

# Perbandingan GoogleNet, ResNet, dan DenseNet pada Dataset CIFAR-10

Silverius Sony Lembang  
Muhammad Takdim  
Richard Enrico Sulieanto

## CONTENTS

<b>I</b>	<b>Introduction</b>	<b>1</b>
<b>II</b>	<b>Problem Definition</b>	<b>1</b>
<b>III</b>	<b>CNN Architecture</b>	<b>1</b>
III-A	GoogLeNet . . . . .	1
III-B	Resnet . . . . .	1
III-C	DenseNet . . . . .	2
<b>IV</b>	<b>Research Methodology</b>	<b>2</b>
IV-A	Dataset . . . . .	2
IV-B	Data Augmentation . . . . .	3
IV-C	Model . . . . .	3
<b>V</b>	<b>Analysis and Interpretation</b>	<b>3</b>
<b>VI</b>	<b>Conclusions and Recommendations</b>	<b>3</b>
	<b>References</b>	<b>3</b>

## LIST OF FIGURES

1	Inception Block . . . . .	1
2	Arsitektur GoogleNet . . . . .	2
3	Resnet Block . . . . .	2
4	Arsitektur DenseNet . . . . .	2

## LIST OF TABLES

I	Konfigurasi <i>optimizer</i> dan <i>hyperparameter</i> untuk masing-masing model . . . . .	3
II	Performa dari masing-masing model CNN . . . . .	3

# Perbandingan GoogLeNet, ResNet, dan DenseNet pada Dataset CIFAR-10

**Abstract**—Arsitektur Convolutional Neural Network (CNN) terus berkembang seiring berjalannya waktu sejak pertama kali dikembangkan oleh Yann LeCun pada tahun 1995. Hingga saat ini terdapat begitu banyak arsitektur CNN yang telah dikembangkan. Saat ini, setidaknya terdapat 3 arsitektur CNN yang umum digunakan yakni GoogLeNet, ResNet dan DenseNet. Akan tetapi, pemilihan arsitektur CNN yang sesuai terkadang menimbulkan masalah tersendiri. Oleh karena itu, dalam *technical report* ini, kami membandingkan ketiga arsitektur CNN tersebut dengan menggunakan dataset CIFAR-10 untuk melihat *validation accuracy* dan *test accuracy* pada masing-masing arsitektur. Hasil yang diperoleh menunjukkan bahwa arsitektur ResNet mengungguli kedua arsitektur CNN lainnya pada bagian *validation accuracy* dan *test accuracy*.

## I. INTRODUCTION

Convolutional Neural Network (CNN) merupakan salah satu jenis dari *neural network* yang berfokus pada data yang berupa gambar. CNN cukup populer digunakan khususnya dalam menyelesaikan masalah-masalah yang berkaitan dengan gambar seperti klasifikasi gambar, segmentasi gambar, dan deteksi objek. Keunggulan dari CNN adalah kemampuannya untuk mengekstrak fitur-fitur dari sebuah gambar sehingga kita tidak perlu melakukan ekstraksi fitur secara manual yang tentunya membutuhkan waktu yang lama dan memerlukan pengetahuan tertentu untuk memilih fitur-fitur pada gambar. CNN memiliki banyak arsitektur yang terus berkembang hingga saat ini. Diantaranya adalah GoogLeNet [1], ResNet [2], dan DenseNet [4]. Dalam *report* ini, kami akan membandingkan tiga arsitektur CNN yakni GoogLeNet/Inception, ResNet, dan DenseNet. Adapun dataset yang akan digunakan adalah CIFAR-10.

## II. PROBLEM DEFINITION

Dalam *technical report* ini, kami berfokus untuk membandingkan beberapa arsitektur CNN yakni GoogLeNet, ResNet dan DenseNet pada dataset CIFAR-10. Secara khusus, pertanyaan-pertanyaan berikut akan dibahas dalam *technical report* ini.

- Bagaimana performa dari masing-masing arsitektur CNN pada dataset CIFAR-10?
- Arsitektur mana yang memberikan performa terbaik pada dataset CIFAR-10?

## III. CNN ARCHITECTURE

### A. GoogLeNet

Pada tahun 2014, GoogLeNet memenangkan ImageNet Large-Scale Visual Recognition Challenge 2014 (ILSVRC14), mengusulkan sebuah struktur yang menggabungkan kelebihan

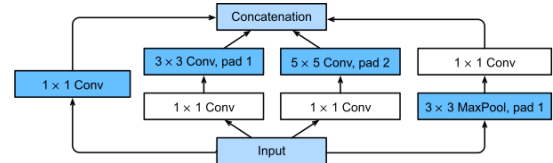


Fig. 1: Inception Block

NiN dan paradigma blok berulang [1]. Salah satu fokus dari GoogLeNet adalah mencari ukuran kernel konvolusi terbaik untuk menghindari masalah overfit dan meminimalisir biaya komputasi yang terjadi ketika menambah ukuran Neural Network untuk meningkatkan performa. Cara mendasar untuk menyelesaikan kedua masalah tersebut adalah dengan beralih dari arsitektur yang sepenuhnya terhubung ke arsitektur yang jarang terhubung, bahkan di dalam konvolusi. Arsitektur Inception dipilih setelah dilakukan penyetselan parameter dan berhasil menunjukkan performa yang optimal.

Berdasarkan Fig. 1, blok inception terdiri atas 4 jalur paralel. Tiga jalur pertama menggunakan lapisan konvolusi dengan ukuran 1x1, 3x3, dan 5x5 untuk mengekstrak informasi dari ukuran spasial yang berbeda. Dua jalur tengah melakukan konvolusi 1x1 pada input untuk mengurangi jumlah saluran, mengurangi kerumitan model. Jalur keempat menggunakan lapisan pooling maksimum 3x3, diikuti oleh lapisan konvolusi 1x1 untuk mengubah jumlah saluran. Keempat jalur semuanya menggunakan padding yang sesuai untuk memberikan input dan output dengan tinggi dan lebar yang sama. Akhirnya, output di sepanjang setiap jalur digabungkan di sepanjang dimensi saluran dan terdiri dari output blok. Beberapa diantaranya yang cukup populer adalah

### B. Resnet

ResNet merupakan jenis deep learning yang diperkenalkan pada tahun 2015 oleh Kaiming He, Xiangyu Zhang, Shaoqing Ren dan Jian Sun. Hingga saat ini paper ResNet merupakan salah satu makalah AI yang paling banyak dikutip, dan telah menjadi dasar untuk jaringan saraf dengan lebih dari 1.000 lapisan [3]. Inti dari ResNet yang diusulkan adalah gagasan bahwa setiap lapisan tambahan harus lebih mudah memuat fungsi identitas sebagai salah satu elemennya. Pertimbangan ini agak mendalam tetapi mereka mengarah pada solusi yang sangat sederhana yaitu blok residual. Alih-alih memodelkan  $xl + 1 = F(xl)$ , Dengan memodelkan  $xl + 1 = xl + F(xl)$  dimana F adalah sebuah pemetaan non-linier (biasanya urutan modul NN seperti konvolusi, fungsi aktivasi, dan normalisasi).

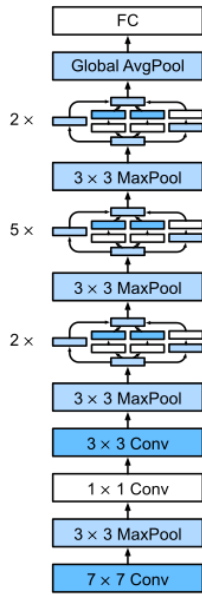


Fig. 2: Arsitektur GoogleNet

Jika kita melakukan backpropagation pada koneksi residual tersebut, kita memperoleh:

$$\frac{\partial x_{l+1}}{\partial x_l} = \mathbf{I} + \frac{\partial F(x_l)}{\partial x_l}$$

Bias terhadap matriks identitas menjamin propagasi gradien yang stabil menjadi kurang terpengaruh dengan sendirinya. Ada banyak varian ResNet yang diusulkan, yang sebagian besar menyangkut fungsi, atau operasi yang diterapkan pada penjumlahan. Di sini kami akan menggunakan varian blok ResNet asli dan blok ResNet Pra-Aktivasi.

Pada Fig.3 dapat kita lihat bahwa Blok ResNet asli menerapkan fungsi aktivasi non-linear, biasanya ReLU, setelah koneksi lewat. Sebaliknya, blok ResNet pra-aktivasi menerapkan non-linearitas di awal F. Keduanya memiliki kelebihan dan kekurangan masing-masing. Namun, untuk jaringan yang sangat dalam, pra-aktivasi ResNet telah menunjukkan kinerja yang lebih baik karena aliran gradien dijamin memiliki matriks identitas seperti yang dihitung di atas, dan tidak dirugikan oleh aktivasi non-linier yang diterapkan padanya. Sebagai perbandingan, di notebook ini, kami menerapkan kedua jenis ResNet sebagai jaringan dangkal.

### C. DenseNet

DenseNet merupakan arsitektur yang diciptakan oleh Gao Huang, Zhuang Liu dan Laurens van der Maaten [4]. DenseNet berfokus masalah yang timbul seiring dengan bertambahnya *layer* dimana informasi yang terdapat pada input akan menghilang seiring dengan informasi tersebut melewati banyak *layer* pada *neural network*. Pada DenseNet, hal ini dilakukan dengan membuat suatu *simple connectivity pattern* yang menghubungkan semua *layer* secara langsung satu sama lain.

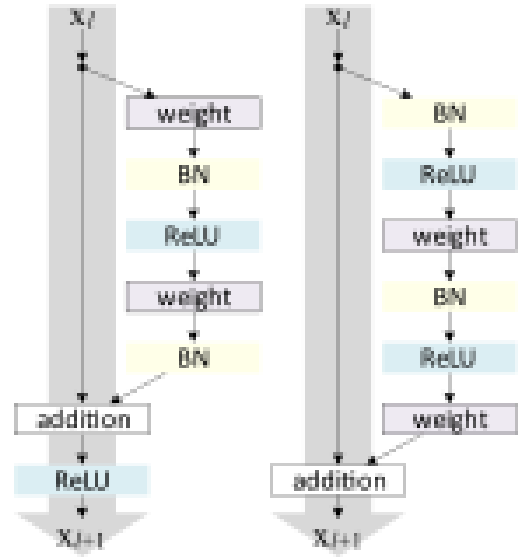


Fig. 3: Resnet Block

Untuk menjaga *nature* dari *feed-forward* setiap *layer* akan menerima input tambahan dari semua *layer* sebelumnya dan akan meneruskan *feature-maps* dari *layer* tersebut ke semua *layer* berikutnya [4]. *Network* pada DenseNet akan dibagi menjadi beberapa dense blocks yang saling terhubung. Diantara setiap dense blocks, terdapat transition layers yang berguna untuk melakukan proses konvolusi dan pooling.

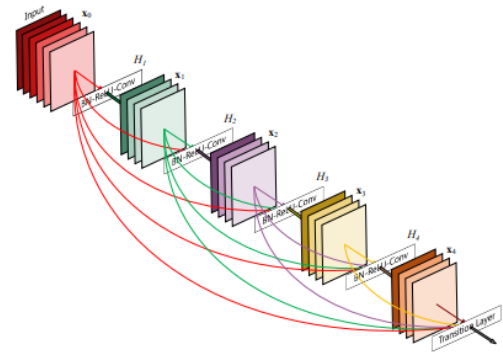


Fig. 4: Arsitektur DenseNet

## IV. RESEARCH METHODOLOGY

### A. Dataset

Dalam technical report ini, kami menggunakan dataset CIFAR-10. CIFAR-10 merupakan dataset yang berisi kumpulan gambar yang terdiri dari 60000 gambar yang dibagi menjadi 10 kelas dimana masing-masing kelas terdiri dari 6000 gambar. Masing-masing data merupakan gambar berwarna

berukuran 32x32. Dataset ini terdiri dari 50000 data latih dan 10000 data uji. [5]. 10 kelas yang terdapat pada dataset CIFAR-10 adalah *airplane, automobile, cat, deer, dog, frog, horse, ship, truck*. Sebelum dataset tersebut digunakan, maka perlu dilakukan normalisasi pada dataset tersebut menggunakan nilai mean dan std dari dataset tersebut.

### B. Data Augmentation

Untuk mencegah terjadinya overfitting, maka akan dilakukan data augmentation. Pada tahap ini, akan dilakukan 2 proses data augmentasi. Pertama, proses data augmentasi dilakukan dengan melakukan flip pada gambar secara horizontal sehingga diperoleh gambar baru dengan perspektif berbeda. Umumnya, proses *flip* tidak akan mengubah ukuran dari gambar tersebut. Proses yang augmentasi kedua adalah melakukan *resize* pada gambar. Proses *resize* dapat mengubah skala dan *aspect ratio* dari gambar. Oleh karena itu, setelah proses ini, gambar akan dicrop dengan ukuran 32x32 agar selanjutnya dapat dimasukkan kedalam model CNN.

### C. Model

Untuk membangun model CNN, kami menggunakan tiga arsitektur CNN yakni GoogleNet, ResNet, dan DenseNet. Untuk arsitektur ResNet, akan digunakan 2 jenis block yakni original ResNet Block dan Pre-Activation ResNet Block. Model GoogleNet pada penelitian ini akan menggunakan *activation function* yakni ReLU. Model GoogleNet menggunakan *optimizer* Adam dengan *learning rate* 0.001 dan *weight decay* 0.0001. Untuk model ResNet, *activation function* yang digunakan adalah ReLU dan *Stochastic Gradient Descent* (SGD). Selain itu, *hyperparameter* yang digunakan adalah *learning rate* yang digunakan sebesar 0.1, *momentum* sebesar 0.9 dan *weight decay* sebesar 0.0001. *Optimizer* dan *hyperparameter* tersebut berlaku untuk 2 jenis ResNet block yang digunakan. Untuk model DenseNet, *optimizer* yang digunakan adalah Adam dan *activation function* yang digunakan adalah ReLU. Selain itu, *hyperparameter* yang digunakan adalah *learning rate* yang digunakan sebesar 0.001 dan *weight decay* sebesar 0.0001. Secara sederhana, *optimizer* dan *hyperparameter* yang digunakan dapat dilihat pada tabel berikut.

Model	Activation	Optimizer	lr	Momentum	Weight Decay
GoogleNet	ReLU	Adam	0.001	-	0.0001
ResNet	ReLU	SGD	0.1	0.9	0.0001
ResNetPreAct	ReLU	SGD	0.1	0.9	0.0001
DenseNet	ReLU	Adam	0.001	-	0.0001

TABLE I: Konfigurasi *optimizer* dan *hyperparameter* untuk masing-masing model

## V. ANALYSIS AND INTERPRETATION

Setelah melakukan proses pembangun model dan pengujian model, performa dari masing-masing model dapat dilihat pada tabel berikut.

Model	Val Accuracy	Test Accuracy	Num Parameters
GoogleNet	90.40%	89.70%	260,650
ResNet	91.84%	91.06%	272,378
ResNetPreAct	91.80%	91.07%	272,250
DenseNet	90.72%	90.23%	239,146

TABLE II: Performa dari masing-masing model CNN

Secara umum, dapat dilihat bahwa semua model memberikan performa yang cukup baik. Dari hasil tersebut, terlihat bahwa GoogleNet memperoleh performa yang paling rendah pada bagian *validation accuracy* dan *test accuracy* masing-masing sebesar 90.40% dan 89.70%. DenseNet memberikan performa yang sedikit lebih baik jika dibandingkan dengan GoogleNet dimana terjadi peningkatan sebesar 0.32% pada *validation accuracy* dan 0.53% pada *test accuracy*. Terakhir, kita dapat melihat performa dari masing-masing model ResNet mampu mengungguli model GoogleNet dan DenseNet dimana terjadi perbedaan hingga lebih dari 1% pada bagian *validation accuracy* dan perbedaan lebih dari 1% untuk *test accuracy* dengan GoogleNet dan perbedaan mendekati 1% jika dibandingkan dengan DenseNet.

## VI. CONCLUSIONS AND RECOMMENDATIONS

Berdasarkan hasil yang diperoleh, ResNet memberikan performa yang terbaik jika dibandingkan dengan 2 arsitektur lainnya. Jika kita ingin menerapkan arsitektur untuk *task* yang lebih kompleks dengan gambar yang lebih besar, maka kita dapat melihat perbedaan yang besar antara GoogleNet dan arsitektur *skip connection* seperti ResNet dan DenseNet. Oleh karena itu, apabila kita ingin memilih arsitektur untuk *task* yang baru, maka memilih arsitektur *ResNet* merupakan langkah yang baik melihat performanya pada dataset CIFAR-10.

## REFERENCES

- [1] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... Rabinovich, A. (2015). Going deeper with convolutions. *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1–9).
- [2] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770–778).
- [3] Crew, B. (2020, July 13). *Google scholar reveals its most influential papers for 2020*. Nature Index. Retrieved April 29, 2022, from <https://www.natureindex.com/news-blog/google-scholar-reveals-most-influential-papers-research-citations-twenty-twenty>
- [4] 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).
- [5] Krizhevsky, A. (n.d.). CIFAR-10 and CIFAR-100 datasets. Retrieved April 29, 2022, from <https://www.cs.toronto.edu/~kriz/cifar.html>