

Peningkatan Model Arsitektur Cnn Vgg-16 Menggunakan Fine Tuning Pada Klasifikasi Ras Kucing Melalui Gambar

REVI MOCHAMAD FIKRY^{1*}, YOULLIA INDRAWATY NURHASANAH¹

¹Informatika, Institut Teknologi Nasional
Email : revifikry@gmail.com

Received 27 01 2023 | Revised 03 02 2023 | Accepted 03 02 2023

ABSTRAK

Peningkatan model arsitektur CNN VGG-16 menggunakan fine tuning pada klasifikasi ras kucing melalui gambar merupakan suatu metode yang dapat digunakan untuk meningkatkan kemampuan sistem dalam mengklasifikasikan ras kucing berdasarkan gambar. Penelitian ini dilakukan dengan menggunakan dataset gambar ras kucing yang terdiri dari 11 jenis ras kucing (Bengal, Birman, Bombay, British Shorthair, Dilute Calico, Egyptian Mau, Maine Coon, Persian, Ragdoll, Sphynx, Tuxedo). Model arsitektur CNN VGG-16 yang telah ditraining sebelumnya dioptimalkan dengan metode fine tuning agar dapat lebih akurat dalam mengklasifikasikan ras kucing. Hasil dari penelitian ini menunjukkan bahwa penerapan fine tuning pada model arsitektur CNN VGG-16 dapat meningkatkan kinerja antara model VGG-16 sebagai berikut : Fine Tuning 0 dengan Fine Tuning 1 adalah 27.7%.

Kata kunci: VGG-16, Fine Tuning, Kucing, CNN, Peningkatan Model VGG

ABSTRACT

Improving the VGG-16 CNN architecture model using fine tuning for cat breed classification through images is a method that can be used to improve the system's ability to classify cat breeds based on images. This research was conducted using a dataset of cat breed images consisting of 11 cat breeds (Bengal, Birman, Bombay, British Shorthair, Dilute Calico, Egyptian Mau, Maine Coon, Persian, Ragdoll, Sphynx, Tuxedo). The previously trained VGG-16 CNN architecture model was then optimized using the fine tuning method to be more accurate in classifying cat breeds. The results of this research show that the implementation of fine tuning on the CNN VGG-16 architecture model can improve performance between the VGG-16 model as follows: Fine Tuning 0 with Fine Tuning 1 is 27.7%.

Keywords: VGG-16, Fine Tuning, Cat, CNN, Improve VGG-16 Model

Commented [L1]: Jenis ras kucing disebutkan disini

Commented [RMF2]: peningkatan performa 0,1 0,2 0,3 0,4

Commented [L3]: Jenis ras kucing disebutkan disini

1. PENDAHULUAN

Banyak jenis ras kucing yang diakui secara internasional terdapat 45 jenis kucing yang bersisilah jelas (Cat Fanciers Assosiation., 2020). Contohnya adalah ras kucing persia yang memiliki 4 jenis ras yang tersebar di dunia yaitu kucing persia flatnose, kucing persia peaknose, kucing persia medium dan yang terakhir adalah kucing persia himalaya. (Kurniawan, 2020). Dikarenakan oleh banyaknya ras kucing yang ada diperlukan suatu cara untuk mengklasifikasikan jenis-jenis kucing tersebut, *Convolutional Neural Network* (CNN) dapat membantu untuk mendeteksi dan mengklasifikasikan jenis jenis ras kucing tersebut.

Commented [RMF4]: Gunakan kalimat yang baik dan benar

Convolutional Neural Network (CNN) merupakan salah satu pengembangan dari jaringan syaraf tiruan yang terinspirasi dari jaringan syaraf manusia dan biasa digunakan pada data gambar untuk mendeteksi dan mengenali suatu objek pada sebuah gambar. *Convolutional Neural Network* adalah salah satu metode *deep learning* yang paling terkemuka, dimana *multiple layer* dilatih dengan cara yang kuat. Metode ini sangat efektif dan biasa digunakan dalam computer vision (Guo et al., 2015)

2. LANDASAN TEORI

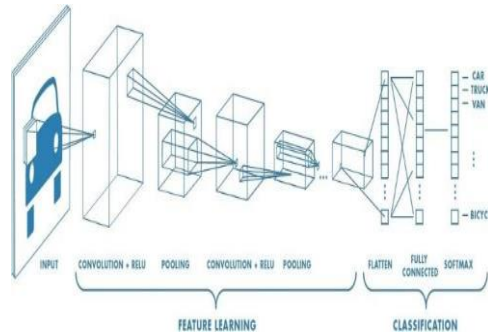
2.1 Deep Learning

Metode *deep learning* tersusun atas layer yang bertingkat/dalam untuk mempelajari suatu ciri dari data dengan level abstraksi yang bertingkat (Lecun et al., 2015). Pendekatan *deep learning* memungkinkan komputer untuk mempelajari suatu model yang rumit dengan cara membangun suatu model yang sederhana (Goodfellow, I., Bengio, Y., & Courville, A., 2016). Berdasarkan hal tersebut, *deep learning* dapat dikatakan sebagai turunan dari *machine learning*, dimana metode ini terdiri dari banyak tingkatan proses informasi non-linear dan abstraksi untuk dapat melakukan *supervised*, *unsupervised learning* dan representation, klasifikasi, dan pengenalan pola (Deng & Yu, 2013).

2.2 CONVOTIONAL NEURAL NETWORK

Convolutional Neural Network adalah salah satu metode *deep learning* yang paling terkemuka, dimana *multiple layer* dilatih dengan cara yang kuat. Metode ini merupakan metode yang sangat efektif dan biasa digunakan pada aplikasi computer vision (Guo et al., 2015), Pada perkembangan CNN terdapat banyak arsitektur yang sering digunakan seperti *AlexNet*, *VGGNet*, *GoogLeNet*, *ResNet*, dll. Secara umum terdapat 3 layer utama pada CNN, yaitu *convolutional layers*, *pooling layers*, dan *fully connected layers*. Pada *convolutional layers*, dilakukan proses operasi konvolusi pada *citra* masukan terhadap *kernel* atau *filter* untuk mendapatkan fitur pada *citra*. Kemudian fungsi *pooling layer* adalah untuk memperkecil dimensi dari fitur yang telah didapat dari convolutional layer. Sedangkan pada *fully connected layer*, citra yang telah diperkecil pada pooling layer dirubah menjadi 1 dimensi dan diklasifikasikan berdasarkan data latih yang dimiliki pada database. Pada Gambar 1 merupakan gambaran arsitektur secara umum CNN.

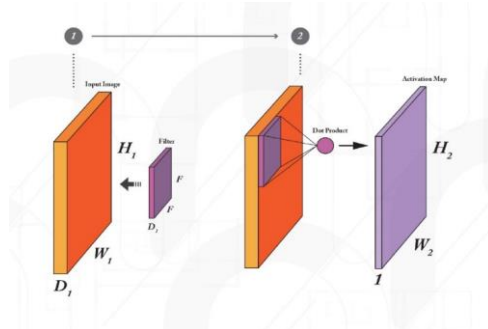
Peningkatan Model Arsitektur Cnn Vgg-16 Menggunakan Fine Tuning Pada Klasifikasi Ras Kucing Melalui Gambar



Gambar. 1 Arsitektur CNN

2.2.1 Convolutional Layer

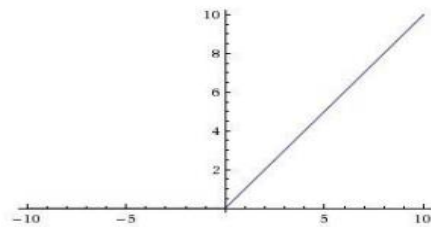
Convolutional Layer bagian yang melakukan operasi konvolusi yaitu mengkombinasikan linier filter terhadap daerah lokal. Layer ini yang pertama kali menerima gambar yang diinputkan pada arsitektur. Bentuk layer ini adalah sebuah filter dengan panjang (pixel), tinggi (pixel), dan tebal sesuai dengan channel image data yang diinputkan. Ketiga filter ini akan bergeser keseluruh bagian gambar. Pergeseran tersebut akan melakukan operasi "dot" antara input dan nilai dari filter tersebut sehingga akan menghasilkan output yang disebut sebagai activation map atau feature map. Gambar 2 menampilkan proses konvolusi yang ada di dalam convolutional layer.



Gambar. 2. Convolutional Layer

2.2.2 Rectified Linear Units (ReLU)

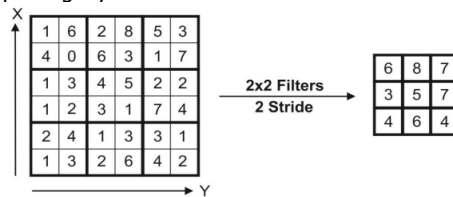
Rectified Linear Units (ReLU) merupakan lapisan aktivasi pada model CNN yang mengaplikasikan fungsi $f(x)=\max(0,x)$ yang berarti fungsi ini melakukan thresholding dengan nilai nol terhadap nilai piksel pada input citra. Aktivasi ini membuat seluruh nilai piksel yang bernilai kurang dari nol pada suatu citra akan dijadikan 0 (Ilahiyah & Nilogiri, 2018).



Gambar. 3 Grafik. ReLU

2.2.3 Pooling Layer

Pooling layer menerima output dari convolutional layer, pada layer ini ukuran data citra akan direduksi. Prinsipnya pooling layer terdiri dari filter dengan ukuran tertentu dan stride/langkah kemudian bergeser keseluruhan area feature map. Sebagian besar arsitektur CNN, metode pooling yang digunakan adalah max pooling. Max pooling membagi output convolutional layer menjadi beberapa grid kemudian setiap pergeseran filter akan mengambil nilai terbesar dari setiap grid. Tergantung pada panjang langkahnya, gambar yang dihasilkan adalah sebagian kecil dari ukuran aslinya yang berguna untuk mengurangi dimensi data, sehingga mengurangi jumlah parameter pada langkah selanjutnya (Shukla & Fricklas, 2018). Gambar 4 menampilkan proses yang ada pada pooling layer

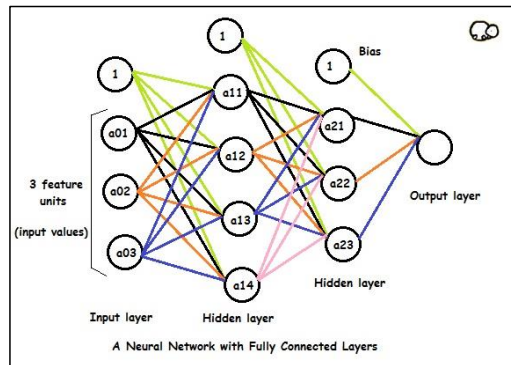


Gambar. 4. Proses Pooling Layer

2.2.4 Fully Connected Layer

Fully connected layer mengambil input dari hasil output pooling layer yang berupa feature map. Feature map tersebut masih berbentuk multidimensional array maka lapisan ini akan melakukan reshape feature map dan menghasilkan vektor sebanyak n-dimensi dimana n adalah jumlah kelas output yang harus dipilih program. Misalnya lapisan terdiri dari 500 neuron, maka akan diterapkan softmax yang mengembalikan daftar probabilitas terbesar untuk masing-masing 10 label kelas sebagai klasifikasi akhir dari jaringan (Darma Nurfita & Ariyanto, 2018). Gambar 5 menampilkan proses yang ada dalam fully connected layer.

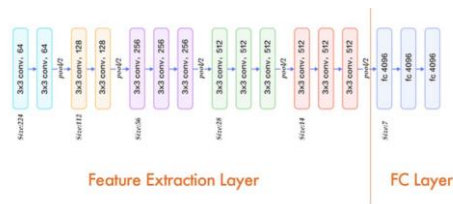
Peningkatan Model Arsitektur Cnn Vgg-16 Menggunakan Fine Tuning Pada Klasifikasi Ras Kucing Melalui Gambar



Gambar. 5. Proses Fully Connected Layer

2.2.5 VGG-16

VGG adalah model arsitektur yang terdiri dari 16 layer terlatih, yaitu layer yang mempunyai bobot (Simonyan & Zisserman, 2014). Gambar model arsitektur VGG dapat dilihat pada Gambar 6



Gambar. 6. Arsitektur VGG-16

2.2.6 Softmax

Softmax merupakan fungsi aktivasi yang digunakan pada layer output. Layer output memiliki banyak kesamaan dengan fully connected layer, yang membedakan kedua layer ini adalah penggunaan fungsi aktivasi softmax pada layer output dan fungsi aktivasi ReLU pada fully-connected layer.

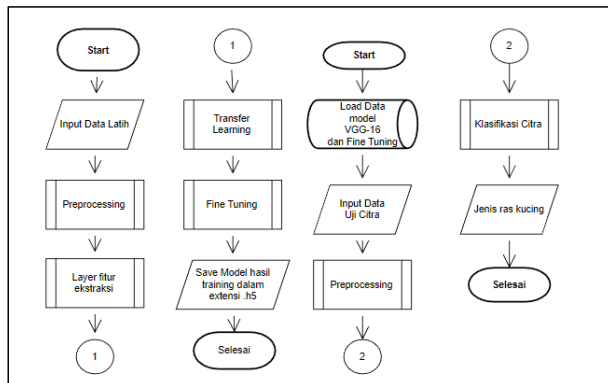
Persamaan fungsi softmax adalah :

$$P_i = \frac{e^{x_i + \log(c)}}{\sum_{k=1}^N e^{x_k + \log(c)}}$$

Pada persamaan softmax x_i merupakan nilai strength neuron dan $\log(c)$ merupakan nilai konstan yang dapat ditentukan, namun secara umum nilai yang ditetapkan adalah $-\max(x)$. Fungsi aktivasi softmax memiliki turunan fungsi yang dijelaskan pada Persamaan : (Bendersky, 2016)

3. METODOLOGI PENELITIAN

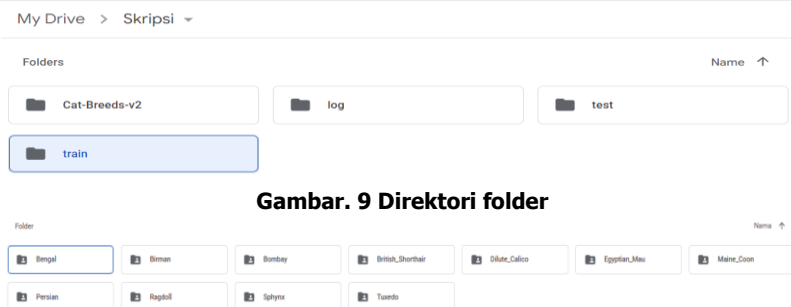
Sebuah penelitian memerlukan tahapan-tahapan yang harus tersusun dengan baik agar pada saat proses pelaksanaan bisa mencapai tujuan dengan baik. Pada penelitian ini dilakukan tahapan yang telah tercantum pada Gambar 8.



Gambar. 8. Flowchart

3.1 Data Latih

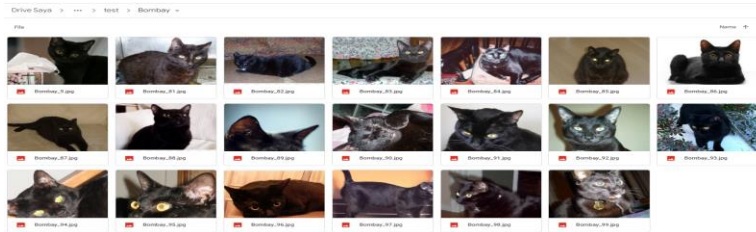
Penelitian ini menggunakan dataset jenis ras kucing yang didapatkan dari <https://www.robots.ox.ac.uk/~vqg/data/pets/>. Data citra yang diperoleh berjumlah 2200 data citra dengan 11 kelas, dimana setiap kelas memiliki 200 data citra. Data pelatihan dibagi menjadi 153 data latih, 27 data validasi dan 20 data uji.



Gambar. 9 Direktori folder

Gambar. 10. Didalam Folder Train

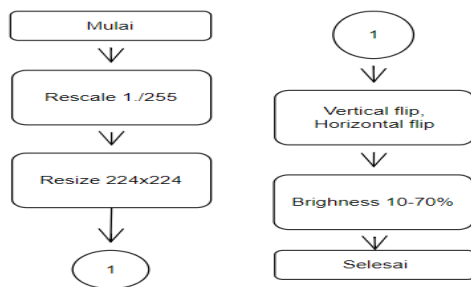
Peningkatan Model Arsitektur Cnn Vgg-16 Menggunakan Fine Tuning Pada Klasifikasi Ras Kucing Melalui Gambar



Gambar.11. Contoh Data Train didalam Folder

3.2 Preprocessing

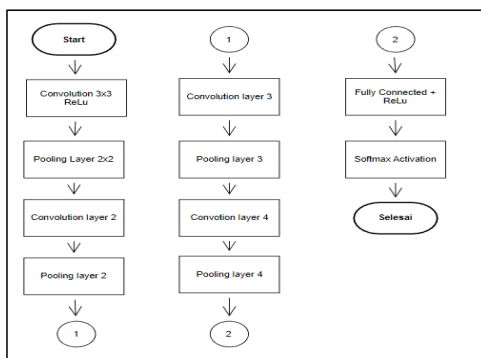
Pada tahap pra-pemrosesan citra dilakukan beberapa proses yang diterapkan pada tiap citra agar data siap digunakan dan dapat disesuaikan pada model arsitektur yang dibangun. Pra-pemrosesan dilakukan menggunakan fitur data generator dari pustaka Keras untuk menerapkan beberapa proses pada tahap pra-pemrosesan yaitu Rescale, Resize, Vertikal dan Horizontal Flip, mengatur kecerahan citra dari 10-70%



Gambar. 12. Flowchart

3.3 VGG-16

Pada sub bab ini akan dijelaskan proses flowchart pada VGG-16 yang penelitian ini gunakan. Gambar 3.4 adalah proses flowchart dari metode VGG-16 yang digunakan.

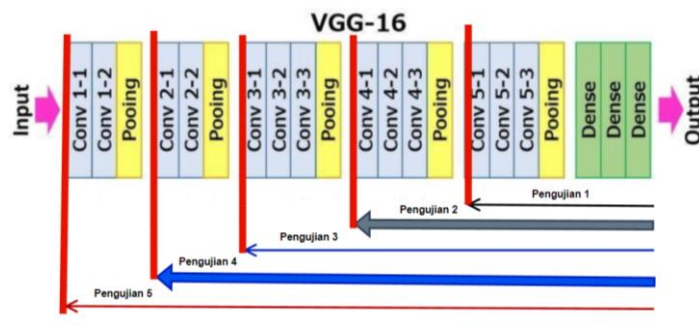


Gambar. 8. VGG-16

Pada proses VGG-16, tahap awal yang dilakukan adalah melakukan Proses input data yang akan dilatih. dari input data yang sudah dimasukkan, data akan di konvolusikan dengan Kernel size 3x3, lalu setelah didapatkan nilai konvolusi proses selanjutnya adalah maxPooling 2x2. Proses VGG-16 terus berulang diantara convolusi dan maxPooling saja. Setelah proses konvolusi dan maxpooling selesai proses terakhir yaitu Fully connected Layer dimana akan mendapatkan output nilai untuk setiap kelasnya. Pada pemrosesan pelatihan data dengan VGG-16 di penelitian ini peneliti menggunakan layer tensorflow dari scratchnya yang sudah disiapkan oleh (Tensorflow Keras Team, 2021) detail dari input, proses, hingga outputnya

3.4 Transfer Learning dan Fine Tuning

Pada sub bab ini akan dijelaskan proses block layer pada Fine Tuning yang penelitian ini gunakan. **Gambar 10** adalah proses block layer dari metode Fine Tuning yang digunakan.



Gambar. 10 Block Layer Fine Tuning

Pada proses Fine Tuning, tahap awal yang dilakukan adalah melanjutkan proses Transfer Learning. Lalu proses Fine Tuning akan mengubah lapisan pretrained layer menjadi non-trainable dan menyisakan beberapa lapisan terakhir dari pretrained layer untuk dilatih sesuai dengan input yang akan dimasukkan. Pada pemrosesan pelatihan data dengan Fine Tuning di penelitian ini menggunakan peneliti menggunakan layer yang sudah dibuat secara sederhana oleh (Tensorflow Keras Team, 2021)

4. IMPLEMENTASI PENGUJIAN

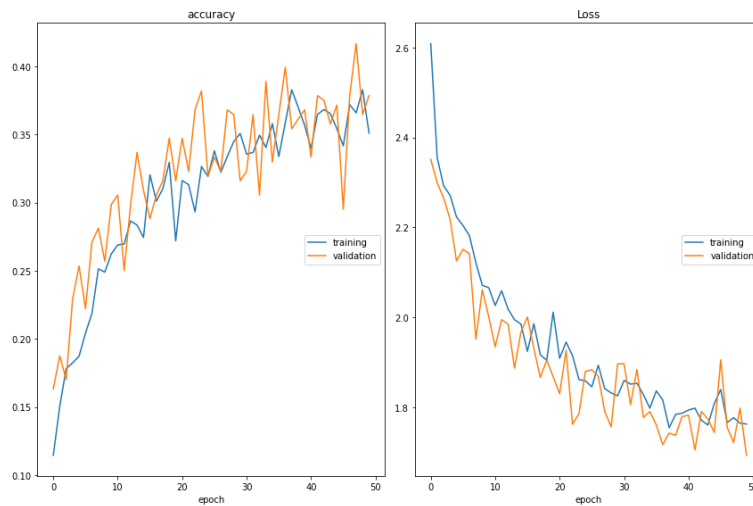
4.1 Pengujian Pelatihan Model Dengan Fine Tuning 0

Dari total pelatihan yang dilakukan menggunakan 50 epoch, dihasilkan riwayat pada epoch 1 loss: 2.604, val_loss: 2.351, accuracy: 0.114, val_accuracy: 0.163. Kemudian nilai accuracy terbesar saat mencapai epoch ke 48 yaitu: loss: 1.755 - accuracy: 0.383 - val_loss: 1.693 - val_accuracy: 0.417. Adapun detail riwayat pelatihan menggunakan Fine Tuning 0 ini ditunjukkan pada Tabel 1

Tabel.1 Pelatihan Dengan Fine Tuning 0

Epoch	Val_accuracy	Accuracy	Val_loss	Loss
1	0.1632	0.114	2.351	2.604
10	0.2621	0.2722	2.3086	2.021
20	0.3421	0.2852	1.7812	1.9231
30	0.3231	0.3324	1.8221	1.8301
40	0.3421	0.3432	1.7723	1.7821
50	0.3612	0.3522	1.7524	1.7732

Peningkatan Model Arsitektur Cnn Vgg-16 Menggunakan Fine Tuning Pada Klasifikasi Ras Kucing Melalui Gambar



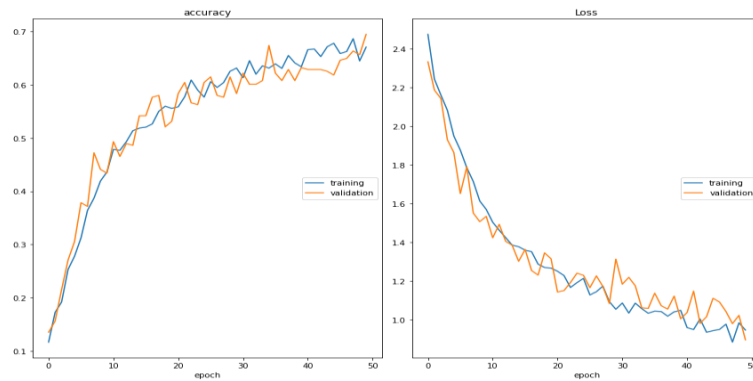
Gambar. 11 Pengujian Pelatihan Model Dengan Fine Tuning 0

4.2 Pengujian Pelatihan Model Dengan Fine Tuning 1

Dari total pelatihan yang dilakukan menggunakan 50 epoch, dihasilkan riwayat pada epoch 1 loss: 2.634, val_loss: 2.373, accuracy: 0.102, val_accuracy: 0.149. Kemudian nilai accuracy mendapatkan nilai akurasi terbesar saat mencapai epoch ke 50 yaitu: loss: 1.747 - accuracy: 0.384 - val_loss: 1.709- val_accuracy: 0.410. Adapun detail riwayat pelatihan menggunakan Learning Rate 0.001 dan Fine Tuning 0 ini ditunjukkan pada Tabel.2

Tabel.2 Pelatihan Dengan Fine Tuning 1

Epoch	Val_accuracy Training	Accuracy Training	Val_loss Training	Loss Training
1	0.135	0.117	2.334	2.475
10	0.318	0.322	1.422	1.453
20	0.512	0.542	1.224	1.252
30	0.572	0.583	1.423	1.124
40	0.621	0.642	0.982	1.223
50	0.694	0.686	0.895	0.883



Gambar. 12 Pengujian Pelatihan Model Dengan Fine Tuning 1

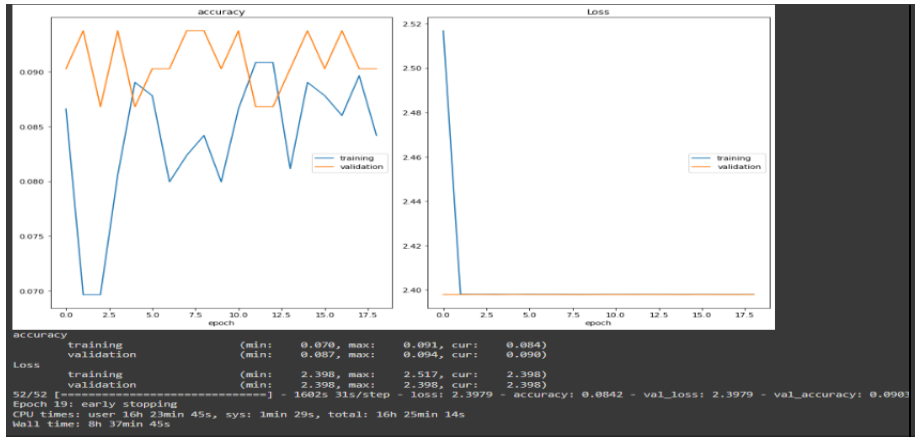
4.3 Pengujian Pelatihan Model Dengan Fine Tuning 2

Hasil pengujian terhadap pelatihan data menggunakan Fine Tuning 2 menghasilkan riwayat berupa loss, val_loss, accuracy, val_accuracy. Peneliti menggunakan total pelatihan sebanyak 30 epoch, namun karena peneliti menggunakan fitur early stopping untuk mencegah overfitting dan mencegah untuk model tidak belajar dari noise pengujian berhenti pada epoch 18, dihasilkan riwayat epoch 1 loss: 2.531, val_loss: 2.4, accuracy: 0.086, val_accuracy: 0.091. Setelah itu grafik dari *proses* training model tidak mengalami peningkatan pada akurasi ataupun pengurangan pada nilai loss sehingga pengujian berhenti pada epoch ke 15. Adapun detail riwayat pelatihan menggunakan Fine Tuning 2 ini ditunjukkan pada Tabel 3.

Tabel.3 Pelatihan Dengan Fine Tuning 2

Epoch	Val_accuracy Training	Accuracy Training	Val_loss Training	Loss Training
1	0.09	0.086	2.4	2.52
5	0.092	0.086	2.4	2.4
10	0.09	0.08	2.4	2.4
15	0.091	0.087	2.4	2.4

Peningkatan Model Arsitektur Cnn Vgg-16 Menggunakan Fine Tuning Pada Klasifikasi Ras Kucing Melalui Gambar



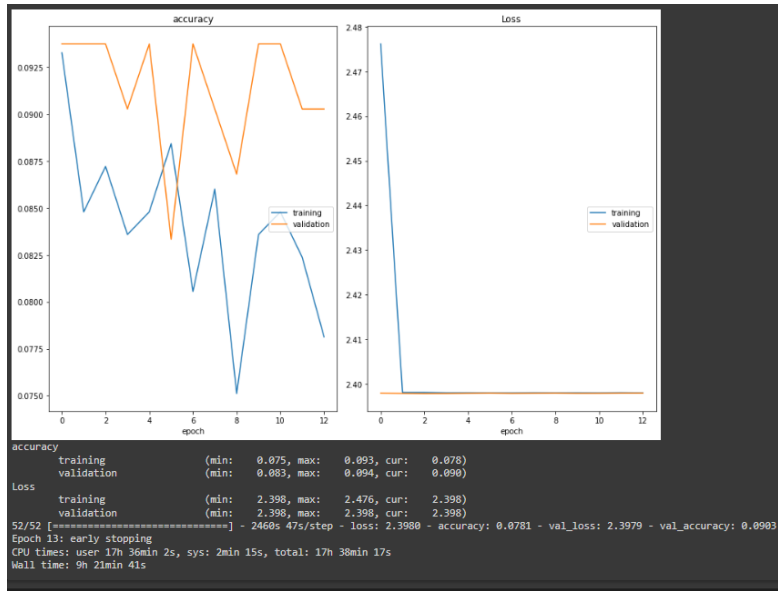
Gambar. 13 Pengujian Pelatihan Model Dengan Fine Tuning 2

4.4 Pengujian Pelatihan Model Dengan Fine Tuning 3

Ditampilkan hasil pengujian terhadap pelatihan data menggunakan Fine Tuning 3 menghasilkan riwayat berupa loss, val_loss, accuracy, val_accuracy. Peneliti menggunakan total pelatihan sebanyak 30 epoch, namun karena peneliti menggunakan fitur early stopping untuk mencegah overfitting dan mencegah untuk model tidak belajar dari noise pengujian berhenti pada epoch 12, dihasilkan riwayat epoch 1 loss: 2.431, val_loss: 2.4, accuracy: 0.092, val_accuracy: 0.092. Setelah itu grafik dari *prose* training model tidak mengalami peningkatan pada akurasi ataupun pengurangan pada nilai loss sehingga pengujian berhenti pada epoch ke 12. Adapun detail riwayat pelatihan menggunakan Fine Tuning 3 ini ditunjukkan pada Tabel 4.

Tabel.4 Pelatihan Dengan Fine Tuning 3

Epoch	Val_accuracy Training	Accuracy Training	Val_loss Training	Loss Training
1	0.092	0.086	2.4	2.52
5	0.086	0.087	2.4	2.4
10	0.093	0.085	2.4	2.4
12	0.91	0.08	2.4	2.4



Gambar. 14 Pengujian Pelatihan Model Dengan Fine Tuning 3

4.5 Pengujian Pelatihan Model Dengan Fine Tuning 4

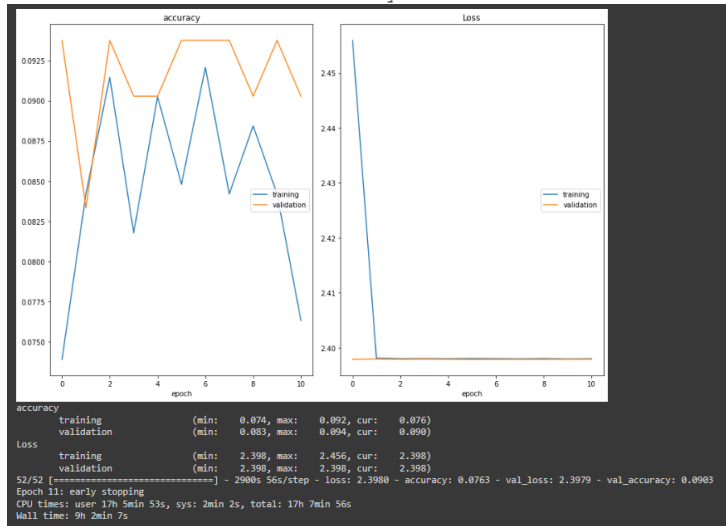
Ditampilkan hasil pengujian terhadap pelatihan data menggunakan Fine Tuning 4 menghasilkan riwayat berupa loss, val_loss, accuracy, val_accuracy. Peneliti menggunakan total pelatihan sebanyak 30 epoch, namun karena peneliti menggunakan fitur early stopping untuk mencegah overfitting dan mencegah untuk model tidak belajar dari noise pengujian berhenti pada epoch 10, dihasilkan riwayat epoch 1 loss: 2.461, val_loss: 2.4, accuracy: 0.0749, val_accuracy: 0.093. Setelah itu grafik dari proses training model tidak mengalami peningkatan pada akurasi ataupun pengurangan pada nilai loss sehingga pengujian berhenti pada epoch ke 10. Adapun detail riwayat pelatihan menggunakan Fine Tuning 4 ini ditunjukkan pada Tabel 4.5

Tabel.5 Pelatihan Dengan Fine Tuning 4

Epoch Training	Val_accuracy Training	Accuracy Training	Val_loss Training	Loss Training
1	0.093	0.0749	2.4	2.45
5	0.093	0.892	2.4	2.4
10	0.091	0.077	2.4	2.4

Peningkatan Model Arsitektur Cnn Vgg-16 Menggunakan Fine Tuning Pada Klasifikasi Ras Kucing Melalui Gambar

Tabel 4. 1 Riwayat Pelatihan



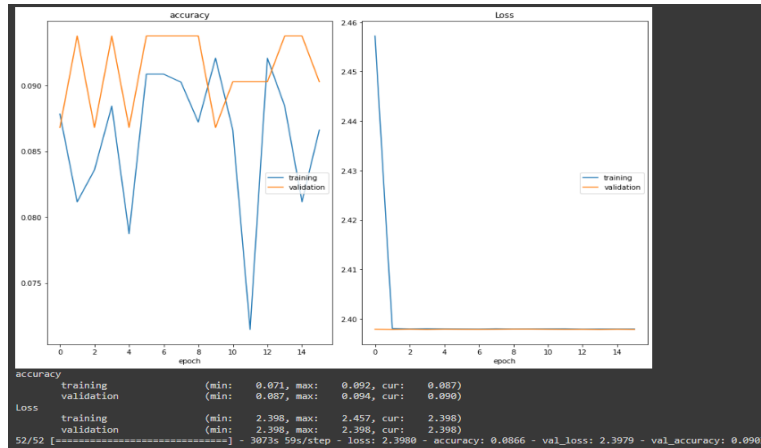
Gambar. 15 Pengujian Pelatihan Model Dengan Fine Tuning 4

4.6 Pengujian Pelatihan Model Dengan Fine Tuning 5

Ditampilkan hasil pengujian terhadap pelatihan data menggunakan Fine Tuning 5 menghasilkan riwayat berupa loss, val_loss, accuracy, val_accuracy. Peneliti menggunakan total pelatihan sebanyak 30 epoch, namun karena peneliti menggunakan fitur early stopping untuk mencegah overfitting dan mencegah untuk model tidak belajar dari noise pengujian berhenti pada epoch 14, dihasilkan riwayat epoch 1 loss: 2.451, val_loss: 2.4, accuracy: 0.086, val_accuracy: 0.086. Setelah itu grafik dari proses training model tidak mengalami peningkatan pada akurasi ataupun pengurangan pada nilai loss sehingga pengujian berhenti pada epoch ke 15. Adapun detail riwayat pelatihan menggunakan Fine Tuning 5 ini ditunjukkan pada Tabel 5.

Tabel.6 Pelatihan Dengan Fine Tuning 5

Epoch	Val_accuracy Training	Accuracy Training	Val_loss Training	Loss Training
1	0.086	0.086	2.4	2.45
5	0.086	0.079	2.4	2.4
10	0.089	0.087	2.4	2.4
14	0.085	0.091	2.4	2.4



Gambar. 16 Pengujian Pelatihan Model Dengan Fine Tuning 5

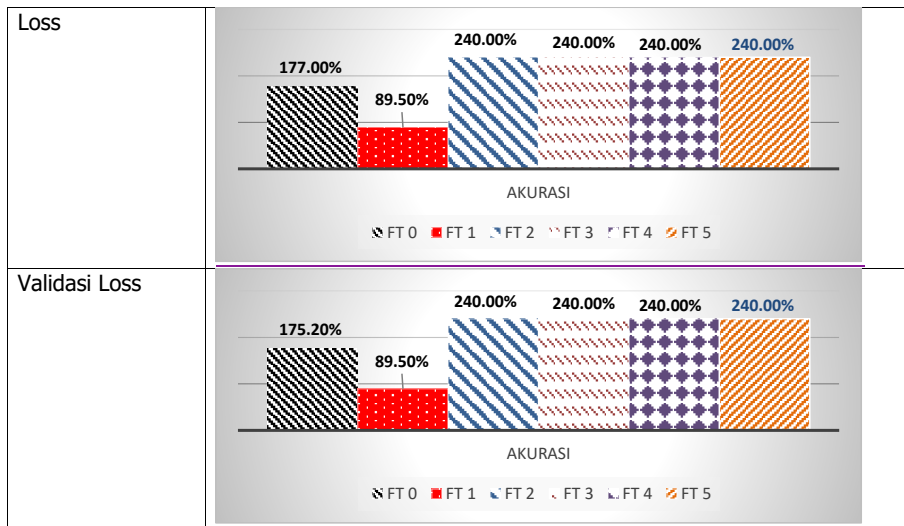
4.7 Perbandingan Nilai Akurasi, Validasi Akurasi, Loss dan Validasi Loss

Tabel 7 adalah perbandingan dari nilai Akurasi, Validasi Akurasi, Loss, dan Validasi Loss yang telah didapatkan dari proses pelatihan yang sudah dilakukan dari hyperparameter learning rate 0.001 hingga 0.0001 dengan Fine Tuning 0 dan 2. Dari perbandingan yang dapat kita lihat bahwa hyperparameter Skenario Pengujian Kinerja Sistem dengan learning rate 0.0001 dan Fine Tuning 2 adalah model yang best fit karena nilai akurasi dan validasi akurasi yang tinggi dengan nilai loss dan validasi loss yang kecil.

Tabel. 7

Perbandingan	Grafik
Akurasi	AKURASI FT 0 FT 1 FT 2 FT 3 FT 4 FT 5
Validasi Akurasi	VALIDASI AKURASI FT 0 FT 1 FT 2 FT 3 FT 4 FT 5

Peningkatan Model Arsitektur Cnn Vgg-16 Menggunakan Fine Tuning Pada Klasifikasi Ras Kucing Melalui Gambar



4.8 Pengujian Dengan Data Testing

Pada eksperimen ini, dilakukan penggabungan model antara metode CNN-VGG-16 dengan Fine Tuning dan melihat hasil dari akurasi, presisi, recall, dan f1- score dari setiap konfigurasi learning rate pada Fine Tuning 0 sampai dengan Fine Tuning 5, hasil akurasi dengan beberapa skenario tersebut akan disajikan di Gambar berdasarkan confusion matrix

Class : 0 = Bengal
 1 = Birman
 2 = Bombay
 3 = British Shorthair
 4 = Dilute Calico
 5 = Egyptian Mau
 6 = Maine Coon
 7 = Persian
 8 = Ragdoll
 9 = Sphynx
 10 = Tuxedo

VGG16 Model Accuracy Fine-Tuning = 0 : 43.18%				
	precision	recall	f1-score	support
0	0.56	0.75	0.64	20
1	0.00	0.00	0.00	20
2	0.42	1.00	0.59	20
3	0.00	0.00	0.00	20
4	0.20	0.05	0.08	20
5	0.65	0.55	0.59	20
6	0.23	0.75	0.35	20
7	0.61	0.55	0.58	20
8	0.00	0.00	0.00	20
9	1.00	0.25	0.40	20
10	0.50	0.85	0.63	20
accuracy			0.43	220
macro avg	0.38	0.43	0.35	220
weighted avg	0.38	0.43	0.35	220

Gambar 17 Fine Tuning 0

Dapat dilihat pada Gambar 17 jika nilai *Precision*, *Recall*, dan *F1 – Score* terburuk terdapat pada ras kucing ke 1 (Birman), kucing ke 3 (British Shorthair), dan kucing ke 8 (Ragdoll).

	precision	recall	f1-score	support
0	0.67	0.60	0.63	20
1	0.67	0.80	0.73	20
2	0.45	0.90	0.60	20
3	1.00	0.50	0.67	20
4	0.50	0.60	0.55	20
5	1.00	0.15	0.26	20
6	0.52	0.75	0.61	20
7	0.76	0.65	0.70	20
8	1.00	0.15	0.26	20
9	0.67	0.70	0.68	20
10	0.58	0.90	0.71	20
accuracy			0.61	220
macro avg	0.71	0.61	0.58	220
weighted avg	0.71	0.61	0.58	220

Gambar 18 Fine Tuning 1

Dapat dilihat pada Gambar 18 dapat dilihat bahwa *Precision* terkecil ada pada ras 2 (Bombay) *Recall* terkecil ada pada ras 5 (Egyptian Mau), 8 (Ragdoll) dan f1-score terkecil ada pada ras nomer 8.

	precision	recall	f1-score	support
0	0.00	0.00	0.00	20
1	0.00	0.00	0.00	20
2	0.09	1.00	0.17	20
3	0.00	0.00	0.00	20
4	0.00	0.00	0.00	20
5	0.00	0.00	0.00	20
6	0.00	0.00	0.00	20
7	0.00	0.00	0.00	20
8	0.00	0.00	0.00	20
9	0.00	0.00	0.00	20
10	0.00	0.00	0.00	20
accuracy			0.09	220
macro avg	0.01	0.09	0.02	220
weighted avg	0.01	0.09	0.02	220

Gambar 19 Fine Tuning 2

Dapat dilihat pada Gambar 19 jika nilai *Precision*, *Recall*, dan *F1 – Score* dari hasil Fine Tuning layer 2 memiliki tingkat akurasi 9%

Peningkatan Model Arsitektur Cnn Vgg-16 Menggunakan Fine Tuning Pada Klasifikasi Ras Kucing Melalui Gambar

	precision	recall	f1-score	support
0	0.00	0.00	0.00	20
1	0.09	1.00	0.17	20
2	0.00	0.00	0.00	20
3	0.00	0.00	0.00	20
4	0.00	0.00	0.00	20
5	0.00	0.00	0.00	20
6	0.00	0.00	0.00	20
7	0.00	0.00	0.00	20
8	0.00	0.00	0.00	20
9	0.00	0.00	0.00	20
10	0.00	0.00	0.00	20
accuracy			0.09	220
macro avg	0.01	0.09	0.02	220
weighted avg	0.01	0.09	0.02	220

Gambar 20 Fine Tuning 3

Dapat dilihat pada Gambar 20 jika nilai *Precision*, *Recall*, dan *F1 – Score* dari hasil Fine Tuning layer 3 memiliki tingkat akurasi 9%

	precision	recall	f1-score	support
0	0.00	0.00	0.00	20
1	0.00	0.00	0.00	20
2	0.00	0.00	0.00	20
3	0.00	0.00	0.00	20
4	0.00	0.00	0.00	20
5	0.00	0.00	0.00	20
6	0.08	0.25	0.12	20
7	0.00	0.00	0.00	20
8	0.00	0.00	0.00	20
9	0.08	0.60	0.14	20
10	0.00	0.00	0.00	20
accuracy			0.08	220
macro avg	0.01	0.08	0.02	220
weighted avg	0.01	0.08	0.02	220

Gambar 21 Fine Tuning 4

Dapat dilihat pada Gambar 21 jika nilai *Precision*, *Recall*, dan *F1 – Score* dari hasil Fine Tuning layer 4 memiliki tingkat akurasi 8%

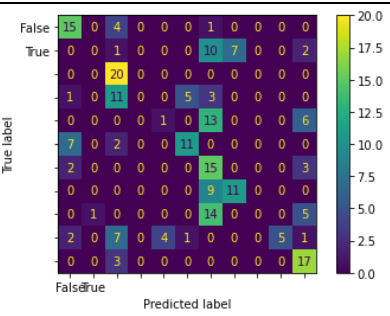
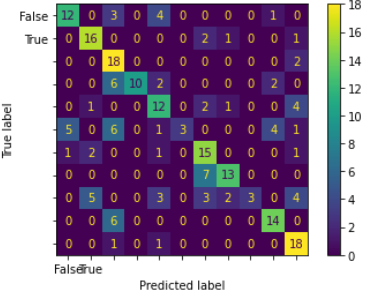
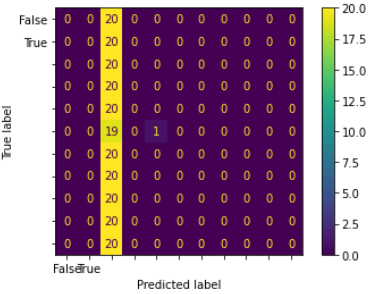
	precision	recall	f1-score	support
0	0.00	0.00	0.00	20
1	0.00	0.00	0.00	20
2	0.00	0.00	0.00	20
3	0.00	0.00	0.00	20
4	0.00	0.00	0.00	20
5	0.00	0.00	0.00	20
6	0.08	0.25	0.12	20
7	0.00	0.00	0.00	20
8	0.00	0.00	0.00	20
9	0.08	0.60	0.14	20
10	0.00	0.00	0.00	20
accuracy			0.08	220
macro avg	0.01	0.08	0.02	220
weighted avg	0.01	0.08	0.02	220

Gambar 22 Fine Tuning 5

Dapat dilihat pada Gambar 22 jika nilai *Precision*, *Recall*, dan *F1 – Score* dari hasil Fine Tuning layer 5 memiliki tingkat akurasi 8%
buruk

Hasil pengujian dari setiap rasnya akan ditampilkan secara rinci pada tabel 4.17 yang akan menampilkan *Confusion Matrix* dari setiap modelnya.

Tabel 4. 2 Tabel Confusion Matrix

Confusion Matrix	Keterangan
 Confusion matrix with Fine Tuning 0, dapat dilihat bahwa kesalahan deteksi terbesar ada pada ras kucing birman (1), Dilute Calico, Egyptian mau dan Ragdoll.	Confusion matrix dengan Fine Tuning 0, dapat dilihat bahwa kesalahan deteksi terbesar ada pada ras kucing birman (1), Dilute Calico, Egyptian mau dan Ragdoll.
 Confusion matrix with Fine Tuning 1, dapat dilihat bahwa kesalahan deteksi terbesar ada pada ras kucing birman dan ragdoll.	Confusion matrix dengan Fine Tuning 1, dapat dilihat bahwa kesalahan deteksi terbesar ada pada ras kucing birman dan ragdoll.
 Dikarenakan oleh akurasi yang sangat kecil confusion matrix Fine Tuning 2 mendeteksi semua input ras sebagai ras ke – 3 (Bombay)	Dikarenakan oleh akurasi yang sangat kecil confusion matrix Fine Tuning 2 mendeteksi semua input ras sebagai ras ke – 3 (Bombay)

Peningkatan Model Arsitektur Cnn Vgg-16 Menggunakan Fine Tuning Pada Klasifikasi Ras Kucing Melalui Gambar

True label False True Predicted label	Dikarenakan oleh akurasi yang sangat kecil confusion matrix Fine Tuning 3 mendeteksi semua input ras sebagai ras ke – 1 (Birman)
True label False True Predicted label	Dikarenakan oleh akurasi yang sangat kecil confusion matrix Fine Tuning 4 mendeteksi semua input ras sebagai ras ke – 6 (Mainecoon) atau 10 (Sphynx)
True label False True Predicted label	Dikarenakan oleh akurasi yang sangat kecil confusion matrix Fine Tuning 5 mendeteksi semua input ras sebagai ras ke – 6 (Mainecoon) atau 10 (Sphynx)

Dapat dilihat pada proses Fine Tuning 0 dan Fine Tuning 1 bahwa. Ras kucing 1 (Birman) sering kali diprediksi sebagai ras kucing 6, ras kucing 5 (Egyptian Mau) sering kali diprediksi sebagai ras kucing 0 (Bengal), dan ras kucing 8 (Ragdoll) sering kali diprediksi sebagai ras kucing 2 (Birman).

Hal ini terjadi karena kemiripan fitur dari ras kucing tersebut yang mengakibatkan nilai *Precision*, *Recall*, dan *F1-Score* dari ras kucing tersebut menjadi kurang baik, kemiripan fitur pada ras kucing dapat dilihat pada Tabel 4.18.

Tabel 4. 3 Tabel Persamaan Kucing

Tabel Kemiripan Ras Kucing			
Ras kucing 1 (Birman) dan 6 (Persia)			
	<i>Birman</i>	<i>Persia</i>	
aRas kucing 5 (Egyptian Mau) dan 0 (Bengal)			
	<i>Egyptian Mau</i>	<i>Bengal</i>	
Ras kucing 8 (Ragdoll) dan 2 (Birman)			
	<i>Ragdoll</i>	<i>Birman</i>	

5. KESIMPULAN

Pada penelitian ini telah diimplementasikan berbagai model menggunakan deep learning CNN dan VGG-16 untuk mengklasifikasikan ras pada kucing. Pada penelitian ini dapat dikatakan bahwa Fine Tuning berhasil untuk meningkatkan performa model, namun terlalu banyak lapisan dari Fine Tuning yang digunakan dapat berpengaruh kepada kinerja model.

Dalam Penelitian ini model yang dengan Fine Tuning 0 mendapatkan nilai akurasi sebesar 35.2%, Fine Tuning 1 sebesar 68.2%, Fine Tuning 2 sebesar 8,7%, Fine Tuning 3 sebesar 9.25%, Fine Tuning 4 sebesar 9.1%, dan Fine Tuning 5 sebesar 9%.

Peningkatan kinerja antara model VGG-16 Fine Tuning 0 dengan Fine Tuning 1 adalah 27.7%, sedangkan untuk model Fine Tuning 2 sampai dengan Fine Tuning 5 model mengalami penurunan hal ini bisa disebabkan oleh *Overfitting*, dataset yang terlalu mirip, penyusunan data yang kurang sesuai, dan kesalahan atau kekurangan pada *preprocessing data*.

DAFTAR PUSTAKA

- [1]. Darna Nurfitra, R., & Ariyanto, G. (2018). *IMPLEMENTASI DEEP LEARNING BERBASIS TENSORFLOW UNTUK PENGENALAN SIDIK JARI*. <http://bias.csr.unibo.it/fvc2004/databases.asp>
- [2]. Deng, L., & Yu, D. (2013). Deep learning: Methods and applications. In *Foundations and Trends in Signal Processing* (Vol. 7, Issues 3–4, pp. 197–387). Now Publishers Inc. <https://doi.org/10.1561/20000000039>
- [3]. Guo, Y., Liu, Y., Oerlemans, A., Lao, S., Wu, S., & Lew, M. S. (2015). *Deep learning for visual understanding: A review*. <https://doi.org/10.1016/j.neucom.2015.09.116>
- [4]. Ilahiyah, S., & Nilogiri, A. (2018). *Implementasi Deep Learning Pada Identifikasi Jenis Tumbuhan Berdasarkan Citra Daun Menggunakan Convolutional Neural Network*.
- [5]. Kurniawan, A. (2020, April 7). *4 Jenis Kucing Persia, Kenali Ciri Khasnya Masing-masing* | merdeka.com. <https://www.merdeka.com/jabar/4-jenis-kucing-persia-kenali-ciri-khasnya-masing-masing-klh.html>
- [6]. Lauw, K. O., Santoso, L. W., & Intan, R. (2020). *Identifikasi Jenis Anjing Berdasarkan Gambar Menggunakan Convolutional Neural Network Berbasis Android*.
- [7]. Lecun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. In *Nature* (Vol. 521, Issue 7553, pp. 436–444). Nature Publishing Group. <https://doi.org/10.1038/nature14539>
- [8]. Ridwan, M., Sistem, J., Univeristas, I., Husni, M., Jakarta, T., & Raya Pd, J. (2018). *SISTEM DETEKSI WAJAH JENIS KUCING DENGAN IMAGE CLASSIFICATION MENGGUNAKAN OPENCV*.
- [9]. Shukla, N., & Fricklas, K. (2018). *ML With TensorFlow*. <https://www.manning.com/books/machine-learning-with-tensorflow>