

KLASIFIKASI KONDISI BAN KENDARAAN MENGGUNAKAN ARSITEKTUR VGG16

Ahmad Fudoli Zaenun Nazhirin^{1*}; Muhammad Rafi Muttaqin²; Teguh Iman Hermanto³

Program Studi Teknik Informatika^{1,2,3}

STT Wastukencana Purwakarta^{1,2,3}

<https://www.stt-wastukencana.ac.id/>^{1,2,3}

ahmadfudoli08@wastukencana.ac.id^{1*}; rafi@wastukencana.ac.id²; teguhiman@wastukencana.ac.id³;



Abstract— Tyres are the main component that a vehicle needs to work with reducing vibration due to uneven road surfaces, protecting the wheels from wear to provide stability between the vehicle and the ground helping to improve acceleration to facilitate travel while driving. Wear ensures stability between the vehicle and the ground helps improve acceleration for easy movement and driving. Caused including components that are often used, tires can experience damage such as the appearance of cracks in the tires. Cracks in tires can be triggered by factors such as age or the cause of the road that has been exceeded. Detection of tire cracks at this time is still carried out conventionally, where users see directly the state of the tire whether the tire is in good condition or cracked. Conventional methods are important because they maintain tire quality and rider safety. The Conventional Method certainly has weaknesses because vehicle users must have good vision and the ability to distinguish normal tires or cracked tires, but this method is considered less effective because it still uses human labor, causing the risk of human error (human negligence) which can hinder the process of identifying tire cracks. Based on this problem, this study will develop a deep learning model that can classify cracked tires using the VGG16 architecture. In this study, the model was created using 8 scenarios by changing the value of epochs, to get the best parameters in making the model. The results of the 8 scenarios carried out in this study are the best scenario obtained in scenarios 1,3,4 which get 98% accuracy in model testing.

Keywords: Tyres, Deep Learning, VGG16

Abstrak—Ban adalah komponen utama yang diperlukan kendaraan untuk bekerja mengurangi getaran akibat permukaan jalan yang tidak rata, melindungi roda dari keausan untuk memberikan stabilitas antara kendaraan dan tanah membantu meningkatkan akselerasi agar memudahkan perjalanan saat mengemudi. pakai memastikan stabilitas antara kendaraan dan tanah membantu meningkatkan akselerasi untuk memudahkan pergerakan dan mengemudi. Disebabkan termasuk komponen yang sering digunakan ban dapat mengalami kerusakan seperti munculnya retak pada ban. Keretakan pada ban dapat dipicu oleh faktor seperti usia atau penyebab jalan yang telah terlampaui. Deteksi pada retak ban pada saat ini masih dilakukan secara Konvensional, dimana pengguna melihat langsung keadaan ban apakah ban dalam keadaan bagus atau retak. Metode konvensional menjadi hal penting karena untuk menjaga kualitas ban dan keselamatan pengendara. Metode Konvensional tentunya memiliki kelemahan karena pengguna kendaraan harus memiliki penglihatan yang baik dan kemampuan dalam membedakan ban normal atau ban retak tetapi metode ini dinilai kurang efektif karena masih menggunakan tenaga manusia sehingga menimbulkan resiko *human error* (kelalaian manusia) yang dapat menghambat proses pengidentifikasian retak ban. Berdasarkan permasalahan ini maka pada penelitian ini menggunakan sebuah model *deep learning* yang dapat melakukan klasifikasi pada ban yang mengalami retak dengan menggunakan arsitektur VGG16. Pada penelitian ini model yang dibuat menggunakan 8 skenario dengan mengubah nilai epoch, untuk mendapatkan parameter terbaik dalam pembuatan model. Hasil dari 8 skenario yang dilakukan pada penelitian ini adalah skenario terbaik didapatkan pada skenario 1,3,4 dimana mendapatkan akurasi 98% pada pengujian model.

Kata kunci: Ban, Deep Learning, VGG16

PENDAHULUAN

Ban adalah komponen utama yang diperlukan kendaraan untuk bekerja mengurangi getaran akibat permukaan jalan yang tidak rata, melindungi roda dari keausan untuk memberikan stabilitas antara kendaraan dan tanah membantu meningkatkan akselerasi agar memudahkan perjalanan saat mengemudi. Pakai memastikan stabilitas antara kendaraan dan tanah membantu meningkatkan akselerasi untuk memudahkan pergerakan dan mengemudi (Ufriandi, 2021). Disebabkan termasuk komponen yang sering digunakan ban dapat mengalami kerusakan seperti munculnya retak pada ban. Keretakan pada ban dapat dipicu oleh faktor seperti usia atau penyebab jalan yang telah terlampaui (Arthono & Permana, 2022). Berdasarkan Soerjanto Tjahjono, Ketua Komite Nasional Keselamatan Transportasi menyatakan bahwa "80% kecelakaan di jalan raya disebabkan masalah ban" (Shidiq et al., 2022). Tanpa penanganan yang tepat, ban rusak dapat menjadi faktor penyebab kecelakaan (Ruusen et al., 2021). Deteksi pada retak ban pada saat ini masih dilakukan secara konvensional, dimana pengguna melihat langsung keadaan ban apakah ban dalam keadaan bagus atau retak.

Metode konvensional menjadi hal penting karena untuk menjaga kualitas ban dan keselamatan pengendara. Metode konvensional tentunya memiliki kelemahan karena pengguna kendaraan harus memiliki penglihatan yang baik dan kemampuan dalam membedakan ban normal atau ban retak tetapi metode ini dinilai kurang efektif karena masih menggunakan tenaga manusia sehingga menimbulkan resiko *human error* (kelalaian manusia) yang dapat menghambat proses pengidentifikasian retak ban. Kemajuan teknologi ini telah membawa dampak dan manfaat bagi kehidupan masyarakat, terutama pada Klasifikasi Citra (Shidiq et al., 2022).

Klasifikasi citra telah diterapkan dalam mengidentifikasi otomatis pada masalah - masalah yang ada (Putra, Kade Bramasta Vikana et al., 2021). Tujuan dari klasifikasi citra ban sendiri adalah untuk mengklasifikasikan input citra ban ke dalam label retak atau bagus. Klasifikasi citra dengan input citra banyak tentunya menjadi satu pekerjaan sulit untuk dipelajari komputer, karena komputer hanya melihat citra berupa deretan nilai piksel dan data piksel saja, oleh karena itu diperlukan Model pembelajaran dalam proses deteksi otomatis retak pada ban. Sehingga kemajuan bidang pengolahan citra saat ini maka *deep learning* yang merupakan bagian dari bidang pengolahan citra banyak digunakan sebagai solusi mempermudah dalam melakukan pengecekan serta mengurangi kemungkinan *human error* pada pengecekan ban (Puspitasari, 2020).

Penelitian mengenai identifikasi objek menggunakan *deep learning* sudah banyak dilakukan, salah satunya penelitian yang berjudul "Klasifikasi Kualitas Buah Salak dengan *Transfer Learning* Arsitektur VGG16", Penelitian bertujuan untuk membedakan kualitas buah salak telah dilakukan. Salak dibagi menjadi dua kelas, yaitu kelas bagus dan kelas jelek. Salak dengan kualitas jelek merupakan salak yang terlalu matang, mengandung cacat busuk, dan mengandung cacat robek. Jumlah data yang digunakan dalam penelitian ini adalah 370 data, yang terdiri dari 190 salak kelas bagus, dan 180 salak kelas jelek. Pelatihan dilakukan menggunakan *Transfer Learning* dengan arsitektur VGG16. Dalam melakukan pelatihan, parameter yang dirubah adalah *learning rate*, epoch dan momentum. Hasil penelitian menunjukkan, arsitektur ini mampu mendapatkan akurasi tertinggi sebesar 95,83% dengan menggunakan *learning rate* = 0,0001 dan momentum = 0,9. Jika menggunakan momentum, maka akurasi terbaik didapatkan dengan menggunakan *learning rate* = 0,001. Penelitian selanjutnya dapat dilakukan untuk meneliti metode-metode *Transfer Learning* lain untuk pengenalan citra ini (Rismiyati & Luthfiarta, 2021).

Penelitian yang berjudul "*Classification of Tobacco Leaf Pests Using VGG16 Transfer Learning*" meneliti *deep learning* yang menggunakan arsitektur VGG16 dengan objek daun tembakau. Dataset yang digunakan berjumlah 500 gambar dan dibagi menjadi tiga kelas yaitu *Glassy*, *Greenspot*, dan . Hasil dari penelitian ini mendapatkan akurasi terbesar yaitu 99% dan *precision* sebesar 99%, penelitian tersebut dapat dijadikan acuan dalam penggunaan arsitektur karena mendapatkan hasil yang bagus (Swasono et al., 2019).

Dan Penelitian yang dilakukan oleh Rozaqi dengan judul "*Implementation of Transfer Learning in the Convolutional Neural Network Algorithm for Identification of Potato Leaf Disease*" Penelitian ini dilakukan dengan beberapa model yaitu VGG16, Inception-V3 dan ResNet, dataset yang digunakan 3 kategori, per kategori memiliki 150 gambar. Proses Klasifikasi mendapatkan akurasi tertinggi sebesar 95%. Penelitian ini menyimpulkan bahwa arsitektur VGG16 mendapatkan hasil yang lebih baik (Rozaqi et al., 2021).

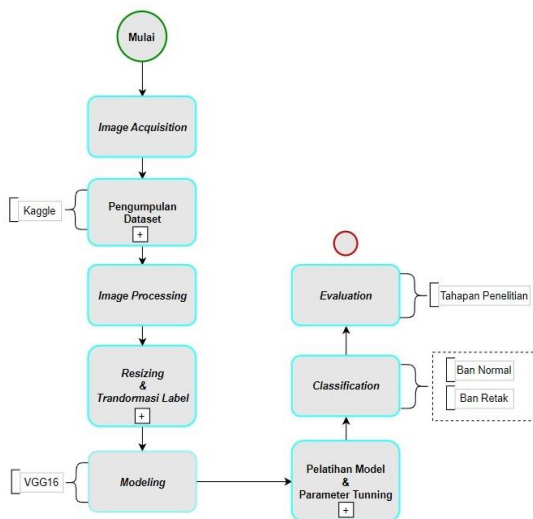
Selanjutnya penelitian dengan judul "*Tire Defect Detection Using Fully Convolutional Network*" oleh Wang, Penelitian ini melakukan klasifikasi pada citra mobil menghasilkan akurasi terbaik 79,91% dari 5 arsitektur yang diuji. Namun pada penelitian tersebut arsitektur *deep learning* yang digunakan adalah VGG16 yang mana proses pelatihannya memerlukan waktu yang lebih banyak dikarenakan VGG16 terdiri dari 5 block CNN (Wang et al., 2019).

VGG16 adalah arsitektur *deep learning* yang terkenal dengan kinerja tinggi dalam pengenalan objek dalam gambar. Arsitektur VGG16 terdiri dari 16 lapisan konvolusi dan lapisan penggabungan maksimum di antara mereka. Lapisan penggabungan maksimum memungkinkan jaringan untuk mengambil informasi penting dari seluruh citra. Arsitektur VGG16 adalah salah satu arsitektur *deep learning* terkenal yang sering digunakan untuk pengenalan objek dalam gambar. Arsitektur VGG16 terdiri dari 16 lapisan konvolusi dan lapisan penggabungan maksimum di antara mereka, memberikan kinerja yang baik pada tugas-tugas pengenalan objek dan klasifikasi gambar (Pramana et al., 2020).

Berdasarkan uraian diatas, penelitian ini bertujuan untuk membuat sebuah model yang dapat mengidentifikasi keretakan pada sebuah ban melalui pendekatan *deep learning* dengan menggunakan *transfer learning*. Kemudian metode *Transfer Learning* yang akan digunakan adalah menguji Arsitektur VGG16 untuk menentukan sebuah nilai akurasi terbaik untuk hasil akhir klasifikasi citra. Hasil akhir dari penelitian ini yaitu untuk mengetahui apakah pengujian citra ban dengan arsitektur VGG16 dapat menghasilkan output terbaik dalam mendeteksi keretakan pada ban berdasarkan evaluasi dari model yang dihasilkan.

BAHAN DAN METODE

Metode penelitian disusun seperti pada Gambar 1. berikut :



Sumber : (Nazhirin et al., 2023)

Gambar 1. Metode Penelitian

1. Image Acquisition

Pada *image acquisition* ini akan dilakukan pengumpulan data yang akan digunakan dalam

penelitian saat ini. Dataset yang akan digunakan dalam penelitian saat ini adalah citra ban . Data yang telah dikumpulkan akan diseleksi sesuai dengan kebutuhan dalam penelitian ini, selanjutnya data akan dipisahkan secara manual ke dalam beberapa folder sesuai dengan kategori ban normal dan retak yang akan diteliti. Dataset didapatkan dari website Kaggle.com. Total data citra yang di kumpulkan berjumlah 1028 citra. Dataset yang dikumpulkan berupa gambar citra kondisi ban normal serta ban yang mengalami keretakan pada bagian carcass casing ban. Citra data tersebut memiliki ukuran 24 Bit dan masih memiliki ukuran pixel yang berbeda beda dengan format JPG. Dimana Data citra tersebut akan di ambil untuk pengujian data latih dan *testing* sebanyak 70/30% seperti yang digunakan oleh (Haksoro & Setiawan, 2021). Untuk data latih terdiri dari 703 data dan untuk data uji terdiri dari 325 data masing masing dari data diatas terbagi menjadi kelas normal dan cracked.

2. Image Proccesing

Setelah dilakukan tahap *Image Acquisition* maka dilanjutkan dengan tahap *Image Processing*. Tahap *Image Processing* adalah tahap perubahan citra atau gambar menjadi resolusi lebih rendah. Citra mengalami proses *resize* menjadi ukuran 224x224 pixel dengan 3 channel RGB (Red, Green , Blue) (Prasetyo et al., 2021). Hal ini dikarenakan citra akan dijadikan sebagai input pada model yang dilatih, oleh karena itu model yang diinput harus disesuaikan dengan ukuran tertentu sebagai patokan pelatihan model .

3. Modelling

Lalu pada tahap ini dilakukan tahap *modelling* dengan merancang model dengan menggunakan arsitektur VGG16. Tahap *modelling* dibangun dengan menerapkan macam-macam teknik pemodelan VGG16 serta beberapa parameternya yang kemudian disesuaikan sedemikian rupa agar menghasilkan nilai yang optimal. Proses *modelling* pada dataset.

4. Classification

Dalam tahap ini model yang telah dibuat mulai diuji coba guna mengetahui kemampuan model dalam pengenalan objek pada citra yang menjadi fokus penelitian. Selanjutnya akan dilakukan pengukuran performa tersebut dengan menggunakan *Confusion Matrix* untuk mengetahui apakah model telah berjalan dengan baik atau tidak. Hasil evaluasi akan menggunakan nilai *accuracy*, *precision*, dan *recall* (Damuri et al., 2021).

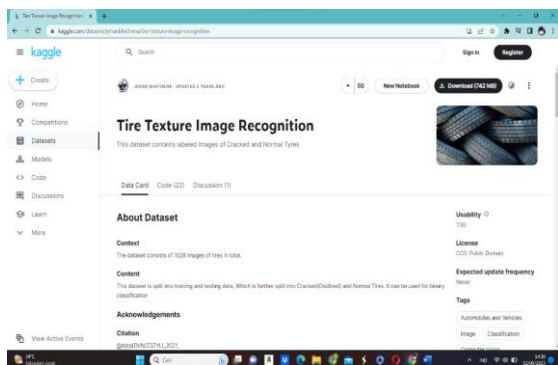
5. Evaluation

Evaluasi merupakan tahap akhir yang bertujuan untuk menentukan apakah tahapan penelitian yang dilakukan sudah mencapai tujuan awal atau tidak berdasarkan hasil yang telah dicapai oleh model .

HASIL DAN PEMBAHASAN

1. Image Acquisition

Image Acquisition adalah langkah di mana tool yang digunakan dalam model juga ditentukan dalam langkah ini. Citra yang digunakan dalam penelitian ini berasal dari Kaggle yang diunggah oleh Jehan Bhathena dari Kuwait. Materi tersebut diunggah pada November 2021. Dataset tersebut berkaitan dengan ban normal dan retak. Sumber dataset yang digunakan dalam penelitian ini ditunjukkan oleh Gambar 2 :



Sumber :

(<https://www.kaggle.com/datasets/jehanbhathena/tire-texture-image-recognition>)

Gambar 2. Sumber Web Kaggle

Terdapat total 1028 citra yang telah dikumpulkan. Dataset citra ini terbagi menjadi dua kategori, yaitu ban dalam keadaan normal dan ban yang mengalami retak. keadaan ban normal ditunjukkan seperti pada gambar 3 sedangkan ban retak ditunjukkan seperti pada gambar 4.



Sumber :

(<https://www.kaggle.com/datasets/jehanbhathena/tire-texture-image-recognition>)

Gambar 3. Ban Normal



Sumber :

(<https://www.kaggle.com/datasets/jehanbhathena/tire-texture-image-recognition>)

Gambar 4. Ban Retak

Berikut ini adalah rincian data yang terlampir dalam tabel 1 untuk referensi lebih lanjut :

Tabel 1. Pembagian Dataset

Jenis data	Sumber data	Data latih	Data testing
Ban normal	Kaggle	376	155
Ban retak	Kaggle	327	210
total		703	325

Sumber : (Nazhirin et al., 2023)

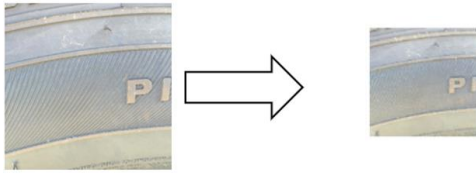
Dari total 1028 data citra yang diperoleh, kemudian dilakukan pemisahan terhadap citra-citra yang memiliki format yang berbeda untuk mengubahnya menjadi format JPG. Setelah data citra dalam format JPG, langkah berikutnya adalah melakukan pengolahan lebih lanjut.

2. Image Processing

Image Processing Tahap pemrosesan gambar dilakukan sebelum melanjutkan ke tahap klasifikasi. Tahap ini sangat penting karena gambar akan menjadi input dalam proses selanjutnya. Mengingat gambar yang diunduh melalui Kaggle memiliki ukuran yang beragam, peneliti perlu melakukan proses *resize* (perubahan ukuran). Ukuran gambar diubah menjadi 224 x 224 piksel dengan format JPG dan tiga channel warna RGB (Red, Green, Blue). Setelah dilakukan tahap *Image Acquisition* maka dilanjutkan dengan tahap *Image Processing*. Tahap *Image Processing* adalah tahap perubahan citra atau gambar menjadi resolusi lebih rendah. Citra mengalami proses *resize* menjadi ukuran 224x224 *pixel* dengan 3 channel RGB (Red, Green , Blue).

Langkah pertama adalah menentukan *library* yang akan digunakan dalam tahap pemodelan. *Library* yang digunakan meliputi: *numpy* untuk manipulasi tabel, *matplotlib* untuk visualisasi data dalam bentuk grafik, dan *Tensorflow* untuk membantu pengembangan model. Untuk

memperlihatkan perbedaan antara gambar sebelum dan setelah proses *resizing* yang awalnya memiliki ukuran 2000×2000 di rubah menjadi 224×224 *pixel* yang dapat dilihat pada Gambar 5.



Sumber : (Nazhirin et al., 2023)

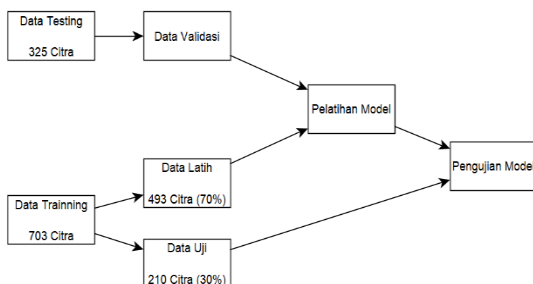
Gambar 5. Proses *Resizing* menjadi 224×224 *pixel*

Setelah perubahan ukuran gambar selesai, langkah selanjutnya adalah mengganti label dengan fungsi biner. Tujuannya adalah untuk melakukan klasifikasi biner (dua kelas), yaitu ketika keluaran model adalah 0 atau 1 (retak dan normal) untuk memudahkan proses pemodelan kedepannya. Langkah berikutnya adalah melakukan transformasi label menggunakan fungsi categorical. Tujuannya adalah untuk mengubah label pada citra menjadi angka 0 dan 1, sehingga memudahkan proses pemodelan selanjutnya. Dataset yang digunakan dalam penelitian ini dibagi dengan perbandingan 70:30, di mana 70% data digunakan sebagai data latih dan 30% data digunakan sebagai data uji.

Data yang diambil dari data latih sebesar 70% akan digunakan untuk melatih model. Pada tahap pelatihan model, digunakan metrik evaluasi berupa akurasi dan loss. Sementara itu, data uji sebesar 30% akan digunakan untuk menguji model yang telah dilatih. Data dari folder pengujian akan digunakan sebagai data validasi untuk menguji kinerja model yang telah dilatih.

Pemilihan pembagian dataset dengan rasio 70:30 dipilih untuk mencapai keseimbangan yang baik pada hasil akhir, serta menghindari terjadinya *overfitting* atau *underfitting* pada model. Skema dataset ini dapat dilihat pada Gambar 6. yang membantu memudahkan pemahaman mengenai pengaturan data.

3. Modelling



Sumber : (Hendrawan, 2022)

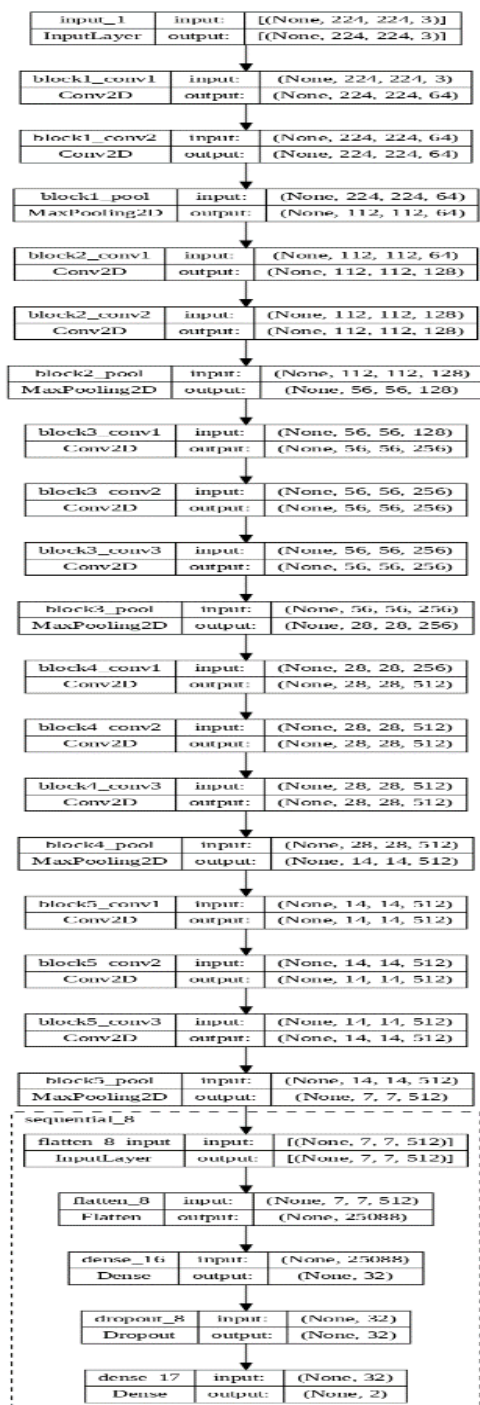
Gambar 6. Skema Pengguna Dataset

Pada tahap ini, dilakukan perancangan model untuk melakukan klasifikasi pada ban yang mengalami keretakan menggunakan Google Colab. Algoritma yang digunakan untuk pembuatan model adalah *Convolutional Neural Network (CNN)* yang memanfaatkan teknik *Transfer Learning* Arsitektur VGG16, yang merupakan sebuah jenis jaringan syaraf konvolusi.

Dalam tahap pemodelan ini, citra hasil pengolahan akan dimasukkan ke dalam dataset yang terdiri dari data latih (*train*) dan data uji (*test*). Pembuatan model dilakukan menggunakan *Tensorflow*, sebuah *library deep learning* yang dikembangkan oleh Google untuk memudahkan pembuatan model. API keras digunakan di atas platform *Tensorflow* untuk mempercepat komputasi model. Model arsitektur yang dibangun akan terdiri

- a). *Flatten Layer* : Digunakan untuk mengubah output dari model VGG16 menjadi vektor satu dimensi sebelum masuk ke *layer Dense*. *Input shape* dari layer ini mengikuti *output shape* dari model VGG16.
- b). *Dense Layer* : Layer ini memiliki 32 unit dengan fungsi aktivasi ReLU. Fungsi aktivasi ReLU digunakan untuk menambahkan non-linearitas pada output layer sebelumnya.
- c). *Dropout Layer*: Digunakan untuk mencegah *overfitting* dengan mematikan secara acak sebagian unit pada layer sebelumnya. *Dropout rate* yang digunakan adalah 0.2, yang berarti 20% unit pada layer sebelumnya akan dimatikan secara acak selama proses *training*.
- d). *Dense Layer*: Layer ini memiliki 2 unit dengan fungsi aktivasi *softmax*. Fungsi aktivasi *softmax* digunakan untuk menghasilkan probabilitas kelas *output* (dalam hal ini 2 kelas).
- e). *Model Layer*: Digunakan untuk menggabungkan model VGG16 dengan *top_model* yang telah dibangun sebelumnya. Model layer ini memiliki input dari model VGG16 dan output dari *top model*.

Selain layer-layer di atas, terdapat juga pengaturan *compile model* yang termasuk dalam arsitektur tersebut, seperti penggunaan *Optimizer Adam* dengan *learning rate* 0.001, loss function categorical crossentropy (untuk tugas klasifikasi multi-kelas), dan metrik akurasi (*accuracy*) untuk evaluasi model. Berikut adalah visualisasi dari arsitektur model VGG16 dari *Optimizer ADAM* dan SGD. Untuk Arsitektur Model VGG16 ditunjukkan pada Gambar 7 :



Sumber : (Nazhirin et al., 2023)
Gambar 7. Arsitektur Model VGG16

Langkah selanjutnya adalah membuat dua fungsi program, termasuk fungsi *Callback* dan fungsi *checkpoint*. Fungsi *Callback* adalah fungsi yang dipanggil secara otomatis oleh program ketika peristiwa atau kondisi tertentu terjadi, sedangkan fungsi *checkpoint* adalah metode yang digunakan dalam pembelajaran mesin untuk mempertahankan versi terbaik dari suatu model selama proses pelatihan. Dengan dua fungsi ini, ini

adalah solusi yang efektif untuk mengikuti dan mengontrol proses pelatihan di tahap akhir secara real-time, sehingga mencapai hasil terbaik.

Salah satu jenis *Callback* yang digunakan dalam proses pemodelan ini adalah menjaga model terbaik selama pelatihan berdasarkan akurasi pelatihan dan validasi data. Model tersimpan terbaik dapat digunakan kembali nanti. Pada model ini, pelatihan model berhenti secara otomatis ketika akurasi pelatihan dan validasi data telah mencapai batas maksimum. Untuk fungsi *loss*, modelnya menggunakan fungsi *loss function* sedangkan untuk metrik, model digunakan sebagai parameter evaluasi. Untuk kode program fungsi *callback* dan *Chekpont* dapat dilihat pada Gambar 8.

```

# Iterate each variance of epochs
for idx, epoch in enumerate(list_epochs):
    # Interface
    print(f"Schema SGB on Epochs: {epoch}")
    # Define checkpoint file model
    checkpoint_path = f"skenario_sgb{idx + 1}/cp_{epoch:04d}.ckpt"
    # Create a callback that saves the model's weights
    cp_callback = tf.keras.callbacks.ModelCheckpoint(filepath=checkpoint_path,
                                                    save_weights_only=True,
                                                    monitor='val_accuracy',
                                                    mode='max',
                                                    save_best_only=True,
                                                    verbose=1)
    # Inherit model
    model = build_model()
    # Calculating computational time
    t = time.perf_counter()
    # Train model
    history = model.fit(
        X_train, y_train,
        epochs=epoch,
        validation_data=(X_test, y_test),
        callbacks=[cp_callback, early_stop])
    
```

Sumber : (Nazhirin et al., 2023)
Gambar 8. Fungsi *Callback* dan *Chekpont*

Langkah selanjutnya adalah melakukan penyetelan parameter tuning pada model. Penyetelan parameter dilakukan untuk mencari nilai parameter terbaik yang dapat digunakan oleh model. Dibawah ini adalah tabel yang memvisualisasikan skenario parameter tuning dan pelatihan model untuk implementasi *deep learning* dengan teknik *Transfer Learning* menggunakan arsitektur VGG16 dan dapat dilihat pada Tabel 2.

Tabel 2. Skenario *Parameter Tuning*

Skenario	Epoch	Optimizer	Learning Rate	Batch Size	Dropout
1	50	ADAM	0,001	16	0,2
2	70	ADAM	0,001	16	0,2
3	90	ADAM	0,001	16	0,2
4	100	ADAM	0,001	16	0,2
5	50	ADAM	0,001	16	0,2
6	70	SGD	0,001	16	0,2
7	90	SGD	0,001	16	0,2
8	100	SGD	0,001	16	0,2

Sumber : (Nazhirin et al., 2023)
Pembahasan Parameter Tuning untuk Klasifikasi Kondisi Ban Kendaraan Menggunakan Arsitektur VGG16 dengan *Optimizer* Adam dan SGD:

- 1) Parameter Tuning dengan *Optimizer* Adam:
 - a. Skenario 1 : Dalam skenario ini, dilakukan pelatihan model dengan menggunakan Arsitektur VGG16, *Optimizer* Adam, Epoch 50, *Batch Size* 16, *Dropout* 0.2, dan berfokus pada klasifikasi kondisi ban kendaraan. Pada akhirnya, model dievaluasi untuk memeriksa

kinerja dan akurasi hasil klasifikasi yang diperoleh.

- b. Skenario 2 : Pada skenario ini, model diperbarui dengan meningkatkan jumlah epoch menjadi 70. Hal ini bertujuan untuk melihat apakah peningkatan jumlah epoch dapat meningkatkan kinerja model dalam mengklasifikasikan kondisi ban kendaraan.
 - c. Skenario 3 : Skenario ini melibatkan pelatihan model dengan Epoch 90, menggunakan *Optimizer Adam*, *Batch Size 16*, *Dropout 0.2*, dan Arsitektur VGG16. Tujuannya adalah untuk menguji apakah peningkatan jumlah epoch dapat menghasilkan peningkatan akurasi yang lebih baik.
 - d. Skenario 4 : Dalam skenario terakhir, model dilatih selama 100 epoch menggunakan *Optimizer Adam*, *Batch Size 16*, *Dropout 0.2*, dan Arsitektur VGG16. Hasil akhirnya akan dievaluasi untuk melihat apakah peningkatan jumlah epoch memberikan manfaat yang signifikan dalam klasifikasi kondisi ban kendaraan.
- 2) Parameter Tuning dengan *Optimizer SGD* :
- a. Skenario 1 : Pada skenario ini, model diperbarui menggunakan *Optimizer SGD*, dengan jumlah epoch sebesar 50. Pelatihan dilakukan dengan Arsitektur VGG16, *Batch Size 16*, *Dropout 0.2*, dan fokus pada klasifikasi kondisi ban kendaraan. Hasil akhirnya akan dievaluasi untuk melihat kinerja model.
 - b. Skenario 2 : Dalam skenario ini, model dilatih selama 70 epoch menggunakan *Optimizer SGD*. Tujuannya adalah untuk melihat apakah peningkatan jumlah epoch dapat meningkatkan kinerja dan akurasi model dalam mengklasifikasikan kondisi ban kendaraan.
 - c. Skenario 3 : Model diperbarui dengan peningkatan jumlah epoch menjadi 90. Pada skenario ini, *Optimizer SGD* digunakan untuk melatih model dengan *Batch Size 16*, *Dropout 0.2*, dan Arsitektur VGG16. Tujuannya adalah untuk mengevaluasi apakah peningkatan jumlah epoch dapat memberikan peningkatan kinerja yang lebih baik.
 - d. Skenario 4 : Skenario terakhir melibatkan pelatihan model selama 100 epoch menggunakan *Optimizer SGD*, *Batch Size 16*, *Dropout 0.2*, dan Arsitektur VGG16. Hasil akhirnya akan dievaluasi untuk melihat manfaat dari peningkatan jumlah epoch dalam klasifikasi kondisi ban kendaraan.

Dalam semua skenario, *Batch Size* tetap 16 dan *Optimizer Dropout* diatur sebesar 0.2 untuk semua pengujian. Evaluasi akhir pada setiap skenario akan

memberikan pemahaman yang lebih baik tentang kinerja dan akurasi model dalam mengklasifikasikan kondisi ban kendaraan menggunakan Arsitektur VGG16 dengan *Optimizer Adam* dan SGD.

Pada tahapan ini, model yang telah dibangun sebelumnya akan diuji menggunakan data uji untuk mengukur akurasinya. Pengujian dilakukan pada 325 data, di mana 110 data digunakan untuk pengujian pada masing-masing kelas dan 210 data digunakan untuk pembuatan *Confusion Matrix* pada setiap kelasnya. Hasil pengujian menunjukkan bahwa pada data normal, model berhasil mengklasifikasikan 324 data sebagai data normal dan hanya 1 data yang salah diklasifikasikan sebagai data tidak normal. Sementara itu, pada data tidak normal, model mampu mengklasifikasikan 323 data dengan benar sebagai data tidak normal. Berikut ini adalah Tabel 3 hasil pengujian model untuk implementasi *deep learning* dengan menggunakan teknik *Transfer Learning* menggunakan arsitektur VGG16 *Optimizer Adam* dan SGD :

Tabel 3. Skenario *Optimizer Adam* dan Sgd

Skenario	Epoch	Loss	Accuracy	Val Loss	Val Accuracy	Optimizer
1	50	0.077	0.973	0.039	0.985	Adam
2	70	0.691	0.535	0.690	0.538	Adam
3	90	0.021	0.998	0.038	0.985	Adam
4	100	0.022	0.999	0.044	0.985	Adam
5	50	0.207	0.922	0.185	0.938	SGD
6	70	0.171	0.947	0.155	0.951	SGD
7	90	0.167	0.952	0.159	0.948	SGD
8	100	0.181	0.932	0.167	0.935	SGD

Sumber : (Nazhirin et al., 2023)

Dalam pengujian model dengan empat skenario yang menggunakan *Optimizer Adam* dan SGD, kita dapat melihat perbedaan kinerja model pada setiap skenario dan epoch yang berbeda.

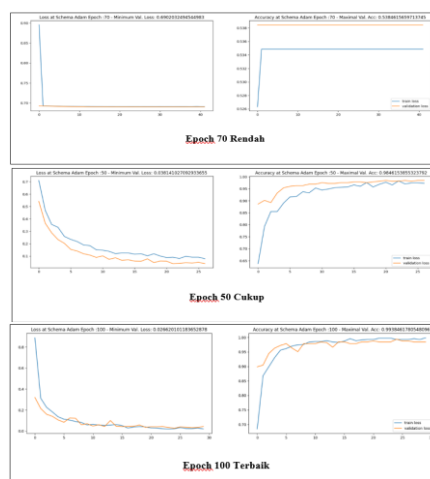
- 1) Pada skenario 1 dengan *Optimizer Adam*, model mencapai tingkat akurasi yang tinggi, yaitu 97.30% pada epoch 50, dengan loss yang relatif rendah. Hal ini menunjukkan kemampuan model dalam mempelajari pola pada data pelatihan. Tingkat akurasi pada data validasi juga tinggi, yaitu 98.46%, yang mengindikasikan kemampuan model untuk secara umum memprediksi dengan benar pada data baru.
- 2) pada skenario 2 dengan *Optimizer Adam*, model menunjukkan performa yang buruk. Tingkat akurasi yang rendah, yaitu 53.49%, pada data pelatihan menunjukkan bahwa model tidak mampu mempelajari pola dengan baik. Tingkat

akurasi pada data validasi juga rendah, yaitu 53.85%, yang menunjukkan ketidakmampuan model untuk melakukan prediksi dengan akurat pada data baru.

- 3) Pada skenario 3 dengan *Optimizer* Adam, model mencapai tingkat akurasi yang sangat tinggi, yaitu 99.72%, pada epoch 90, dengan loss yang rendah. Ini menunjukkan bahwa model berhasil mempelajari pola yang ada dalam data pelatihan. Tingkat akurasi pada data validasi juga tinggi, yaitu 98.46%, menunjukkan kemampuan model untuk melakukan prediksi yang baik pada data baru.
 - 4) Skenario 4 dengan *Optimizer* Adam menunjukkan hasil yang serupa dengan skenario 3. Model mencapai tingkat akurasi yang sangat tinggi, yaitu 99.86%, pada epoch 100, dengan *loss* yang rendah. Tingkat akurasi pada data validasi juga tinggi, yaitu 98.46%. Hasil ini mengindikasikan kemampuan model untuk melakukan prediksi yang akurat pada data baru.
 - 5) Selanjutnya, dalam pengujian model dengan *Optimizer* SGD, hasilnya sedikit berbeda. Pada skenario 1, model mencapai tingkat akurasi yang cukup tinggi, yaitu 92.18%, pada epoch 50, dengan *loss* yang lebih tinggi dibandingkan dengan skenario menggunakan *Optimizer* Adam. Tingkat akurasi pada data validasi juga cukup tinggi, yaitu 93.85%.
 - 6) Pada skenario 2 dengan *Optimizer* SGD, model menunjukkan peningkatan performa. Tingkat akurasi yang lebih tinggi, yaitu 94.74%, pada epoch 70, dengan *loss* yang lebih rendah dibandingkan dengan skenario sebelumnya. Tingkat akurasi pada data validasi juga meningkat, yaitu 95.08%.
 - 7) Skenario 3 dengan *Optimizer* SGD menunjukkan tingkat akurasi yang lebih tinggi, yaitu 95.16%, pada epoch 90, dengan *loss* yang rendah. Tingkat akurasi pada data validasi juga tinggi, yaitu 94.77%, yang mengindikasikan performa yang baik dalam prediksi data baru.
 - 8) Namun, pada skenario 4 dengan *Optimizer* SGD, model menunjukkan sedikit penurunan performa. Tingkat akurasi yang lebih rendah, yaitu 93.17%, pada epoch 100, dengan *loss* yang sedikit lebih tinggi dibandingkan dengan skenario sebelumnya. Tingkat akurasi pada data validasi juga sedikit lebih rendah, yaitu 93.54%.
- Secara keseluruhan, penggunaan *Optimizer* Adam memberikan hasil yang lebih baik dibandingkan dengan SGD dalam hal tingkat akurasi dan kemampuan model untuk mempelajari pola pada data pelatihan dan memprediksi pada data baru. Skenario 1 dan 3 dengan *Optimizer* Adam menunjukkan hasil yang paling baik, sementara skenario 2 dengan *Optimizer* Adam dan skenario 4 dengan *Optimizer* SGD menunjukkan performa yang

paling buruk. Dalam memilih *Optimizer*, perlu mempertimbangkan performa model pada data pelatihan dan data validasi untuk memastikan kemampuan model dalam melakukan prediksi yang akurat pada data baru.

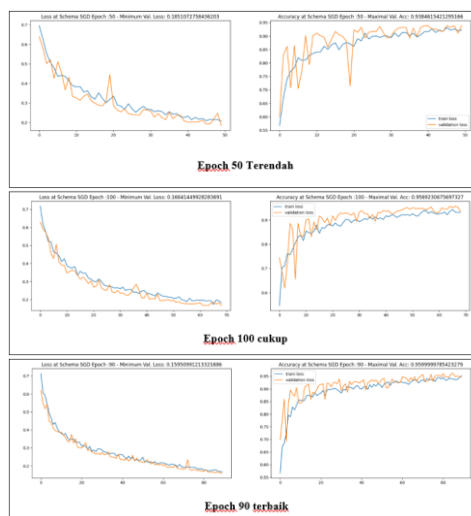
Pemilihan skenario dan jumlah epoch yang tepat serta pemilihan *Optimizer* yang sesuai menjadi faktor penting dalam meningkatkan performa model. Evaluasi lebih lanjut dengan variasi parameter dan dataset yang lebih besar diperlukan untuk memahami performa model secara lebih komprehensif dalam berbagai konteks dan tugas. Dalam grafik visualisasi yang dilampirkan, terdapat kategori yang digunakan untuk menggambarkan akurasi pengujian (*Test Accuracy*) dan kerugian pengujian (*Test Loss*) yaitu kategori rendah, cukup, dan terbaik. untuk visualisasi Grafik dengan *Optimizer* Adam dapat dilihat pada Gambar 9 :



Sumber : (Nazhirin et al., 2023)

Gambar 9. Grafik dengan *Optimizer* Adam

Dan untuk visualisasi Grafik dengan *Optimizer* SGD dapat dilihat pada Gambar 10 :

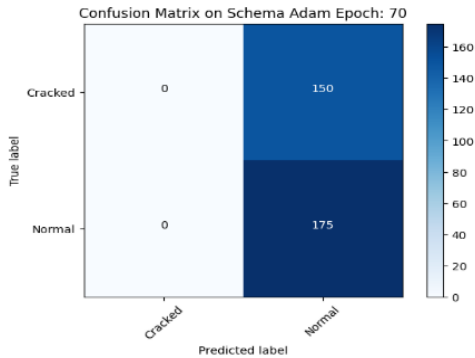


Sumber : (Nazhirin et al., 2023)

Gambar 10. Grafik dengan *Optimizer* Sgd

4. Classification

Pada tahap classification model yang telah selesai dibuat akan diuji untuk mengetahui hasil model yang dibuat model akan menghasilkan nilai *precision*, *recall*, dan *accuracy*, *confussion matriks*. pada visualisasi hasil perhitungan untuk klasifikasi yang dilampirkan, Bebebrapa kategori saja menyesuaikan dengan hasil pengujian yang sudah terlampir diatas. Berikut adalah visualisasinya dapat dilihat pada Gambar 11. Sampai Gambar 18.



Sumber : (Nazhirin et al., 2023)

Gambar 11. Confusion Matrix Optimizer Adam, epoch 70

Dalam hasil *Confusion Matrix* yang diberikan, terdapat beberapa informasi yang dapat digunakan untuk membahas *true/false positive* dan *false negative* :

Optimizer Adam, epoch 70:

Confusion Matrix Optimizer Adam, epoch 70 yang ditunjukkan pada Gambar 11. menunjukan bahwa tidak ada sampel yang terklasifikasi sebagai *Cracked* (class 0) yang berhasil diprediksi dengan benar (*True Positive*). Semua sampel *Cracked* diprediksi sebagai *Normal* (*False Negative*).

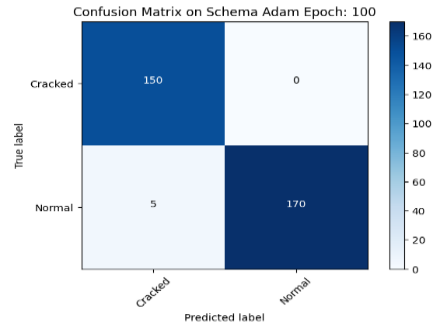
Classification Report:				
	precision	recall	f1-score	support
Cracked	0.00	0.00	0.00	150
Normal	0.54	1.00	0.70	175
accuracy			0.54	325
macro avg	0.27	0.50	0.35	325
weighted avg	0.29	0.54	0.38	325

Sumber : (Nazhirin et al., 2023)

Gambar 12. Klasifikasi *Optimizer Adam*, epoch 70

Hal ini dikonfirmasi oleh *Classification Report* yang menunjukkan *recall* (*true positive rate*) untuk kelas *Cracked* sebesar 0.00, yang ditunjukan pada Gambar 12. menandakan bahwa tidak ada sampel *Cracked* yang berhasil diprediksi dengan benar.

Dalam kesimpulan, model dengan *Optimizer Adam* pada epoch 70 memiliki performa yang buruk dalam mengenali sampel-sampel *Cracked*, dengan tidak ada sampel yang berhasil diprediksi dengan benar.



Sumber : (Nazhirin et al., 2023)

Gambar 13. Confusion Matrix Optimizer Adam, epoch 100

Optimizer Adam, epoch 100:

Confusion Matrix Optimizer Adam, epoch 100 yang ditunjukkan pada Gambar 13. menunjukkan bahwa terdapat 5 sampel yang termasuk ke dalam kelas *Cracked* (*True Positive*), namun salah diklasifikasikan sebagai *Normal* (*False Negative*).

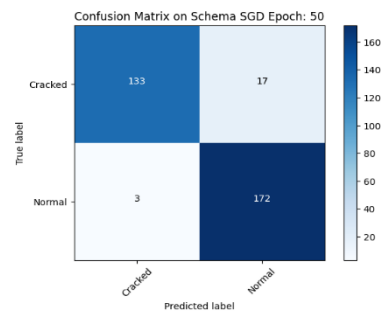
Classification Report:				
	precision	recall	f1-score	support
Cracked	0.97	1.00	0.98	150
Normal	1.00	0.97	0.99	175
accuracy			0.98	325
macro avg	0.98	0.99	0.98	325
weighted avg	0.99	0.98	0.98	325

Sumber : (Nazhirin et al., 2023)

Gambar 14. Klasifikasi *Optimizer Adam*, epoch 100

Hasil prediksi label dari *Classification Report* menunjukkan *recall* sebesar 1.00 yang ditunjukan pada Gambar 14. untuk kelas *Cracked*, menandakan bahwa semua sampel *Cracked* berhasil diprediksi dengan benar.

Dapat disimpulkan bahwa pada epoch 100, model dengan *Optimizer Adam* memiliki performa yang baik dalam mengenali dan memprediksi sampel-sampel *Cracked*, dengan semua sampel *Cracked* berhasil diprediksi dengan benar.



Sumber : (Nazhirin et al., 2023)

Gambar 15. Confusion Matrix Optimizer SGD, epoch 50

Optimizer SGD, epoch 50:

Confusion Matrix Optimizer SGD, epoch 50 yang ditunjukkan pada Gambar 15. menunjukkan bahwa terdapat 17 sampel yang termasuk ke dalam kelas *Cracked* dan salah diklasifikasikan sebagai *Normal*

(*False Positive*), serta terdapat 3 sampel yang termasuk ke dalam kelas Normal dan salah diklasifikasikan sebagai *Cracked* (*False Negative*).

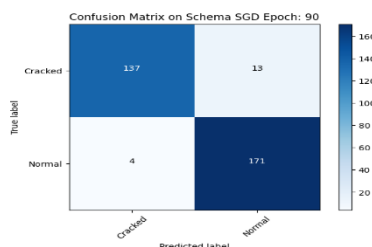
Classification Report:				
	precision	recall	f1-score	support
Cracked	0.98	0.89	0.93	150
Normal	0.91	0.98	0.95	175
accuracy			0.94	325
macro avg	0.94	0.93	0.94	325
weighted avg	0.94	0.94	0.94	325

Sumber : (Nazhirin et al., 2023)

Gambar 16. Klasifikasi *Optimizer* SGD, epoch 50

Classification Report yang ditunjukkan pada Gambar 16. menunjukkan precision sebesar 0.98 untuk kelas *Cracked*, menandakan bahwa sebagian besar sampel yang diprediksi sebagai *Cracked* adalah benar-benar *Cracked*. Namun, *recall* sebesar 0.89 menunjukkan bahwa sebagian dari sampel *Cracked* tidak berhasil terdeteksi dengan baik oleh model.

Dalam kesimpulan, model dengan *Optimizer* SGD pada epoch 50 memiliki performa yang baik dalam memprediksi sampel-sampel *Cracked*, namun terdapat beberapa sampel yang tidak berhasil terdeteksi dengan baik (*False Negative*).



Sumber : (Nazhirin et al., 2023)

Gambar 17. *Confusion Matrix* *Optimizer* SGD, epoch 90

Optimizer SGD, epoch 90:

Confusion Matrix *Optimizer* SGD, epoch 90 yang ditunjukkan pada Gambar 17. menunjukkan bahwa terdapat 13 sampel yang termasuk ke dalam kelas *Cracked* dan salah diklasifikasikan sebagai Normal (*False Positive*), serta terdapat 4 sampel yang termasuk ke dalam kelas Normal dan salah diklasifikasikan sebagai *Cracked* (*False Negative*).

Classification Report:				
	precision	recall	f1-score	support
Cracked	0.97	0.91	0.94	150
Normal	0.93	0.98	0.95	175
accuracy			0.95	325
macro avg	0.95	0.95	0.95	325
weighted avg	0.95	0.95	0.95	325

Sumber : (Nazhirin et al., 2023)

Gambar 18. Klasifikasi *Optimizer* SGD, epoch 90

Hasil prediksi label dari *Classification Report* yang ditunjukkan pada Gambar 18. menunjukkan nilai *precision* dan *recall* yang tinggi untuk kedua kelas, menandakan performa yang baik dalam memprediksi kelas *Cracked* dan Normal.

Dapat disimpulkan bahwa pada epoch 90, model dengan *Optimizer* SGD memiliki performa yang baik dalam memprediksi sampel-sampel *Cracked* dan Normal, dengan sebagian kecil sampel yang salah diklasifikasikan (*False Positive* dan *False Negative*).

Dengan demikian, dapat disimpulkan bahwa model dengan *Optimizer* Adam pada epoch 70 memiliki performa yang buruk dalam mengenali sampel-sampel *Cracked*, sementara model dengan *Optimizer* Adam pada epoch 100 dan SGD pada epoch 50 dan 90 memiliki performa yang baik dalam mengenali dan memprediksi sampel-sampel *Cracked* dan Normal, meskipun terdapat beberapa kesalahan klasifikasi.

5. Evaluation

Evaluasi adalah sebuah tahapan dimana tahap-tahap penelitian yang sebelumnya telah dilakukan akan kembali dilakukan pengecekan guna menentukan apakah penelitian yang dilakukan telah sesuai dengan apa yang ditentukan pada tahap awal.

a. Evaluasi Hasil

Tahapan ini adalah untuk evaluasi terhadap hasil yang didapatkan dari model dengan menggunakan arsitektur VGG16 adapun hasil yang diperoleh dapat dilihat pada tabel 4 :

Skenario	Akurasi
1	98%
2	54%
3	98%
4	98%
5	94%
6	95%
7	95%
8	94%

Sumber : (Nazhirin et al., 2023)

Dari 8 skenario yang dilakukan skenario ke 1,3,4 menjadi model terbaik dengan akurasi 98%.

b. Review Proses Tahapan Penelitian

Pada tahap ini akan dilakukan pengecekan pada tiap tahapan yang telah dilakukan guna memastikan tidak adanya faktor yang terlewat. Tahapan-tahapan pada penelitian ini dapat dilihat pada tabel 5.

Tabel 5. Review Proses Tahapan Penelitian

Tahapan	Keterangan
1. <i>Image Acquisition</i>	Tahapan pada <i>image acquisition</i> yang telah dilakukan telah sesuai dengan tujuan awal yang ditentukan.
1) Pengumpulan data	

Tahapan	Keterangan
2. <i>Image Processing</i> 1) <i>Resizing</i> 2) Transformasi label	Tahapan pada <i>Image Processing</i> yang telah dilakukan sesuai dengan tujuan yang telah ditentukan dimana data siap digunakan untuk modelling.
3. <i>Modelling</i> 1) Pelatihan model 2) Parameter tunning	Tahapan pada <i>modelling</i> yang telah dilakukan sesuai dengan penentuan spesifikasi model yang akan dibuat.
4. <i>Classification</i> 1) Ban normal 2) Ban rusak	Tahapan pada <i>classification</i> yang telah dilakukan sesuai dengan tujuan awal penelitian dimana model dapat melakukan klasifikasi pada ban kendaraan.

Sumber : (Nazhirin et al., 2023)

KESIMPULAN

Pada penelitian ini dilakukan pengembangan model untuk melakukan klasifikasi pada kelayakan ban kendaraan dengan menggunakan metode *deep learning* arsitektur VGG16. Untuk dataset yang digunakan dalam pembuatan model menggunakan data *opensource* yang berasal dari kaggle yang berjumlah 1028 data citra yang berisi ban normal dan ban rusak yang mengalami keretakan. Pada penelitian ini dilakukan 8 skenario dimana *parameter tunning* dilakukan dengan mengubah nilai epoch dan *optimaizer* guna mendapatkan model dengan *dropout 0.2* atau *optimaizer adam* dengan *optimaizer sgd* mendapatkan hasil yang baik pada pelatihannya sehingga dapat disebut arsitektur VGG16 sangat efektif dalam membedakan kondisi ban yang berbeda, seperti ban *cracked*, atau ban yang dalam kondisi normal. Pada pengujian yang dilakukan dengan 8 skenario nilai akurasi terbaik yang didapatkan adalah 98%. Dalam kesimpulannya, penelitian ini menunjukkan bahwa arsitektur VGG16 efektif dalam melakukan klasifikasi kondisi ban kendaraan dan memberikan akurasi yang tinggi. Adapun saran untuk penelitian selanjutnya yaitu eksploarsi penggunaan dataset yang lebih luas dan dataset yang lebih variatif, termasuk berbagai jenis ban kendaraan dan

evaluasi performa model pada kondisi pencahayaan yang berbeda.

REFERENSI

Arthono, A., & Permana, V. A. (2022). Perencanaan Perkerasan Lentur Jalan Raya Menggunakan Metode Analisa Komponen SNI 1732-1989-F Ruas Jalan Raya Mulya Sari Kecamatan Pamanukan Sampai Kecamatan Binong Kabupaten Subang Propinsi Jawa Barat. *Jurnal Komposit*, 6(1), 41. <https://doi.org/10.32832/komposit.v6i1.6740>

Damuri, A., Riyanto, U., Rusdianto, H., & Aminudin, M. (2021). Implementasi Data Mining dengan Algoritma Naïve Bayes Untuk Klasifikasi Kelayakan Penerima Bantuan Sembako. *JURIKOM (Jurnal Riset Komputer)*, 8(6), 219. <https://doi.org/10.30865/jurikom.v8i6.3655>

Haksoro, E. I., & Setiawan, A. (2021). Pengenalan Jamur Yang Dapat Dikonsumsi Menggunakan Metode Transfer Learning Pada Convolutional Neural Network. *Jurnal ELTIKOM*, 5(2), 81–91. <https://doi.org/10.31961/eltikom.v5i2.428>

Pramana, A. L., Setyati, E., & Kristian, Y. (2020). Model Cnn Lenet Dalam Pengenalan Jenis Golongan Kendaraan. *Institut Sains Dan Teknologi Terpadu Surabaya*, 13(2), 65–69.

Prasetyo, E., Purbaningtyas, R., Adityo, R. D., Prabowo, E. T., & Ferdiansyah, A. I. (2021). Perbandingan Convolution Neural Network Untuk Klasifikasi Kesegaran Ikan Bandeng Pada Citra Mata. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, 8(3), 601. <https://doi.org/10.25126/jtiik.2021834369>

Puspitasari, L. (2020). Analisa Performance Ban Pada Unit Produksi Overburden Hd-785 Terhadap Produktivitas Tambang Batubara. *Kurvatek*, 5(1), 69–79. <https://doi.org/10.33579/krvtk.v5i1.1775>

Putra, Kade Bramasta Vikana, Bayupati, I. P. A., & Arsa, D. M. S. (2021). Klasifikasi Citra Daging Menggunakan Deep Learning dengan Optimisasi Hard Voting. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, 5(4), 656–662. <https://doi.org/10.29207/resti.v5i4.3247>

Rismiyati, R., & Luthfiarta, A. (2021). VGG16 Transfer Learning Architecture for Salak Fruit Quality Classification. *Telematika*, 18(1), 37. <https://doi.org/10.31315/telematika.v18i1.4025>

Rozaqi, A. J., Sunyoto, A., & Arief, M. R. (2021). Implementasi Transfer Learning pada Algoritma Convolutional Neural Network Untuk Identifikasi Penyakit Daun Kentang. *Procedia of Engineering and Life Science*, 1(1). <https://pels.umsida.ac.id/index.php/PELS/article/view/820>

Ruusen, A. S., Veibe, S., & Lembong, R. R. (2021).

- Penegakan Hukum Pidana Karena Kelalaian Pengemudi Kendaraan Yang Mengakibatkan Kecelakaan Lalu Lintas. *Lex Crimen*, 10(2), 97–108.
- Shidiq, A. L. A., SUhartono, E., & Saidah, S. (2022). Klasifikasi Kecacatan Ban Untuk Mengendalikan Kualitas Produk Menggunakan Model CNN Dengan Arsitektur VGG-16 Classification Of Tire Defect To Control Product Quality Using Cnn Model With VGG-16 Architecture. *Telkom University*, 8(6), 3216–3225. www.kaggle.com.
- Swasono, D. I., Tjandrasa, H., & Fathicah, C. (2019). Classification of tobacco leaf pests using VGG16 transfer learning. *Proceedings of 2019 International Conference on Information and Communication Technology and Systems, ICTS 2019*, 176–181. <https://doi.org/10.1109/ICTS.2019.8850946>
- Ufriandi, A. (2021). Analisis Tingkat Keausan Terhadap Pemakaian Ban Merek a, B Dan C Menggunakan Ban Standar 90/90-14 46 P. *Jurnal Surya Teknik*, 8(1), 282–288. <https://doi.org/10.37859/jst.v8i1.2678>
- Wang, R., Guo, Q., Lu, S., & Zhang, C. (2019). Tire Defect Detection Using Fully Convolutional Network. *IEEE Access*, 7, 43502–43510. <https://doi.org/10.1109/ACCESS.2019.2908483>.
- Hendarawan, I. E. (2022). Vehicle Tire Crack Classification Using ResNet50 Architecture. *SATIN - Sains dan Teknologi Informasi*, 9(1). <https://doi.org/10.33372/stn.v9i1.902>