

IMPLEMENTASI DEEP LEARNING DENGAN MENGGUNAKAN ALGORITMA CONVOLUTIONAL NEURAL NETWORK UNTUK MENGIDENTIFIKASI JENIS IKAN LAUT

Nada Aqila Mahmud*¹⁾, Budi Hartono²⁾

1. Universitas Stikubank, Semarang, Indonesia
2. Universitas Stikubank, Semarang, Indonesia

Article Info

Kata Kunci: *Resnet-50; Convolutional Neural Network; Artificial Intelligence; Deep Learning; Ikan Air Laut*

Keywords: *Resnet-50; Convolutional Neural Network; Artificial Intelligence; Deep Learning; Seawater Fish*

Article history:

Received 1 February 2024
Revised 15 February 2024
Accepted 29 February 2024
Available online 1 June 2024

DOI :

<https://doi.org/10.29100/jipi.v9i2.4477>

* Corresponding author.
Nada Aqila Mahmud
E-mail address:
nadaaqila59@gmail.com

ABSTRAK

Indonesia memiliki wilayah laut yang luas sehingga diperlukan pengelolaan yang tepat untuk memanfaatkan potensi kekayaan sumber daya alam agar dapat memberikan keuntungan bagi kehidupan masyarakat terutama pada sektor perikanan. Untuk membantu dalam mengelola potensi sektor kelautan Indonesia yang besar diperlukan inovasi yang baru, hal ini dapat disolusikan dengan sistem identifikasi menggunakan metode deep learning yang menggunakan algoritma convolutional neural network dengan arsitektur ResNet-50. Data yang digunakan terdiri dari 900 citra data yang terbagi menjadi 9 kelas yaitu Black Sea Spart, Gilt-Head Bream, House Mackerel, Red Mullet, Red Sea Bream, Sea Bass, Shrimp, Striped Red Mullet, dan Trout yang diambil dari website Kaggle. Dalam membangun model klasifikasi diterapkan konfigurasi dari epoch dan learning rate dengan perbandingan data latih, data validasi, dan data uji sebesar 60:20:20 dan didapatkan performa dari sistem memiliki kecenderungan meningkat dengan bertambahnya jumlah epoch. Performa terbaik dihasilkan oleh model dengan epoch 100 dan learning rate 0.001 dengan akurasi sebesar 97.92%. sehingga dapat disimpulkan bahwa arsitektur ResNet-50 dapat mengidentifikasi jenis ikan laut.

ABSTRACT

Indonesia has a large sea area so proper management is needed to utilize the potential wealth of natural resources in order to provide benefits for people's lives, especially in the fisheries sector. To help manage the huge potential of Indonesia's marine sector, new innovations are needed, this can be solved by an identification system using deep learning methods that use convolutional neural network algorithms with the ResNet-50 architecture. The data used consists of 900 data images divided into 9 classes, namely Black Sea Spart, Gilt-Head Bream, House Mackerel, Red Mullet, Red Sea Bream, Sea Bass, Shrimp, Striped Red Mullet, and Trout taken from the Kaggle website. In building a classification model, a configuration of epochs and learning rates is applied with a comparison of training data, validation data, and test data of 60:20:20 and the performance of the system tends to increase with the increase in the number of epochs. The best performance was produced by a model with an epoch of 100 and a learning rate of 0.001 with an accuracy of 97.92%. So it can be concluded that the ResNet-50 architecture can identify the type of marine fish.

I. PENDAHULUAN

INDONESIA sebagai negara kepulauan terbesar di dunia dan terluas di Asia Tenggara, dengan jumlah pulau mencapai 17.499 yang terbagi secara merata baik pulau kecil maupun pulau besar. Memiliki total luas wilayah 7.81 juta km², dengan 3.1 juta km² merupakan luas wilayah perairan dan 2.7 juta km² masuk dalam Zona Ekonomi Eksklusif (ZEE), hanya sekitar 2.01 juta km² saja luas daratan yang dimiliki Indonesia (Badan Informasi Geospasial, 2021).[1]

Indonesia juga merupakan salah satu negara yang memiliki garis pantai terpanjang kedua setelah Kanada dengan panjang 99.093 km². Dengan letak geografis tersebut, Indonesia mempunyai iklim laut yang lembab, kelembapan udara yang tinggi, dan suhu yang tidak begitu berbeda antara siang dan malam.

Pengelolaan sumber daya kelautan yang masih tergolong rendah ditunjukkan dengan timbulnya maraknya pencurian ikan. Di sisi lain, perikanan juga menjadi salah satu sektor yang diandalkan untuk pembangunan nasional selain dari sektor pertanian. (Direktorat Jenderal Pengelolaan Ruang Laut, 2019) hasil ekspor perikanan Indonesia mencapai Rp 73.681.883.000 yang dimana hasil tersebut naik sebesar 10.1% dibandingkan pada tahun 2018 sedangkan pada tahun 2021, nilai ekspor produk perikanan meningkat 17,7%. [2]

Dalam proses untuk meningkatkan nilai ekspor dalam perikanan, Menteri Kelautan dan Perikanan Sakti Wahyu Trenggono menargetkan, ekspor perikanan pada tahun 2023 sebesar USD 7,66 Miliar atau 30,37 juta ton, jumlah tersebut terdiri dari perikanan tangkap sebanyak 8,73 juta ton, dan perikanan budi daya sebesar 21, 58 juta ton. pertumbuhan untuk produk domestik bruto ditargetkan di angka 5-6%.

Untuk menjaga ekosistem laut, pemerintah mengeluarkan peraturan untuk tetap mempertahankan wilayah konservasi yang menerapkan konsep ekonomi biru yang berkelanjutan atau *sustainable blue economy*. Dalam rangka itu pula, pemerintah mengeluarkan Peraturan Pemerintah (PP) Nomor 11 Tahun 2023 tentang penangkapan Ikan Terukur yang ditandatangani oleh presiden Joko Widodo pada tanggal 6 Maret 2023. Berkaitan dengan itu, Menteri Sakti Wahyu Trenggono mengatakan bahwa luas Kawasan konservasi diperluas sebesar 29,1 juta hektar dan proporsi tangkap jenis ikan yang berada dalam Batasan biologis yang aman yaitu 76% dan akan didukung dengan sarana, prasarana, dan pengawasan yang optimal. Komoditas utama ekspor perikanan Indonesia meliputi udang dengan nilai USD 1.997,49 juta, Tuna-Cakalang-Tongkol senilai USD 865,73 juta, Cumi-Sotong-Gurita sebesar USD 657,71 juta, Rumput Laut sebesar USD 554,96 juta dan Rajungan-Kepiting sebesar USD 450,55 juta. Tujuan negara eksportir utama meliputi Amerika Serikat, Tiongkok, Jepang, Asean, serta 27 Uni Eropa. [3]

Untuk membantu dalam mengelola potensi sektor kelautan Indonesia yang besar diperlukan inovasi yang baru, hal ini dapat disolusikan dengan sistem identifikasi menggunakan metode *deep learning* yang merupakan sub-bagian dari *Machine Learning* yang terinspirasi dari jaringan saraf tiruan manusia dengan menggunakan algoritma *Convolutional Neural Network* atau CNN. Pada penelitian ini akan memanfaatkan hasil pembelajaran dari algoritma CNN yaitu ResNet-50 yang digunakan untuk mengidentifikasi citra jenis ikan laut, cara kerja dari ResNet50 adalah dengan memanfaatkan *skip connection* dan terdapat *batch normalization* sehingga mampu menyelesaikan masalah pada *vanishing gradient* yang banyak dimiliki oleh arsitektur CNN lainnya. Maka dari itu manfaat dari penelitian ini adalah untuk mengetahui akurasi yang dihasilkan dari metode ResNet-50 dalam mengidentifikasi jenis ikan laut sehingga akan mempermudah dalam mengelola potensi kelautan Indonesia. Selain itu dapat menjadi pertimbangan untuk penelitian selanjutnya yang akan menggunakan ResNet-50 dalam mengklasifikasikan citra gambar.

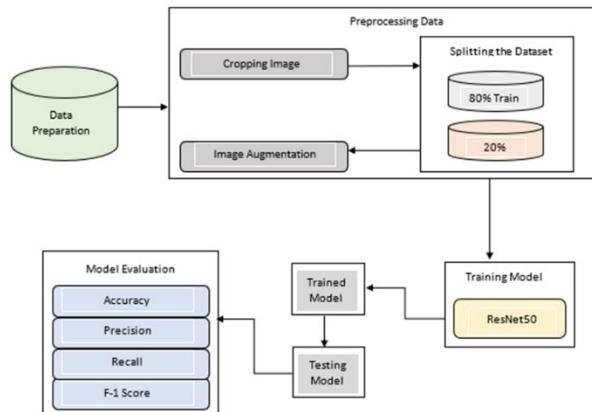
Pada penelitian sebelumnya membahas penggunaan metode dari CNN untuk pengolahan citra. Sistem ini digunakan untuk membedakan ikan yang segar dan tidak segar berdasarkan dari warna mata pada ikan dan menggunakan *data training* sebanyak 24 citra yang terdiri dari 12 citra ikan segar dan 12 citra ikan tidak segar. Penelitian ini menghasilkan nilai akurasi 100% dengan nilai *loss* sebesar 0.00000199354 yang didapatkan dari jumlah *epoch* adalah 5 [4]. Selain itu dengan menambahkan jumlah *epoch* pada saat proses *training* akan membuat *model* menjadi semakin baik.

Penelitian yang membahas tentang perbandingan model AlexNet dan ResNet dalam mengklasifikasikan citra Bunga yang menggunakan *dataset* berjumlah 7370 citra gambar yang terbagi menjadi 6552 sebagai *data training* dan 818 sebagai data uji. Menghasilkan kinerja ResNet lebih baik dibandingkan dengan kinerja dari AlexNet, hasil akurasi yang didapatkan dari model AlexNet hanya sebesar 90.2% sedangkan model ResNet menghasilkan nilai akurasi sebesar 97.3% [5]. Dengan jumlah *dataset* yang tergolong sedikit, *model* akan tetap mampu menghasilkan nilai akurasi yang tinggi.

Untuk penelitian selanjutnya membahas tentang klasifikasi sidik jari menggunakan ResNet50, *dataset* yang digunakan mempunyai dimensi 512x512 dengan total *dataset* berjumlah 2100 citra yang terbagi menjadi 429 citra yang memiliki pola *arch*, 403 citra berpola *tented arch*, 402 citra berpola *left loop*, 410 citra berpola *right loop*, dan 456 citra berpola *whorl*. Penelitian tersebut dapat mengidentifikasi 5 pola sidik jari dengan akurasi pelatihan sebesar 99.52%, akurasi validasi 95.05%, *loss* pelatihan 0.016 dan *loss* validasi 0.229 yang menggunakan *learning rate* 0.01 [6]. Selain itu dimensi citra yang kecil dapat mempercepat proses komputasi *model* dalam mempelajari *dataset* saat proses *training*.

II. METODE PENELITIAN

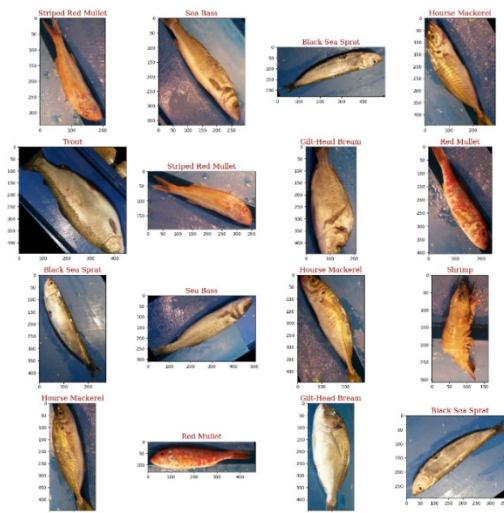
Untuk mempermudah dalam mengetahui alur dari penelitian ini, maka tahapan-tahapan yang akan dilakukan dijabarkan seperti pada gambar 1.



Gambar. 1. Alur Penelitian

A. Data Preparation

Penelitian ini menggunakan *dataset* yang berasal dari repositori website Kaggle yang berjudul *A Large Fish dataset* dengan jumlah total *dataset* adalah 9000 dari 9 kelas yaitu *Black Sea Spirt*, *Gilt-Head Bream*, *Hourse Mackerel*, *Red Mullet*, *Red Sea Bream*, *Sea Bass*, *Shrimp*, *Striped Red Mullet*, dan *Trout*. Untuk data yang diambil dalam penelitian ini hanya berjumlah 100 untuk masing-masing kelasnya sehingga total *dataset* yang digunakan pada penelitian ini sebanyak 900 gambar. Tampilan *dataset* citra dapat dilihat pada gambar 2.



Gambar. 2. Tampilan Dataset Ikan

Pada *dataset* yang digunakan sebanyak 100 gambar dengan 9 kelas akan terbagi menjadi 20% untuk *data testing* dan 80% untuk *data training*.

B. Data Preprocessing

Sebelum membangun model CNN untuk pelatihan, data terlebih dahulu dilakukan pengolahan dengan beberapa tahapan untuk mengurangi tingkat terjadinya *overfitting* atau *underfitting*, maka dilakukan beberapa proses sebagai berikut:

1. Cropping Image

Cropping image dilakukan dengan tujuan untuk mendapatkan lebih banyak objek ikan dibandingkan objek selain ikan. Proses ini cukup penting dilakukan karena jika tidak, komputer akan menganalisis semua objek yang terdapat pada citra. Hasil *cropping image* dapat dilihat pada gambar 3.



Gambar. 3. Hasil Proses Cropping

2. Splitting Dataset

Setelah citra selesai dilakukan *cropping*, maka citra dibagi menjadi data *testing* dan *training* yang bertujuan untuk memudahkan nanti saat proses *image augmentation*. Pembagian data menjadi *training* dan *testing* menggunakan bantuan fungsi dari *library* Keras yaitu *train_test_split*.

3. Image Augmentation

Saat membuat model menggunakan *machine learning*, teknik *image augmentation* dilakukan untuk memperbanyak data *training* dan meningkatkan ukuran dari dataset sebagai solusi dari terbatasnya data yang tersedia saat proses *training* tanpa harus mencari data yang baru untuk model *machine learning*. Proses ini dapat dilakukan dengan bantuan dari *library* dari Keras *ImageDataGenerator* dengan menambahkan parameter-parameter yang disesuaikan dengan kebutuhan citra. Berikut adalah parameter yang digunakan dalam penelitian ini pada tabel 1.

TABEL I
PARAMETER IMAGE AUGMENTATION

Augmentasi	Nilai
<i>Rescale</i>	1./255
<i>Rotation_range</i>	40
<i>Width_shift_range</i>	0.2
<i>Height_shift_range</i>	0.2
<i>Shear_range</i>	0.2
<i>Zoom_range</i>	0.2
<i>Horizontal_flip</i>	True
<i>Fill_mode</i>	nearest
<i>Target_size</i>	224x224
<i>Validation_split</i>	0.2

a. Rescale

Rescale berfungsi untuk mengubah nilai skala citra, proses ini termasuk ke dalam normalisasi citra dengan membagi piksel yang terkecil dengan piksel terbesar.

b. Random Rotation

Random_rotation memungkinkan model menjadi *invariant* terhadap orientasi dari objek secara acak memutar gambar dari 0 hingga 360° menggunakan nilai *integer* dengan argumen *rotation_range*. Saat gambar dalam proses rotasi, beberapa piksel akan pindah atau keluar dari gambar dan membuat gambar memiliki area yang kosong yang seharusnya terisi. Sehingga diperlukan argumen untuk mengisi area yang kosong tersebut, salah satunya adalah menggunakan argumen *fill_mode* dengan menggunakan nilai *default nearest* yang menggantikan area kosong tersebut dengan nilai piksel terdekat.

c. Random Shift

Saat melakukan *image augmentation*, besar kemungkinan bahwa objek tidak akan selalu berada di tengah gambar. Untuk mengatasi masalah tersebut, dapat dilakukan dengan mengubah piksel menjadi horizontal atau vertikal. Argumen yang digunakan adalah *height_shift_range* yang akan mengubah gambar menjadi vertikal dan *width_shift_range* yang akan mengubah gambar menjadi horizontal.

d. *Random Flips*

Pada *image augmentation*, *flipping images* juga dapat membantu model *machine learning* untuk mendapatkan hasil yang optimal tetapi penggunaan *flipping images* pada *image augmentation* disesuaikan dengan dataset yang digunakan. Argumen yang digunakan untuk *flipping images* adalah *horizontal_flip* atau *vertical_flip* dengan nilai *False* atau *True*.

e. *Random Zoom*

Zoom pada augmentasi gambar ini akan secara acak memperbesar atau memperkecil gambar. Parameter yang digunakan adalah *zoom_range* dengan menspesifikkan nilai apakah besar atau kecil, jika ingin menspesifikkan nilai dengan *float* maka, *zoom* gambar akan menghasilkan nilai antara $[1-zoom_range, 1+zoom_range]$.

f. *Target Size*

Argumen ini berfungsi untuk melakukan proses *resize* terhadap data citra untuk mengecilkan ukuran citra yang asli karena jika terlalu besar *model* akan terlalu berat dan sulit untuk melakukan proses *training*.

g. *Validation Split*

Di dalam proses augmentasi terjadi pembagian data untuk *validation* yang hasilnya akan digunakan saat proses evaluasi model. Data validasi berasal dari data *training* yang dibagi sebanyak 20%.

Selanjutnya merupakan penggunaan metode *flow_from_dataframe()* yang berfungsi untuk membaca nama dan target nilai dari *dataframe*. Pada penelitian ini, data *training* terbagi lagi sebanyak 20% yang digunakan untuk data *validation*, sehingga pembagian data *training*, data *testing*, dan data *validation*, dapat dilihat pada tabel 2.

TABEL II
PEMBAGIAN DATASET

No	Nama Dataset	%	Jumlah Data
1.	Training	60	576
2.	Validation	20	144
3.	Testing	20	180
Jumlah		100	900

C. Pelatihan Model

Tahap ini merupakan proses model dalam mempelajari *dataset* yang sudah melalui *preprocessing*. Penelitian ini menggunakan algoritma dari *deep learning* yaitu *Convolutional Neural Network* dan menggunakan arsitektur ResNet-50, alur dari arsitektur ResNet-50 dapat dilihat pada tabel 3.

TABEL III
ARSITEK RESTNET-50

Layer name	Output size	18-layer	34-layer	50-layer	101-layer	152-layer
Conv1	112x112			7x7, 64, stride 2		
Conv2_x	56x56			3x3 max-pool, stride 2		
		$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$
Conv3_x	28x28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
Conv4_x	14x14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
Conv5_x	7x7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
FLOPs	1x1	1.8x10 ⁹	3.6x10 ⁹	Average pool, 1000-d fc, softmax 3.8x10 ⁹	7.6x10 ⁹	11.3x10 ⁹

Pada dasarnya *Convolutional Neural Network* terbagi menjadi 2 bagian besar, yaitu *Feature Extraction Layer* dan *Classification Layer*. Lapisan *Feature Extraction* terjadi perubahan *image* menjadi sebuah *features (encoding)* yang berupa angka untuk mempresentasikan citra tersebut. Di dalam lapisan ini terdapat 3 proses yaitu proses konvolusi yang terjadi di *convolutional layer*. Proses konvolusi menjadi salah satu tahapan yang paling penting dikarenakan pada tahapan ini akan mengesktrak fitur dari citra input dengan

memanfaatkan struktur yang terdiri dari *neuron* sehingga membentuk sebuah *filter*. Persamaan dari operasi konvolusi dapat dilihat sebagai berikut

$$h(x) = f(x) * g(x)$$

Sedangkan untuk mengukur dimensi *output* dari proses konvolusi dapat menggunakan persamaan berikut

$$n_{out} = \left(\frac{N - F + 2P}{S} \right) + 1$$

Pooling layer merupakan lapisan yang berfungsi untuk mengurangi ukuran dari parameter *feature map* yang dihasilkan dari proses konvolusi menggunakan teknik *down-sampling* sehingga mampu mengendalikan model agar tidak terjadi *overfitting*. Terdapat metode *pooling* yang sering digunakan yaitu *average pooling* dan *max-pooling*, *average pooling* akan mengambil nilai rata-rata dari *kernel* sedangkan *max-pooling* akan memilih nilai tertinggi.[7]

Activation Function Rectified Linear Unit (ReLU) adalah salah satu jenis fungsi aktivasi pada *deep learning* yang mampu menyelesaikan masalah *non-linear*, ReLU akan mengaplikasikan fungsi $f(x) = \max(0, x)$ yang akan mengembalikan nilai 0 apabila menerima negatif dan akan mengembalikan nilai x jika menerima positif.[8]

Pada *classification layer* terjadi beberapa proses diantaranya ada proses *flatten* yang berfungsi untuk mengubah nilai input matriks yang terdapat dari hasil *convolutional layer* menjadi vektor berbentuk satu dimensi, proses ini perlu dilakukan dikarenakan jika tidak akan menyebabkan kehilangan informasi dan data menjadi tidak *reversible*.

Fully Connected Layer, lapisan ini akan menghubungkan semua neuron dari aktivitas sebelumnya ke neuron di lapisan selanjutnya. *Fully connected layer* terdiri dari beberapa *hidden layer*, *action function*, *output layer*, dan *loss function*. Input yang digunakan berasal dari *feature map* yang telah melalui proses *flatten* yang berbentuk satu dimensi.

Softmax memiliki fungsi untuk menghasilkan klasifikasi dengan menghitung probabilitas pada setiap kelas dengan distribusi dasar dan nilainya akan diubah menjadi vektor dengan rentang nilai antara 0 dan 1 yang apabila dijumlahkan akan bernilai 1.[9] Persamaan dari *softmax* dapat dilihat pada persamaan berikut.

$$f_j(z) = \frac{e^{z_j}}{\sum_k e^{z_k}}$$

D. Pengujian Model

Testing model diperlukan untuk mendapatkan data gambar ketika data selesai diklasifikasikan. Dalam menguji data latih dapat menggunakan fungsi *fitmodel* yang berisi *epoch*, *callbacks*, *batch size*. *Epoch* adalah satu kali iterasi dari *dataset training* yang digunakan untuk melatih *neural network*, sedangkan *batch size* merupakan jumlah sampel data yang digunakan untuk melatih *neural network*.

E. Evaluasi Model

Penelitian ini menggunakan beberapa parameter untuk mengetahui performa dari *model* yang telah dihasilkan, Adapun parameter skenario tersebut adalah:

- 1) Pengujian terkait adanya pengaruh *preprocessing* sebelum membangun *model* klasifikasi terhadap hasil tingkat akurasi dan *loss*.
- 2) Pengujian dari jumlah *epoch* yang diusulkan, pengujian ini dilakukan untuk mengetahui seberapa tingkat akurasi dan *loss* yang dihasilkan model pada proses *training* dan *validation* menggunakan *epoch* 20, 40, 60, 80, 100 dan besarnya *learning rate* yang digunakan. Pada penelitian ini *learning rate* yang digunakan berjumlah 2 yaitu 0.001 dan 0.0001 untuk mengetahui performa yang dihasilkan menggunakan nilai akurasi dan *loss*.

Uji coba akan dilakukan berulang hingga sistem menghasilkan hasil akurasi dan *loss* yang baik berdasarkan nilai *learning rate* dan *epoch*, selain itu juga memperhatikan hasil dari *precision*, *recall*, dan *f1-score* sehingga model *Convolutional Neural Network* mampu mengidentifikasi jenis ikan dengan optimal.

III. HASIL DAN PEMBAHASAN PENELITIAN

Bagian ini menjelaskan tentang percobaan yang telah dilakukan. percobaan dilakukan dengan membandingkan hasil dari data training dan data validation menggunakan arsitektur ResNet-50. Dalam penelitian ini dilakukan beberapa skenario untuk menguji model CNN dengan pengaruh *preprocessing*, pengaruh jumlah *epoch*, dan besarnya nilai *learning rate*.

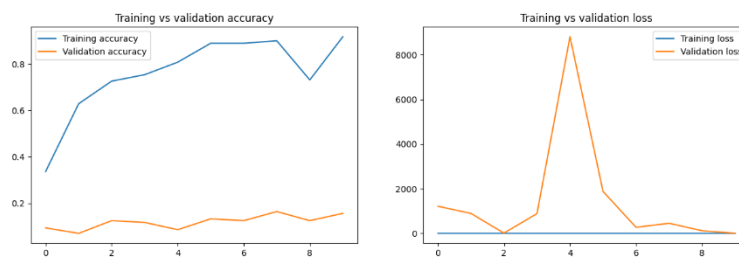
A. Pengujian Pengaruh *Preprocessing*

Tahap ini dilakukan pengujian terhadap skenario pertama yaitu pengaruh adanya *preprocessing* yang dilakukan sebelum proses pelatihan *model* dijalankan. *Preprocessing* yang dilakukan melalui tiga tahapan yaitu *cropping*, *splitting dataset*, dan *image augmentation*. Pengujian ini dilakukan untuk mengetahui apakah model mengalami *overfitting* atau tidak. Tabel 4 merupakan hasil pengujian dari data yang tidak dilakukan *preprocessing* terhadap nilai akurasi dan *loss* pada *training* dan *validation*.

TABEL IV
HASIL PENGUJIAN TANPA PREPROCESSING

Learning Rate	Epoch	Training		Validation	
		Accuracy	Loss	Accuracy	Loss
0.001	20	91.67%	0.2865	15.62%	4.4488

Meskipun hasil akurasi pada saat *training* mendapatkan nilai yang tinggi, namun dalam grafik prosesnya. Model tersebut mengalami *underfitting* sehingga hasil dari model pelatihan tersebut tidak bisa digunakan. Dikarenakan dalam proses *training*, komputer menganalisis semua objek yang terdapat pada citra dan ukuran citra yang terlalu besar. Maka hasil dari grafik dengan pengujian data citra tanpa melalui proses *preprocessing* dapat dilihat pada gambar 5.



Gambar. 4. Hasil Akurasi & Loss Training dan Validation Data Tanpa Preprocessing

Sedangkan model yang menggunakan proses uji *preprocessing* mendapatkan nilai yang baik tanpa mengalami *underfitting* pada saat proses pelatihan dilakukan. Tabel 4 merupakan hasil pengujian *preprocessing* terhadap hasil akurasi dan *loss* pada saat proses *training* dan *validation*.

B. Pengaruh Jumlah *Epoch* dan Besarnya Nilai *Learning Rate*

Uji percobaan dengan skenario kedua yaitu pengaruh dari jumlah *epoch* dan besarnya nilai *learning rate* saat proses *training*, menggunakan data *training* dan data *validation*. Penentuan dari jumlah *epoch* bertujuan untuk memberikan model CNN performa yang baik sedangkan untuk *learning rate* bertujuan untuk mempengaruhi proses model CNN saat *training* apakah akan berjalan cepat atau lambat.

TABEL V
HASIL UJI TRAINING DAN VALIDATION

Percobaan	Learning Rate	Epoch	Training		Validation	
			Accuracy	Loss	Accuracy	Loss
1	0.001	20	94.44%	0.28107	90.97%	0.38831
2	0.001	40	97.22%	0.10607	93.06%	0.26812
3	0.001	60	96.53%	0.21097	96.53%	0.13994
4	0.001	80	96.18%	0.22767	95.83%	0.25362
5	0.001	100	97.92%	0.069413	97.92%	0.065838
6	0.0001	20	93.75%	0.16189	96.53%	0.17176
7	0.0001	40	95.31%	0.11324	93.06%	0.21366

8	0.0001	60	96.70%	0.091653	96.53%	0.086277
9	0.0001	80	96.74%	0.07301	97.92%	0.066742
10	0.0001	100	97.22%	0.092081	96.53%	0.08223

Percobaan ini dilakukan untuk menentukan jumlah *epoch* dan nilai *learning rate* yang optimal untuk memberikan performa yang terbaik untuk model CNN. Penelitian ini menggunakan beberapa nilai *epoch*, yaitu 20, 40, 60, 80, dan 100, selain itu menggunakan *learning rate* sebesar 0.001 dan 0.0001.

Berdasarkan tabel 5 nilai akurasi tertinggi dihasilkan oleh *learning rate* 0.001 dengan jumlah *epoch* yang sebesar 100 sedangkan untuk nilai akurasi terendah didapatkan dari *epoch* 20 dengan *learning rate* 0.0001 dan untuk nilai akurasi terendah dihasilkan oleh *epoch* 20 dengan nilai akurasi 94.44%. Sedangkan untuk *learning rate* 0.0001 hasil akurasi tertinggi dengan Jumlah *epoch* 100 menghasilkan nilai akurasi 97.92% dan untuk *epoch* 20 menghasilkan tingkat akurasi yang rendah dengan nilai sebesar 93.75% dengan *learning rate* 0.0001.

Dapat dilihat bahwa nilai *epoch* dan *learning rate* akan mempengaruhi hasil akurasi hal ini dikarenakan semakin besar jumlah *epoch* maka semakin tinggi pula hasil akurasi yang dihasilkan, sedangkan besar atau kecilnya *learning rate* yang digunakan akan mempengaruhi waktu komputasi saat proses *training*. Maka dari itu, dapat diambil kesimpulan bahwa *epoch* 100 dengan *learning rate* 0.001 merupakan hasil yang optimal untuk membuat model klasifikasi jenis ikan air laut menggunakan arsitektur ResNet-50.

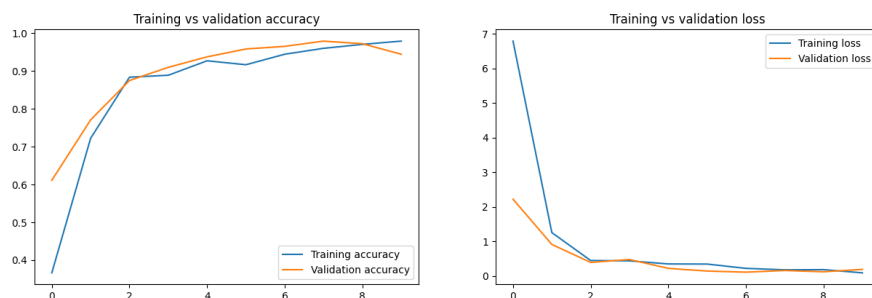
C. Pengujian Mode

Hasil dari proses *training* menunjukkan bahwa model menghasilkan nilai terbaik pada percobaan ke-5 dengan *epoch* 60 dan menggunakan 900 data uji citra sehingga untuk menguji skenario yang telah dibuat maka menggunakan hasil dari percobaan ke-5.

TABEL VI
HASIL UJI TESTING

Percobaan	Learning Rate	Epoch	Testing	
			Accuracy	Loss
5	0.001	100	97.78%	0.095742

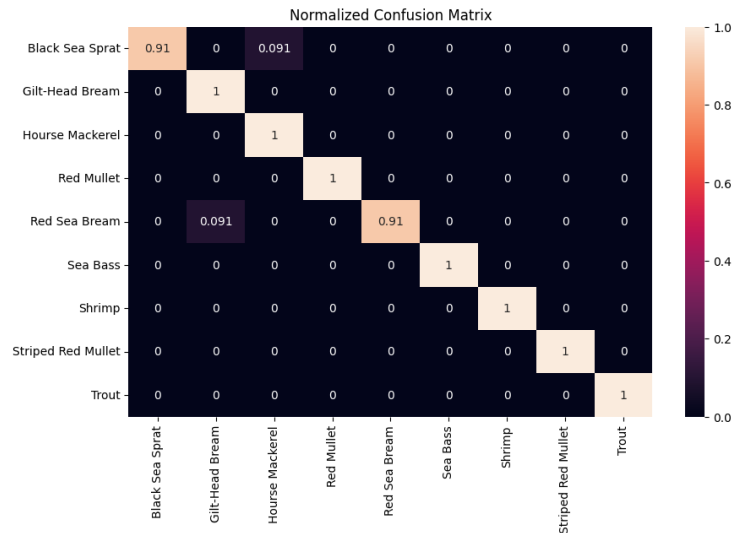
Tingkat akurasi dan loss yang dihasilkan pada saat proses *testing* menghasilkan nilai yang dapat dilihat pada tabel 6. Nilai *learning rate* 0.001 memberikan nilai *loss* lebih kecil dibandingkan dengan nilai akurasi. Artinya bahwa nilai *loss* yang dihasilkan pada *learning rate* 0.001 mengalami penurunan pada saat *training* sehingga membuat nilai akurasi yang didapatkan semakin meningkat. Grafik perhitungan *epoch* dapat dilihat pada gambar 6



Gambar. 5. Hasil Akurasi & Loss Training dan Validation

D. Evaluasi Model

Untuk mengetahui informasi lebih detail tentang performa *Convolutional Neural Network* menggunakan arsitektur ResNet-50, maka model CNN dievaluasi menggunakan metode *Confusion Matrix* untuk mengetahui nilai dari *precision*, *recall*, dan *f1-score*. Hasil tersebut dapat dilihat pada gambar 7.



Gambar. 7. Confusion Matrix

Didapatkan rata-rata dari *dataset* sebanyak 900 dengan 9 kelas dengan dimensi citra adalah 224x224 dengan hasil dari nilai *accuracy* yang didapatkan pada epoch 100 dengan learning rate 0.001 menghasilkan nilai *accuracy* sebesar 98%, kemudian untuk nilai *precision* sebesar 98%, untuk nilai *recall* sebesar 98% dan yang terakhir nilai *f1-score* sebesar 97%. Sehingga berdasarkan nilai *precision*, *recall*, dan *f1-score* tersebut, penentuan jumlah *epoch* akan mempengaruhi *model* dalam mempelajari *dataset* seperti yang dapat dilihat pada tabel 7. Selain itu nilai dari *learning rate* akan mempengaruhi waktu komputasi yang diperlukan saat *model* dalam proses *training*, dalam kasus ini digunakan 2 macam *learning rate*, 0.001 & 0.0001. Dengan hasil yang didapatkan dari kedua *learning rate* tersebut, dapat disimpulkan bahwa semakin besar *learning rate* maka akan semakin cepat pula sistem dalam mempelajari *dataset* dan dapat menghasilkan akurasi yang tinggi. Dengan demikian penggunaan *learning rate* bernilai 0.001 memiliki keunggulan dengan proses komputasi yang lebih cepat dibandingkan dengan 0.0001. Apabila dibandingkan dengan penelitian sebelumnya yang menggunakan jumlah *epoch* yang relatif kecil, *dataset* berjumlah ribuan, dan dimensi citra yang besar, penelitian ini dengan menggunakan arsitektur ResNet-50 tetap mampu menghasilkan nilai akurasi yang tinggi.

TABEL VII
 HASIL PRECISION, RECALL & F1-SCORE

	Precision	Recall	F1-score
Black Sea Spart	1.00	0.91	0.95
Gilt-Head Bream	0.87	1.00	0.93
Hourse Mackerel	0.95	1.00	0.98
Red Mullet	1.00	1.00	1.00
Red Sea Bream	1.00	0.91	0.95
Sea Bass	1.00	1.00	1.00
Shrimp	1.00	1.00	1.00
Striped Red Mullet	1.00	1.00	1.00
Trout	1.00	1.00	1.00

III. KESIMPULAN

Berdasarkan hasil penelitian diatas dapat disimpulkan bahwa penggunaan metode *Convolutional Neural Network* menggunakan arsitektur ResNet-50 yang dipengaruhi oleh besarnya *epoch* dan nilai *learning rate* sangat mempengaruhi performa *model* dalam mengklasifikasikan objek pada saat *training*. Performa model yang terbaik dalam menghasilkan nilai *accuracy* dan *loss* terjadi pada percobaan ke-5 dengan *epoch* 100 dan *learning rate* 0.001, dengan seiring meningkatnya jumlah *epoch* dan nilai dari *learning rate*, *model* akan mampu

mempelajari karakteristik dan mempercepat proses *training dataset*. Untuk penelitian lebih lanjut disarankan untuk meningkatkan *epoch* latih yang berguna untuk mengetahui tingkat fluktuasi sistem dan menerapkan *image augmentation* pada saat *preprocessing* untuk menghindari terjadinya *underfitting* atau *overfitting*. Selain itu, dapat ditambahkan lebih lagi dataset untuk jenis ikan lainnya.

DAFTAR PUSTAKA

- [1] Direktorat Jenderal Penguatan Daya Saing Produk Kelautan dan Perikanan.(2022).*Ekspor Perikanan Tumbuh 10.66% di 2022*. Diakses pada 2023, dari <https://kkp.go.id/djpdspkp/artikel/47840-ekspor-perikanan-tumbuh-10-66-di-2022>.
- [2] PUSHIDROSAL.(2018).*Data Kelautan yang Menjadi Rujukan Nasional Diluncurkan*.Diakses pada 2023,dari <https://www.pushidrosal.id/berita/5256/DATA-KELAUTAN-YANG-MENJADI-RUJUKAN-NASIONAL--DILUNCURKAN/#:~:text=Luas%20perairan%20Indonesia%206.400.000,Indonesia%2C%20sesuai%20dengan%20UU%20no>
- [3] Indonesia.go.id.(2023).*Pengusahaan Sektor Perikanan Hanya di Wilayah Terukur*.Diakses pada 2023, dari <https://indonesia.go.id/kategori/editorial/6950/pengusahaan-sektor-perikanan-hanya-di-wilayah-terukur?lang=1>.
- [4] Arif Agusetyawan.(2020).*Pengolahan Citra untuk Membedakan Ikan Segar dan Ikan Tidak Segar Menggunakan Convolutional Neural Network*.5(1),hal. 11-19.
- [5] Bana Falakhi, Elmira F.A.,Muhamad Rizaldi, Renata Rizki R.A. & Novanto Yudistira.(2022).*Perbandingan Model AlexNet dan ResNet dalam Klasifikasi Citra Memanfaatkan Transfer Learning* .9(1), hal. 70-78.
- [6] Novelita Dwi Wiranda, Ledy Novamizanti, & Syamsul Rizal.(2020).*Convolutional Neural Network Pada Klasifikasi Sidik Jari Menggunakan Resnet-50*.1(2),hal. 61-68.
- [7] Pulung, A.N,Indah Fenriana, & Rudy Arijanto,Mkom.(2020).*Implementasi Deep Learning Menggunakan convolutional neural network (CNN) pada Ekspresi Manusia*.2(1),hal. 12-21.
- [8] Sarirotul Ilahiyah, Agung Nilogiri.(2018). *Implementasi Deep Learning pada Identifikasi Jenis Tumbuhan Berdasarkan Citra Daun Menggunakan Convolutional Neural Network*.3(2),hal. 49-56
- [9] Femil Paraijun, Rosida N.A, Dwina K.(2022). *Implementasi Algoritma Convolutional Neural Network Dalam Mengklasifikasikan Kesegaran Buah Berdasarkan Citra Buah*.11(1),hal. 1-9.
- [10] Dahria, Muhammad (Agustus 2008). *Kecerdasan Buatan (Artificial Intelligence)*.5(2),hal 185-196.
- [11] P.F. Ambara Dewi, I.G.A. Ari Widarti, dan D.P. Sukraniti.(Februari 2018). *Pengetahuan Ibu Tentang Ikan dan Pola Konsumsi Ikan pada Balita di Desa Kedonganan Kabupaten Badung*.7(1), hal. 16-20. Tersedia: <https://ejournal.poltekkes-denpasar.ac.id/index.php/JIG/article/view/jig7104>
- [12] E.B. Sinaga, dan M.F. Astuti Tanjung (Agustus 2022). *Pemanfaatan Zat Besi Dalam Makanan Laut (Ikan Laut) Pada Ibu Hamil untuk Mencegah Stunting Di Kelurahan Pulau Simardan Tanjung Balai*.4(2),hal. 26-29.
- [13] .(September 2020).*Manfaat Asupan Gizi Ikan Laut Untuk Mencegah Penyakit dan Menjaga Kesehatan Tubuh Bagi Masyarakat Pesisir*.1(2),hal. 92-95.
- [14] A.O.Puspita Dewi.(2020).*Kecerdasan Buatan sebagai Konsep Baru pada Perpustakaan*.4(4),hal. 453-461.
- [15] Betta Mahesh.(Januari 2020).*Machine Learning Algorithms – A Review*.9(1),hal. 381-386.
- [16] Ahmad Roihan, P.A. Sunarya, dan A.S. Rafika.(April 2020).*Pemanfaatan Machine Learning dalam Berbagai Bidang: Review paper*.5(1),hal. 75-82.
- [17] Qifang Bi, K.E. Goodman, Joshua Kaminsky, and Justin Lessler.(October 2019).*What is Machine Learning? A Primer for the Epidemiologist*.188(12),hal. 2222-2239.
- [18] Bell, Jason.(2015)*Machine Learning: Hands-on for Developers and Technical Professionals*.Canada, hal. 1-4.
- [19] .Wantimpres.go.id.(2017).*Potensi Perikanan Indonesia*.Diakses pada 2023, pada <https://wantimpres.go.id/id/2017/04/potensi-perikanan-indonesia/>.
- [20] Qayyum, Rafay.(2022).*Introduction to Pooling Layers in CNN*. Diakses pada 2023,dari <https://pub.towardsai.net/introduction-to-pooling-layers-in-cnn-dafe61eabe34>.
- [21] Dharmadi. Richard.(2018).*Mengenal Convolutional Layer dan Pooling Layer*.Diakses pada 2023, dari <https://medium.com/nodeflux/mengenal-convolutional-layer-dan-pooling-layer-3c6f5c393ab2>
- [22] *Rectified Linear Units (ReLU) in Deep Learning*.Diakses pada 2023,dari <https://www.kaggle.com/code/dansbecker/rectified-linear-units-relu-in-deep-learning/notebook>.
- [23] *Convolutional Neural Networks*.Diakses pada 2023,dari <https://cs231n.github.io/convolutional-networks/#pool>.
- [24] MB Herlambang.(2019).*Deep Learning: Convolutional Neural Networks*.Diakses pada 2023, dari <https://www.megabagus.id/deep-learning-convolutional-neural-networks/6/>.
- [25] Syihab Irawan, Fiki.(2020).*Convolutional Neural Network (CNN)*.Diakses pada 2023, dari [https://kotakode.com/blogs/2707/Convolutional-Neural-Network-\(CNN\)](https://kotakode.com/blogs/2707/Convolutional-Neural-Network-(CNN)).
- [26] Bhandari, aniruddha.(2020).*Image Augmentation on The Fly using Keras ImageDataGenerator*.Diakses pada 2023,dari <https://www.analyticsvidhya.com/blog/2020/08/image-augmentation-on-the-fly-using-keras-imagedatagenerator/>.
- [27] Bhattacharyya, Saptashwa.(2020).*Understand and Implement ResNet-50 with TensorFlow 2.0*.Diakses pada 2023, dari <https://towardsdatascience.com/understand-and-implement-resnet-50-with-tensorflow-2-0-1190b9b52691>.
- [28] Asyrofi, Rakha.(2020).*Neural Network dalam Tensorflow*.Diakses pada 2023, dari <https://asyrofist.medium.com/perspektif-pemangku-kepentingan-556ba82b914e>.
- [29] Duong, Andre.(2019).*Keras Callbacks Explained in Three Minutes “A gentle introduction to callbacks in Keras. Learn about EarlyStopping, ModelCheckpoint, and other callback functions with code examples.”*.Diakses pada 2023, dari <https://www.kdnuggets.com/2019/08/keras-callbacks-explained-three-minutes.html>.