

Classification of meat using the convolutional neural network

Detty Purnamasari¹, Koko Bachrudin², Dede Herman Suryana³, Robert³

¹Specialization in Information Technology, Universitas Gunadarma, Jakarta, Indonesia

²Specialization in Computer Information System, Universitas Gunadarma, Jakarta, Indonesia

³Specialization in Information Technology, Master of Electrical Engineering, Universitas Gunadarma, Jakarta, Indonesia

Article Info

Article history:

Received Sep 30, 2022

Revised Feb 16, 2023

Accepted Mar 10, 2023

Keywords:

Classification

Convolutional neural network

Deep learning

Image processing

Meat

ABSTRACT

Every animal meat has different color and texture, for example, beef has a dark red color with a chewy texture, while pork has a pale red color and smooth fiber. A previous study has classified types of meat using gray level co-occurrence matrix (GLCM), hue saturation value (HSV), and color intensity. In this research, we created meat classification between beef, pork, and horse meat using a convolutional neural network (CNN) develop in jupyter notebook, using the MobileNetV2 model, and 315 meat images as a dataset divided into 3 groups, 70% image for the training dataset, 20% image for the testing dataset, and 10% image for validation dataset. Before dividing the image into 3 groups, the image is resized to 224×224, and convert the color to grayscale. The model is trained with a training dataset, the epoch of 50, and Adam optimizer, the results show an accuracy of 93.15%.

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Dede Herman Suryana

Specialization in Information Technology, Master of Electrical Engineering, Universitas Gunadarma, Jakarta, Indonesia

Email: angamei.of.tei1@gmail.com

1. INTRODUCTION

Meat is a high nutritious food as it is filled with proteins, vitamins and many other nutrients that have many useful health effects [1]. With the economic growth and development of people's living standards, meat has become one of the important foods in the human diet. Meat quality such as meat color, tenderness, meat texture and juiciness can affect consumers that the important point in choosing the best meat is based on its texture, therefore will have an impact on consumers in purchasing meat with the best quality [2].

Animal meat has various colors and textures. For example, beef has dark red color with a chewy texture, the fiber in beef is rough and tight, has thick and hard fat, from the aroma typical of the smell of beef. Pork has pale red color, the fiber in pork is smooth and loose and has a soft texture, the fat in pork is thick and soft with a distinctive fishy aroma. Horse meat has dark red color with a thick texture with tenuous fibers [3]. There is also previous research using meat image by extracting its feature like color and texture and classified using k nearest neighbor (KNN), support vector machine (SVM), and neural network (NN) using neural net from RapidMiner [4].

The characteristic of horse meat is dark red color with a slight brown tint for older horses, because the color of the foal is similar to the color of beef. Horse meat is characterized by thin muscle fibers, interspersed with fatty tissue. Foal has a more delicate structure and is easier to digest than beef or pork. Tenderness of horse meat unlike other meats, is due to the presence of high level of connective tissue protein, as well as higher thermal resistance [5].

Classification of meat recognition can be done with several ways. The first way is by using human sight, second way can also use meat fat sample and chemical to classified meat, the third way is by using image between two types of meat and classification model. There are several related papers on meat classification has

been done, one of research using electronic nose and support vector machine (SVM) to detect pork adulteration in beef [6]. Some other research using raman spectroscopy and pure fat samples to distinguish between beef and horse meat [7].

In the branch of artificial intelligence, there is something specific that is learning process known as deep learning. Deep learning approach is an algorithm of answers to solve problems in human life today. Deep learning has proven to be superior in image processing [8].

The deep learning method that is often used in the field of image classification is convolutional neural networks [9]. Convolutional neural network (CNN) is type of multi-layer neural network inspired by mechanism of human thought. Convolutional neural network is a deep learning model consisting of several layers: Convolutional, pooling, and fully connected layers [10].

There are very few researches on meat classification by using image, therefore this research will create meat classification using MobileNetV2 and 3 types of meat which is beef, pork, and horse meat. MobileNetV2, is one of the convolutional neural network (CNN) architectures that can be used to overcome the need for excessive computing resources. The difference between the MobileNetV2 architecture and the CNN architecture in general is the use of a convolution layer with a filter thickness that matches the thickness of the input image. The MobileNetV2 architecture divides convolution into depthwise convolution and pointwise convolution [11]. This research expects MobileNetV2 to be able to classify meat between beef, pork, and horse meat.

2. METHOD

Convolutional neural network (CNN) is included in deep learning because it includes the depth of the network, deep learning is a branch of machine learning that can make computers do work like humans [12]–[16]. Convolutional neural network (CNN) is an enhancement of the multilayer perceptron (MLP) which is built to process two-dimensional data [17], [18]. The convolution layer uses filters to extract an object. This filter is a weight used to identify objects such as edges, curves, or colors. The output of the convolution is in the form of a linear transformation [19]–[22]. As illustrated in Figure 1, the CNN structure consists of input, process, augmentation, classification process and output. The extraction process carried out by CNN contains a hidden arrangement or what is commonly called a hidden layer, the hidden layer is a convolution arrangement, and there is reactified linear unit (ReLu), and pooling. CNN moves hierarchically so that the output of the initial convolution is used as input for the next convolution array [23], [24].

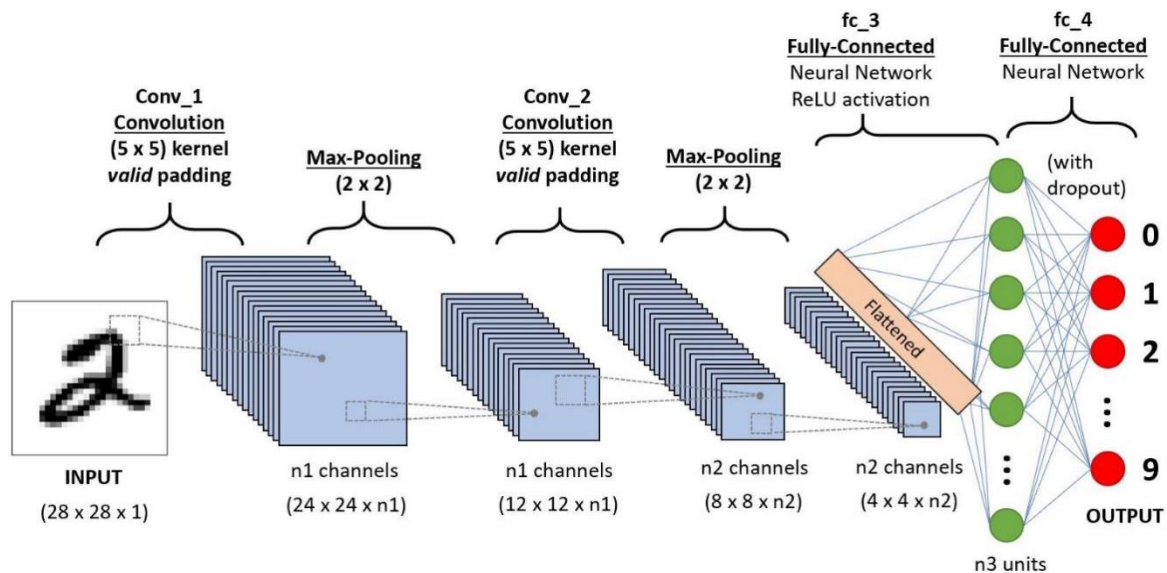


Figure 1. CNN layers

Before meat image dataset classified using CNN, preprocessing is carried out on the meat image dataset so that the meat image dataset can be fitted into MobileNetV2 as an input. The classification of meat types consists of several steps, these steps are image dataset acquisition, pre-processing, extraction features,

creating a model, training and validating a model with a training and validation dataset, testing a trained model with a test dataset. The flowchart detection system is shown in Figure 2.

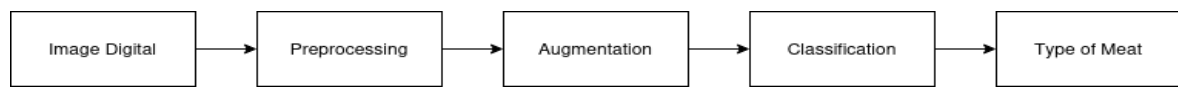


Figure 2. Flowchart detection system

2.1. Dataset

The data used in this research is obtained from internet, the dataset consist of 3 types of meat: beef, pork, and horse meat that already divided into 3 folders with total of 315 images, all of the images also have extension of .jpg. The images size varies and need to be resized into one size, the purpose of resizing image to one size is because CNN model only accepts fixed input. The dataset for beef uses breasts and thighs because these parts are easy to detect because they have good texture and fiber. For horse meat use only the breast because the part is very visible fiber to distinguish the type. For pork use the breast and thigh because it is easy to detect. The dataset for 3 types of meat took only the meat part, not the fat or bones. The meat images sample shown in Figure 3.



Figure 3. Meat image sample

2.2. Pre-Processing

2.2.1. Cropping and labelling image

Pre-processing of the original image data aims to improve image quality, for example removing noise in an image, sharpening the intensity of the edges of objects and removing the effect of blur. Noise can be interpreted as information that is recorded in the image, but this information is not needed. The convolutional neural network model of MobileNetV2 used in this research require fixed image input size width and height of 224×224, therefore all of the images in dataset cropped with the provisions of cropping starting from the left, right, top, and bottom of the image. The result obtained in the process of cutting this image is to emphasize the image in the middle, because it makes it easier for researchers to take important information about the image

such as color, texture, and fibers contained in the three meat images. This research use softmax activation in output layer, softmax classifier is another form of logistic regression algorithm that can be used to classify more than two classes. Softmax can produce more impulsive results and has a better probabilistic interpretation than other classification algorithms. Softmax can calculate probabilities for all labels [25]. Therefore, all of the image will be labeled with following rules: [1,0,0] for beef, [0,1,0] for horse meat, [0,0,1] for pork. The result of labeled image shown in Figure 4.

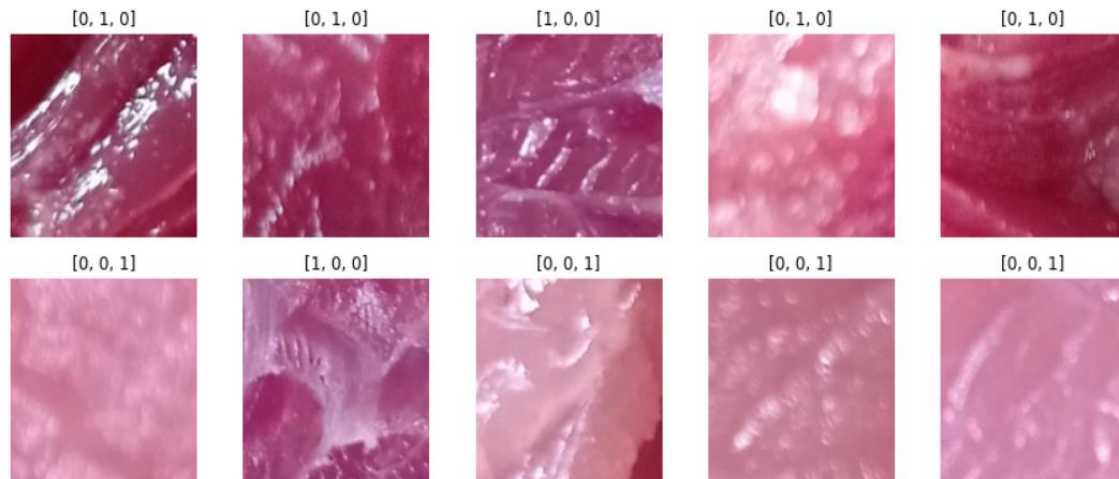


Figure 4. Result of labeling images

2.2.3. Split dataset

Splitting data is an instruction to split data into two or three parts. The first data section is used to develop the model, in this section it is called the data train. The second part is used to evaluate the performance of the model, in this part it is called test data. The third section is used to test the performance of the model that has been built, in this section it is called data validation [26]. In this research the ratio between training dataset, validation dataset, and testing dataset is 70%:20%:10%. After splitting dataset, here the following amount of each type dataset: training dataset 255 images, validation dataset 37 images, and testing dataset 73 images. Next is to convert color.

2.2.4. Convert image color to grayscale

Grayscale image is an image with one channel. Each pixel only represents the amount of light in an image, so it only produces intensity information. There are several methods to convert an RGB image into a grayscale image including the average method and the weighted method. In changing the color image using the average method following the formula $G = 1/3 (R + G + B)$. The Average method is considered less efficient for use on the human sense of sight to view an image. So that the weighted method was made where this method focuses on the green color component, because in human eyesight is more sensitive to the color green. The formula used in the weighted method is $G = 0.299R + 0.587G + 0.114B$ [27]. As illustrated in Figure 5, the images are still in red, green, blue (RGB) color model and to convert it to grayscale color model using weighted average grayscale. We need to convert the image to array using python library.

The array holds 3 attributes from images. First attribute hold image width coordinates, second attribute hold image height coordinates, third attribute hold image color channel. the array width and height start from image top left corner as coordinate [0,0]. Each coordinate hold 3 value which define image color start with red color, green color, and blue color. From Figure 5 we can see that the first coordinates [0,0] hold value of red color of 174, green color of 56, and blue color of 91. After all images converted into array, the next step is to convert image color to grayscale using weighted average grayscale. The result of converted image from RGB to grayscale shown in Figure 6.

It can be seen from Figure 6, that each coordinate of image array holds same value of red, green, and blue. It is because of color conversion from RGB to weighted average grayscale. As an example, we can see from Figure 4 for coordinates [0,0] hold red at 174, green at 56, blue at 91, if we using weighted average grayscale the average of RGB color will be calculated like this $(174 \times 0.299) + (56 \times 0.587) + (91 \times 0.114)$ which will result value of 94, the then assign to all color channel of corresponding coordinates.


```

RGB
[[ [174  56  91]
  [166  48  83]
  [165  47  82]
  ...
  [161  43  78]
  [162  44  79]
  [158  40  75]]

```

Figure 5. Image array

```

Grayscale
[[ [ 94  94  94]
  [ 86  86  86]
  [ 85  85  85]
  ...
  [ 81  81  81]
  [ 82  82  82]
  [ 78  78  78]]

```

Figure 6. Grayscale image array

In Figure 7 shows the result of some images color model conversion from RGB to Grayscale. From the result we can see that pork has whiter color while horse meat and beef have similar color with horse has darker spot like picture in the left top corner and right top corner. The last step of pre-processing is scaling image.

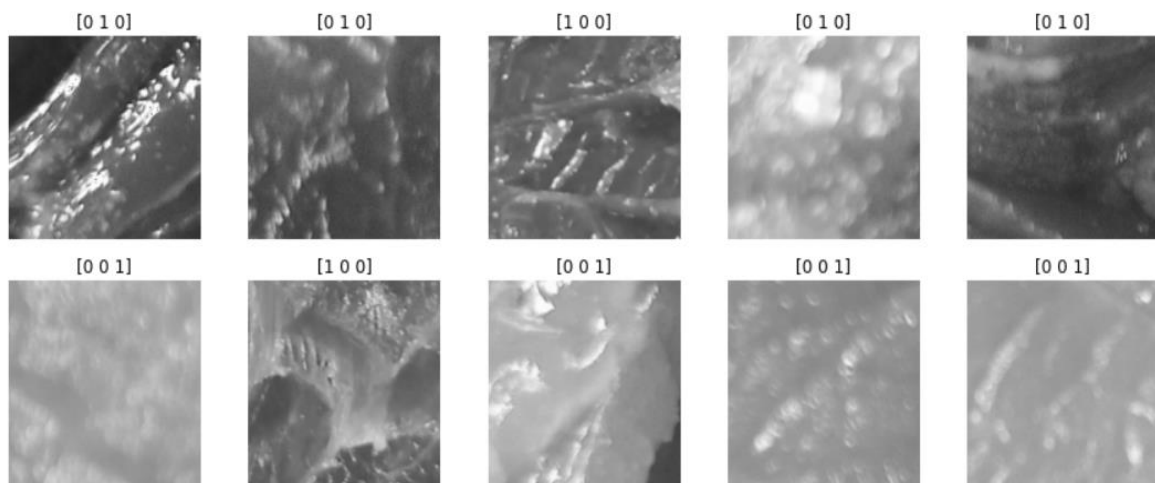


Figure 7. Grayscale image

2.2.5. Scaling image

The second to last step before using the dataset is to scaling image range of 0 to 1. This step also is required because the model that used in this research only accept color range between 0 to 1 [28]. At the moment the dataset color range are still in 0 to 255. To scaling it we divide the color range with maximum color range which is 255. As an example, for image in coordinates [0,0] with color value of 94. To scale it from [0...255] to [0...1] we divide it with 255, the calculation will be $94/255$ which the result is 0.3682745. The result then assigns to all color channels (red, green, blue) of their corresponding coordinates. The result of scaling image color ranges from [0,255] to [0,1] shown in Figure 8.

```

Grayscale + Scale
[[ [0.36862745 0.36862745 0.36862745]
  [0.3372549  0.3372549  0.3372549 ]
  [0.33333333 0.33333333 0.33333333]
  ...
  [0.31764706 0.31764706 0.31764706]
  [0.32156863 0.32156863 0.32156863]
  [0.30588235 0.30588235 0.30588235]]

[[ [0.34509804 0.34509804 0.34509804]
  [0.34117647 0.34117647 0.34117647]
  [0.31372549 0.31372549 0.31372549]]

```

Figure 8. Scaled image array

2.2.6. Data Augmentation

The augmentation method is the most appropriate method to solve such problems because it can create artificial images through various processing means or combinations of several processing, such as random rotation, shifting, contrast stretching, and sharpening. without changing the label of the image. The data available on the source used still amounts to 293 images. This amount of data is very small to be used in the image classification implementation process, so an augmentation process is needed to produce more data to get more optimal performance. The last step of pre-processing is to add to the dataset, since rotation and shift will cause to lose some part of the image and fill it with the nearest pixel value, to keep all the pixels of the image, augmentation only flips the image either vertically or horizontally.

2.3. Classification Using MobileNetV2

The CNN model used in this research is the MobileNetV2 architecture, which is an architecture developed by TensorFlow. Input images on the CNN model using images that are size 224×224. The input image will then be processed first by the MobileNetV2 architecture with an output of 1,280, dropout 0.5, dense 3 with softmax activation and adam optimizer [29].

Dropout is an algorithmic technique used for training data and serves to prevent data overfitting and provides a way to efficiently combine many different architecture types [30]. Dense is a function to add a fully connected layer. The keras_layer is the MobileNetV2 architecture with an output shape of 1,280, followed by a dropout with the rate of 0.5 and an output layer with activation softmax and the number of outputs is 3, these results can be seen in Figure 9.

Model: "sequential"

Layer (type)	Output Shape	Param #
keras_layer (KerasLayer)	(None, 1280)	2257984
dropout (Dropout)	(None, 1280)	0
dense (Dense)	(None, 3)	3843
Total params: 2,261,827		
Trainable params: 3,843		
Non-trainable params: 2,257,984		

Figure 9. Model summary

3. RESULTS AND DISCUSSION

The CNN MobileNetV2 model is trained with train dataset and validate with validation dataset that has been pre-processed, the model will be trained with epoch of 50. The training time per epoch took around 2 seconds with the longest training time 5 seconds only once, and the rest training time is 3 seconds in 8 epoch and 2 seconds in 41 epochs with average of 2.22 seconds training time per epoch and took around 111 seconds to train and validate the model. As illustrated in Figure 10, we can see that the model train accuracy increases from epoch 1 to epoch around 40 and decrease at several last epoch, and train loss decrease as epoch increases, it can be said that the train accuracy is high with low loss. The accuracy of validation not as high as train accuracy but it also can be said that validation accuracy is high because it is hit 97.22% at the end of training with loss 0.1081 at the end of training.

After the model is trained, it is used to classify the meat using a test dataset. As illustrated in Figure 11, meat yield is classified by model, out of 32 pictures there is 1 image classified in the wrong type. We can see from those results there is one image that is classified as horse meat but is actually meat, while the other in the example is classified correctly. From these results, it can be said that the model has an accuracy of 96.88%. The model then evaluates with the test dataset to get a loss value, and after evaluating we get a loss of 0.10.

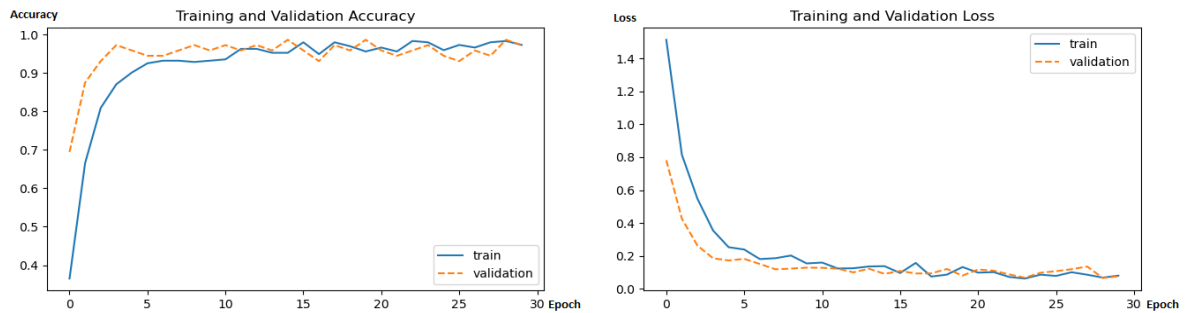


Figure 10. Model train history per epoch

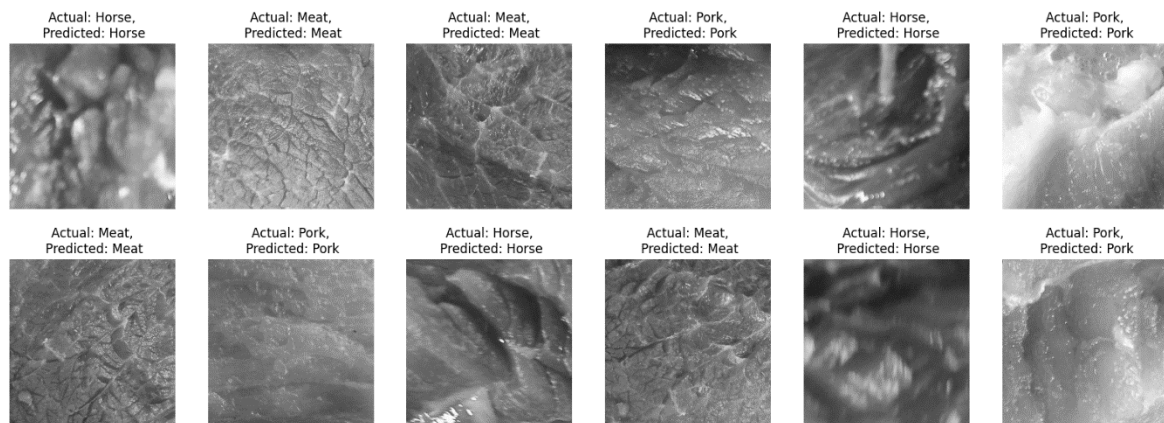


Figure 11. Example of meat classified by the model

4. CONCLUSION

In this research we create system using MobileNetV2 CNN model architecture. Using beef, pork, horse meat dataset from Kaggle with total of 315 images that crop size to 224×224 and split into 3 datasets, the CNN using MobileNetV2 architecture able to achieve accuracy of 93.15% with dropout rate of 0.5 and output using softmax activation. It can be concluded that the research was successfully carried out with very high excellent accuracy in classifying between beef, pork, and horse meat. For further research, it can be developed again by adding more data image of each meat, more types of meat such as chicken.

ACKNOWLEDGEMENTS

We would like thanks Universitas Gunadarma and the DGX UG Development Team.




REFERENCES

- [1] A. Arsalane, A. Klilou, N. El Barbri, and A. Tabyaoui, "Artificial vision and embedded systems as alternative tools for evaluating beef meat freshness," *6th International Conference on Optimization and Applications, ICOA 2020 - Proceedings*, 2020, doi: 10.1109/ICOA49421.2020.9094503.
- [2] F. Tao and Y. Peng, "A method for nondestructive prediction of pork meat quality and safety attributes by hyperspectral imaging technique," *Journal of Food Engineering*, vol. 126, pp. 98–106, 2014, doi: 10.1016/j.jfoodeng.2013.11.006.
- [3] D.-S. Kim and N. Joo, "Texture characteristics of horse meat for the elderly based on the enzyme treatment," *Food Science of Animal Resources*, vol. 40, no. 1, pp. 74–86, 2020, doi: 10.5851/kosfa.2019.e86.
- [4] J. Chairunnisa *et al.*, "Pork and beef features extractions," *Proceedings - 2018 International Seminar on Application for Technology of Information and Communication: Creative Technology for Human Life, iSemantic 2018*, pp. 295–298, 2018, doi: 10.1109/ISEMANTIC.2018.8549765.
- [5] R. Stanislawczyk, M. Rudy, and S. Rudy, "The quality of horsemeat and selected methods of improving the properties of this raw material," *Processes*, vol. 9, no. 9, 2021, doi: 10.3390/pr9091672.
- [6] R. Sarno, K. Triyana, S. I. Sabilla, D. R. Wijaya, D. Sunaryono, and C. Fatichah, "Detecting pork adulteration in beef for halal authentication using an optimized electronic nose system," *IEEE Access*, 2020, doi: 10.1109/ACCESS.2020.3043394.
- [7] I. H. Boyaci, H. T. Temiz, R. S. Uysal, H. M. Velioglu, R. J. Yadegari, and M. M. Rishkan, "A novel method for discrimination of beef and horsemeat using Raman spectroscopy," *Food Chemistry*, vol. 148, pp. 37–41, 2014, doi: 10.1016/j.foodchem.2013.10.006.
- [8] Y. Gulzar, Y. Hamid, A. B. Soomro, A. A. Alwan, and L. Journaux, "A convolution neural network-based seed classification




- system," *Symmetry*, vol. 12, no. 12, pp. 1–18, 2020, doi: 10.3390/sym12122018.
- [9] J. Li *et al.*, "A shallow convolutional neural network for apple classification," *IEEE Access*, vol. 8, pp. 111683–111692, 2020, doi: 10.1109/ACCESS.2020.3002882.
 - [10] F. Sultana, A. Sufian, and P. Dutta, "Advancements in image classification using convolutional neural network," *Proceedings - 2018 4th IEEE International Conference on Research in Computational Intelligence and Communication Networks, ICRICIN 2018*, pp. 122–129, 2018, doi: 10.1109/ICRCICIN.2018.8718718.
 - [11] A. G. Howard *et al.*, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, [Online]. Available: <http://arxiv.org/abs/1704.04861>.
 - [12] T. Huang and X. Yuan, "Detection and classification of various image operations using deep learning technology," *Proceedings - International Conference on Machine Learning and Cybernetics*, vol. 1, pp. 50–55, 2018, doi: 10.1109/ICMLC.2018.8526999.
 - [13] H. S. Nagamani and H. Sarojadevi, "Tomato leaf disease detection using deep learning techniques," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 1, pp. 305–311, 2022, doi: 10.14569/IJACSA.2022.0130138.
 - [14] E. K. Nithish, M. Kaushik, P. Prakash, R. Ajay, and S. Veni, "Tomato leaf disease detection using convolutional neural network with data augmentation," *Proceedings of the 5th International Conference on Communication and Electronics Systems, ICCES 2020*, pp. 1125–1132, 2020, doi: 10.1109/ICCES48766.2020.09138030.
 - [15] K. P. Akshai and J. Anitha, "Plant disease classification using deep learning," *2021 3rd International Conference on Signal Processing and Communication, ICPSC 2021*, pp. 407–411, 2021, doi: 10.1109/ICSPC51351.2021.9451696.
 - [16] B. Akshaya and M. T. Kala, "Convolutional neural network based image classification and new class detection," *Proceedings of 2020 IEEE International Conference on Power, Instrumentation, Control and Computing, PICC 2020*, 2020, doi: 10.1109/PICC51425.2020.9362375.
 - [17] U. Mittal, R. Potnuru, and P. Chawla, "Vehicle detection and classification using improved faster region based convolution neural network," *ICRITO 2020 - IEEE 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)*, pp. 511–514, 2020, doi: 10.1109/ICRITO48877.2020.9197805.
 - [18] A. Elhassouny and F. Smarandache, "Smart mobile application to recognize tomato leaf diseases using convolutional neural networks," *Proceedings of 2019 International Conference of Computer Science and Renewable Energies, ICCSRE 2019*, 2019, doi: 10.1109/ICCSRE.2019.8807737.
 - [19] R. S. Samosir, E. Abdurachman, F. L. Gaol, and B. S. Sabarguna, "Brain tumor segmentation using double density dual tree complex wavelet transform combined with convolutional neural network and genetic algorithm," *IAES International Journal of Artificial Intelligence*, vol. 11, no. 4, pp. 1373–1383, 2022, doi: 10.11591/ijai.v11.i4.pp1373-1383.
 - [20] H. Ajra, M. K. Nahar, L. Sarkar, and M. S. Islam, "Disease detection of plant leaf using image processing and CNN with preventive measures," *ETCCE 2020 - International Conference on Emerging Technology in Computing, Communication and Electronics*, 2020, doi: 10.1109/ETCCE51779.2020.9350890.
 - [21] R. Karthik, M. Hariharan, S. Anand, P. Mathikshara, A. Johnson, and R. Menaka, "Attention embedded residual CNN for disease detection in tomato leaves," *Applied Soft Computing Journal*, vol. 86, 2020, doi: 10.1016/j.asoc.2019.105933.
 - [22] M. Sardogan, A. Tuncer, and Y. Ozen, "Plant leaf disease detection and classification based on CNN with LVQ algorithm," *UBMK 2018 - 3rd International Conference on Computer Science and Engineering*, pp. 382–385, 2018, doi: 10.1109/UBMK.2018.8566635.
 - [23] M. A. Al Mamun, D. Z. Karim, S. N. Pinku, and T. A. Bushra, "TLNet: A deep CNN model for prediction of tomato leaf diseases," *ICCIT 2020 - 23rd International Conference on Computer and Information Technology, Proceedings*, 2020, doi: 10.1109/ICCIT51783.2020.9392664.
 - [24] O. M. Khanday, S. Dadvandipour, and M. A. Lone, "Effect of filter sizes on image classification in CNN: A case study on CFIR10 and fashion-MNIST datasets," *IAES International Journal of Artificial Intelligence*, vol. 10, no. 4, pp. 872–878, 2021, doi: 10.11591/ijai.v10.i4.pp872-878.
 - [25] J. Xiao, J. Wang, S. Cao, and B. Li, "Application of a novel and improved VGG-19 network in the detection of workers wearing masks," *Journal of Physics: Conference Series*, vol. 1518, no. 1, 2020, doi: 10.1088/1742-6596/1518/1/012041.
 - [26] J. Faraway, "Data splitting strategies for reducing the effect of model selection on inference," *Computing Science and Statistics*, 1998, [Online]. Available: <http://people.bath.ac.uk/jjf23/papers/interface98.ps>.
 - [27] K. Kumar, J.-P. Li, Z. -, and R. Ahmed, "Content based image retrieval using gray scale weighted average method," *International Journal of Advanced Computer Science and Applications*, vol. 7, no. 1, 2016, doi: 10.14569/ijacsa.2016.070101.
 - [28] S. Roy, A. Mitra, and S. K. Setua, "Color image representation using multivector," *Proceedings - International Conference on Intelligent Systems, Modelling and Simulation, ISMS*, vol. 2015-Septe, pp. 357–363, 2015, doi: 10.1109/ISMS.2014.66.
 - [29] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 4510–4520, 2018, doi: 10.1109/CVPR.2018.00474.
 - [30] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, pp. 1929–1958, 2014.

BIOGRAPHIES OF AUTHORS






Detty Purnamasari    has educational background in Information System for Bachelor Degree, Master Degree, and Information Technology for Doctoral. She is finished her studied at Universitas Gunadarma Jakarta-Indonesia. She is lecturer, researcher, and Management Team for High Performance Computing in Universitas Gunadarma. Her research topic about Information System, Artificial Intelligence, Machine Learning, Deep Learning, Database, Ontology, Semantics. He can be reached via email: detty@staff.gunadarma.ac.id






Koko Bachrudin    has educational background in Informatics for Bachelor Degree, Software Engineering for Master Degree, and he is a PhD student in Doctoral of Information Technology at Universitas Gunadarma. He is a Lecturer in Computer Science Department of Universitas Gunadarma, and Management Team for High Performance Computing in Universitas Gunadarma. The research interest is in computer security, Quality of Service and Internet characteristic, Artificial Intelligence. He is Currently working in Universitas Gunadarma and ddoeth.id. He can be reached via email: koko@staff.gunadarma.ac.id



Dede Herman Suryana    holds a Bachelor of Engineering (S.T.) in Informatics Engineering, is currently pursuing a Master of Engineering (M.T.) in Electrical Engineering in addition to several certificates and professional skills. He currently teaches and is a staff at Gunadarma University. His research areas of interest include Artificial Intelligent, Image Processing, Data Science, Web Developer and Mobile Application. He can be reached via email: hermansuryana@staff.gunadarma.ac.id



Robert    holds a Bachelor of Engineering (S.T.) in Informatics Engineering, is currently pursuing a Master of Engineering (M.T.) in Electrical Engineering in addition to several certificates and professional skills. He currently teaches and is a staff at Gunadarma University. His research areas of interest include Artificial Intelligent, Image Processing, and Data Science. He can be reached via email: robertug@staff.gunadarma.ac.id