature and humidity measurements with a wireless sensor network by sending a short message (i.e., SMS) to the user's smartphone of any unexpected changes in measurements [4].

On the other hand, any irrigation system requires the identification of the plant under investigation. Recent advances in machine learning technology help to develop automatic identification systems by the acquisition of millions of digital plant photos with smartphones, leading to mobile-based automatic plant identification applications. There have been many approaches developed to identify different flower species considering the main features of flowers such as color, texture, and shape. The support vector machine (SVM) has been widely used as the most effective machine learning-based classifier in image classification [5]. However, hand-crafted traditional discriminative features such as histogram of oriented gradients [6], scale-invariant feature transform [7] used in the SVM classification method cannot be easily applied to numerous flower classes. Moreover, the classification method applied to a single flower dataset is not guaranteed to achieve the same performance on a different flower dataset due to the complexity of the problem. Besides, the extracted features might not be generalizable to other flower images with changing conditions such as rotation, scaling, and illumination.

The deep learning methods, especially convolutional neural network (CNN) architectures have recently gained preference on image classification due to their superior performance in accuracy compared to the classical machine learning methods which rely on mostly hand-crafted methods. The CNN takes the intensities of pixels

*Corresponding author e-mail address: merdal.ozbek@ikcu.edu.tr

Received: 16/07/2021; Accepted: 25/08/2021

from an input image and gradually adjusts the parameters (model weights) during training until optimizing the algorithm in order to neural network make a prediction as closely as possible. The detection of a flower from its image depends on the efficient models of the deep learning architectures based on a CNN. There have been studies to classify from various images [8, 9], and various types of plants including flowers [10, 11]. In those works, efficient deep learning architectures like Inception V3 [12], MobileNet [13], and ResNet-50 [14] have been used to obtain higher recognition rates [9, 11, 15]. Moreover, the information of the model obtained from one type of flower can also be transferred to another by transfer learning. Thus the present study uses CNNs along with transfer learning to efficiently recognize flower species in real-time. In order to implement the control and traceability of the system, and display real-time information to the users, the sensors reading the moisture level of the soil are integrated to an Android application on a smartphone.

The remainder of the paper is organized as follows: Section 2 gives the background information and mentions the key concepts. Section 3 explains the system architecture and implementation of the proposed flower recognition-based flower status tracker and self-irrigation system (FloTIS). Section 4 reports the accuracy performance of the models for flower classification and demonstrates the irrigation system integrated with the designed Android smartphone application. Section 5 summarizes and concludes the paper.

2. Background

The proposed system combines the concepts of deep learning to detect the flower type, and IoT to transfer the flower data in order to check the status and control the automatic irrigation of the flower via smartphones.

The deep learning models for image classification utilizes mostly the CNN architectures. A typical CNN is composed of three stages [16]. In the first stage, several convolutions are performed in parallel to produce a set of linear activations. The so-called convolution layer uses filters that perform convolution operations and outputs feature maps. Each linear activation is run through a nonlinear activation function in the following stage. In the third stage, a pooling function is used to modify the output of the layer. The pooling is essentially a down-sampling operation where commonly used values of the maximum or the average are taken. The concatenation of layers provides to work with data that has a grid-structured topology and scales to large sizes. This is followed by a final fully connected layer that outputs the desired classification class scores. This overall approach has been the most successful on two-dimensional cases, i.e., on images.

The CNN model is first introduced to perform recognition of hand-written digits from images and it has shown significant performance improvement compared to earlier state-of-the-art machine learning techniques [17]. Later it has been extended by many studies and many architectures have been proposed with an increased number of layers that can achieve lower error rates [9, 18].

On the other hand, IoT aims to create a better environment for humanity, where objects around us are organized according to our needs and act without explicit instructions. Therefore, the fundamental promise of IoT is to monitor and control the connected things [19]. This is mainly built up by sensors to collect information, identifiers to identify the source of data, software to analyze the data, and most importantly the Internet connectivity to communicate and notify. The explosion of smart and mobile devices further boosted the usability of IoT-based solutions for everyday life problems.

Initially, the dominant technology behind the IoT was Radio-Frequency Identification (RFID). Today with further technological achievements, a diverse set of architectures and enabling technologies drive the standardization of IoT in several application domains. There are several surveys that focus on technologies such as wireless sensor networks, communication protocols, cloud platforms, fog computing [20, 21, 22]. Concepts combined with technology are reflected on the applications name such as smart home, smart city, or smart farming. Particularly for agriculture, sensing for soil moisture and nutrients, controlling water usage for plant growth, and determining custom fertilizer are some simple uses of IoT [23].

A natural solution to using these smart systems is employing smartphones. Simple applications in smartphones aid in monitoring and controlling the objects together with the underlying technologies of IoT. A common operating system in smartphones is Android. The software tool generating those applications helps to handle the high computation for deep learning classification using the cloud services, necessary communication protocols to access and control the objects as well as notify users at any step of the usage.

3. System Architecture and Implementation

This section describes the theoretical foundation and system implementation of our proposed system in detail.

3.1. Flower Recognition System

In order to classify the type of flower and determine the required water based on the soil moisture level, the first step is image classification. Image classification is the pioneering work that boosted the deep learning architectures based on CNNs after AlexNet [24]. Today, most of the deep learning architectures utilize the pre-trained models on large-scale datasets such as ImageNet [25] which contains 1.4 million images over 1000 object classes for training their models using a technique called transfer learning. The transfer learning method keeps the parameters of the CNN model's previous layer and removes the last layer, then retrains the last layer for new categories. This helps to train networks for classification with a small number of training samples like the 20-category flower dataset that we used for this study.

In order to determine the most effective model for flower classification, experiments were carried out with three different CNN algorithms.

MobileNet V1: MobileNets are mobile-first computer vision models designed by Google researchers in order to run onto smartphones with limited memory. MobileNets uses depthwise separable convolution architecture that significantly reduces the parameters when compared to networks with the same depth in the networks. This lightweight network structure provides an efficient model for mobile and low-power embedded vision applications [26]. MobileNet V1 network structure has 28 layers and requires only one-eighth of the computation cost. This architecture, as seen in Figure 1, has an image input size of 224-by-224 with 3 channels.

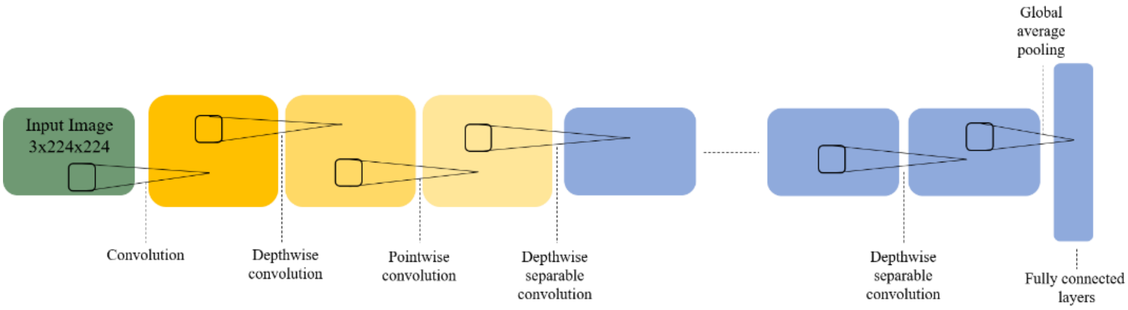

Figure 1. *MobileNet V1 architecture.*

Resnet-50: The ResNet-50 (Residual Network-50) is the first deep-CNN (>50 layers) architecture, as seen in Figure 2, with 50-layer deep that utilized residual learning [27] to manage more complex image recognition tasks and improve the accuracy of the model. In deep networks, with the increasing layers, accuracy gets saturated and then degrades rapidly. Residual learning has been introduced to address this degradation problem. The residual networks are made up of residual blocks that comprise a skip connection to make it easier to learn the identity function and help to solve the problem of degradation of training accuracy. Residual networks are the deepest ever presented on ImageNet and still have lower complexity than VGG networks [28]. This network structure has an image input size of 3-by-224-by-224.

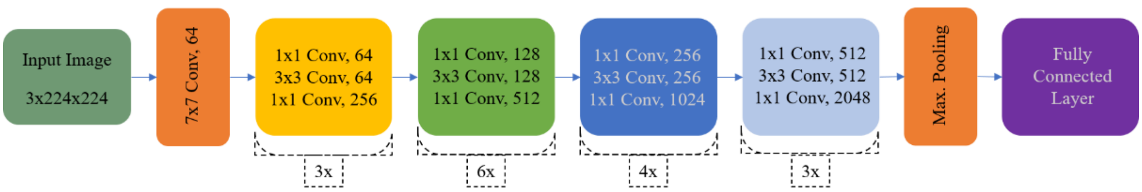

Figure 2. *Resnet-50 architecture.*

Inception V3: Inception V3 is a CNN structure developed by Keras [29], which is pre-trained in ImageNet. The Inception V3 network consists of 48-layers of convolutional, pooling, and fully connected layers as shown in Figure 3 with an image input size 3-by-299-by-299. Compared to the previous versions (Inception V1 [30] and V2 [31]), the Inception V3 network structure uses a convolution kernel splitting method [32] to achieve different scale perceptions. Through the splitting method, it does not only reduce the number of network parameters and computational costs without hampering accuracy but also increases the network depth and extracts the spatial features more effectively.

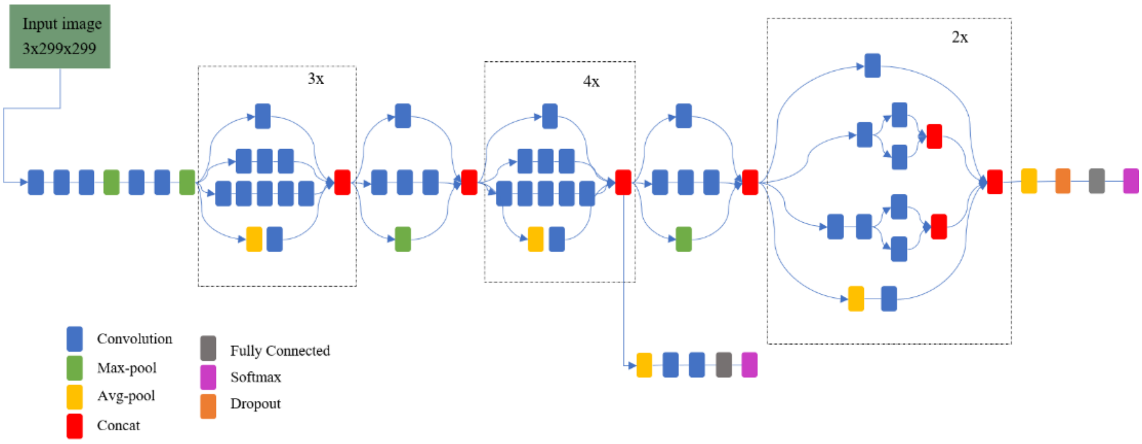

Figure 3. *Inception V3 architecture.*

3.2. Implementation

The system utilizes two microcontrollers for electronic device monitoring, data collection, and data saving; Arduino UNO [33] and Raspberry Pi [34]. The irrigation system is composed of a vertical submersible low-power water pump to supply water and a resistive soil moisture sensor to detect the humidity level of the soil. The sensor sends the moisture level of the soil to the Raspberry Pi through the Arduino UNO with serial communication. Then, the received value is compared with the predefined soil moisture range of the flower given in Table 1 determined from [35], for every 12 hours. The flower water status is then forwarded to the smartphone application through the Firebase [36] server. If the moisture level is below the desired range, the automated irrigation system provides the required measure of water for 2 seconds till it reaches the determined water content range.

Table 1. *Soil moisture levels of flowers in our dataset.*

Flower		**0% - 20%** **Dry soil**	**21% - 40%** **Drained soil**	**41% - 60%** **Moist soil**	**61% - 80%** **Wet soil**
Azalea			✔		
Bluebell			✔		
Buttercup					✔
Cactus	✔	✔			
Coltsfoot				✔	
Cowslip				✔	
Crocus	✔	✔			
Daffodil				✔	
Daisy				✔	
Dandelion			✔		
Fritillary			✔		
Geranium				✔	
Iris			✔		
Lily Valley			✔		
Pansy				✔	
Snowdrop			✔		
Sunflower			✔		
Tigerlily					
Tulip			✔		
Windflower				✔	

The overall proposed smart flower status tracker and self-irrigation system is depicted as a block diagram and is presented in Figure 4.

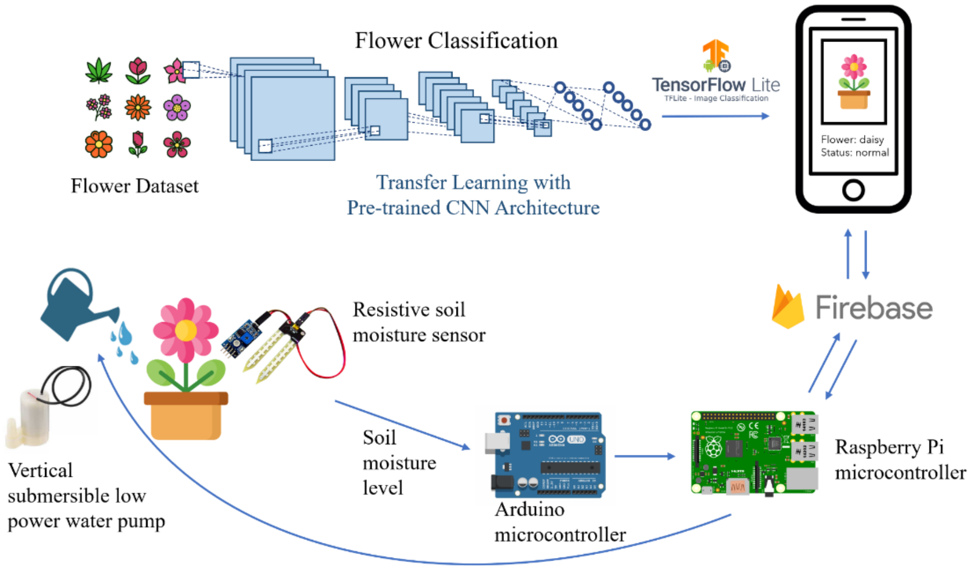

Figure 4. *Schematic of the flower status tracker and self-irrigation system on smartphones.*

4. Results

In this study, we used the Oxford-17 Flower Dataset [37] with each species has 80 flower images. Three more types of flower images (80 images per type) were also added to obtain 20 species. This new dataset is split into two parts where 70% are used as a training set, 30% are used as a validation set. For the classification of images, the three aforementioned models were selected as MobileNet V1, ResNet-50, and Inception V3. The optimizer for training these models is the Stochastic Gradient Descent (SGD) [38] optimizer with a selected batch size of 32 and a learning rate of 0.001. The training of all the models is completed in 50 epochs where the validation accuracy has at the peak level. Then the model is exported to TensorFlow-Lite [39] model and is deployed into the application to perform the classification on Android-based smartphone devices and manage to detect flowers.

The training and validation accuracy of the models with the transfer learning technique is presented in Table 2. The best validation accuracy is found with the Inception V3 model; therefore, this model is deployed to the Android application.

Table 2. *Performance evaluation scores.*

Model	Train Accuracy	Validation Accuracy
MobileNet V1	0.9758	0.8973
ResNet-50	0.9880	0.9010
Inception V3	**0.9955**	**0.9312**

Table 3. *Performance comparison of our proposed model and other state-of-the-art models on Oxford-17 Flower Dataset.*

Model	Accuracy
VGG 16-transfer [40]	0.8353
VGG 19-transfer [40]	0.8471
VA ResNet-50 [41]	0.8570
Fusion Descriptor and SVM [7]	0.8617
Our proposed model	**0. 9312**

A comparison of the different models on the Oxford-17 Flower Dataset is also performed. Based on the accuracy values presented in Table 3, our proposed Inception V3 model has a significant improvement as compared to other methods.

Figure 5 presents the setup prototype of FloTIS. The system is tested on the geranium. The flower name is identified by taking a photo with the camera of the smartphone from the application menu. The photo is predicted as "geranium" and displayed by the application.

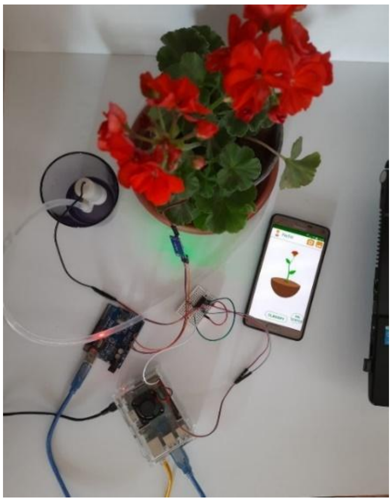

Figure 5. *Setup for the prototype of FloTIS.*

The flower name information is then sent to the controller through the server. As the soil moisture levels have been stored for each flower, the requirement for geranium soil moisture is moist soil and the level should be between 41% and 60% as given in Table 1. In this work, three categories are selected to notify the user and then to supply water to the flower. These are determined as "need water!", "normal", "excessive water!". For example, when the measured value is 23%, the water is insufficient, then the notification is "need water!". When the flower is watered and the moisture level is increased to 56%, it is notified as "normal". When the moisture level is measured as 65% that exceeds the upper limit for the geranium, the notification becomes "excessive water!". The corresponding statuses of the water displayed on the smartphone application are shown in Figure 6.

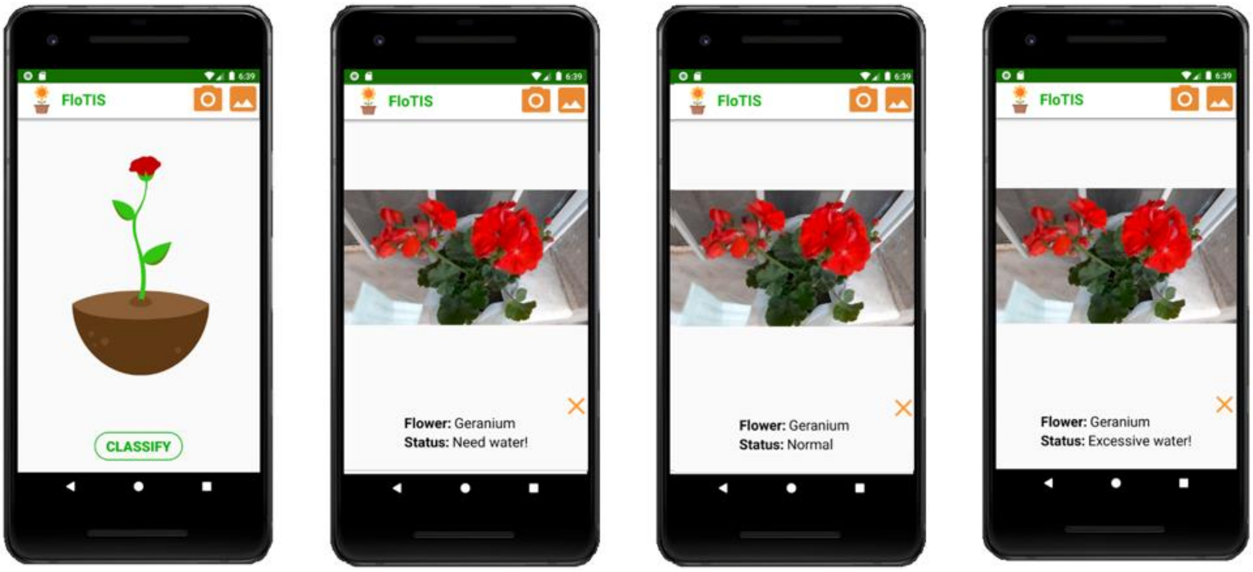

Figure 6. *Water statuses of the recognized flower by the smartphone application.*

5. Conclusion

In this study, an automatic self-irrigation and flower status tracker system is proposed. It combines deep learning architectures to identify the flower, a detection and irrigation system to sense and then provide the plant with a certain level of water and hence prevent overwatering, into a smartphone application named FloTIS integrating all systems into an IoT framework.

The advantage of the system lies in detecting the flower type using image recognition and then providing the correct execution of watering for different flower types. The user-friendly FloTIS application displays corresponding water levels of the flower while it controls and tracks the irrigation system managing the desired levels accordingly.

As the flower image dataset used in this work is limited, only three deep learning architectures were selected for flower identification. A comparison of the other classification algorithms denoted the efficiency of the proposed method in terms of the accuracy scores. Note also that those were pre-trained algorithms and they can be used easily in a higher number of dataset images wherever available. Therefore, the proposed system can be implemented on a large scale for farming, especially for water conservation due to prevailing conditions and water shortages.

Declaration of Interest

The authors declare that there is no conflict of interest.

References

[1] A. Patil, M. Beldar, A. Naik, and S. Deshpande, "Smart farming using Arduino and data mining," In 3rd International Conference on Computing for Sustainable Global Development (INDIACom), 2016, pp. 1913-1917.
[2] R. Ratasuk, B. Vejlgaard, N. Mangalvedhe, and A. Ghosh, "NB-IoT system for M2M communication," In Proc. IEEE Wireless Communications and Networking Conference, 2016, pp. 428-432.
[3] A. Kumar, A. Surendra, H. Mohan, K. M. Valliappan, and N. Kirthika, "Internet of things based smart irrigation using regression algorithm," In Proc. International Conference on Intelligent Computing, Instrumentation and Control Technologies, 2017, pp. 1652-1657.
[4] M. Mancuso and F. Bustaffa, "A wireless sensors network for monitoring environmental variables in a tomato greenhouse," In IEEE International Workshop on Factory Communication Systems, 2006, pp. 107-110.
[5] H. H. Lee, X. H. Li, K. W. Chung, and K. S. Hong. "Flower image recognition using multi-class SVM," Applied Mechanics and Materials, vol. 284-287, pp. 3106-3110, 2013.
[6] W. Zhou, S. Gao, L. Zhang, and X. Lou. "Histogram of oriented gradients feature extraction from raw Bayer pattern images," IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 67, no. 5, pp. 946-950, 2020.
[7] W. Liu, Y. Rao, B. Fan, J. Song, and Q. Wang. "Flower classification using fusion descriptor and SVM," In Proc. IEEE International Smart Cities Conference (ISC2), 2017, pp. 1-4.
[8] M. Hussain, J. J. Bird, and D. R. Faria, "A study on CNN transfer learning for image classification," In UK Workshop on Computational Intelligence, 2018, pp. 191-202.
[9] N. Sharma, V. Jain, and A. Mishra, "An analysis of convolutional neural networks for image classification," Procedia Computer Science, vol. 132, pp. 377-384, 2018.
[10] I. Gogul and V. S. Kumar, "Flower species recognition system using convolution neural networks and transfer learning," In 4th International Conference on Signal Processing, Communication and Networking, 2017, pp. 1-6.
[11] X. Xia, C. Xu, and B. Nan, "Inception-v3 for flower classification," In 2nd International Conference on Image, Vision and Computing, 2017, pp. 783-787.
[12] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," In Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2818-2826.
[13] D. Sinha and M. El-Sharkawy. "Thin mobilenet: An enhanced mobilenet architecture," In IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference, 2019, pp. 280-285.
[14] K. He, X. Zhang, S. Ren, J. Sun, "Deep residual learning for image recognition", In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770-778.
[15] W. Wang, Y. Li, T. Zou, X. Wang, J. You, and Y. Lou, "A novel image classification approach via dense-MobileNet models," Mobile Information Systems, vol. 2020, Article ID 7602384, 8 pages, 2020.
[16] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning, MIT Press, Cambridge, 2016.
[17] Y. LeCun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, and L. Jackel, "Handwritten digit recognition with a backpropagation network," in: Advances in Neural Information Processing Systems (NIPS), vol. 2, 1989, pp. 396-404.
[18] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie, and L. Farhan, "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," Journal of Big Data, vol. 8, no. 53, pp. 1-74, 2021.
[19] A. Rayes and S. Salam, Internet of Things From Hype to Reality: The Road to Digitization, 2nd, Springer, 2019.
[20] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: A survey on enabling technologies, protocols, and applications," IEEE Communications Surveys & Tutorials, vol. 17, no. 4, pp. 2347-2376, Fourthquarter 2015.
[21] I. U. Din et al., "The Internet of things: A review of enabled technologies and future challenges," IEEE Access, vol. 7, pp. 7606-7640, 2019.

[22] A. Khanna and S. Kaur, "Internet of things (IoT), applications and challenges: A comprehensive review," Wireless Personal Communications, vol. 114, pp. 1687-1762, 2020.
[23] H. F. Atlam, R. J. Walters, and G. B. Wills, "Internet of things: State-of-the-art, challenges, applications, and open issues," International Journal of Intelligent Computing Research (IJICR), vol. 9, no. 3, pp. 928-938, 2018.
[24] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in: Advances in Neural Information Processing Systems (NIPS), vol. 25, 2012, pp. 1097-1105.
[25] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "Imagenet large scale visual recognition challenge," 2015, pp. 211-252.
[26] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications,", arXiv:1704.04861, 2017.
[27] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," In Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770-778.
[28] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition,", arXiv:1409.1556, 2014.
[29] A. Gulli and S. Pal, Deep Learning with Keras, Packt Publishing, 2017.
[30] S. M. Sam, K. Kamardin, N. N. A. Sjarif, and N. Mohamed, "Offline signature verification using deep learning convolutional neural network (CNN) architectures GoogLeNet Inception-v1 and Inception-v3," Procedia Computer Science, vol. 161, pp. 475-483, 2019.
[31] S. R. Bose and V. S. Kumar, "Efficient inception V2 based deep convolutional neural network for real-time hand action recognition," IET Image Processing, vol. 14, pp. 688-696, 2019.
[32] M. Kim and L. Rigazio, "Deep clustered convolutional kernels," In Proceedings of the 1st International Workshop on Feature Extraction: Modern Questions and Challenges at NIPS 2015, PMLR vol. 44, pp. 160-172, 2015.
[33] Y. A. Badamasi, "The working principle of an Arduino," In 11th International Conference on Electronics, Computer and Computation (ICECCO), 2014, pp. 1-4.
[34] N. S. Yamanoor, and S. Yamanoor, "High quality, low cost education with the Raspberry Pi," In IEEE Global Humanitarian Technology Conference (GHTC), 2017, pp. 1-5.
[35] [Online]. Available: https://www.groww.fr/en/plants.
[36] L. Moroney, "The firebase realtime database," in The Definitive Guide to Firebase, Apress, 2017, pp. 51-71.
[37] M.-E. Nilsback and A. Zisserman, "A visual vocabulary for flower classification," In Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2006, pp. 1447-1454.
[38] L. Bottou, "Stochastic gradient descent tricks," In: Montavon G., Orr G. B., Müller K. R. (eds) Neural Networks: Tricks of the Trade. Lecture Notes in Computer Science, vol. 7700. Springer, Berlin, Heidelberg, 2012, pp. 421-436.
[39] O. Alsing, "Mobile object detection using tensorflow lite and transfer learning," Master thesis, KTH Royal Institute of Technology, 2018.
[40] Y. Wu, X. Qin, Y. Pan, and C. Yuan, "Convolution neural network based transfer learning for classification of flowers," In Proc. IEEE 3rd International Conference on Signal and Image Processing, 2018, pp. 562-566.
[41] S. Cao and B. Song, "Visual attentional-driven deep learning method for flower recognition," Mathematical Biosciences and Engineering, vol. 18, no. 3, pp. 1981-1991, 2021.