Mid Term Technical Report
CNN Inception, ResNet, dan DenseNet

Faradias Izza A. F., Truly Roselyne I. R. dan Nurul Salsabila S.
Program Studi Sistem Informasi, Departemen Matematika
Universitas Hasanudddin

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Mid Term Technical Report
CNN Inception, ResNet, dan DenseNet

***Abstract*—*Convolutional Neural Network* adalah jenis *multi-layer neural network* yang dirancang untuk mengenali pola visual langsung dari gambar piksel dengan pra-pemrosesan minimal. Proyek ImageNet adalah database visual ekstensif yang dirancang untuk digunakan dalam penelitian perangkat lunak pengenalan objek visual. Proyek ImageNet menjalankan kontes perangkat lunak tahunan, *ImageNet Large Scale Visual Recognition Challenge* (ILSVRC), di mana program perangkat lunak bersaing untuk mengklasifikasikan dan mendeteksi objek dengan benar. Laporan ini membandingkan tiga arsitektur CNN yang berbeda untuk klasifikasi citra menggunakan dataset CIFAR10. Kinerja jaringan dibandingkan sesuai dengan akurasi validasi dan set uji masing-masing model. Menurut hasil kinerja yang diperoleh, ResNet mengungguli DenseNet dan GoogleNet lebih dari 1% pada set validasi. Pada saat yang sama, ada perbedaan kecil antara kedua versi ResNet, yaitu versi ResNet asli dan pra-aktivasi.**

I. Introduction

Selama beberapa tahun terakhir, jumlah aplikasi dalam klasifikasi citra telah meningkat secara signifikan. Klasifikasi citra digunakan di banyak bidang, seperti pencitraan medis, identifikasi objek dalam citra satelit, sistem kontrol lalu lintas, deteksi lampu rem, penglihatan mesin, dan masih banyak lagi. Tugas-tugas ini memerlukan kumpulan data skala besar yang diberi label dengan tepat, dan sebagian besar mencakup berbagai jenis gambar, dari anjing atau kucing hingga lanskap, jalan, dan sebagainya [1].

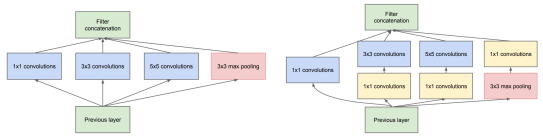
Salah satu permasalahan dalam *computer vision* adalah pengklasifikasian objek pada citra secara umum, bagaimana menduplikasi kemampuan manusia dalam memahami informasi citra sehingga komputer dapat mengenali objek seperti manusia. Tujuan dari klasifikasi citra adalah untuk mengidentifikasi dan menggambarkan tingkat keabuan (atau warna) yang unik, fitur-fitur yang terjadi dalam sebuah gambar dalam kaitannya dengan objek yang benar-benar diwakili oleh fitur-fitur ini di lapangan. [2].

Convolutional Neural Network merupakan metode *deep learning* yang dapat mendeteksi dan mengenali suatu objek dalam citra digital. Selama beberapa tahun terakhir, *Deep Learning* telah menunjukkan kinerja yang luar biasa. Hal ini didukung oleh komputasi yang lebih kuat, kumpulan data yang besar, dan teknik untuk melatih jaringan yang lebih dalam. Baru-baru ini, banyak arsitektur menarik dari *deep CNN* yang telah dilaporkan muncul. Hal ini disebabkan ketersediaan data dalam jumlah besar dan peningkatan teknologi perangkat keras. Laporan ini mereproduksi dan membandingkan tiga arsitektur klasifikasi gambar berbeda yang dilatih pada dataset CIFAR10, yang terdiri dari kumpulan gambar yang biasa digunakan untuk melatih *machine learning* dan algoritma *computer vision*. Arsitektur CNN yang dibandingkan adalah modul Inception dari GoogleNet, ResNet, dan DenseNet yang masing-masing menjadi juara atau runner-up dalam kompetisi ImageNet yang menjadi barometer kemajuan dalam pembelajaran terawasi untuk *computer vision* sejak tahun 2010.

II. Networks and Dataset

A. Inception

Pada tahun 2014, GoogLeNet memenangkan *ImageNet Challenge* dengan mengusulkan struktur yang menggabungkan kekuatan NiN dan paradigma blok berulang. Salah satu fokus laporan ini adalah untuk menjawab pertanyaan tentang ukuran kernel konvolusi mana yang terbaik. Jaringan populer sebelumnya menggunakan pilihan sekecil 1×1 hingga 11×11. Satu wawasan dalam laporan ini adalah bahwa kadang-kadang dapat menguntungkan untuk menggunakan kombinasi kernel dengan berbagai ukuran [3].



(a) Inception module, naïve version (b) Inception module with dimension reductions

Fig. 1. Struktur Inception blok

Ide utama dari arsitektur Inception didasarkan pada mencari tahu bagaimana struktur *sparse* lokal yang optimal dalam jaringan visi konvolusional dapat didekati dan ditutupi oleh komponen padat yang tersedia. Untuk menghindari masalah *patchalignment*, inkarnasi arsitektur Inception saat ini dibatasi untuk menyaring ukuran 1×1, 3×3 dan 5×5, namun keputusan ini lebih didasarkan pada kenyamanan daripada kebutuhan. Ini juga berarti bahwa arsitektur yang disarankan adalah kombinasi dari semua lapisan tersebut dengan bank filter keluarannya yang digabungkan menjadi satu vektor keluaran yang membentuk masukan dari tahap berikutnya. Selain itu, karena operasi penyatuan sangat penting untuk keberhasilan dalam jaringan konvolusi mutakhir saat ini. Hal tersebut menunjukkan bahwa menambahkan jalur penyatuan paralel alternatif di setiap tahap tersebut harus memiliki efek menguntungkan tambahan [4].

Salah satu aspek menguntungkan utama dari arsitektur ini adalah memungkinkan untuk meningkatkan jumlah unit pada setiap tahap secara signifikan tanpa ledakan yang tidak terkendali dalam kompleksitas komputasi. Penggunaan pengurangan dimensi di mana-mana memungkinkan untuk melindungi

sejumlah besar filter input dari tahap terakhir ke lapisan berikutnya, pertama-tama mengurangi dimensinya sebelum menggulungnya dengan ukuran tambalan yang besar. Aspek praktis lain yang berguna dari desain ini adalah selaras dengan intuisi bahwa informasi visual harus diproses pada berbagai skala dan kemudian digabungkan sehingga tahap berikutnya dapat mengabstraksikan fitur dari skala yang berbeda secara bersamaan.

B. ResNet

ResNet, yang diusulkan pada tahun 2015 oleh para peneliti di Microsoft Research, memperkenalkan arsitektur baru yang disebut *Residual Network*. Untuk mengatasi masalah gradien menghilang/meledak, arsitektur ini memperkenalkan konsep yang disebut Jaringan Residual. ResNet menggunakan teknik yang disebut *skip connections*. *Skip connections* melewatkan pelatihan dari beberapa lapisan dan terhubung langsung ke output. Pendekatan di balik jaringan ini bukannya mempelajari pemetaan yang mendasarinya; memungkinkan jaringan agar sesuai dengan pemetaan residual [5].

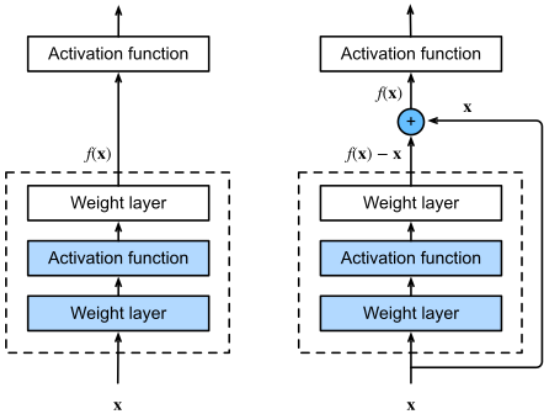


Fig. 2. *Plain Building Blocks* (kiri) dan *Residual Building Blocks* (kanan)

Residual Networks lebih tahan terhadap degradasi sehubungan dengan jaringan biasa. Dalam sebuah jaringan, motivasi utama selain menghasilkan koneksi sisa adalah menyediakan koneksi alternatif ke koneksi biasa. Sambungan sisa memiliki koefisien konstan yang sama dengan 1, tidak termasuk sambungan sisa dari prosedur kereta api. Dalam prosedur kereta, jika koefisien koneksi reguler cenderung konvergen ke 0, jalan pintas residual menjamin integritas jaringan. Ketika koneksi reguler di-shortcut oleh koneksi residual, data kumulatif dihitung sebelum koneksi reguler diteruskan ke seluruh jaringan. Koneksi alternatif memberikan kesempatan kepada jaringan untuk dapat menggunakan pintasan ini alih-alih koneksi biasa bila diperlukan [6].

ResNet mengikuti desain lapisan konvolusi penuh 3×3 dari VGG. Blok residual memiliki dua lapisan konvolusi 3×3 dengan jumlah saluran keluaran yang sama. Setiap lapisan konvolusi diikuti oleh lapisan normalisasi batch dan fungsi aktivasi ReLU. Kemudian, peneliti melewatkan dua operasi konvolusi ini dan menambahkan input langsung sebelum fungsi aktivasi ReLU terakhir. Jenis desain ini mensyaratkan bahwa output dari dua lapisan konvolusi harus memiliki bentuk yang sama dengan input, sehingga dapat dijumlahkan. Jika kita ingin mengubah jumlah saluran, kita perlu memperkenalkan lapisan konvolusi 1×1 tambahan untuk mengubah input menjadi bentuk yang diinginkan untuk operasi penjumlahan [3].

C. DenseNet

DenseNet adalah arsitektur modern CNN untuk pengenalan objek visual yang telah memperoleh *state-of-the-art* dengan parameter yang lebih sedikit. Dengan beberapa modifikasi utama, DenseNet sangat mirip dengan ResNet. DenseNet, bersama dengan atribut gabungannya (.), menggabungkan output lapisan sebelumnya dengan lapisan masa depan, sementara ResNet menggunakan atribut aditif (+) untuk menggabungkan lapisan sebelumnya dengan lapisan masa depan. Arsitektur DenseNet bertujuan untuk memperbaiki masalah ini dengan menghubungkan semua lapisan secara padat [7].

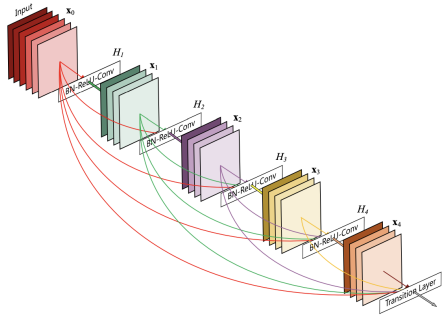


Fig. 3. Arsitektur DenseNet

Perbedaan utama antara ResNet dan DenseNet adalah bahwa dalam kasus terakhir, output digabungkan (dilambangkan dengan [,]) daripada ditambahkan. Nama DenseNet muncul dari fakta bahwa grafik ketergantungan antar variabel menjadi cukup padat. Lapisan terakhir dari rantai tersebut terhubung erat ke semua lapisan sebelumnya [3]. Sambungan padat atau *dense connections* ditunjukkan pada Fig. 4.

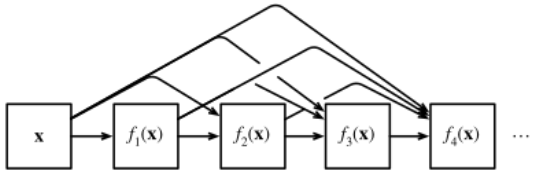


Fig. 4. *Dense connections* pada DenseNet.

Kemungkinan efek kontra-intuitif dari konektivitas padat ini polanya adalah membutuhkan lebih sedikit parameter daripada jaringan konvolusi tradisional, karena tidak perlu mempelajari kembali peta fitur yang berlebihan. Selain efisiensi parameter yang lebih baik, satu keuntungan besar DenseNet adalah peningkatan aliran informasi dan gradien di seluruh jaringan, yang

membuatnya mudah untuk dilatih. Setiap lapisan memiliki akses langsung ke gradien dari fungsi kerugian dan sinyal input asli, yang mengarah ke pengawasan mendalam yang implisit. Ini membantu pelatihan arsitektur jaringan yang lebih dalam [8].

D. Dataset

Basis data gambar yang digunakan dalam laporan ini adalah dataset CIFAR10 (*Canadian Institute for Advanced Research, 10 class*) [9]. Dataset CIFAR-10 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. *Training batch* pada dataset ini terdiri dari 5000 gambar untuk tiap kelas [10].

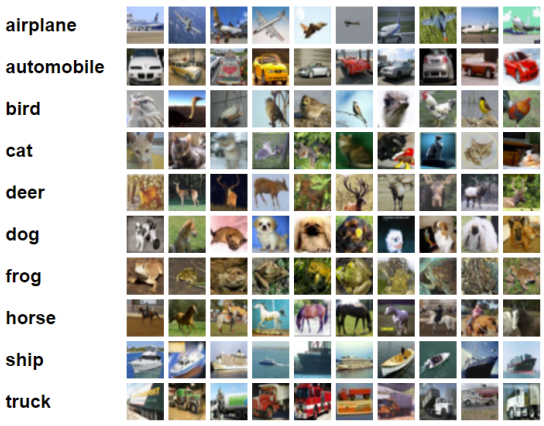


Fig. 5. Kelas sampel dalam dataset CIFAR10, serta 10 gambar acak dari masing-masing kelas

III. Experiments

A. Pre-processing

Sebelum melakukan *training* pada dataset, dilakukan *preprocessing* pada data dengan mean nol. Sebagai langkah pertama, kami menghitung mean dan standar deviasi dari dataset CIFAR10. Informasi ini akan digunakan untuk normalisasi data. Selain itu, kami melakukan augmentasi data selama proses *training*. Ini mengurangi risiko *overfitting* dan membantu CNN untuk menggeneralisasi lebih baik. Jenis augmentasi data yang akan digunakan adalah *Random Horizontal Flip* dan *Random Resized Crop*. Random Horizontal Flip akan membalik setiap gambar secara horizontal dengan peluang 50%, dan Random Resize Crop akan menskalakan gambar dalam rentang kecil sementara mengubah rasio aspek kemudian meng-*crop* gambar dalam ukuran sebelumnya. Oleh karena itu, nilai piksel aktual berubah sementara konten atau semantik keseluruhan gambar tetap sama. Terakhir, kami secara acak membagi gambar menjadi set pelatihan dan set validasi.

B. Networks

Dalam membangun model CNN, kami menggunakan 3 arsitektur CNN, yaitu *GoogLeNet*, *ResNet*, dan *DenseNet*. Kami menggunakan library PyTorch Lightning untuk menyederhanakan kode yang dibutuhkan untuk melatih, mengevaluasi, dan menguji model. Pytorch juga menangani *logging* ke TensorBoard dan menggunakan konsep callback.

Pada model *GoogLeNet*, blok Inception menerapkan empat blok konvolusi secara terpisah pada satu blok yang sama, yaitu konvolusi 1x1, 3x3, 5x5, dan operasi max pooling. Arsitektur *GoogLeNet* terdiri dari beberapa blok Inception dengan max pooling sesekali untuk mengurangi tinggi dan lebar feature map. Kami melakukan pelatihan model pada hampir 200 epochs. Pada penelitian ini, model *GoogLeNet* menggunakan *activation function* RelU dan *optimizer* Adam. Nilai *learning rate* pada proses *training* adalah 0.001 dengan penurunan 0.1 setelah 100 dan 150 epochs.

Pada model *ResNet*, digunakan dua jenis blok sebagai perbandingan, yaitu *original ResNet block* dan *Pre-Activation ResNet block*. Arsitektur *ResNet* secara keseluruhan terdiri dari penumpukan beberapa blok ResNet, di mana beberapa di antaranya melakukan *downsampling* input. Pada penelitian ini, model *ResNet* mempunyai [3,3,3] blok, yang artinya model ini memiliki 3 kali grup dari 3 blok ResNet, di mana *subsampling* berlangsung di blok keempat dan ketujuh. Tiga grup ini beroperasi pada resolusi 32×32, 16×16, dan 8×8. Pada penelitian ini, model *ResNet* menggunakan *activation function* RelU dan *optimizer* Stochastic Gradient Descent dengan momentum. Nilai *learning rate* pada proses *training* adalah 0.1 dengan penurunan 0.1 setelah 100 dan 150 epochs dan nilai momentum 0.9.

Pada model *DenseNet*, kami membagi implementasi tiap *layer* di *DenseNet* menjadi tiga bagian: *DenseLayer*, *DenseBlock*, dan *TransitionLayer*. Modul *DenseLayer* mengimplementasikan satu *layer* di dalam *dense block*. Modul ini menerapkan konvolusi 1x1 untuk pengurangan dimensi dengan konvolusi 3x3 berikutnya. Modul *DenseBlock* merangkum beberapa *dense layers* yang diterapkan secara berurutan. Setiap *dense layer* mengambil input asli sebagai input yang digabungkan dengan *feature map* pada lapisan sebelumnya. Terakhir, *TransitionLayer* mengambil output final dari *dense block* sebagai input dan mengurangi dimensi *channel*-nya menggunakan konvolusi 1x1. Untuk mengurangi dimensi tinggi dan lebar, kami mengambil pendekatan yang sedikit berbeda dari pada *ResNet* dan menerapkan *average pooling* dengan ukuran kernel 2 dan *stride* 2. Model *DenseNet* dilatih menggunakan *activation function* RelU dan *optimizer* Adam. Nilai *learning rate* pada proses pelatihan adalah 0.001 dengan penurunan 0.1 setelah 100 dan 150 epochs.

IV. Results and Conclusion

Berdasarkan hasil perbandingan yang diperoleh, ResNet mengungguli GoogleNet dan DenseNet, memberikan performa yang lebih unggul untuk dataset CIFAR dan implementasinya

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 I. PERBANDINGAN TIAP-TIAP MODEL

yang sederhana. ResNet adalah yang tercepat karena DenseNet dan GoogleNet memiliki lebih banyak layer yang diterapkan secara berurutan dalam implementasi primitif. Namun, jika ingin menerapkan model pada task yang lebih kompleks dengan gambar yang lebih besar dan lebih banyak layer di dalam jaringan, kita bisa melihat perbedaan yang cukup besar antara *GoogLeNet* dan arsitektur *skip-connection* seperti *ResNet* dan *DenseNet*

REFERENCES

[1] L. Oscar, R. Ian, and R. Aditya, "Image Classification with Classic and Deep Learning Techniques" *CoRR*, abs/2105.04895, May 14, 2021. [Full Text]. Available: Arxiv, https://arxiv.org/abs/2105.04895. Accessed on: Apr. 26, 2022.

[2] S. Kavish, "Image Classification Techniques," *Medium*, May. 8, 2020. [Online]. Available: https://medium.com/analytics-vidhya/image-classification-techniques-83fd87011cac. [Accessed: Apr. 26, 2022].

[3] Z. Aston, L. Zachary, M. Li and S. Alexander. *Dive into Deep Learning*, 2020. [E-book] https://d2l.ai. Accessed on: Apr. 27, 2022.

[4] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., … Rabinovich, A., "Going deeper with convolutions." *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp(1-9), 2015, https://ieeexplore.ieee.org/document/7298594. Accessed on: Apr. 26, 2022.

[5] GeeksforGeeks, "Residual Networks (ResNet) – Deep Learning," *GeeksforGeeks*, Jan. 27, 2022. [Online]. Available: https://www.geeksforgeeks.org/residual-networks-resnet-deep-learning/. [Accessed: Apr. 26, 2022].

[6] He, K., Zhang, X., Ren, S., and Sun, J., "Deep residual learning for image recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp(770-778), 2016, https://ieeexplore.ieee.org/document/7780459. Accessed on: Apr. 26, 2022.

[7] Hasan N, Bao Y, Shawon A, and Huang Y., "DenseNet Convolutional Neural Networks Application for Predicting COVID-19 Using CT Image." *SN Computer Science*, vol 2 no 5, 2021, https://pubmed.ncbi.nlm.nih.gov/34337432/. Accessed on: Apr. 26, 2022.

[8] Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. "Densely connected convolutional networks." *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp(4700-4708), 2017, https://ieeexplore.ieee.org/document/8099726. Accessed on: Apr. 26, 2022.

[9] Alex. K, Vinod. N, and Geoffrey. H "CIFAR-10 (Canadian Institute for Advanced Research), Available: http://www.cs.toronto.edu/~kriz/cifar.html. [Accessed: Apr. 26, 2022].

[10] Alex. K"Learning Multiple Layers of Features from Tiny Images", Apr 8, 2009,https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf. Accessed on: Apr. 26, 2022.