

Implementasi Metode Convolutional Neural Network untuk Pengenalan Plat Nomor Kendaraan

*Veran Goraldy Cosmas¹⁾, Wahyuni²⁾, dan Pitrasacha Adytia³⁾

^{1,2,3}Teknik Informatika, STMIK Widya Cipta Dharma
^{1,2,3}Jl. M. Yamin, Gn. Kelua, Kec. Samarinda Ulu, Kota Samarinda, Kalimantan Timur 75123
Email: 2041007@wicida.ac.id¹⁾, pitra@wicida.ac.id

ABSTRAK

Penelitian ini bertujuan untuk mengembangkan sistem pengenalan plat nomor kendaraan berbasis teknologi kecerdasan buatan guna mendukung kebutuhan sistem transportasi cerdas, seperti e-parking dan identifikasi kendaraan. Tantangan utama dalam penelitian ini meliputi variasi kondisi pencahayaan, sudut pengambilan gambar, desain plat nomor, serta kerusakan fisik pada plat. Untuk mengatasi tantangan tersebut, penelitian ini mengimplementasikan metode Convolutional Neural Network (CNN) yang telah terbukti efektif dalam mengenali pola kompleks pada citra. Metode pengembangan sistem mengikuti kerangka kerja SKKNI Nomor 299 Tahun 2020, yang meliputi tahapan Business Understanding, Data Understanding, Data Preparation, Modeling, dan Model Evaluation. Proses preprocessing dilakukan menggunakan OpenCV untuk mendeteksi area karakter, sedangkan model CNN dilatih dengan memanfaatkan perangkat lunak seperti Jupyter Notebook (Python) dan Visual Studio Code.

Kata Kunci: Pengenalan Plat Nomor, *Convolutional Neural Network (CNN)*, *OpenCV*.

Implementation Of Convolutional Neural Network Method for Vehicle License Plate Recognition

ABSTRACT

This research aims to develop a vehicle license plate recognition system based on artificial intelligence technology to support the needs of intelligent transportation systems, such as e-parking and vehicle identification. The main challenges in this study include variations in lighting conditions, image capture angles, license plate designs, and physical damage to plates. To address these challenges, this research implements the Convolutional Neural Network (CNN) method, which has been proven effective in recognizing complex patterns in images. The system development methodology follows the SKKNI Framework Number 299 of 2020, which consists of Business Understanding, Data Understanding, Data Preparation, Modeling, and Model Evaluation stages. The preprocessing process is carried out using OpenCV to detect character areas, while the CNN model is trained using software such as Jupyter Notebook (Python) and Visual Studio Code.

Keywords: License Plate Recognition, *Convolutional Neural Network (CNN)*, *OpenCV*

1. PENDAHULUAN

Perkembangan teknologi kecerdasan buatan telah merambah ke berbagai sektor, termasuk dalam bidang transportasi yang vital. Dengan banyaknya jumlah kendaraan di jalan raya, kebutuhan untuk memantau dan mengendalikan lalu lintas menjadi semakin mendesak. Dalam konteks ini, kemampuan untuk mengenali plat nomor kendaraan secara akurat dan efisien menjadi sangat penting, tidak hanya untuk pengawasan lalu lintas, tetapi juga untuk aplikasi seperti pelaporan pelanggaran hingga manajemen parkir.

Salah satu tantangan utama dalam pengenalan plat nomor kendaraan adalah adanya variasi dalam kondisi pencahayaan, sudut pengambilan gambar, jenis plat nomor seperti warna pada plat nomor kendaraan, serta

kerusakan atau keausan pada plat nomor itu sendiri. Variabilitas ini sering kali menyulitkan proses pengenalan, mengurangi keakuratan dan efisiensi sistem yang ada saat ini.

Penggunaan teknik pengolahan citra dan pembelajaran mesin, khususnya Convolutional Neural Network (CNN), menawarkan solusi yang sangat menjanjikan dalam mengatasi tantangan tersebut. Convolutional Neural Network (CNN), sebagai salah satu dari jaringan saraf tiruan yang dirancang untuk melakukan ekstraksi fitur dari data sebuah citra secara matematis. Dengan kemampuannya dalam mengenali berbagai macam pola yang cukup kompleks, Convolutional Neural Network (CNN) telah berhasil diterapkan dalam berbagai tugas seperti, segmentasi

gambar, klasifikasi gambar, deteksi objek, generatif 1 2 gambar hingga peningkatan kualitas gambar. salah satunya pada kasus ini dilakukan pengenalan pada plat nomor kendaraan .Karena itulah dipilihlah Judul ini karena relevansinya dengan kebutuhan saat ini untuk teknologi yang dapat meningkatkan keamanan dan efisiensi sistem transportasi. Implementasi Convolutional Neural Network (CNN) dalam pengenalan plat nomor tidak hanya membuka jalan untuk peningkatan sistem pengawasan lalu lintas yang ada, tetapi juga menjanjikan peningkatan dalam aplikasi-aplikasi seperti sistem e-parking yang efisien.

Melalui penelitian ini, diharapkan dapat menunjukkan bagaimana penerapan metode Convolutional Neural Network (CNN) dalam pengenalan plat nomor kendaraan dapat mengatasi masalah-masalah yang ada dan secara signifikan meningkatkan kinerja sistem pengawasan lalu lintas ataupun manajemen parkir. Dengan demikian, penelitian ini tidak hanya berkontribusi pada pengetahuan akademis tapi juga memiliki dampak praktis yang luas dalam peningkatan infrastruktur transportasi modern.

Naskah ditulis dengan bahasa Indonesia sesuai EYD, dan istilah asing harus dicetak miring (*italic*).

Pendahuluan harus berisi: latar belakang penelitian dilakukan, permasalahan, *state of art*, kebaruan, kontribusi dan solusi untuk penyelesaian masalah.

2. RUANG LINGKUP

Dalam penelitian ini permasalahan mencakup:

1. Cakupan permasalahan

Berdasarkan latar belakang diatas maka rumusan masalah yang dapat dikemukakan adalah “Bagaimana mengimplementasikan metode convolutional neural network (CNN) untuk merancang sistem cerdas untuk melakukan pengenalan pada plat nomor kendaraan?”.

2. Batasan-batasan penelitian

Untuk menghindari meluasnya masalah yang akan dibahas dalam penelitian ini, maka penulis menetapkan batasan masalah yaitu sebagai berikut:

1 Citra yang digunakan adalah gambar dari plat nomor kendaraan yang bersih tanpa adanya objek lain yang menghalangi dan diambil dengan kamera.

2 Penggunaan dataset yang relevan dan representatif untuk melatih dan menguji model pengenalan plat nomor kendaraan

3 Format file gambar yang diperoleh untuk sampel berupa *.jpg, *.jpeg *.png.

4 Menghasilkan sebuah prototype aplikasi berbasis website yang dibangun dengan Bahasa pemrograman python.

5 Pengembangan aplikasi dilakukan hingga evaluasi model.

6 Untuk implementasinya menggunakan web dan menggunakan library python yaitu framwork Flask

3. Rencana hasil yang didapatkan

3. BAHAN DAN METODE

3.1 Machine Learning

Menurut Cholissodin, I, .dkk (2020), Machine Learning (ML) atau mesin membelajar merupakan cabang dari Artificial Intelligence AI yang berfokus untuk belajar dari data (learn from data), yaitu fokus pada pengembangan sistem yang mampu belajar secara “mandiri” tanpa harus berulang kali diprogram manusia. ML membutuhkan Data yang valid sebagai bahan belajar (ketika proses training) sebelum digunakan ketika testing untuk hasil output yang optimal. Hasil pengembangan produk bernasis AI (Machine Learning dan Teknik lainnya misal dengan Optimasi, etc) ini harapannya dapat lebih memberikan kemudahan dan langsung dapat diterapkan di masyarakat luas atau bahkan masuk ke industri dalam skala nasional dan internasional

3.2 Deep Learning

Menurut Danukusumo, (2017), Deep Learning merupakan salah satu bidang dari machine learning yang memanfaatkan jaringan syaraf tiruan untuk implementasi permasalahan dengan dataset yang besar. Teknik deep learning memberikan arsitektur yang sangat kuat untuk supervised learning. Dengan menambahkan lebih banyak lapisan maka model pembelajaran tersebut bisa mewakili data citra berlabel dengan lebih baik. Pada machine learning terdapat teknik untuk menggunakan ekstraksi fitur dari data pelatihan dan algoritma 13 pembelajaran khusus untuk mengklasifikasi citra maupun untuk mengenali suara. Namun, metode ini masih memiliki beberapa kekurangan baik dalam hal kecepatan dan akurasi.

Aplikasi konsep jaringan syaraf tiruan yang dalam (banyak lapisan) dapat ditanggihkan pada algoritma machine learning yang sudah ada sehingga komputer sekarang bisa belajar dengan kecepatan, akurasi, dan skala yang besar. Prinsip ini terus berkembang hingga deep learning semakin sering digunakan pada komunitas riset dan industri untuk membantu memecahkan banyak masalah data besar seperti Computer vision, Speech recognition, dan Natural Language Processing.

Feature Engineering adalah salah satu fitur utama dari deep learning untuk mengekstrak pola yang berguna dari data yang akan memudahkan model untuk membedakan kelas. Feature Engineering juga merupakan teknik yang paling penting untuk mencapai hasil yang baik pada tugas prediksi. Namun, sulit untuk dipelajari dan dikuasai karena kumpulan data dan jenis data yang berbeda memerlukan pendekatan teknik yang berbeda juga.

Dalam deep learning, metode CNN atau Convolutional Neural Network sangatlah bagus dalam menemukan fitur yang baik pada citra ke lapisan berikutnya untuk membentuk hipotesis nonlinier yang dapat meningkatkan kekompleksitasan sebuah model. Model yang kompleks tentunya akan membutuhkan waktu pelatihan yang lama sehingga di dunia deep learning penggunaan GPU sudah sangatlah umum.

3.3 Algoritma CNN

Menurut Trisiawan dan Yuliza (2022), CNN adalah satu dari beberapa macam model yang ada pada *deep learning*. Struktur CNN terdiri dari feature learning layer dan fully connected layer. *Feature learning layer* dalam CNN terdiri dari beberapa layer, yaitu layer konvolusi, fungsi aktivasi (*ReLU*), *max pooling* dan *dropout regularization*. CNN bekerja secara hierarki, sehingga output pada layer konvolusi pertama digunakan sebagai input pada layer konvolusi selanjutnya.

3.4 Layer Konvolusi

Layer konvolusi merupakan layer pertama dalam arsitektur CNN. Konvolusi adalah istilah matematis dimana pengaplikasian sebuah fungsi pada output fungsi lain secara berulang. Dalam pengolahan gambar, konvolusi berarti mengaplikasikan sebuah kernel (kotak kuning) pada gambar di semua bagian yang memungkinkan seperti yang ditunjukkan pada Gambar 2.1 Kotak biru secara keseluruhan adalah gambar input. Kernel bergerak dari sudut kiri atas ke kanan bawah sehingga hasil konvolusi dari gambar tersebut berupa feature map (kotak hijau)

Gambar 2. 1 Operasi Konvolusi

$$h[m,n] = \sum \sum h[j,k]f[m-j,n-k]$$

Dimana:

f = input

h = kernel

m = baris pada feature map

n = kolom pada feature map

3.5 Fungsi Aktivasi (*ReLU*)

ReLU adalah salah satu jenis fungsi aktivasi yang bersifat *non-linear*. Nilai *output* dari *neuron* bisa dinyatakan sebagai 0 jika inputnya adalah negatif. Jika nilai input adalah positif, maka *output* dari *neuron* adalah nilai input aktivasi itu sendiri. Adapun persamaan untuk adalah sebagai berikut.

$$F(x) = \max(0, x)$$

3.6 Max Pooling

Max pooling merupakan operasi yang membagi output dari layer konvolusi menjadi beberapa grid kecil lalu mengambil nilai maksimal dari setiap grid untuk menyusun matriks gambar yang telah direduksi seperti pada Gambar 2.2, grid yang berwarna merah, hijau, kuning dan biru merupakan kelompok grid yang akan dipilih nilai maksimumnya. Sehingga hasil proses tersebut dapat dilihat pada kumpulan grid disebelah kanannya. Proses tersebut memastikan fitur yang

didapatkan akan sama meskipun objek gambar mengalami translasi (pergeseran).

Gambar 2. 2 Max Pooling

3.7 Fully Connected layer

Fully connected layer merupakan layer dimana semua neuron aktivasi dari layer sebelumnya terhubung semua dengan neuron di layer selanjutnya untuk menentukan fitur mana yang paling berkorelasi dengan kelas tertentu. Fungsi dari layer ini adalah untuk menyatukan semua node menjadi satu dimensi

3.8 Tensorflow

Menurut Pangestu, M. A., & Bunyamin, H. (2018), (2022), TensorFlow adalah framework machine learning yang bekerja dalam skala besar dan dalam environment yang heterogeneous. TensorFlow digunakan untuk melakukan eksperimen model deep learning, melatih model pada dataset yang berukuran 18 besar, dan membuatnya layak diproduksi. Selain itu TensorFlow juga mendukung training dan inference berskala besar dengan menggunakan ratusan server yang menggunakan Graphic Processing Unit (GPU) untuk training secara efisien. Penelitian ini menggunakan TensorFlow sebagai backend framework untuk Keras.

3.9 Keras

Menurut Pangestu, M. A., & Bunyamin, H. (2018). Keras adalah High Level Neural Network API yang dapat dijalankan di atas framework-framework machine learning seperti TensorFlow, CNTK, atau Theano. Keras ditulis dalam bahasa Python. Lebih lanjut, Keras menyediakan API yang mempermudah user dalam membangun arsitektur ANN. Keras juga menyediakan Keras Applications yang merupakan deep learning models yang digunakan bersama dengan pre trained weights. Model-model tersebut dapat digunakan untuk memprediksi, melakukan feature extraction atau fine-tuning. Model-model tersebut diantaranya adalah Xception, VGG16, VGG19, ResNet50, InceptionV3, InceptionResNetV2, MobileNet, DenseNet & NasNet. Penelitian ini akan membandingkan tingkat akurasi dari tiga Pre-trained model Keras yaitu ResNet50, Xception, dan VGG16.

3.10 Opencv

Menurut Fitrah, G. F. R. (2020), *OpenCV (Open-Source Vision Library)*, adalah sebuah *library open-source* yang dikembangkan oleh intel yang fokusnya untuk menyederhanakan programming terkait citra digital. *OpenCV* dapat digunakan dalam berbagai hal khususnya dalam pengenalan suatu objek seperti, pengenalan wajah, deteksi wajah dan berbagai jenis metode AI dengan pengimplementasian berbagai algoritma sederhana terkait computer vision

3.11 SKKNI Nomor 299 Tahun 2020

Menurut keputusan Menteri Republik Indonesia Nomor 299 Tahun 2020 tentang penetapan standar kompetensi kerja nasional indonesia kategori informasi dan komunikasi golongan pokok aktivitas pemrograman, konsultasi komputer dan kegiatan yang berhubungan dengan itu bidang keahlian Artificial Intelligence sub bidang Data Science.

| TUJUAN UTAMA | FUNGSI KUNCI | FUNGSI UTAMA | FUNGSI DASAR |
|---|-----------------------------------|-----------------------------------|---------------------------------------|
| Menemukan pengetahuan, insight atau pola yang bermanfaat dari data untuk berbagai keperluan | Menganalisis kebutuhan organisasi | Business understanding | Menentukan objektif bisnis |
| | | | Menentukan tujuan teknis data science |
| | | | Membuat rencana proyek data science |
| | | <i>Data understanding</i> | Mengumpulkan data |
| | | | Menelaah data |
| | | | Memvalidasi data |
| | Mengembangkan model | <i>Data preparation</i> | Memilah data |
| | | | Membersihkan data |
| | | | Menentukan label |
| | | | Mengintegrasikan data |
| | | <i>Modeling</i> | Membangun skenario pengujian |
| | | | Membangun model |
| | | <i>Model evaluation</i> | Mengevaluasi hasil pemodelan |
| | | | Melakukan review proses pemodelan |
| | | Menggunakan model yang dihasilkan | <i>Deployment</i> |
| | Melakukan deployment model | | |
| Membuat rencana pemeliharaan | | | |
| Melakukan pemeliharaan model | | | |
| <i>Evaluation</i> | Melakukan review | | |

| | | | |
|--|--|--|-----------------------------------|
| | | | proyek <i>data science</i> |
| | | | Membuat laporan akhir proyek data |

4. PEMBAHASAN

Metode Pengembangan sistem yang digunakan pada penelitian ini ialah menggunakan metode SKKNI (Standar Kompetensi Kerja Nasional Indonesia) No. 299 tahun 2020, bidang keahlian Artificial Intelligence, sub bidang Data Science. Yang terdiri dari tujuh fungsi utama seperti :

- 1 business understanding
- 2 data understanding
- 3 data preparation
- 4 modeling
- 5 model evaluation
- 6 Deployment
- 7 Evaluation

4.1 Business understanding

Tahapan pemahaman bisnis bertujuan untuk mendefinisikan masalah bisnis yang akan diselesaikan serta tujuan dari proyek penelitian ini. Dalam penelitian ini masalah utama yang dihadapi adalah bagaimana mengimplementasikan metode Convolutional Neural Network (CNN) untuk mengenali plat nomor kendaraan tingkat akurasi yang tinggi. Sesuai dengan fungsi dasar dalam SKKNI Nomor 299 Tahun 2020, bagian ini mencakup tiga elemen utama: Menentukan objektif bisnis, Menentukan tujuan teknis data science, dan Membuat rencana proyek data science seperti :

1. Menentukan objektif Bisnis
2. Menentukan Tujuan Teknis Data Science
3. Membuat Rencana Proyek Data Science

4.2 Data Understanding

Berdasarkan fungsi dasar yang tercantum dalam SKKNI Nomor 299 Tahun 2020, tahap ini mencakup mengumpulkan data, menelaah data, dan memvalidasi data. Proses ini penting untuk memastikan bahwa dataset yang digunakan memiliki kualitas yang memadai untuk mendukung pengembangan model yang andal.

1. Mengumpulkan data

Dataset yang digunakan dalam penelitian ini diperoleh dari repositori GitHub, yang menyediakan total gambar sebanyak 362. Data berupa gambar berbagai karakter alfanumerik yang terdapat pada plat kendaraan. Pemilihan repositori ini dilakukan berdasarkan kualitas dan kelengkapan dataset, sehingga dapat mencerminkan kondisi nyata yang mungkin ditemui di lapangan. Dataset yang diambil memiliki struktur yang memadai, dengan label yang terorganisir berdasarkan karakter pada plat nomor, sehingga mempermudah proses pelatihan dan validasi model.

2. Menelaah Data Pada tahap ini, memastikan setiap karakter alfanumerik (angka dan huruf) memiliki representasi yang memadai untuk mendukung pelatihan model secara adil. Proses ini dilakukan dengan menganalisis distribusi data untuk mengetahui apakah jumlah sampel pada setiap kelas sudah seimbang.

3. Memvalidasi Data

Tahapan ini dilakukan untuk memastikan bahwa data yang digunakan bebas dari error dan duplikasi. Setiap gambar diperiksa untuk memastikan bahwa gambar dalam format yang konsisten, yaitu pada format .png. Skrip di bawah ini berfungsi untuk mengonversi file gambar dengan format tertentu (seperti .jpg, .jpeg, .bmp, .tiff, dan .gif) menjadi format PNG di dalam direktori tertentu, termasuk subdirektoriya

4.3 Data Preparation

Pada tahap data preparation, dilakukan berbagai proses untuk memastikan dataset siap digunakan dalam pelatihan model. Langkah pertama adalah melakukan *import library* yang di butuhkan kemudian melakukan upload data gambar yang sudah disiapkan. Selanjutnya standarisasi ukuran gambar, di mana semua gambar diubah menjadi dimensi yang seragam agar sesuai dengan input model *Convolutional Neural Network (CNN)*.

1. Import library

Pada skrip diatas merupakan skrip untuk memanggil library tensorflow yang memiliki berbagai macam fungsi untuk melakukan pengolahan gambar seperti, augmentasi data , melakukan *preprocessing*, membuat model *Convolutional Neural Network (CNN)* hingga *multi layer perceptron* (jaringan saraf tiruan) ,hingga dapat melakukan penyimpanan model dengan format file H5 dan Keras .

2. Upload Data

Pada skrip diatas merupakan tampilan skrip untuk mengupload data dari folder path yang ada di windows explore dan di masukan kedalam variabel agar memudahkan dalam melakukan pemanggilan

data pada saat melakukan augmentasi dan preprocessing

3. Preprocessing

Pada skrip diatas merupakan skrip yang digunakan untuk meyiapkan parameter yang akan berfungsi untuk mengubah ukuran gambar dengan lebar 40 pixel dan tinggi 40 pixel (40x40) sehingga mempermudah dalam persiapan data training dan validation serta jumlah batch yang merupakan jumlah gambar yang akan di input pada model pada saat training

4. preprocessing

Pada Skripdiats sebuah objek yang bertugas untuk mengubah dan membagi data gambar untuk melatih model Convolutional Neural Network. Pertama, Gambar secara default memiliki nilai piksel dalam rentang [0, 255]. Dengan menggunakan *rescale = 1./255*, setiap nilai piksel akan dinormalisasi menjadi rentang [0, 1]. Langkah ini penting untuk membantu proses pembelajaran model menjadi lebih stabil dan konvergen lebih cepat. Kemudian menerapkan transformasi shear atau geser pada gambar dengan intensitas tertentu. Nilai 0.2 menandakan tingkat pergeseran sudut yang akan diterapkan secara acak. Hasilnya, beberapa gambar pada saat pelatihan akan sedikit “miring”, yang membantu model menjadi lebih robust terhadap variasi sudut atau kemiringan objek pada gambar asli. Serta melakukan zoom in atau zoom out secara acak dengan kisaran 20%. Dengan adanya zoom acak, model akan mengenali terhadap perubahan jarak objek dari kamera atau variasi ukuran objek. selain itu data dibagi menjadi 2 bagian yaitu 80% untuk melatih model dan 20% untuk menguji kinerja model. Semua langkah ini bertujuan untuk meningkatkan kemampuan model dalam mengenali pola pada gambar yang bervariasi dan mengurangi kemungkinan model terkaky bergantung pada data pelatihan tertentu

4.4 Permodelan

```
1. train_generator = train_datagen.flow_from_directory(  
2.     path,  
3.     target_size=(IMG_HEIGHT,IMG_WIDTH),  
4.     batch_size=32,  
5.     class_mode='categorical',  
6.     subset='training', # set sebagai training data  
7.     color_mode='grayscale') # set colormode as grayscale  
8.  
9. validation_generator = train_datagen.flow_from_directory(  
10.    path,  
11.    target_size=(IMG_HEIGHT,IMG_WIDTH),  
12.    batch_size=32,  
13.    class_mode='categorical',  
14.    color_mode='grayscale', #set sebagai training data  
15.    subset='validation') # set colormode as grayscale  
Output:  
Found 578 images belonging to 36 classes.  
Found 144 images belonging to 36 classes.
```

Skrip diatas bertujuan untuk mempersiapkan data yang akan dipakai dalam pembuatan model Convolutional Neural Network (CNN). Skrip ini membagi data menjadi dua bagian: data pelatihan (train) dan data validasi (validation). Pembagian ini penting agar model CNN dapat belajar dari data pelatihan dan kemudian diuji pada data validasi untuk melihat seberapa baik model tersebut bekerja pada data yang belum pernah dilihat sebelumnya

```
1. from tensorflow.keras import layers, models  
2. model = models.Sequential(  
3. # Layer pertama dengan input shape  
4. layers.Conv2D(8, (3, 3), activation='relu', padding='same',  
input_shape=(IMG_HEIGHT,IMG_WIDTH,1)),  
5. layers.MaxPool2D(),  
6. layers.Conv2D(16, 3, padding='same', activation='relu'),  
7. layers.MaxPooling2D(),  
8. layers.Conv2D(32, 3, padding='same', activation='relu'),  
9. layers.MaxPooling2D(),  
10. layers.Conv2D(64, 3, padding='same', activation='relu'),  
11. layers.MaxPooling2D(),  
12. layers.Flatten(),  
13. # Fully connected layer  
14. layers.Dense(128, activation='relu'),  
15. # Output layer untuk 36 kelas  
16. layers.Dense(36, activation='softmax')  
17.])
```

Pada skrip diatas menjelaskan proses pembuatan model CNN. Alur ini dimulai pada lapisan pertama dengan pengambilan input gambar berukuran 40x40 pixel dengan 1 channel warna, kemudian melakukan operasi konvolusi pada setiap pixel yang dilewati oleh

kernel berukuran 3x3, kemudian dilakukan fungsi aktivasi Relu(Retified Linear Unit),selanjutnya hasilnya diteruskan ke lapisan pooling. Gambar tersebut kemudian melalui beberapa lapisan kovolusi dan pooling beberapa kali secaraberulang. Setelah itu , hasilnya di ubah menjadi matrix 1 dimensi menggunakan fungsi Flatten() dan diteruskan ke lapisan fully connecteed untuk menghasil kan prediksi di lapisan terakhir

```
1. model.compile(optimizer='adam',  
2.     Loss='categorical_crossentropy',  
3.     metrics=['accuracy'])
```

Pada skrip diatas merupakan tahap berikutnya setelah pembuatan model CNN adalah melakukan konfigurasi untuk pelatihan dan pengujian model dengan menggunakan skrip model.compile(optimizer='adam',loss='categorical_crossentropy', metrics= ['accuracy']). Proses ini mencakup tiga komponen utama:

1.Optimizer

Optimizer 'Adam': secara otomatis menyesuaikan laju pembelajaran setiap parameter, memadukan keunggulan metode AdaGrad dan RMSProp, sehingga proses pelatihan model menjadi lebih cepat, stabil, dan umumnya cocok untuk berbagai jenis masalah tanpa memerlukan banyak penyesuaian hyperparameter.

2.Fungsi Kerugian (loss function)

Fungsi kerugian sparse categorical crossentropy: Digunakan untuk masalah klasifikasi dengan lebih dari dua kelas. Ini mengukur seberapa baik prediksi model dibandingkan dengan label sebenarnya dan memberikan umpan balik untuk mengurangi kesalahan.

3.Metrik Akurasi

Metrik akurasi: Menunjukkan persentase keakuratan model dalam melakukan prediksi dengan benar kemudian dibandingkan dengan total prediksi, membantu mengevaluasi kinerja model selama pelatihan dan pengujian.

```
1. model_checkpoint = ModelCheckpoint(path_saving_model, save_best_only=True,  
2.     verbose=1, monitor = 'val_accuracy', mode= 'auto')  
3. reduce_lr = ReduceLRonPlateau(monitor='val_accuracy', factor= 0.2,patience= 3 ,  
4.     verbose=1 , mode='auto')
```

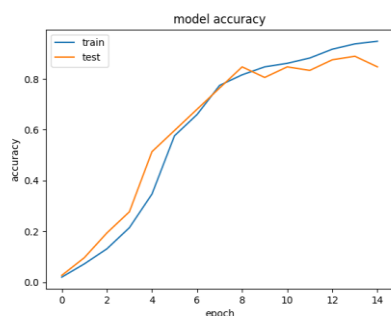
Pada skrip tersebut melakukan konfigurasi callback untuk memaksimalkan kinerja pelatihan model. Pertama, variabel model_checkpoint didefinisikan menggunakan objek ModelCheckpoint dengan parameter save_best_only=True. Strategi ini bertujuan untuk hanya menyimpan bobot model terbaik berdasarkan metrik val_accuracy yang dipantau. Dengan demikian, sistem dapat menghindari penumpukan file bobot yang tidak dibutuhkan, sekaligus memastikan bahwa model yang tersimpan selalu

mewakili performa terbaik yang telah dicapai selama proses pelatihan. Penggunaan verbose=1 memungkinkan peneliti memantau proses penyimpanan model pada setiap peningkatan kinerja yang terdeteksi. Selanjutnya, variabel reduce_lr diatur melalui ReduceLROnPlateau guna menyesuaikan laju pembelajaran (learning rate) secara dinamis ketika metrik kinerja tidak mengalami peningkatan setelah beberapa epoch. Argumen factor=0.2 menunjukkan bahwa laju pembelajaran akan dikurangi menjadi 20% dari nilai 49 sebelumnya ketika kondisi ini terpenuhi, sedangkan patience=3 menetapkan jumlah epoch yang ditunggu sebelum penurunan laju pembelajaran dilakukan. Dengan menetapkan monitor='val_accuracy' dan verbose=1, peneliti dapat dengan mudah memantau kapan dan mengapa laju pembelajaran mengalami penyesuaian, sehingga dapat segera mengamati efeknya terhadap proses konvergensi model

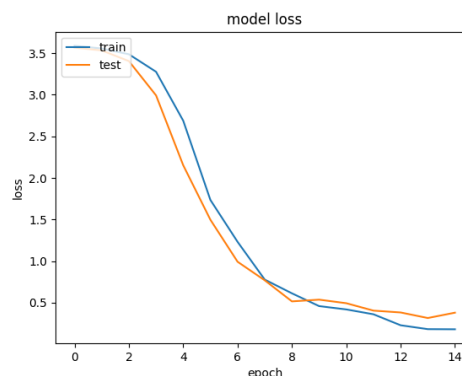
```
1. history=model.fit(x=train_generator, validation_data=validation_generator,
2. epochs=15,shuffle= True,callbacks=[model_checkpoint, reduce_lr])
```

Terakhir pada skrip tersebut proses pelatihan dimulai dengan pemanggilan model.fit() yang menggunakan train_generator sebagai data latih dan validation_generator sebagai data validasi. Parameter epochs=15 dipilih untuk memberikan kesempatan model beradaptasi secara bertahap terhadap pola dalam data. Opsi shuffle=True memastikan bahwa data diacak setiap epoch, mencegah model mengingat urutan data secara spesifik. Sementara itu, pengintegrasian kedua callback (model_checkpoint dan reduce_lr) melalui argumen callbacks=[model_checkpoint, reduce_lr] memastikan bahwa model selalu mencoba meningkatkan kinerjanya, menyimpan bobot terbaik yang ditemukan, serta mengoptimalkan laju pembelajaran secara adaptif sesuai respon model terhadap metrik yang dipantau. Semua hasil pelatihan, termasuk tren peningkatan atau penurunan kinerja, direkam dalam objek history

4.5 Model Evaluation



Gambar 4. 1 Grafik Akurasi



Gambar 4. 2 grafik loss function

Kedua grafik ini tidak hanya divisualisasikan, tetapi juga disimpan sebagai file gambar dengan format PNG, accuracy_plot.png untuk grafik akurasi dan loss_plot.png untuk grafik loss. Hal ini dilakukan untuk memastikan dokumentasi hasil pelatihan model tersedia dan dapat dianalisis lebih lanjut. Visualisasi ini menjadi alat penting dalam evaluasi performa model, memberikan informasi yang esensial untuk mendeteksi potensi overfitting atau underfitting, serta menjadi dasar keputusan untuk penyesuaian model di iterasi berikutnya. Grafik akurasi dan loss yang dihasilkan selama proses pelatihan memberikan gambaran penting tentang kinerja model yang dikembangkan. Pada Gambar 4.16, terlihat bahwa akurasi model meningkat secara konsisten pada awal pelatihan, baik untuk data pelatihan maupun validasi. Pada epoch awal (0-5), peningkatan akurasi terjadi cukup tajam, menunjukkan bahwa model berhasil mempelajari pola dasar dari dataset. Namun, setelah sekitar epoch ke-10, akurasi validasi mulai stabil dengan sedikit fluktuasi, sementara akurasi pelatihan terus meningkat.

```
1. acc = history.history['accuracy'][-1]
2. val_acc = history.history['val_accuracy'][-1]
3. print(f'Hasil akhir akurasi : {acc*100:.2f}% | validasi akurasi :
      {val_acc*100:.2f}%')
Output :
Hasil akhir akurasi : 92.36% | validasi akurasi : 87.50%
```

Skrip diatas menunjukkan hasil akhir akurasi sistem pada data latih yang telah berhasil mencapai angka sebesar 92,36%. Angka ini menunjukkan bahwa model memiliki kemampuan yang cukup baik dalam mengenali pola-pola yang terdapat pada dataset pelatihan. Selain itu, ketika dievaluasi menggunakan data validasi, model juga mencatat nilai akurasi sebesar 87,50%. Nilai akurasi validasi

ini memberikan gambaran mengenai sejauh mana model mampu mempertahankan kinerjanya ketika dihadapkan dengan data baru yang tidak digunakan dalam proses pelatihan. Perbedaan antara akurasi pelatihan dan akurasi validasi yang tidak terlalu besar menandakan bahwa model tidak mengalami overfitting yang berlebihan. Dengan kata lain, model tidak hanya sekadar “mengingat” pola dari data latih, tetapi juga dapat menggeneralisasikan pengetahuan tersebut pada data yang belum pernah dilihat sebelumnya. Hal ini memberikan indikasi bahwa metode dan parameter yang dipilih dalam pelatihan relatif seimbang. Meskipun demikian, masih ada ruang untuk perbaikan. Tingkat akurasi yang telah dicapai sangat baik, tetapi peningkatan lebih lanjut dapat dilakukan dengan mengeksplorasi berbagai pendekatan, seperti penyesuaian arsitektur 54 model, pemilihan hyperparameter yang lebih tepat, penambahan data latih, atau penerapan teknik augmentasi data yang lebih beragam. Langkah-langkah ini diharapkan dapat meningkatkan kemampuan model dalam mengklasifikasikan data secara akurat dan konsisten, sehingga hasil yang diperoleh dapat semakin mendekati tingkat kinerja yang diharapkan saat di implementasikan ke aplikasi.

```
1. import keras
2. keras.saving.save_model(model,filepath='./Best-
model/best_model.h5')
```

Skrip diatas menampilkan skrip yang digunakan untuk menyimpan file dalam format .h5, Skrip ini bertujuan untuk memastikan bahwa data yang diolah dapat disimpan dengan aman dan mudah diakses kembali di kemudian hari. Dalam proses penyimpanan ini, langkah-langkah penting meliputi pengaturan nama file, menentukan lokasi penyimpanan, dan memastikan bahwa data tersimpan dalam struktur yang benar.

4.6 Deployment

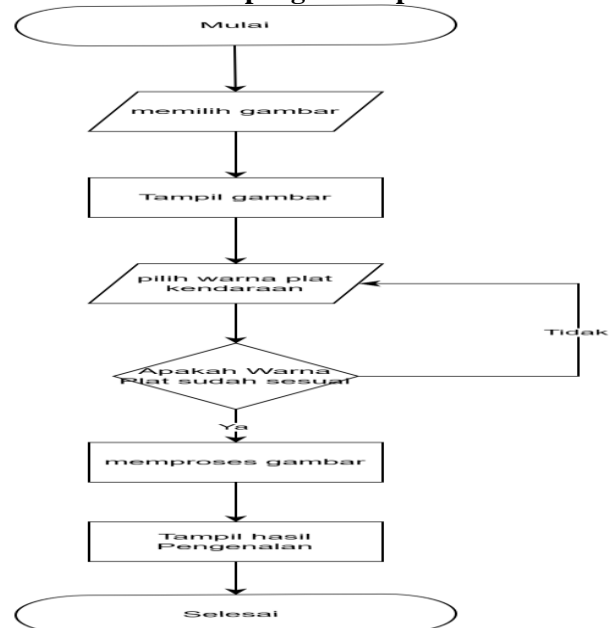
Tahap deployment bertujuan untuk menghubungkan hasil penelitian yang sebelumnya hanya diuji dalam lingkungan internal ke sebuah prototipe aplikasi berbasis web. Dengan cara ini, model yang telah dibangun dan dievaluasi dapat diakses langsung oleh pengguna melalui antarmuka yang sederhana dan mudah dipahami. Melalui prototipe tersebut, pengguna dapat mencoba sendiri fungsi prediksi atau klasifikasi yang disediakan oleh model, sehingga memberikan gambaran yang lebih nyata dan praktis mengenai kinerja sistem serta potensi penerapannya dalam konteks yang lebih luas.

4.6.1 Preoses Deployment

Proses deployment dalam penelitian ini dirancang untuk memastikan alur sistem pengenalan plat nomor kendaraan berjalan secara efisien dari awal hingga akhir. Sistem ini berbasis aplikasi web yang memungkinkan pengguna untuk mengunggah gambar plat nomor kendaraan, memprosesnya melalui backend, dan mendapatkan hasil prediksi berupa karakter alfanumerik. Backend sistem menggunakan metode Convolutional Neural Network (CNN) untuk melakukan prediksi, sedangkan preprocessing citra dilakukan menggunakan OpenCV untuk mengekstrak karakter dari gambar plat nomor. Antarmuka web dibangun menggunakan framework Flask, yang memfasilitasi komunikasi antara frontend dan backend.

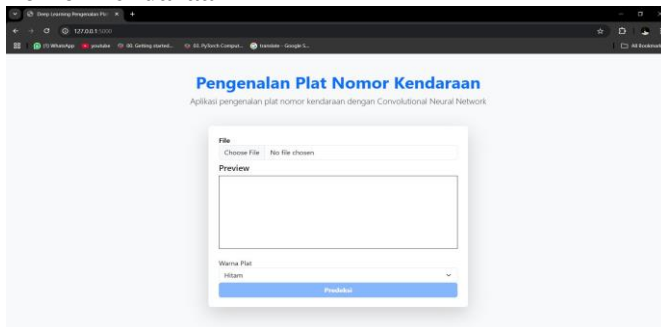
Setelah gambar diunggah oleh pengguna, backend memproses citra dengan langkah-langkah seperti konversi warna, thresholding, dan ekstraksi kontur untuk mendeteksi Region of Interest (ROI), yaitu area yang berisi karakter plat nomor. ROI ini kemudian diteruskan ke model CNN yang telah dilatih sebelumnya untuk menghasilkan prediksi karakter. Hasil prediksi berupa nomor plat kendaraan dikirim kembali ke frontend dan ditampilkan kepada pengguna melalui antarmuka yang sederhana dan interaktif. Proses deployment ini dirancang untuk memberikan pengalaman pengguna yang intuitif serta memanfaatkan teknologi machine learning dan pengolahan citra untuk menghasilkan sistem yang akurat dan andal.

4.6.2 Flowchart pengenalan plat kendaraan



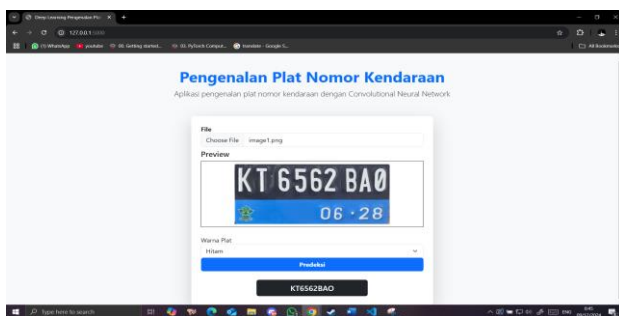
Gambar 4. 3 Flowchart pengenalan plat nomor kendaraan

4.6.3 Tampilan halaman pengenalan plat nomor kendaraan



Gambar 4. 4 Tampilan awal aplikasi pengenalan plat nomor kendaraan

Pada gambar 4.4 merupakan tampilan awal website, halaman utama menampilkan judul "Pengenalan Plat Nomor Kendaraan", yang memberikan informasi kepada pengguna mengenai fungsi utama aplikasi ini. Deskripsi singkat di bawah judul menjelaskan bahwa aplikasi ini menggunakan pendekatan berbasis Convolutional Neural Network (CNN) untuk mengenali karakter pada plat nomor kendaraan. Formulir untuk mengunggah file ditampilkan di tengah halaman, dengan tombol "Choose File" yang memungkinkan pengguna memilih gambar dari perangkat mereka. Di bawahnya terdapat kotak Preview yang dirancang untuk menampilkan gambar yang diunggah, bersama dengan opsi dropdown untuk memilih warna plat nomor kendaraan



Gambar 4. 5 Menampilkan Hasil Pengenalan Plat Nomor

Pada Gambar 4.5 setelah menginput gambar tampilan pratinjau gambar plat nomor kendaraan setelah file dipilih oleh pengguna melalui tombol "Choose File". Gambar yang ditampilkan di area Preview memungkinkan pengguna untuk memverifikasi file yang diunggah sebelum melanjutkan ke tahap prediksi. Setelah memastikan gambar yang dipilih sesuai, pengguna dapat menggunakan tombol "Prediksi" untuk memulai proses analisis. Tombol ini akan menghubungkan file gambar ke model Convolutional Neural

Network (CNN) yang telah dikembangkan untuk mengenali karakter pada plat nomor kendaraan. Dan hasil pengenalan akan tampil dibagian bawah tombol

4.7 Evaluation

Tahap selanjutnya ialah implementasi atau pengujian dimana sistem pengenalan plat akan diuji menggunakan metode pengujian sistem yaitu black-box testing yang berfungsi untuk melihat kesalahan sistem prototype ini dapat berjalan dengan lancar.

4.7.1 Black-Box Testing

Pada tahap ini, dilakukan pengujian sistem secara menyeluruh untuk memastikan bahwa sistem berfungsi dengan baik dan memenuhi seluruh persyaratan yang telah ditentukan. Pengujian bertujuan untuk mengevaluasi apakah prototype yang dikembangkan mampu beroperasi sesuai dengan fungsi yang dirancang dan dapat diandalkan dalam berbagai kondisi erasional. Setiap komponen dan fitur sistem diuji secara komprehensif untuk memastikan bahwa tidak ada kesalahan atau kekurangan yang terlewatkan. Melalui pengujian yang mendetail, masalah yang muncul dapat diidentifikasi dan diperbaiki sebelum sistem diluncurkan secara penuh. Evaluasi ini juga mencakup pengujian kinerja sistem dalam berbagai skenario penggunaan untuk memastikan bahwa sistem mampu menangani kondisi nyata yang beragam dan memenuhi harapan pengguna

| No | Input | Yang Diharapkan | Hasil |
|----|------------------------------------|--|----------|
| 1 | Penguji mengakses website tersebut | Menampilkan tampilan awal website | Berhasil |
| 2 | Fitur pilih media | Bisa mengupload file ke sistem pengenalan plat nomor kendaraan | Berhasil |
| 3 | Fitur deteksi media | Mampu mengenali plat nomor kendaraan | Berhasil |

5. KESIMPULAN

Penelitian ini berhasil mengembangkan sistem pengenalan plat nomor kendaraan berbasis Convolutional Neural Network (CNN) yang memiliki tingkat akurasi yang cukup tinggi. Melalui proses pengembangan menggunakan metode SKKNI, tujuh tahapan utama (Business Understanding, Data Understanding, Data Preparation, Modeling, Model Evaluation, Deployment,

dan Evaluation) telah diterapkan untuk memastikan bahwa sistem ini mampu beroperasi secara optimal. Berdasarkan hasil pengujian, model yang dikembangkan mencapai akurasi 92.36% pada data pelatihan dan 87.50% pada data validasi, menunjukkan performa yang baik dalam mengenali pola alfanumerik pada plat nomor kendaraan dalam berbagai kondisi. Keunggulan sistem ini terlihat pada kemampuannya untuk mengenali karakter plat nomor dengan cepat dan efisien. Selain itu, prototipe berbasis web yang dihasilkan memudahkan pengguna untuk mengunggah gambar plat nomor kendaraan dan mendapatkan hasil prediksi dalam waktu singkat. Penelitian ini juga menunjukkan bahwa penggunaan teknik augmentasi data dan preprocessing berbasis OpenCV memberikan kontribusi signifikan dalam meningkatkan performa model. Namun, beberapa tantangan tetap ada, seperti kesalahan prediksi pada karakter dengan bentuk visual yang mirip (misalnya, "O" dan "Q") serta kebutuhan untuk meningkatkan akurasi pada karakter dengan distribusi data yang tidak seimbang. Dengan demikian, penelitian ini tidak hanya memberikan solusi yang relevan untuk pengawasan lalu lintas dan manajemen parkir, tetapi juga memberikan kontribusi dalam pengembangan teknologi transportasi cerdas di masa depan.

6. SARAN

Penelitian ini masih memiliki peluang untuk dikembangkan lebih lanjut, terutama dalam meningkatkan akurasi dan keandalannya. Diharapkan hasil penelitian ini dapat menjadi pijakan untuk pengembangan teknologi serupa di masa mendatang dan memberikan manfaat nyata bagi sektor transportasi. Untuk meningkatkan kualitas sistem, beberapa langkah yang disarankan meliputi:

1. Menggunakan dataset yang lebih besar dan beragam untuk memastikan representasi data yang lebih merata dan mendukung performa model.
2. Melakukan augmentasi data dengan teknik yang lebih bervariasi, seperti perubahan intensitas cahaya, rotasi,

dan penghilangan noise, untuk meningkatkan generalisasi model.

3. Meningkatkan efisiensi sistem untuk dapat memproses data secara real time pada skala besar.

4. Mengembangkan sistem ini lebih lanjut dengan mengintegrasikan sebuah sistem deteksi objek sehingga dapat secara langsung melakukan pengenalan plat nomor dan membacanya.

7. DAFTAR PUSTAKA

- Cholissodin, I., Sutrisno, Soebroto, A. A., Hasanah, U., & Febiola, Y. I. (2020). *AI, Machine Learning & Deep Learning (Teori & Implementasi) "from Basic Science to High Scientific Solution for Any Problem" Versi 1.01*. 317. <http://bit.ly/3piOnnU>
- Danukusumo, K. P., & Maslim, M. (2017, September). Indonesia ancient temple classification using convolutional neural network. In 2017 international conference on control, electronics, renewable energy and communications (ICCREC) (pp. 50-54). IEEE
- Pangestu, M. A., & Bunyamin, H. (2018). Analisis Performa dan Pengembangan Sistem Deteksi Ras Anjing pada Gambar dengan Menggunakan Pre-Trained CNN Model. *Jurnal Teknik Informatika Dan Sistem Informasi*, 4(2), 2443–2229. <http://dx.doi.org/10.28932/jutisi.v4i2.828>
- Teori, L., Vision, C., & Wajah, D. (n.d.). *12.10115406_Geri Fitrah Ridjki Ramadhan_Bab 2*. 11–34.
- Trisiawan, I. K., & Yuliza, Y. (2022). Penerapan Multi-Label Image Classification Menggunakan Metode Convolutional Neural Network (CNN) Untuk Sortir Botol Minuman. *Jurnal Teknologi Elektro*, 13(1), 48. <https://doi.org/10.22441/jte.2022.v13i1.009>