



# Classification Analysis of Back Propagation-Optimized CNN Performance in Image Processing

Putrama Alkhairi<sup>1</sup>, Agus Perdana Windarto<sup>2\*</sup>

<sup>1,2</sup>Information Systems, STIKOM Tunas Bangsa, Pematang Siantar, Indonesia  
Email: putrama@amiktunasbangsa.ac.id<sup>1</sup>, <sup>2\*</sup>agus.perdana@amiktunasbangsa.ac.id

## Abstract

This study aims to optimize the performance of the Convolutional Neural Network (CNN) in the image classification task by applying data augmentation and fine-tuning techniques to a case study of mammal classification. In this study, we took a fairly complex image classification dataset and used the CNN model as a basis for training and evaluating the performance of the model compared to Back propagation. From this study, the CNN VGG16 architecture optimized with ADAM optimization has been compared with the Back propagation optimization of SGD. We also conducted a literature review on several related studies and basic concepts in CNN, such as convolution, pooling, and fully connected layers. The research methodology involves creating datasets using data augmentation techniques, model training using fine-tuning techniques, and testing model performance using a number of evaluation metrics, including accuracy, precision, and recall. The results of this study indicate that the techniques used have succeeded in improving the performance of the CNN model in complex image classification tasks with accuracy in identifying and monitoring animal species more accurately, with an accuracy of 91.18% for the best model. Model accuracy increased by 2% after applying data augmentation and fine-tuning techniques to the CNN model. These results indicate that the techniques applied in this study can be a good alternative in improving the performance of the CNN model in the image classification task.

Keywords: CNN, VGG16 Architecture, Back propagation, Optimization, Classification, Animals

## 1. Introduction

Image processing has become an increasingly important research topic in recent years. However, the use of the neural convolution model (CNN) for image classification often faces challenges, such as image blur or intensity variations. Convolutional neural networks (CNNs) have been widely used for classification and image recognition. Convolutional Neural Network (CNN) is a neural network model that is very effective in image processing and pattern recognition [1], [2]. CNN consists of several layers, each of which has a specific function in image processing, such as convolution, pooling, and fully-connected layers. For better accuracy, CNN requires both intensive and extensive computations. For real-time processing, CNN is usually accelerated by a parallel processor such as a graphics processing unit (GPU). Although GPUs speed up computing, the substantial increase in power limits their applicability to embedded systems. For low power and high performance digital systems, field programmable gate arrays (FPGA) and application specific integrated circuits (ASIC) have been used for CNN accelerators in recent years [3], [4].

The increasing use of Convolutional Neural Network (CNN) models in creating computer vision models makes this model architecture a natural focus for performance optimization efforts. Likewise, optimization with the use of optimization techniques involving CNN hidden layers. The inference performance of the Standard VGG16 architecture CNN model leaves significant room for improvement [5], [6]. With an important machine learning approach inspired by the nervous system in the human mind. It involves an input layer, a hidden layer, and an output layer, and aims to adjust the optimization type in relation to the weight of each neuron in the ANN following the training process. The performance of the CNN structure is strongly influenced by the amount and variety of training and optimization data used [7], [8].

To test this optimization model there needs to be a comparison. In this comparison with backpropagation, where optimization is done on CNN performance for image classification is usually done through techniques such as fine-tuning and data augmentation [9], [10]. Fine-tuning is done by utilizing previously trained weights on different datasets to be applied to a new image classification dataset, while data augmentation is done by enlarging the dataset used to train the model by adding small variations to the training data [11], [12].

The main difference between optimization techniques such as backpropagation and techniques used in CNN performance for image classification is in their purpose. The purpose of backpropagation is to determine the most optimal weight by minimizing the loss function defined in the model. Meanwhile, the techniques used for CNN performance aim to improve model performance on specific image classification datasets, by utilizing techniques such as fine-tuning and data augmentation to improve model accuracy and generalization [13], [14].

Therefore, doing the CNN architecture VGG16 model for efficient performance and selecting the right optimizer is also very important in CNN and deserves to be researched and tested. There are various optimizers that can be used, such as Adam and SGD. Choosing the right optimizer can improve model performance and speed up training [15], [16].

In the previous research conducted by "Tian Yuan, Weiqiang Liu et al [17]" entitled "High Performance CNN Accelerators Based on Hardware and Algorithm Co-Optimization" resulted in a hardware-oriented CNN compression strategy, it was proposed that the deep neural network (DNN) model be divided into "nopruning layer (NP-layers)" and "pruning layers (P-layers)". The NP layer has regular weight distribution for parallel computation and high performance. The P-layers are irregular due to trimming, but produce a high compression ratio. Uniform and incremental quantization schemes are used to achieve a tradeoff between compression ratio and processing efficiency at small loss of accuracy. For a hardware accelerator on a single FPGA chip without using off-chip memory, the 27.5× compression ratio is achieved with a top-5 accuracy of 0.44%.

Whereas the results of research conducted by "Desi Irfan et al [18]" "Comparison Of SGD, Rmsprop, And Adam Optimization In Animal Classification Using CNNs" show that the CNN method with the Adam optimization function produces the highest accuracy compared to the SGD optimization method and RMSprop. The model trained using the Adam optimization function achieved an accuracy of 89.81% on the test, demonstrating the feasibility of the approach [19]-[20].

Thus, the results of the analysis of previous research and the results of the author's analysis, the author created the CNN model of the VGG16 architecture to streamline performance with the Adam optimizer to improve model performance and speed up training. For this reason, this model was tested by comparing it with the backpropagation algorithm model with the SGD optimization function. With a case study of the classification of mammals.

## 2. Method

In this study a mammal species classification system was designed to determine accuracy using digital image processing methods. Figure 1 shows the block diagram of the system designed in this study [21], [22].



**Figure 1.** General System Block Diagram

In general, the systematic system block diagram shown in Figure 1 is Animal image collection, preprocessing with resizing and data augmentation stages, Training, and Testing

### 2.1 Image Collection

The data used in this research is secondary data. According to the Big Indonesian Dictionary, animal data is a synonym for animal or animal. According to the World Wildlife Fund, many rare animals such as Sumatran elephants, Asian elephants, African elephants, blue whales, hawksbill turtles, orangutans, Javan rhinoceros, dugongs, hippos, turtles, polar bears, penguins and many others. The data is sourced from Kaggle. The reason the author takes data from Kaggle is because of the reliability of the dataset that has been tested. Keep in mind, using high-resolution training images can also be used to get better accuracy [23].



**Figure 2.** Animal Pictures

### 2.2 Pengamatan Awal

Results To have a foundation that we can build on a comparison of how well our model is prohibited from doing something naturally, we used pre-trained VGG-19 with the following structure:

**Table 1.** Hyperparameter Model CNN

| No | Layer                           | Output Shape  |
|----|---------------------------------|---------------|
| 1  | Batch Size                      | 128           |
| 2  | Crop size                       | 64            |
| 3  | Input Layer                     | 3 x 64 x 64   |
| 4  | nn.model                        | 64 x 4 x 4    |
| 5  | Global Average pooling 2d layer | 3, 8, 3, 1, 1 |
| 6  | Dropout                         | 10%           |

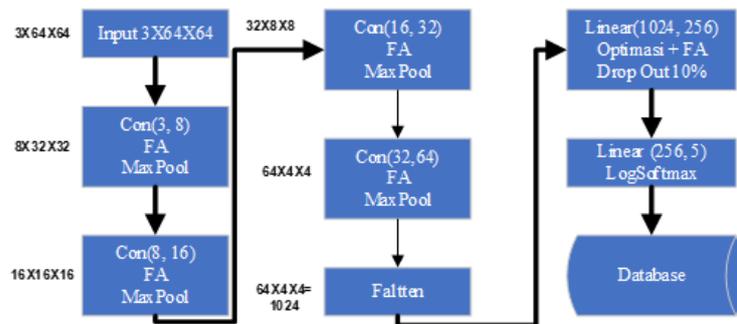
In all the experiments we've done, the only change we've made is to replace the Optimization Function with Adam, RMSProp, and SGD with the appropriate Learning Rate.

### 2.3 Preprocessing.

After the image collection process, pre-processing is carried out to optimize image quality, as well as facilitate and enhance the system's ability to identify objects. Pre-processing augmentation is done by resizing and augmenting data [24].

### 2.4 Training.

At the training stage, a learning process is carried out on the image which then produces a model output that will be stored for use in the testing process. Modeling is the process of training and training image data in identifying objects and categorizing them according to their class [25],[26].



**Figure 3.** The Flowchart of the Stages of the Training System refers to VGG16

In this study the method refers to the very popular VGG16 architecture and has been tested using 2 layers shown in Figure 3. The input size is 64 x 64 x 3. In the first convolution using a total of 10 kernels with a 3x3 matrix with a valid padding value, ReLu activation used in this convolution process as non-Linearity. The pooling process, especially max pooling, uses a size of 2x2 and in the second convolution stage uses a total of 20 kernels with a 5x5 matrix, uses ReLU with a valid padding value. Next, flatten is to change the output of the convolution process in the form of a matrix into a vector which will be forwarded to the classification process using MLP (MultiLayer Perceptron) with a predetermined number of neurons in the hidden layer. In SGD Optimization and Adam will be applied to nodes for weight optimization and bias with the default Learning rate using the softmax activation function according to the number of classes, in this study there were 5 classes of neurons. The image class is then classified based on the value of neurons in the hidden layer using the softmax activation function.

### 2.5 Testing.

Fig.4 shows the flowchart of the system testing phase. The testing phase is the process of classifying animal species by testing the test image data and comparing it with the results of the training model from the training image data stored in the database. Image data was taken as much as 1000 for the original data then 3000 augmentation image data. The image that has been taken will be processed by the CNN algorithm to produce system output in the form of Animal Class information [27], [28].

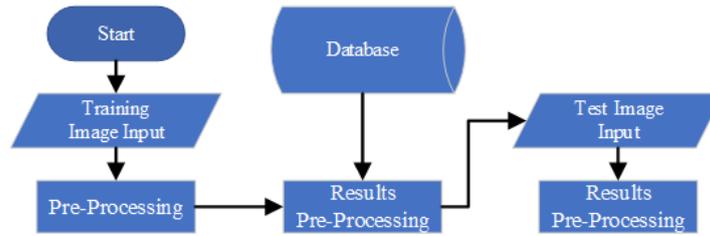


Figure 4. Flowchart of System Testing Stage

2.6 Testing.

Gambar 5 adalah proses Backpropagation yang digunakan untuk memperbarui bobot dalam jaringan saraf [29].

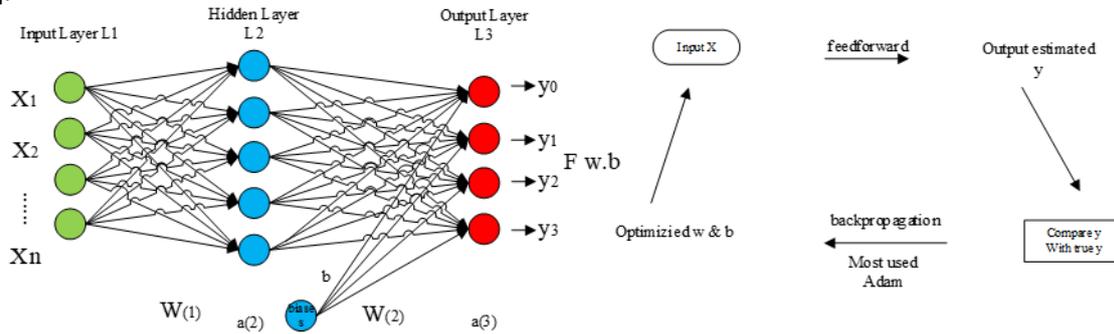


Figure 5. Weight and Bias Optimization Process

In Figure 5 it can be seen that the gradient of the parameter model is sampled iteratively, behind the direction of the network weights, to find a new weight that minimizes the error value in terms of classification. The first optimizer proposed is Adaptive Moment Estimation (Adam), where at every iteration  $t$ , Adam updates the model parameters as follows [28], [30]:

$$\begin{aligned}
 m(t) &= \beta_1 * m(t-1) + (1-\beta_1) * \nabla f(w(t-1), x(t)) \\
 v(t) &= \beta_2 * v(t-1) + (1-\beta_2) * (\nabla f(w(t-1), x(t)))^2 \\
 w(t+1) &= w(t) - \alpha * m(t) / (\sqrt{v(t)} + \epsilon)
 \end{aligned}$$

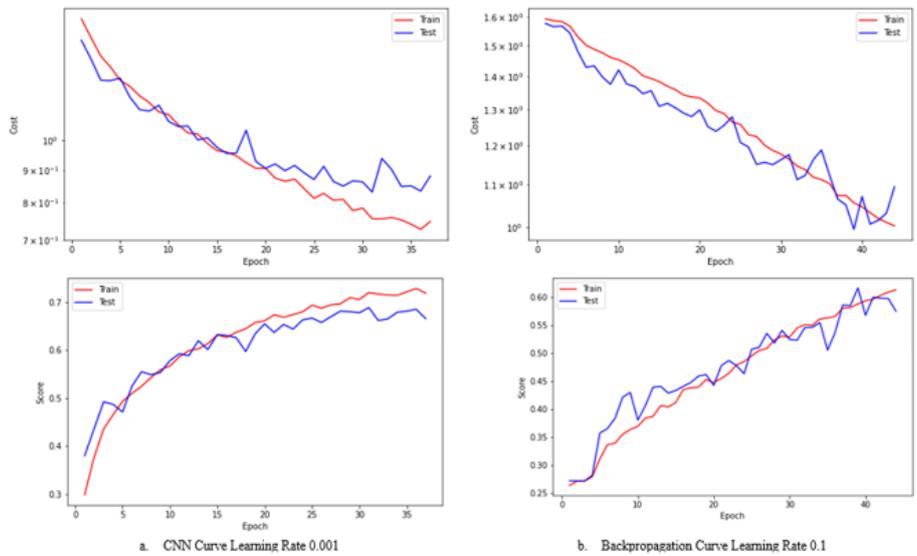
where  $m(t)$  and  $v(t)$  are the first and second moments of the gradient at iteration  $t$ ,  $\beta_1$  and  $\beta_2$  are the moment coefficients,  $\epsilon$  is a small number to prevent division by zero, and  $\nabla f(w(t-1), x(t))$  is the gradient of the objective function  $f$  at point  $w(t-1)$  for the training data sample  $x(t)$ .

3. Result and Discussion

Experiments on systems that have been designed using the CNN method with architecture referring to the VGG16 architecture and compared with backpropagation in image processing in determining the type of animal from the dataset are divided into five classes, namely cats, bulls, elephants, horses and sheep with consideration. The test system is formed by utilizing hyperparameter changes in the data before augmentation and after augmentation. The hyperparameters used are changes to the Optimizer type, namely Adam for the CNN optimization model and SGD for the backpropagation optimization model, changes to the batch size, namely 128, changes to the best learning rate value in each model, namely 0.001 for the CNN VGG16 architecture, 0.1 for the backpropagation model and the number of training iterations. (epochs) in this case using an early stop.

3.1 Data Testing dan Analysis

The first data to be tested is the original data of 9875 images. In the training process, the data used was 0.8 out of a total of 8138 data. While in the process of testing the data used amounted to 20% of the total data or 1737 data. This test uses three optimizers, namely Adam for CNN with a learning rate of 0.001 and SGD with a backpropagation learning rate of 0.1 with parameters, namely batch size 128, for Epoch here using an early stop, where when the model cannot get accuracy and loses again, the process automatically stops.

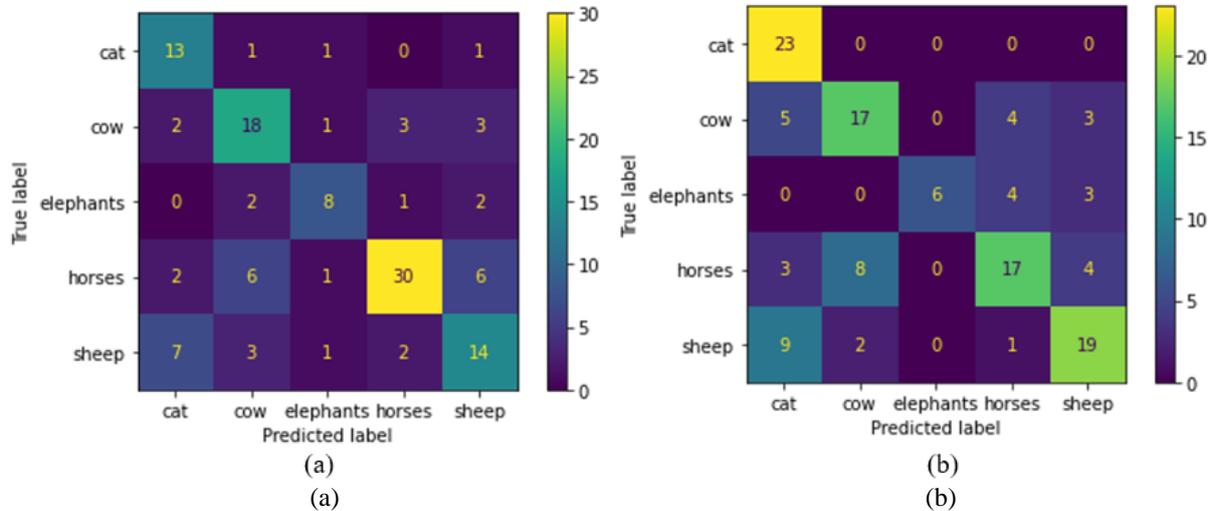


**Figure 6.** Graph of CNN Training and Testing Optimization and Backpropagation

In Figure 6 it can be seen that the graph of the results of the training and testing process learning rate data is 0.001 CNN and 0.1 for backpropagation is good, the red graph shows the training process and the blue graph shows the Testing process touching consistently which indicates that the model is working not overfit.

### 3.2. Confusion Matrix

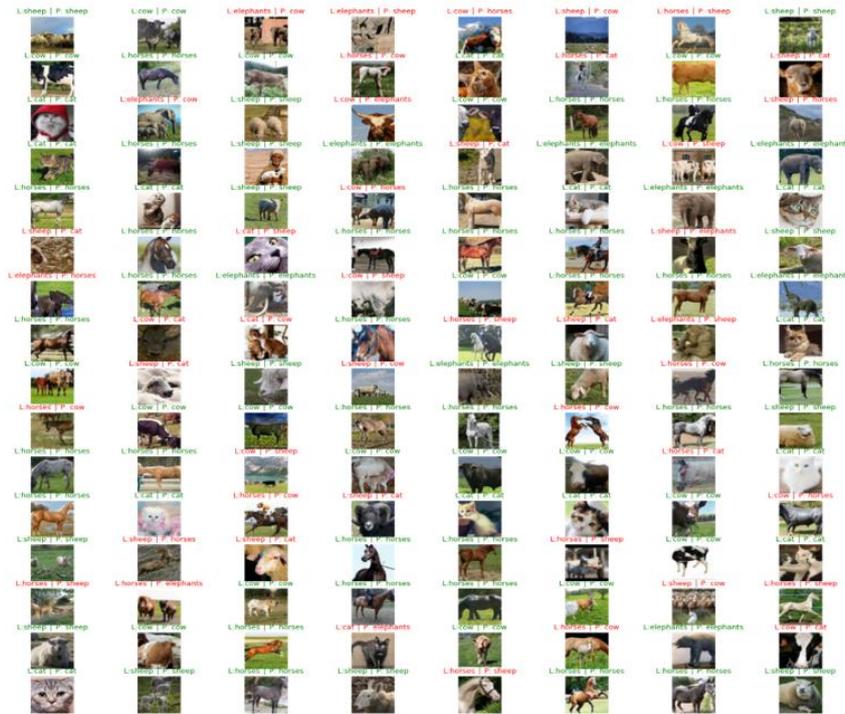
Figure 7 shows a plot of the confusion matrix results from the training process using two optimization algorithms. On the left side of the plot are the True Labels of the 5 animal classes which are the actualization of the actual animal classes and at the bottom are the Predicted labels which are predictions of the training process.



**Figure 7.** Confusion Matrix ADAM for CNN and SGD for Backpropagation

### 3.3. Visualizes data

To see how the prediction visualization results from the built model will be displayed in the form of an image. As explained in the previous explanation that the image data taken was 128 images. To set the display, dimensions of 5 rows and 8 columns are used which only display 40 images.



**Figure 8.** Predicted image visualization using CNN with Adam Optimizer

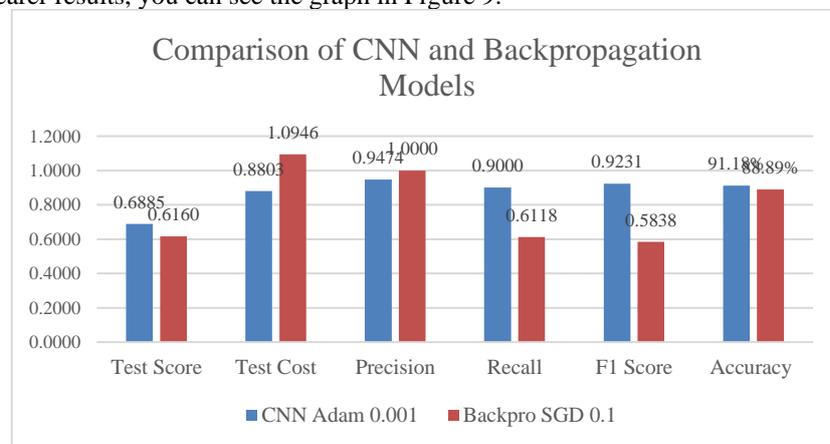
From Figure 8 it can be seen that the actual label will be compared with the predicted label where if the actual label and predicted label show the same class, it will be green, in other words True Positive. And if the actual label and predicted label show a different class, it will be a False Positive in red. Prediction errors can occur due to image factors which can be caused by several factors such as test data in this case in the form of unclear images, background influences such as properties, similarity of colors and shapes, and so on. As in the first row fourth column example, there is a decapitated tiger, the larger machine is represented by an elephant. This can be overcome by adding a tiger image without a head to the training data.

Improvements in the accuracy and learning speed values can be seen in table 2. Comparison of Apple to Apple, using the same dataset. In table 2 below are the test results of the two models.

**Table 2.** Comparison of two Adaptive Learning Rate Models

| Algoritma | Optimasi | LR    | Test Score | Test Cost | Best Epoch | Precision | Recall | F1 Score | Accuracy |
|-----------|----------|-------|------------|-----------|------------|-----------|--------|----------|----------|
| CNN       | Adam     | 0.001 | 0,6885     | 0,8803    | 37         | 0,9474    | 0,9000 | 0,9231   | 91,18%   |
| Backpro   | SGD      | 0.1   | 0,6160     | 1,0946    | 44         | 1,0000    | 0,6118 | 0,5838   | 88,89%   |

In table 2, the results of testing the animal image dataset above can be seen in the model which shows an increase in accuracy in the CNN architecture VGG16 model with Adam optimization, a significant increase of around 2%. For clearer results, you can see the graph in Figure 9.



**Figure 9.** Comparison of the Optimized CNN model and standard Backpro

In Figure 9. It can be seen that the problem with the speed of the backpropagation learning process and the very long time for convergence can be seen in the test cost and minimum local problems that make artificial neural networks (ANN) often stuck at the local minimum have started to be resolved, and the accuracy value is improved by this model can be applied. In the model proposed by the author in improving the accuracy value obtained by using Adam's optimization with a learning rate of 0.001 with the highest accuracy value of 91.18%.

#### 4. Conclusion

In this study, we can recommend a solution model to help future researchers or writers identify and monitor animal species more accurately, with an accuracy of 91.18% for the best model. The solution offered is that it can assist in monitoring animal species, especially those that are protected in the future, cheaper, faster and more reliable. This research also proves that with an appropriate learning rate in each estimation function, Adam is superior to SGD. Suggestions for further research are to compare the effect of the activation function with the same dataset and optimizer in this study to further improve the performance of the previous model.

#### Acknowledgements

Thank you to Mr. Agus Perdana Windarto for his support in carrying out this research and for his guidance and continued support.

#### References

- [1] A. A. SG. Mas Karunia Maharani, Komang Oka Saputra, and Ni Made Ary Esta Dewi Wirastuti, "Komparasi Metode Backpropagation Neural Network dan Convolutional Neural Network Pada Pengenalan Pola Tulisan Tangan," *J. Comput. Sci. Informatics Eng.*, vol. 6, no. 1, pp. 56–63, 2022, doi: 10.29303/jcosine.v6i1.431.
- [2] Y. Liu, Y. Wang, R. Yu, M. Li, V. Sharma, and Y. Wang, "Optimizing CNN model inference on CPUs," *Proc. 2019 USENIX Annu. Tech. Conf. USENIX ATC 2019*, pp. 1025–1039, 2019.
- [3] N. Irtiza Trinto and M. Eunus Ali, "Detecting Multilabel Sentiment and Emotions from Bangla YouTube Comments," *2018 Int. Conf. Bangla Speech Lang. Process. ICBSLP 2018*, no. January 2019, 2018, doi: 10.1109/ICBSLP.2018.8554875.
- [4] R. Leluc and F. Portier, "SGD with Coordinate Sampling: Theory and Practice," vol. 23, pp. 1–47, 2021, [Online]. Available: <http://arxiv.org/abs/2105.11818>
- [5] H. Kör, H. Erbay, and A. H. Yurttakal, "Diagnosing and differentiating viral pneumonia and COVID-19 using X-ray images," *Multimed. Tools Appl.*, vol. 81, no. 27, pp. 39041–39057, 2022, doi: 10.1007/s11042-022-13071-z.
- [6] X. Yan, Y. Xu, X. Xing, B. Cui, Z. Guo, and T. Guo, "Trustworthy Network Anomaly Detection Based on an Adaptive Learning Rate and Momentum in IIoT," *IEEE Trans. Ind. Informatics*, vol. 16, no. 9, pp. 6182–6192, 2020, doi: 10.1109/TII.2020.2975227.
- [7] L. Gaur, U. Bhatia, N. Z. Jhanjhi, G. Muhammad, and M. Masud, "Medical image-based detection of COVID-19 using Deep Convolution Neural Networks," *Multimed. Syst.*, no. 0123456789, 2021, doi: 10.1007/s00530-021-00794-6.
- [8] A. Hedström *et al.*, "Quantus: An Explainable AI Toolkit for Responsible Evaluation of Neural Network Explanations," vol. 24, pp. 1–11, 2022, [Online]. Available: <http://arxiv.org/abs/2202.06861>
- [9] M. P. Covid-, "Optimasi Deep Learning untuk Prediksi Saham di," vol. 7, no. 2, pp. 133–140, 2021.
- [10] M. I. Jordan, T. Lin, and M. Zampetakis, "First-Order Algorithms for Nonlinear Generalized Nash Equilibrium Problems," vol. 24, pp. 1–46, 2022, [Online]. Available: <http://arxiv.org/abs/2204.03132>
- [11] R. Kumar *et al.*, "Classification of COVID-19 from chest x-ray images using deep features and correlation coefficient," *Multimed. Tools Appl.*, vol. 81, no. 19, pp. 27631–27655, 2022, doi: 10.1007/s11042-022-12500-3.
- [12] E. M. Achour, F. Malgouyres, and F. Mamalet, "Existence, Stability and Scalability of Orthogonal Convolutional Neural Networks," vol. 23, pp. 1–56, 2021, [Online]. Available: <http://arxiv.org/abs/2108.05623>
- [13] S. Kiziloluk and E. Sert, "COVID-CCD-Net: COVID-19 and colon cancer diagnosis system with optimized CNN hyperparameters using gradient-based optimizer," *Med. Biol. Eng. Comput.*, vol. 60, no. 6, pp. 1595–1612, 2022, doi: 10.1007/s11517-022-02553-9.
- [14] F. Cecon *et al.*, "OMLT: Optimization & Machine Learning Toolkit," vol. 23, pp. 1–8, 2022, [Online]. Available: <http://arxiv.org/abs/2202.02414>
- [15] B. Xu, D. Martín, M. Khishe, and R. Boostani, "COVID-19 diagnosis using chest CT scans and deep convolutional neural networks evolved by IP-based sine-cosine algorithm," *Med. Biol. Eng. Comput.*, vol. 60, no. 10, pp. 2931–2949, 2022, doi: 10.1007/s11517-022-02637-6.
- [16] I. Arrieta-ibarra, P. Gujral, J. Tannen, M. Tygert, and C. Xu, "Metrics of calibration for probabilistic predictions," *Tech. Rep.*, vol. 23, pp. 1–50, 2022.
- [17] T. Yuan, W. Liu, J. Han, and F. Lombardi, "High Performance CNN Accelerators Based on Hardware and Algorithm Co-Optimization," *IEEE Trans. Circuits Syst. I Regul. Pap.*, vol. 68, no. 1, pp. 250–263, 2021, doi: 10.1109/TCSI.2020.3030663.
- [18] D. Irfan and T. S. Gunawan, "COMPARISON OF SGD , RMSProp , AND ADAM OPTIMATION IN ANIMAL CLASSIFICATION USING CNNs," *2nd Int. Conf. Infomation Sci. anda Technol. Innov.*, 2023.

- [19] L. Yu *et al.*, “Prediction of pathologic stage in non-small cell lung cancer using machine learning algorithm based on CT image feature analysis,” *BMC Cancer*, vol. 19, no. 1, pp. 1–12, 2019, doi: 10.1186/s12885-019-5646-9.
- [20] H. Wang *et al.*, “Comparison of machine learning methods for classifying mediastinal lymph node metastasis of non-small cell lung cancer from 18F-FDG PET/CT images,” *EJNMMI Res.*, vol. 7, no. 1, 2017, doi: 10.1186/s13550-017-0260-9.
- [21] S. S. A. Laros, D. B. M. Dickerscheid, S. P. Blazis, and J. A. van der Heide, “Machine learning classification of mediastinal lymph node metastasis in NSCLC: a multicentre study in a Western European patient population,” *EJNMMI Phys.*, vol. 9, no. 1, 2022, doi: 10.1186/s40658-022-00494-8.
- [22] N. Hasan, Y. Bao, A. Shawon, and Y. Huang, “DenseNet Convolutional Neural Networks Application for Predicting COVID-19 Using CT Image,” *SN Comput. Sci.*, vol. 2, no. 5, pp. 1–11, 2021, doi: 10.1007/s42979-021-00782-7.
- [23] J. Tachella, D. Chen, and M. Davies, “Sensing Theorems for Unsupervised Learning in Linear Inverse Problems,” vol. 24, pp. 1–45, 2022, [Online]. Available: <http://arxiv.org/abs/2203.12513>
- [24] L. T. K. Hien, D. N. Phan, and N. Gillis, “An Inertial Block Majorization Minimization Framework for Nonsmooth Nonconvex Optimization,” vol. 24, no. 30468160, pp. 1–41, 2020, [Online]. Available: <http://arxiv.org/abs/2010.12133>
- [25] N. Ho, C.-Y. Yang, and M. I. Jordan, “Convergence Rates for Gaussian Mixtures of Experts,” vol. 23, pp. 1–81, 2019, [Online]. Available: <http://arxiv.org/abs/1907.04377>
- [26] T. Ikeuchi, “Python package for causal discovery based on LiNGAM,” vol. 24, pp. 1–8, 2023.
- [27] K. Ji and Y. Liang, “Lower Bounds and Accelerated Algorithms for Bilevel Optimization,” vol. 23, pp. 1–56, 2021, [Online]. Available: <http://arxiv.org/abs/2102.03926>
- [28] A. Mahillo and D. Mart, “Discrete Variational Calculus for Accelerated Optimization,” vol. 24, pp. 1–33, 2023.
- [29] M. Unser, “Ridges , Neural Networks , and the Radon Transform,” vol. 24, pp. 1–33, 2023.
- [30] S. Duan and J. C. Principe, “Labels, Information, and Computation: Efficient, Privacy-Preserving Learning Using Sufficient Labels,” vol. 24, pp. 1–35, 2021, [Online]. Available: <http://arxiv.org/abs/2104.09015>