Enhancing Deeper Layers with Residual Network on CNN Architecture: A Review

A. Supani[1,2(✉)], Y. Andriani[1,2], Indarto[1,2], H. Saputra[1,2], A. Bahri Joni[1,2], D. Alfian[1,2], A.Taqwa[1,2], and A. Silvia H.[1,2]

[1] Computer Department, State Polytechnic of Sriwijaya, Palembang, Indonesia
ahyarsupani@polsri.ac.id
[2] Mathematics Department, Sriwijaya University, Palembang, Indonesia

Abstract. The Convolution Neural Network (CNN) architecture is well-suited to performing both detection and classification tasks on image data. The inclusion of layers in the CNN improves its performance whilst training. Adding a lot, on the other hand, will cause the architecture to lose or explode gradients while learning training data. To address this issue, a mechanism for inserting the residual network between two layer blocks, ReLu activation function, and Batch Normalization must be added. In this paper, we examine various past studies that used residual networks in CNN design to validate model performance improvements. The examination of this study's findings reveals highly substantial outcomes for the prediction of classification and detection tasks for picture data. We infer from previous research findings that the as have adds a deeper layer to the CNN without losing the gradient. In this paper, we examine various past studies that used residual networks in CNN design to validate model performance improvements. The examination of this study's findings reveals highly substantial outcomes for the prediction of classification and detection tasks for picture data. We infer from previous research findings that the as have adds a deeper layer to the CNN without losing the gradient.

Keywords: Residual Network (Resnet) · Classification · Detection · Convolution Neural Network

1 Introduction

CNN is a deep learning model that is quite popular for analyzing image data. In CNN, adding an inner layer will become increasingly advantageous, but in the specific situation, doing so will also result in a gradient explosion/vanishing when training. Loss of this gradient will lead to poor or lower prediction results during training. To overcome this problem, by adding a method to the CNN architecture consisting of residual network, ReLu activation function, and Batch Normalization, the CNN will increase the inner layer without losing or exploding its gradient during training, as in the Residual Neural Network (resnet) architecture.

Kaiming He et al. [1] developed a type of architecture termed as Resnet that is highly popular. Due to the state of the art in classification, detection, and segmentation tasks

N. L. Husni et al. (Eds.): FIRST-ESCSI 2022, AHE 14, pp. 449–457, 2023.
https://doi.org/10.2991/978-94-6463-118-0_46

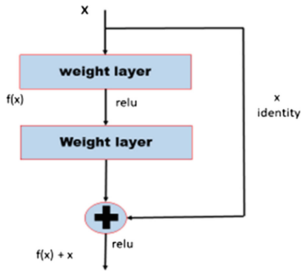


Fig. 1. Skip connection between two layer blocks forms residual network as the foundation of the Resnet architecture [1]

at the time, it was fairly groundbreaking. A CNN architecture that has a high depth is one of the important things in building a CNN model that has good performance. But it that has a high depth also has problems, namely the vanishing gradient problem, which is a situation where the gradient results studied by the model do not can reach the first layer because it is multiplied so many times that the first layer does not receive any gradient, or in short, this causes a CNN to be unable to learn from the calculated error [2]. The architecture of Resnet also varies, with layers ranging from 18 to 152 and commencing at 18, 34, 50, 101, and so on [1]. The Resnet34 architecture, which is a resnet architecture with 34 layers, is used in the study by Kamal HM et al. It was selected because it performed well in the ILSVRC competition [1]. Training data will be taken from the entire dataset to develop the CNN model, and validation data will be utilized to evaluate the model's performance. The accuracy, training and validation losses, as well as the error matrix, will be applied to evaluate how well the system is performing. Additionally, this statistic is employed in numerous research with CNN [3–5]. The suggestion made at the time by Kaiming He et al. was to employ a residual block, which is a block that appears in each layer of the CNN Resnet architecture and serves as its foundation; an overview of this block is shown in Fig. 1.

2 Related Works

Dinis LR [7] in the stenosis object of a coronary artery is one of a few analogous works with resnet architecture that has a detection task. This step's goal is to identify every stenosis that is visible in a frame and estimate its location. It is possible to approximate this to an object detection/recognition issue where the stenosis is the object of interest by using the annotated bounding boxes for the stenosis in the ideal interval, as shown in Fig. 2.

This work detects stenosis of the left or right coronary artery as a result of a different model, based on the single shot detector RetinaNet architecture was assembled as the second stage of the framework to automatically detect stenosis location. Table 1 displays the outcome of work [7].

Comparing Dinis LR's method to other authors confirms its superior performance, which achieves 0.72/0.70 recall and 0.82/0.84 precision for one detection, 0.73/0.68 recall and 0.72/0.74 precision for five detections on the RCA/LCA, respectively. It

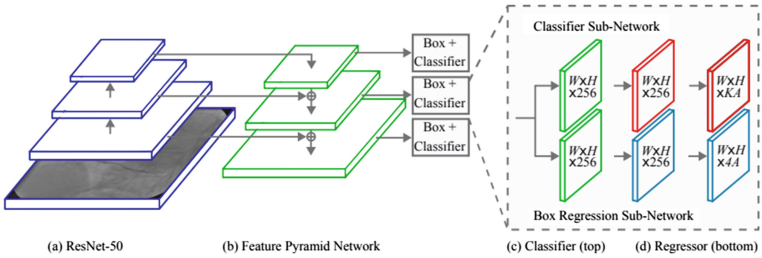


Fig. 2. A RetinaNet: (a) ResNet-50, (b) Feature Pyramid Network, (c) classifier (top), and (d) classifier (bottom)

Table 1. Metrics for detecting stenosis in comparison to previous work.

	Author	IoU for one detection		IoU for one detection	
		recall	precision	recall	precision
LCA	B	**0.72**	0.80	**0.73**	0.64
	BG	0.64	0.714	0.61	0.64
	NL	0.68	**0.82**	0.51	**0.72**
	BGNL	0.65	0.79	0.63	0.71
	Cong et al.	0.71	–	–	–
RCA	B	0.68	0.79	0.65	0.65
	BG	**0.70**	**0.84**	**0.68**	**0.74**
	NL	0.65	0.81	0.56	0.68
	BGNL	0.58	0.73	0.51	0.64
	Cong et al.	0.60	–	–	–

indicates that the depth layers of RetinaNet have increased and were not the result of gradient loss or explosion.

Next, Kaiming He et al. [1] use a residual network in CNN in their work. They compare the performance of 18-layer and 34-layer residual nets (ResNets). The baseline architectures are the same as the plain nets described above, with the exception that a shortcut connection is added to each pair of 3 x 3 filters, as shown in Fig. 3. (right). They use identity mapping for all shortcuts and zero-padding for increasing dimensions in the first comparison on Table 2 and Fig. 4 (right). As a result, they have no additional parameters when compared to their plain counterparts.

Table 2 and Fig. 4 provide three major observations for Kaiming et al. First, with residual learning, the situation is reversed: the 34-layer ResNet outperforms the 18-layer ResNet (by 2.8%). Furthermore, the 34-layer ResNet has a significantly lower training error and is generalizable to validation data. This indicates that the degradation problem has been adequately addressed in this setting, and they are able to obtain accuracy gains from increased depth. Second, when compared to its plain counterpart, the 34-layer

Table 2. Top-1 error (%, 10-crop testing) on ImageNet validation. In this case, the ResNets have no additional parameters when compared to their plain counterparts.

# layers	Plain	Resnet
18	27.94	27.88
34	28.54	25.03

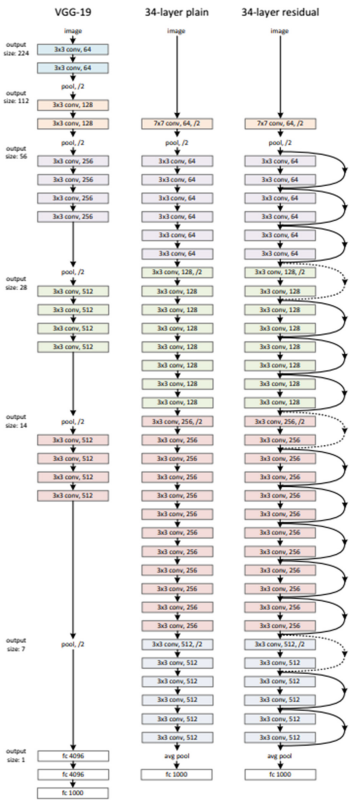


Fig. 3. Example of the overall architecture of Resnet [6]

ResNet reduces top-1 error by 3.5% in Table 2, owing to successfully reduced training error (Fig. 4 right vs. left). This comparison validates residual learning's effectiveness on extremely deep systems. Finally, they note that while the 18-layer plain/residual nets are comparable in accuracy (Table 2), the 18-layer ResNet converges faster (Fig. 4. Right vs. left). When the net is "not too deep" (18 layers in this case), the current SGD solver can still find good solutions to the plain net. In this case, the ResNet facilitates optimization by allowing for faster convergence at an early stage.

Architectures with a deeper bottleneck. Following that, we will go over our ImageNet deeper nets. We modify the building block as a bottleneck design due to concerns about

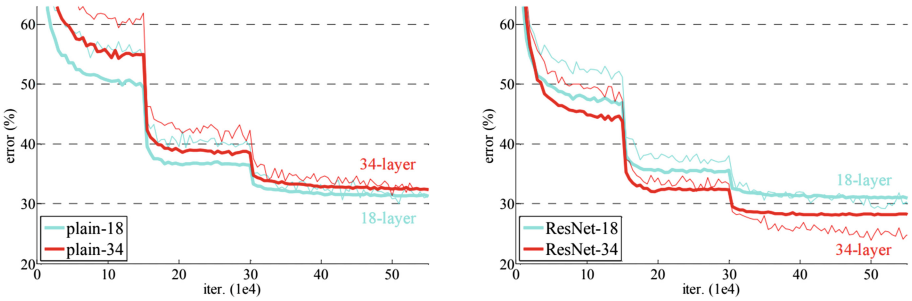
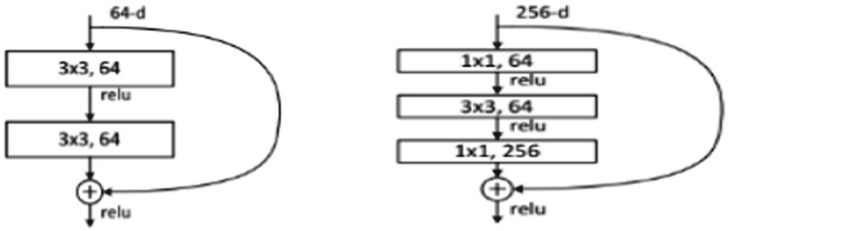


Fig. 4. Palembang ImageNet training. The thin curves represent training error, while the bold curves represent validation error of the center crops. Plain networks with 18 and 34 layers are shown on the left. ResNets with 18 and 34 layers, respectively. The residual networks in this plot have no extra parameters when compared to their plain counterparts.



Left: a building block (on 56×56 feature maps) as in Figure 3 for ResNet34

Right: a "bottleneck" building block for ResNet-50/1

Fig. 5. A deeper residual function F for ImageNet

the amount of training time we can afford. We use a three-layer stack instead of two for each residual function (Fig. 5). The three layers are 11 convolutions, 33 convolutions, and 11 convolutions, with the 11 layers responsible for reducing and then increasing (restoring) dimensions, leaving the 33 layer as a bottleneck with smaller input/output dimensions. Figure 5 depicts an example in which both designs have a comparable time complexity. For bottleneck architectures, parameter-free identity shortcuts are especially important. When the identity shortcut in Fig. 5 (right) is replaced by projection, the time complexity and model size are doubled.

Comparisons with State-of-the-Art Methods.
Kaiming et al. compare their results to the best single-model results previously obtained. Their baseline 34-layer ResNets achieved extremely competitive accuracy. Their 152-layer ResNet has a 4.49% single-model top-5 validation error. This single-model outcome outperforms all previous ensemble outcomes (Table 3). They create an ensemble by combining six models of varying depths (only with two 152-layer ones at the time of submitting). This results in a top-5 error rate of 3.57% on the test set (Table 3). This entry took first place in the 2015 ILSVRC.

Table 3. Ensemble error rates (%). The top-5 errors are on ImageNet's test set and are reported by the test server.

Method	Top-5 err. (test)
VGG [4] (ILSVRC'14)	7.32
GoodLeNet [5] (ILSVRC'14)	6.66
VGG [4] (v5)	6.8
PReLU-net [8]	4.94
BN-inception [9]	4.82
ResNet (ILSVRC'15)	3.57

Kamal HM [6] also runs Resnet for classification. A convolutional neural network model capable of classifying images from the 2012 ILSVRC Imagenet dataset was developed in this study. At the training stage, the CNN model will be trained using the Resnet34 architecture. A learning rate search algorithm will be run prior to beginning training. This algorithm is run after the training is complete to evaluate the new learning rate that must be changed based on the model's loss. The images in the training data will be processed by the CNN model, which will pass each image through every part of the Resnet34 architecture; the result of this process is the prediction result of the images that enter the model. The model will evaluate the existing dataset in the validation data for each epoch to determine how well the model can classify images that have not been seen before during training. The best training model's results will be evaluated and analyzed for performance. Three main scenarios will be created to determine the performance of the CNN architectural model built in image classification and to evaluate the experimental results. The first scenario uses a 64x64 pixel image, which is a smaller resolution than the standard image resolution used in the CNN model. The second scenario uses an image with a resolution of 224 x 224 pixels, which is the image resolution used by various architectures in classifying ImageNet [1, 4]. The third option is to use a smaller image, 64x64, and then retrain with 224x224 pixels. Each of the three main scenarios will be trained with and without augmentation.

The accuracy for augmentation data is 49.32%, while the accuracy for data without augmentation is 50.52%. The accuracy percentage is lower than when using augmentation data in similar image sizes of 64 x 64. In Tables 4, 5, the model using augmentation data has a low accuracy performance, but the overall model performance is quite good. Although the accuracy of using augmentation data is lower, the number of error classes is lower and more evenly distributed. It indicates that depth layers exist in the Resnet architecture and that the losses are not gradient.

The classification of multi-class images on the ImageNet dataset was performed using the Resnet34 architecture in this study. Using the cyclical learning rate method to determine the learning rate where the initial learning rate is 1x10–2 and this value is reduced after the model's accuracy reaches saturation. Based on the results of the simulation, the best accuracy is obtained without the use of augmentation, with an accuracy of 75.82%, as shown in Fig. 6.

Table 4. Errors in the top 5 classes of prediction results on models with an image size of 64x64

#	Classes		Sum of error
1	European Green Lizard	Carolina Anole	18
2	Partridge	Ruffed Grouse	17
3	Smooth Green Snake	Green Mamba	16
4	Saharan Horned Viper	Sidewinder	15
5	European Garden Spider	Barn Spider	14

Table 5. The error of the top 5 classes of prediction results on a model with an image size of 64x64 with augmentation.

#	Classes		Sum of error
1	Tiger Shark	Hammerhead Shark	20
2	Carolina Anole	European Green Lizard	12
3	Saharan Horned Viper	Sidewinder	12
4	Bald Eagle	Kite	12
5	Great White Shark	Hammerhead Shark	12

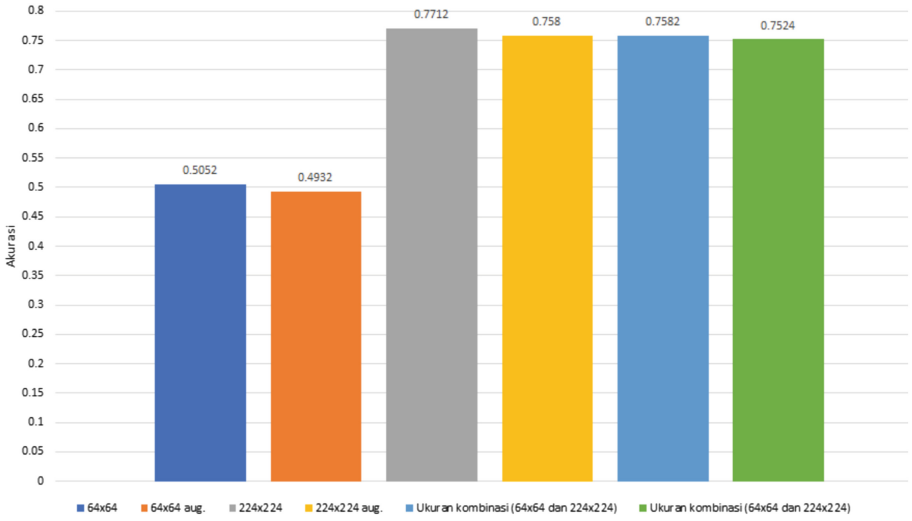


Fig. 6. Accuracy obtained from the 6 scenarios which carried out.

However, the error matrix evaluation results show that the model that does not use augmentation and changes the image size has the best error matrix with the 11 highest errors, despite having a slightly lower accuracy of 75.24%. It can be concluded that (1) augmentation and data augmentation will provide the model with more data, but if not balanced with good tuning, the model will perform poorly. (2) Reducing and then increasing the size of the trained image can make the model predict more evenly while maintaining the same accuracy. To ensure that this scenario can be carried out in the future, it is expected that a larger architecture and a greater number of classes will be used.

3 Discussion

In this post, we will go through three past research that used CNN architecture with residual blocks or Resnet. The results of the first investigation by Kamal Hasan et al. [6] reveal that image classification with the Resnet34 architecture generates superior performance predictions than CNN without residual or plain CNN with augmentation data or not. This suggests that the CNN layers are growing rather than shrinking. In this Resnet34 architecture, 34 convolutions have been formed with 16 blocks of skip connections, and each block has 2 layers plus 1 start and 1 final layer, for a total of 34 layers.

The second study, by Dinis LR. et al. [7], recognizes left and right coronary arteries using Retinanet architecture composed of Resnet50, Feature Pyramid Network, Classifier, and Regressor. On the RCA/LCA, the results show 0.72/0.70 recall, 0.82/0.84 precision with one detection, 0.73/0.68 recall, and 0.72/0.74 precision for five detections. Retinanet, when combined with Resnet50, outperforms other architectures. This condition demonstrates that the Retinanet architecture adds its inner layer while retaining the gradient by including Resnet50 at the start of the Retinanet architecture.

Finally, Kaiming et al. [1] used a comparison of plain CNN and CNN with the residual network (Resnet) to achieve classification performance on ILSVRC 2015 image data with a top 5 error of 3.57%. Kaiming et al. demonstrate greater performance during classification training by adding a residual network called Resnet to the plain CNN. This suggests that the plain CNN with Resnet has grown its inner layer while maintaining its gradient. So, Resnet outperforms plain CNN.

As described in [13, 14], the CNN architecture can be evolved into numerous additional architectures, and the basic architecture of CNN can be improved by adding residual networks or residual blocks to increase performance or reduce error.

4 Conclusion and Future Work

We conclude that using a residual network on CNN improves architectural performance in terms of detection and classification tasks. With block residuals, several other architectures derived from the basic CNN architecture can be modified. We did not investigate other research for segmentation in this article, nor did we attempt to segment images by incorporating a residual network into an architecture. In terms of future work, it is critical to forecast using an image dataset segmentation task.

References

1. K. He, X. Zhang, S. Ren, and J. Sun, 2015, Deep residual learning for image recognition. arXiv:1512.03385, Dec 2015. arXiv: 1512.03385.
2. Y. Hu, A. E. G. Huber, J. Anumula, and S. Liu. Overcoming the vanishing gradient problem in plain recurrent networks. CoRR, abs/1801.06105, 2018.
3. A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. Communications of the ACM, 60(6):84–90, May 2017
4. K. Simonyan and A. Zisserman, 2014, Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556 [cs], Sep 2014. arXiv: 1409.1556.
5. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. arXiv:1409.4842 [cs], Sep 2014. arXiv: 1409.4842
6. Kamal Hasan Mahmud , Adiwijaya , Said Al Faraby, 2019, Klasifikasi Citra Multi-Kelas Menggunakan Convolutional Neural Network, e-Proceeding of Engineering : Vol.6, No.1 April 2019, Page 2127.
7. Dinis L. Rodrigues , Miguel Nobre Menezes , Fausto J. Pinto , and Arlindo L. Oliveira, 2021, Automated Detection of Coronary Artery Stenosis in X-ray Angiography using Deep Neural Networks, arXiv:2103.02969v1 [eess.IV]4 Mar 2021.
8. K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In ICCV, 2015.
9. S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In ICML, 2015.
10. S. Hochreiter and J. Schmidhuber. Long short-term memory. Neural computation, 9(8):1735–1780, 1997.
11. K. Chatfield, V. Lempitsky, A. Vedaldi, and A. Zisserman. The devil is in the details: an evaluation of recent feature encoding methods. In BMVC, 2011.
12. Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. IEEE Transactions on Neural Networks, 5(2):157–166, 1994.
13. Alireza Zaeemzadeh, Nazanin Rahnavard, Mubarak Shah, 2020, Norm-Preservation: Why Residual Networks Can Become Extremely Deep?, UNIVERSITY OF BIRMINGHAM, 15 June 2020, IEEE Xplore.
14. Motahareh Aghalari, Ali Aghagolzadeh, Mehdi Ezoji, Brain tumor image segmentation via asymmetric/symmetric UNet based on two-pathway-residual blocks, Biomedical Signal Processing and Control, Volume 69, August 2021, 102841.