ISSN 2278-3091
Volume 9, No.3, May - June 2020
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse24932020.pdf
https://doi.org/10.30534/ijatcse/2020/24932020

Applicability of Various Pre-Trained Deep Convolutional Neural Networks for Pneumonia Classification based on X-Ray Images

Julio Christian Young[1], Alethea Suryadibrata[2]
[1]Universitas Multimedia Nusantara, Indonesia, julio.christian@umn.ac.id
[2]Universitas Multimedia Nusantara, Indonesia, alethea@umn.ac.id

ABSTRACT

Along with the development of machine learning methodologies, in the last few decades, a variety of machine learning techniques began to solve many problems in various fields. Encouraged by the development of computer hardware technology, deep learning has become one of machine learning methods that have become incredibly attractive due to the performance of such methods. By using deep learning, the researchers managed to find solutions to a variety of complex problems that had not yet had an optimal solution. In the field of bioinformatics and computer-aided diagnosis, previous researches show that deep learning can be utilized to predict the structure of protein compounds, detecting lumbar spinal stenosis regions, or other biomedical related prediction problems. In this study, we are experimenting with various pre-trained deep convolutional neural networks (CNN) architectures, namely, densely connected convolutional networks (DenseNet), VGG-16, Inception, and Inception-ResNet, to build another classifier on top of before-mentioned pre-trained models for classify each patient condition based on their chest X-Ray images. In this study, our resulted model needs to able to predict whether a patient's lungs have a normal condition or have contracted either with viral or bacterial pneumonia-related conditions based on the given chest X-Ray image. Our experiment shows that by using a variation of DenseNet, DenseNet201,deep CNN is able to achieve the accuracy scores of 93.87% for the training data and 82.7% for the validation data.

Key words: Deep convolutional neural networks, pneumonia classification, x-ray images

1. INTRODUCTION

In the last few decades, a variety of machine learning techniques began to solve many problems in various fields. Encouraged by the development of computer hardware technology, deep learning has become one of machine learning methods that have become incredibly attractive due to the performance of such methods. The term deep learning was first introduced by Ivakhnenko (1970) through the group method of data handling (GMDH) algorithm as one of the heuristics approaches for solving complex problems. In general, the term deep learning refers to an artificial neural network (NN) architecture that consists of a big number of hidden layers as well as hidden neurons within a network [1].

By using deep learning, the researchers managed to find solutions to a variety of complex problems that had not yet had an optimal solution. For example, researchers [2]developed daily water level forecasting using Whale Optimization Algorithm and NN. In [3] researchers also use NN for crude oil export forecasting. Besides that, in the field of computer vision, NN is used for human body poses recognition [4]. Due to the fascinating performance of deep learning methods, deep learning began to be applied in various fields that require sophisticated and critical solutions. This means that the solution given needs to be very reliable since it involves the critical aspect of a human being, e.g., bioinformatics, computer-aided diagnosis, and computer-aided surgery systems.

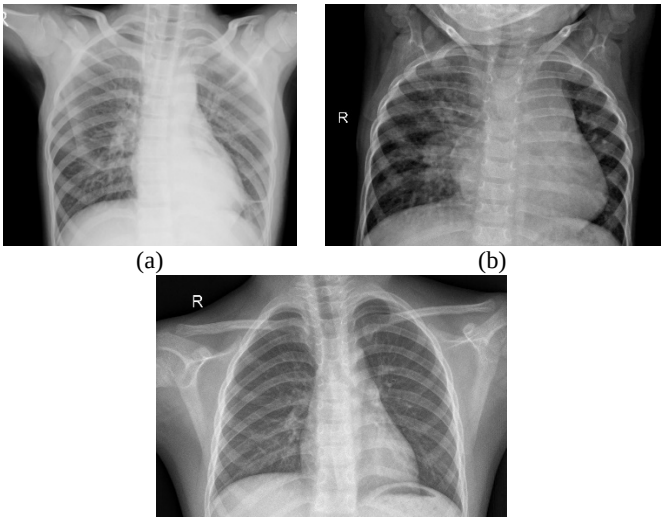
In the field of bioinformatics and computer-aided diagnosis, deep learning can be utilized to predict the structure of protein compounds [5], lumbar spinal stenosis detection system based on MRI images [6, 7], or classification system based on certain patterns in a collection of biomedical images [8, 9]. Previous research by Gulshan et al. [10], shows the implementation of deep CNN (DCNN) for classifying patients' diabetes level based on their retinal eye image. In another study [11], by the use of a similar method, researchers managed to classify the condition of gastric infections based on patient's gastric images.

Besides having good performance, deep learning methods also do not require their users to do the features extraction in its implementation manually. For example, in the implementation of CNN, the convolutional and pooling layers in the networks can be used to capture some image features that characterize each corresponding class in the dataset [12]. However, the main problem of deep learning implementations is that the learning process tends to require a massive amount of computational resources and times [13].

For addressing the mentioned problem above, transfer learning is a mechanism that often used by researchers [12].

Transfer learning is a knowledge transfer mechanism where a model which hadpreviously been trained for a certain task is reused as a starting pointinanother model for a different task. In the case of CNN, transfer learning can be done by the use of convolution and pooling layers that have previously been trained. By using pre-trained convolution and pooling layers, the learning process for a new task can be reduced, so it only focused on the fully-connected layers part.

In this study, we are experimenting with various CNN architectures, namely, densely connected convolutional network (DenseNet) [14], VGG-16 [15], Inception [16] and Inception-ResNet [17], to classify each patient condition based on their chest x-ray images. All of the mentioned architectures above are quite popular and have a fascinating performance for image recognition and localization tasks. Each architecture used in our experiment has been trained for a Large Scale Visual Recognition Challenge 2017.The dataset used in this research consists of Chest X-ray images (anterior-posterior) that has been selected from retrospective cohorts of paediatric patients with one to five years long treatment in Guangzhou Women and Children's Medical Center. All chest X-ray imaging was obtained from patients' routine clinical care in the hospital. The dataset consists of 3 categories, namely, normal, viral, and bacterial pneumonia, where each category has 1538, 1178, and 2780 images, respectively. Figure 1, 2, and 3respectively show X-ray image of viral pneumonia, bacterial pneumonia, and normal lung condition in the provided dataset.



(a) (b)

(c)

Figure 1:(a) Viral infection case (b) Bacterial infection case (c) Normal lung condition

As seen in Figure 1c, normal chest X-ray depicts clear lungs without any areas without abnormal opacification within, while in pneumonia case (Figure 1b), chest X-ray depicts some area with more radio-opaque characteristics in the lungs. Moreover, for viral pneumonia case (Figure 1a), both lungs have more dispersed opacity, while in bacterial pneumonia case, the opacity is more noticeable in one part of the lungs.

2. FUNDAMENTALS

2.1 Convolutional Neural Network

The first idea of Convolutional neural networks (CNN) was introduced by Zhang [18] under the name Shift invariant artificial neural networks (SIANN). CNN is an implementation variant of deep feed-forward artificial neural network that often used to analyze images data. As a deep learning method, the idea of CNN is mimicking the connectivity of neurons in the visual cortex of the animal brain. Each neuron in the cortex only responds to stimuli in certain regions of the visual field known as receptive fields. The receptive part of each neuron has several parts that partially overlap and cover all visual fields seen by the eye [19]. CNN is generally composed of convolutional, activation, pooling, and fully-connected layers.

2.2 Convolutional Layers

The principal idea of the convolutional layers is to obtain a collection of features based on an input volume. In conventional feed-forward NN architecture, Multilayer Perceptron (MLP), there are several drawbacks for using it to solve image processing tasks.The first one is the amount of parameters within the model can rapidly become unmanageable as the input image become larger. As an example, for a 299 * 299colorful image (3 color channel), each neuron in the first hidden layer have around 260000 weights that need to be adjusted during the training process. Moreover, MLP is unable to processing objects' spatial information within an image as MLP reacts differently to an image and its shifted version.

For tackling this problem, instead of directly processing the input image using MLP ,in CNN architecture, the input image is processed first through a stack of convolutional and pooling layers for the image features extraction process. Afterwards, the extracted features will be used as an input for a classification task in MLP. In CNN, each convolutional layer works as receptive fields which represented as a collection of m filters, where each filter is represented by$n * n$ matrix. Within a convolutional layer, each filter will move across the entire image from the top-left to bottom-right, producing a matrix of features (features map) from the input. Each value within a features map is a dot product of n*n matrix with the window that slide across the input image.

2.3 Pooling Layers

A pooling layer is another compositional part of a CNN and mostly placed after the convolutional layer in CNN architecture. A pooling layer runs on each feature map separately and partitions it into a collection of non-overlapping rectangles to reduce the spatial size of the features map. Max and average pooling layers are the two most popular version of pooling layers which often used in

CNN, though in practice, the max pooling layers usually work better. The idea max pooling is taking the largest element within a spatial neighbourhood (e.g., 3x3 window) from the rectified feature maps produced by the previous layer. By using a similar idea, the average pooling layers work by taking the average from the rectified feature map.

2.4 Fully-Connected Layers

The fully-connected (FC) layer consists of a collection of neurons at the end of CNN. Similarly to regular Multilayer Perceptron, each neuron in an FC layer is connected to all activations in the preceding layer. As the purpose of convolutional and pooling layers is to capture useful features from the input image, the purpose of the FC layers is to use the produced features for classifying it into its corresponding target. The last layer in the FC layers part (the output layer) often uses the softmax function for producing a vector of values within zero to one that sum to one.

2.5 Dropout Layers

Dropout layers are functioning as regularization in neural networks which reduce co-dependency amongst each neuron during the training phase. It can lead a model to overfit of training data. Dropout layers work by shutting down part of neural networks during a particular forward and backward phases. By doing so, dropout layers might benefit a neuron in a neural network to discover more robust features that are beneficial in conjunction with many different arbitrary subsets of the other neurons.

2.6 ImageNet Pretrained CNN

ImageNet is an ongoing research effort that consists of more than fourteen millions of manually human-annotated and quality-controlled images into almost 22,000 separate object categories. The purpose of the ImageNet project is to be an easily accessible image database to all researchers around the world. Every year, from 2010 to 2017, based on the ImageNet dataset, the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) is held to evaluates algorithms for high performance andlarge scale object detection and image classification. Based on ILSVRC, the following section will show some CNN architecture that works best and dominating the challenge.

2.7 VGG-16

VGG-16 is one from famous deep convolutional neural network submitted to ILSVRC-2014 [15]. VGG-16 is invented by Visual Geometry Group (VGG) from University of Oxford. VGG-16 take an input of an RGB image with a fixed size of 224x224. In VGG-16, the input image is processed through a stack of convolutional layers with a relatively small receptive field: 3x3. By using small receptive fields, VGG-16 has fewer parameters compared to another CNN architectures like AlexNet and ZFNet. With fewer parameters to be learnt, VGG-16 is faster to converge and reduced the overfitting problem. By using smaller receptive fields, VGG-16 designed to have deeper convolutional layers configuration. Another research by Goodfellow at el. (2014) shows that deep ConvNets led to better performance.

2.8 DenseNet

Densely connected convolutional networks (DenseNet) is one of deep CNN developed by Huang, Liu, and van der Maaten [14]. Unlike a standard CNN, in DenseNet, the input image goes through multiple convolutional layers and each convolutional layer passes on its feature maps to all subsequent layers by the use of channel-wise concatenation. The idea of doing so is each layer receiving a “collective knowledge” from all preceding layers. As each layer receives extractedfeatures from all preceding layers, the designed network architecture can be leaner and compact, thus lead to a higher computational and memory efficiency. There are several implementations of DenseNet (e.g., DensetNet121, DenseNet169,and DenseNet201), where the number for each implementation denotes the depth of the ImageNet models. All implementations of DenseNet are taking an input image with a width and height of 224 and 3 color channels.

2.9 Inception

There are several versions of Inception. In this research, the term Inception is referring to Inception v3 that developed by Szegedy et al. [16]. Similarly to VGG16, the idea of Inception is to create a more in-depth neural network architecture. However, as a deeper model requires more data to prevent it from overfitting and requires more computational resources, Inception work by replacing fully connected architectures inside of convolutional layers with sparsely connected architectures. By doing so, Inception has a very depth convolutional layers configuration with less number of parameters (requiring less computational resources) compared to other CNN architectures. For example, compared to VGG-16 with the depth of 23 that has 138,357,544 parameters, Inception with the depth of 159 only consists of 23,851,784 parameters. Inception takes an input image with the size of 299x299.

2.10 Inception-ResNet

Inception-ResNet is a modification of Inception v3 model which uses some principal techniques in ResNet implementation. By implementing the concept of residual connections in ResNet, Szegedy et al. [17] able to create deeper neural networks (Inception-ResNet convolutional part has 572 layers with 55,873,736 parameters) which lead to even better performance for ILSVRC image classification benchmark. Inception Res-Net takes an input with the size of 299x299.

3. RESEARCH METHODOLOGY

In this research, patients' conditions already separated into three classes, which are normal, and bacterial or viral pneumonia infections. All of the data is randomly divided into a ratio of 70:30. 70% of the data is used by the machine to study the patterns of each class, while the remaining data will be used to validate the learning process that has been carried out by the machine. For each CNN architecture (DenseNet121, DensetNet169, DensetNet201, VGG-16, and Inception-ResNet) which will be used as a feature extractor, the learning and validation process will be carried out alternately using the same data.

A scaling process is carried out when the image has a different size to adjusts the number of inputs that can be received by each architecture. For DenseNet121, DenseNet169, DenseNet201, and VGG-16 architectures, the images will be scaled to 224 * 224, whereas for Inception-ResNet architectures, the images will be scaled to 299 * 299. The result of the scaling process will be used as an input to each of the models that have been trained to solve the problem of object detection and localization in the ILSVRC2017 competition.

Based on the output of a pre-trained model, the flattening process will be carried out for transforming the outputted matrices into a 1-dimensional vector. To create a new classifier for the pneumonia classification task, we add fully-connected layers on top of a pre-trained model. In this research, the input layer of the fully-connected layers consists of a number of neurons equal to the length of the flattened image features, while the output layer consists of three neurons where each neuron represents each label in the dataset. Between the input and output layer, we add one hidden layer which consists of 250 neurons and one dropout layer.

We decide to use 250 neurons in the hidden layer as on our preliminary experimentation involving a pre-trained model with high dimensional output features (DenseNet201 and Inception-ResNet). The output shows that there is no significant impact as we add more neurons in the hidden layer. The dropout rate in the designed model is set to 0.5 to prevent itto overfitting to the training data. Moreover, every neuron in the hidden layer will use the rectified linear unit (ReLU) with the definition of $f(x) = \max(0, x)$ and every neuron in the output layer will use softmax function with the definition of:

$$f(x_i) = \frac{\exp(x_i)}{\sum_{j=0}^{k} \exp(x_j)} \; untuk\; i = 0, 1, 2, \dots, k \; \#(1)$$

ReLU was chosen to introduce nonlinearity in the designed model and generally make the learning process faster than traditional function such as hyperbolic tan or sigmoidal function. On the other hand, softmax function is used as its capability to normalize a vector of *K* number into a probability distribution consisting of *K* probabilities. In our case, each value in a vector output represents a probability of each target class.

Finally, to train the designed model, the epoch and batch size is set to 20 and 32 respectively; and root mean square propagation (RMSprop) is used as an optimizer in the training process. RMSprop is a gradient descent algorithm with momentum so our model may converge to the optimal solution faster compared to the traditional gradient descent algorithm. We use default Keras implementation for the RMSprop. The learning rate is set to 10^{-3} and rho to 0.9. For further detail about RMSprop reader can refer to [20].

4. EXPERIMENT

Based on the previously described research methodology, Table 1 shows our experiment results.

Table 1: Experiment Results

Architecture	Training Accuracy	Validation Accuracy
DenseNet121	92.5%	81.21%
DenseNet169	93.07%	82.70%
DenseNet201	93.87%	82.70%
VGG-16	85.75%	80.33%
Inception	47.7%	47.7%
Inception-ResNet	72.84%	70.59%

For the convenience of the reader, we visualize the experiment results in Figure 2.

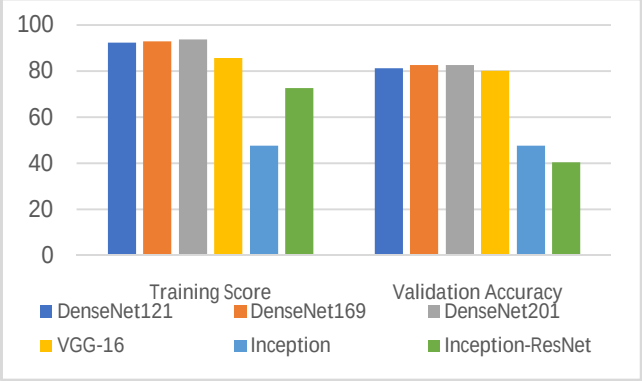


Figure 2: Visualization of our experiment results.

Figure 2 shows that a designed architecture by using DenseNet201 as a feature extractor has the best accuracy scores for both training and validation data. Following the previous experiment result, we decided to double up the number of epoch in the DenseNet201 architecture settings. We got a better accuracy score of 93.87% for the training data and the same accuracy for the validation part. Moreover, following the two promising results, we triple up the number of epoch to 60 to see if there is any accuracy improvement for the training and testing data. However, the resulting model shows the same performance as a model that has been trained for 40 epoch.

5. CONCLUSION

Based on our experiment, we got the best accuracy score for the testing data by 82.7% and the training data by 93.87% using DenseNet201 as a feature extractor. However, compared to VGG-16, as the number of produced features in DenseNet201 is bigger than VGG-16, the learning process of DenseNet201 involving more learnable parameters thus lead to more training time within an epoch. Furthermore, though Inception-ResNet and DenseNet201 are producing a similar amount of features, in our case, Inception-Resnet based model is not achieving classification accuracy as best as DensetNet201. Therefore, as VGG-16 based model with a smaller amount of features producing better accuracy score, it can be concluded that the number of produced features produced might not affecting the performance of a new pre-trained based model. Finally, by using pre-trained CNN as feature extractor, we can build a classifier which might have relatively same performance as fully-trained CNN with significantly lower computation resources and times.

6. FUTURE WORKS

Another research in [21] shows that combining several pre-trained models can create a better classifier rather than using a single pre-trained model for the music classification task. Therefore, the next experiment that worth to try is combining several pre-trained models as a feature extractor and evaluating the performance gains through such combination. However, combining several pre-trained models requiring more computational power and memory as loading several pre-trained models requiring more GPU memory. Moreover, as several pre-trained models produce more high dimensional features, we might need to add more neurons in the fully convolutional layers of our designed CNN so it can completely utilize before-mentioned features.

On the other hand, we can also combine several flattened image features from each pre-trained models as a single vector and applying a dimensionality reduction technique (DR) such as Principal Component Analysis [23] or Uniform Manifold Approximation and Projection [22], etc for lowering the combined features' dimension. By doing so, the number of neurons in the hidden layers might not needed to be adjusted as the reduced features might have significantly lower dimension compared to the un-preprocessed features .Even though the reduced features have a smaller vector size, some previous studies show that using DR in preprocessing steps will not reduce the classification model performances or in some cases even increased such performances[24,25].

REFERENCES

1. Goodfellow, I., Bengio, Y., & Courville, A. (2016). **Deep Learning**. MIT Press.
2. Louis, et al. (2019). **Development of Whale Optimization Neural Network for Daily Water Level Forecasting**In *International Journal of Advanced Trends in Computer Science and Engineering (IJATCSE)*, Vol. 8 No. 3, pp. 354-362. https://doi.org/10.30534/ijatcse/2019/04832019
3. Alam, T. (2019), **Crude Oil Export: A Comparative Analysis using Artificial Neural Network and ARIMA** in *International Journal of Advanced Trends in Computer Science and Engineering (IJATCSE)*, Vol. 8, No. 5, pp. 2546-2550 https://doi.org/10.30534/ijatcse/2019/102852019
4. Al-Qerem, A., &Alahmad, A. (2019).**Human Body Poses Recognition Using Neural Networks with Data Augmentation***in International Journal of Advanced Trends in Computer Science and Engineering (IJATCSE)*, Vol. 8 No. 5, pp. 2117-2120 https://doi.org/10.30534/ijatcse/2019/40852019
5. L. Bai dan L. Yang. (2017). **A Unified Deep Learning Model for Protein Structure Prediction** in *3rd IEEE International Conference on Cybernetics (CYBCONF)*, Exeter, pp. 1-6.
6. Kafri, A. A. A., et al. (2018).**Segmentation of lumbar spine MRI images for stenosis detection using patch-based pixel classification neural network.** 2018 IEEE Congress on Evolutionary Computation (CEC), pp. 1-8. https://doi.org/10.1109/CEC.2018.8477893
7. Kafri, A. A. A., et al. (2019). **Boundary delineation of MRI images for lumbar spinal stenosis detection through semantic segmentation using deep neural networks**in IEEE Access, Vol. 7, pp. 43487-43501.
8. Plis, S. M., Hjelm, D. R., Salakhutdinov, R., et al. **Deep learning for neuroimaging: a validation study**. Frontiers in neuroscience 2014;8. https://doi.org/10.3389/fnins.2014.00229
9. Shin, H. C., Roth, H. R., Gao, M., Lu, L., Xu, Z., Nogues, I., ... & Summers, R. M. (2016). **Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning.** IEEE transactions on medical imaging, 35(5), 1285-1298.
10. Gulshan, V., Peng, L., Coram, M., Stumpe, M. C., Wu, D., Narayanaswamy, A., ... & Kim, R. (2016). **Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs**. Jama, 316(22), 2402-2410.
11. Zhu, R., Zhang, R., &Xue, D. (2015, October). **Lesion detection of endoscopy images based on convolutional neural network features.** In 2015 8th International Congress on Image and Signal Processing (CISP) (pp. 372-376). IEEE. https://doi.org/10.1109/CISP.2015.7407907
12. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., ... & Darrell, T. (2014, November). **Caffe: Convolutional architecture for fast feature embedding.** In Proceedings of the 22nd ACM international conference on Multimedia (pp. 675-678). ACM.
13. Sze, V., Chen, Y. H., Einer, J., Suleiman, A., & Zhang, Z. (2017, April). **Hardware for machine learning:**

Challenges and opportunities. In Custom Integrated Circuits Conference (CICC), 2017 IEEE (pp. 1-8). IEEE. https://doi.org/10.1109/CICC.2017.7993626

14. Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). **Densely connected convolutional networks**. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4700-4708).
15. Simonyan, K., & Zisserman, A. (2014). **Very deep convolutional networks for large-scale image recognition.** arXiv preprint arXiv:1409.1556.
16. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., &Wojna, Z. (2016). **Rethinking the inception architecture for computer vision**. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2818-2826). https://doi.org/10.1109/CVPR.2016.308
17. Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. A. (2017, February). **Inception-v4, inception-resnet and the impact of residual connections on learning**. In Thirty-first AAAI conference on artificial intelligence.
18. Zhang, W. (1988). **Shift-invariant pattern recognition neural network and its optical architecture** in Proceedings of annual conference of the Japan Scoeity of Applied Physics.
19. Matsugu, M., Mori, K., Mitari, Y., & Kaneda, Y. (2003). **Subject independent facial expression recognition with robust face detection using a convolutional neural network.** Neural Networks, 16(5-6), 555-559.
20. Hinton, G., Srivastava, N., &Swersky, K. (2012). **Neural networks for machine learning lecture 6a overview of mini-batch gradient descent.** Cited on, 14(8).
21. Lee, J., & Nam, J. (2017). **Multi-level and multi-scale feature aggregation using pretrained convolutional neural networks for music auto-tagging.** IEEE signal processing letters, 24(8), 1208-1212. https://doi.org/10.1109/LSP.2017.2713830
22. McInnes, L., Healy, J., Saul, N., &Groberger, L. (2018). **UMAP: Uniform Manifold Approximation and Projection**. *Journal of Open Source Software*, *3*(29), 861.
23. Jolliffe, I.T., 2002. **Principal Component Analysis**, second edition, New York: Springer-Verlag New York, Inc.
24. Fu, X., & Wang, L. (2003). **Data dimensionality reduction with application to simplifying RBF network structure and improving classification performance**. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, *33*(3), 399-409. https://doi.org/10.1109/TSMCB.2003.810911
25. Nasution, M. Z. F., Sitompul, O. S., & Ramli, M. (2018, March). **PCA based feature reduction to improve the accuracy of decision tree C4. 5 classification**. In *Journal of Physics: Conference Series* (Vol. 978, No. 1, p. 012058). IOP Publishing. https://doi.org/10.1088/1742-6596/978/1/012058