COMPARISON OF DENSENET AND RESNET ARCHITECTURES FOR CLASSIFICATION ON WHITE BLOOD CELL

[1]Jessica Betha Yolanda
[1]Program Studi Teknik Informatika Fakultas Ilmu Komputer, Universitas Katholik Soegijapranata
[1]20k10039@unika.ac.id

Abstract

White blood cell examination is useful in determining disease conditions. Accuracy in white blood cell classification is crucial, as errors can result in improper disease detection. Currently, white blood cell classification in healthcare is still done manually, so a system is needed to reduce classification errors. Researchers have tried to classify white blood cells using CNN, ResNet architecture provides good performance. Similar to ResNet, DenseNet also has "skip connections", but the difference is that it has fewer parameters. DenseNet has been evaluated on competitive datasets and the results outperformed ResNet. So in this study, we suspect that the DenseNet architecture can provide higher performance than ResNet in classifying white blood cells. We wanted to compare the performance of ResNet and DenseNet for classifying white blood cells. We used certain parameters and divided the dataset with various comparisons on the two architectures. Then the performance of the model will be evaluated by calculating sensitivity, specificity, and accuracy. Thus, the results of this study can be implemented in the health sector and can help health workers when classifying white blood cells with the right predictions. The best result from the comparison of both DenseNet and ResNet architectures is the DenseNet architecture. DenseNet obtained the results of sensitivity 64,85234678, specificity 88,26916351, and accuracy 82,40852433.

Keywords: white blood cell, cnn, densenet, resnet

Introduction

For disease diagnosis, hematology tests or complete blood tests have become very important in recent years. To determine a patient's medical condition, white blood examination can be of great help. In the classification of white cells, accuracy is crucial as incorrect classification of white cells will lead to improper diagnosis of diseases. Until now, WBC detection and categorization is done manually by experts in many medical centers [1]. This is because manual classification still allows errors. Therefore, the system should be designed in a way that can reduce the error rate.

Using Convolutional Neural Network researchers have tried to classify white blood cells. Various CNN architectures have been used by researchers to achieve a high level of performance. In 2018 Macawile et al. performed white blood cell classification by comparing 3 CNN architectures namely AlexNet, GoogleNet, and ResNet-101. Based on their research, the highest accuracy result obtained was ResNet-101 of 97.552%[2].

The ResNet-101 architecture is a CNN that is 101 layers deep and has "skip connections" that are used to mitigate the vanishing gradient problem or the condition where the gradient shrinks due to layers that are too deep. Similar to ResNet-101, the DenseNet-169 architecture also has "skip connections" but the

difference is that it has fewer parameters. Huang et al. have evaluated DenseNets on four competitive benchmark datasets (CIFAR-10, CIFAR-100, SVHN, and ImageNet) with the result that DenseNets has been shown to outperform ResNets[3]. So in this study, we suspect that the DenseNet-169 architecture is able to provide higher performance than the ResNet-101 architecture in classifying white blood cells.

Sejumlah penelitian telah dilakukan untuk mengatasi tantangan dalam klasifikasi sel darah putih. Cheng et al. [4] menggunakan metode Faster RCNN untuk menemukan sel darah putih dalam gambar darah smear besar, meningkatkan F1-Score, recall, dan presisi. Macawile et al. [2] membandingkan tiga model CNN dan menemukan bahwa ResNet-101 memiliki akurasi tertinggi sebesar 97.552%. Gautam and Bhadauria [5] menggunakan fitur morfologi untuk mengklasifikasikan sel darah putih, tetapi hanya mencapai akurasi 73%. Treebupachatsakul and Poomrittigul [6] mengklasifikasikan bakteri menggunakan model LeNet CNN, mencapai akurasi sebesar 75%. Manik et al. [7] menggunakan ekstraksi fitur dan jaringan saraf tiruan (ANN) untuk mengklasifikasikan tiga jenis sel darah putih dengan akurasi 98.9%. Yu et al. [8] menggabungkan hasil klasifikasi beberapa model CNN menggunakan mekanisme voting, dengan akurasi sekitar 88.5%. Rosyadi et al. [9] menggunakan metode K-Means untuk mengklasifikasikan sel darah putih dengan akurasi bervariasi tergantung fitur yang digunakan. Liang et al. [10] menggabungkan CNN dan RNN dengan akurasi tertinggi sebesar 90.79%. Ridoy and Islam[1] menggunakan algoritma Lightweight CNN dan mencapai AUC score 0.99 untuk klasifikasi multikelas. Terakhir, Gautam et al. [11] menggunakan fitur morfologi dan Naïve Bayes dengan peningkatan akurasi menjadi 80.88%. Studi terbaru oleh Huang et al. [3] menyoroti keefektifan DenseNet, menunjukkan kemampuannya mengatasi masalah vanishing-gradient dan mengurangi jumlah parameter.

Dengan demikian, penelitian ini bermaksud untuk membandingkan arsitektur DenseNet-169 dan ResNet-101 dalam mengklasifikasikan sel darah putih. Dengan demikian, hasil dari penelitian ini dapat diimplementasikan dalam bidang kesehatan. Dengan tujuan untuk membantu tenaga kesehatan ketika mengklasifikasikan sel darah putih dengan prediksi yang tepat.

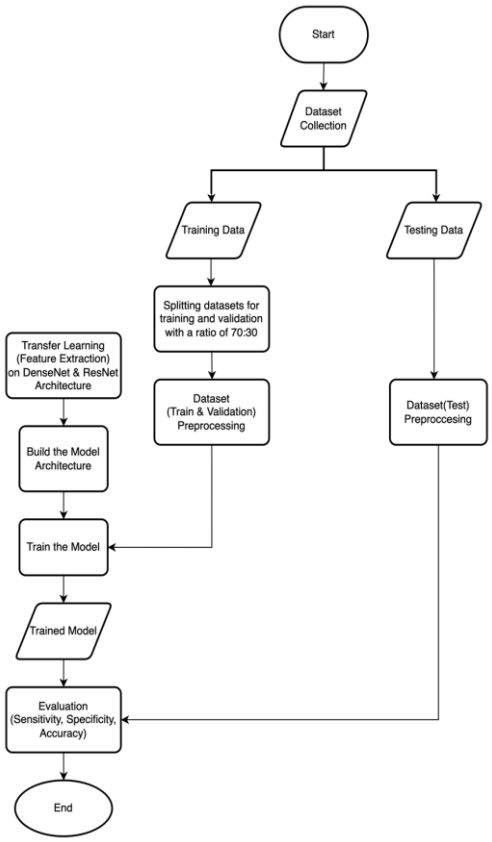

Figure 1: Workflow

Research Method

Dataset Collection

In Figure 1, the researcher starts by collecting datasets from Kaggle, specifically the Blood Cell Image dataset derived from the BCCD dataset. This dataset, which was created for blood cell detection, consists of about 12,500 images that have been augmented with associated labels. The images, which were originally 640 x 480, were resized to 320 x 240 to speed up model training. The Kaggle dataset is divided into 9,957 training images and 2,487 testing images, covering the cell categories of eosinophils, lymphocytes, monocytes, and neutrophils.

Splitting Data

In this research, the dataset is divided into 3 parts: training, validation, and testing. However, from the available datasets only training data and testing data. As listed in Figure 1, researchers divided the training dataset with a ratio of 70:30 for training data and validation data. The researcher used the validation_split parameter of ImageDataGenerator to split the training dataset into 2 parts, 70% for training and 30% for validation.

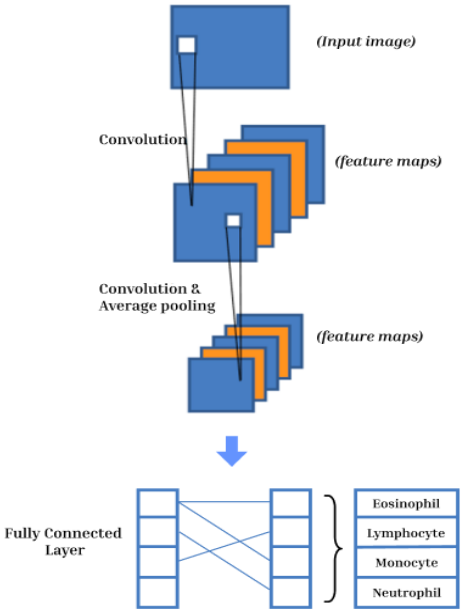

Figure 2: An Architecture of CNN

Dataset Preproccessing

In using CNN, the first and most important step before training or testing on a dataset is to ensure that the image dimensions are the same as the CNN requirements. The available dataset dimension size is 320 x 240, while DenseNet and ResNet require an input size of 224 x 224. So, researcher use the parameter in ImageDataGenerator i.e. .flow_from_directory which can be used to load the image and at the same time can be used to change the dataset dimensions to 224 x 224 with the target_size parameter.

Transfer Learning

To identify different types of white blood cells, researchers used transfer learning methods. As shown in Figure 1, before building the model, the researchers performed the transfer learning method first. Transfer learning is a method of reusing features that have been learned by the CNN model on a particular dataset and transferring those features to be used on other datasets. Using this technique can not only save time, but because the trained CNN has proven to be strong and provide results with high accuracy [2]. So, in this study, researchers used the Densenet and ResNet models that were pre-trained using the ImageNet dataset, then discarded the layers above the final layer. The final layer serves as a classification that will be used to predict the class. Researchers use the extracted features as input for the model that we will use later.

Build the Model Architecture

This research uses 3 layers as an architectural model, namely, convolutional layer, pool layer, and fully-connected layer. However, keep in mind that the use of CNN must ensure the dimensions of the input image with the dimensions of the CNN architecture used as the preprocessing process we have done above. The DenseNet and Resnet architectures have the same dimensions of 224 x 224.

Convolution Layer

The convolution layer initiates the architectural model, generating a "feature map" by convolving the input image with 3x3 and 5x5 filters. These filters scan the image to detect features such as lines, producing high responses on the feature map when identified. Various activation functions, including ReLu, Softmax, Tanh, Sigmoid, and Elu, were explored in the study. The research tested filter sizes of 32, 64, and 128, employing weight initialization techniques such as "he_normal" and "glorot_uniform."

Pool Layer

In this layer, Global Average Pooling (GAP) is employed due to its ability to reduce the number of parameters and prevent overfitting. GAP calculates the average of all feature maps, generating a single number for each feature map, as opposed to conventional pooling techniques that consider adjacent areas. The researchers opt for GAP to enhance efficiency compared to traditional pooling methods.

Fully Connected Layer

Following the preceding layer, the data is directed to the Softmax layer, producing an N-dimensional vector (N=4 classes in this study) with probability values for each class. A dropout layer is incorporated to mitigate overfitting. The study employs cross-entropy as the loss function to quantify the disparity between the actual and target outputs.

Train the Model

As shown in Figure 1, datasets that have been preprocessed and models that have been built will be trained. The model training process is carried out using the training dataset and validation dataset using a ratio of 70:30. Training using the training dataset, which is the process of training the model using a labeled dataset. After training using the training dataset, training using the validation dataset is carried out, which is useful for measuring the model's ability to datasets that have never been seen before. Researchers use validation datasets so that the model does not experience overfitting, which is a condition where the model is too focused on training data and cannot generalize data that has never been seen before. In this study using the RMSprop (Root Mean Square Propagation), SGD, and Adam optimizers, the three of which will be tested which one gives the best results and the loss function used is Categorical Crossentropy.

Evaluation

A test dataset was prepared to evaluate the performance of the DenseNet and Resnet models. Performance was measured using sensitivity, specificity, and accuracy, which are represented in a confusion matrix table with the categories True Positive (TP), False Negative (FN), True Negative (TN), and False Positive

$TPR = \frac{TP}{TP + FN}$	(1)
$TNR = \frac{TN}{TN + FP}$	(2)
$ACC = \frac{TP + TN}{TP + TN + FP + FN}$	(3)

(FP). Sensitivity measures the percentage of samples that actually belong to the class and are identified as such, specificity indicates the percentage of samples that actually do not belong to the class and are correctly identified, while accuracy is calculated as the ratio of correctly classified samples to total samples.

First, sensitivity(1) or True Positive Rate (TPR) indicates the percentage of samples that actually belong to the class and are identified as such. Secondly, specificity or True Negative Rate (TNR)(2) indicates the percentage of samples that do not actually belong to the class and are identified as such. Thirdly, accuracy (ACC)(3) is calculated as the ratio of all samples belonging to a class and the total number of samples.

Result and Discussion

In this study, several results were obtained from experimenting with various hyperparameters on both algorithms, namely DenseNet169 and ResNet101. We tried one hyperparameter at a time, and the results included sensitivity, specificity, and accuracy. The hyperparameters that gave the best results for the three evaluation metrics will be used for experiments on other hyperparameter settings with the aim of optimizing model performance.

Table 1: Result for Densenet169 and ResNet101

No	Algorithm	Avg Sensitivity	Avg Specificity	Avg Accuracy	Params
1	DenseNet169	69,7165068	89,90312682	84,86127865	479396
2	ResNet101	64,85234678	88,26916351	82,40852433	589988

In experiments that have been carried out by researchers by trying various parameters. We chose to use filter size 32, kernel size (3,3), sigmoid as convolution activation, he_normal as initialization, adam as optimization, learning rate 0.001, dropout rate 0.3, batch size 32 for both DenseNet169 and ResNet101 architectures. By using these hyperparameters, the sensitivity, specificity, and accuracy evaluation results obtained for both architectures achieved the highest values compared to other hyperparameters as shown in Table 1. Overall the evaluation results obtained, the DenseNet169 architecture obtained higher results compared to the ResNet101 architecture. However, when viewed from the params or weights required, the DenseNet169 architecture requires less weight than the ResNet101 architecture.

As discussed earlier, the researchers tried various hyperparameters to get the model with the highest evaluation results. The results showed that using filter 32 gave better evaluation performance compared to filters 64 and 128. Filter 32 also requires fewer params or weights compared to filters 64 and 128. In research [12], said that a large number of filters will give poor value to the generalization of the classifier. So the researcher chose to use 32 filters to classify white blood cells. In this study, the use of kernel size (3,3) also provides high evaluation results and smaller weights compared to kernel size (5,5). In [13] the kernel size (3,3) obtained better results and the use of kernel size (5,5) was too easy to remember.

When comparing activation functions, sigmoid provides the best evaluation results compared to other activation functions namely relu, softmax, tanh, and elu. Relu and elu gave low scores in terms of sensitivity, specificity, and accuracy. Sigmoid has a good ability in classification and is able to overcome vanishing gradient [14]. Meanwhile, the use of he_normal in this study gives better results than glorot_uniform. According to [15] the use of he_normal managed to provide a very low error rate.

In the optimizer experiment, Adam was able to provide more optimal results than using sgd and rmsprop. With Adam's ability to use adaptive methods, he is able to change the learning rate for epochs so as to achieve better accuracy [16]. The researcher chose to use a learning rate of 0.001 with the best results among other learning rates. A high learning rate does not always provide efficient results, but a low learning rate can provide higher accuracy results [17]. The use of a dropout rate of 0.3 prevents overfitting of the model and is able to provide stable results such as research [18]. Furthermore, the use of a batch size of 32 with a small learning rate will provide excellent accuracy results [19], as has been done by researchers using a learning rate of 0.001.

The experimental results obtained, the DenseNet169 architecture provides better results than the ResNet101 architecture. When viewed from the characteristics of the DenseNet169 architecture, this architecture can directly receive input from all previous layers. This characteristic allows the DenseNet169 architecture model to extract complex features. ResNet101, on the other hand, only accepts input from the previous layer. The model with DenseNet169 architecture is more efficient because the number of parameters required is less than ResNet101 as shown in Table 4.17. Although DenseNet169 has more layers, the dense connectivity, where each layer receives all inputs from the previous layer, makes the DenseNet architecture more efficient in parameter usage.

Conclusion

Based on the results obtained, the evaluation results of DenseNet169 in classifying white blood cells get a sensitivity value of 69.7165068, specificity 89.90312682, and accuracy of 84.86127865. Meanwhile, the Resnet101 architecture is unable to outperform the results of the DenseNet169 architecture with the evaluation results of the sensitivity value of 64,85234678, specificity of 88,26916351, and accuracy of 82,40852433.

The sensitivity, specificity, and accuracy values of the various hyperparameter experiments on the DenseNet169 architecture produce higher values than the ResNet101 architecture due to the characteristics of the Densenet169 architecture each layer receives all input from the previous layer. While ResNet101 only receives input from the previous layer. From the results of various hyperparameter experiments, the following hyperparameters are able to provide maximum results in both architectures, filter size 32, kernel size (3,3), sigmoid as activation convolution, he_normal as initializer, adam as optimizer, learning rate 0.001, dropout rate 0.3, batch size 32.

With fewer params or weights, Densenet169 still provides higher sensitivity, specificity, and accuracy values than ResNet101 which requires more params or weights, this is due to the architecture of Densenet169 which receives input from all previous layers. For future research, researchers can try various other hyperparameters to get better accuracy and performance.

Daftar Pustaka

[1] Md. A. R. Ridoy and Md. R. Islam, “A Lightweight Convolutional Neural Network for White Blood Cells Classification,” in *2020 23rd International Conference on Computer and Information Technology (ICCIT)*, DHAKA, Bangladesh: IEEE, Dec. 2020, pp. 1–5. doi: 10.1109/ICCIT51783.2020.9392649.

[2] M. J. Macawile, V. V. Quinones, A. Ballado, J. D. Cruz, and M. V. Caya, “White blood cell classification and counting using convolutional neural network,” in *2018 3rd International Conference on*

Control and Robotics Engineering (ICCRE), Nagoya: IEEE, Apr. 2018, pp. 259–263. doi: 10.1109/ICCRE.2018.8376476.
[3] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely Connected Convolutional Networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI: IEEE, Jul. 2017, pp. 2261–2269. doi: 10.1109/CVPR.2017.243.
[4] S. Cheng, Y. Suhua, and J. Shaofeng, "Improved faster RCNN for white blood cells detection in blood smear image," in *2019 14th IEEE International Conference on Electronic Measurement & Instruments (ICEMI)*, Changsha, China: IEEE, Nov. 2019, pp. 1677–1682. doi: 10.1109/ICEMI46757.2019.9101445.
[5] A. Gautam and H. Bhadauria, "Classification of white blood cells based on morphological features," in *2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, Delhi, India: IEEE, Sep. 2014, pp. 2363–2368. doi: 10.1109/ICACCI.2014.6968362.
[6] T. Treebupachatsakul and S. Poomrittigul, "Bacteria Classification using Image Processing and Deep learning," in *2019 34th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC)*, JeJu, Korea (South): IEEE, Jun. 2019, pp. 1–3. doi: 10.1109/ITC-CSCC.2019.8793320.
[7] S. Manik, L. M. Saini, and N. Vadera, "Counting and classification of white blood cell using Artificial Neural Network (ANN)," in *2016 IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES)*, Delhi, India: IEEE, Jul. 2016, pp. 1–5. doi: 10.1109/ICPEICES.2016.7853644.
[8] W. Yu *et al.*, "Automatic classification of leukocytes using deep neural network," in *2017 IEEE 12th International Conference on ASIC (ASICON)*, Guiyang: IEEE, Oct. 2017, pp. 1041–1044. doi: 10.1109/ASICON.2017.8252657.
[9] T. Rosyadi, A. Arif, Nopriadi, B. Achmad, and Faridah, "Classification of leukocyte images using K-Means Clustering based on geometry features," in *2016 6th International Annual Engineering Seminar (InAES)*, Yogyakarta, Indonesia: IEEE, Aug. 2016, pp. 245–249. doi: 10.1109/INAES.2016.7821942.
[10] G. Liang, H. Hong, W. Xie, and L. Zheng, "Combining Convolutional Neural Network With Recursive Neural Network for Blood Cell Image Classification," *IEEE Access*, vol. 6, pp. 36188–36197, 2018, doi: 10.1109/ACCESS.2018.2846685.
[11] A. Gautam, P. Singh, B. Raman, and H. Bhadauria, "Automatic classification of leukocytes using morphological features and Naïve Bayes classifier," in *2016 IEEE Region 10 Conference (TENCON)*, Singapore: IEEE, Nov. 2016, pp. 1023–1027. doi: 10.1109/TENCON.2016.7848161.
[12] A. U. R. Durrani, N. Minallah, N. Aziz, J. Frnda, W. Khan, and J. Nedoma, 'Effect of hyperparameters on the performance of ConvLSTM based deep neural network in crop classification', *PLoS ONE*, vol. 18, no. 2, p. e0275653, Feb. 2023, doi: 10.1371/journal.pone.0275653.
[13] S. Ozturk, U. Ozkaya, B. Akdemir, and L. Seyfi, 'Convolution Kernel Size Effect on Convolutional Neural Network in Histopathological Image Processing Applications', in *2018 International Symposium on Fundamentals of Electrical Engineering (ISFEE)*, Bucharest, Romania: IEEE, Nov. 2018, pp. 1–5. doi: 10.1109/ISFEE.2018.8742484.
[14] S. Sharma, S. Sharma, and A. Athaiya, 'ACTIVATION FUNCTIONS IN NEURAL NETWORKS', *IJEAST*, vol. 04, no. 12, pp. 310–316, May 2020, doi: 10.33564/IJEAST.2020.v04i12.054.

[15] L. Datta, ‘A Survey on Activation Functions and their relation with Xavier and He Normal Initialization’. arXiv, Mar. 18, 2020. Accessed: Nov. 30, 2023. [Online]. Available: http://arxiv.org/abs/2004.06632
[16] A. Ramdan, V. Zilvan, E. Suryawati, H. F. Pardede, and V. P. Rahadi, ‘Tea clone classification using deep CNN with residual and densely connections’, *Jurnal Teknologi dan Sistem Komputer*, vol. 8, no. 4, pp. 289–296, Oct. 2020, doi: 10.14710/jtsiskom.2020.13768.
[17] B. M. Mathunjwa, Y.-T. Lin, C.-H. Lin, M. F. Abbod, and J.-S. Shieh, ‘ECG arrhythmia classification by using a recurrence plot and convolutional neural network’, *Biomedical Signal Processing and Control*, vol. 64, p. 102262, Feb. 2021, doi: 10.1016/j.bspc.2020.102262.
[18] X. Liang *et al.*, ‘R-Drop: Regularized Dropout for Neural Networks’. arXiv, Oct. 29, 2021. Accessed: Dec. 01, 2023. [Online]. Available: http://arxiv.org/abs/2106.14448
[19]I. Kandel and M. Castelli, ‘The effect of batch size on the generalizability of the convolutional neural networks on a histopathology dataset’, *ICT Express*, vol. 6, no. 4, pp. 312–315, Dec. 2020, doi: 10.1016/j.icte.2020.04.010.