

Inception, ResNet and DenseNet Menggunakan Pytorch Lightning

Indah Nurawaliah
Muhammad Ikhsan Ilma
Muhammad Farid Aqthar
Sistem Informasi-Universitas Hasanuddin

CONTENTS

I	Pendahuluan	1
II	Defini Masalah	1
III	Arsitektur	1
III-A	GoogleNet/Inception	1
III-B	ResNet	1
III-C	DenseNet	2
IV	Metode Penelitian	2
IV-A	Sumber Data	2
IV-B	Alur Kerja	2
IV-B1	Normalisasi dan Augmentasi	2
IV-B2	Visualisasi	2
IV-B3	Data Training	2
V	Analisis dan Interpretasi	3
VI	Kesimpulan dan Saran	3
	References	3

LIST OF FIGURES

1	GoogleNet	1
2	ResNet	2
3	DenseNet	2
4	Hasil Performa tiap arsitektur	3

LIST OF TABLES

Inception, ResNet and DenseNet Menggunakan Pytorch Lightning

Abstract—Ada berbagai macam arsitektur Convolutional Neural Network(CNN) Beberapa yang paling berdampak, dan masih relevan hingga saat ini, adalah arsitektur GoogleNet/Inception, ResNet, dan DenseNet. Semuanya adalah model terancang saat diusulkan, dan ide inti dari jaringan ini adalah fondasi untuk sebagian besar arsitektur terancang saat ini.

I. PENDAHULUAN

Convolutional Neural Network (CNN) adalah salah satu jenis neural network yang biasa digunakan pada data image. CNN bisa digunakan untuk mendeteksi dan mengenali object pada sebuah image. CNN memanfaatkan proses konvolusi dengan mengerjakan sebuah kernel konvolusi (filter) berukuran tertentu ke sebuah gambar, komputer mendapatkan informasi representatif baru dari hasil perkalian bagian gambar tersebut dengan filter yang digunakan. Convolutional Neural Network (CNN) termasuk dalam jenis deep learning karena kedalaman jaringannya. Deep learning adalah cabang dari machine learning yang dapat mengajarkan komputer untuk melakukan pekerjaan selayaknya manusia, seperti komputer dapat belajar dari proses training. CNN memiliki berbagai macam arsitektur yang dapat di gunakan, beberapa yang paling berdampak, dan masih relevan hingga saat ini adalah GoogleNet/Inception ResNet dan DenseNet.

II. DEFINI MASALAH

CNN memiliki fungsi untuk melakukan ekstraksi fitur. Fitur-fitur perlu didapatkan guna proses atau tugas seperti klasifikasi, clustering ataupun regresi. Pada machine learning konvensional dilakukan ekstraksi fitur manual, artinya ditentukan terlebih dahulu fitur-fitur yang diekstraksi. Beberapa arsitektur CNN yang sedang populer adalah inception resNet dan denseNet. Very deep convolutional networks telah menjadi pusat pengembangan dalam performa image recognition belakangan ini. Contohnya adalah arsitektur inception yang menghasilkan performa yang sangat baik dengan komputasi yang relatif rendah. densenet memiliki arsitektur yang menghubungkan setiap layer ke layer lain dengan cara feed-forward. ResNet memiliki konsep shortcut connection yang mencegah sistem dari kehilangan banyak informasi selama training dilakukan.

Tujuan dari laporan ini adalah untuk memahami arsitektur pada Convolutional Neural Network (CNN) dan mempelajari cara mengimplementasikannya menggunakan arsitektur inception resNet dan denseNet menggunakan pytorch lightning. Serta membandingkan akurasi yang dihasilkan pada setiap arsitektur

III. ARSITEKTUR

Convolution Neural Network (CNN) adalah perkembangan metode dari Multi Layer Perception (MLP). Namun, CNN memiliki jumlah dimensi yang lebih banyak dibandingkan dengan MLP. CNN memiliki masukan (input) array mulai dari dua dimensi hingga lebih. Pada laporan ini menggunakan beberapa arsitektur CNN yaitu inception/googleNet, resNet, dan denseNet.

A. GoogleNet/Inception

Googlenet adalah sebuah model dan arsitektur berdasarkan modifikasi CNN. Arsitektur ini dikembangkan oleh Google dan telah melakukan training dengan jutaan gambar. Karena basisnya dari CNN, maka pengembangannya tidak berbeda jauh dari modifikasi yang dilakukan Alexnet. GoogleNet memiliki layer dan convolution dengan jumlah banyak. Secara ringkas, Googlenet memiliki lima layer hingga 13 layer.

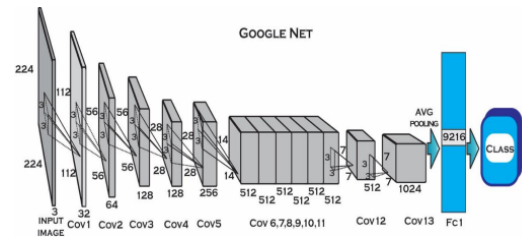


Fig. 1. GoogleNet

B. ResNet

Residual neural network atau yang biasa disebut sebagai Resnet adalah salah satu jenis arsitektur yang cukup populer, arsitektur ini dibuat oleh Kaiming He et al. Arsitektur ini cukup revolusioner pada saat itu karena arsitektur ini menjadi state of the art pada saat itu tidak hanya dalam klasifikasi, namun dalam semua kategori termasuk object detection, dan semantic segmentation. Arsitektur CNN yang memiliki kedalaman tinggi adalah salah satu hal penting dalam membangun model CNN yang memiliki performa yang baik, namun model CNN yang memiliki kedalaman yang tinggi juga memiliki masalah, yaitu vanishing gradient problem, yaitu suatu keadaan dimana hasil gradien yang dipelajari oleh model, tidak dapat mencapai layer pertama karena mengalami perkalian berkali-kali sehingga layer pertama tidak menerima gradien apa-apa, atau secara singkatnya, hal ini menyebabkan suatu CNN tidak dapat belajar dari error yang telah dikalkulasi. Resnet memiliki

berbagai macam jenis arsitektur, mulai dari 18, 34, 50, 101, sampai 152 layer. Pada penelitian ini digunakan arsitektur Resnet34, yaitu arsitektur resnet yang memiliki 34 layer, arsitektur ini dipilih karena arsitektur ini memiliki performa yang baik pada kompetisi ILSVRC dan karena keterbatasan hardware peneliti, maka arsitektur yang dipilih hanya memiliki 34 layer. Dari dataset keseluruhan, akan digunakan data training untuk melatih model CNN yang dibuat, dan data validasi untuk mengukur performa model yang dibuat. Performa dari sistem yang dibangun akan diukur dengan menggunakan metrik evaluasi akurasi, training loss, validation loss dan melihat matriks kekeliruan.

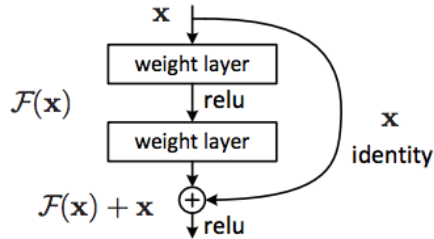


Fig. 2. ResNet

Blok ini menambahkan suatu jalan pintas yang berfungsi sebagai fungsi identitas, yang secara tidak langsung akan melewati proses training untuk satu layer atau lebih, sehingga membuat sesuatu yang bernama residual block.

C. DenseNet

Convolutional Neural Network (CNN) dirancang untuk memproses suatu data yang ada dalam bentuk banyak array. Ada berbagai macam bentuk CNN yaitu 1D untuk sinyal dan urutan biasanya digunakan untuk bahasa, 2D untuk gambar atau suara; dan 3D untuk video atau gambar volumetric.

Dense Convolutional Network (DenseNet), yang menghubungkan setiap lapisan/blok ke setiap lapisan/blok lainnya dengan cara umpan maju. Sedangkan jaringan konvolusional tradisional dengan L lapisan memiliki koneksi L - satu antara setiap lapisan dan lapisan berikutnya jaringan memiliki koneksi langsung L (L + 1) / 2. DenseNet memiliki beberapa keunggulan menarik: meringankan masalah gradien-gradien, memperkuat penyebaran fitur, mendorong penggunaan kembali fitur, dan secara substansial mengurangi jumlah parameter.

Pada gambar 3 untuk setiap komposisi lapisan menggunakan batch normalization, ReLU activation dan convolution dengan filter 3x3. Pada setiap blok ada masukan berupa matriks sesuai dengan pixel citra kemudian masuk ke proses batch normalization untuk mengurangi adanya overfitting pada saat proses training, ReLU activation digunakan untuk mengubah nilai x menjadi 0 jika nilai x tersebut bernilai negatif, sedangkan sebaliknya untuk nilai x tetap dipertahankan apabila nilai tidak kurang dari 0, convolution dengan filter 3x3 proses citra matriks yang sudah dilakukan operasi ReLU activation akan dikalikan dengan matriks convolution dengan filter 3x3 dan

keluaran yang dihasilkan berupa nilai matriks yang sudah di proses sebelumnya.

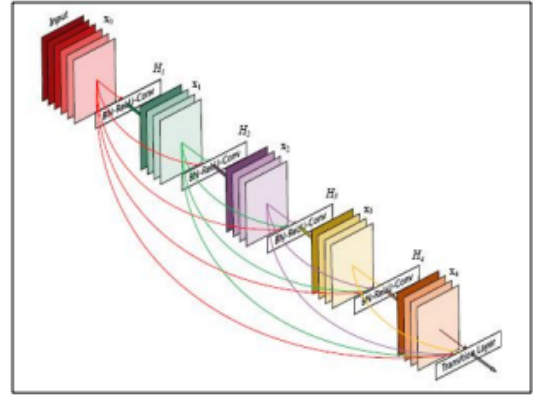


Fig. 3. DenseNet

IV. METODE PENELITIAN

A. Sumber Data

Dalam laporan ini menggunakan dataset CIFAR-10 yang terdiri dari 60.000 gambar berwarna 32x32 dalam 10 kelas, dengan 6000 gambar per kelas. Ada 50000 gambar pelatihan dan 10.000 gambar uji. Dataset dibagi menjadi lima batch pelatihan dan satu batch pengujian, masing-masing dengan 10.000 gambar. Kumpulan tes berisi tepat 1000 gambar yang dipilih secara acak dari setiap kelas. Kumpulan pelatihan berisi gambar yang tersisa dalam urutan acak, tetapi beberapa kumpulan pelatihan mungkin berisi lebih banyak gambar dari satu kelas daripada yang lain.

B. Alur Kerja

1) *Normalisasi dan Augmentasi*: Pertama yang dilakukan adalah normalize yang akan menormalkan data kami. Selain itu, kami akan menggunakan augmentasi data untuk mengurangi risiko overfitting dan membantu untuk menggeneralisasi lebih baik. Secara khusus, kami akan menerapkan dua augmentasi acak.

Kemudian melakukan Augmentasi kedua untuk men-skalakan gambar dalam rentang kecil, sementara pada akhirnya mengubah rasio aspek, dan memotongnya ke ukuran sebelumnya. Oleh karena itu, nilai piksel aktual berubah sementara konten atau semantik keseluruhan gambar tetap sama.

2) *Visualisasi*: Setelah melakukan augmentasi maka akan dilakukan visualisasi untuk memvisualisasikan beberapa gambar dari set pelatihan, dan bagaimana tampilannya setelah augmentasi data acak.

3) *Data Training*: Pada proses ini akan dilakukan training dan testing pada setiap model arsitektur Convolutional Neural Network(CNN) yaitu googleNet ResNet dan DenseNet.

V. ANALISIS DAN INTERPRETASI

Dalam laporan ini dilakukan proses pengujian dataset CIFAR10 menggunakan arsitektur Convolutional Neural Network(CNN) yaitu googleNet ResNet dan DenseNet. Mengklasifikasikan gambar menggunakan citra data training yang dilakukan menggunakan PyTorch Lightning dengan 200 epoch.

Pada arsitektur googleNet konsisten dengan multiple Inception blok dengan maks pooling untuk mengurangi tinggi dan lebar fitur. GoogleNet asli dirancang untuk ukuran gambar ImageNet (224x224 piksel) dan memiliki hampir 7 juta parameter. Pada saat training dataset CIFAR10 dengan ukuran gambar 32x32, kita tidak memerlukan arsitektur yang berat, dan sebaliknya, menerapkan versi yang lebih kecil. Jumlah saluran untuk pengurangan dimensi dan keluaran per filter (1x1, 3x3, 5x5, dan pooling maks) perlu ditentukan secara manual dan dapat diubah. Secara umum memiliki filter terbanyak untuk konvolusi 3x3, karena filter tersebut cukup kuat untuk mempertimbangkan konteks sementara membutuhkan hampir sepertiga dari parameter konvolusi 5x5.

Pada arsitektur resNet. Satu perbedaan pada train GoogleNet adalah menggunakan SGD dengan momentum optimizer Adam. Optimizer Adam sering menghasilkan akurasi yang sedikit lebih buruk pada ResNets. Tidak jelas mengapa kinerja Adam lebih buruk dalam konteks ini, tetapi satu penjelasan yang mungkin terkait dengan permukaan kerugian ResNet. ResNet telah terbukti menghasilkan permukaan kerugian yang lebih halus daripada jaringan tanpa melewati koneksi.

Pada arsitektur denseNet untuk menentukan jumlah lapisan, kami menggunakan notasi yang sama seperti di ResNets dan meneruskan daftar int yang mewakili jumlah lapisan per blok. Setelah setiap blok padat kecuali yang terakhir, kami menerapkan lapisan transisi untuk mengurangi dimensi sebesar 2. Terakhir, dilakukan train network. Berbeda dengan ResNet, DenseNet tidak menunjukkan masalah apa pun dengan Adam. Hyperparameter lainnya dipilih untuk menghasilkan jaringan dengan ukuran parameter yang sama seperti ResNet dan GoogleNet. Umumnya, ketika merancang jaringan yang sangat dalam, parameter DenseNet lebih efisien daripada ResNet dalam mencapai kinerja yang serupa. Berikut adalah hasil loss dan accuracy pada DenseNet.

Berikut adalah hasil dari performa tiap-tiap model arsitektur:

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

Fig. 4. Hasil Performa tiap arsitektur

VI. KESIMPULAN DAN SARAN

Secara umum, kita dapat menyimpulkan bahwa ResNet adalah arsitektur yang sederhana namun kuat. Jika kita akan menerapkan model pada tugas yang lebih kompleks dengan

gambar yang lebih besar dan lebih banyak lapisan di dalam jaringan, kemungkinan kita akan melihat kesenjangan yang lebih besar antara GoogleNet dan arsitektur skip-connection seperti ResNet dan DenseNet. Perbandingan dengan model yang lebih dalam di CIFAR10 misalnya dapat ditemukan di sini. Menariknya, DenseNet mengungguli ResNet asli pada pengaturan mereka tetapi berada di belakang ResNet Pra-Aktivasi. Model terbaik, Jaringan Jalur Ganda sebenarnya merupakan kombinasi dari ResNet dan DenseNet yang menunjukkan bahwa keduanya menawarkan keuntungan yang berbeda.

Biasanya, memulai dengan ResNet adalah ide yang bagus mengingat kinerja yang unggul dari kumpulan data CIFAR dan implementasinya yang sederhana. Selain itu, untuk nomor parameter yang kami pilih di sini, ResNet adalah yang tercepat karena DenseNet dan GoogleNet memiliki lebih banyak lapisan yang diterapkan secara berurutan dalam implementasi primitif kami. Namun, jika Anda memiliki tugas yang sangat sulit, seperti segmentasi semantik pada gambar HD, disarankan menggunakan varian ResNet dan DenseNet yang lebih kompleks.

REFERENCES

- [1] L. Deng, O. Abdel-Hamid and D. Yu "A deep convolutional neural network using heterogeneous pooling for trading acoustic invariance with phonetic confusion," 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, 2013, pp. 6669-6673. Available: <http://www.vmsk.org/Layman.pdf>. [Accessed: Dec. 3, 2003].
- [2] Hu, Wei, et al. "Deep convolutional neural networks for hyperspectral image classification." *Journal of Sensors* 2015 (2015).
- [3] W. Setiawan, M. . Utoyo, and R. Rulaningtyas, "Classification of neo-vascularization using convolutional neural network model," *TELKOMNIKA*, vol. 17, no. 1, pp. 463-473, 2019.
- [4] M. K. Hasan, Adiwijaya , S. A. Faraby "Klasifikasi Citra Multi-Kelas Menggunakan Convolutional Neural Network", *e-Proceeding of Engineering* : Vol.6, No.1 April 2019 — Page 2127, ISSN : 2355-9365.
- [5] S. I. Ahmad "arsitektur Convolutional Neural Network (CNN) Untuk Klasifikasi Jenis Dan Kesegaran Buah Pada Neraca Buah", Tesis – If185401, Available: https://repository.its.ac.id/73567/1/05111850010020-Master_Thesis.pdf
- [6] Pardede Jasman, D. . Putra, " Implementasi DenseNet Untuk Mengidentifikasi Kanker Kulit Melanoma", *Jurnal Teknik Informatika dan Sistem Informasi* Volume 6 Nomor 3 Desember 2020, p-ISSN : 2443-2210, e-ISSN : 2443-2229