

A COMPARISON PRE-TRAINED MODELS FOR AUTOMATIC INDONESIAN LICENSE PLATE RECOGNITION

Sahid Bismantoko¹, M. Rosyidi¹, Umi Chasanah¹, Tri Widodo¹

¹ Center of Technology for System and Infrastructure of Transportation
Agency for the Assessment and Application of Technology
e-mail: sahid.bismantoko@bppt.go.id

ABSTRACT

Automatic License Plate Recognition is related to the Intelligent Transportation System (ITS) that supports the road's e-law enforcement system. In the case of the Indonesian license plate, with various colour rules for font and background, and sometimes vehicle owners modify their license plate font format, this is a challenge in the image processing approach. This research utilizes pre-trained of AlexNet, VGGNet, and ResNet to determine the optimum model of Indonesian character license plate recognition. Three pre-trained approaches in CNN-based detection for reducing time for a build if model from scratch. The experiment shows that using the pre-trained ResNet model gives a better result than another two approaches. The optimum results were obtained at epoch 50 with an accuracy of 99.9% and computation time of 26 minutes. This experiment results fulfil the goal of this research.

Keywords : ALPR; ITS; CNN; AlexNet; VGGNet; ResNet

INTRODUCTION

The availability of CCTV and cloud computing technology with high-performance processing units has made the application of deep learning computation to Automatic License Plate Recognition (ALPR) an exciting research field. The implementation of ALPR has several challenges, including; diversity of license plates (LP) formats, illumination changes, and motion blur. ALPR's challenge in Indonesian vehicle LPs is the variety of characters and background colours based on vehicle type **Figure 1**. Also, font modification by vehicle owners is a challenge in the image processing approach.



Figure 1. Indonesian License Plate

Image processing is one of the Computer Vision techniques to read or recognize the LP characters. The most famous image processing tools that successfully detect character accurately are deep convolution neural network (CNN). Deep CNN solves the character recognition problems and works well with image analysis, media recreation, natural language processing, and other recognize issues. Especially for character recognition, CNN accuracy has increased than traditional NN. This study intends to identify a suitable pre-trained method to train the character recognition of Indonesian LP.

RELATED WORKS

Three CNN-based approaches will be compared in this research. The related works focus on pre-trained methods to training Indonesian LP character recognition.

Rajaraman et al. [1] proposed pre-trained CNN-based deep learning models for classifying parasitized and uninfected cells. The experiment determines the optimal model layers for feature extraction from the underlying

data. This research utilizes pre-trained for feature extraction of malaria parasite detection in thin blood smear images, not applying all of the systems for a solution. The last analysis used a statistical approach. This pre-trained method becomes an idea for applying another pre-trained method for this research.

Sasongko et al. [2] use ResNet as a transfer learning technique for solving Indonesia toll road vehicle classification. Similar research conducted by Abubakar et al. [3] use transfer learning by leveraging pre-trained deep learning models due to deficient datasets to discriminate two classes of skin injuries; burnt skin and injured skin. An experiment conducted using pre-trained includes ResNet50, ResNet101, and ResNet152. The focus of both types of research is using Resnet as the main transfer learning.

Liang et al. [4] used transfer learning on the AlexNet CNN to improve the LPs localization. The final AlexNet layer has been modified to localize the LP into two output classes: the LP class and the background class. Furthermore, transfer learning is carried out in 3 iterations. The first iteration to learn LP characters with labels ground-truth includes only the LP character. In the next iteration, all network false-positive locations were labelled in training set 2 as background classes, while the actual LP locations had been labelled as LP class. The final iteration is re-labelling so that the ground-truth bounding box covers the LP's edges and the LP character. The goal is to train the network to find out that the character is in LP.

Li et al. [5] adopt VGG-16 to extract low-level CNN features. The local features for LP detection are extracted using 13 convolution layers VGG-16, while the fully connected layers are unused. Since LP is relatively small in an image, to overcome the LP feature vanished after pooling, two pooling layers are applied. The resulting feature map is a quarter of the original input image and is used as the basis for detection and recognition.

Many researches conduct a comparison of CNN transfer learning to solve classification problems. Brodzicki et al. [6] conduct research for bacteria classification and make the comparison of VGG-19, ResNet50, Xception, and DenseNet121. Taló et al. [7] solve Multi-class Histopathology Image Classification using DenseNet-161 and ResNet-50 pre-trained. Jing et al. [8] reviewed five 2DConvNets architectures for image feature learning, namely

AlexNet, VGG, GoogLeNet, ResNet, and DenseNet.

This research will compare ResNet, VGGNet, and AlexNet as base transfer learning for Indonesian LP character recognition. Accuracy and computation time analysis is applied to determine the optimum model for Indonesian LP's application.

METHODS

Character recognition of Indonesian LP in this study was solved using a character classification approach. Three transfer learning methods were used as a comparison to solve that classification problem. The evaluation is based on accuracy, epoch, and computational time.

A. AlexNet Pre-trained

Alex Krizhevsky et al. [9] proposed AlexNet to compete in the ImageNet competition. This model's success (which took first place in the 2012 ImageNet competition) convinced many computer vision communities to take a severe look at deep learning for computer vision tasks. AlexNet contained eight layers. Convolutional Layers were in the first five layers, several layers followed by a max-pooling layer, and the rest is fully connected layers. The architecture of AlexNet is shown in **Table 1**.

B. VGG16 Pre-trained

The VGG network offers a deeper yet simpler variant of the convolutional structures [10]. VGG16 contains four stack convolutional layers. At first, two stacks perform two convolutional and max pooling. The third and fourth stack has three convolutional layers and max-pooling layers, respectively. And the last three layers perform fully connected layers. The architecture of VGG16 is shown in **Table 2**.

C. ResNet Pre-trained

The Residual Network stands for ResNet, was proposed by Kaiming He et al. [11] in their work.

ResNet was using residual modules, specifically the residual identity model. ResNet consists of two convolutional layers with the same number of filters and small filter size. The result of the first convolution will be combined

with the output of the second convolution. Depict as a graph, the module's input is added to the module output and is called a shortcut connection.

ResNet network uses an 18-layer plain network architecture inspired by VGG-19, the shortcut connection is added. These shortcut connections then convert the architecture into the residual network, as shown in **Table 3**.

Table 1. AlexNet architecture

Layer	Output	Param
Conv2d-1	[-1, 64, 15, 15]	23,296
ReLU-2	[-1, 64, 15, 15]	0
MaxPool2d-3	[-1, 64, 7, 7]	0
Conv2d-4	[-1, 192, 7, 7]	307,392
ReLU-5	[-1, 192, 7, 7]	0
MaxPool2d-6	[-1, 192, 3, 3]	0
Conv2d-7	[-1, 384, 3, 3]	663,936
ReLU-8	[-1, 384, 3, 3]	0
Conv2d-9	[-1, 256, 3, 3]	884,992
ReLU-10	[-1, 256, 3, 3]	0
Conv2d-11	[-1, 256, 3, 3]	590,080
ReLU-12	[-1, 256, 3, 3]	0
MaxPool2d-13	[-1, 256, 1, 1]	0
AdaptiveAvgPool2d-14	[-1, 256, 6, 6]	0
Dropout-15	[-1, 9216]	0
Linear-16	[-1, 4096]	37,752,832
ReLU-17	[-1, 4096]	0
Dropout-18	[-1, 4096]	0
Linear-19	[-1, 4096]	16,781,312
ReLU-20	[-1, 4096]	0
Linear-21	[-1, 36]	147,492

Table 2. VGG16 architecture

Layer	Output	Param
Conv2d-1	[-1, 64, 64, 64]	1,792
ReLU-2	[-1, 64, 64, 64]	0
Conv2d-3	[-1, 64, 64, 64]	36,928
ReLU-4	[-1, 64, 64, 64]	0
MaxPool2d-5	[-1, 64, 32, 32]	0
Conv2d-6	[-1, 128, 32, 32]	73,856
ReLU-7	[-1, 128, 32, 32]	0
Conv2d-8	[-1, 128, 32, 32]	147,584
ReLU-9	[-1, 128, 32, 32]	0
MaxPool2d-10	[-1, 128, 16, 16]	0
Conv2d-11	[-1, 256, 16, 16]	295,168
ReLU-12	[-1, 256, 16, 16]	0
Conv2d-13	[-1, 256, 16, 16]	590,080
ReLU-14	[-1, 256, 16, 16]	0
Conv2d-15	[-1, 256, 16, 16]	590,080
ReLU-16	[-1, 256, 16, 16]	0
MaxPool2d-17	[-1, 256, 8, 8]	0
Conv2d-18	[-1, 512, 8, 8]	1,180,160
ReLU-19	[-1, 512, 8, 8]	0
Conv2d-20	[-1, 512, 8, 8]	2,359,808
ReLU-21	[-1, 512, 8, 8]	0
Conv2d-22	[-1, 512, 8, 8]	2,359,808
ReLU-23	[-1, 512, 8, 8]	0
MaxPool2d-24	[-1, 512, 4, 4]	0
Conv2d-25	[-1, 512, 4, 4]	2,359,808
ReLU-26	[-1, 512, 4, 4]	0
Conv2d-27	[-1, 512, 4, 4]	2,359,808
ReLU-28	[-1, 512, 4, 4]	0
Conv2d-29	[-1, 512, 4, 4]	2,359,808
ReLU-30	[-1, 512, 4, 4]	0
MaxPool2d-31	[-1, 512, 2, 2]	0
AdaptiveAvgPool2d-32	[-1, 512, 7, 7]	0
Linear-33	[-1, 4096]	102,764,544
ReLU-34	[-1, 4096]	0
Dropout-35	[-1, 4096]	0
Linear-36	[-1, 4096]	16,781,312
ReLU-37	[-1, 4096]	0
Dropout-38	[-1, 4096]	0
Linear-39	[-1, 36]	147,492

Table 3. ResNet18 architecture

Layer	Output	Param
Conv2d-1	[-1, 64, 32, 32]	9,408
BatchNorm2d-2	[-1, 64, 32, 32]	128
ReLU-3	[-1, 64, 32, 32]	0
MaxPool2d-4	[-1, 64, 16, 16]	0
Conv2d-5	[-1, 64, 16, 16]	36,864
BatchNorm2d-6	[-1, 64, 16, 16]	128
ReLU-7	[-1, 64, 16, 16]	0
Conv2d-8	[-1, 64, 16, 16]	36,864
BatchNorm2d-9	[-1, 64, 16, 16]	128
ReLU-10	[-1, 64, 16, 16]	0
BasicBlock-11	[-1, 64, 16, 16]	0
Conv2d-12	[-1, 64, 16, 16]	36,864
BatchNorm2d-13	[-1, 64, 16, 16]	128
ReLU-14	[-1, 64, 16, 16]	0
Conv2d-15	[-1, 64, 16, 16]	36,864
BatchNorm2d-16	[-1, 64, 16, 16]	128
ReLU-17	[-1, 64, 16, 16]	0
BasicBlock-18	[-1, 64, 16, 16]	0
Conv2d-19	[-1, 128, 8, 8]	73,728
BatchNorm2d-20	[-1, 128, 8, 8]	256
ReLU-21	[-1, 128, 8, 8]	0
Conv2d-22	[-1, 128, 8, 8]	147,456
BatchNorm2d-23	[-1, 128, 8, 8]	256
Conv2d-24	[-1, 128, 8, 8]	8,192
BatchNorm2d-25	[-1, 128, 8, 8]	256
ReLU-26	[-1, 128, 8, 8]	0
BasicBlock-27	[-1, 128, 8, 8]	0
Conv2d-28	[-1, 128, 8, 8]	147,456
BatchNorm2d-29	[-1, 128, 8, 8]	256
ReLU-30	[-1, 128, 8, 8]	0
Conv2d-31	[-1, 128, 8, 8]	147,456
BatchNorm2d-32	[-1, 128, 8, 8]	256
ReLU-33	[-1, 128, 8, 8]	0
BasicBlock-34	[-1, 128, 8, 8]	0
Conv2d-35	[-1, 256, 4, 4]	294,912
BatchNorm2d-36	[-1, 256, 4, 4]	512
ReLU-37	[-1, 256, 4, 4]	0
Conv2d-38	[-1, 256, 4, 4]	589,824
BatchNorm2d-39	[-1, 256, 4, 4]	512
Conv2d-40	[-1, 256, 4, 4]	32,768
BatchNorm2d-41	[-1, 256, 4, 4]	512
ReLU-42	[-1, 256, 4, 4]	0
BasicBlock-43	[-1, 256, 4, 4]	0
Conv2d-44	[-1, 256, 4, 4]	589,824
BatchNorm2d-45	[-1, 256, 4, 4]	512

ReLU-46	[-1, 256, 4, 4]	0
Conv2d-47	[-1, 256, 4, 4]	589,824
BatchNorm2d-48	[-1, 256, 4, 4]	512
ReLU-49	[-1, 256, 4, 4]	0
BasicBlock-50	[-1, 256, 4, 4]	0
Conv2d-51	[-1, 512, 2, 2]	1,179,648
BatchNorm2d-52	[-1, 512, 2, 2]	1,024
ReLU-53	[-1, 512, 2, 2]	0
Conv2d-54	[-1, 512, 2, 2]	2,359,296
BatchNorm2d-55	[-1, 512, 2, 2]	1,024
Conv2d-56	[-1, 512, 2, 2]	131,072
BatchNorm2d-57	[-1, 512, 2, 2]	1,024
ReLU-58	[-1, 512, 2, 2]	0
BasicBlock-59	[-1, 512, 2, 2]	0
Conv2d-60	[-1, 512, 2, 2]	2,359,296
BatchNorm2d-61	[-1, 512, 2, 2]	1,024
ReLU-62	[-1, 512, 2, 2]	0
Conv2d-63	[-1, 512, 2, 2]	2,359,296
BatchNorm2d-64	[-1, 512, 2, 2]	1,024
ReLU-65	[-1, 512, 2, 2]	0
BasicBlock-66	[-1, 512, 2, 2]	0
AdaptiveAvgPool2d-67	[-1, 512, 1, 1]	0
Linear-68	[-1, 36]	18,468

EXPERIMENTAL SETTINGS

A. Transfer Learning Approach

Transfer learning techniques are applied to the AlexNet, VGG16, and ResNet18 pre-trained CNN architectures. An experiment was carried out using epoch [30,50,100] to analyse those methods. Accuracy calculations were carried out for each algorithm with the three epoch parameters. The computation time for each parameter is also performed as an evaluation.

B. Dataset

The sample dataset for Indonesian LP characters is shown in **Figure 2**. These characters have been segmented from the LP image. It is split into two parts: training dataset and test dataset. **Table 4** shows the ratio of LP character that used in this research. There are 37,783 datasets with a distribution: 80% is used for training, and the remaining 20% is used for testing.



Figure 2. Indonesian LP Character Dataset

Table 4. Dataset ratio

Dataset	Amount
Training	30,226
Test	7,556
Total	37,783

C. Environment

The ALPR experiment's environment setting is 4 cores CPU with 16 Gigabytes of RAM and GPU P100-PCIE-16GB with 2 CPU cores and 13 Gigabytes RAMS. A batch of train and test are 800 and 300.

EXPERIMENTAL RESULT

A. Evaluation

Figure 3 shows the evaluation result of the three methods that were used. There are three parameters to make the evaluation; accuracy, loss, and computation time. Each transfer learning approach is tested through three epoch arrangements; 30, 50, and 100. Each part will explain the reliability of the pre-trained model for itself and compare it to each other.

The first step of the experiment configuration is 30 epochs. AlexNet reaches 80.625% at the final epoch. Initial accuracy in the first batch on the first epoch is 26.150% and has increased in each step. AlexNet gets the highest accuracy score at 83.55%. Meanwhile, VGG16 had an initial accuracy score of 40.150% at the first batch in the first epoch and continues increasing until getting 84.575% as the highest value. The final epoch VGG16 gets 84.213% as an accuracy score. ResNet18 got a higher accuracy score than the two previous models. The first batch in the first epoch ResNet gets 63.700% and had increased to 99.425% on the final epoch. The highest accuracy score of ResNet18 in the 30 epochs configuration is 99.650%.

The second configuration is 50 epochs. AlexNet gets the highest accuracy score at 84% and reaches 83.112% at the final epoch.

Meanwhile, VGG16's highest accuracy score is 88.875% and 87.300% as the final value. ResNet18 is still better than the two previous models. ResNet reaches the highest accuracy score and final accuracy score in 99.900% and 99.862%, respectively.

The final configuration is 100 epochs. ResNet18 still gets the best accuracy compared to AlexNet and VGG16. ResNet18 gets the highest accuracy score and final score in 99.983%, 99.737%, respectively. On the other hand, AlexNet's highest accuracy score is 86.800%, and the final score is 85.963%. VGG16 gets 84.850% as the highest accuracy score and 84% in the last step.

The computation time calculation in Table 5 shows the time increase of about two times at epoch 30 to 50 and 50 to 100 on the AlexNet and VGG16 methods. Whereas in the ResNet18 method, the initial computation time is 16 minutes, almost twice that of AlexNet, but in the next epoch, the increase in computation time is not as high as the previous two methods. ResNet18 computation time on epoch 50 is 26 minutes, and epoch 100 is 48 minutes.

The amount of data used in the model validation process was 7557, following the test sets' predetermined number. The validated model results from training based on the number of epochs 30, 50, and 100.

The validation result in Table 6 shows that ResNet18 has a high accuracy level for each epoch setting, which is around 99%. The VGG16 validation value at epoch 30 is approximately 60% and reaches the best value at epoch 100. The AlexNet validation value tends to be stable above the 80% value for all epochs.

B. Comparison

All transfer learning approaches show promising results when applied to the Indonesian LP character dataset. The comparison of these methods can be explained as:

1. ResNet18 offers the highest value of training data and test accuracy. VGG16 accuracy is better than AlexNet, but the loss score has the opposite result; AlexNet is better than VGG16.
2. The smallest loss value is ResNet18, and the loss reduction is more reliable than the other two transfer learning.
3. ResNet18 computation time is like AlexNet, while VGG16 is a little slower than others.

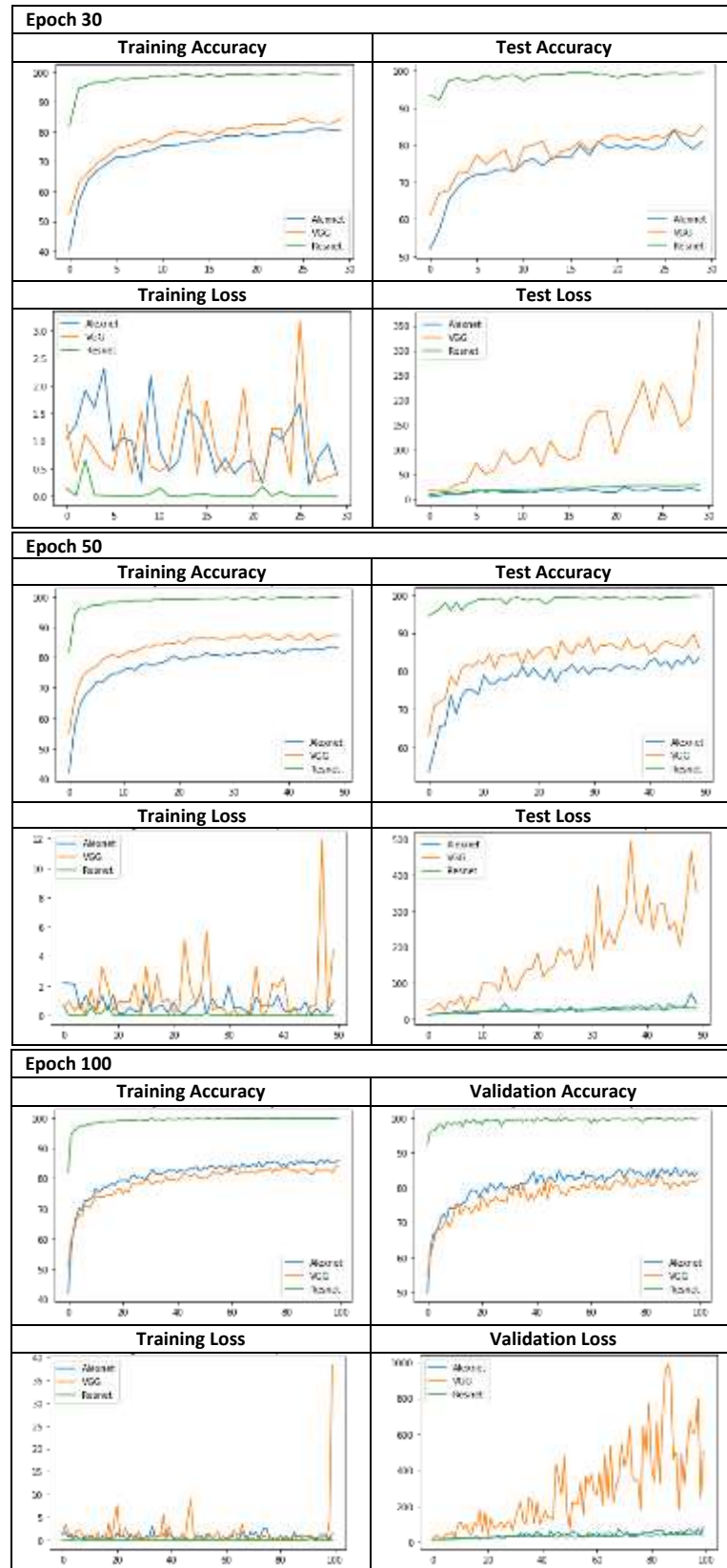


Figure 3. Comparison of transfer learning AlexNet, VGGNet, and ResNet

Table 5. Computation time

No	Models	Epoch 30	Epoch 50	Epoch 100
1	AlexNet	9 minutes	17 minutes	31 minutes
2	VGG16	18 minutes	31 minutes	60 minutes
3	ResNet18	16 minutes	26 minutes	48 minutes

Table 6. Validation Result

No	Models	Epoch 30	Epoch 50	Epoch 100
1	AlexNet	81.143%	80.865%	84.862%
2	VGG	60.487%	67.156%	85.378%
3	Resnet	99.563%	99.524%	99.762%

CONCLUSION

Based on the experiment result:

1. Three approaches of transfer learning have been applied to recognize the character of Indonesian LP. ResNet18 has the highest accuracy level for each epoch setting, reaches 99%.
2. ResNet18 is the most promising transfer learning method compared to AlexNet and VGG16 because the smallest loss value and loss line decrease each epoch and consistency of the accuracy training and test dataset.

FUTURE RESEARCH

Based on the conclusions:

1. Add another transfer learning approach to get more comparable models.
2. The epoch's number should be increased to prove ResNet accuracy and apply to another method in the future.

AUTHOR INFORMATION

Corresponding Authors

Email: sahid.bismantoko@bppt.go.id

Email: m.rosyidi@bppt.go.id

Email: umi.chasanah@bppt.go.id

Email: tri.widodo@bppt.go.id

Author Contributions

All Authors have contributed equally to this work.

ACKNOWLEDGEMENT

Thanks to the Center of Technology for System and Infrastructure of Transportation, Agency for the Assessment and Application of Technology, that fully support this research related to the technical support and research funding.

REFERENCES

- [1] S. Rajaraman et al., "Pre-trained convolutional neural networks as feature extractors toward improved malaria parasite detection in thin blood smear images," PeerJ, 2018.
- [2] A. T. Sasongko and M. Ivan Fanany, "Indonesia Toll Road Vehicle Classification Using Transfer Learning with Pre-Trained Resnet Models," 2019 2nd Int. Semin. Res. Inf. Technol. Intell. Syst. ISRITI 2019, pp. 373–378, 2019.
- [3] A. Abubakar, M. Ajuji, and I. U. Yahya, "Comparison of deep transfer learning techniques in human skin burns discrimination," Appl. Syst. Innov., vol. 3, no. 2, pp. 1–15, 2020.
- [4] T. E. Liang, U. U. Sheikh, and M. N. H. Mohd, "Malaysian car plate localization using region-based convolutional neural network," Bull. Electr. Eng. Informatics, vol. 9, no. 1, pp. 411–419, 2020.
- [5] H. Li, P. Wang, and C. Shen, "Toward End-to-End Car License Plate Detection and Recognition with Deep Neural

- Networks," IEEE Trans. Intell. Transp. Syst., vol. 20, no. 3, pp. 1126–1136, 2019.
- [6] A. Brodzicki, J. Jaworek-Korjakowska, P. Kleczek, M. Garland, and M. Bogyo, "Pre-trained deep convolutional neural network for clostridioides difficile bacteria cytotoxicity classification based on fluorescence images," Sensors (Switzerland), 2020.
- [7] M. Talo, "Convolutional neural networks for multi-class histopathology image classification," arXiv, 2019.
- [8] L. Jing and Y. Tian, "Self-supervised visual feature learning with deep neural networks: A survey," arXiv, pp. 1–24, 2019.
- [9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in Advances in Neural Information Processing Systems, 2012.
- [10] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc., pp. 1–14, 2015.
- [11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2016.