

ANALISIS PERBANDINGAN MODEL CNN VGG16 DAN DENSENET121 MENGGUNAKAN KERANGKA KERJA TENSORFLOW UNTUK DETEKSI JENIS HEWAN

Comparison Analysis of CNN Model VGG16 and DenseNet121 using TensorFlow Framework to Detect Animal Images

Wahyu Wijaya Kusuma^{*,1)}, R. Rizal Isnanto¹⁾, Adnan Fauzi¹⁾

¹⁾Program Studi Teknik Komputer, Fakultas Teknik, Universitas Diponegoro
Jl. Prof. Soedarto, SH, Kampus Undip Tembalang, Semarang, Indonesia 50275

How to cite: W.W. Kusuma, R.R. Isnanto, and A. Fauzi, "Analisis Perbandingan Model CNN VGG16 dan DenseNet121 Menggunakan Kerangka Kerja TensorFlow untuk Deteksi Jenis Hewan," *Jurnal Teknik Komputer*, vol. 1, no. 4, pp. 141-147, 2023. doi: 10.14710/jtk.v1i4.37009 [Online].

Abstract – This research will be comparing the model performance of VGG16 and DenseNet121 in classifying animal images. The dataset used is dataset of 10 different kinds of animals. The amount of dataset taken is 14500. 10000 for training, 2500 for validation, and 2000 for evaluating. These two models each are trained for 30 epochs long. Based on the research results, VGG16 has the accuracy of 0,97, Precision 0,90, Recall 0,90 and F1 Score 0,90. For the DenseNet121 model, this model has the Accuracy of 0,98, Precision, 0,95, Recall 0,95, and F1 Score 0,95. Based on this research, it can be concluded that DenseNet121 model can classify animal images better than VGG16 model.

Keywords – Image Classification, Convolutional Neural Network, Animals, VGG16, DenseNet121

Abstrak – Penelitian ini membandingkan kinerja dari model CNN VGG16 dan DenseNet121 dalam mendeteksi jenis hewan. Dataset yang digunakan adalah dataset citra 10 jenis hewan. Jumlah dataset yang diambil adalah 14500. 10000 untuk pelatihan, 2500 untuk validasi, dan 2000 untuk pengujian. Kedua model dilatih masing-masing selama 30 epoch. Berdasarkan hasil penelitian, VGG16 memiliki tingkat Akurasi 0,97, Presisi 0,90, Recall 0,90 dan F1 Score 0,90. Sementara model DenseNet121 memiliki tingkat Akurasi 0,98, Presisi 0,95, Recall 0,95, dan F1 Score 0,95. Dari hasil penelitian, dapat disimpulkan bahwa model DenseNet121 mampu mengklasifikasikan gambar jenis hewan lebih baik dari model VGG16.

Kata kunci – Klasifikasi Gambar, Convolutional Neural Network, Hewan, VGG16, DenseNet121

I. PENDAHULUAN

Pada era sekarang dunia sudah berada di era digital. Era dimana setiap aspek kehidupan manusia berkaitan dengan komputasi. Seiring berkembangnya zaman, manusia terus menembangkan teknologi untuk meringankan pekerjaannya. Salah satunya adalah teknologi kecerdasan buatan atau yang biasa disebut *Artificial Intelligence (AI)*.

Salah satu cabang yang ada pada kecerdasan buatan adalah *Computer Vision* atau penglihatan komputer. *Computer Vision* adalah ilmu yang mempelajari bagaimana sebuah komputer bisa mengenali sebuah objek. Salah satu permasalahan dari cabang ilmu ini adalah deteksi objek dan klasifikasi gambar. Deteksi objek merupakan salah satu topik dalam kecerdasan buatan yang terus dipelajari dan dikembangkan dengan sangat pesat saat ini. Deteksi objek merupakan teknologi dari komputer yang berhubungan dengan komputer agar bisa mengenali sebuah objek dalam suatu citra digital yang dapat berupa warna atau objek.

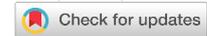
Ada beberapa metode agar komputer dapat mengenali suatu objek pada sebuah citra digital. Salah satunya adalah *Convolutional Neural Network (CNN)*, yang merupakan sebuah metode *Deep Learning* yang paling sering digunakan untuk memecahkan masalah deteksi objek dan klasifikasi gambar.

CNN sejatinya bukan teknologi yang sangat baru. Metode *Deep Learning* ini pertama kali muncul pada tahun 1998, dengan nama arsitekturnya yaitu LeNet, namun metode ini kurang mendapat perhatian karena keterbatasan teknologi pada waktu itu dengan data yang sedikit, komputer yang kurang memadai dan salah dalam menerapkan fungsi non linear[1].

Penelitian tentang arsitektur *Deep Learning* mulai meledak pada tahun 2012 diawali dengan diadakannya kompetisi *ImageNet Large Scale Visual Recognition Challenge (ILSVRC)* yang berhasil dimenangkan oleh arsitektur AlexNet. Arsitektur ini yang pertama kali menggunakan fungsi ReLU untuk melatih neuron pada

^{*})Penulis Korespondensi (Wahyu Wijaya Kusuma)

Email : wahyuwk000@gmail.com



jaringan sarafnya[2]. Berkat kompetisi ini lah dan munculnya ReLU, mulai banyak bermunculan model CNN yang memiliki performa yang sangat bagus. Diawali dengan munculnya AlexNet yang kala itu yang pertama kali menggunakan ReLU dan melaporkan bahwa ReLU 6x lebih cepat dalam waktu pelatihnnya dibanding fungsi aktivasi sebelumnya, tanh. Setelah AlexNet mulai banyak bermunculan model model lainnya, contohnya saja VGG16 dan DenseNet.

VGG16 pertama kali muncul pada tahun 2014, model ini terdiri dari 5 layer konvolusi dan pooling lalu diakhiri dengan fully connected layer. Model ini memiliki jumlah parameter yang banyak yaitu 138 juta parameter. Sementara itu DenseNet muncul pada tahun 2016, model ini berbeda dengan VGG16, karena memiliki parameter yang jauh lebih sedikit yaitu 8 juta parameter. Model ini memanfaatkan setiap layer awalnya yang terus terhubung pada layer berikutnya hingga layer akhir, menjadikan semua ciri yang sudah dipelajari diawal, terus dipakai hingga akhir.

Harapanya, penelitian ini mampu menganalisa performa model *Convolutional Neural Network (CNN)* untuk mendeteksi sebuah citra hewan dan mengklasifikasinya sesuai dengan jenis hewan tersebut.

II. TINJAUAN LITERATUR

A. Kajian Penelitian Terdahulu

Penelitian terdahulu dilakukan oleh Tanuwijaua dkk dengan judul “Modifikasi Arsitektur VGG16 untuk Klasifikasi Citra Digital Rempah-Rempah Indonesia”. Penelitian tersebut membandingkan hasil deteksi dan klasifikasi citra digital rempah-rempah menggunakan model CNN VGG16 dan AlexNet. Berdasarkan penelitian tersebut dapat disimpulkan bahwa model VGG16 memiliki hasil evaluasi yang lebih baik dari model AlexNet dalam hal mendeteksi dan klasifikasi citra digital rempah-rempah[3].

Penelitian terdahulu yang lain dilakukan oleh Rachmawati dkk dengan judul “Analisis Performa Pretrained Model Convolutional Neural Network dalam Mendeteksi Penyakit Tuberculosis”. Penelitian tersebut membandingkan hasil deteksi dan klasifikasi citra rontgen dada menggunakan model CNN Xception, InceptionResNetV2 dan DenseNet121. Berdasarkan hasil penelitian tersebut model CNN DenseNet121 memiliki tingkat keakurasian paling tinggi dibanding model CNN lainnya dalam mendeteksi penyakit tuberculosis pada rontgen dada[4].

Penelitian terdahulu yang lain dilakukan oleh Thunder dkk dengan judul “Implementasi Algoritma Convolutional Neural Networks di Microsoft Azure untuk Mendeteksi Jenis Kebutaan Mata yang dialami Penderita Penyakit Diabetes”. Penelitian tersebut membandingkan hasil deteksi dan klasifikasi citra kebutaan mata menggunakan model CNN InceptionV3 dan VGG16. Berdasarkan hasil penelitian tersebut model CNN VGG16 memiliki perubahan proses yang maksimal

mampu menghasilkan tingkat akurasi yang lebih rasional jika dibandingkan dengan Inception V3[5].

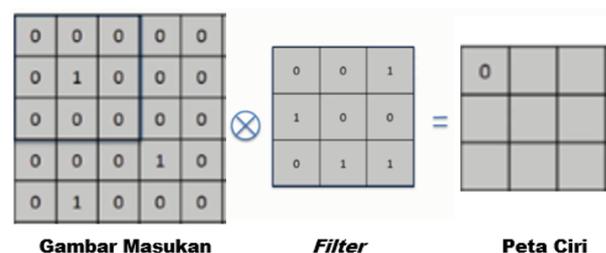
Perbedaan ide dalam penelitian terdahulu dengan penelitian saat ini terletak pada model CNN dan dataset yang digunakan. Dalam penelitian terdahulu belum ada yang membandingkan secara langsung model CNN VGG16 dan DenseNet121 dengan menggunakan citra hewan. Sedangkan dalam penelitian kali ini digunakan citra hewan yang berjumlah 12500 data pelatihan dan validasi dan 2000 data untuk pengujian serta terdiri dari 10 jenis hewan yaitu anjing, ayam, domba, gajah, kucing, kupu-kupu, laba-laba, sapi, tupai.

B. Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) adalah salah satu jenis jaringan saraf tiruan yang biasa digunakan dalam data gambar. Mereka sangat berguna untuk menganalisis dan memahami pola yang kompleks. CNN dapat digunakan untuk mengidentifikasi objek dalam sebuah gambar. CNN adalah teknik yang digunakan oleh manusia untuk menciptakan persepsi visual. Ini mirip dengan bagaimana mamalia menghasilkan gambar dan persepsi. *Convolutional Neural Network* mirip dengan jaringan saraf tiruan biasanya, tetapi mereka termasuk neuron dengan bobot, bias, dan fungsi aktivasi[6]. CNN menggunakan operasi konvolusi sebagai dasar dari algoritma nya. Nama konvolusi sendiri merupakan operasi aljabar linear yang mengalikan matriks dari tapis pada citra yang akan di proses, sehingga proses ini disebut lapisan konvolusi dan merupakan salah satu jenis layer yang dimiliki oleh CNN. Input dari CNN adalah kumpulan piksel yang berupa larik[3]. Arsitektur CNN pada umumnya bisa dilihat pada Gambar 2.1, masukan CNN yang berupa kumpulan piksel yang telah dirubah ke larik akan diproses pada lapisan konvolusi. Keluaran dari lapisan konvolusi ini adalah peta ciri. Peta ciri tersebut kemudian akan disederhanakan pada lapisan *pooling* lalu diproses lagi pada lapisan konvolusi berikutnya hingga ke lapisan *fully connected* yang akan menghasilkan keluaran berupa klasifikasi citra[7].

C. Lapisan Konvolusi

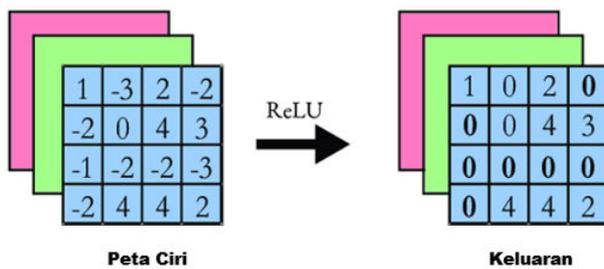
Pada Lapisan konvolusi adalah tempat dimana proses terjadinya operasi konvolusi, yaitu proses perkalian antara nilai masukan dan nilai tapis (*filter*). Keluaran dari proses ini adalah yang disebut sebagai peta ciri yang nantinya akan disederhanakan di lapisan penyaringan[7]. Contoh dari operasi dari lapisan konvolusi bisa dilihat pada Gambar 1.



Gambar 1. Operasi Konvolusi

D. Aktifasi ReLU

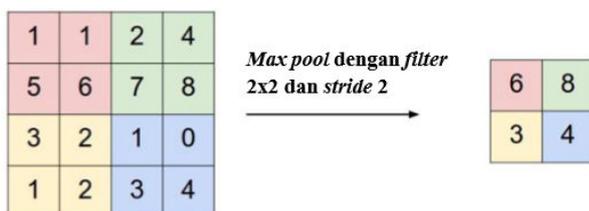
Fungsi aktifasi adalah proses sesudah lapisan konvolusi dan sebelum lapisan *pooling*. Proses ini memungkinkan waktu pelatihan menjadi lebih cepat dan lebih efektif dalam memetakan nilai keluaran peta ciri berdasarkan fungsi aktifasi yang digunakan[8]. Fungsi aktifasi yang paling sering digunakan saat ini adalah fungsi aktifasi ReLU, fungsi aktifasi ini akan memetakan nilai negatif menjadi nol dan memertahankan nilai positif. Contoh operasi pada aktifasi ReLU dapat dilihat pada Gambar 2.



Gambar 2. Operasi Aktifasi ReLU

E. Lapisan Pooling

Lapisan *pooling* adalah lapisan yang berfungsi untuk menyederhanakan keluaran lapisan konvolusi. Prinsipnya, lapisan *pooling* terdiri dari sebuah tapis dengan ukuran dan *stride* tertentu. Tujuan utama dari lapisan ini adalah untuk mempercepat waktu pelatihan karena jumlah parameter yang harus diperbaharui semakin sedikit serta mengatasi *overfitting*[11]. *Pooling* yang biasa digunakan adalah *Max Pooling* dan *Average Pooling*. *Max Pooling* akan mengambil nilai terbesar dari satu area peta ciri sementara *Average Pooling* mengambil nilai rata-rata. Contoh operasi pada lapisan *pooling* menggunakan *Max Pooling* dapat dilihat pada Gambar 3, keluaran dari lapisan konvolusi yang bernilai 4 x 4 disederhanakan menjadi 2 x 2.

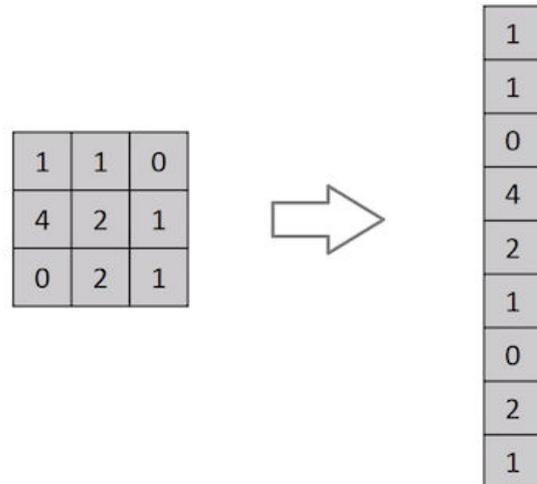


Gambar 3. Operasi Pooling

F. Lapisan Fully Connected

Lapisan *fully connected* adalah lapisan keluaran dari model CNN. Keluaran dari lapisan *pooling* yang masih berupa larik akan diubah menjadi vektor dengan fungsi *flatten* agar bisa digunakan sebagai input pada aktifasi *Softmax* yang nantinya akan menghasilkan keluaran

berupa probabilitas klasifikasi gambar. Contoh operasi pada fungsi *flatten* dapat dilihat pada Gambar 4.



Gambar 4. Fungsi *flatten*

G. Fungsi Softmax

Softmax merupakan fungsi yang berfungsi untuk menghitung probabilitas dari citra dan mengklasifikasinya. Fungsi ini bekerja dengan merubah vektor bilangan nyata menjadi bilangan yang jika dijumlahkan berjumlah 1. Masukan dari vektor bisa berupa bilangan positif lebih dari satu ataupun nol, *softmax* nantinya akan merubah masukan tersebut menjadi probabilitas diantara bilangan 0 sampai 1[15]. Jika masukannya besar maka probabilitasnya juga besar, begitu juga sebaliknya tetapi keluarannya akan tetap diantara 0 sampai 1. Untuk inilah fungsi *softmax* berada pada lapisan terakhir dari CNN.

H. Confusion Matrix

Untuk dapat mengetahui apakah sebuah model CNN sudah bekerja dengan baik atau belum, perlu untuk mengevaluasi model tersebut. Evaluasi model dilakukan menggunakan *Confusion Matrix* yang dapat dilihat pada Gambar 5.

		NILAI AKTUAL	
		POSITIF	NEGATIF
NILAI PREDIKSI	POSITIF	TP	FP
	NEGATIF	FN	TN

Gambar 5. Tabel *Confusion Matrix*

Dari tabel *Confusion Matrix* seperti pada Gambar 1, akan dihasilkan nilai *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN) yang



nantinya dapat dihitung nilai dari Akurasi, Presisi, *Recall*, dan *F1 Score* arsitektur CNN tersebut.

$$\text{Akurasi} = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

$$\text{Presisi} = \frac{TP}{TP+FP} \quad (2)$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad (3)$$

$$\text{F1 Score} = \frac{2 \cdot (\text{Presisi} \cdot \text{Recall})}{\text{Presisi} + \text{Recall}} \quad (4)$$

I. Flask

Flask adalah sebuah kerangka kerja aplikasi web yang ditulis menggunakan bahasa pemrograman Python. Flask berfungsi sebagai kerangka kerja aplikasi dan tampilan dari sebuah web. Sebagian besar komponen umum yang ada pada aplikasi web seperti Basis data dan Form validasi tidak disediakan di Flask. Pengguna dapat menggunakan ekstensi untuk menggunakan komponen tersebut dalam Flask. Flask dapat diinstall menggunakan *packet manager* dari Python. Keunggulan dari Flask itu sendiri diantaranya ringan untuk dijalankan, dapat menangani fungsi permintaan HTTP serta banyak dokumentasi dan komunitas yang mengembangkannya [12]

III. METODE PENELITIAN

Penelitian yang dilakukan dalam analisis model *Convolutional Neural Network* (CNN) adalah dengan menggunakan metode ekstraksi ciri, metode ini menggunakan model *Convolutional Neural Network* (CNN) yang sudah dilatih sebelumnya di ImageNet, model VGG16 dan DenseNet121 akan mengekstraksi fitur yang berbeda dari sampel yang baru.

A. Alur Penelitian

Penelitian dan proses pelatihan model CNN dilakukan menggunakan Google Colab dan menggunakan bahasa Python. Setelah berhasil dilatih maka akan dibandingkan hasil evaluasi dari kedua model CNN VGG16 dan DenseNet121.

B. Dataset

Pengambilan data untuk mendeteksi gambar diambil dari situs Kaggle (<https://www.kaggle.com/datasets/alessiocorrado99/animals10>), citra yang digunakan adalah gambar hewan yang terdiri atas 10 kelas yaitu anjing, kuda, laba-laba, kupu-kupu, domba, sapi, kucing, dan tupai. Sampel yang diambil berjumlah 14500 yang terdiri dari 10000 gambar untuk dilatih, 2500 gambar untuk validasi dan 2000 gambar untuk tes evaluasi.

Data pelatihan berfungsi sebagai masukan yang mengawali proses dari pelatihan model, sebagai acuan untuk memberi nilai awal bobot dari jaringan syaraf

tiruan model CNN tersebut. Data validasi adalah data yang dimasukan setelah semua data pelatihan selesai dilatih dalam 1 *epoch* saat pelatihan, data ini berfungsi untuk mengatur *hyperparameter* dari model CNN serta membandingkan perubahan dari bobot pada jaringan syaraf tiruan apakah mempengaruhi akurasi dari model tersebut.

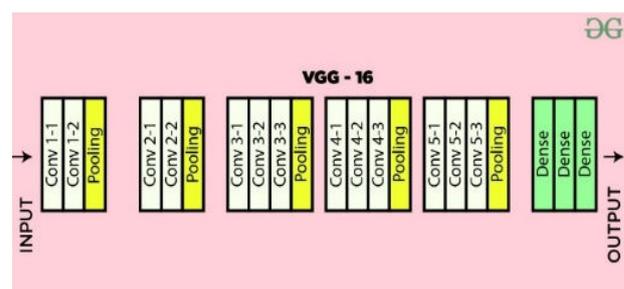
Terakhir, data test adalah data yang berfungsi untuk mengevaluasi performa dari model tersebut. Berbeda dari data pelatihan dan validasi, data ini tidak mempengaruhi lagi bobot dari jaringan syaraf tiruan pada model. Evaluasi biasanya dilakukan menggunakan *confusion matrix* untuk mencari tahu nilai dari Presisi, Akurasi, *Recall*, dan *F1-Score*.

C. Prapengolahan

Prapengolahan adalah proses yang digunakan untuk memudahkan dalam ekstraksi ciri dengan memanipulasi citra tersebut. Fungsi utama proses ini adalah mendapatkan data yang mendalam dari citra tersebut. Prapengolahan dilakukan menggunakan fungsi *ImageDataGenerator*. Pada algoritma yang digunakan dalam penelitian ini, Prapengolahan yang dilakukan adalah *rescale*, rotasi, *shift*, *zoom*, dan *flipRescale* berfungsi agar data dari gambar yang kita masukan yang memiliki nilai RGB 0-255 dapat diskala ulang dalam jangkauan 0-1 agar model CNN dapat lebih mudah mempelajari gambar tersebut, rotasi berarti gambar tersebut akan diputar sesuai dengan sudut yang dimasukan sebagai nilai, *shift* berarti gambar tersebut akan secara acak digeser secara horisontal dan vertikal dengan nilai sesuai yang kita masukan, *zoom* berarti data akan diperbesar maupun diperkecil sesuai dengan nilai yang dimasukan, *flip* berarti data akan diputar secara horisontal maupun vertikal

D. VGG16

Model yang digunakan dalam penelitian ini adalah salah satunya dengan menggunakan VGG16. VGG16 merupakan model CNN yang memanfaatkan convolutional lapisan dengan spesifikasi convolutional tapis yang kecil (3x3). Dengan ukuran tapis konvolusional tersebut, kedalaman jaringan syaraf tiruan dapat dimodifikasi dengan lebih banyak lapisan konvolusional untuk menghasilkan tingkat keakurasian yang lebih tinggi. Model VGG16 mempunyai 19 lapisan yang terdiri dari 16 lapisan konvolusional dan 3 lapisan yang *fully connected*. Arsitektur dari model VGG16 dapat dilihat pada Gambar 6[9].

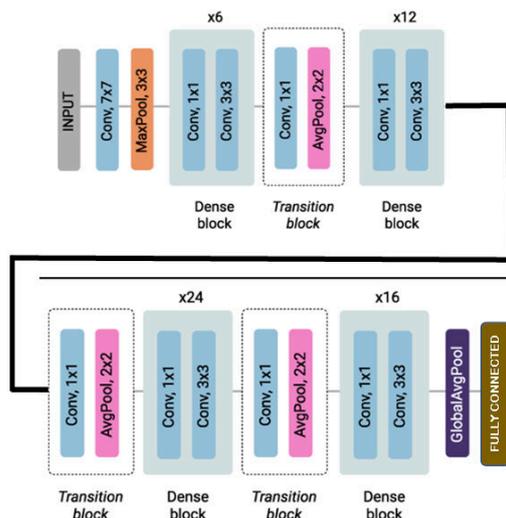


Gambar 6. Arsitektur VGG16



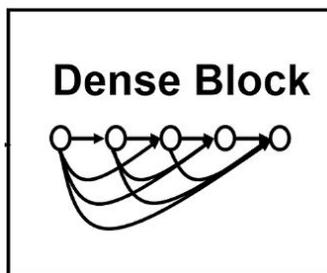
E. DenseNet121

Model lain yang digunakan pada penelitian ini adalah DenseNet121. DenseNet121 adalah sebuah CNN yang terhubung dalam artian menghubungkan seluruh keluaran dari sebuah lapisan dan menggunakan kembali keluaran tersebut sebagai masukan pada lapisan selanjutnya. Arsitektur dari model DenseNet121 dapat dilihat pada Gambar 7[10].



Gambar 7. Arsitektur DenseNet121

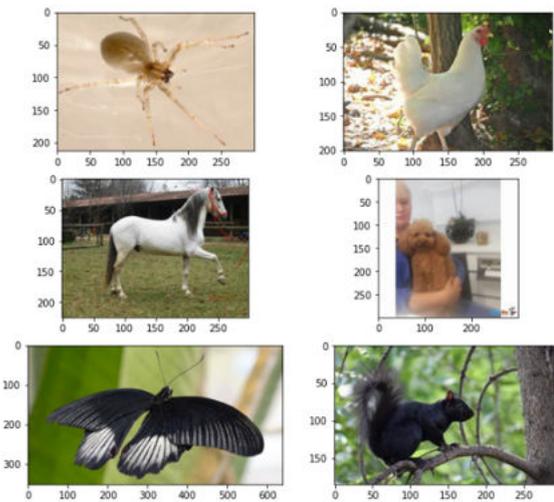
Masing-masing blok *Dense* terhubung dengan blok *Dense* berikutnya sehingga memungkinkan blok *Dense* terakhir bisa menggunakan keluaran dari blok *Dense* pertama, ilustrasi bagaimana blok *Dense* bisa terus terhubung dapat dilihat pada Gambar 8.



Gambar 8. Ilustrasi Blok *Dense*

IV. HASIL DAN PEMBAHASAN

Pengambilan data untuk mendeteksi gambar diambil dari situs Kaggle (<https://www.kaggle.com/datasets/alessiocorrado99/animals10>), citra yang digunakan adalah gambar hewan yang terdiri dari 10 kelas yaitu anjing, kuda, laba-laba, kupu-kupu, domba, sapi, kucing, dan tupai. Sampel yang diambil berjumlah 14500 yang terdiri dari 10000 gambar untuk dilatih, 2500 gambar untuk validasi dan 2000 gambar untuk tes pengujian. Contoh sebagian citra dari dataset.dapat dilihat pada Gambar 9.



Gambar 9. Contoh citra dari dataset

A. Prapengolahan

Pada algoritma yang digunakan dalam penelitian ini, Prapengolahan yang dilakukan adalah *rescale*, *rotasi*, *shift*, *zoom*, dan *flip*. *Rescale* berfungsi agar data dari gambar yang dimasukkan yang memiliki nilai RGB 0-255 dapat diskala ulang dalam jangkauan 0-1 agar model CNN dapat lebih mudah mempelajari gambar tersebut, *rotasi* berarti gambar tersebut akan diputar secara acak dengan sudut maksimal 180 derajat, *shift* berarti gambar tersebut akan secara acak digeser secara horisontal dan vertikal sebesar (-20px) - 20 px, *zoom* berarti data akan diperbesar maupun diperkecil sebesar 80% sampai 120%, *flip* berarti data akan diputar secara horisontal maupun vertikal

B. Pelatihan Model

Pelatihan model CNN VGG16 dan DenseNet121 ini akan menggunakan masukan (224,224,3) sesuai dengan masukan yang digunakan yang digunakan kedua model tersebut pada ImageNet. *Step* tiap *epoch* yang digunakan adalah 200 agar seluruh sampel gambar dapat digunakan sebagai masukan. Karena keterbatasan sumber daya, *epoch* yang digunakan pada pelatihan adalah 30 agar tidak memakan waktu terlalu lama saat menggunakan Google Colab. Hasil dari pelatihan kedua model dapat dilihat pada Tabel 1.

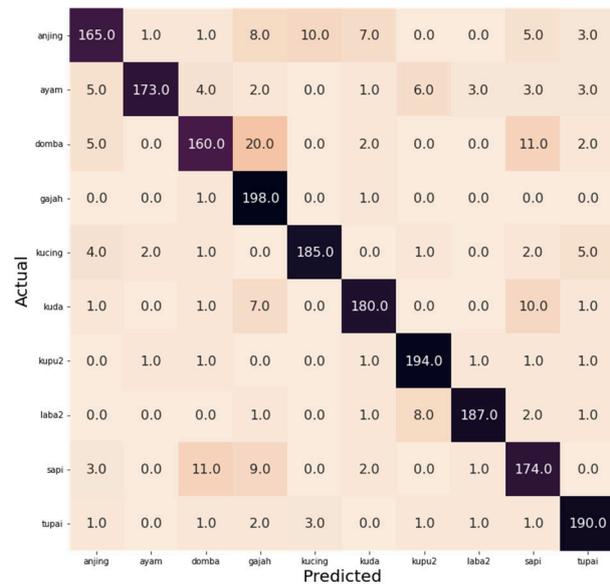
Tabel 1. Hasil pelatihan model

Metrik	VGG16	DenseNet121
<i>Accuracy</i>	0,9481	0,9089
<i>Loss</i>	0,1550	0,2876
<i>Val_accuracy</i>	0,8652	0,9476
<i>Val_loss</i>	0,4447	0,1995

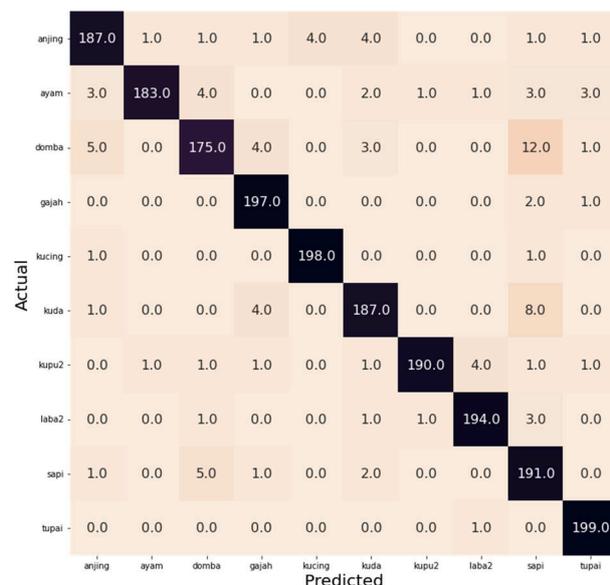


C. Evaluasi Model

Setelah model dilatih maka selanjutnya adalah proses evaluasi. Proses evaluasi dilakukan dengan memanfaatkan confusion matrix untuk mencari nilai Akurasi, Presisi, *Recall*, dan *F1 Score*. Evaluasi ini berfungsi untuk mencari tahu bagaimana nantinya performa model saat dihadapkan dengan citra yang sebelumnya belum pernah dilihat oleh model. Evaluasi dilakukan menggunakan *Confusion Matrix*. Gambar tabel *Confusion Matrix* dari model VGG16 dan DenseNet121 dapat dilihat pada Gambar 10 dan Gambar 11.



Gambar 10. Tabel *Confusion Matrix* VGG16



Gambar 11. Tabel *Confusion Matrix* DenseNet121

Dari tabel confusion matrix, Dari tabel confusion matrix akan didapat nilai-nilai *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN). Kemudian, dari nilai-nilai tersebut dapat dihitung nilai dari Akurasi, *Recall*, Presisi, dan *F1 Score*. Nilai

dari Akurasi, *Recall*, Presisi, dan *F1 Score* dari masing-masing model dapat dilihat pada Gambar 12 dan Gambar 13.

Classification Report:

	precision	recall	f1-score	support
anjing	0.90	0.82	0.86	200
ayam	0.98	0.86	0.92	200
domba	0.88	0.80	0.84	200
gajah	0.80	0.99	0.89	200
kucing	0.93	0.93	0.93	200
kuda	0.92	0.90	0.91	200
kupu2	0.92	0.97	0.95	200
laba2	0.97	0.94	0.95	200
sapi	0.83	0.87	0.85	200
tupai	0.92	0.95	0.94	200
accuracy			0.90	2000
macro avg	0.91	0.90	0.90	2000
weighted avg	0.91	0.90	0.90	2000

Gambar 12. Hasil Evaluasi Model VGG16

Classification Report:

	precision	recall	f1-score	support
anjing	0.94	0.94	0.94	200
ayam	0.99	0.92	0.95	200
domba	0.94	0.88	0.90	200
gajah	0.95	0.98	0.97	200
kucing	0.98	0.99	0.99	200
kuda	0.94	0.94	0.94	200
kupu2	0.99	0.95	0.97	200
laba2	0.97	0.97	0.97	200
sapi	0.86	0.95	0.91	200
tupai	0.97	0.99	0.98	200
accuracy			0.95	2000
macro avg	0.95	0.95	0.95	2000
weighted avg	0.95	0.95	0.95	2000

Gambar 13. Hasil Evaluasi Model DenseNet121

Dari hasil evaluasi menggunakan *confusion matrix* di atas, perbandingan hasil evaluasi model VGG16 dan DenseNet121 dapat dilihat pada Tabel 2.

Tabel 2. Perbandingan hasil evaluasi model

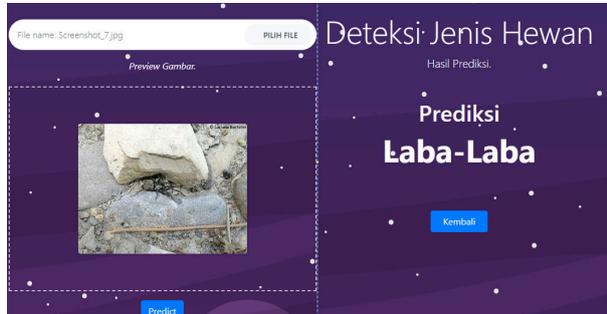
Metrik	VGG16	DenseNet121
Akurasi	0,97	0,98
Presisi	0,90	0,95
Recall	0,90	0,95
F1 Score	0,90	0,95

D. Menjalankan Model pada Aplikasi Web

Setelah dievaluasi, model yang telah dibuat dijalankan pada sebuah aplikasi web sederhana menggunakan kerangka kerja Flask. Cara penggunaannya



pun cukup sederhana, pengguna cukup memasukan gambar lalu tekan tombol prediksi, nantinya model akan memproses gambar dan mengklasifikasikan jenis hewan yang ada di gambar tersebut. Contoh dari hasil prediksi aplikasi web dapat dilihat pada Gambar 14.



Gambar 14. Hasil Prediksi Model Pada Aplikasi Web

V. PENUTUP

A. Kesimpulan

Dari hasil penelitian dan analisis model *Convolutional Neural Network* (CNN) menggunakan model VGG16 dan DenseNet121, dapat disimpulkan bahwa. Berdasarkan hasil pelatihan, model DenseNet121 memiliki tingkat akurasi yang lebih tinggi dari VGG16 dalam mengklasifikasi gambar hewan. Berdasarkan hasil evaluasi menggunakan *confusion matrix*, model DenseNet121 lebih baik dalam mengklasifikasikan gambar hewan daripada VGG16.

DAFTAR PUSTAKA

- [1] Lecun et al, "Gradient based learning applied to document recognition", *Proceedings of IEEE*, no.86 November, pp.2278-2324, 1998.
- [2] Krizhevsky, Alex, Ilya Sutskever, dan Geoffrey E. Hinton, "Imagenet classification with deep convolutional neural networks.", *Communications of the ACM* 60.6 84-90, 2017.
- [3] Tanuwijaya, E. dan A. Roseanne, "Modifikasi Arsitektur VGG16 untuk Klasifikasi Citra Digital Rempah-Rempah Indonesia", *Matrik: Jurnal Manajemen, Teknik Informatika, dan Rekayasa Komputer*, 21(1), 189-196, 2021.
- [4] Rochmawanti, O., F. Utaminingrum, dan F.A. Bachtiar, "Analisis Performa Pre-Trained Model Convolutional Neural Network dalam Mendeteksi Penyakit Tuberkulosis", *Jurnal Teknologi Informasi dan Ilmu Komputer*, 8(4), 805-814, 2021..
- [5] Thunder, Bintang. "Implementasi Algoritma Convolutional Neural Networks di Microsoft Azure untuk Mendeteksi Jenis Kebutaan Mata yang Dialami Penderita Penyakit Diabetes.", 2018.
- [6] Suhardin, Ikbal, Andi Patombongi, dan Andi Muhammad Islah. "Mengidentifikasi Jenis Tanaman Berdasarkan Citra Daun Menggunakan Algoritma Convolutional Neural Network." *Simtek: Jurnal Sistem Informasi dan Teknik Komputer* 6.2, 100-108, 202.
- [7] Danukusumo, Kefin Pudi, "Implementasi Deep Learning Menggunakan Convolutional Neural Network Untuk Klasifikasi Citra Candi Berbasis Gpu". *E-Journal UAJY*, 2017.
- [8] Rahman, Fatur, "Implementasi Convolutional Neural Networks menggunakan VGG-19 Net pada Image Depth Data Hand Posture", *Diss. Universitas Hasanuddin*, 2020
- [9] Simonyan, Karen, dan Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition.", *arXiv preprint arXiv:1409.1556*, 2014.
- [10] Huang, G., Z. Liu, L. van der Maaten, dan K. Q. Weinberger, "Densely Connected Convolutional Networks", *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, ---(---), 2261-2269, 2018
- [11] Samuel Sena, "Pengenalan Deep Learning Part 7 : Convolutional Neural Network", 13 November 2017, <https://medium.com/@samuelsena/pengenalan-deep-learning-part-7-convolutional-neural-network-cnn-b003b477dc94>, diakses pada 10 November 2022
- [12] Putra, M., dan M. Putera, "Analisis Perbandingan metode SOAP dan REST yang digunakan pada Framework Flask untuk membangun Web Service.", *SCAN-Jurnal Teknologi Informasi dan Komunikasi*, 14 (2) 1-7 2019.
- [13] Pangestu, Ridho Aji, Basuki Rahmat, dan Fetty Tri Anggraeny, "Implementasi algoritma CNN untuk klasifikasi citra lahan dan perhitungan luas.", *JIFOSI: Jurnal Informatika dan Sistem Informasi*, 1 (1), 649-654, 2020



©2023 This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/).