JURNAL TEKNOINFO
Volume 18, Nomor 1, Januari 2024, Page 121-132
ISSN: 1693-0010(Print), ISSN: 2615-224X(Online)
Available online at https://ejurnal.teknokrat.ac.id/index.php/teknoinfo/index

DETEKSI PENYAKIT RETINOPATI DIABETES MENGGUNAKAN CITRA MATA DENGAN IMPLEMENTASI DEEP LEARNING CNN

Muhammad Zahir[1)], Rizal Adi Saputra[2)]

[1]Jurusan Teknik Informatika, Universitas Halu Oleo
[2] Jurusan Teknik Informatika, Universitas Halu Oleo
[1] H.E.A. Mokodompi, Kota Kendari
Email: [1]zahirmuhammad260@gmail.com, [2]rizaladisaputra@uho.ac.id*

Abstract

Abstract−Diabetes is one of the most common types of disease in Indonesia. One of the main causes of this disease is increased blood sugar or glucose levels. Proper treatment of this disease can be done one way with early detection. Classification of diabetes is a very important process in detecting this disease. The use of Convolutional Neural Networks (CNN) in recent years has shown promising results as a classification method. In this study we used an approach for diabetes classification with the Convolutional Neural Network (CNN) model, where we used a dataset consisting of two types of classifications, namely diabetes and non-diabetic datasets to train and test the CNN model. The dataset is in the form of eye fundus images from diabetic patients and normal patients. We carried out several stages in the classification process, first we prepared a dataset of eye fundus images, secondly, data pre-processing included reducing the image pixel value, converting the image to a grayscale image, giving the image a name, third, training and testing stages of image data, fourth, creating a model. sequential, and the fifth stores the model results. This study uses the implementation of Deep learning with Jupyter notebook as one of the dataset processing media and Spyder as a user interface or GUI display media. The results of our study show satisfactory predictive accuracy in the diabetes classification process. This study shows that the use of the Convolutional Neural Network (CNN) model can be used as a fairly effective tool in predicting diabetes classification which can later detect patients who are likely to have diabetes. The results of this study indicate that the predictions made obtained an accuracy value of 96%, these results are fairly good.

Keyword: *Detection, Diabetic Retinopathy, Deep Learning, Convolutional Neural Network, Classification*

Abstrak

Diabetes merupakan salah satu jenis penyakit yang paling umum terjadi di Indonesia. Salah satu penyebab utama terjadinya penyakit ini yaitu meningkatnya kadar gula darah atau glukosa. Perawatan yang tepat pada penyakit ini dapat dilakukan salah satunya dengan deteksi dini. Klasifikasi diabetes merupakan proses yang sangat penting dalam melakukan deteksi pada penyakit ini. Penggunaan *Convolutional Neural Network* (*CNN*) dalam beberapa tahun terakhir memberikan hasil yang menjanjikan sebagai salah satu metode klasifikasi. Dalam penelitian ini kami menggunakan sebuah pendekatan untuk klasifikasi diabetes dengan model *Convolutional Neural Network* (*CNN*), dimana kami menggunakan dataset yang terdiri dari dua jenis klasifikasi yakni dataset diabetes dan tidak diabetes untuk melatih dan menguji model *CNN*. Dataset tersebut berupa citra atau gambar fundus mata dari pasien penderita diabetes dan pasien normal. Kami melakukan beberapa tahapan dalam proses klasifikasi, pertama kami menyiapkan dataset citra fundus mata, kedua pra-pemrosesan data meliputi pengurangan nilai *pixel* gambar, mengubah gambar menjadi gambar *grayscale*, memberi nama pada gambar, ketiga tahapan latihan dan pengujian terhadap data citra, keempat membuat model *sequential*, dan kelima menyimpan hasil model. Penelitian ini menggunakan implementasi *Deep learning* dengan *Jupyter notebook* sebagai salah satu media pengolahan dataset dan *Spyde*r sebagai media tampilan *user interface* atau *GUI*. Hasil penelitian ini menunjukan prediksi akurasi yang cukup memuaskan dalam proses klasifikasi diabetes. Penelitian ini menunjukan bahwa penggunaan model *Convolutional Neural Network* (*CNN*) dapat digunakan sebagai alat yang efektif dalam klasifikasi prediksi diabetes yang nantinya dapat mendeteksi pasien yang kemungkinan menderita diabetes. Hasil dari penelitian ini menunjukan bahwa prediksi yang dilakukan memperoleh nilai akurasi 96%, hasil tersebut terbilang baik.

Kata Kunci: Deteksi, Retinopati Diabetes, Deep Learning, Convolutional Neural Network, Klasifikasi

1. PENDAHULUAN

Pada tahun 2020, 1,1 miliar orang di seluruh dunia telah mengalami gangguan penglihatan. Jumlah ini diperkirakan akan terus bertambah hingga mencapai 1,76 miliar orang pada tahun 2050. Penyebab paling umum yang sering terjadi hingga menyebabkan kebutaan pada anak dan remaja adalah penyakit mata yang dapat dicegah jika diketahui dan diobati

JURNAL TEKNOINFO
Volume 18, Nomor 1, Januari 2024, Page 121-132
ISSN: 1693-0010(Print), ISSN: 2615-224X(Online)
Available online at https://ejurnal.teknokrat.ac.id/index.php/teknoinfo/index

sejak dini. Mata (oculus) adalah salah satu organ vital terpenting dalam tubuh. Mata adalah salah satu dari lima indra manusia [1].

Pemeriksaan fundus banyak digunakan untuk mendeteksi penyakit yang gejalanya berhubungan dengan mata manusia. Gambar fundus digital dapat memberikan informasi tentang perubahan patologis yang disebabkan oleh kelainan mata dan penyakit sistemik [2]. Pemeriksaan fundus banyak digunakan untuk mendeteksi penyakit yang gejalanya berhubungan dengan mata manusia. Gambar fundus digital dapat memberikan informasi tentang perubahan patologis yang disebabkan oleh kelainan mata dan penyakit sistemik. Pengembangan metode klasifikasi diabetes menggunakan *Convolutional Neural Network (CNN)* dalam konteks pengolahan citra medis telah menarik minat peneliti sebagai solusi yang potensial dalam diagnosis dan deteksi dini penyakit tersebut. Diabetes, sebagai salah satu penyakit kronis yang mempengaruhi jutaan orang di seluruh dunia, memerlukan upaya yang serius dalam pengidentifikasian dan klasifikasi yang akurat. Dalam penelitian ini, kami akan menjelaskan metode *CNN* yang dikembangkan secara khusus untuk mengenali dan mengklasifikasikan gambar atau citra medis terkait diabetes.

Retinopati Diabetes merupakan salah satu dari beberapa jenis penyakit diabetes yang dapat diidentifikasi lewat mata. Penyakit tersebut juga dapat menyebabkan kebutaan bagi penderitanya, hal ini dikarenakan terdapat beberapa luka pada mata yang dapat merusak retina mata. Gejala pertama adalah pelebaran pembuluh darah di mata. Jika hal ini terus berlanjut, akan terbentuk pembuluh darah baru yang dapat menyumbat retina mata dan berujung pada kebutaan. [3]. Penyebab hilangnya penglihatan dalam lima puluh tahun terakhir salah satunya adalah retinopati diabetes. Gejala-gejala yang terjadi tidak dapat diamati secara langsung (kasat mata), oleh karena itu dibutuhkan citra atau gambar dari fundus mata. Dan dari foto fundus tersebut kita dapat melakukan pengamatan namun agar proses pengamatan dapat lebih efisien dan optimal maka dibutuhkan sebuah program sistem yang dapat membantu analisis dan juga klasifikasi terhadap citra atau gambar fundus tersebut. *Deep learning* Pembelajaran mendalam (*Deep learning)* adalah cabang pembelajaran mesin yang terdiri dari algoritma pemodelan abstrak tingkat tinggi untuk data dengan menggunakan serangkaian fungsi transformasi nonlinier yang diatur ke dalam level dan kedalaman. [4]. Model *Convolutional Neural Network (CNN)* adalah pendekatan pemodelan untuk pembelajaran mendalam, sebuah konsep pembelajaran mesin. Model *Convolutional Neural Network (CNN)* adalah kelas jaringan saraf tiruan yang paling umum digunakan untuk analisis citra visual, di mana model arsitekturnya terdiri dari lapisan simpul yang mencakup lapisan masukan, satu atau lebih lapisan tersembunyi, dan lapisan keluaran. [5]. Model penelitian ini menggunakan *Convolutional Neural Network* (*CNN*) untuk klasifikasi data berupa citra fundus mata untuk deteksi penyakit retinopati diabetes. Penelitian ini menggunakan aplikasi *Anaconda* sebagai media untuk membuat model dan juga sistem klasifikasi data.

Penelitian ini bukanlah penelitian baru melainkan telah banyak peneliti yang telah melakukan penelitian ini. Diantaranya Syamsul rizal, dkk, melakukan penelitian dengan judul "*Deep Learning* untuk Klasifikasi *Diabetic Retinopathy* menggunakan Model *EfficientNet*" dimana mereka melakukan pengujian data dengan empat scenario, dan didapatlah kesimpulan bahwa model *EfficientNet* dapat meningkatkan akurasi dalam mengklasifikasikan 5 stadium retinopati diabetik dibandingkan dengan model lainnya. Setelah *pretreatment* kulit kayu dalam penelitian ini, akurasi meningkat menjadi 79,8%. penelitian ini juga mengungkapkan bahwa tidak disarankan untuk mengubah ukuran dari citra baik itu memperbesar atau sebaliknya karena dapat mempengaruhi nilai akurasi. Studi ini juga menunjukkan bahwa pada penelitian selanjutnya diharapkan penggunaan jumlah materi yang seimbang antar kategori. Dalam hal ini terdapat perbedaan jumlah masing-masing kategori bahan yang tidak berbeda secara signifikan satu sama lain. Hal tersebut dilakukan guna menghindari kesalahan pembacaan data pada sistem, dimana sistem hanya akan melatih kelas yang dominan nantinya [3].

Klasifikasi retinopati diabetes juga pernah dilakukan oleh Syafiq Hilmi Abdullah dengan judul "Diabetic *Retinopathy Classification Based on Fundus Image Processing and Deep Learning"* mengembangkan sistem *Convolutional Neural Network (CNN)* yang menggunakan model *EfficientNet* untuk melakukan klasifikasi secara efektif dan efisien. Dataset yang digunakan adalah APTOS 2019 *Blindness Detection,* yang berisi 3.662 citra *RGB* yang terbagi dalam lima kategori: No *DR*, *Mild NPDR*, *Moderate NPDR, Severe NPDR* dan *Proliferate DR*. Hasil akhir menunjukkan bahwa model terbaik melewati pengoptimal AdaMax, menggunakan tingkat pembelajaran 0,001, dan ukuran tumpukan 32, dengan akurasi 82,096%, nilai presisi 67,6%, nilai *recall* 63,4%, dan skor f-1 64,6. %. Berdasarkan hasil penelitian, kinerja sistem menunjukkan bahwa model yang dibuat dapat digunakan sebagai sistem deteksi dini dan mempersingkat waktu pemeriksaan medis pada pasien retinopati diabetik [6].

Penerapan model *Convolutional Neural Network* (*CNN*) untuk klasifikasi penyakit retinopati diabetes menggunakan citra fundus mata pernah dilakukan oleh Eko Hari Rachmawanto dengan judul "*Convolutional Neural Network (CNN)* untuk Klasifikasi Citra Penyakit *Diabetes Retinopathy*" Penelitian ini mengoptimalkan klasifikasi citra retina mata yang didiagnosis retinopati menggunakan algoritma *Convolutional Neural Network (CNN)* dengan tujuan untuk mengidentifikasi pembuluh darah pada retina mata yang didiagnosis *DR*. Tujuan utama dari penelitian ini adalah untuk menguji kinerja *CNN* dalam proses klasifikasi citra *DR* dengan dataset yang cukup besar tanpa *pre-processing* untuk menyimpulkan bahwa *CNN* saja sudah mampu mengklasifikasikan objek dengan benar. Gambar berasal dari database

JURNAL TEKNOINFO
Volume 18, Nomor 1, Januari 2024, Page 121-132
ISSN: 1693-0010(Print), ISSN: 2615-224X(Online)
Available online at https://ejurnal.teknokrat.ac.id/index.php/teknoinfo/index

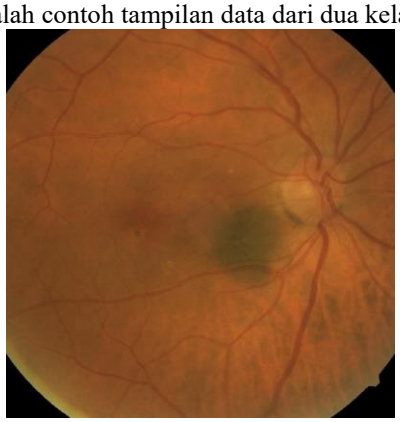
Kaggle, yang berisi total 88.702 gambar yang diurutkan berdasarkan 88.000 tanggal. Hasil klasifikasi citra yang benar sebanyak 82445 dengan persentase 93,68%, sedangkan citra yang salah klasifikasi sebanyak 5555 citra dengan tingkat kesalahan 6,32% [7].

Dari beberapa penelitian yang telah dilakukan tersebut maka peneliti mengangkat sebuah judul "Deteksi Penyakit Retinopati Diabetes Menggunakan Citra Fundus Mata Dengan Implementasi *Deep Learning CNN*" penelitian ini dibangun menggunakan model *Convolutional Neural Network (CNN)* dengan menggunakan *Anaconda jupyter notebook* sebagai media pemrosesan, pengolahan dan klasifikasi data serta penggunaan *Anaconda spyder* sebagai alat untuk menampilkan tampilan *GUI* agar memudahkan dalam melakukan proses interaksi user dan sistem.

2. METODE PENELITIAN

2.1 Dataset

Dataset dalam penelitian ini merupakan dataset sekunder yang diperoleh dari Kaggle (https://www.kaggle.com/datasets/gunavenkatdoddi/eye-diseases-classification). Dataset tersebut terdiri dari empat kelas yakni katarak, glukoma, retinopati diabetes, dan normal. Dimana data katarak terdiri dari 1038 file gambar, data glukoma terdiri dari 1007 file gambar, data normal terdiri dari 802 file gambar dan terakhir data retinopati diabetes terdiri dari 796 file gambar. Namun pada penelitian ini, peneliti hanya menggunakan dua jenis kelas dari empat kelas tersebut dimana kelas yang dipakai adalah kelas retinopati diabetes yang kemudian dibagi lagi menjadi dua *path* atau folder yakni folder pelatihan dan pengujian. Kemudian kelas yang kedua adalah kelas normal dimana kelas ini dibagi menjadi dua *path* yakni path pelatihan dan pengujian. Berikut adalah contoh tampilan data dari dua kelas tersebut.



Gambar 1. Citra atau Gambar Mata Retinopati Diabetes

Gambar 2. Citra atau Gambar Mata Normal

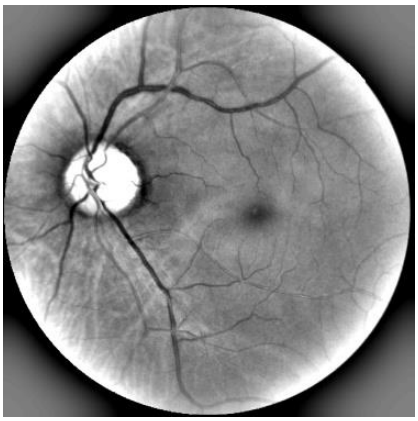
Gambar diatas merupakan salah satu contoh yang akan digunakan pada proses klasifikasi.

2.2 *Preprocessing Data*

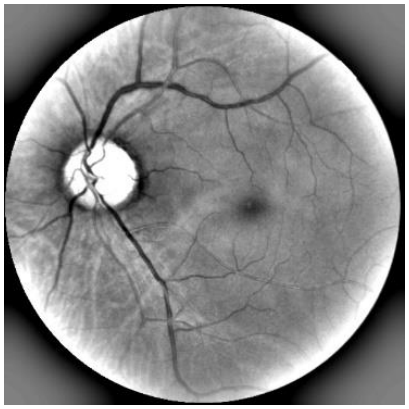
Tahapan selanjutnya adalah *preprocessing data.* Langkah ini dapat menyederhanakan jumlah matriks pada gambar. Jika gambar RGB memiliki tiga matriks, masing-masing mewakili saluran warna, maka gambar skala abu-abu hanya memiliki satu matriks. Pada gambar jenis ini, warna dinyatakan dalam intensitas antara 0 dan 255. Nilai 0 menunjukkan

JURNAL TEKNOINFO
Volume 18, Nomor 1, Januari 2024, Page 121-132
ISSN: 1693-0010(Print), ISSN: 2615-224X(Online)
Available online at https://ejurnal.teknokrat.ac.id/index.php/teknoinfo/index

warna hitam dan nilai 255 menunjukkan warna putih [8]. Data berupa citra fundus mata akan dilakukan pengubahan ukuran yaitu *resize* gambar, agar ukuran dari setiap gambar yang ada menjadi sama besar. Setelah semua data memiliki ukuran yang sama maka selanjutnya data akan diubah warnanya menjadi *grayscale*. Selain itu, setelah dilakukan *resizing* data, data yang terkumpul dibagi menjadi dua bagian yaitu data latih dan data uji. . Setelah itu masuk kedalam tahapan berikutnya yaitu pemberian nama atau pelabelan untuk tiap-tiap data yaitu data diabetes dan normal (*nondiabetes*) [9]. Pada tahap ini data sudah menjadi data mentah siap pakai untuk digunakan pada proses tahapan selanjutnya. Data tahapan *preprocessing* dapat dilihat pada gambar berikut.



Gambar 3. Citra Mata Diabetes *Preprocessing*



Gambar 4. Citra Mata Normal *Preprocessing*

2.3 Proses Latih *Data*

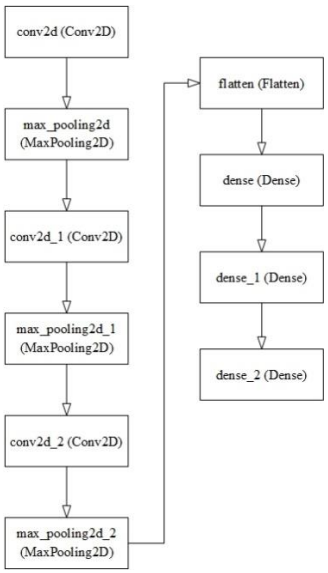
Dalam pelatihan, kumpulan data dibagi menjadi dua bagian, sehingga terdiri dari 80% data latih dan 20% untuk data uji. Data uji diperlukan untuk mengkonfirmasi keakuratan model yang dibuat. Untuk membandingkan dan menemukan nilai akurasi terbaik, diterapkan beberapa skenario [3]. Proses ini dilakukan 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 di lakukan iterasi sebanyak 150 *epoch* dan 4 *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 [10]. Model yang dikembangkan terdiri dari jumlah layer yang digunakan, spesifikasi filter, spesifikasi ukuran kernel dan spesifikasi fungsi aktivasi, serta spesifikasi pool [9]. Model *CNN* terdiri dari beberapa *layer* yaitu *convolutional layer, activation function, pooling layer, flat layer* dan *fully connected layer*. Lapisan pertama adalah *convolutional layer* dimana matriks citra masukan dikalikan dengan *filter*, kemudian hasil perkalian tersebut dijumlahkan dan disimpan dalam matriks baru yang disebut *feature map*. Perkalian dilakukan dengan konvolusi sesuai dengan dimensi citra masukan, dan persamaan fungsi lapisan ini diberikan pada Persamaan 1 [4].

JURNAL TEKNOINFO
Volume 18, Nomor 1, Januari 2024, Page 121-132
ISSN: 1693-0010(Print), ISSN: 2615-224X(Online)
Available online at https://ejurnal.teknokrat.ac.id/index.php/teknoinfo/index

Berikut adalah contoh dari layer *Convolutional Neural Network (CNN)* yang akan dibuat dan digunakan.



Gambar 5. *Layer CNN*

$$(f * g)(n) = \sum_{-\infty}^{\infty} f(m)g(n-m) \quad (1)$$

Pada titik ini, model *CNN* untuk klasifikasi retinopati diabetik dikembangkan. Sebelum model dibuat, data harus dikumpulkan terlebih dahulu dan dataset dipecah menjadi data pelatihan dan data uji. Tujuan *augmentasi* adalah untuk meningkatkan akurasi model dengan mengekstraksi informasi tambahan yang berguna [11]. Berbeda dengan proses *extension* pada saat *pre-processing* yang hanya berfokus pada peningkatan jumlah *record* kelas yang berat, proses *extension* pada tahap ini juga bertujuan untuk meningkatkan performa akurasi dari model yang dibangun. Proses penambahan pada tahap ini dilakukan dengan menskalakan skala citra dan membalik citra secara horizontal [12]. Kemudian membagi kumpulan data atau dataset menjadi data latih dengan data latih 80%, data uji 20% dan batch size 4%.

3. HASIL DAN PEMBAHASAN

3.1 *Preprocessing Model*

Preprocessing model dalam klasifikasi gambar atau citra merupakan serangkaian langkah atau metode yang digunakan untuk mengolah data citra dan mempersiapkannya sebelum diberikan kepada model klasifikasi. Tujuan utama dari proses klasifikasi adalah meningkatkan kemampuan dan kualitas model dalam mengenali dan mengklasifikasikan fitur dan objek dalam gambar atau citra menjadi lebih akurat, dengan melibatkan beberapa tahapan yang umum digunakan seperti normalisasi, penghilangan *noise,* pembersihan gambar atau citra, dan peningkatan kontras gambar atau citra.

Preprocessing model merupakan tahapan awal dari seluruh rangkaian kerja alur program yang akan dibuat. Dimana pada tahapan ini, data akan di duplikasi dengan ukuran dan warna yang akan diubah melalui tahapan ini. Pada tahapan ini, diperlukan folder baru yang terdiri atas folder training dan testing dimana didalam folder tersebut akan terdapat sebuah folder baru yang masih kosong, dan diberi nama diabetes dan *nondiabetes.* Folder tersebut dibuat agar nantinya dapat menjadi tempat penyimpanan untuk file gambar hasil pengolahan pada tahap *preprocessing.* Adapun alur kerja dari kode program dapat dilihat pada gambar 5.

JURNAL TEKNOINFO
Volume 18, Nomor 1, Januari 2024, Page 121-132
ISSN: 1693-0010(Print), ISSN: 2615-224X(Online)
Available online at https://ejurnal.teknokrat.ac.id/index.php/teknoinfo/index

```
[ ]: !pip install tensorflow opencv-python matplotlib

[1]: from matplotlib.pyplot import imread, imshow, subplots, show
     from matplotlib import pyplot as plt
     import matplotlib.image as mpimg
     import os
     import cv2
     import numpy as np
     import zipfile
     from os import listdir
     from os.path import isfile, join
```

Gambar 6. *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 mendukung beberapa bahasa pemrograman salah satunya adalah *python*. *Opencv* dapat mendeteksi, mengidentifikasi, dan mengklasifikasikan berbagai macam objek yang ada dialam ini. *Opencv* merupakan salah satu *software library* yang biasa digunakan dalam teknik pengolahan citra digital [13]. *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 [14]. Pustaka (*Library*) *Tensorflow* menggabungkan komputasi aljabar dengan teknik pengoptimalan terjemahan yang mendukung komputasi [15]. Pustaka *Matplotlib* adalah pustaka *Python* yang menangani pembuatan plot seperti menggambar, mengedit, menyimpan, dan proses lainnya [16]. *Import os* digunakan agar operating system dapat membaca program yang kita buat, serta *library numpy* digunakan untuk perhitungan *python.*

```
: IMG_SIZE = 612

: path_trainDIB = 'Training/diabetes'

: n = 1
  for filename in os.listdir(path_trainDIB):
      image_trainDIB = cv2.imread(f"{path_trainDIB}/{filename}")
      image_trainDIB = cv2.cvtColor(image_trainDIB, cv2.COLOR_BGR2GRAY)
      image_trainDIB = cv2.resize(image_trainDIB, (IMG_SIZE, IMG_SIZE))
      image_trainDIB = cv2.addWeighted(image_trainDIB,4, cv2.GaussianBlur(image_trainDIB , (0,0), 40) ,-4 , 128)
      cv2.imwrite(f"Data_Kurangi_Filter/Training/diabetes/Diabetes_{n}.png", image_trainDIB)
      n += 1

: path_trainNonDIB = 'Training/non_diabetes'

: n = 1
  for filename in os.listdir(path_trainNonDIB):
      image_trainNonDIB = cv2.imread(f"{path_trainNonDIB}/{filename}")
      image_trainNonDIB = cv2.cvtColor(image_trainNonDIB, cv2.COLOR_BGR2GRAY)
      image_trainNonDIB = cv2.resize(image_trainNonDIB, (IMG_SIZE, IMG_SIZE))
      image_trainNonDIB = cv2.addWeighted(image_trainNonDIB,4, cv2.GaussianBlur(image_trainNonDIB , (0,0), 40) ,-4 , 128)
      cv2.imwrite(f"Data_Kurangi_Filter/Training/non_diabetes/NonDiabetes_{n}.png", image_trainNonDIB)
      n += 1
```

Gambar 7. *Image Preprocessing*

Selanjutnya, setelah *library* telah terpasang semua, maka proses akan dimulai dengan penyamaan ukuran semua data citra atau gambar (seperti gambar 6), disini menggunakan ukuran 612. Fungsi dari penyamaan atau pengurangan nilai pixel citra adalah untuk mengurangi beban komputasi pada laptop, dan mengubah menjadi bentuk persegi, dimana nilai pixel citra berbanding lurus dengan beban komputasi komputer. Setelah itu data *training* atau latih akan diproses menggunakan *library module cv2* dari *opencv,* lalu data tersebut akan disimpan pada folder yang telah dibuat sebelumnya. Lakukan yang sama untuk *path* atau folder pada *testing* atau data uji.

3.2 *Classification Model*

Tahap selanjutnya adalah *classification model*. Tahapan ini diawali dengan memasukan perpustakaan (*import library*) seperti pada gambar 7.

JURNAL TEKNOINFO
Volume 18, Nomor 1, Januari 2024, Page 121-132
ISSN: 1693-0010(Print), ISSN: 2615-224X(Online)
Available online at https://ejurnal.teknokrat.ac.id/index.php/teknoinfo/index

```
import os
import numpy as np
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Conv2D, MaxPool2D, Flatten
import tensorflow as tf
from tensorflow.keras.preprocessing import image
import matplotlib.image as mpimg
import matplotlib.pyplot as plt
import cv2
%matplotlib inline
```

Gambar 8. *Import Library*

```
print('total diabetes training images :', len(os.listdir('Data_Kurangi_Filter/Training/diabetes')))
print('total non diabetes training images :', len(os.listdir('Data_Kurangi_Filter/Training/non_diabetes')))
print('total diabetes testing images :', len(os.listdir('Data_Kurangi_Filter/Testing/diabetes')))
print('total non diabetes testing images :', len(os.listdir('Data_Kurangi_Filter/Testing/non_diabetes')))
```

```
total diabetes training images : 637
total non diabetes training images : 642
total diabetes testing images : 159
total non diabetes testing images : 160
```

```
train_dir = os.path.join('Data_Kurangi_Filter/Training')
testing_dir = os.path.join('Data_Kurangi_Filter/Testing')
```

```
class_names = ['Diabetic', 'Normal']
print(class_names)
```

```
['Diabetic', 'Normal']
```

Gambar 9. Membaca Folder

Kemudian, untuk mengetahui jumlah banyak data yang dimiliki pada sebuah folder dapat menuliskan kode pada gambar 8. Untuk membaca atau mengambil informasi data yang telah dilakukan *preprocessing* dapat dilakukan dengan "*os.path.join*". Selanjutnya membuat nama kelas (*label*) yang terdiri dari dua kelas yang diberi nama *diabetic* dan normal.

```
train_datagen = ImageDataGenerator(
    rescale=1./255,
    horizontal_flip=True,
    zoom_range=0.3,
    shear_range=0.2,
    validation_split=0.1) # set validalition split
```

```
train_generator = train_datagen.flow_from_directory(
    train_dir,
    target_size=(150, 150),
    batch_size=4,
    class_mode='categorical',
    subset='training') # set as training data
```

```
Found 1152 images belonging to 2 classes.
```

```
train_generator.class_indices
```

```
{'diabetes': 0, 'non_diabetes': 1}
```

Gambar 10. Augmentasi Data

Augmentasi data bertujuan agar sistem dapat membaca data walaupun data diambil dari sudut yang beraneka ragam atau kondisi secara *real time.* Augmentasi bertujuan agar data dapat dikenali walaupun kondisi data mengalami perubahan. Batch_size adalah jaringan syaraf yang dilewati sejumlah data dalam suatu waktu, dan menjadi penentu banyaknya sampel yang dikerjakan sebelum pembaharuan parameter model internal.

```
model = Sequential()
model.add(Conv2D(64,input_shape=(150,150,3),kernel_size=(3,3), activation="relu"))
model.add(MaxPool2D(pool_size=(2,2),strides=(2,2)))
model.add(Conv2D(128, kernel_size=(3,3), activation="relu"))
model.add(MaxPool2D(pool_size=(2,2), strides=(2,2)))
model.add(Conv2D(256, kernel_size=(3,3), activation="relu"))
model.add(MaxPool2D(pool_size=(2,2), strides=(2,2)))
model.add(Flatten())
model.add(Dense(units=512,activation="relu"))
model.add(Dense(units=512,activation="relu"))
model.add(Dense(units=2, activation="softmax"))
```

```
model.summary()
```

Gambar 11. *Layer Convolutional Neural Network*

JURNAL TEKNOINFO
Volume 18, Nomor 1, Januari 2024, Page 121-132
ISSN: 1693-0010(Print), ISSN: 2615-224X(Online)
Available online at https://ejurnal.teknokrat.ac.id/index.php/teknoinfo/index

Layer Convolutional Neural Network (CNN), terbagi kedalam beberapa macam. Adapun yang digunakan dalam penelitian ini yaitu 10 layer yang terdiri dari *Convolutional Layer* berguna untuk meng-ekstraksi berbagai macam fitur yang ada dalam gambar/citra inputan. Fungsi aktivasi yang digunakan adalah "*Relu*", fungsi yang lazim digunakan karena memiliki perhitungan komputasi yang efisien dan sederhana. Kedua, ada "*MaxPool*" berfungsi untuk mempertahankan informasi penting hasil reduksi dimensi spasial dari (*feature map*) yang dihasilkan oleh lapisan konvolusional. Ketiga, "*Flatten*" merupakan operasi yang dapat meratakan matriks multidimensional menjadi vektor linear. Keempat, "*Dense*" digunakan karena kemampuannya yang mampu menggabungkan informasi penting yang terkandung dalam fitur-fitur yang telah diekstraksi sebelumnya dengan pola yang lebih kompleks dan abstrak.

Tabel 1. *Layer convolutional neural network*

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 148, 148, 64)	1792
max_pooling2d (MaxPooling2D)	(None, 74, 74, 64)	0
conv2d_1 (Conv2D)	(None, 72, 72, 128)	73856
max_pooling2d_1 (MaxPooling2D)	(None, 36, 36, 128)	0
conv2d_2 (Conv2D)	(None, 34, 34, 256)	295168
max_pooling2d_2 (MaxPooling2D)	(None, 17, 17, 256)	0
flatten (Flatten)	(None, 73984)	0
dense (Dense)	(None, 512)	37880320
dense_1 (Dense)	(None, 512)	262656
dense_2 (Dense)	(None, 2)	1026
	Total params: 38,514,818	
	Trainable params: 38,514,818	
	Non-trainable params: 0	

Tabel 1. Merupakan tampilan hasil pengolahan dalam bentuk layer tabel yang dilakukan pada gambar 10.

```
model.compile(loss='binary_crossentropy',optimizer=tf.optimizers.Adamax(learning_rate=0.001),metrics=['accuracy'])
```

```
epoch = 150
history = model.fit(train_generator,validation_data=validation_generator,epochs=epoch,verbose=2,steps_per_epoch=50,validation_steps=5)
```

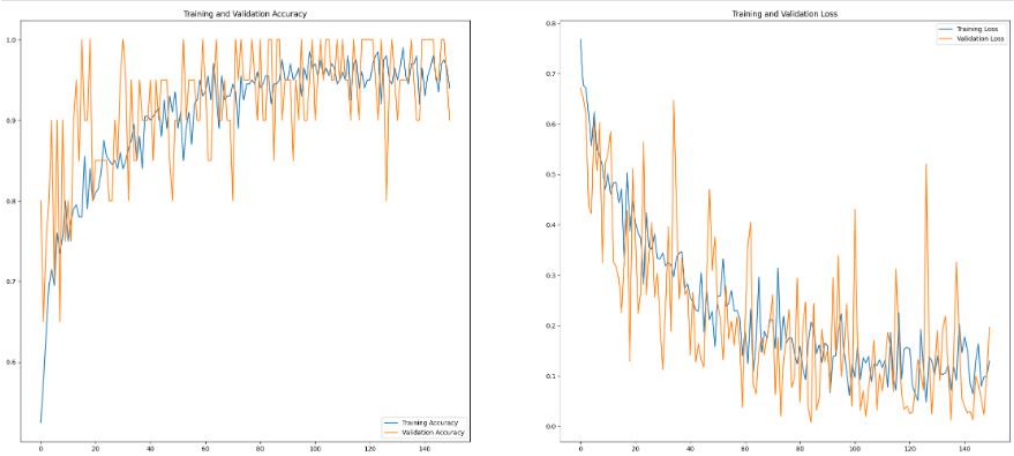
Gambar 12. *Model Convolutional Neural Network*

Untuk model *optimizer* gambar 11. Digunakan "*binary_crossentropy*" karena hanya terdapat dua klasifikasi, berbeda dengan klasifikasi lebih dari dua (*multiple*) yang biasanya menggunakan *adamax. Learning_rate* dengan nilai 0.001.

```
acc = history.history['accuracy']
val_acc = history.history['val_accuracy']

loss = history.history['loss']
val_loss = history.history['val_loss']

epochs_range = range(150)

plt.figure(figsize=(28, 12))
plt.subplot(1, 2, 1)
plt.plot(epochs_range, acc, label='Training Accuracy')
plt.plot(epochs_range, val_acc, label='Validation Accuracy')
plt.legend(loc='lower right')
plt.title('Training and Validation Accuracy')

plt.subplot(1, 2, 2)
plt.plot(epochs_range, loss, label='Training Loss')
plt.plot(epochs_range, val_loss, label='Validation Loss')
plt.legend(loc='upper right')
plt.title('Training and Validation Loss')
plt.show()
```

Gambar 13. *Plot Code*

JURNAL TEKNOINFO
Volume 18, Nomor 1, Januari 2024, Page 121-132
ISSN: 1693-0010(Print), ISSN: 2615-224X(Online)
Available online at https://ejurnal.teknokrat.ac.id/index.php/teknoinfo/index



Gambar 14. *Testing Plot*

Kemudian (gambar 12 dan gambar 13), merupakan tampilan plot penyebaran yang dapat diamati. Dimana nilai dari akurasi latih dan validasi akurasi terlihat naik ke angka satu, hal ini menunjukan hasil dari iterasi sudah cukup baik. Begitu juga dengan nilai dari kegagalan latih dan validasi terlihat menurun hal ini menunjukan data yang di latih sudah cukup baik.

```
import keras
IMG_SIZE = 512

Pred_Path = "Testing/diabetes/10233_left.jpeg"
Pred_result_Path = "Prediksi.png"

img = cv2.imread(Pred_Path)
img_process = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
img_process = cv2.resize(img_process, (IMG_SIZE, IMG_SIZE))
img_process = cv2.addWeighted(img_process,4, cv2.GaussianBlur(img_process, (0,0) , 40) , -4 ,128)
cv2.imwrite(Pred_result_Path, img_process)

import keras
img_plot = tf.keras.utils.load_img(Pred_Path, target_size=(150, 150))
img_pred = tf.keras.utils.load_img(Pred_result_Path, target_size=(150, 150))
img_array = tf.keras.utils.img_to_array(img_pred)
img_array = tf.expand_dims(img_array, 0)

predictions = model.predict(img_array)
score = tf.nn.sigmoid(predictions[0])
imgplot = plt.imshow(img_plot)

print(
    "This image most likely belongs to {} with a {:.2f} percent confidence."
    .format(class_names[np.argmax(score)], 100 * np.max(score))
)
```

Gambar 15. *Testing Code*

Selanjutnya adalah percobaan prediksi (gambar 14), gambar yang digunakan adalah gambar yang ada di dalam folder uji coba, dengan gambar penderita diabetes. Proses ini menggunakan module cv2 dari *opencv* dan juga keras dari *tensorflow.*

JURNAL TEKNOINFO
Volume 18, Nomor 1, Januari 2024, Page 121-132
ISSN: 1693-0010(Print), ISSN: 2615-224X(Online)
Available online at https://ejurnal.teknokrat.ac.id/index.php/teknoinfo/index

Gambar 16. *Output Testing*

Kemudian hasil dari pengujian tersebut pada gambar 15, dimana sistem menjawab bahwa data tersebut benar diabetes dengan *confidence* 73%.

```
Testing_datagen = ImageDataGenerator(rescale=1./255)

test_generator = Testing_datagen.flow_from_directory(
    testing_dir,
    target_size=(150, 150),
    batch_size=4,
    class_mode='categorical')

Found 319 images belonging to 2 classes.

test_accu = model.evaluate(test_generator)
print('The testing accuracy is :',test_accu[1]*100, '%')

80/80 [==============================] - 10s 119ms/step - loss: 0.0923 - accuracy: 0.9655
The testing accuracy is : 96.55172228813171 %

model.save('GUI/ModelML/ModelDR.h5')

new_model = tf.keras.models.load_model('GUI/ModelML/ModelDR.h5')

new_model.summary()
```

Gambar 17. *Testing Accuracy*

Setelah itu, dilakukan pelatihan data akurasi dengan data acak, dan menghasilkan persentase 96%. Hal ini cukup baik karena sistem sudah mampu mencapai akurasi tinggi diatas 90%. Setelah melakukan proses pelatihan data maka data tersebut akan disimpan dengan nama modelDR5.

Tabel 2. *Layer convolutional neural network*

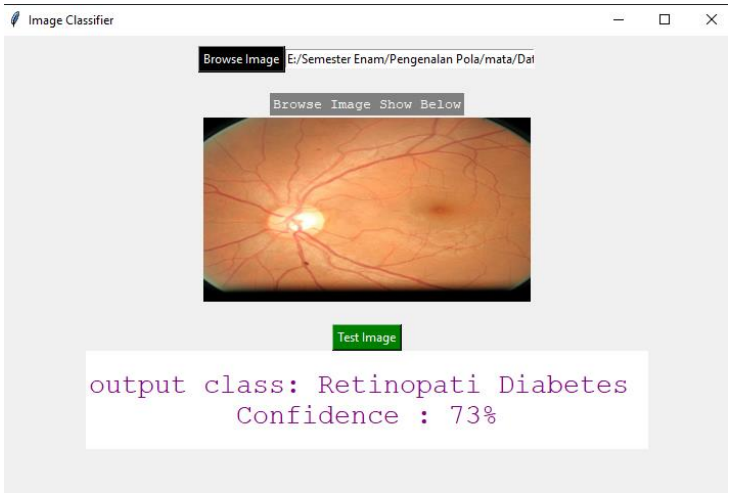
Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 148, 148, 64)	1792
max_pooling2d (MaxPooling2D)	(None, 74, 74, 64)	0
conv2d_1 (Conv2D)	(None, 72, 72, 128)	73856
max_pooling2d_1 (MaxPooling2D)	(None, 36, 36, 128)	0
conv2d_2 (Conv2D)	(None, 34, 34, 256)	295168
max_pooling2d_2 (MaxPooling2D)	(None, 17, 17, 256)	0
flatten (Flatten)	(None, 73984)	0
dense (Dense)	(None, 512)	37880320
dense_1 (Dense)	(None, 512)	262656
dense_2 (Dense)	(None, 2)	1026
	Total params: 38,514,818	
	Trainable params: 38,514,818	
	Non-trainable params: 0	

Kemudian pada tabel 2 menunjukan tampilan layer yang sama seperti pada tabel sebelumnya namun hal in perlu dilakukan karena akan menyimpan model pelatihan baru. Tujuan dari penambahan layer pada Convolutional Neural Network (CNN) adalah untuk meningkatkan kemampuan model dalam mengekstraksi fitur yang relevan dari data citra. Layer-layer ini dirancang untuk mengenali pola-pola visual yang berbeda dalam citra dan memperoleh representasi yang semakin abstrak dan diskriminatif seiring dengan kedalaman arsitektur model. Pemberian layer pada CNN bertujuan untuk memperoleh representasi fitur yang lebih mendalam dan kompleks dari citra, sehingga memungkinkan model untuk mempelajari karakteristik yang lebih spesifik dari objek atau pola yang ada dalam citra.

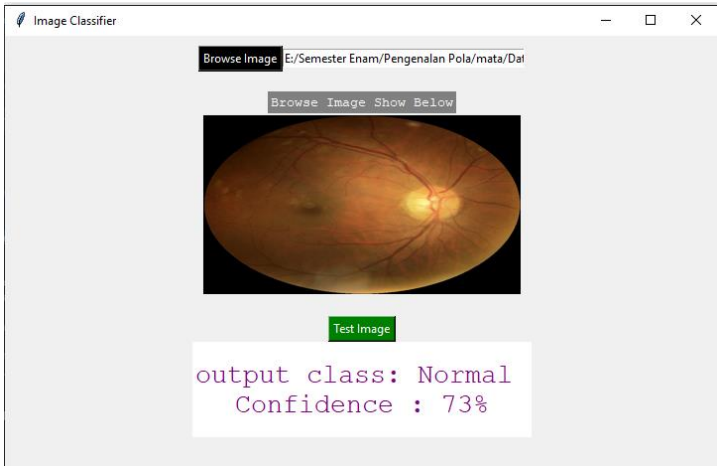
3.3 Program *GUI*

Implementasi akhir dari program ini dibuat dalam bentuk *GUI*, agar memudahkan user untuk melakukan uji coba pada sistem yang telah di bangun. Program *GUI* ini dibangun dengan bahasa pemrograman *python* dan anaconda spyder sebagai media pengolahan kode program, serta penggunaan library *tkinter* sebagai alat pengembangan aplikasi *GUI*, hal ini dilakukan karena tkinter mudah untuk di gunakan.

JURNAL TEKNOINFO
Volume 18, Nomor 1, Januari 2024, Page 121-132
ISSN: 1693-0010(Print), ISSN: 2615-224X(Online)
Available online at https://ejurnal.teknokrat.ac.id/index.php/teknoinfo/index



Gambar 18. *GUI Diabetic Retinopathy*



Gambar 18. *GUI Normal*

4. KESIMPULAN

Klasifikasi data gambar menggunakan algoritma *Convolutional Neural Network* (*CNN*) menjadi salah satu tantangan yang kompleks dalam pengolahan data gambar/citra. *Convolutional Neural Network* (*CNN*) memang sudah mencapai hasil yang cukup mengagumkan dalam aplikasi pengolahan gambar/citra, namun masih terdapat beberapa resiko masalah yang dapat mempengaruhi kinerja serta akurasi klasifikasi. Kesimpulan ini akan membahas masalah-masalah yang umum dihadapi saat melakukan implementasi algoritma *Convolutional Neural Network* (*CNN*) terhadap data gambar. Masalah umum yang pertama adalah adanya keterbatasan data yang sesuai atau relevan. Untuk implementasi *Convolutional Neural Network* (*CNN*) diperlukan sejumlah dataset yang besar, beragam, dan representatif agar dapat mempelajari dan menghasilkan pola dan fitur yang tepat pada gambar/citra. *Overfitting* adalah masalah yang sering terjadi apabila jumlah data yang dimiliki cenderung terbatas. Dimana, model tidak dapat melakukan generalisasi dengan baik karena data dianggap asing oleh sistem, terutama pada data-data yang baru dilihat. Ada beberapa cara agar kesalahan ini tidak terjadi salah satunya dengan augmentasi data. Augmentasi data berfungsi untuk menghasilkan variasi tambahan dalam dataset pelatihan yang ada. Masalah berikutnya adalah ukuran dan kompleksitas gambar/citra. Gambar/citra yang memiliki banyak objek atau detail kompleks atau resolusi tinggi dapat mempengaruhi kinerja *Convolutional Neural Network* (*CNN*). Komputasi dan pelatihan akan memakan waktu lebih lama apabila gambar memiliki ukuran yang besar. Gambar/citra yang kompleks juga akan mempersulit model untuk menemukan fitur yang tepat dan relevan, namun untuk mengatasi masalah tersebut dapat dilakukan dengan menerapkan ekstraksi fitur sebelum data gambar/citra dimasukkan

JURNAL TEKNOINFO
Volume 18, Nomor 1, Januari 2024, Page 121-132
ISSN: 1693-0010(Print), ISSN: 2615-224X(Online)
Available online at https://ejurnal.teknokrat.ac.id/index.php/teknoinfo/index

kedalam model *Convolutional Neural Network* (*CNN*). Selain itu, ketergantungan pada data pelatihan juga dapat menyebabkan masalah. Jika dataset pelatihan tidak mencakup secara representatif populasi yang ingin diklasifikasikan, model CNN dapat menghasilkan prediksi yang tidak akurat atau bias. Misalnya, jika data pelatihan hanya mencakup objek dalam kondisi pencahayaan atau sudut pandang tertentu, sementara data pengujian memiliki variasi yang lebih luas, maka performa model akan menurun. Untuk mengatasi masalah ini, penting untuk memiliki dataset pelatihan yang representatif dan melakukan validasi lintas-validasi silang untuk menguji kinerja model pada data yang belum pernah diterima sebelumnya. Penelitian ini menggunakan data yang bersumber dari *Kaggle*, lalu kemudian dilakukan pengolahan yang terdiri dari 802 file gambar mata normal dan 796 file gambar mata penderita retinopati diabetes. 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. Dimana 20% data tersebut akan digunakan untuk data uji coba dan untuk 80% data gambar tersebut akan dipakai untuk data latih. Adapun hasil dari latihan data tersebut diperoleh hasil yang cukup memuaskan, dimana diperoleh data akurasi prediksi 96%. Harapan peneliti kepada penelitian selanjutnya adalah dapat menerapkan aplikasi *Convolutional Neural Network* (*CNN*) dengan model yang lebih baik lagi dan dengan jumlah data yang lebih beragam serta penggunaan layer yang lebih variatif.

DAFTAR PUSTAKA

[1] R. Indraswari, W. Herulambang, and R. Rokhana, “Deteksi Penyakit Mata Pada Citra Fundus Menggunakan Convolutional Neural Network (CNN) Ocular Disease Detection on Fundus Images Using Convolutional Neural Network (CNN).” [Online]. Available: https://www.kaggle.com/datasets/jr2ngb/cataractdataset\

[2] Saputra, Rizal Adi, Yuwanda Purnamasari Pasrun, and Amaliya Nurani Basyarah. "Macular Edema Classification Using Self-Organizing Map and Generalized Learning Vector Quantization." *Jurnal Ilmu Komputer dan Informasi* 7.2 (2014): 54-60.

[3] 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, p. 693, Aug. 2020, doi: 10.26760/elkomika.v8i3.693.

[4] S. Frangky Handono, F. Tri Anggraeny, and B. Rahmat, “IMPLEMENTASI CONVOLUTIONAL NEURAL NETWORK (CNN) UNTUK DETEKSI RETINOPATI DIABETIK,” 2020.

[5] C. Lubis, “Jurnal Ilmu Komputer dan Sistem Informasi KLASIFIKASI PENYAKIT MATA MENGGUNAKAN CNN.” [Online]. Available: https://www.kaggle.com/nafizimtiazkhan/cataract-

[6] S. H. Abdullah, R. Magdalena, and R. Y. N. Fu’adah, “KLASIFIKASI DIABETIC RETINOPATHY BERBASIS PENGOLAHAN CITRA FUNDUS DAN DEEP LEARNING,” *JOURNAL OF ELECTRICAL AND SYSTEM CONTROL ENGINEERING*, vol. 5, no. 2, pp. 84–90, Feb. 2022, doi: 10.31289/jesce.v5i2.5659.

[7] E. Hari Rachmawanto, “Convolutional Neural Network (CNN) untuk Klasifikasi Citra Penyakit Diabetes Retinopathy,” *SKANIKA: Sistem Komputer dan Teknik Informatika*, vol. 5, no. 2, pp. 167–176, 2022.

[8] “Algoritma Kohonen dalam Mengubah Citra Graylevel Menjadi Citra Biner.”

[9] N. Fadlia and R. Kosasih, “KLASIFIKASI JENIS KENDARAAN MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK (CNN),” *Jurnal Ilmiah Teknologi dan Rekayasa*, vol. 24, no. 3, pp. 207–215, 2019, doi: 10.35760/tr.2019.v24i3.2397.

[10] N. Fadlia and R. Kosasih, “KLASIFIKASI JENIS KENDARAAN MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK (CNN),” *Jurnal Ilmiah Teknologi dan Rekayasa*, vol. 24, no. 3, pp. 207–215, 2019, doi: 10.35760/tr.2019.v24i3.2397.

[11] K. Hasan Mahmud and S. Al Faraby, “Klasifikasi Citra Multi-Kelas Menggunakan Convolutional Neural Network.”

[12] K. Tingkat Keparahan Retinopati Diabetik Berdasarkan Citra Fundus Menggunakan, M. A. Hendriyawan, and W. R. Saputro, “Diabetic Retinopathy Severity Level Classification Based on Fundus Image Using Convolutional Neural Network (CNN),” pp. 13–2021.

[13] A. A. N. G. Sapteka, P. A. S. Dharma, K. A. Widyatmika, I. N. Suparta, I. M. S. Yasa, and A. A. N. G. Sapteka, “Pendeteksi Penggunaan Masker Wajah dengan ESP32Cam Menggunakan OpenCV dan Tensorflow,” *Majalah Ilmiah Teknologi Elektro*, vol. 21, no. 2, p. 155, Dec. 2022, doi: 10.24843/mite.2022.v21i02.p01.

[14] 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, p. 359, Apr. 2021, doi: 10.26760/elkomika.v9i2.359.

[15] M. Malik, “Deteksi Suhu Tubuh dan Masker Wajah dengan MLX90614, Opencv, Keras/Tensorflow, dan Deep Learning,” vol. 6, no. 1, 2022.

[16] 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, Aug. 2020, doi: 10.33489/mibj.v18i2.220.