Materials Research Proceedings 43 (2024) 205-213 https://doi.org/10.21741/9781644903216-27

Date fruit type classification using convolutional neural networks

Abdullah ALAVI[1,a], Md Faysal AHAMED[1,b], Ali ALBELADI[1,c], Mohamed MOHANDES[1,d] *

[1]King Fahd University of Petroleum and Minerals, Department of Electrical Engineering, Dhahran, Saudi Arabia

[a]g202313690@kfupm.edu.sa, [b]g202309930@kfupm.edu.sa, [c]albeladi@kfupm.edu.sa, [d]mohandes@kfupm.edu.sa

Keywords: Date Fruit Type Classification, Convolutional Neural Network, Squeezenet, Pretrained Network, Transfer Learning

Abstract. Classification of objects is an important task for convolutional neural networks (CNNs). They have been applied to numerous fields with excellent results. In this study, we use CNNs to classify five categories of Sukkari dates, namely Galaxy, Mufattal, Nagad, Qishr, and Ruttab. Transfer learning is when a pretrained model is taken and only the final layers are trained to make a prediction. In this paper, we used the following five models: SqueezeNet, GoogLeNet, EfficientNet-b0, ShuffleNet, and MobileNet V2. The results show that SqueezeNet outperforms the other networks with a classification accuracy of 92% on the testing set. The testing accuracy for GoogLeNet, EfficientNet-b0, ShuffleNet, and MobileNet V2, on the other hand are 85.14%, 82.86%, 89.14%, and 87.43%, respectively. As this is a classification task, other metrics like precision, recall, and F1 score are also evaluated. These values for the SqueezeNet on the testing set are 92.67%, 92%, and 92.33%, respectively. ShuffleNet was second with values of 89.41%, 89.14%, and 89.28%, respectively. EfficientNet scored the lowest with 83.10%, 82.86%, and 82.98%, respectively.

Introduction

The recent breakthroughs in computer vision and artificial intelligence (AI) have led to myriad applications, ranging from facial recognition to self-driving cars. All such applications have a common theme, that it is relatively easy for people to solve with good accuracy but nearly impossible to program and implement on a computer [1]. To solve this problem, AI-based systems need to possess the ability to extract the patterns from the raw data and produce an output based on this knowledge [1, 2, 3, 4]. Agriculture is one of the areas that fall in this category and has benefitted from the developments of computer vision. Applications include machines to sort fruits, automatic fruit harvesting, and fruit scanners in markets [5, 6].

Deep Learning (DL) is a subset of AI that has gained significant popularity in recent years. It has a high level of abstraction and can automatically learn patterns from images [7]. Convolutional Neural Network (CNN) [8] is a popular architecture for applications that involve image processing [1, 4, 9, 10]. CNNs use a convolution operation in at least one of the layers [1, 11]. CNNs have started to gain popularity after 2012, when Krizhevsky et al. [12] won the ILSVRC competition on ImageNet [13]. Since then, they have found various applications in computer vision including fruit classification and detection [14, 15, 16, 17, 18].

Dates fruits are popular in the Middle East, North Africa, and Southwest Asia [19]. In the Kingdom of Saudi Arabia (KSA), date palm trees occupy nearly a quarter of the total cultivated land [20]. Various types of dates are cultivated and they vary greatly in terms of their size, color, and taste [21]. The recent success of AI and DL on inspecting a variety of fruits has inspired a number of works pertaining to date fruits. Date fruit quality classification, for instance, has gained traction among several researchers [22, 23]. Alresheedi et al. [24] compared the accuracy of several machine learning models with CNN. The dataset consisted of nine classes of date fruits. The authors found that the CNN model boasted the highest accuracy. In [25], a framework is proposed for date recognition. 500 images of three types of date fruits were used. The framework was based on a deep CNN and achieved an accuracy of 89.2%. In [20], Faisal et al. proposed a solution consisting of three estimation functions to classify date fruits based on maturity, type, and mass. The work used a Support Vector Machine (SVM) and achieved an accuracy of 99% among all the estimation functions. In [26], the authors focused on sorting dates based on their health and maturity. Four date types in different maturity stages along with defective dates were used in a CNN model based on VGG-16. The accuracy reported was 97%. In [27], Perez et al. used Medjool dates to compare the performance of eight different CNNs. The target was to sort the dates based on maturity stage. Out of all models, VGG-19 performed the best with an accuracy of 99.32%. In [26], the authors used transfer learning with fine-tuning using two pretrained networks, namely AlexNet and VGGNet. They used a dataset of 8,000 images separated into 5 classes and achieved an accuracy of 97.25%. In [28], the authors implemented a machine vision framework to deploy in a harvesting robot. The framework used three models to classify according to the type, maturity, and harvesting decision. The models are based on pretrained AlexNet and VGG-16. The VGG-16 model achieved accuracy of 99.01%, 97.25%, and 98.59% on date type, maturity, and harvesting decision classification, respectively. In [29], the authors used a dataset of 1,658 images belonging to nine categories of dates. The model used MobileNet V2 and resulted in an accuracy of 96%.

Although, numerous works have been published using individual pretrained models, there has been very little regarding the comparison of the performances of these various pretrained state-of-the-art models. Hardware and datasets for training and testing are different in published works, which make it unfair to compare performances. In this work, we present a comparison between various pretrained CNN models and observe how they perform on the same dataset. The goal of the CNN models is to classify five subclasses of dates of the same family. The paper is organized as follows. Section II describes the CNN models used. Section III explains the explains the dataset and the methodology. Section IV discusses the results. Finally, the conclusion is presented in Section V.

Overview of the CNN Models

A number of popular state-of-the-art pretrained networks like VGG-16, ResNet, Inception, and AlexNet are implemented frequently. However, due to their great number of layers, they usually require a long time to train even on powerful hardware. To mitigate this issue, researchers are looking for ways to minimize the size and training time by restructuring the CNN in various ways while maintain a comparable accuracy. In this paper, we implemented five such models and compared their performance in this classification task. The models are SqueezeNet [30], GoogLeNet [31], EfficientNet-b0 [32], ShuffleNet [33], and MobileNet V2 [34]. A brief summary is provided below about each model and the techniques they used to improve the efficiency.

In SqueezeNet, the authors achieved accuracy comparable to that of AlexNet [12] while using $1/50^{th}$ of its parameters. They also reduced the size to less than 0.5 MB, which is 510 times smaller than AlexNet. They achieved this by replacing the 3x3 filters with 1x1 filters. They also decreased the number of input channels and down sampling the later layers in the network. These strategies were then incorporated with other modifications and packed in what is known as a *Fire module.*

In GoogLeNet, the authors managed to increase the depth and width of the network without increasing the computational demand. They have achieved this by implementing the *Inception module*. This module applies 1x1, 3x3, 5x5 parallel convolutions along with dimensionality reduction. This helped in capturing details of varying sizes.

In EfficientNet, the primary motivation was how to scale up a CNN. Generally, CNNs are developed with certain computational constraint in mind. If the model performs satisfactorily, then it is scaled up to further increase the accuracy. Scaling up can be done by increasing the depth, width or resolution. Instead of scaling up arbitrarily, the authors proposed a relationship between the three parameters known as *compound coefficient.* Using this technique, they have managed to reduce the size and increase the speed when compared to existing state-of-the-art CNNs.

In ShuffleNet, the primary motivation was to design a CNN that can run on mobile devices with extremely limited hardware. They used two new operations, namely pointwise group convolutions and channel shuffling. Using this, the ShuffleNet architecture achieved a superior performance, outperforming MobileNet in terms of accuracy in the ImageNet top-1 error on a computational power of 40 MFLOPS. On ARM-based computing hardware, ShuffleNet achieved a 13x speedup over AlexNet while maintaining a similar level of accuracy.

In MobileNetV2, the authors implemented what is known as an inverted residual structure. In this structure, the shortcut connections are between the thin bottleneck layers. Then, in the intermediate expansion layers, lightweight depth wise convolutions are performed to filter features in order to introduce non-linearity. Also, in the narrow layers, they discovered that it is important to remove non-linearities. This helped in maintaining representational power. On the ImageNet dataset, the MobileNet architecture improved the state-of-the-art on various performance measures in addition to reducing the model complexity.

Dataset and Methodology

The dataset used in this paper came from a real date palm plantation. It consists of images of dates that belong to five subcategories of Sukkari dates. They are known as Galaxy, Ruttab, Mufattal, Qishr, and Nagad. The original images were cropped to 500 by 500 pixels with RGB channels of size 8 bits each with white background. The images were then down sampled to 250 by 250 to reduce the size. Fig. 1 illustrates the five classes of images. As can be observed, the subclasses bear similar resemblance to each other. However, upon closer inspection, it can be seen that they have slightly different texture.

The dataset contains a total of 1,689 images that include the five classes of dates mentioned above. The images were not equally divided between the five classes, rather each class had roughly 400-440 images. For testing, a total of 175 images were set aside from the 1,689 images, and each class contained 35 images. The remaining images were used for training and validation with a split of 80% for training and 20% for validation, respectively. After splitting the dataset, the five CNNs mentioned in the previous section are loaded and trained. Fig. 2 illustrates the flowchart of the overall process of the collection of the images, cropping and down sampling, and the training phase of the CNNs.

The pretrained CNNs were originally trained on the ImageNet dataset [35]. For this study, only the final convolutional, classification and softmax layer were modified to produce the outputs of 5 classes. Also, the learning rate of the final layers were increased. This is discussed further in the next section. Apart from that, no other parameters of the CNNs were modified and the default values were used for simulation. The size of the input layers and the subsequent layers and their connections were also kept the same. During the training, validation, and testing phase, the images were automatically resized to fit the size of the default input layer of the pretrained networks.

Materials Research Proceedings 43 (2024) 205-213 https://doi.org/10.21741/9781644903216-27

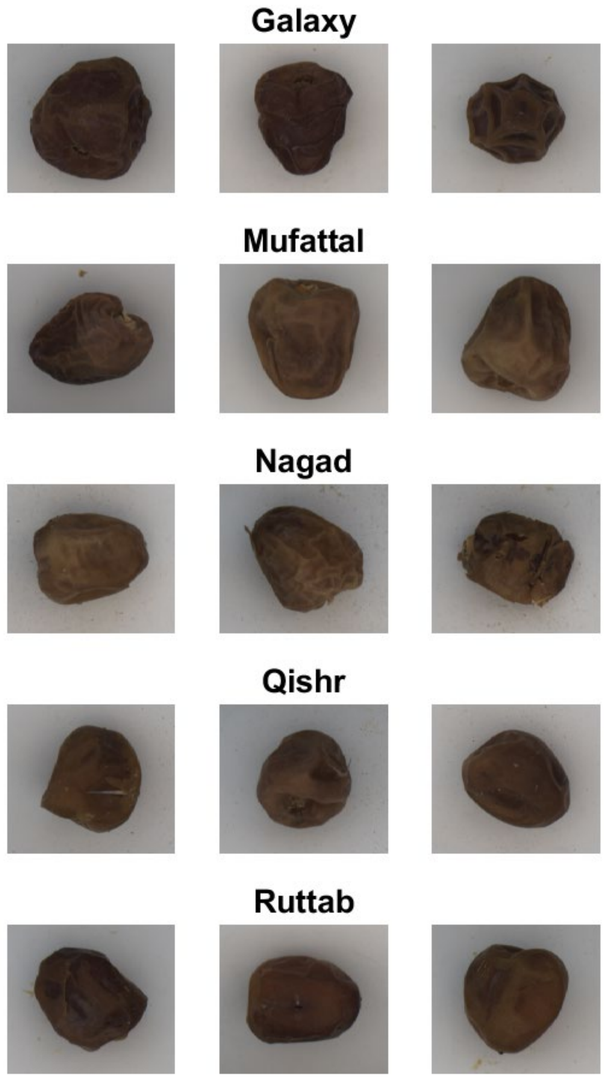

Fig. 1. Five classes of dates.

Materials Research Forum LLC
https://doi.org/10.21741/9781644903216-27

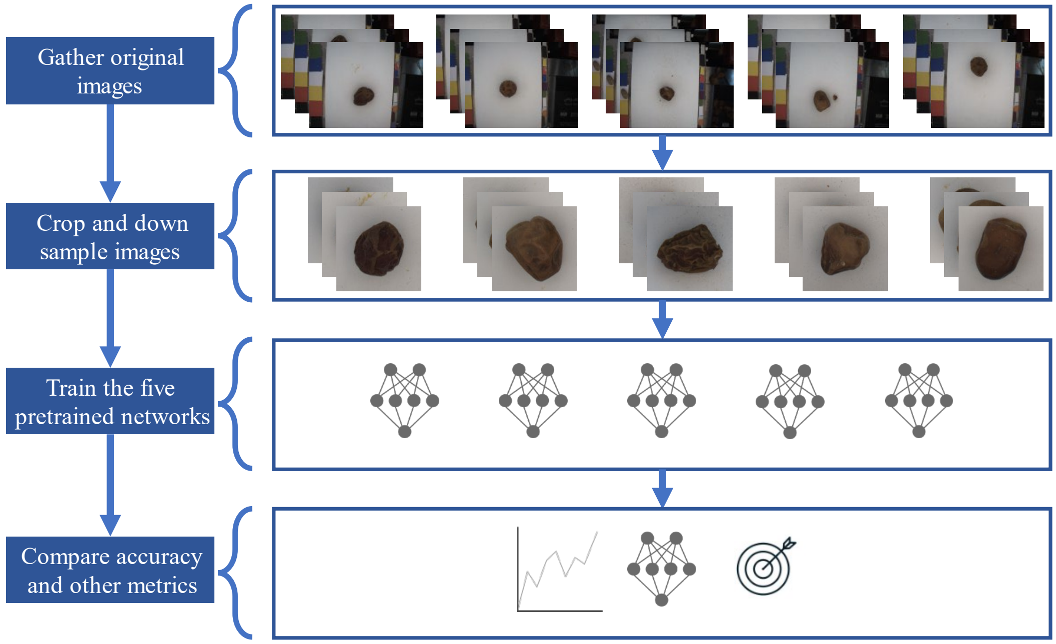

Fig. 2. Flowchart of the overall simulation.

Simulation Results

The results of the simulation provide an overview of the performances of the various networks. MATLAB 2023b along with the built-in Deep Learning Toolbox were used to train, validate, and test the networks. The networks were trained on a Lenovo IdeaPad Gaming 3 laptop. The networks were specifically trained on the GPU only. The GPU on the laptop is Nvidia GeForce RTX 4050 (6 GB). The CPU is an AMD Ryzen 7-7735HS. The RAM is 16 GB. As the networks are pretrained, the training does not need to be as extensive because the previous layers have already learned the detailed features. The epochs have been set to 10 with a batch size of 32 images. The validation frequency has been set to every two iterations. Cross-entropy loss was used with a train-test-validation split of 72-18-10, respectively. The learning rate for all layers except the last layer has been set to 0.0001. For the final layer, it has been set to 10 for both the weights and biases. For augmenting the images, random X and Y-reflections were applied. Also, random X and Y-translation has been applied varying from -30 to 30 pixels. Table 1 summarizes the parameters used for the training of the network.

Table 1. Training parameters for the CNNs.

Training Parameters	**Setting**
Epochs	10
Mini Batch Size	32
Total Iterations	480
Validation Frequency (Iterations)	2
Train-Val-Test Split	72-18-10
Loss Function	Cross-entropy
Overall Learning Rate	0.0001
Final Layer Weight and Bias Learning Rate	10
Image Augmentation	X-reflection
	Y-reflection
	X-translation (30 to -30 pixels)
	Y-translation (30 to -30 pixels)

Table 2 summarizes the results of the testing for all the models. The primary metric is the accuracy. However, as the task is classification, other metrics such as precision, recall and F1 score are also of great significance. The training time is also included in the table. The results on the test set indicate that SqueezeNet outperforms the other networks. The overall accuracy of SqueezeNet is 92%, with precision, recall, and F1 score of 92.67%, 92%, and 92.33%, respectively. In the second place was ShuffleNet with an accuracy of 89.14%, with precision, recall, and F1 score of 89.41%, 89.14%, and 89.28%, respectively. EfficientNet recorded the lowest accuracy at 82.86%, with precision, recall, and F1 score of 83.10%, 82.86%, and 82.98%, respectively. In terms of the training time, SqueezeNet was also the quickest, taking 9 minutes and 17 seconds to train. GoogLeNet was the second at 14 minutes and 46 seconds. EfficientNet took the longest with 34 minutes and 17 seconds.

Table 2. Summary of the results of testing and training time.

CNN	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)	Training Time (mm:ss)
SqueezeNet	92	92.67	92	92.33	09:17
ShuffleNet	89.14	89.41	89.14	89.28	15:20
MobileNet	87.43	87.65	87.43	87.54	21:05
GoogLeNet	85.14	85.68	85.14	85.41	14:46
EfficientNet	82.86	83.10	82.86	82.98	34:17

Fig. 3 illustrates the confusion matrix for SqueezeNet in the test set. The vertical axis is the actual class. The horizontal axis is the predicted class. The diagonal elements represent the correct predictions while the off-diagonal elements represent the misclassified predictions. The outer horizontal percentages give the precision for each class, whereas the vertical percentages give the recall for each class.

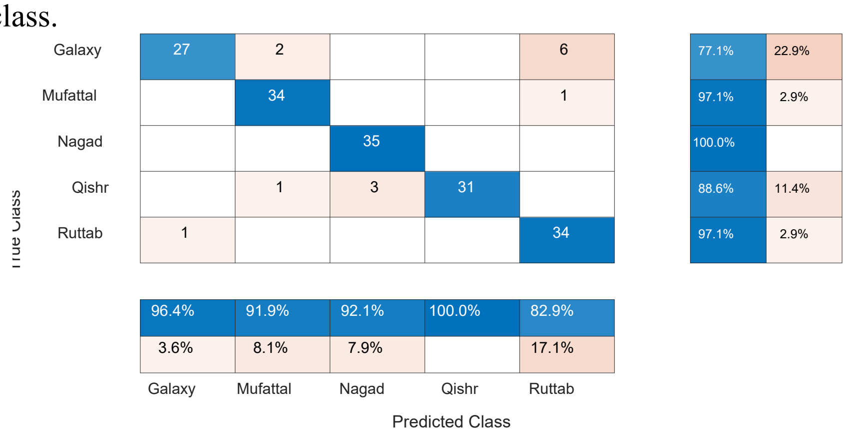

Fig. 3. Confusion matrix for SqueezeNet in the Test set.

Fig. 4 illustrates how the accuracy of the SqueezeNet evolved as the training and validation iterations progress. The number of iterations for the 10 epochs is 480 (for batch size of 32 images). As can be seen, the model converges quickly as it is a pretrained network. Fig. 5 illustrates the loss vs. the iterations. The loss will decrease as the iterations progress. Accuracy and loss are inversely related. Therefore, as accuracy increases, loss decreases.

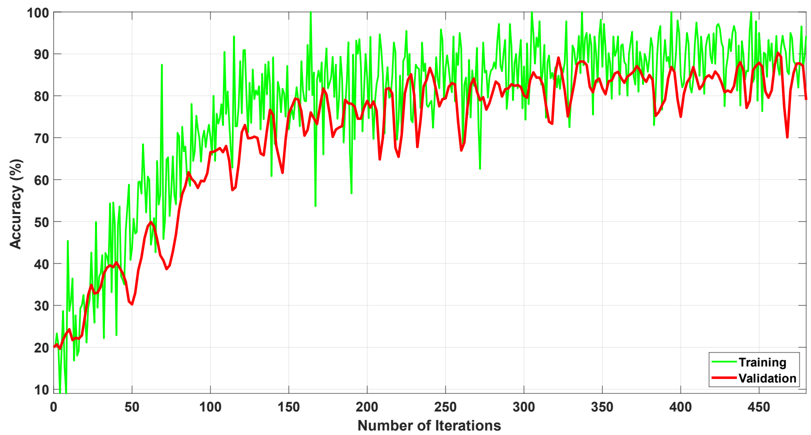

Fig. 4. Accuracy of SqueezeNet vs. the number of iterations.

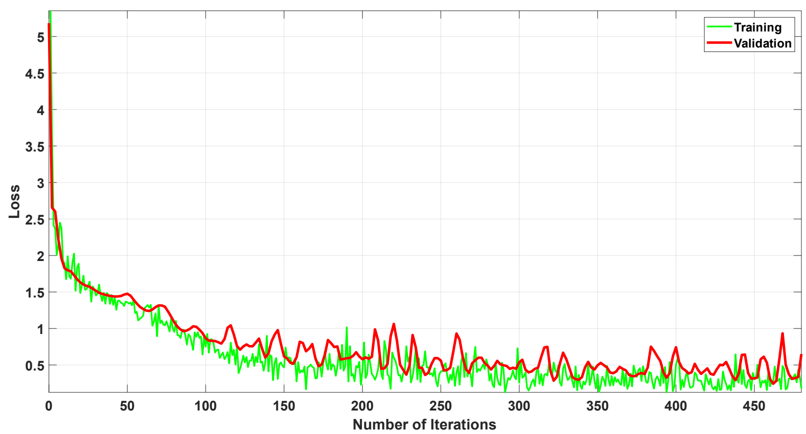

Fig. 5. Loss of SqueezeNet vs. the number of iterations.

Conclusion

Fruit classification is an important area of research for industries as they proceed towards automating the classification process. Various researches have been published in this area using various fruit datasets. However, most studies either include a single model or train a model from a scratch. Comparisons between various papers on different models are usually not homogeneous in nature as they are trained on different datasets and/or different hardware which can significantly affect the results. In this study, we used transfer learning on five popular pretrained CNNs to classify five subcategories of Sukkari dates. We used the same dataset of 1,689 regular RGB images and run the simulation on the same hardware for an even comparison. Comparing the results of the training for the CNNs, it is observed that SqueezeNet performs the best in terms of classification accuracy and the training time. The overall testing accuracy of SqueezeNet is 92% and took 9 minutes and 17 seconds to train on Nvidia GeForce RTX 4050 (6 GB) GPU. The second highest accuracy recorded was the ShuffleNet with a testing accuracy of 89.14%, whereas the second fastest in terms of training time was the GoogLeNet with a training time of 14 minutes and 46 seconds. For the future, we plan to perform further studies on how to improve the accuracy of these models and include other types of date fruits.

References

[1] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. MIT Press, 2016.
[2] B. Coppin, Artificial intelligence illuminated. Jones & Bartlett Learning, 2004.

[3] M. I. Jordan and T. M. Mitchell, “Machine learning: Trends, perspectives, and prospects,” Science (1979), vol. 349, no. 6245, pp. 255–260, 2015. https://doi.org/10.1126/science.aaa8415
[4] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” Nature, vol. 521, no. 7553, pp. 436–444, 2015. https://doi.org/10.1038/nature14539
[5] E. Rachmawati, I. Supriana, and M. L. Khodra, “Toward a new approach in fruit recognition using hybrid RGBD features and fruit hierarchy property,” in 2017 4th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI), 2017, pp. 1–6. https://doi.org/10.1109/EECSI.2017.8239110
[6] Y. Tao and J. Zhou, “Automatic apple recognition based on the fusion of color and 3D feature for robotic fruit picking,” Comput Electron Agric, vol. 142, pp. 388–396, 2017. https://doi.org/10.1016/j.compag.2017.09.019
[7] J. Naranjo-Torres, M. Mora, R. Hernández-Garcia, R. J. Barrientos, C. Fredes, and A. Valenzuela, “A review of convolutional neural network applied to fruit image processing,” Applied Sciences, vol. 10, no. 10, p. 3443, 2020. https://doi.org/10.3390/app10103443
[8] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” Proceedings of the IEEE, vol. 86, no. 11, pp. 2278–2324, 1998. https://doi.org/10.1109/5.726791
[9] Y. Guo and others, “Deep learning for visual understanding.” Neurocomput, 2017.
[10] M. D. Zeiler and R. Fergus, “Visualizing and understanding convolutional networks,” in Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13, 2014, pp. 818–833. https://doi.org/10.1007/978-3-319-10590-1_53
[11] R. Singh and S. Balasundaram, “Application of extreme learning machine method for time series analysis,” International Journal of Computer and Information Engineering, vol. 1, no. 11, pp. 3407–3413, 2007.
[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” Adv Neural Inf Process Syst, vol. 25, 2012.
[13] O. Russakovsky et al., “Imagenet large scale visual recognition challenge,” Int J Comput Vis, vol. 115, pp. 211–252, 2015. https://doi.org/10.1007/s11263-015-0816-y
[14] Y. Lu, “Food image recognition by using convolutional neural networks (cnns),” arXiv preprint arXiv:1612.00983, 2016.
[15] Y.-D. Zhang et al., “Image based fruit category classification by 13-layer deep convolutional neural network and data augmentation,” Multimed Tools Appl, vol. 78, pp. 3613–3632, 2019. https://doi.org/10.1007/s11042-017-5243-3
[16] J. Steinbrener, K. Posch, and R. Leitner, “Hyperspectral fruit and vegetable classification using convolutional neural networks,” Comput Electron Agric, vol. 162, pp. 364–372, 2019. https://doi.org/10.1016/j.compag.2019.04.019
[17] S. W. Chen et al., “Counting apples and oranges with deep learning: A data-driven approach,” IEEE Robot Autom Lett, vol. 2, no. 2, pp. 781–788, 2017. https://doi.org/10.1109/LRA.2017.2651944
[18] S. Bargoti and J. Underwood, “Deep fruit detection in orchards,” in 2017 IEEE international conference on robotics and automation (ICRA), 2017, pp. 3626–3633. https://doi.org/10.1109/ICRA.2017.7989417
[19] K. Albarrak, Y. Gulzar, Y. Hamid, A. Mehmood, and A. B. Soomro, “A deep learning-based model for date fruit classification,” Sustainability, vol. 14, no. 10, p. 6339, 2022. https://doi.org/10.3390/su14106339
[20] M. Faisal, F. Albogamy, H. Elgibreen, M. Algabri, and F. A. Alqershi, “Deep learning and computer vision for estimating date fruits type, maturity level, and weight,” IEEE Access, vol. 8, pp. 206770–206782, 2020. https://doi.org/10.1109/ACCESS.2020.3037948

[21] A. Alsirhani, M. H. Siddiqi, A. M. Mostafa, M. Ezz, and A. A. Mahmoud, "A novel classification model of date fruit dataset using deep transfer learning," Electronics (Basel), vol. 12, no. 3, p. 665, 2023. https://doi.org/10.3390/electronics12030665

[22] D. Chaudhari and S. Waghmare, "Machine vision based fruit classification and grading—a review," in ICCCE 2021: Proceedings of the 4th International Conference on Communications and Cyber Physical Engineering, 2022, pp. 775–781. https://doi.org/10.1007/978-981-16-7985-8_81

[23] T. Shoshan, A. Bechar, Y. Cohen, A. Sadowsky, and S. Berman, "Segmentation and motion parameter estimation for robotic Medjoul-date thinning," Precis Agric, vol. 23, no. 2, pp. 514–537, 2022. https://doi.org/10.1007/s11119-021-09847-2

[24] K. M. Alresheedi, S. Aladhadh, R. U. Khan, and A. M. Qamar, "Dates Fruit Recognition: From Classical Fusion to Deep Learning.," Computer Systems Science & Engineering, vol. 40, no. 1, 2022. https://doi.org/10.32604/csse.2022.017931

[25] A. Magsi, J. A. Mahar, S. H. Danwar, and others, "Date fruit recognition using feature extraction techniques and deep convolutional neural network," Indian J Sci Technol, vol. 12, no. 32, pp. 1–12, 2019. https://doi.org/10.17485/ijst/2019/v12i32/146441

[26] A. Nasiri, A. Taheri-Garavand, and Y.-D. Zhang, "Image-based deep learning automated sorting of date fruit," Postharvest Biol Technol, vol. 153, pp. 133–141, 2019. https://doi.org/10.1016/j.postharvbio.2019.04.003

[27] B. D. Pérez-Pérez, J. P. Garcia Vazquez, and R. Salomón-Torres, "Evaluation of convolutional neural networks' hyperparameters with transfer learning to determine sorting of ripe medjool dates," Agriculture, vol. 11, no. 2, p. 115, 2021. https://doi.org/10.3390/agriculture11020115

[28] H. Altaheri, M. Alsulaiman, and G. Muhammad, "Date fruit classification for robotic harvesting in a natural environment using deep learning," IEEE Access, vol. 7, pp. 117115–117133, 2019. https://doi.org/10.1109/ACCESS.2019.2936536

[29] M. F. Nadhif and S. Dwiasnati, "Classification of Date Fruit Types Using CNN Algorithm Based on Type," MALCOM: Indonesian Journal of Machine Learning and Computer Science, vol. 3, no. 1, pp. 36–42, 2023. https://doi.org/10.57152/malcom.v3i1.724

[30] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size," arXiv preprint arXiv:1602.07360, 2016.

[31] C. Szegedy et al., "Going deeper with convolutions," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1–9. https://doi.org/10.1109/CVPR.2015.7298594

[32] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in International conference on machine learning, 2019, pp. 6105–6114.

[33] X. Zhang, X. Zhou, M. Lin, and J. Sun, "Shufflenet: An extremely efficient convolutional neural network for mobile devices," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 6848–6856. https://doi.org/10.1109/CVPR.2018.00716

[34] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 4510–4520. https://doi.org/10.1109/CVPR.2018.00474

[35] "Pretrained Deep Neural Networks." Accessed: Feb. 25, 2024. [Online]. Available: https://www.mathworks.com/help/deeplearning/ug/pretrained-convolutional-neural-networks.html