

KLASIFIKASI CITRA KOMPONEN SEPEDA MOTOR MENGGUNAKAN ALGORITMA CNN DENGAN ARSITEKTUR MOBILENET

Oleh:

Renaldi Anggarkusuma¹, Izmy Alwiah Musdar^{2*}, Hasniati³

^{1,2,3}Program Studi Teknik Informatika, STMIK Kharisma Makassar

²Program Studi Sistem Informasi, UIN Alauddin Makassar

e-mail: ¹renaldianggarkusuma_20@kharisma.ac.id, ²izmyalwiah@kharisma.ac.id,

³hasniati@kharisma.ac.id

Abstrak: Pengenalan citra merupakan sub-kategori teknologi visi komputer yang digunakan untuk mengklasifikasikan citra ke dalam kategori tertentu. Tujuan dari penelitian ini adalah untuk membuat model CNN dengan arsitektur MobileNet untuk mengklasifikasikan citra komponen sepeda motor serta mengukur tingkat akurasi yang dihasilkan oleh model. Pembuatan model deep learning CNN menggunakan library TensorFlow. Data awal untuk proses pelatihan yang digunakan sebanyak 50 citra yang dibagi kedalam 5 jenis kategori yaitu citra busi, kampas rem, laher, kiprok dan rumah roller. Data-data tersebut dilakukan teknik augmentasi berupa rotasi, pergeseran, dan pencerminan citra (flip). Penelitian ini berhasil membuat model CNN menggunakan arsitektur MobileNet yang mampu mengklasifikasikan citra komponen sepeda motor. Pengujian model mobilenet dilakukan dengan menggunakan 20 data uji dimana 10 diantaranya menggunakan filter motion blur. Dari hasil pengujian diperoleh hasil bahwa kinerja akurasi model CNN dengan arsitektur MobileNet dalam mengklasifikasikan citra komponen sepeda motor sebesar 95% untuk citra tanpa proses blur dan 90% untuk citra yang blur.

Kata kunci: Deep Learning, Klasifikasi Citra, Convolutional Neural Network, MobileNet

Abstract: Image recognition is a sub-category of computer vision technology used to classify images into specific categories. The purpose of this research is to create a CNN model with the MobileNet architecture to classify motorcycle component images and measure the accuracy level produced by the model. The creation of the deep learning CNN model uses the TensorFlow library. The initial data for the training process consists of 50 images divided into 5 categories: spark plugs, brake pads, bearings, regulators, and roller housings. These data undergo augmentation techniques such as rotation, shifting, and image flipping. This research successfully developed a CNN model using the MobileNet architecture that can classify motorcycle component images. The MobileNet model was tested using 20 test data, with 10 of them subjected to a motion blur filter. The test results showed that the accuracy performance of the CNN model with the MobileNet architecture in classifying motorcycle component images is 85%, and the accuracy of image classification did not significantly decrease when the motion blur filter was applied.

Keywords: Deep Learning, Image Classification, Convolutional Neural Network, MobileNet

1. PENDAHULUAN

Sepeda motor merupakan salah satu kendaraan yang banyak digunakan sebagai kendaraan pribadi. Sepeda motor merupakan kendaraan yang dapat mengantarkan manusia maupun barang ke tempat tujuan dalam waktu singkat, sehingga banyak digunakan oleh

* Corresponding author : Izmy Alwiah Musdar (izmyalwiah@kharisma.ac.id)

masyarakat di seluruh dunia khususnya di negara berkembang karena harganya yang terjangkau. Indonesia merupakan negara ketiga dengan pengguna sepeda motor terbanyak menurut hasil survei yang dilakukan oleh pusat penelitian pew [1].

Terdapat banyak sekali komponen yang ada didalam sebuah sepeda motor mulai dari komponen yang harus diganti secara berkala sampai pada komponen yang jarang rusak atau diganti. Hal ini tentu sangat sulit untuk diketahui bagi orang yang tidak mengetahui komponen-komponen tersebut. Maka diperlukan sebuah teknologi untuk pengenalan citra dari komponen-komponen tersebut. Metode pengenalan citra yang digunakan adalah Convolutional Neural Network atau CNN.

CNN (Convolutional Neural Network) merupakan salah satu implementasi dari deep learning yang digunakan untuk pemrosesan citra digital [2], [3]. CNN yang merupakan jenis utama dari jaringan syaraf multilayer yang terinspirasi oleh mekanisme sistem optik makhluk hidup [4]. Convolutional Neural Network terdiri dari beberapa lapisan yang berbeda yaitu, lapisan konvolusi, lapisan pooling, lapisan fully connected dan lapisan output dengan simpul atau yang disebut dengan neuron sebagai penghubung lapisan-lapisan tersebut neuron memiliki parameter seperti weight, bias dan activation function [5]. Pada tahun 2012 kemampuan CNN dianggap sebagai model terbaik untuk menyelesaikan permasalahan deteksi objek dan pengenalan objek. Saat itu CNN mampu mengenali citra digital dengan sangat baik menggunakan arsitektur AlexNet yang diperkenalkan oleh Krizhevsky et al (2012) [6]. Dengan memanfaatkan kemampuan GPU CNN memberikan kinerja yang luar biasa dalam mengklasifikasikan citra.

Ada banyak arsitektur di dalam algoritma CNN diantaranya yaitu Visual Geometry Group (VGG) Net, Residual Network (ResNet) 50, GoogleNet dan Inception-V3 namun arsitektur-arsitektur tersebut menggunakan sumber daya komputasi yang besar untuk mendapatkan akurasi yang baik [7].

MobileNets merupakan salah satu arsitektur CNN yang dirancang khusus untuk perangkat mobile yang dapat digunakan untuk mengatasi masalah akan kebutuhan sumber daya komputasi yang berlebih yang diperkenalkan pertama kali oleh peneliti Google Andrew G. Howard dkk pada tahun 2017 [8]. Perbedaan antara MobileNet dengan arsitektur yang lain adalah MobileNet memiliki jumlah parameter yang lebih sedikit yang menyebabkan waktu train menjadi lebih cepat. Oleh karena itu, pada penelitian ini digunakan algoritma CNN dengan arsitektur MobileNet untuk pengenalan citra komponen sepeda motor.

2. Convolutional Neural Network

CNN merupakan salah satu implementasi dari deep learning yang banyak digunakan untuk pemrosesan data dua dimensi seperti pengenalan gambar, pengolahan video dan pengenalan tulisan tangan. Convolutional Neural Network (CNN) adalah pengembangan dari multilayer perceptron [3]. CNN termasuk dalam jenis Deep Neural Network, karena kedalaman jaringan yang tinggi.

Convolutional Neural Network terdiri dari beberapa lapisan yang berbeda yaitu, lapisan konvolusi, lapisan pooling, lapisan fully connected dan lapisan output. Struktur dari algoritma CNN dapat dilihat pada Gambar 1 [5], [9].

a. Lapisan Konvolusi

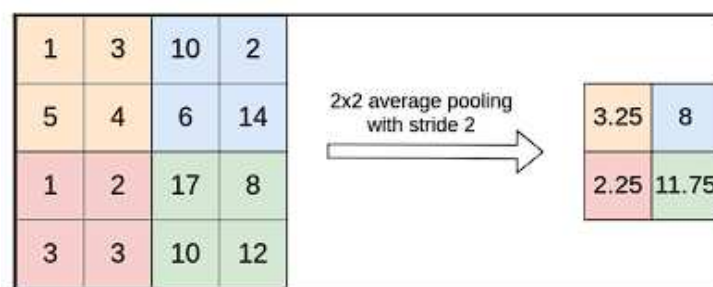
Lapisan konvolusi merupakan operasi konvolusi pada data *input*. Operasi konvolusi merupakan sebuah operasi matematika berupa perkalian antar dua buah matriks yang dilakukan pada setiap piksel *input* dengan *kernel* atau filter [10]. Lapisan konvolusi terdiri dari *neuron* yang tersusun sedemikian rupa sehingga membentuk sebuah filter dengan panjang dan tinggi (piksel). Kemudian filter tersebut digeser mulai dari sudut kiri atas hingga kanan bawah dari sebuah citra. Pergeseran filter diatur menggunakan stride, stride adalah besaran jumlah piksel yang dilewati saat pergeseran filter. Operasi “*dot*” dilakukan setiap kali filter digeser sehingga menghasilkan sebuah *output* atau biasa disebut sebagai *activation map* atau *feature map*.

b. Lapisan Pooling

Lapisan *pooling* digunakan untuk mengurangi dimensi data setelah melalui proses konvolusi. Proses pooling dapat digunakan untuk menangkap fitur yang penting dari data dan meningkatkan invariansi terhadap perubahan pada data asli. Proses *pooling* mengurangi jumlah parameter yang harus di perbarui. Terdapat beberapa jenis operasi pooling yaitu max pooling dan average pooling [10]. Max pooling mengambil nilai maksimum untuk masing-masing *grid* yang dibagi dari output lapisan konvolusi sedangkan average pooling mengambil nilai rata-rata dari setiap *grid* untuk menghasilkan *feature map* yang lebih ringkas.

1) Average Pooling

Average pooling adalah operasi pooling yang menghitung nilai rata-rata dari setiap output area pooling kernel [11]. Pada Gambar 1 menunjukkan contoh dari operasi average pooling yang mengambil nilai rata-rata untuk mengurangi dimensi data.



Gambar 1. Average Pooling

[Sumber : Vasilev dkk, 2019]

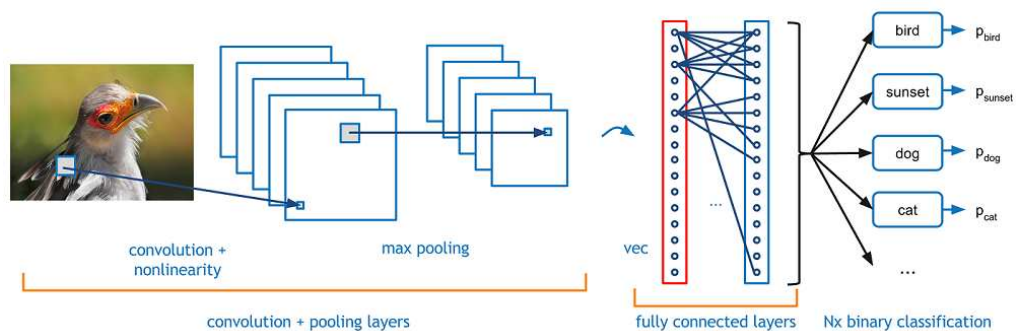
c. Lapisan Fully Connected

Lapisan *fully-connected* menghubungkan seluruh neuron pada lapisan sebelumnya dan setelahnya seperti halnya jaringan syaraf tiruan biasa. Setiap *neuron* dalam lapisan konvolusi harus diubah menjadi representasi data satu dimensi sebelum dihubungkan ke seluruh *neuron*

pada lapisan *fully connected* [10]. Jumlah output (neuron) dalam lapisan *fully connected* biasanya sama dengan jumlah kelas yang diprediksi.

d. Lapisan Output

Lapisan *output* adalah lapisan akhir yang memberikan hasil akhir klasifikasi pada citra [5]. Struktur dari algoritma CNN ditunjukkan pada Gambar 2 [9].



Gambar 2. Struktur Algoritma CNN

[Sumber : Desphande dkk, 2016]

2.1 Arsitektur MobileNet

MobileNet merupakan salah satu arsitektur didalam algoritma Convolutional Neural Network (CNN) yang dirancang khusus untuk perangkat mobile yang memiliki sumber daya komputasi yang terbatas. Tujuan dari MobileNet adalah menyediakan jaringan syaraf tiruan yang ringan sekaligus efisien dalam pengenalan objek yang sehingga dapat diimplementasikan pada smartphone, kamera pintar, dan drone [7]. Perbedaan arsitektur MobileNet dan arsitektur CNN lainnya terletak pada penggunaan lapisan atau layer konvolusi. Lapisan konvolusi pada MobileNet menggunakan depthwise separable convolution.

Layer inti dari arsitektur MobileNet adalah depthwise separable convolution, depthwise separable convolution terdiri dari 2 layer utama yaitu, depthwise convolution dan pointwise convolution [8]. Depthwise convolution menggunakan filter 3x3 sedangkan Pointwise convolution menggunakan filter 1 x 1 yang kemudian digunakan untuk membuat kombinasi linear dari output depthwise convolution. Batchnorm dan fungsi aktivasi ReLU digunakan setelah setiap konvolusi dilakukan. Penggunaan depthwise separable convolution mengurangi komputasi dan ukuran dari model yang dibuat [7], [8], [12], [13]. Arsitektur tubuh MobileNet ditunjukkan pada Tabel 1.

2.2 TensorFlow

Tensorflow adalah perpustakaan perangkat lunak terbuka untuk komputasi numerik yang menggunakan grafik aliran data. Tensorflow dapat menerapkan model *machine learning* di lingkungan yang beragam seperti *browser*, *mikrokontroler*, CPU, GPU, FPGA. Tensorflow sering digunakan di banyak area yang terkait dengan pembelajaran mesin seperti pengenalan suara, pengenalan citra dan pemrosesan bahasa alami [14].

Tabel 1: MobileNet Body Architecture
[Sumber : Howard dkk, 2017]

Type / Stride	Filter Shape	Input Size
Conv / s2	3 x 3 x 3 x 32	224 x 224 x 3
Conv dw / s1	3 x 3 x 32 dw	112 x 112 x 32
Conv / s1	1 x 1 x 32 x 64	112 x 112 x 32
Conv dw / s2	3 x 3 x 64 dw	112 x 112 x 64
Conv / s1	1 x 1 x 64 x 128	56 x 56 x 64
Conv dw / s1	3 x 3 x 128 dw	56 x 56 x 128
Conv / s1	1 x 1 x 128 x 128	56 x 56 x 128
Conv dw / s2	3 x 3 x 128 dw	56 x 56 x 128
Conv / s1	1 x 1 x 128 x 256	28 x 28 x 128
Conv dw / s1	3 x 3 x 256 dw	28 x 28 x 256
Conv / s1	1 x 1 x 256 x 256	28 x 28 x 256
Conv dw / s2	3 x 3 x 256 dw	30 x 28 x 256
Conv / s1	1 x 1 x 256 x 512	14 x 14 x 256
5 x Conv dw / s1	3 x 3 x 512 dw	14 x 14 x 512
5 x Conv / s1	1 x 1 x 512	14 x 14 x 512
Conv dw / s2	3 x 3 x 512 dw	14 x 14 x 512
Conv / s1	1 x 1 x 512 x 1024	7 x 7 x 512
Conv dw / s2	3 x 3 x 1024 dw	7 x 7 x 1024
Conv / s1	1 x 1 x 1024 x 1024	7 x 7 x 1024
Avg Pool / s1	Pool 7 x 7	7 x 7 x 1024
FC / s1	1024 x 1000	1 x 1 x 1024
Softmax / s1	Classifier	1 x 1 x 1000

3 METODE PENELITIAN

3.1 Jenis Data dan Sumber Data

Pada penelitian ini jenis data yang digunakan adalah jenis data kualitatif. Penulis menggunakan 5 jenis sparepart yaitu citra busi, kampas rem, kiprok, laher dan rumah roller yang diambil dari hasil pencarian menggunakan media google gambar, setiap kategori terdiri dari 17 gambar dengan total citra sebanyak 85 citra.

3.2 Metode Pengumpulan Data

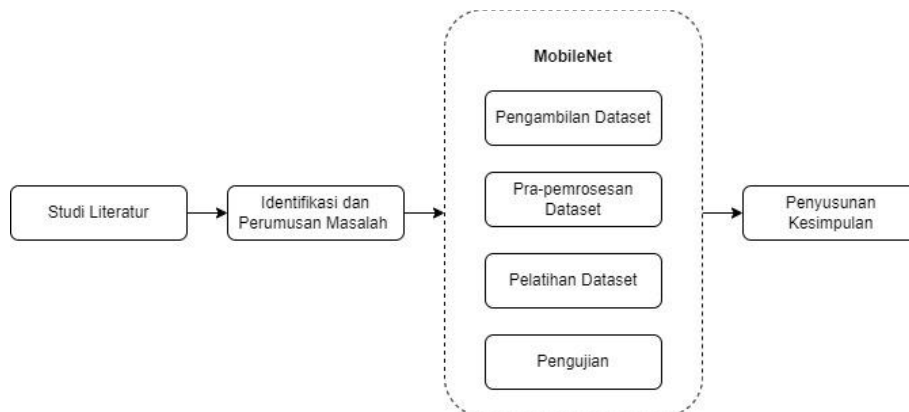
Pengumpulan data dilakukan dengan cara observasi dan studi dokumen. Data yang dikumpulkan oleh penulis berupa citra masing-masing sparepart motor sebanyak 17 citra busi, 17 citra kampas rem, 17 citra kiprok, 17 citra laher dan 17 citra rumah roller dengan total sebanyak 85 citra. Proses pengambilan data citra diperoleh dari beberapa sumber yang didapatkan dari basis data yang tersedia untuk publik yaitu Google gambar. Contoh dataset ditunjukkan pada Gambar 3.



Gambar 3. Contoh Dataset

3.3 Alur Penelitian

Penelitian ini dilakukan dalam beberapa tahap. Tahapan penelitian dimulai dari studi literatur, identifikasi dan perumusan masalah, pengambilan dataset, pra pemrosesan dataset, pelatihan dataset, pengujian dan penyusunan kesimpulan diakhir penelitan. Tahapan penelitian yang dilakukan dapat dilihat pada Gambar 4.



Gambar 4. Tahapan Penelitian

1) Studi Literatur

Studi literatur dilakukan untuk mendapatkan informasi awal terkait topik penelitian yang dipilih dalam makalah-makalah di jurnal, seperti “Implementasi Deep Learning Pada Identifikasi Jenis Tumbuhan Berdasarkan Citra Daun Menggunakan Convolutional Neural Network” [4], “Implementasi Convolutional Neural Network Untuk Klasifikasi Tanaman Rimpang Secara Virtual” [7] dan “Pengenalan Gambar Botol Plastik dan Kaleng Minuman Menggunakan Metode Convolutional Neural Network [15].

2) Identifikasi dan Perumusan Masalah

Pada tahap ini penulis menyusun rumusan masalah yang menjadi fokus permasalahan yang dikaji dalam penelitian ini, yaitu bagaimana membuat model CNN dengan arsitektur MobileNet dan bagaimana tingkat akurasi yang dihasilkan.

3) Pengambilan Dataset

Pengumpulan data dilakukan dengan menggunakan citra yang dikumpulkan melalui media google gambar. Terdapat 5 jenis *sparepart* yang digunakan sebagai data latih. Pemilihan jenis *sparepart* berdasarkan 5 *sparepart* yang paling mudah rusak yaitu, rumah roller, laher, kampas rem, kiprok, busi.

4) Pra-pemrosesan Dataset

Pada tahap ini citra yang telah dikumpulkan yang terdiri dari 5 jenis *sparepart* akan diubah ukurannya dan diberi label untuk digunakan sebagai data latih. Selain itu digunakan teknik *augmentasi* data berupa rotasi, pergeseran dan pencerminan citra untuk memperbanyak data latih.

5) Pelatihan Dataset

Pada tahap ini dataset citra *sparepart* digunakan untuk proses pelatihan dan klasifikasi menggunakan arsitektur MobileNet. Pada arsitektur ini pelatihan dataset dimulai dengan lapisan konvolusi pertama diikuti dengan 13 lapisan *depthwise separable convolution*, proses *pooling* dan lapisan *output*.

6) Pengujian

Pada tahap ini proses pengujian dilakukan terhadap model yang telah dibuat dengan menggunakan 20 data citra. Sejumlah sepuluh citra diambil dari google gambar sebagai data asli dengan rincian masing-masing sebanyak dua citra per kategori. Kemudian sepuluh citra lainnya diperoleh dari proses *motion blur* dengan nilai derajat 359 terhadap citra yang diperoleh dari data asli. Proses *blur* dilakukan menggunakan *tools* PINETOOLS [15]. Kemudian dilakukan perhitungan akurasi menggunakan rumus pada Persamaan 1.

$$\text{Akurasi} = \frac{\text{Jumlah prediksi benar}}{\text{Jumlah data yang diuji}} \times 100\% \quad (1)$$

7) Penyusunan Kesimpulan

Penarikan kesimpulan diambil dari hasil pengujian yang dilakukan menggunakan 10 citra tanpa proses blur dan 10 citra yang melewati proses blur. Dari hasil pengujian tersebut kemudian dilihat bagaimana tingkat akurasi yang dihasilkan oleh model dan apakah terdapat perbedaan yang signifikan dalam memprediksi citra blur dan citra yang tidak blur.

4 HASIL DAN PEMBAHASAN

4.1 Dataset dan Pra Pemrosesan Dataset

Dataset yang telah disiapkan sebelumnya sebanyak 85 citra dimuat dan dikategorikan kedalam lima kelas yang berbeda. Citra-citra tersebut dipisahkan berdasarkan folder yang masing-masing memiliki 17 citra yang berbeda yang berkaitan dengan kelas tersebut. Dari 17 citra tersebut dipisahkan lagi ke dalam 3 dataset yaitu, dataset yang digunakan untuk pelatihan, dataset untuk melakukan validasi, dan dataset untuk melakukan percobaan terhadap model. Dataset pelatihan terdiri dari 10 citra, dataset validasi terdiri dari 5 citra dan dataset percobaan terdiri dari 2 citra untuk masing-masing jenis komponen. Contoh pelabelan dataset ditunjukkan pada Gambar 5.



Gambar 5. Dataset Yang Telah Diberi Label

Seluruh citra yang diambil saat proses pengambilan dataset memiliki ukuran yang berbeda-beda oleh karena itu, pada tahap ini data citra tersebut dilakukan *resize* ke dimensi 224 x 224 piksel dengan 3 channel RGB sebagai lapisan input.

4.2 Augmentasi Data

Augmentasi data merupakan proses memodifikasi atau memanipulasi suatu citra untuk memberikan variasi tambahan dalam dataset pelatihan agar mesin dapat belajar dari berbagai citra yang berbeda untuk mencegah terjadinya *overfitting* [25]. Contoh augmentasi terhadap data citra adalah mengubah ukuran citra, membalik, memutar dan pencahayaan. Operasi augmentasi yang dilakukan adalah *rotation_range* sebesar 20, *width_shift_range* sebesar 0.2, *height_shift_range* sebesar 0.2 dan *horizontal_flip*. Contoh citra hasil augmentasi dapat dilihat pada Gambar 6.



Gambar 6. Contoh Hasil Augmentasi Citra

4.3 Implementasi Model MobileNet

Penelitian ini menggunakan model MobileNet yang telah disediakan oleh TensorFlow atau Keras. Model yang dibangun pada penelitian ini menggunakan 13 lapisan *depthwise separable convolution* untuk melakukan ekstraksi fitur pada citra. Setelah proses ekstraksi fitur selesai lapisan berikutnya adalah *global average pooling* untuk menghitung rata-rata dari setiap *channel* fitur. Ini mengurangi dimensi dari *feature map* menjadi vektor satu dimensi. Selanjutnya lapisan *fully connected* digantikan dengan lapisan *dense* untuk melakukan klasifikasi. Jumlah

neuron di lapisan dense disesuaikan dengan jumlah kelas yang diprediksi. Jumlah *neuron* yang digunakan pada penelitian ini sebanyak 5 *neuron*. Fungsi aktivasi yang digunakan pada lapisan dense untuk melakukan klasifikasi terakhir menggunakan fungsi aktivasi softmax.

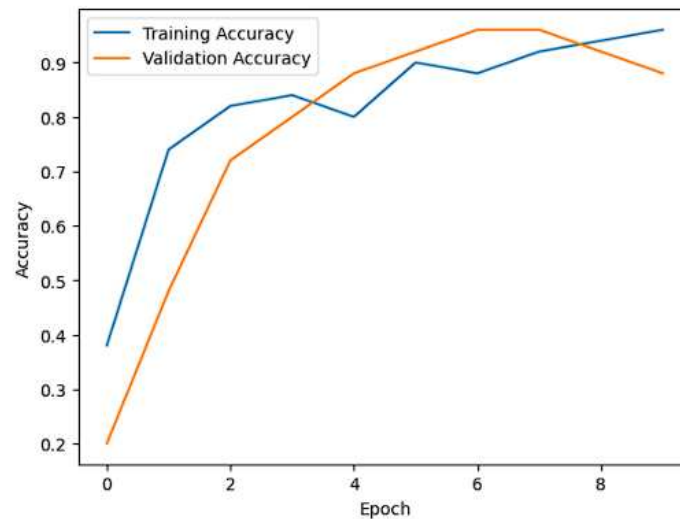
4.4 Pelatihan

Pelatihan model membutuhkan beberapa parameter diantaranya adalah batch pelatihan, batch validasi dan jumlah epoch. Batch pelatihan dan batch validasi merupakan data yang telah diproses pada tahap pra pemrosesan dataset. *Batch* pelatihan digunakan untuk melatih dataset sedangkan batch validasi digunakan untuk mengevaluasi kinerja model selama proses pelatihan berlangsung. Adapun nilai epoch yang digunakan sebanyak sepuluh epoch. proses untuk memperbarui bobot dan bias untuk mengurangi *loss* secara keseluruhan atau yang dikenal dengan *backpropagation* menggunakan adam *optimizer* [16] dengan *learning rate* sebesar 0.0001. Penggunaan optimasi Adam dipilih karena optimasi ini dapat menyesuaikan *learning rate* secara adaptif. Fungsi *loss* yang digunakan adalah *categorical cross entropy* karena data pelatihan berbentuk kategori sehingga pengevaluasian *loss* dilakukan berdasarkan kategori data dan metrik untuk pengukuran performa atau kinerja model menggunakan *accuracy*. *Accuracy* merupakan metrik yang mengukur sejauh mana model memberikan prediksi yang benar dibandingkan dengan total jumlah sampel. Semakin kecil *loss* nya semakin baik model dan semakin baik tingkat akurasinya seperti pada Tabel 2.

Tabel 2: Hasil Loss Dan Akurasi

epoch	loss	accuracy	val_loss	val_accuracy
1	1.761039615	0.379999995	2.68641305	0.200000003
2	0.9184165	0.74000001	1.640923977	0.479999989
3	0.566094339	0.819999993	0.777953327	0.720000029
4	0.503638983	0.839999974	0.512393534	0.800000012
5	0.635247946	0.800000012	0.33697778	0.879999995
6	0.437191099	0.899999976	0.303214401	0.920000017
7	0.393525481	0.879999995	0.205244243	0.959999979
8	0.28216207	0.920000017	0.181724593	0.959999979
9	0.338273406	0.939999998	0.216687381	0.920000017
10	0.186282173	0.959999979	0.189365685	0.879999995

Hasil yang didapatkan setelah melakukan pelatihan terhadap dataset citra komponen sepeda motor menghasilkan nilai akurasi tertinggi sebesar 0.96 pada epoch ke sepuluh untuk hasil data training dan 0.96 pada epoch ke tujuh dan epoch ke delapan untuk hasil data validasi. Grafik akurasi model dapat dilihat pada Gambar 7.

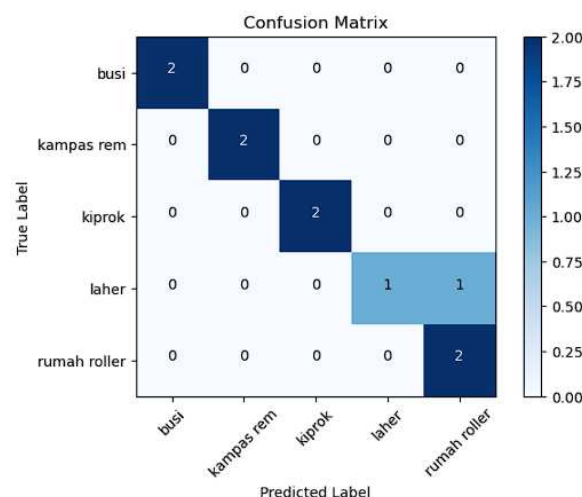


Gambar 7. Hasil Akurasi Model

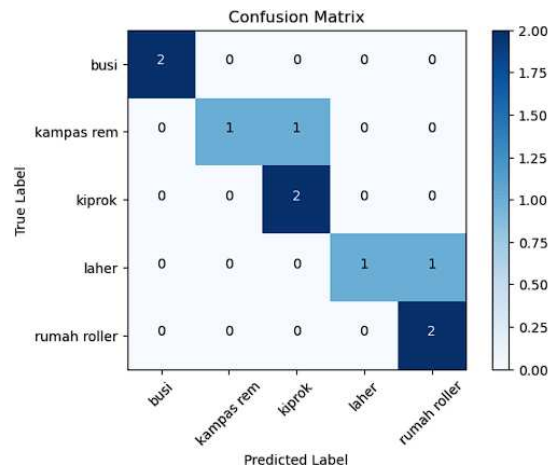
4.5 Pengujian

Pengujian dilakukan menggunakan 20 data uji yang 10 diantaranya diterapkan efek *motion blur* untuk mengaburkan citra. Efek *motion blur* digunakan untuk melihat apakah model mampu mengenali citra dalam kondisi citra yang ditangkap kurang jelas kemudian di plot kedalam confusion matrix.

- Hasil *confusion matrix* menggunakan data uji sebanyak 10 buah citra tanpa proses blur menunjukan bahwa 9 dari 10 data uji berhasil diklasifikasikan dengan benar. Adapun terdapat kekeliruan pada klasifikasi terhadap citra laher yang ditunjukkan pada Gambar 8.

Gambar 8. *Confusion Matrix* Untuk Citra Tanpa Proses Motion Blur

- Hasil *confusion matrix* menggunakan data uji sebanyak 10 buah citra dengan proses blur menunjukan bahwa 8 dari 10 data uji berhasil diklasifikasikan dengan benar. Adapun terdapat kekeliruan pada klasifikasi terhadap citra kampus rem dan laher masing-masing sebanyak satu citra yang ditunjukkan pada Gambar 9.

Gambar 9. *Confusion Matrix* Untuk Citra Blur

Berdasarkan hasil pengujian, akurasi yang diperoleh dapat dihitung sebagai berikut:

$$\text{Akurasi percobaan 1} = \frac{19}{20} \times 100\% = 95\%$$

$$\text{Akurasi percobaan 2} = \frac{18}{20} \times 100\% = 90\%$$

5. KESIMPULAN

Berdasarkan hasil pembuatan dan pengujian model CNN dengan menggunakan arsitektur MobileNet yang dilakukan pada penelitian ini, maka dapat ditarik kesimpulan sebagai berikut: Penelitian ini berhasil membuat model CNN menggunakan arsitektur MobileNet yang mampu mengklasifikasikan citra komponen sepeda motor. Pengujian model mobilenet dilakukan dengan menggunakan 20 data uji. Dari hasil pengujian diperoleh hasil bahwa kinerja akurasi model CNN dengan arsitektur MobileNet dalam mengklasifikasikan citra komponen sepeda motor sebesar 95% untuk citra tanpa proses blur dan 90% untuk citra yang blur. Hasil pengujian menunjukkan tingkat ketepatan klasifikasi citra menggunakan citra yang diblur tidak menurun secara signifikan.

DAFTAR PUSTAKA

- [1] S. H. I. Baik, "Jadi Negara Pengguna Motor Terbanyak ke-3, Berapa Jumlah Penduduk Indonesia?," *sohib.indonesiabaik.id*, 2023. <https://sohib.indonesiabaik.id/article/negara-pengguna-motor-terbanyak-L8Xnt> (accessed May 09, 2023).
- [2] F. Felix, J. Wijaya, S. P. Sutra, P. W. Kosasih, and P. Sirait, "Implementasi Convolutional Neural Network Untuk Identifikasi Jenis Tanaman Melalui Daun," *J. SIFO Mikroskil*, vol. 21, no. 1, pp. 1–10, 2020, doi: 10.55601/jsm.v21i1.672.
- [3] S. R. Suartika E. P, I Wayan, Wijaya Arya Yudhi, "Klasifikasi Citra Menggunakan Convolutional Neural Network (Cnn) Pada Caltech 101," *J. Tek. ITS*, vol. 5, no. 1, p. 76, 2016, [Online]. Available: <http://repository.its.ac.id/48842/>
- [4] F. Sultana, A. Sufian, and P. Dutta, "Advancements in Image Classification using Convolutional Neural Network," May 2019, doi: 10.1109/ICRCICN.2018.8718718.
- [5] E. Chen, X. Wu, C. Wang, and Y. Du, "Application of improved convolutional neural network in image classification," *Proc. - 2019 Int. Conf. Mach. Learn. Big Data Bus.*

- Intell. MLBDBI 2019*, pp. 109–113, 2019, doi: 10.1109/MLBDBI48998.2019.00027.
- [6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks", [Online]. Available: <http://code.google.com/p/cuda-convnet/>
 - [7] A. M. S. Darmatasia, "Implementasi Convolutional Neural Network Untuk Klasifikasi Tanaman Rimpang Secara Virtual," *J. INSTEK (Informatika Sains dan Teknol.*, vol. 8, no. 1, pp. 122–131, 2023.
 - [8] A. G. Howard *et al.*, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," 2017, [Online]. Available: <http://arxiv.org/abs/1704.04861>
 - [9] A. Deshpande, "A Beginner's Guide To Understanding Convolutional Neural Networks," *Github*, 2016. <https://adeshpande3.github.io/A-Beginner%27s-Guide-To-Understanding-Convolutional-Neural-Networks/>
 - [10] R. Valentina, S. Rostianingsih, A. N. Tjondrowiguno, and J. S. Surabaya, "Pengenalan Gambar Botol Plastik dan Kaleng Minuman Menggunakan Metode Convolutional Neural Network," *J. Infra*, vol. 8, no. 1, pp. 249–254, 2020.
 - [11] I. Vasilev, D. Slater, G. Spacagna, P. Roelants, and V. Zocca, *Python Deep Learning: Exploring deep learning techniques, neural network architectures and GANs with PyTorch, Keras and TensorFlow*, 2nd Ed. Birmingham: Packt Publishing Ltd., 2019. [Online]. Available: <https://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=2002295&site=eds-live>
 - [12] M. R. Islam, N. Tasnim, and S. B. Shuvo, "MobileNet Model for Classifying Local Birds of Bangladesh from Image Content Using Convolutional Neural Network," *2019 10th Int. Conf. Comput. Commun. Netw. Technol. ICCCNT 2019*, pp. 1–4, 2019, doi: 10.1109/ICCCNT45670.2019.8944403.
 - [13] W. Sae-Lim, W. Wettayaprasit, and P. Aiyarak, "Convolutional Neural Networks Using MobileNet for Skin Lesion Classification," *JCSSE 2019 - 16th Int. Jt. Conf. Comput. Sci. Softw. Eng. Knowl. Evol. Toward Singul. Man-Machine Intell.*, pp. 242–247, 2019, doi: 10.1109/JCSSE.2019.8864155.
 - [14] J. D. Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, M. K. Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, P. T. Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, and G. B. Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng, "TensorFlow: A System for Large-Scale Machine LearningNo Title," in *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI '16)*, 2016, pp. 265–283.
 - [15] pinetools.com, "PINETOOLS." [Online]. Available: <https://pinetools.com/>
 - [16] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, pp. 1–15, 2015.