

Fruits Recognition using Deep Convolutional Neural Network for Low Computing Device

Irene Anindaputri Iswanto^{1*}, Amadeus Suryo Winoto², Michael Kristianus³

¹⁻³ Computer Science Department, School of Computer Science,
Bina Nusantara University,
Jakarta, Indonesia 11480
irene.iswanto@binus.ac.id; amadeus.winoto@binus.ac.id;
michael.kristianus@binus.ac.id

*Correspondence: irene.iswanto@binus.ac.id

Abstract– Artificial intelligence is one of the most developed fields in Computer Science where a lot of researches had been done to make the computer smarter to perform human-like task. One of the most common human-life task research that had been done is object recognition. Convolutional Neural Network is one of the most popular deep learning model to perform a good object recognition. While improving CNN model can be done by simply increasing the depth of its architecture, some researchers prove that as the CNN architecture go deeper, the accuracy will get worse. ResNet, with their residual layer, successfully lift the limitation, but ResNet by itself is too heavy for a mobile or low computing device. This paper proposes a new model which could reach the accuracy of ResNet while having faster prediction time. The proposed model and other state-of-the-art models had been trained on our own fruits and vegetables dataset. The result shows that the proposed model can reach the same accuracy as Resnet110 and overcome the accuracy of DenseNet121 while being faster than those models.

Keywords: Deep Convolutional Neural Network; Deep Learning; Image Recognition; Low Computing Device

I. INTRODUCTION

Artificial Intelligence development played an important role in attempt to make the daily work easier, like helping us in making agenda, reading texts, translating or summarizing them, cooking some foods, and many others. Artificial Intelligence is divided into various sub-field, such as: Neural Networks, Evolutionary Computation, Vision, Robotics, Expert Systems, Speech Processing, Natural

Language Processing, Planning, and Machine Learning (Khaleel et al., 2023).

An example of Artificial Intelligence technology is Automated Online Inventory System or well known as “Smart Fridge (Avinash et al., 2020), (Zhou et al., 2021). The automated online inventory system has database, user interface, and an ability of object classification, using image recognition or RFID (Fujiwara et al., 2018). Some research on classifying the object in smart fridge system used RFID and Barcode to classify or recognize the ingredients (Miniaoui et al., 2019) (Nejekar et al., 2022). The use of RFID tag seems simpler and require low cost, however it may lacks some additional information that can be used for more advanced task, such as the condition of the object. Then another idea is proposed which uses Convolutional Neural Network to classify the object image and display the output data into a tablet that has been planted in the Fridge’s door (Buzzelli et al., 2018), (Ashwathan et al., 2022). Instead of identifying the object using RFID tag, image recognition method is performed to identify the type and the condition of the object in the inventory.

Since Convolutional Neural Network (CNN) won the image classification competition 2012 (ILSVRC12), lots of attention has been paid to deep layer CNN study (Han et al., 2018). CNN is widely used because of its advantages that enables raw images to be processed by an end to end system to some categories output without creating a specific feature extractor. In 2012, a more advanced CNN architecture called as AlexNet (Krizhevsky et al., 2017) (Ni et al., 2021) was developed that contains of 8 neural network layers, 5 convolutional layers, and 3 fully connected layers. This AlexNet has outperformed the performance of the traditional CNN due to some additional layers added into it.

Every layer added to the network will contribute to different feature learning, yet there is a maximum threshold for depth with the traditional CNN model (Pei et al., 2021). At certain depth CNN may lose its performance due to the famous vanishing gradient problem (Dhruv & Naskar, 2020). Since then, a vast improved CNN architecture had been developed to obtain higher performance in both accuracy and time processing while also overcoming the vanishing gradient problem caused by deeper network. VGG-16 (Simonyan & Zisserman, 2014) (Parashar et al., 2022), InceptionNet (Szegedy et al., n.d.) (Nikhitha et al., 2019), ResNet (He et al., n.d.) (Yang et al., 2022), DenseNet (Dong et al., 2022; Huang et al., n.d.) are some of the advanced CNN architecture that had been developed. However despite the advantages, each of these CNN architectures also have several drawbacks.

Since Deep Convolutional Neural Network (AlexNet) won the image classification competition in 2012, AlexNet is widely used to solve image classification tasks (Ni et al., 2021) (S. Li et al., 2021) (Titoriya & Sachdeva, 2019) (Patino-Saucedo et al., 2018). There are four major highlights that were presented by AlexNet: (a) ReLu activation function was used instead of tanh to add non-linearity, (b) the introduction of dropout layer to deal with overfitting, (c) the use of stacked pooling to reduce the size of the network, and (d) input with fix size 256x256 pixels (Krizhevsky et al., 2017). AlexNet contains five convolutional layers and three fully connected layers with eight layers in total. Since AlexNet, the state-of-the-art CNN architecture is going deeper and deeper. VGG Network (Simonyan & Zisserman, 2014) and GoogleNet (Inception v1) (Szegedy et al., n.d.) have 19 and 22 layers respectively. The intuition behind this architecture is by adding more layers to the architecture it will progressively make it capable to learn more complex features. The first layer will learn edges, the second layer learns shapes, the third layer learns objects, and so on. However there is a maximum threshold for depth with the traditional CNN model (Pei et al., 2021). It shows that when the depth of CNN architecture reach certain number (56 in the experiment), the training error is increased. This happened due to a famous problem called as vanishing gradient problem. Thus it can be concluded that constructing deeper CNN is not a best solution to obtain high performance.

Residual Network or ResNet is a neural network architecture that implements Residual Learning to overcome this vanishing gradient problem. Previously, depth of DCNN are capped from 16 to 19 layers. Residual learning can help achieving deeper network while still producing high performance for image classification and image segmentation (B. Li & He, 2018). Residual mapping is easier to optimize rather than normal unreferenced mapping because if an identity mapping is optimal, it would be easier to push the residual to zero rather than fitting an identity mapping to a stack of nonlinear layers.

YOLO is a neural network architecture that focuses on detection speed for real-time image recognition (Latha et al., 2022) (Liang et al., 2022) (Xu et al., 2022). It has advantages in: (a) speed because it uses a single network

as a pipeline, (b) lower background error because YOLO see image as a whole when making a prediction which could make it understand contextual information, and (c) endurance because it is highly generalizable resulting in lower chance of breaking when handling unexpected input (Redmon et al., n.d.). Nevertheless, YOLO accuracy still falls behind and struggling to detect small object. YOLO works by implementing Darknet framework as its backbone. In the first version, they add four convolutional layers and two fully connected layers, then last layer make class probabilities prediction and find bounding box coordinates. In the second version, a new model similar to VGG models is proposed and they use global average pooling to make prediction, compress the feature representation, batch normalization to stabilize training, speed up convergence, and regularize model (Redmon & Farhadi, n.d.). The final model has 19 convolutional layers and 5 maxpooling layers. In the third version, the network is improved by adding Residual learning and become remarkably deeper (Redmon & Farhadi, 2018). It has 53 convolutional layers and thus named as Darknet-53. Redmon and Farhadi stated that Darknet-53 performance is similar to ResNet model but with fewer floating-point operation which is make it more efficient and faster.

Another well known CNN-based architecture is GoogleNet or InceptionNet. The most convenient way to improve neural network performance is by enlarging its size. By size it could mean the depth of the network or the width - the number of units in each level. In the third version of InceptionNet, the idea to improve the network by using factorizing Convolutions is proposed. Factorizing convolutions can be performed by replacing a convolution layer with a bigger kernel to multiple convolution layers with smaller kernel (Szegedy et al., n.d.). This can lessen computational cost by reducing the number of parameters while maintain the network efficiency. Factorization will give very good result on medium grid-size when in feature maps in range 12 and 20.

Lastly, there is Densely Connected Convolutional Network (DenseNet) architecture that utilize short path characteristic that has been implemented in many network topologies (Zhong et al., 2020) (Herman et al., 2021). It is a simple connectivity pattern with short path characteristic to establish maximum information flow in the network by connecting each layer directly (Huang et al., n.d.). It means that DenseNet preserve feed-forward nature by append all preceding inputs to the result of current layer. DenseNet and ResNet might look similar but DenseNet always concatenate the feature before passed into a layer rather than adding them like ResNet.

By analyzing all the advantages obtained from every state-of-the-art CNN-based architecture above, a proposed architecture is presented. The proposed architecture will hold the base concept of inceptionNet while also applying other CNN-based architecture to increase its performance in accuracy and processing time.

This paper focus on the task of Fruits Image Classification, which is important part in the Automated Online Inventory System or Smart Fridge System.

Strength and drawbacks for each CNN architecture are analyzed, then a new Deep Convolutional Neural Network Architecture is proposed by combining some of the existing CNN architectures to overcome each of the architecture drawbacks for the task of fruits image classification. To prove that the proposed model can compete with the state-of-the-art architecture, an evaluation system will be performed to evaluate the accuracy and processing time along with the complexity.

II. METHODS

The research conducted in this paper will be executed by following the steps and procedures shows in the Figure 1. Each of the step will be explained in the following subsections:

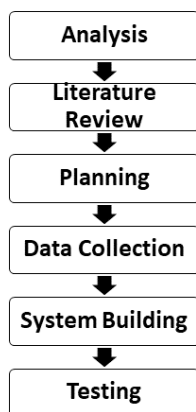


Figure 1. Research Framework Flowchart

2.1. Analysis, Literature Review, and Planning

This step analyzes the requirement of the system and the CNN models to be developed. Convolutional Neural Network itself has a limitation in the form of the accuracy loss caused by over fitting or under fitting. Deep Convolutional Neural Network such as ResNet and DenseNet can be used to solve those problems by either adding or concatenating the input to the output. Thus ResNet and DenseNet are taken as a comparison and goal for the accuracy.

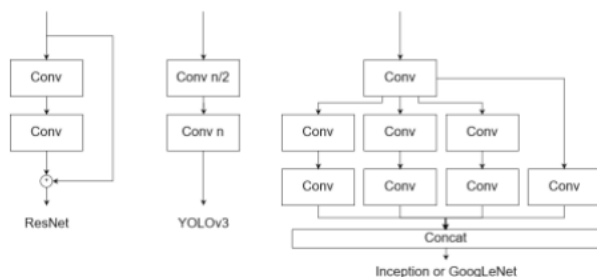


Figure 2. State-of-the-art CNN-Based Architectures

As stated in the previous section, in this paper some novel CNN-based architectures (show in Figure 2) will be presented by utilizing the concept of some existing state-of-the-art CNN-based architectures with few modifications as explained follows:

- **ResNet:** ResNet, as we can see in the Figure 2, has properties where the input of the block were added to the result of the block using what is called as skip connection. However in the proposed model, the input

could be processed differently than the block to get a better result

- **DenseNet:** In a way, DenseNet is almost identical with ResNet, where the input is now concatenated to the result rather than adding it.
- **YOLO:** As for YOLO, we used the bottleneck blocks which convert the input to smaller layer, then goes to bigger layer as explained in the following equation or stack it twice.

$$n \rightarrow \frac{n}{4} \rightarrow n \quad (1)$$

$$n \rightarrow \frac{n}{4} \rightarrow n \rightarrow \frac{n}{4} \rightarrow n \quad (2)$$

- **InceptionNet:** As for the InceptionNet, the wide block is used to reach higher accuracy with lower number of parameters. But there are two types of blocks, which is the original and factorized blocks. The original blocks mainly uses convolution layers, while the factorized blocks may use multiple smaller convolution layers or using the 1x n x1 convolution layers.

2.2. Data Collection

Images for the dataset will be collected by crawling on the internet to get images of objects to be recognized, which includes: apples, avocados, bananas, carrots, garlic, grapes, kiwis, mangoes, oranges, pineapples, tomatoes. Raw dataset consists of various number of images collected. After that, the dataset will be normalized by equalizing the number of images in each class. An observation will be made on whether dataset with equal number of images will affect the performance of CNN model. Then, the model will be trained using the normalized dataset, and after analyzing the result of each class, both of the previously made dataset will be combined by taking the highest class accuracy after comparing both classes to make it better.

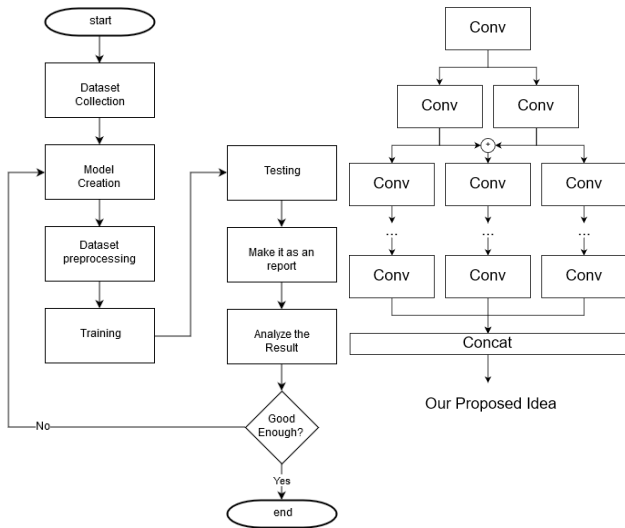
To test the stability of the convolutional model, the model will be trained using the data with different numbers of images. And the model will be tested with two types of validation tests. The validation test itself consisted of 66 images and 88 images. Table 1 shows the details of each dataset.

Table 1. Student Distribution Frequency

Fruits	Dataset 1		Dataset 2		Dataset 3	
	Train	Test	Train	Test	Train	Test
Apple	58	5	60	8	58	8
Avocado	50	8	60	8	50	8
Banana	36	6	60	8	36	9
Carrot	27	6	60	8	27	8
Garlic	36	8	60	8	36	8
Grape	41	8	60	8	60	8
Kiwi	60	5	60	8	60	8
Mango	45	6	60	8	60	10
Orange	35	8	60	8	45	9
Pineapple	34	8	60	8	60	8
Tomato	51	7	60	8	51	8
Total	473	75	660	88	543	92

2.3. Model Building

The research will be done by following the flowchart presented in the Figure 3. Convolutional Neural Network (CNN) model creation is the main purpose of this methodology. First, the standard CNN will be used. Afterwards, improving the accuracy of the standard CNN by adding and removing some layer, or changing the number of kernel used, or changing the kernel size or implementing the ideas from the state-of-art architectures as mentioned in the previous subsection, while constantly train and evaluate the CNN made. This paper will implement the proposed model into the CNN bit by bit.



(a) CNN Model Development Structure (b) Proposed Architecture Structure

Figure 3. Our CNN Model Development and Proposed Model

And as the state-of-art architectures proves, mostly, CNN will have a better accuracy if they are made deep (He et al., n.d.). However there is a different idea, where the Inception is made wider and deeper than other CNN architectures (Szegedy et al., n.d.). Hence, the best architecture that made is found. The proposed model was developed using the concept of

InceptionNet which consisted of three main line (strand) which later on combines into one as shown in the Figure 4.

Table 2. Top-5 Proposed Architecture

Model	Base Architecture	Depth	Layer	Information	State-of-the-art Model
CNN_3	Standard CNN	37	54	Residual: Conv, conv	ResNet and YOLO
CNN_5	Standard CNN	40	61	Residual: Stacked Bottleneck, Bottleneck	ResNet and YOLO
CNN_11	Proposed Architecture	40	125	Factorization Layer	Residual and Inception Net
CNN_29	Proposed Architecture	44	151	3 blocks each strand	ResNet and YOLO
CNN_30	Proposed Architecture	48	173	4 blocks each strand	ResNet, InceptionNet, and YOLO

There are missing parts where it could be filled by the layers that already explained briefly in previous subsection such as Residual layer, Batch Normalization layer, Dropout

layer, Inception layer, Stacked Bottleneck layer.

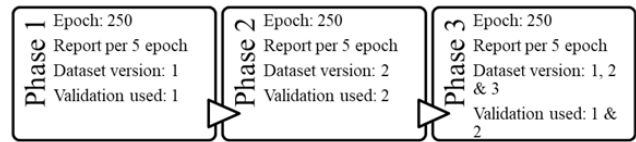


Figure 4. Training Phase

After creating the models, the image dataset will be resized into 112, then normalizing the images as in equation 3 which means for every i and j from 0 until the respective weight and height, the member of matrix x_{ij} will be divided with 255. The result of equation 3 will be normalized matrix where all of the member will be $0 < x_{ij} < 1$.

$$X_{i,j} = \frac{x_{i,j}}{255} \quad (3)$$

Model training regime is shown in Figure 4. Other training menu to compare the top-7 model to the state-of-art architectures will be proposed. This process results in training one model by one model until a full list of 30 models have been gotten in the end which follows the Phase 1 training. Followed by training in Phase 2 resulted in the accuracy from validation set 2. After the accuracy of validation set 1 and validation set 2 has been gotten, then continuing to calculate the average accuracy of each model as in attachment 1 until 3 in L-1 and L-2, which then gives us the top-5 models as can be seen in table 2 along with each of the model differences.

III. RESULTS AND DISCUSSION

3.1. Experiment Results

To compare the proposed model with the state-of-art architectures, training the top 5 models along with DenseNet121, ResNet50, and ResNet110 is necessary. To compare the models, the evaluation of the current model every 5 epoch which shows the validation accuracy like previously done before is also needed, and training the models with data augmentation or image processing to improve the accuracy. The data augmentation setting is as follows:

- Image are re-scaled or normalized by dividing it by 255.
- Image may be rotated randomly within -20 to 20 degrees.
- Image may be flipped horizontally.
- Image may be shifted from -10% to 10% from the original size horizontally or vertically.

Table 3. Results of Batch Prediction on Validation Data

Model	Val_1		Val_2		Average	
	Acc	Time	Acc	Time	Acc	Time
CNN_3	90,91%	162	81,82%	166	86,36%	164
CNN_5	90,91%	105	78,41%	103	84,66%	104
CNN_11	86,36%	37	78,41%	38	82,39%	38
CNN_29	92,42%	49	81,82%	49	87,12%	49
CNN_30	95,45%	38	85,23%	37	90,34%	38
DenseNet121	95,45%	46	84,09%	44	89,77%	45
ResNet50	95,45%	74	85,23%	73	90,34%	74
ResNet110	95,45%	157	85,23%	143	90,34%	150

Table 3 shows that CNN_30 is considered as the best model by having the high accuracy and fast prediction time. We also applied a four different filter to each images dataset such as: vignette, paler color tone, adding more contrast, and other random filter. After the CNN_30 and state-of-art models are used to predict the base images and the filtered images, the result of prediction is analyzed. The results shows that the proposed model, CNN_30 successfully reach the same accuracy performance with ResNet50 and ResNet110 and outperformed other state-of-the-art architectures such as DenseNet121. In terms of time, it also shows that the proposed model CNN_30 is faster than any other state-of-the-art models.

3.2. Discussion

Figure 5 shows the mapping of accuracy, time, and complexity of each model which uses the y-axis as accuracy, x-axis as complexity and the area of the circle is the time used. As seen in Figure 5 CNN_30 can predict as good as the other architecture such as ResNet50 and ResNet110. The CNN_30 model also runs faster than DenseNet121, ResNet50, and ResNet110. This statement also supported by the result of the experiment on filtered images, which shows that CNN_30 can work better than ResNet50 and DenseNet121 and can give equal performance with ResNet110.

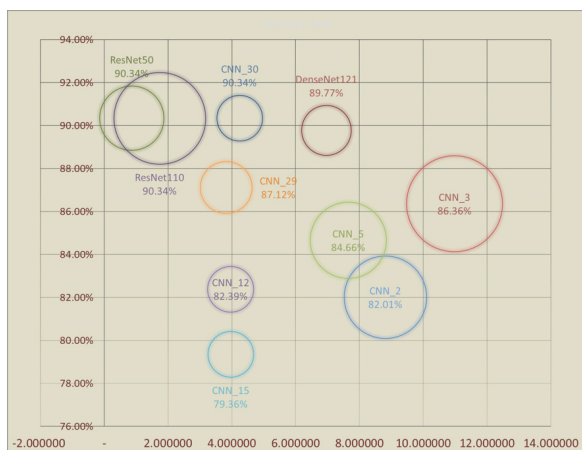


Figure 5. Mapping of each Model Accuracy, Time, and Complexity

In addition, in terms of the model size, our proposed model CNN_30 is also considered as light size model, compare to other state-of-the-art architectures as shown in the Table 4. It means our proposed model CNN_30 is also suitable to be implemented in the mobile device or embedded system for smart fridge system which requires small memory.

Table 4. Size of each Model

Model	Model Complexity	Model Size (kb)
CNN_3	10.980.427	42.895
CNN_5	7.649.211	29.884
CNN_11	3.962.707	15.510
CNN_29	3.825.187	14.982
CNN_30	4.248.431	16.626
DenseNet121	6.965.131	27.286
ResNet50	863.403	3.388
ResNet110	1.740.363	6.827

IV. CONCLUSIONS

Based on the experiment, conclusions that can be made are:

- CNN_30 has been made and capable to classify fruit and vegetables even with only 60 images per class and 660 training images in total.
- CNN_30 reaches 90.34% accuracy, better than DenseNet, and on par with ResNet50 and ResNet110
- CNN_30 predicts each image within 38 ms in average, which is faster than ResNet50 (74 ms), ResNet110 (150 ms), and DenseNet121 (45 ms).
- CNN_30 is also suitable to be implemented in a mobile or embedded device system due to its small model size.

REFERENCES

- Ashwathan, R., Asnath, V. P. Y., Geetha, S., & Kalaivani, K. (2022). Object Detection in IoT-Based Smart Refrigerators Using CNN. In *The Industrial Internet of Things (IIoT)* (pp. 281–300). Wiley. <https://doi.org/10.1002/9781119769026.ch11>
- Avinash, N. J., Pinto, R., Bhat, S., Chetan, R., & Moorthy, H. R. (2020, December 10). Smart Fridge for Global Users Based on IOT Using Deep Learning. *ICPECTS 2020 - IEEE 2nd International Conference on Power, Energy, Control and Transmission Systems, Proceedings*. <https://doi.org/10.1109/ICPECTS49113.2020.9337016>
- Buzzelli, M., Belotti, F., & Schettini, R. (2018). Recognition of Edible Vegetables and Fruits for Smart Home Appliances. *2018 IEEE 8th International Conference on Consumer Electronics - Berlin (ICCE-Berlin)*, 1–4. <https://doi.org/10.1109/ICCE-Berlin.2018.8576236>
- Dhruv, P., & Naskar, S. (2020). *Image Classification Using Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN): A Review* (pp. 367–381). https://doi.org/10.1007/978-981-15-1884-3_34
- Dong, Y., Huang, Y., Xu, B., Li, B., & Guo, B. (2022). Bruise detection and classification in jujube using thermal imaging and <scp>DenseNet</scp>. *Journal of Food Process Engineering*, 45(3). <https://doi.org/10.1111/jfpe.13981>
- Fujiwara, M., Moriya, K., Sasaki, W., Fujimoto, M., Arakawa, Y., & Yasumoto, K. (2018, August 13). A smart fridge for efficient foodstuff management with weight sensor and voice interface. *ACM International Conference Proceeding Series*. <https://doi.org/10.1145/3229710.3229727>
- Han, D., Liu, Q., & Fan, W. (2018). A new image classification method using CNN transfer learning and web data augmentation. *Expert Systems with Ap-*

- plications*, 95, 43–56. <https://doi.org/10.1016/j.eswa.2017.11.028>
- He, K., Zhang, X., Ren, S., & Sun, J. (n.d.). *Deep Residual Learning for Image Recognition*. <http://image-net.org/challenges/LSVRC/2015/>
- Herman, H., Cenggoro, T. W., Susanto, A., & Pardamean, B. (2021). Deep Learning for Oil Palm Fruit Ripeness Classification with DenseNet. *2021 International Conference on Information Management and Technology (ICIMTech)*, 116–119. <https://doi.org/10.1109/ICIMTech53080.2021.9534988>
- Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (n.d.). *Densely Connected Convolutional Networks*. <https://github.com/liuzhuang13/DenseNet>
- Khaleel, M., Ahmed, A. A., & Alsharif, A. (2023). Artificial Intelligence in Engineering. *Brilliance: Research of Artificial Intelligence*, 3(1), 32–42. <https://doi.org/10.47709/brilliance.v3i1.2170>
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6), 84–90. <https://doi.org/10.1145/3065386>
- Latha, R. S., Sreekanth, G. R., Rajadevi, R., Nivetha, S. K., Kumar, K. A., Akash, V., Bhuvanesh, S., & Anbarasu, P. (2022). Fruits and Vegetables Recognition using YOLO. *2022 International Conference on Computer Communication and Informatics (ICCCI)*, 1–6. <https://doi.org/10.1109/ICCCI54379.2022.9740820>
- Li, B., & He, Y. (2018). An Improved ResNet Based on the Adjustable Shortcut Connections. *IEEE Access*, 6, 18967–18974. <https://doi.org/10.1109/ACCESS.2018.2814605>
- Li, S., Wang, L., Li, J., & Yao, Y. (2021). Image Classification Algorithm Based on Improved AlexNet. *Journal of Physics: Conference Series*, 1813(1). <https://doi.org/10.1088/1742-6596/1813/1/012051>
- Liang, X., Jia, X., Huang, W., He, X., Li, L., Fan, S., Li, J., Zhao, C., & Zhang, C. (2022). Real-Time Grading of Defect Apples Using Semantic Segmentation Combination with a Pruned YOLO V4 Network. *Foods*, 11(19). <https://doi.org/10.3390/foods11193150>
- Miniaoui, S., Atalla, S., & Bin Hashim, K. F. (2019). Introducing Innovative Item Management Process Towards Providing Smart Fridges. *2019 2nd International Conference on Communication Engineering and Technology (ICCET)*, 62–67. <https://doi.org/10.1109/ICCET.2019.8726900>
- Nejekar, S. M., Nataraj, K. R., Rekha, K. R., Sheela, S., Pooja, P., & Nafeesa, K. S. (2022). *Raspberry Pi Based Smart Refrigerator to Recognize Fruits and Vegetables* (pp. 1055–1065). https://doi.org/10.1007/978-981-16-3690-5_100
- Ni, J., Gao, J., Li, J., Yang, H., Hao, Z., & Han, Z. (2021). E-AlexNet: quality evaluation of strawberry based on machine learning. *Journal of Food Measurement and Characterization*, 15(5), 4530–4541. <https://doi.org/10.1007/s11694-021-01010-9>
- Nikhitha, M., Roopa Sri, S., & Uma Maheswari, B. (2019). Fruit Recognition and Grade of Disease Detection using Inception V3 Model. *2019 3rd International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, 1040–1043. <https://doi.org/10.1109/ICECA.2019.8822095>
- Parashar, N., Mishra, A., & Mishra, Y. (2022). *Fruits Classification and Grading Using VGG-16 Approach* (pp. 379–387). https://doi.org/10.1007/978-981-19-0976-4_31
- Patino-Saucedo, A., Rostro-Gonzalez, H., & Conradt, J. (2018). Tropical fruits classification using an alexnet-type convolutional neural network and image augmentation. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11304 LNCS, 371–379. https://doi.org/10.1007/978-3-030-04212-7_32
- Pei, Y., Huang, Y., Zou, Q., Zhang, X., & Wang, S. (2021). Effects of Image Degradation and Degradation Removal to CNN-Based Image Classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(4), 1239–1253. <https://doi.org/10.1109/TPAMI.2019.2950923>
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (n.d.). *You Only Look Once: Unified, Real-Time Object Detection*. <http://pjreddie.com/yolo/>
- Redmon, J., & Farhadi, A. (n.d.). *YOLO9000: Better, Faster, Stronger*. <http://pjreddie.com/yolo9000/>
- Redmon, J., & Farhadi, A. (2018). *YOLOv3: An Incremental Improvement*. <http://arxiv.org/abs/1804.02767>
- Simonyan, K., & Zisserman, A. (2014). *Very Deep Convolutional Networks for Large-Scale Image Recognition*. <http://arxiv.org/abs/1409.1556>
- Szegedy, C., Vanhoucke, V., Ioffe, S., & Shlens, J. (n.d.). *Rethinking the Inception Architecture for Computer Vision*.
- Titoriya, A., & Sachdeva, S. (2019). Breast Cancer Histopathology Image Classification using AlexNet. *2019 4th International Conference on Information Systems and Computer Networks (ISCON)*, 708–712. <https://doi.org/10.1109/ISCON47742.2019.9036160>
- Xu, C., Liu, Z., & Tan, J. K. (2022). *Fruits and Vegetables Detection using the Improved YOLOv3*.
- Yang, Y., Wang, L., Huang, M., Zhu, Q., & Wang, R. (2022). Polarization imaging based bruise detection of nectarine by using ResNet-18 and ghost

bottleneck. *Postharvest Biology and Technology*, 189, 111916. <https://doi.org/10.1016/j.postharvbio.2022.111916>

Zhong, Z., Zheng, M., Mai, H., Zhao, J., & Liu, X. (2020). Cancer image classification based on DenseNet model. *Journal of Physics: Conference Series*, 1651(1). <https://doi.org/10.1088/1742-6596/1651/1/012143>

Zhou, Y., Shi, L., & Yuan, B. (2021). A Generative Adversarial Network-based Framework for Fruit and Vegetable Occlusion Detection in Smart Refrigerators. *2021 International Conference on Signal Processing and Machine Learning (CONF-SP-ML)*, 290–295. <https://doi.org/10.1109/CONF-SPML54095.2021.00063>