Volume 2 No 1 Maret 2022
e-ISSN 0000-0000
DOI: http://dx.doi.org/10.53514/jco.v4i1.483

Penerapan Deteksi Pada Objek Kendaraan Motor Dan Mobil Menggunakan Metode Convolution Neural Network Berbasis Python

Alfin Falakhi

Universitas Mercu Buana

*Corresponding author email: *alfinplv@gmail.com*

ABSTRAK

Dengan memantau dan mengumpulkan data volume kendaraan di jalan raya, kita dapat mengendalikan penetapan lalu lintas. Majoritas proses pendataan volume kendaraan yang melewati jalan raya masih dilakukan secara manual. Inilah berarti bahwa beberapa individu disebutkan untuk berada di lapangan dan mencatat atau menghitung semua mobil dan kendaraan motor yang lewat. Pengiraan penghitungan manual memiliki banyak kekurangan, seperti pengumpulan data yang lama dan memerlukan banyak sumber daya manusia. Untuk menyederhanakan pengendalian lalu lintas, diperlukan sistem penghitung dan deteksi kendaraan otomatis yang tepat, baik mobil maupun motor, berdasarkan ketidakpastian tersebut. Sebuah sistem deteksi kendaraan motor dan mobil yang menggunakan metode Convolutional Neural Network berbasis Python sedang dibangun

***Kata Kunci**: Data Volume Kendaraan, Deteksi Kendaraan, Kepadatan Lalu Lintas, Neural Network*

I. Pendahuluan

Kita dapat mengatur kepadatan lalu lintas dengan mengelola dan mengumpulkan data jumlah mobil di jalan raya. Data volume kendaraan dapat digunakan untuk menghitung kepadatan kendaraan, memprediksi kemacetan, dan menjadi acuan perbaikan jalan seperti pelebaran jalan, penambahan jalan baru atau penataan jalur baru, penataan rambu lalu lintas, dan peningkatan infrastruktur. Pendataan volume kendaraan yang melalui jalan raya mayoritas masih dilakukan secara manual. Artinya, satu tim pegawai bertugas mencatat atau menghitung semua mobil dan kendaraan bermotor yang melewati jalan tersebut. Saat ini, kemajuan teknis tidak hanya terjadi di bidang teknologi, tetapi juga dalam komputerisasi; misalnya, visi komputer berkembang pesat. Salah satu masalah dengan komputer vision adalah pengenalan objek pada kendaraan bermotor dan pengenalan kendaraan/mobil. Identifikasi kendaraan dengan pelatihan mendalam adalah metode yang sedang dikembangkan untuk mereplikasi kemampuan manusia untuk menginterpretasikan informasi dari foto, memungkinkan komputer untuk mengenali hal-hal dalam foto dengan cara yang sama seperti yang dilakukan manusia.

Deep Learning, dengan keterampilannya yang luar bisa dalam visi komputer, dapat memodelkan banyak bentuk data yang rumit, seperti data gambar. Convolutional Neural Network (CNN) adalah salah satu pendekatan pembelajaran deep learning yang paling efektif untuk identifikasi gambar saat ini. Namun, CNN memiliki kekurangan: prosedur pelatihan model yang panjang. CNN dirancang untuk pembelajaran mendalam, atau pemrosesan gambar. CNN mampu mengatasi masalah yang menantang dan mengungguli penelitian terkait, menjadikannya pilihan yang menarik untuk aplikasi CNN. CNN adalah jaringan saraf tiruan arsitektur berbobot dengan lapisan tersembunyi.

Kemampuan klasifikasi CNN sangat bermanfaat untuk pengolahan data gambar. Metode pengajaran supervisi memungkinkan penggunaan Convolutional Neural Network untuk klasifikasi data yang sudah terlabel. Tujuan dari metode pengajaran supervisi adalah untuk mengelompokkan data ke dalam data yang sudah ada karena data yang akan dilatih sudah ada dan variabel yang telahita kita targetkan. Selain melakukan deteksi dan segmentasi objek, CNN sering digunakan untuk mengenali benda atau pemandangan

 DOI: http://dx.doi.org/10.53514/jco.v4i1.483

II. Landasan Teori

A. Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) adalah jenis algoritma deep learning yang digunakan untuk memproses data gambar dan video dengan memanfaatkan konsep konvolusi. CNN terdiri dari beberapa layer, di antaranya adalah convolutional layer, pooling layer, dan fully connected layer. Berikut adalah rumus atau formula yang digunakan dalam CNN:
Convolutional Layer Output = (Input – Filter + 2 * Padding) / Stride + 1
Input: ukuran input gambar atau feature map
Filter: ukuran filter atau kernel yang digunakan
Padding: jumlah piksel yang ditambahkan pada tepi gambar
Stride: jumlah piksel yang digeser pada setiap operasi konvolusi Pooling Layer Output = (Input – Pooling Size) / Stride + 1
Input: ukuran input gambar atau feature map
Pooling Size: ukuran pooling window
Stride: jumlah piksel yang digeser pada setiap operasi pooling
Fully Connected Layer Output = Input x Weight + Bias
Input: vektor input dari layer sebelumnya
Weight: matriks bobot yang menghubungkan neuron pada layer sebelumnya dengan neuron pada layer saat ini
Bias: vektor bias yang ditambahkan pada setiap neuron pada layer saat ini
CNN digunakan untuk melakukan klasifikasi, deteksi objek, segmentasi, dan pengenalan wajah pada gambar. CNN telah digunakan dalam berbagai aplikasi, seperti deteksi masker wajah dan deteksi penyakit kulit wajah.

B. Komputer Vision

Komputer vision adalah cabang ilmu komputer yang mempelajari cara komputer untuk memahami gambar dan video. Pengetahuan tentang komputer vision diperlukan untuk memahami bagaimana sistem deteksi gambar bekerja.

C. Jaringan saraf tiruan

Jaringan saraf tiruan adalah model matematika yang diilhami oleh otak manusia. Pengetahuan tentang jaringan saraf tiruan diperlukan untuk memahami bagaimana metode Convolutional Neural Network bekerja.

D. Python

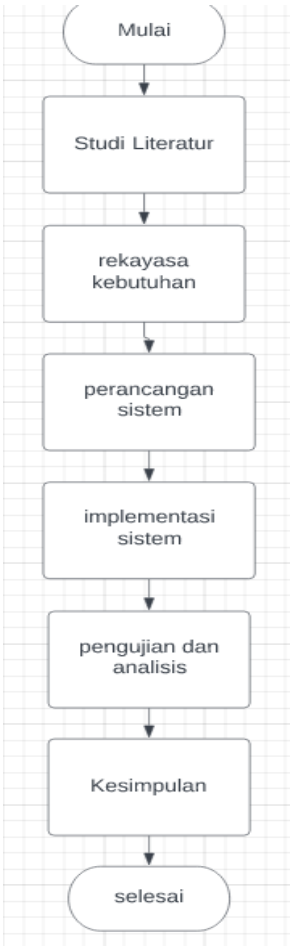
Python adalah bahasa pemrograman yang populer digunakan untuk pengembangan perangkat lunak. Pengetahuan tentang Python diperlukan untuk mengembangkan sistem deteksi gambar.Metode Penelitian.

III. Metode Penelitian

A. Tahapan Penelitian

Proses penelitian adalah langkah-langkah yang dilakukan selama penelitian. Penelitian system deteksi kendaraan motor dan mobil ini dimulai dengan penelitian literatur. Setelah itu, Perancangan termasuk Data

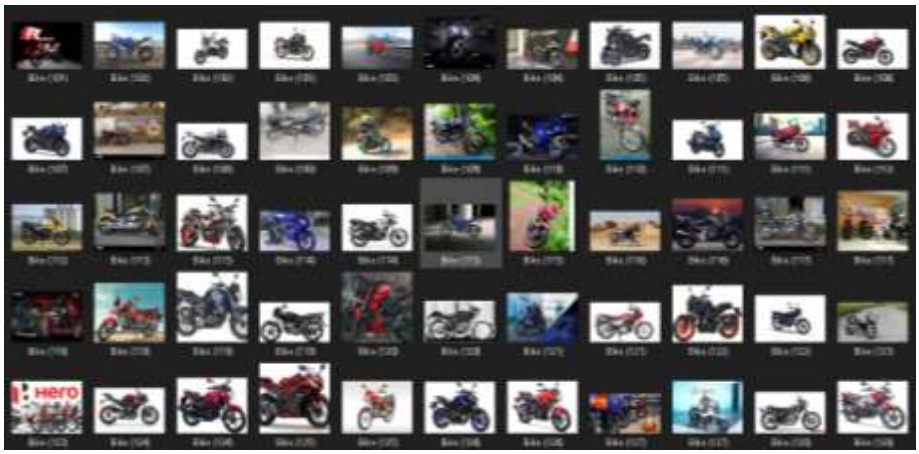
DOI: http://dx.doi.org/10.53514/jco.v4i1.483



B. Data

Dataset gambar kendaraan digunakan dalam berbagai konteks, termasuk untuk melatih kendaraan otonom, deteksi objek, dan pengenalan plat nomor kendaraan. Selain itu, penelitian tentang deteksi dan pengenalan jenis kendaraan menggunakan metode Convolutional Neural Network (CNN) juga telah dilakukan. Dataset gambar kendaraan dapat digunakan untuk berbagai tujuan, termasuk pengembangan dan pelatihan model AI terkait kendaraan. Data yang digunakan sebanyak 2000 gambar motor dan 2000 gambar mobil.

C. Gambar



Gambar 1 Data Gambar Motor

DOI: http://dx.doi.org/10.53514/jco.v4i1.483

Gambar 2 Data Gambar Mobil

IV. Hasil Dan Pembahasan

A. Hasil

Simulasi dijalankan pada Web GOOGLE COLLAB dengan menggunakan Bahasa Python berdasarkan metode CNN. Untuk perancangan perangkat lunak dapat di lihat pada gambar di bawah :

```
!pip install -q kaggle
!mkdir ~/.kaggle
!cp 'kaggle.json' ~/.kaggle
!chmod 600 ~/.kaggle/'kaggle.json'
!kaggle datasets download -d 'utkarshsaxenadn/car-vs-bike-classification-dataset'
```

```
[ ] import zipfile
    import os
    import shutil
    import tensorflow as tf

    from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

```
[ ] dataset_zip = zipfile.ZipFile('car-vs-bike-classification-dataset.zip', 'r')
    dataset_zip.extractall()
    dataset_zip.close()
```

```
[ ] dataset_dir = 'dataset'

    train_dir = os.path.join(dataset_dir, 'train') # 'dataset/train'
    val_dir = os.path.join(dataset_dir, 'val') # 'dataset/train'

    car_train_dir = os.path.join(train_dir, 'car')
```

Gambar 3 Coding perancangan perangkat lunak

 DOI: http://dx.doi.org/10.53514/jco.v4i1.483

```
[ ]  dataset_dir = 'dataset'

     train_dir = os.path.join(dataset_dir, 'train') # 'dataset/train'
     val_dir = os.path.join(dataset_dir, 'val') # 'dataset/train'

     car_train_dir = os.path.join(train_dir, 'car')
     bike_train_dir = os.path.join(train_dir, 'bike')

     car_val_dir = os.path.join(val_dir, 'car')
     bike_val_dir = os.path.join(val_dir, 'bike')
```

```
[ ]  os.mkdir(dataset_dir)
     os.mkdir(train_dir)
     os.mkdir(val_dir)
     os.mkdir(car_train_dir)
     os.mkdir(bike_train_dir)
     os.mkdir(car_val_dir)
     os.mkdir(bike_val_dir)
```

```
[ ]  training_portion = 0.8
     car_train_length = training_portion * len(os.listdir('Car-Bike-Dataset/Car'))
     bike_train_length = training_portion * len(os.listdir('Car-Bike-Dataset/Bike'))
```

Gambar 4 Coding perancangan perangkat lunak

```
[ ]  fileCounter = 0
     for file in os.listdir('Car-Bike-Dataset/Car'):

       src = os.path.join('Car-Bike-Dataset/Car', file) #Car-Bike-Dataset/Car/Car (217).jpeg
       if fileCounter <= car_train_length:
         dest = car_train_dir
       else:
         dest = car_val_dir

       shutil.move(src, dest)
       fileCounter += 1
       print("Move {} from {} to {}".format(file, src, dest))

     print("BANYAK DATA CAR UNTUK TRAINING: {} DATA".format(len(os.listdir(car_train_dir))))
     print("BANYAK DATA CAR UNTUK VALIDATION: {} DATA".format(len(os.listdir(car_val_dir))))
```

```
     fileCounter = 0
     for file in os.listdir('Car-Bike-Dataset/Bike'):

       src = os.path.join('Car-Bike-Dataset/Bike', file)
       if fileCounter <= bike_train_length:
         dest = bike_train_dir
       else:
         dest = bike_val_dir
```

Gambar 5 Coding perancangan perangkat lunak

 DOI: http://dx.doi.org/10.53514/jco.v4i1.483

```
[ ] training = ImageDataGenerator(
        rescale = 1./255,
        horizontal_flip = True,
        vertical_flip = True
    )

    validation = ImageDataGenerator(
        rescale = 1./255
    )

    train_generator = training.flow_from_directory(
        train_dir,
        target_size = (150, 150),
        batch_size = 20,
        class_mode = 'binary',
        shuffle = True
    )

    val_generator = training.flow_from_directory(
        val_dir,
        target_size = (150, 150),
        batch_size = 20,
        class_mode = 'binary',
        shuffle = True
    )
```

Gambar 6 Coding perancangan perangkat lunak

```
[ ]     target_size = (150, 150),
        batch_size = 20,
        class_mode = 'binary',
        shuffle = True
    )

[ ] train_generator.class_indices

[ ] model = tf.keras.models.Sequential([
        tf.keras.layers.Conv2D(8, (3, 3), activation = 'relu', input_shape = (150, 150, 3)),
        tf.keras.layers.MaxPooling2D(2, 2),

        # tf.keras.layers.Conv2D(8, (3, 3), activation = 'relu'),
        # tf.keras.layers.MaxPooling2D(2, 2),

        tf.keras.layers.Flatten(),

        tf.keras.layers.Dense(8, activation = 'relu'),
        tf.keras.layers.Dense(1, activation = 'sigmoid')
    ])

    model.summary()

[ ] model.compile(
```

Gambar 7 Coding perancangan perangkat lunak

 DOI: http://dx.doi.org/10.53514/jco.v4i1.483

```
model.fit(
    train_generator,
    validation_data = val_generator,
    epochs = 5
)
```

```
import PIL
from PIL import Image

for file in os.listdir(car_train_dir):
  img = Image.open(os.path.join(car_train_dir, file))

  if img.format == 'PNG' and img.mode != 'RGBA':
    img.convert('RGBA').save(os.path.join(car_train_dir, file))
```

```
for file in os.listdir(bike_train_dir):
  img = Image.open(os.path.join(bike_train_dir, file))

  if img.format == 'PNG' and img.mode != 'RGBA':
    img.convert('RGBA').save(os.path.join(bike_train_dir, file))
```

```
for file in os.listdir(car_val_dir):
  img = Image.open(os.path.join(car_val_dir, file))
```

Gambar 8 Coding perancangan perangkat lunak

```
img = Image.open(os.path.join(bike_val_dir, file))

if img.format == 'PNG' and img.mode != 'RGBA':
  img.convert('RGBA').save(os.path.join(bike_val_dir, file))
```

```
model.save('model.h5')
```

```
my_model = tf.keras.models.load_model('model.h5')
```

```
my_model.summary()
```

```
my_model.get_config()
# INPUT SHAPE -> (None, 150, 150, 3),
```

```
image_file = 'imagemotor.jfif'
image = tf.keras.utils.load_img(image_file, target_size = (150, 150))
```

```
image_array = tf.keras.utils.img_to_array(image)
```

```
image_array.shape
```

```
image_array = image_array / 255.
```

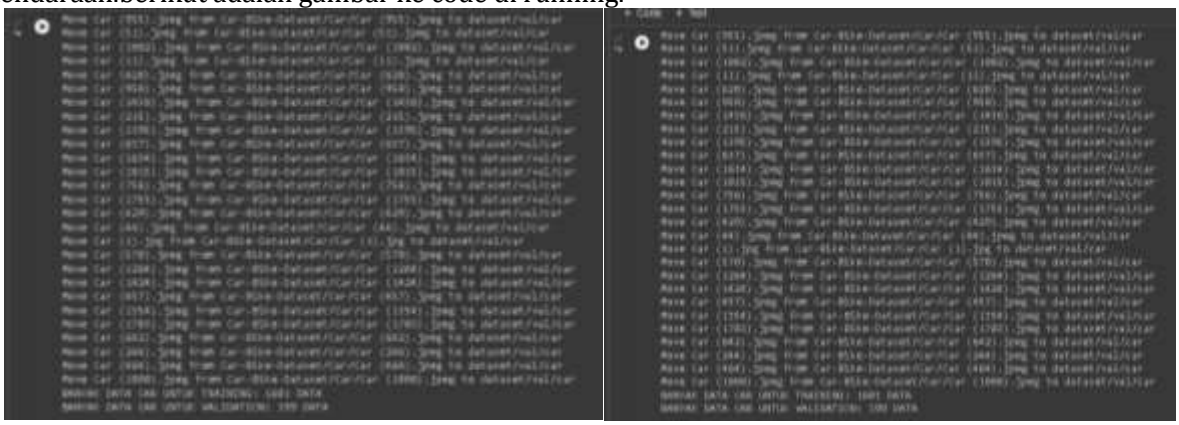
Gambar 9 Coding perancangan perangkat lunak

 DOI: http://dx.doi.org/10.53514/jco.v4i1.483

```
[ ]  image_array.shape

[ ]  image_array = image_array / 255.

[ ]  import numpy as np
     image_array = np.array([image_array])

[ ]  image_array.shape

[ ]  predict_result = my_model.predict(image_array)

[ ]  predict_result = predict_result[0][0]

     if predict_result < 0.5:
       print("MOTOR")
     else:
       print("MOBIL")
```

Gambar 10 Coding perancangan perangkat lunak

Jumlah data train adalah 1601 dan jumlah data validation adalah 399 untuk kedua buah kendaraan.berikut adalah gambar ke code di running.



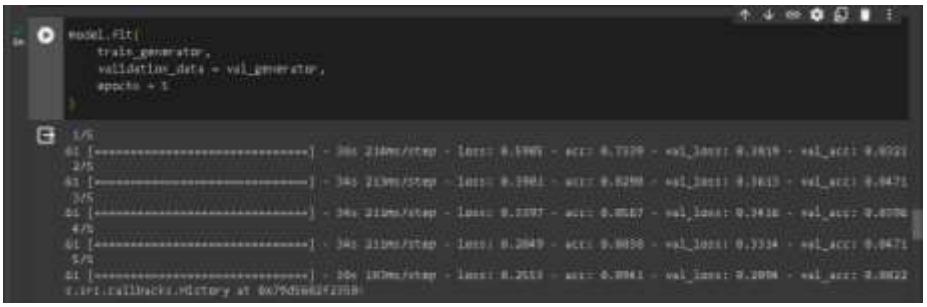
Gambar 11 Data train dan validation

```
model = tf.keras.models.Sequential([
    tf.keras.layers.Conv2D(8, (3, 3), activation = 'relu', input_shape = (150, 150, 3)),
    tf.keras.layers.MaxPooling2D(2, 2),

    # tf.keras.layers.Conv2D(8, (3, 3), activation = 'relu'),
    # tf.keras.layers.MaxPooling2D(2, 2),

    tf.keras.layers.Flatten(),

    tf.keras.layers.Dense(8, activation = 'relu'),
    tf.keras.layers.Dense(1, activation = 'sigmoid')
])
```

Gambar 12 Model convolution

Berdasarkan hasil penelitian yang telah dilakukan menunjukan akurasi saat train dan pada saat validation beda tipis,ini artinya bagus.yaitu pada data train akurasinya mencapai 0,8941 atau 89% dan pada validation akurasinya mencapai 0.8822 atau 89%. berikut dapat dilihat pada gambar dibawah ini.

 DOI: http://dx.doi.org/10.53514/jco.v4i1.483



Gambar 13 Akurasi

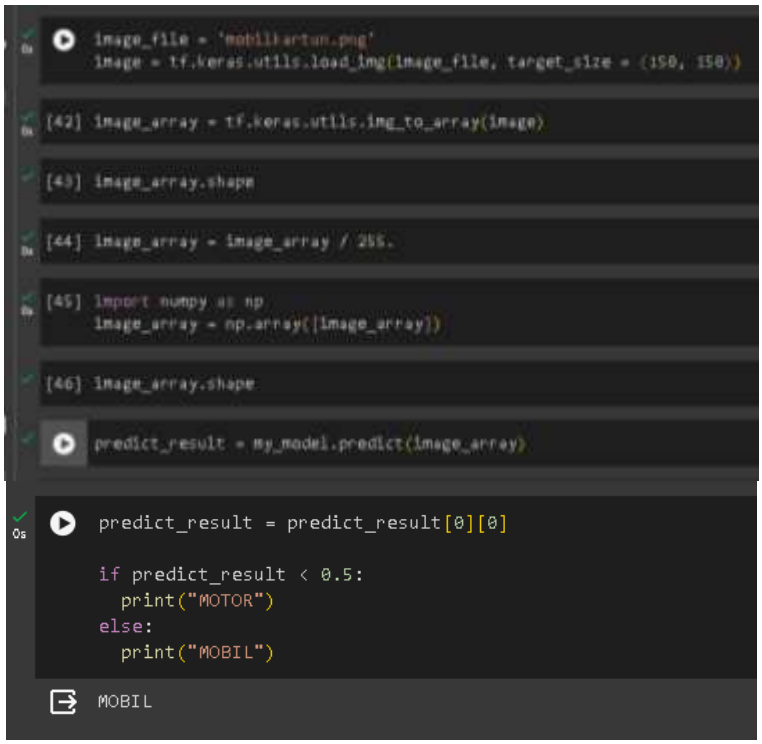
Berikut kami akan menyimpan model untuk kami coba memprediksi gambar secara langsung.

Gambar 14 Save model

Kami akan mencoba dengan memasukan gambar mobil yang tidak biasa yaitu kita akan menggunakan gambar mobil kartun.bisa dilihat pada gambar dibawah ini.

Gambar 15 Mobilkartun.png

 DOI: http://dx.doi.org/10.53514/jco.v4i1.483



Gambar 16 Hasil Deteksi.

B. Pembahasan

Berdasarkan hasil penelitian yang telah dilakukan menunjukan akurasi saat train dan pada saat validation beda tipis,ini artinya bagus.yaitu pada data train akurasinya mencapai 0,8941 atau 89% dan pada validation akurasinya mencapai 0.8822 atau 89%. Artinya akurasi nya sudah bagus karena akurasi pada saat Latihan dan pada saat validasi sama sehingga system dapat dikatakan berhasil.hal tersebut dapat dibuktikan Ketika kami mncoba memasukan mobil kartun yang tentunya berbeda dengan mobil asli akan tetapi system tetap bisa mendeteksi bahwa itu adalah mobil.

V. Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, dapat disimpulkan bahwa sistem deteksi gambar kendaraan motor dan mobil menggunakan metode Convolutional Neural Network berbasis Python dapat bekerja dengan baik. Berdasarkan hasil penelitian yang telah dilakukan menunjukan akurasi saat train dan pada saat validation beda tipis,ini artinya bagus.yaitu pada data train akurasinya mencapai 0,8941 atau 89% dan pada validation akurasinya mencapai 0.8822 atau 89%.
Sistem ini memiliki potensi untuk digunakan dalam berbagai aplikasi, seperti keamanan lalu lintas, pemantauan lalu lintas, otomatisasi parkir, dan asisten pengemudi. Sistem ini dapat membantu meningkatkan keselamatan lalu lintas, efisiensi lalu lintas, dan kenyamanan berkendara

Daftar Rujukan

Prasmatio, R. M. (1970, January 1). *Deteksi Dan pengenalan ikan dengan menggunakan algoritma convolutional neural network*. Welcome to UPN Veteran Jatim Repository - UPN Veteran Jatim Repository. http://repository.upnjatim.ac.id/id/eprint/1732

Prastika, I. W., & Zuliarso, E. (n.d.). *Deteksi penyakit kulit wajah menggunakan tensorflow dengan metode convolutional neural network*. Jurnal Manajemen Informatika dan Sistem Informasi. https://doi.org/10.36595/misi.v4i2.418

 DOI: http://dx.doi.org/10.53514/jco.v4i1.483

Fadlia, N., & Kosasih, R. (2019). Klasifikasi Jenis Kendaraan Menggunakan metode Convolutional Neural Network (CNN). *Jurnal Ilmiah Teknologi Dan Rekayasa*, *24*(3), 207–215. https://doi.org/10.35760/tr.2019.v24i3.2397

Rochmawanti, O., Utaminingrum, F., & Bachtiar, F. A. (2021). Analisis performa pre-trained model convolutional neural network Dalam Mendeteksi Penyakit tuberkulosis. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, *8*(4), 805–814. https://doi.org/10.25126/jtiik.2021844441

Saputra, V. P., Latifa, U., & Ibrahim, I. (2023, August 20). *Simulasi Detection Counter Pada objek Kendaraan motor dan Mobil Menggunakan metode convolutional neural network Berbasis Python*. Zenodo. https://doi.org/10.5281/zenodo.8265040