

Inception, ResNet, and DenseNet

Indira Septianita Larasati (H071191023)
Sakinah Yunus (H071191046)
A. Muh. Rizqullah Awaluddin (H071191053)
Computer Science
Hasanuddin University

CONTENTS

I	Introduction	1
II	Related Work	1
III	Architecture	1
III-A	Inception	1
III-B	ResNet	2
III-C	DenseNet	2
IV	Research Metodology	3
IV-A	Dataset	3
IV-B	Data Augmentation	3
IV-C	Model	3
IV-D	Pytorch Lightning	3
V	Conclusion	3

Inception, ResNet, and DenseNet

Abstract—Saat ini deep learning menarik banyak perhatian dalam hal pengembangan sistem yang cepat dan akurat, otomatis dalam hal pengidentifikasian gambar, contohnya adalah *Convolutional Neural Network* (CNN). CNN adalah salah satu jenis neural network yang biasa digunakan pada data image. Pemilihan arsitektur CNN yang sesuai terkadang menimbulkan masalah tersendiri. Implementasi ketiga model yang akan dilakukan yakni Inception, ResNet, dan DenseNet dengan menggunakan dataset CIFAR. Hasil uji coba menunjukkan hasil untuk model GoogleNet dengan akurasi 90.40%, model ResNet dengan akurasi 91.84%, dan model DenseNet dengan akurasi 90.72%.

I. INTRODUCTION

Convolutional Neural Networks (CNN) adalah metode pembelajaran mesin dengan meniru cara kerja sistem saraf otak manusia. CNN dapat menerima input berupa gambar, menentukan aspek atau obyek apa saja dalam sebuah gambar yang bisa digunakan mesin untuk “belajar” mengenali gambar, dan membedakan antara satu gambar dengan yang lainnya. Dengan arsitektur seperti itu, CNN dapat dilatih untuk memahami detail sebuah gambar dengan lebih baik.

Beberapa tahun lalu, banyak model CNN yang tealah diusulkan yang difokuskan pada peningkatan akurasi, kemudian meminimalkan jumlah parameter, dan pengadaptasian model CNN untuk sistem embedded dan mobile. CNN meniru cara sel-sel saraf manusia dalam berkomunikasi dengan neuron yang saling berhubungan yang memiliki arsitektur yang sama. Arsitektur pertama dari CNN yaitu LeNet yang muncul pada tahun 1998 tidak terlalu mendapat perhatian disebabkan keterbatasan yang ada seperti dataset ujicoba yang kecil, computer yang lambat dan salah dalam menerapkan fungsi linear. Pada tahun 2012, muncul AlexNet yang merupakan model dari CNN yang memenangkan kompetisi ImageNet. Alexnet mirip dengan LeNet-5 tetapi lebih signifikan dalam beberapa cara yang berdampak pada perkembangan Artificial Intelligence. Setelah AlexNet, arsitektur atau model lain dari CNN mulai bermunculan seperti VGG, GoogleNet, ResNet, Inception-V3, InceptionResNet-V2, DenseNet, SqueezeNet, dan lain.lain.

Dalam penggunaan CNN, mesin perlu dilatih dengan image dalam jumlah yang cukup besar, semakin banyak jumlah image yang dipergunakan maka mesin akan cenderung lebih pandai. Dalam hal peningkatan akurasi, model dari arsitektur yang digunakan akan sangat signifikan. Beberapa perbandingan menunjukkan arsitektur yang berbeda untuk melakukan sejumlah training dengan hasil akurasi yang semakin tinggi. Arsitektur CNN dengan layer yang dalam tentu akan mempengaruhi waktu operasi dan performa dari infrastruktur computer yang digunakan. Maka dari itu pengembangan arsitektur CNN yang dilakukan sampai saat ini lebih cenderung kepada penambahan layer yang semakin dalam, ketimbang untuk merancang layer yang efisien. CNN dengan arsitektur Inception, ResNet, dan

DenseNet digunakan dalam penelitian ini untuk mendapatkan perbandingan hasil akurasi dari masing-masing model.

II. RELATED WORK

Deep Learning dan CNN telah terbukti sangat efektif dalam beberapa masalah computer seperti deteksi objek dan klasifikasi gambar. Dalam hal akurasi dan biaya komputasi arsitektur CNN terbaik adalah model Inception yaitu pemenang kompetisi ImageNet Large Scale Visual Recognition Competition (ILSVRC) pada tahun 2014, dan arsitektur Residual Network (ResNet) yaitu pemenang tantangan ILSVRC pada tahun 2015. Sebuah penelitian menggabungkan ide inti dari kedua model ini ke dalam arsitektur hibrida untuk meningkatkan kinerja klasifikasi HIS. Pengujian dilakukan pada empat dataset dengan memperoleh akurasi 95,31% pada dataset Universitas Pavia, 99,02 pada dataset Pavia Center, 95,33% pada dataset Salinas dan 90, 57% pada dataset Indian Pines.

Pada penelitian lain, Faster R-CNN berbasis DenseNet digabungkan dengan Fitur Gabor digunakan untuk meneteksi kelainan pada esofagus. Penggabungan dilakukan untuk meningkatkan detail tekstur pada tahap deteksi. Arsitektur diuji coba pada dua dataset (Kvasir dan MICCAI 2015). Mengenai Kvasir, hasilnya menunjukkan kinerja yang luar biasa dengan recall 90,2% dan presisi 92,1% dengan mAP 75,9%. Sedangkan untuk dataset MICCAI 2015, model ini mampu mengungguli performa state-of-the-art dengan 95% recall dan 91% presisi dengan nilai mAP 84%. Hasil ini menunjukkan bahwa sistem mampu mendeteksi dengan kinerja yang baik tanpa campur tangan manusia.

Selain itu, terdapat penelitian yang menggunakan arsitektur Inception-CNN untuk mendeteksi kesalahan sensor Aero-Engine. Penelitian ini memberikan hasil dengan akurasi 95,41% yang meningkatkan akurasi prediksi sebesar 17,27% dan 12,69% dibandingkan dengan algoritma jaringan non-saraf dengan kinerja terbaik. Selain itu, metode ini menyederhanakan kesalahan yang terdiri dari beberapa algoritma fusi menjadi satu algoritma deteksi yang mengurangi kesulitan algoritma.

III. ARCHITECTURE

A. Inception

Inception adalah arsitektur *neural network* yang digunakan untuk klasifikasi objek pada kompetisi ImageNet Large-Scale Visual Recognition Challenge 2014 (ILSVRC14). Perkembangan dari arsitektur ini yang dikumpulkan untuk kompetisi adalah GoogleNet, jaringan dengan kedalaman 22 *layer*. GoogleNet memiliki tingkat akurasi lebih tinggi (top-5% *error* 6.67% dibandingkan dengan top-5% *error* 15.3%). GoogleNet mendapatkan predikat sebagai arsitektur

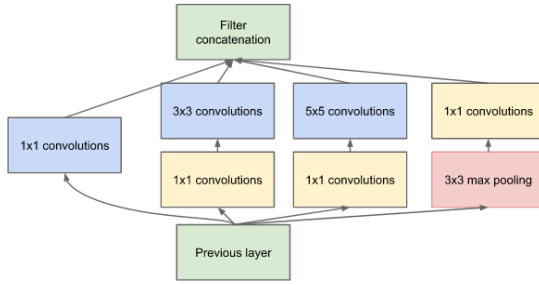


Fig. 1. Inception Block

kinerja paling baik. Kelebihan googlenet terletak pada inception modules. Inception modules terdiri dari sejumlah convolution kecil yang digunakan untuk mereduksi.

Berdasarkan Fig.1, Blok Inception menerapkan empat blok konvolusi secara terpisah pada peta fitur yang sama yakni konvolusi 1x1, 3x3, dan 5x5, dan operasi max pool. Ini akan memungkinkan jaringan untuk melihat data yang sama dengan bidang reseptif yang berbeda. Konvolusi 1x1 tambahan sebelum konvolusi 3x3 dan 5x5 digunakan untuk reduksi dimensi. Ini sangat penting karena fitur dari semua cabang digabungkan setelahnya, dan jika tidak ingin ada perbedaan ukuran fitur. Konvolusi 5x5, 25 kali lebih mahal daripada konvolusi 1x1, dapat menghemat banyak komputasi dan parameter dengan mengurangi dimensi sebelum konvolusi besar.

B. ResNet

Model ini berfokus pada computational accuracy yang merupakan model dengan lapisan paling banyak. Hal ini memungkinkan karena resnet menggunakan Residual Network sehingga tidak perlu khawatir masalah diminishing gradient yang membuat performa model menurun jika model terlalu banyak lapisan. Resnet sendiri memiliki akurasi yang tinggi dibanding dengan model lainnya namun kecepatan training modelnya sendiri juga cepat meskipun tidak secepat GoogleNet ataupun Inception dengan memberikan solusi berupa blok residual. Adapun modelnya

$$x_l + 1 = F(x_l)$$

dengan kita memodelkan menjadi

$$x_l + 1 = x_l + F(x_l)$$

yang dimana F merupakan pemetaan non linear.

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

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

ResNet terbukti berhasil menghasilkan visualisasi lebih halus yang dapat menggeneralisasikan (with skip connection) dibandingkan jaringan pembelajaran tingkat lanjut (without skip connection), seperti sumbu x dan y pada gambar dibawah ini :

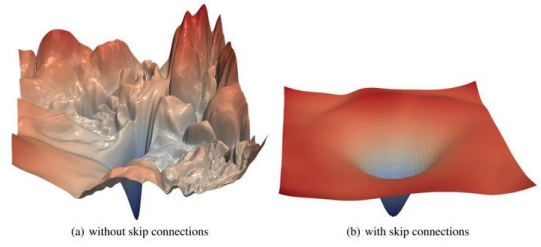


Fig. 2. Visualisasi ResNet

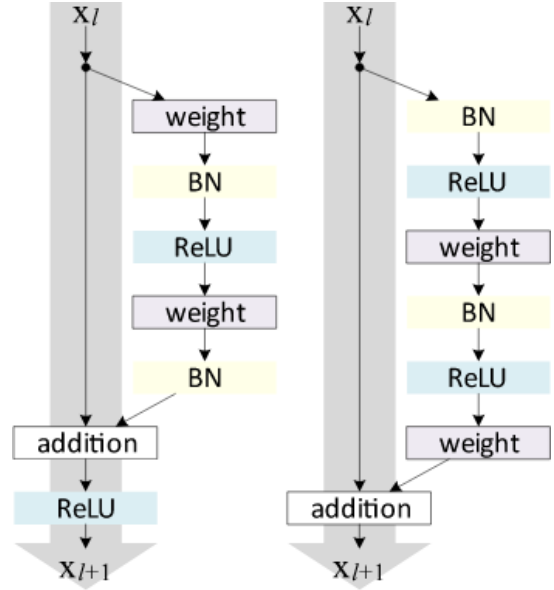


Fig. 3. ResNet Block

C. DenseNet

DenseNet terbagi menjadi tiga bagian yakni DenseLayer, DenseBlock dan TransitionLayer. Pada DenseNet sendiri berfokus pada 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. Setiap dense blocks, terdapat transition layers yang berguna untuk melakukan proses konvolusi dan pooling.

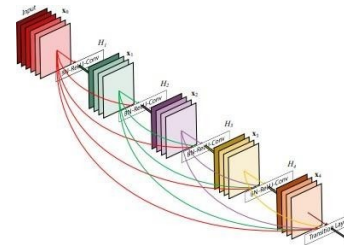


Fig. 4. Arsitektur DenseNet

Pada gambar 4 arsitektur DenSet terdapat lima blok pertumbuhan $k = 4$, setiap lapisan sebagai input semua fitur (Huang et al., 2017). Pada gambar 1, x_0 yang dilewatkan melalui Convolution Network. Jaringan terdiri dari lapisan L , masing-masing yang mengimplementasikan transformasi non-linier $HI(-)$, di mana l adalah indeks lapisan. $HI(-)$ dapat menjadi fungsi komposit operasi seperti Batch Normalization (BN) (Ioffe Szegedy, 2015), unit linier yang diperbaiki (ReLU) (Xavier Glorot, Antoine Bordes, 2011), Pooling (Y. LeCun, L. Bottou, Y. Bengio, 1998), atau Convolution (Konv). Output dari 'lapisan ke- th disebut sebagai x' .

Diterapkan konvolusi 1×1 untuk pengurangan dimensi dengan konvolusi 3×3 berikutnya. Hasil output digabungkan ke aslinya dan dikembalikan. Perhatikan bahwa kami menerapkan Normalisasi Batch sebagai lapisan pertama dari setiap blok. Ini memungkinkan aktivasi yang sedikit berbeda untuk fitur yang sama ke lapisan yang berbeda, pada tergantung apa yang dibutuhkan.

IV. RESEARCH METODOLOGY

A. Dataset

Dataset ini menggunakan dataset CIFAR-10. Dataset digunakan untuk mentrain dan mengevaluasi data dari CIFAR-10 yang 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 32×32 piksel. Dataset ini terdiri dari 50.000 data latih dan 10000 data uji yang terbagi menjadi 10 kelas yang terdapat pada dataset CIFAR-10. Sebelum dataset tersebut digunakan perlu dilakukan normalisasi pada dataset tersebut menggunakan nilai mean dan menormalisasikan dari dataset tersebut. Hal ini dilakukan untuk mengurangi risiko overfitting dan membantu proses pengerjaan CNN dalam menggeneralisasikan menghasilkan output yang terbaik.

B. Data Augmentation

Dilakukan dua augmentasi secara random dengan tujuan untuk menghindari terjadinya overfitting. Oleh karena itu dilakukan pada beberapa tahap. Pertama, proses data augmentasi dilakukan dengan melakukan flip pada gambar secara horizontal dengan pendekatan 50% 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 resolusi 32×32 agar selanjutnya dapat dimasukkan kedalam model CNN.

C. Model

Dalam membangun model CNN diperlukan tiga arsitektur CNN yakni GoogleNet, ResNet, dan DenseNet. Untuk arsitektur ResNet, akan digunakan dua 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 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 sehingga menghasilkan akurasi pada masing masing model yang dimana GoogleNet adalah model untuk mendapatkan kinerja terendah pada set validasi dan pengujian, meskipun sangat dekat dengan DenseNet. Pencarian hyperparameter yang tepat untuk semua ukuran untuk mwlalui GoogleNet kemungkinan akan meningkatkan akurasi model ke tingkat yang sama, tetapi ini juga rumit mengingat banyaknya hyperparameter. ResNet mengungguli DenseNet dan GoogleNet lebih dari 1% pada set validasi, sementara ada perbedaan kecil antara kedua versi, asli dan pra-aktivasi. Adapun tabel masing masing tingkat akurasi model.

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. 5. Tingkat Akurasi Setiap Model

D. Pytorch Lightning

Pytorch Lightning merupakan framework yang digunakan pada pengujian model arsitektur GoogleNet, ResNet dan DesNet. Pada PyTorch Lightning, didefinisikan `pl.LightningModule` yang mengorganisasikan kode yang digunakan menjadi 5 bagian utama yaitu, Initialization, Optimizers, Training loop, Validation loop, dan Test loop.

V. CONCLUSION

Berdasarkan penjelasan yang didapatkan menghasilkan kesimpulan berupa model pada Resnet yang merupakan arsitektur sederhana namun akurat sehingga menghasilkan peforma lebih baik dibandingkan GoogleNet dan DesNet. Jika ingin menerapkan pada konteks yang kompleks dan gambar ukuran lebih besar maka adanya ketidakseimbangan antara GoolheNet dan arsitektur skipconnection seperti ResNet dan DesNet. Jika dilihat pada implementasi penggunaan dari dataset CIFAR10 maka DesNet lebih unggul dibandingkan ResNet. Namun menunjukkan kombinasi ResNet dan DesNet yang saling memiliki masing masing keunggulan tersendiri.