



PENERAPAN ALGORITMA CONVOLUTIONAL NEURAL NETWORK(CNN) DETEKSI KESEHATAN MATA

Rizki Eka Mulyani¹⁾, Muhammad Rafli Erfiyanto²⁾, Fathur Rizqi Putra Pratama³⁾, Tengku Dimas Aditya⁴⁾, Muhammad Fikri Hidayattullah⁵⁾

¹²³⁴⁵ Teknik Informatika, Politeknik Harapan Bersama

email: ¹rizkiekamulyani123@gmail.com, ²raflieriyanto810@gmail.com, ³ezzdarkrap2@gmail.com,
⁴tengkudimasadt@gmail.com, ⁵fikri@poltekegal.ac.id

ARTICLE INFO

Article History:

Received : 13 Juni 2024

Accepted : 19 November 2024

Published : 19 Desember 2024

Keywords:

Eye Health,
Vision,
Convolutional Neural
Network (CNN),
Eye Diseases,
Disease Detection

IEEE style in citing this article:

R. E. Mulyani, M. R. Erfiyanto, F. Rizqi, T. D. Aditya, M. F. Hidayattullah, "Penerapan Algoritma Convolutional Neural Network(CNN) Deteksi Kesehatan Mata", Jurnal ilmiah.informatika, vol. 9, no. 2, pp. 142-xx, Des. 2024.

ABSTRACT

Human senses play a crucial role in perceiving and interacting with the environment. One of the most vital senses is vision, facilitated by the eyes, which provide up to 80% of the information needed for daily activities. Despite its importance, eye health is often neglected, and eye disorders can lead to discomfort, visual impairment, or even total blindness. Diseases such as cataracts, glaucoma, diabetic retinopathy, and other common eye conditions pose significant threats to visual health. Indonesia, with one of the highest blindness rates in the world, faces challenges in providing adequate eye care, especially in remote areas with limited access to ophthalmologists. To address this issue, machine learning techniques, particularly Convolutional Neural Networks (CNN), have been utilized for image classification tasks, including medical image analysis. CNN models, especially the VGG-19 architecture, have proven effective in classifying and diagnosing eye diseases based on retinal images with an accuracy rate of 96%. This study aims to implement a CNN method using the VGG-19 architecture to classify various eye diseases, such as cataracts, glaucoma, diabetic retinopathy, and other eye conditions. The study involves the use of a high-quality dataset consisting of 800 labeled images across eight categories of eye diseases, with the goal of achieving accurate classification. The dataset is preprocessed, and the deep learning model is trained to improve classification outcomes. This research contributes to the potential application of CNN in assisting the diagnosis of eye diseases.

1. PENDAHULUAN

Indra merupakan alat penting dalam tubuh manusia yang berfungsi untuk mengenali atau merasakan sesuatu dari lingkungan sekitarnya. Informasi yang diterima oleh indra kemudian diproses secara otomatis oleh tubuh, sehingga manusia dapat memperoleh dan mengolah informasi tersebut untuk mendukung aktivitas sehari-hari. Pada umumnya, manusia memiliki lima jenis indra yang dikenal sebagai panca indra. Salah satu indra utama tersebut adalah indra penglihatan atau mata. Mata (*oculus*) adalah salah satu organ vital terpenting dalam tubuh[1]. Mata adalah salah satu dari lima indra manusia [1]. Mata bekerja dengan cara menerima gambar visual, yang diteruskan ke otak untuk diolah menjadi informasi yang dapat dipahami. Fungsi penglihatan ini menjadikan mata sebagai salah satu indra yang paling sering digunakan dalam berbagai aktivitas, seperti bekerja, bersekolah, dan aktivitas sehari-hari lainnya.

Untuk menjaga kesehatan mata, penting untuk mengonsumsi makanan bergizi seimbang, rutin berolahraga, dan mengurangi paparan sinar *ultraviolet* dari perangkat elektronik. Langkah-langkah ini dapat membantu mencegah atau meminimalkan efek buruk akibat aktivitas mata yang berlebihan.

Efek buruk tersebut dapat memicu gangguan kesehatan mata, seperti katarak, glaukoma, dan penyakit retina. Jika tidak segera ditangani, gangguan mata ini berpotensi menyebabkan kebutaan. Berdasarkan data dari Kementerian Kesehatan Republik Indonesia, Indonesia menempati posisi kedua sebagai negara dengan angka kebutaan tertinggi di dunia setelah Etiopia, dengan tingkat prevalensi kebutaan di atas 1% [2]. Berdasarkan laporan dari Vision Atlas pada tahun

2020, terdapat sebanyak 1,1 miliar orang yang kehilangan penglihatan di seluruh dunia [3]. Dari jumlah tersebut, sebanyak 90 juta orang di antaranya adalah anak-anak dan remaja, di mana 2,1 juta orang di antaranya mengalami kebutaan [3]. Jumlah ini diproyeksikan akan terus bertambah hingga mencapai 1,76 miliar orang di seluruh dunia yang mengalami kehilangan penglihatan pada tahun 2050 [3].

Penyebab utama kebutaan untuk anak-anak dan remaja adalah penyakit mata [4]. Diperkirakan bahwa sebanyak 40% dari kebutaan yang terjadi pada anak-anak dan remaja akibat penyakit mata dapat dicegah apabila penyakit tersebut dapat dideteksi dan ditangani lebih dini [4]. Hal ini memperparah kondisi kesehatan mata masyarakat, terutama bagi kelompok usia lanjut yang rentan terhadap penyakit mata seperti katarak.

Ada berbagai metode yang dapat digunakan dalam pengolahan citra, salah satunya adalah Convolutional Neural Network (CNN). CNN merupakan pengembangan dari Multi-Layer Perceptron (MLP) yang dirancang khusus untuk menangani data berdimensi dua. Metode ini termasuk dalam kategori deep learning karena memiliki arsitektur jaringan yang mendalam dan sering diterapkan pada pengolahan data citra. Sementara itu, Multi-Layer Perceptron tidak ideal untuk klasifikasi citra karena tidak mampu mempertahankan informasi spasial dari data citra dan memperlakukan setiap piksel sebagai fitur yang berdiri sendiri, sehingga menghasilkan akurasi yang kurang optimal. Hal ini disebabkan oleh fakta bahwa CNN berusaha meniru sistem pengenalan gambar pada visual cortex manusia, yang memberinya kemampuan untuk mengolah gambar [5].

Penerapan metode Convolutional Neural Network (CNN) untuk mengklasifikasikan suatu penyakit telah diterapkan pada penelitian-penelitian terdahulu. Penerapan metode CNN untuk mengklasifikasi 4 jenis penyakit mata dengan menggunakan citra fundus mata mendapatkan akurasi *training* sebesar 98,37% [6], penerapan metode CNN untuk mengklasifikasi penyakit mata katarak dengan menggunakan citra fundus mata mendapatkan akurasi sebesar 91,41% [7], dan penerapan metode CNN untuk mengklasifikasi penyakit mata *diabetic retinopathy* dengan menggunakan citra fundus mata mendapatkan akurasi sebesar 90,10% [8], penerapan metode CNN untuk mendeteksi penyakit mata *diabetic retinopathy* dengan menggunakan citra fundus mata mendapatkan akurasi sebesar 68%.

Salah satu arsitektur CNN yang terkenal adalah VGG-19. Penggunaan Convolutional Neural Network dengan model VGG19 adalah pilihan yang baik untuk tugas klasifikasi citra semacam ini, karena CNN dikenal efektif dalam mengekstraksi fitur-fitur visual dari citra dan VGG19 adalah salah satu arsitektur yang terbukti berhasil dalam berbagai tugas pengolahan citra. Satu hal yang menonjol dari arsitektur VGG adalah penggunaan kernel konvolusi dengan ukuran yang relatif kecil, yaitu 3x3, dengan stride 1. Pendekatan ini memungkinkan model untuk mengekstrak fitur-fitur yang lebih kompleks melalui rangkaian konvolusi yang dalam. Namun, perlu diperhatikan bahwa kedalaman yang lebih besar juga dapat memerlukan lebih banyak komputasi, memori, dan waktu pelatihan. Ketika merancang atau memilih arsitektur CNN untuk tugas-tugas pengolahan citra, informasi tentang arsitektur seperti VGG sangat berharga. [9]

Keunggulan VGG-19 terletak pada kedalamannya, yang memungkinkan model untuk mempelajari fitur-fitur visual secara mendetail. Penelitian oleh Wahyudi et al. (2023) menunjukkan bahwa penerapan VGG-19 untuk mendeteksi penyakit mata seperti retinopati diabetik berhasil mencapai tingkat akurasi hingga 99%. [10] Hal ini menunjukkan potensi besar dari penggunaan VGG-19 dalam membantu proses diagnosa penyakit mata.

Berlandaskan pada penjelasan penelitian sebelumnya, disimpulkan bahwa metode Convolutional Neural Network (CNN) dan arsitektur VGG-19 mampu digunakan secara efektif untuk mengklasifikasikan suatu objek. Oleh karena itu, penelitian ini akan memanfaatkan metode CNN dengan arsitektur VGG-19 untuk mengklasifikasikan penyakit mata pada manusia.

Berdasarkan penjelasan tersebut, permasalahan yang diangkat dalam penelitian ini adalah bagaimana cara menerapkan metode CNN dengan arsitektur VGG-19 untuk mengklasifikasikan berbagai jenis penyakit mata. Tujuan dari penelitian ini adalah untuk mengimplementasikan metode CNN dengan arsitektur VGG-19 dalam mengidentifikasi penyakit mata pada manusia serta mengevaluasi tingkat akurasi yang dihasilkan dari penelitian ini.

2. METODE PENELITIAN

2.1 Dataset

Penelitian ini mengadopsi pendekatan kuantitatif dengan analisis data yang didasarkan pada dataset yang diperoleh dari dua sumber utama. Pertama, dataset yang digunakan dalam penelitian ini diperoleh dari platform *Kaggle*, yang

merupakan sumber data terbuka yang memiliki berbagai macam dataset yang relevan dengan topik penelitian ini. Dataset yang dipilih memiliki kualitas data yang tinggi, lengkap, dan dapat diakses secara bebas untuk keperluan penelitian ilmiah. Selain itu, untuk melengkapi dataset yang ada, pencarian dataset tambahan dilakukan melalui pencarian sumber-sumber terbuka di internet, terutama menggunakan mesin pencari Google. Proses pencarian ini bertujuan untuk menemukan dataset yang relevan dan sesuai dengan topik penelitian, serta memastikan data yang diperoleh mencakup berbagai kondisi yang diperlukan untuk analisis.

Dataset tersebut terdiri dari 800 gambar dengan delapan label yakni mata katarak, mata *glaucoma*, mata *diabetic retinopathy*, mata merah, mata iritasi, mata belekan, mata normal dan bukan mata. Dimana setiap label terdiri dari 100 file gambar. Dari setiap label tersebut kemudian dibagi lagi menjadi dua *path* atau folder yakni folder pelatihan dan pengujian.

2.2 Preprocessing Data

Tahap ini merupakan proses pemrosesan data yang dilakukan sebelum data digunakan dalam pelatihan dan evaluasi. Langkah pertama dalam pra-pemrosesan data adalah membagi data yang telah dikumpulkan menjadi dua bagian, yaitu data latih (untuk pelatihan) dan data validasi (untuk evaluasi). Setelah itu, dilakukan *resize* pada gambar agar dimensi setiap gambar konsisten, yang bertujuan untuk mempercepat proses pelatihan model. Langkah berikutnya adalah melakukan augmentasi data untuk memperbanyak jumlah data

pelatihan, yang diharapkan dapat meningkatkan akurasi model.

2.3 Proses Latih Data

Dalam pelatihan data dibagi menjadi dua bagian, sehingga terdiri dari 80% data latih dan 20% untuk data uji. Data uji diperlukan untuk mengkonfirmasi keakuratan model yang dibuat. Untuk membandingkan dan menemukan nilai akurasi terbaik, diterapkan beberapa skenario [11]. Proses ini dilakukan agar hasil output yang didapatkan dalam data testing nanti dapat menghasilkan output yang diharapkan yakni lebih dari 50% akurasi klasifikasi. Dalam proses pelatihan data citra akan dilakukan iterasi sebanyak 50 *epoch* dan 32 *batch_size*.

2.4 Convolutional Neural Network (CNN)

Jaringan saraf *convolutional* (CNN) adalah salah satu varian dari jaringan saraf tiruan dengan bobot dan beberapa lapisan tersembunyi yang disusun sebagai arsitektur. Perancangan CNN merupakan langkah dalam mengembangkan model yang digunakan untuk melatih data untuk mengidentifikasi objek yang diinginkan. Model yang dikembangkan terdiri dari jumlah layer yang digunakan, spesifikasi filter, spesifikasi ukuran kernel dan spesifikasi fungsi aktivasi, serta spesifikasi pool [10]. Proses dimulai dengan pemilihan arsitektur dasar CNN yang telah dilatih sebelumnya, yaitu VGG16, yang digunakan sebagai *feature extractor*.

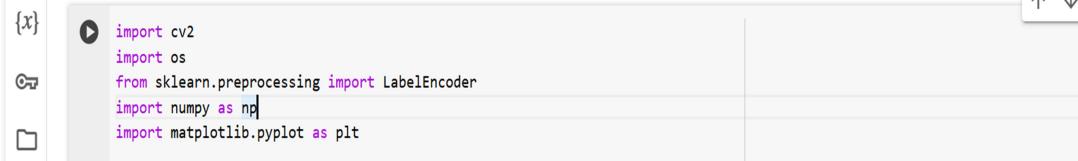
Pada tahap selanjutnya, model diadaptasi dengan menambahkan lapisan-lapisan tambahan untuk meningkatkan kemampuan klasifikasi.

Lapisan pertama adalah *Global Average Pooling* yang bertujuan untuk mereduksi dimensi data keluaran dari lapisan konvolusional menjadi vektor satu dimensi, sehingga dapat digunakan untuk klasifikasi lebih lanjut. Kemudian, dua lapisan *dense* dengan aktivasi ReLU ditambahkan untuk menangkap hubungan non-linear yang lebih kompleks antara fitur-fitur yang telah dipelajari. Lapisan terakhir adalah lapisan *dense* dengan aktivasi softmax yang digunakan untuk menghasilkan probabilitas untuk setiap kelas output, yang disesuaikan dengan jumlah kelas dalam dataset. Model ini dilatih menggunakan data gambar yang telah diproses dengan teknik augmentasi untuk meningkatkan variasi data dan mengurangi risiko overfitting. Selama pelatihan, perhitungan bobot kelas juga dilakukan untuk menangani ketidakseimbangan kelas, memastikan model dapat memberikan perhatian yang lebih besar pada kelas yang jarang muncul.

3. HASIL DAN PEMBAHASAN

3.1 Preprocessing Model

Pemrosesan awal (preprocessing) dalam klasifikasi gambar merupakan



```

{x}   import cv2
      import os
      from sklearn.preprocessing import LabelEncoder
      import numpy as np
      import matplotlib.pyplot as plt
  
```

Gambar 1. *Install, Import and Initialization Library*

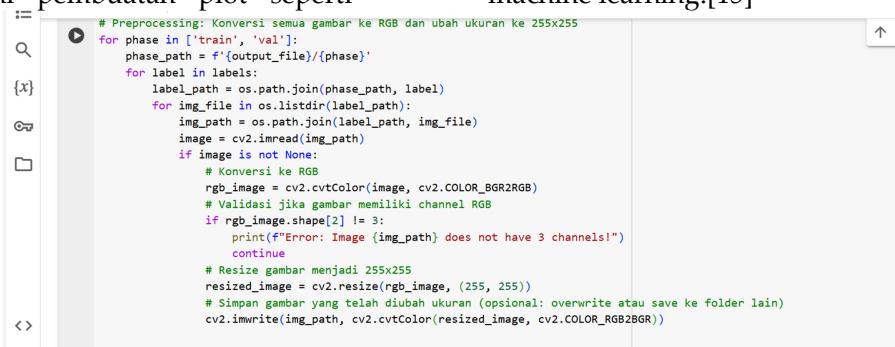
Pemasangan perpustakaan (*installation library*) perlu dilakukan agar program dapat berjalan dengan baik, ada beberapa perpustakaan yang perlu dipasang, yaitu *tensorflow*, *opencv*, dan juga *matplotlib*. *Opencv* adalah *library* gratis yang dapat digunakan untuk melakukan pengolahan citra, *library* ini juga

rangkaian langkah yang digunakan untuk mempersiapkan data citra sebelum diberikan ke model klasifikasi. Proses ini bertujuan untuk meningkatkan kualitas data sehingga model dapat lebih efektif dalam mengenali dan mengklasifikasikan objek atau fitur dalam gambar. Beberapa langkah yang biasanya dilakukan meliputi normalisasi, pengurangan noise, pembersihan citra, serta peningkatan kontras.

Tahap preprocessing ini menjadi langkah awal dalam alur kerja program yang dirancang. Pada tahap ini, data gambar akan diduplikasi, kemudian diubah ukurannya dan warnanya sesuai kebutuhan. Folder baru juga dibuat untuk mengorganisasi data hasil pengolahan, terdiri dari folder *training* dan *testing*. Di dalam masing-masing folder tersebut terdapat subfolder bernama *diabetes* dan *nondiabetes* yang masih kosong. Subfolder ini digunakan untuk menyimpan gambar hasil preprocessing sehingga data lebih terstruktur.

mendukung beberapa bahasa pemrograman salah satunya adalah *python*. *Opencv* dapat mendekripsi, mengidentifikasi, dan mengklasifikasikan berbagai macam objek. *Opencv* adalah salah satu *software library* yang biasa digunakan dalam teknik pengolahan citra digital [11]. *Tensorflow* adalah antarmuka

yang digunakan untuk mengekspresikan algoritme pembelajaran mesin dan menjalankan perintah menggunakan informasi yang ditemukan di objek yang dapat dikenali atau objek yang dapat membedakan satu objek dari objek lainnya [12]. Pustaka (*Library*) *Tensorflow* menggabungkan komputasi aljabar dengan teknik pengoptimalan terjemahan yang mendukung komputasi [13]. Pustaka *Matplotlib* adalah pustaka *Python* yang menangani pembuatan plot seperti



```

# Preprocessing: Konversi semua gambar ke RGB dan ubah ukuran ke 255x255
for phase in ['train', 'val']:
    phase_path = f'{output_file}/{phase}'
    for label in labels:
        label_path = os.path.join(phase_path, label)
        for img_file in os.listdir(label_path):
            img_path = os.path.join(label_path, img_file)
            image = cv2.imread(img_path)
            if image is not None:
                # Konversi ke RGB
                rgb_image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
                # Validasi jika gambar memiliki channel RGB
                if rgb_image.shape[2] != 3:
                    print(f"Error: Image {img_path} does not have 3 channels!")
                    continue
                # Resize gambar menjadi 255x255
                resized_image = cv2.resize(rgb_image, (255, 255))
                # Simpan gambar yang telah diubah ukuran (optional: overwrite atau save ke folder lain)
                cv2.imwrite(img_path, cv2.cvtColor(resized_image, cv2.COLOR_RGB2BGR))

```

Gambar 2. *Install, Import and Initialization Library*

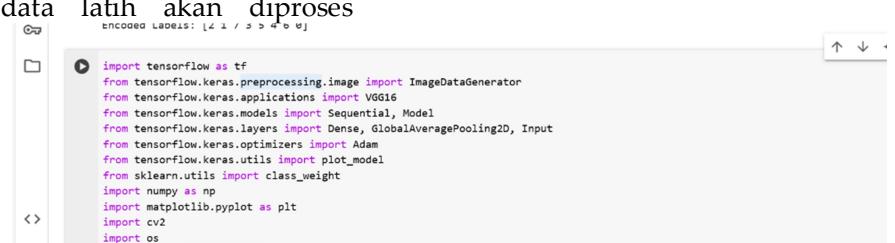
Selanjutnya, setelah semua pustaka telah terpasang, proses akan dimulai dengan menyamakan ukuran semua data citra atau gambar (seperti gambar 2), disini disamakan menjadi 255×255 piksel. Proses ini bertujuan untuk mengurangi beban komputasi pada perangkat serta memastikan gambar berada dalam bentuk persegi. Semakin besar ukuran piksel, semakin tinggi beban komputasi yang diperlukan. Setelah proses penyamaan ukuran, data latih akan diproses

menggambar, mengedit, menyimpan, dan proses lainnya [14]. *Import os* digunakan agar operating system dapat membaca program yang kita buat, serta *library numpy* digunakan untuk perhitungan *python*. *LabelEncoder* preprocessing data, di mana atribut kategori atau numerik diproses menjadi format terstruktur, sesuai dengan peran *LabelEncoder* yang mengubah data kategori menjadi nilai numerik untuk memudahkan pengolahan data pada algoritma machine learning.[15]

menggunakan pustaka modul *cv2* dari OpenCV dan disimpan ke dalam folder yang telah ditentukan sebelumnya. Langkah yang sama juga diterapkan pada data uji dengan menggunakan path atau folder yang berbeda.

3.2 Classification Model

Tahap selanjutnya adalah *classification model*. Tahapan ini diawali dengan memasukan perpustakaan (*import library*) seperti pada gambar 3

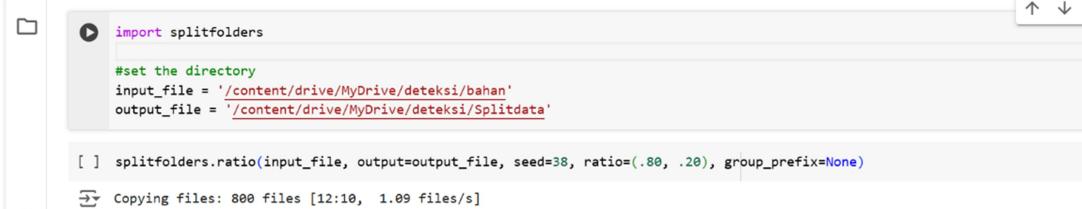


```

import tensorflow as tf
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.applications import VGG16
from tensorflow.keras.models import Sequential, Model
from tensorflow.keras.layers import Dense, GlobalAveragePooling2D, Input
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.utils import plot_model
from sklearn.utils import class_weight
import numpy as np
import matplotlib.pyplot as plt
import cv2
import os

```

Gambar 3. *Import Library*



```

import splitfolders

#set the directory
input_file = '/content/drive/MyDrive/deteksi/bahan'
output_file = '/content/drive/MyDrive/deteksi/Splitedata'

[ ] splitfolders.ratio(input_file, output=output_file, seed=38, ratio=(.80, .20), group_prefix=None)

Copying files: 800 files [12:10, 1.09 files/s]

```

Gambar 4. Membaca Folder

Kemudian dilakukan pembagian dataset yang terletak pada direktori tertentu di Google Drive.

Dataset utama, yang berisi data mentah untuk deteksi, dipisahkan menjadi dua bagian, yaitu data pelatihan (80%) dan data pengujian (20%). Pembagian ini dilakukan menggunakan metode pembagian berbasis rasio, dengan angka acak (seed) tertentu untuk memastikan

konsistensi hasil pembagian. Hasilnya akan disimpan pada direktori tujuan yang telah ditentukan, sehingga data yang telah diatur siap digunakan untuk proses pelatihan model. Parameter tambahan seperti *group_prefix* tidak digunakan, menandakan bahwa data tidak dikelompokkan berdasarkan prefiks nama file.



```

[ ] # Data augmentasi untuk training
train_datagen = ImageDataGenerator(
    rescale=1./255,
    rotation_range=30,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    brightness_range=[0.9, 1.1],
    horizontal_flip=True,
    fill_mode='nearest'
)

[ ] # Data augmentasi untuk validasi
test_datagen = ImageDataGenerator(rescale=1./255)

train_generator = train_datagen.flow_from_directory(
    '/content/drive/MyDrive/deteksi/Splitedata/train',
    target_size=(150, 150),
    batch_size=32,
    class_mode='categorical'
)

validation_generator = test_datagen.flow_from_directory(
    '/content/drive/MyDrive/deteksi/Splitedata/val',
    target_size=(150, 150),
    batch_size=32,
    class_mode='categorical',
    shuffle=False
)

Found 640 images belonging to 8 classes.
Found 160 images belonging to 8 classes.

```

Gambar 5. Augmentasi Data

Proses ini melibatkan augmentasi data untuk memperkaya variasi gambar dalam dataset pelatihan. Data pelatihan diubah dengan teknik seperti rotasi, pergeseran, zoom, perubahan kecerahan, dan pembalikan horizontal, serta disesuaikan skala pikselnya. Data validasi hanya melalui proses *rescale*

tanpa augmentasi tambahan. Dataset pelatihan dan validasi diambil dari direktori yang telah dipisahkan sebelumnya, dengan ukuran gambar distandardkan ke 150x150 piksel, batch sebesar 32, dan klasifikasi berbasis kategori. Data validasi diatur tanpa pengacakan untuk memastikan konsistensi evaluasi.



```
[1] # Preprocessing Model
base_model = VGG16(weights='imagenet', include_top=False, input_shape=(150, 150, 3))
base_model.trainable = True

Download data from https://storage.googleapis.com/tensorflow/keras-applications/vgg16/vgg16_weights_tf_dim_ordering_tf_kernels.h5
58889256/58889256 4s 0us/step
```

Gambar 6. Preprocessing Model

Selanjutnya Preprocessing Model. Pada langkah ini, digunakan model *VGG16* yang telah dilatih sebelumnya dengan bobot dari dataset *ImageNet* sebagai dasar (*base model*). Bagian atas model (lapisan klasifikasi) dihapus, sehingga hanya bagian ekstraksi fitur yang digunakan. Model disesuaikan untuk menerima input gambar dengan ukuran 150x150 piksel dan 3 saluran warna (RGB). Selain itu,

parameter *trainable* diatur ke *True* agar lapisan-lapisan dalam model dasar dapat dilatih ulang untuk menyesuaikan dengan dataset spesifik yang digunakan. *Layer Convolutional Neural Network(CNN)* dengan beberapa tahapan. Dimulai dengan mendefinisikan input gambar berukuran 150x150 piksel dengan 3 saluran warna (RGB).



```
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Dense, GlobalAveragePooling2D, Input

# Menyusun model dengan menggunakan Functional API
inputs = Input(shape=(150, 150, 3))
x = base_model(inputs, training=True)
x = GlobalAveragePooling2D()(x)
x = Dense(256, activation='relu')(x)
x = Dense(128, activation='relu')(x)
outputs = Dense(8, activation='softmax')(x) # Sesuaikan jumlah kelas

model = Model(inputs, outputs)
```

Gambar 7. Layer Convolutional Neural Network

Fitur diekstraksi menggunakan *base model* yang telah disiapkan sebelumnya. Lapisan *Global Average Pooling* digunakan untuk mengurangi dimensi fitur, diikuti oleh dua lapisan *Dense* dengan aktivasi ReLU untuk

mempelajari pola data. Akhirnya, lapisan keluaran dengan 8 neuron dan aktivasi *softmax* ditambahkan untuk klasifikasi ke dalam 8 kelas yang ditentukan.

Layer (Type)	Output Shape	Description
Input (InputLayer)	(150,150,3)	Input gambar RGB
Base Model (VGG16)	(4,4,512)	Ekstraksi fitur menggunakan VGG16
GlobalAveragePooling2D	(512)	Mereduksi dimensi fitur
Dense (Dense)	(256)	Lapisan fully connected, aktivasi ReLU
Dense (Dense)	(128)	Lapisan fully connected, aktivasi ReLU
Dense (Dense)	(8)	Lapisan output, aktivasi softmax

Tabel 1. Layer convolutional neural network

Tabel di atas menampilkan struktur atau arsitektur model

neural network yang telah dibuat.

Kemudian Model dikompilasi untuk mempersiapkannya dalam proses pelatihan. Optimizer yang digunakan adalah *Adam* dengan laju pembelajaran sebesar 0.0001, yang memberikan keseimbangan antara kecepatan konvergensi dan stabilitas

pelatihan. Fungsi *loss* yang diterapkan adalah *categorical_crossentropy*, sesuai untuk tugas klasifikasi multikelas. Metrik evaluasi yang digunakan adalah *accuracy*, untuk memantau seberapa baik model memprediksi kelas dengan benar selama pelatihan.

```
# Menghitung class weight
class_weights = class_weight.compute_class_weight(
    'balanced',
    classes=np.unique(train_generator.classes),
    y=train_generator.classes
)
class_weights_dict = dict(enumerate(class_weights))

[ ] history = model.fit(
    train_generator,
    epochs=50,
    validation_data=validation_generator,
    class_weight=class_weights_dict,
)

```

Epoch 1/50
/usr/local/lib/python3.10/dist-packages/keras/src/trainers/data_adapters/py_dataset_adapter.py:122: UserWarning: Your 'PyDataset' sub-class did not call 'super().__init__()'!
20/20 [=====] - 50s 90ms/step - accuracy: 0.1942 - loss: 2.0915 - val_accuracy: 0.3313 - val_loss: 1.6934
Epoch 2/50
20/20 [=====] - 16s 634ms/step - accuracy: 0.2820 - loss: 1.7697 - val_accuracy: 0.4437 - val_loss: 1.4796
Epoch 3/50

Gambar 9. Menghitung *class weight*

Pada langkah ini, dilakukan perhitungan bobot kelas untuk menangani ketidakseimbangan kelas dalam dataset pelatihan. Bobot kelas dihitung dengan menggunakan metode *compute_class_weight* yang mempertimbangkan distribusi kelas dalam data pelatihan dan menyesuaikan bobotnya agar model tidak condong ke kelas yang lebih

dominan. Setelah itu, pelatihan model dimulai dengan menggunakan data pelatihan yang telah disiapkan sebelumnya dan data validasi untuk evaluasi. Proses pelatihan dilakukan selama 50 epoch, dengan menggunakan bobot kelas yang telah dihitung untuk memastikan model belajar secara seimbang dari setiap kelas yang ada.

Setelah proses pelatihan selesai, langkah berikutnya adalah evaluasi model pada data validasi untuk mengukur kinerjanya. Model diuji menggunakan data validasi yang telah dipersiapkan sebelumnya, dan dua

metrik utama yang dihitung adalah *loss* dan *accuracy*. *Loss* mengukur seberapa besar kesalahan model dalam membuat prediksi, sementara *accuracy* mengukur persentase prediksi yang benar. Hasil *accuracy*

Penerapan Algoritma Convolutional Neural Network

DOI : <https://doi.org/10.35316/ijmi.9.2.142-155>

© 2024 Jurnal Ilmiah Informatika with CC BY NC licence

R. E. Mulyani dkk/ JIMI 9 (2) pp.142-155
validasi dicetak dalam bentuk persentase untuk memberikan gambaran seberapa baik model dalam

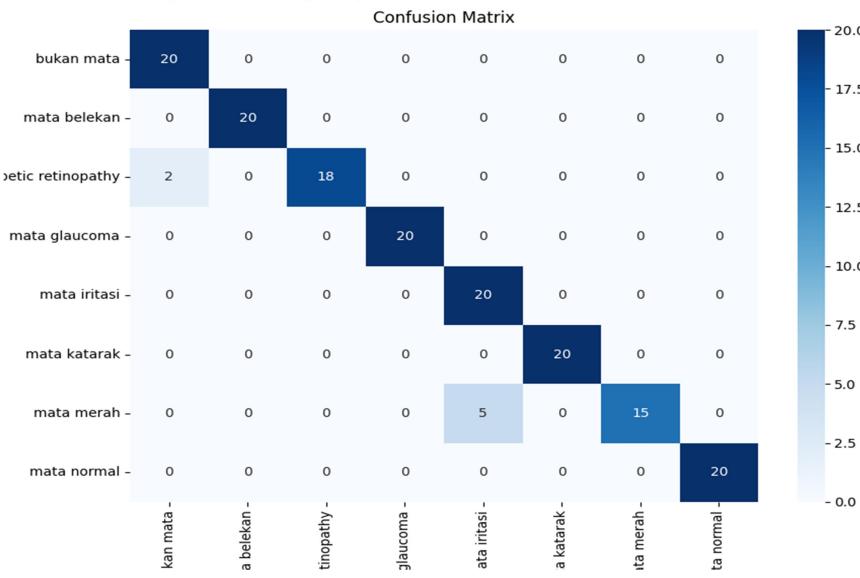
mengklasifikasikan data yang belum pernah dilihat sebelumnya.

		precision	recall	f1-score	support
	bukan mata	0.91	1.00	0.95	20
	mata belekan	1.00	1.00	1.00	20
	mata diabetic retinopathy	1.00	0.90	0.95	20
	mata glaucoma	1.00	1.00	1.00	20
	mata iritasi	0.80	1.00	0.89	20
	mata katarak	1.00	1.00	1.00	20
	mata merah	1.00	0.75	0.86	20
	mata normal	1.00	1.00	1.00	20
		accuracy		0.96	160
		macro avg	0.96	0.96	160
		weighted avg	0.96	0.96	160

Gambar 11. Classification Report

Gambar 11 ini menunjukkan evaluasi kinerja model klasifikasi berdasarkan metrik *precision*, *recall*, dan *f1-score* untuk setiap kelas. Model menunjukkan *accuracy* sebesar 96%, artinya 96% prediksi benar dari total 160 sampel validasi. Kelas seperti "mata belekan", "mata glaucoma", dan "mata katarak" menunjukkan performa sempurna dengan *precision*,

recall, dan *f1-score* mencapai 1.00. Namun, kelas "mata merah" memiliki *recall* lebih rendah (0.75), meskipun *precision* dan *f1-score* tetap tinggi. Rata-rata *macro* dan *weighted* untuk semua metrik mencapai 0.96, menunjukkan performa model yang konsisten dan baik secara keseluruhan. Berikut juga gambar confusion Matrixnya.



Gambar 12. Confusion Matrix

Selanjutnya prediksi atau inferensi menggunakan model yang telah dilatih. Pada tahapan ini, gambar baru yang belum pernah dilihat sebelumnya dimuat, diproses sesuai

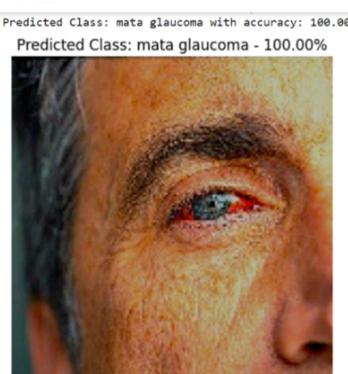
dengan format input model, dan kemudian diprediksi kelasnya berdasarkan model yang sudah terlatih.



Gambar 13. Prediksi Bukan Mata



Gambar 14. Prediksi Mata Belekan

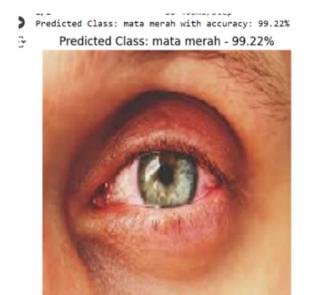
Gambar 15. Prediksi Mata *Diabetic Retinopathy*Gambar 16. Prediksi Mata *Glaukoma*



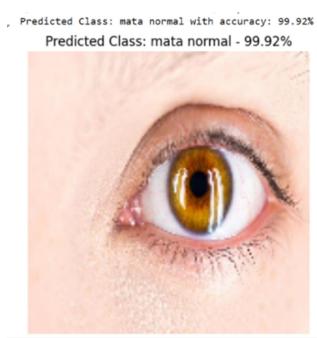
Gambar 17. Prediksi Mata Iritasi



Gambar 18. Prediksi Mata Katarak



Gambar 19. Prediksi Mata Merah



Gambar 20. Prediksi Mata Normal

4. SIMPULAN

Klasifikasi data gambar menggunakan algoritma *Convolutional Neural Network (CNN)* dengan arsitektur VGG-19 untuk mengklasifikasikan penyakit mata. Indra penglihatan atau mata memiliki peran yang sangat penting dalam kehidupan manusia, namun kesehatan mata sering terabaikan, dengan berbagai penyakit seperti katarak, glaukoma, dan retinopati diabetik dapat menyebabkan gangguan penglihatan permanen. Dengan meningkatnya prevalensi kebutaan, terutama di Indonesia, teknologi berbasis CNN menawarkan solusi untuk membantu dalam diagnosa otomatis penyakit mata.

Metode CNN, khususnya arsitektur VGG-19, terbukti efektif dalam mengklasifikasikan citra medis dan telah digunakan dalam beberapa penelitian sebelumnya dengan hasil yang cukup menjanjikan. Penggunaan dataset gambar mata yang terdiri dari 800 gambar dengan delapan label penyakit mata memungkinkan model untuk dilatih dengan data yang cukup bervariasi. Proses preprocessing data penting untuk meningkatkan kualitas gambar agar model dapat bekerja secara optimal. Selain itu, pemilihan arsitektur VGG-19 terbukti mampu menangkap fitur visual yang kompleks dalam gambar untuk meningkatkan akurasi klasifikasi.

5. REFERENSI

- [1]. R. Indraswari, W. Herulambang, and R. Rokhana, "Deteksi Penyakit Mata Pada Citra Fundus Menggunakan Convolutional Neural Network (CNN) Ocular Disease Detection on Fundus

Kemudian, data tersebut dibagi lagi menjadi dua kategori, yaitu 20% dari data keseluruhan file gambar mata normal dan begitu juga dengan file gambar data penderita retinopati diabetes. Dari 20% data tersebut, sebagian akan digunakan untuk data uji coba, sementara 80% sisanya digunakan untuk data latih. Adapun hasil dari pelatihan data tersebut menunjukkan hasil yang cukup memuaskan, dengan diperoleh tingkat akurasi prediksi sebesar 96%. Penelitian ini membuktikan bahwa dengan menggunakan metode CNN dan arsitektur VGG-19, proses klasifikasi penyakit mata dapat dilakukan secara efektif, yang berpotensi membantu dalam pengobatan dan diagnosa penyakit mata, terutama di daerah dengan keterbatasan tenaga medis. Harapan peneliti pada penelitian selanjutnya adalah dapat menerapkan aplikasi *Convolutional Neural Network (CNN)* dengan model yang lebih baik lagi, serta dengan jumlah data yang lebih beragam dan penggunaan layer yang lebih variatif, sehingga hasil yang diperoleh dapat lebih optimal dan dapat diterapkan dalam skala yang lebih luas. Akurasi tinggi pada model ini menunjukkan potensi besar penggunaan deep learning untuk mendeteksi penyakit mata secara otomatis, yang dapat meningkatkan kualitas pelayanan kesehatan mata di masa depan.

Images Using Convolutional Neural Network (CNN)," vol. 21, no. 2

- [2]. A. Sudrajat, "Faktor risiko terjadinya katarak terhadap katarak senil pada petani di wilayah kerja Puskesmas Tempurejo Kabupaten Jember,"

- Multidisciplinary Journal, vol. 4, no. 2, pp. 1–6, 2020.
- [3]. The International Agency for the Prevention of Blindness (IAPB), "Vision Atlas," 2020.
- [4]. M. J. Burton et al., "The Lancet Global Health Commission on Global Eye Health: vision beyond 2020," *Lancet Glob. Heal.*, vol. 9, no. 4, pp. e489–e551, 2021, doi: 10.1016/S2214-109X(20)30488-5.
- [5]. F. N. Cahya, N. Hardi, D. Riana, and S. Hadiyanti, "Klasifikasi penyakit mata menggunakan Convolutional Neural Network (CNN)," *SISTEMASI*, vol. 10, no. 3, pp. 1–8, 2021.
- [6]. G. J. Bu'ulolo, A. Jacobus, and F. D. Kambey, "Identifikasi citra penyakit mata katarak menggunakan Convolutional Neural Network," *Jurnal Teknik Informatika*, vol. 16, no. 4, pp. 375–382, 2021.
- [7]. J. V. N. Rakamawati, "Klasifikasi Diabetic Retinopathy berdasarkan foto fundus menggunakan Convolutional Neural Network (CNN) jenis DenseNet," p. 83, 2021.
- [8]. M. Parikesit, "Deteksi penyakit diabetes retinopati menggunakan Visual Geometry Group (VGG) 19," unpublished, 2020.
- [9]. S. Rizal, N. Ibrahim, N. K. C. Pratiwi, S. Saidah, and R. Y. N. Fu'adah, "Deep learning untuk klasifikasi diabetic retinopathy menggunakan model EfficientNet," *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, vol. 8, no. 3, pp. 693–700, 2020.
- [10]. D. Rosmala and F. Arieffansyah, "Identifikasi citra X-ray tulang penyakit osteoporosis menggunakan Visual Geometry Group (VGG) 19," *Journal MIND*, vol. 9, no. 1, pp. 1–6, 2020.
- [11]. A. A. N. G. Sapteka et al., "Pendeteksi penggunaan masker wajah dengan ESP32Cam menggunakan OpenCV dan Tensorflow," *Majalah Ilmiah Teknologi Elektro*, vol. 21, no. 2, pp. 155–160, 2022.
- [12]. H. G. Ghifari, D. Darlis, and A. Hartaman, "Pendeteksi golongan darah manusia berbasis TensorFlow menggunakan ESP32-CAM," *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, vol. 9, no. 2, pp. 359–364, 2021.
- [13]. M. Malik, "Deteksi suhu tubuh dan masker wajah dengan MLX90614, OpenCV, Keras/TensorFlow, dan deep learning," vol. 6, no. 1, pp. 1–5, 2022.
- [14]. H. Setiawan, E. Utami, and H. Al Fatta, "Penerapan ARIMA dan artificial neural network untuk prediksi penderita DBD di Kabupaten Sragen," *Majalah Ilmiah Bahari Jogja*, vol. 18, no. 2, pp. 64–78, 2020.
- [15]. Saputra et al., "Analisis data sentimen kepuasan pengguna e-wallet menggunakan metode K-Nearest Neighbor," *Jurnal Riset Sistem Informasi Dan Teknik Informatika (JURASIK)*, vol. 9, no. 2, pp. 1021–1029, 2024.