

**KLASIFIKASI TINGKAT KARAT PADA BAUT
MENGUNAKAN METODE *CONVOLUTIONAL NEURAL
NETWORK* (CNN) MODEL RESNET18**



Disusun Oleh:

Devan Yusfa Sukmadya	210491100027
Difa Muhammad	210491100035
Reyhan Anandra Rachmanto	210491100038
Afif Maulana Agung Izzul Haq	210491100056

**PROGRAM STUDI TEKNIK MEKATRONIKA
JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS TRUNOJOYO MADURA
BANGKALAN
2024**

DAFTAR ISI

KATA PENGANTAR	iv
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan masalah.....	2
1.3. Tujuan Penelitian.....	2
1.4. Batasan Masalah.....	2
1.5. Manfaat Penelitian.....	3
BAB II TINJAUAN PUSTAKA.....	4
2.1. BAUT()	4
2.2. Karat (Korosi).....	4
2.3. Augmentasi.....	5
2.4. Convolutional Neural Networks.....	5
2.5. Deep Convolutional Neural Networks	5
2.6. DEEP RESIDUAL NETWORK (Resnet).....	6
BAB III METODE PENELITIAN.....	7
3.1. Alur Penelitian.....	7
BAB IV HASIL DAN PEMBAHASAN	12
4.1. Preprocessing.....	12
4.2. Augmentasi Data	12
4.3. Pelatihan data	14
4.4. Klasifikasi Tingkat Karat Menggunakan DCNN model RESNET18	17
DAFTAR PUSTAKA	20

DAFTAR GAMBAR

Gambar 2.1. Baut.....	4
Gambar 3.1. Alur Penelitian	7
Gambar 3.2. Dataset Tingkat Karat	8
Gambar 3.3. Resnet18 Pada Matlab	9
Gambar 3.4. Arsitektur Resnet18	9
Gambar 3.5. Training Dataset	10
Gambar 3.6. Confusion Matrix.....	11
Gambar 4.1. Data Hasil Resize.....	12
Gambar 4.2. Fitur Augmentasi data	13
Gambar 4.3. Penggunaan fitur imageDataAugmenter.....	13
Gambar 4.4. Opsi Pelatihan	16
Gambar 4.5. Hasil Training	16
Gambar 4.6. Confussion Matrix	17
Gambar 4.7. Modifikasi Layer Akhir Upsampling.....	17
Gambar 4.8. Input Contoh Gambar dari file.....	17
Gambar 4.9. Baut Bersih	18
Gambar 4.10. Hasil Plot scores	18
Gambar 4.11. Baut Agak Berkarat	18
Gambar 4.12. Hasil Plot Baut Agak Berkarat	19
Gambar 4.13. Baut Karat Parah.....	19
Gambar 4.14. Hasil Plot Baut Karat Parah.....	19

KATA PENGANTAR

Puji dan syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa atas rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan laporan penelitian dengan judul "Klasifikasi Tingkat Karat pada Baut Menggunakan Metode Convolutional Neural Network (CNN) Model ResNet18". Laporan ini merupakan salah satu syarat untuk menyelesaikan tugas akhir pada program studi Teknik Mekatronika.

Penelitian ini bertujuan untuk mengembangkan model klasifikasi yang mampu mengidentifikasi dan mengklasifikasikan tingkat karat pada baut menggunakan metode Convolutional Neural Network (CNN) dengan arsitektur ResNet18. Pemilihan metode ini didasarkan pada kemampuannya yang telah terbukti dalam mengatasi masalah klasifikasi gambar dengan akurasi yang tinggi serta kemampuannya mengekstraksi fitur-fitur penting dari gambar dengan efisien.

Penulis menyadari bahwa laporan ini masih jauh dari sempurna. Oleh karena itu, penulis sangat mengharapkan kritik dan saran yang membangun dari pembaca demi perbaikan dan penyempurnaan laporan ini di masa yang akan datang.

Akhir kata, penulis berharap semoga laporan penelitian ini dapat bermanfaat bagi semua pihak yang membacanya serta dapat memberikan kontribusi bagi perkembangan ilmu pengetahuan dan teknologi, khususnya dalam bidang klasifikasi gambar menggunakan metode deep learning

Bangkalan, 01 Juli 2024

Kelompok 03

BAB I PENDAHULUAN

1.1. Latar Belakang

Karat adalah salah satu masalah yang umum dihadapi dalam berbagai industri yang menggunakan komponen logam, terutama baut dan mur. Proses pengkaratan dapat menyebabkan penurunan kualitas dan kekuatan bahan, yang berdampak pada keselamatan dan keandalan struktur. Oleh karena itu, deteksi dan klasifikasi tingkat karat pada baut sangat penting untuk menjaga kualitas dan integritas produk. Namun, deteksi manual karat sering kali memerlukan waktu dan tenaga yang banyak serta rentan terhadap kesalahan manusia. Untuk mengatasi masalah ini, penggunaan teknologi canggih seperti metode Convolutional Neural Network (CNN) menawarkan solusi yang lebih efisien dan akurat.

CNN telah terbukti sangat efektif dalam berbagai tugas pengolahan gambar, termasuk deteksi dan klasifikasi objek. Model CNN seperti ResNet18, yang merupakan salah satu arsitektur jaringan saraf konvolusi yang paling banyak digunakan, telah menunjukkan kinerja yang sangat baik dalam pengenalan pola dan fitur visual dari gambar. ResNet18 dikenal karena kemampuannya dalam mengatasi masalah vanishing gradient dan memungkinkan pelatihan jaringan yang lebih dalam dengan lebih efisien melalui penggunaan residual blocks. Oleh karena itu, menerapkan model ResNet18 untuk mengklasifikasikan tingkat karat pada baut dapat memberikan hasil yang lebih akurat dan konsisten dibandingkan dengan metode manual.

Dengan mengadopsi model CNN seperti ResNet18 untuk klasifikasi tingkat karat, proses evaluasi kualitas dapat dilakukan secara otomatis dan lebih cepat. Model ini akan dilatih menggunakan dataset gambar baut dengan berbagai tingkat karat, memungkinkan sistem untuk belajar membedakan antara tingkat karat yang berbeda berdasarkan fitur visual yang ada pada gambar. Hasil dari klasifikasi ini tidak hanya akan meningkatkan efisiensi dalam proses inspeksi, tetapi juga akan mengurangi ketergantungan pada penilaian manual dan potensi kesalahan manusia. Penggunaan CNN untuk klasifikasi tingkat karat pada baut adalah langkah maju dalam penerapan teknologi machine learning di industri manufaktur dan perawatan logam. Dengan meningkatkan akurasi dan kecepatan deteksi karat, teknologi ini

dapat membantu perusahaan dalam mengelola pemeliharaan dan kualitas produk secara lebih efektif. Oleh karena itu, penelitian ini bertujuan untuk mengevaluasi kinerja model ResNet18 dalam klasifikasi tingkat karat pada baut, dan untuk menunjukkan potensi penerapan teknologi CNN dalam meningkatkan proses inspeksi dan pemeliharaan produk logam.

1.2. Rumusan masalah

Berikut merupakan rumusan masalah yang dikerjakan dalam penelitian ini sebagai berikut.

1. Bagaimana cara mengklasifikasikan tingkat karat pada baut secara akurat menggunakan model Convolutional Neural Network (CNN)?
2. Apa kinerja model ResNet18 dalam mengidentifikasi tingkat karat dibandingkan dengan metode klasifikasi gambar lainnya?
3. Bagaimana pengaruh berbagai parameter dan teknik pra-pemrosesan gambar terhadap akurasi klasifikasi tingkat karat?

1.3. Tujuan Penelitian

Berikut merupakan tujuan penelitian yang dikerjakan dalam penelitian ini sebagai berikut.

1. Mengembangkan dan menerapkan model CNN dengan arsitektur ResNet18 untuk mengklasifikasikan tingkat karat pada baut.
2. Mengevaluasi kinerja model ResNet18 dalam klasifikasi tingkat karat dan membandingkannya dengan metode klasifikasi gambar lainnya.
3. Menentukan pengaruh teknik pra-pemrosesan gambar dan parameter pelatihan terhadap akurasi klasifikasi.

1.4. Batasan Masalah

Berikut merupakan batasan masalah yang dikerjakan dalam penelitian ini sebagai berikut.

1. Penelitian ini akan menggunakan dataset gambar baut dengan tingkat karat yang sudah dikumpulkan sebelumnya.
2. Fokus penelitian ini adalah pada penggunaan model ResNet18. Model CNN lainnya tidak akan dieksplorasi secara mendalam, sehingga hasil yang diperoleh mungkin berbeda jika menggunakan arsitektur yang berbeda.

3. Penelitian ini akan membatasi pada beberapa teknik pra-pemrosesan gambar dasar seperti normalisasi dan augmentasi data, tanpa mengeksplorasi teknik pra-pemrosesan yang lebih canggih atau khusus.

1.5. Manfaat Penelitian

Berikut merupakan manfaat masalah yang dikerjakan dalam penelitian ini sebagai berikut.

1. Penggunaan model CNN seperti ResNet18 dalam klasifikasi tingkat karat dapat meningkatkan akurasi dan konsistensi dalam proses inspeksi karat pada baut, mengurangi kemungkinan kesalahan manusia dalam penilaian.
2. Penelitian ini akan menyediakan solusi otomatis untuk mengidentifikasi tingkat karat, yang dapat menghemat waktu dan biaya dalam proses pengujian dan pemeliharaan komponen logam di industri.
3. Hasil dari penelitian ini dapat memberikan panduan bagi pengembangan sistem inspeksi otomatis yang lebih baik di masa depan, terutama dalam industri manufaktur dan pemeliharaan yang memerlukan penilaian kualitas secara rutin.

BAB II TINJAUAN PUSTAKA

2.1. BAUT()

Baut merupakan komponen penting dalam berbagai aplikasi industri karena kemampuannya untuk menggabungkan dua atau lebih bagian dengan aman dan stabil. Kualitas baut sangat berpengaruh terhadap keamanan dan efisiensi struktur atau mesin yang menggunakannya. Proses manufaktur baut yang tepat mencakup pemilihan bahan, pemanasan, pembentukan, pembuatan ulir, dan penyelesaian yang dirancang untuk menghasilkan baut yang memenuhi spesifikasi ketat. Setiap tahap dirancang untuk memastikan bahwa baut yang dihasilkan memenuhi spesifikasi ketat yang diperlukan untuk aplikasi tertentu [1]. Berikut merupakan contoh gambar dari baut yang akan ditunjukkan pada Gambar 2.1.



Gambar 2.1. Baut

2.2. Karat (Korosi)

Korosi adalah salah satu masalah utama yang mempengaruhi kinerja dan umur baut. Korosi pada baut dapat menyebabkan penurunan kekuatan material, sehingga meningkatkan risiko kegagalan struktural. Jenis korosi yang umum terjadi pada baut termasuk korosi galvanik, korosi celah, dan stress-corrosion cracking. Faktor-faktor lingkungan seperti kelembaban, kandungan garam, dan eksposur terhadap bahan kimia tertentu dapat mempercepat proses korosi. Penelitian menunjukkan bahwa pemahaman yang lebih baik tentang mekanisme korosi dan faktor-faktor yang mempengaruhinya dapat membantu dalam pengembangan metode perlindungan yang lebih efektif [2].

Korosi bisa disebut sebagai kerusakan atau degradasi logam akibat reaksi dengan lingkungan yang korosif. Korosi dapat juga diartikan sebagai serangan yang merusak logam karena logam bereaksi secara kimia atau elektrokimia dengan lingkungan. Ada definisi lain yang mengatakan bahwa korosi adalah kebalikan dari

proses ekstraksi logam dari bijih mineralnya. Contohnya, bijih mineral logam besi di alam bebas ada dalam bentuk senyawa besi oksida atau besi sulfida, setelah diekstraksi dan diolah, akan dihasilkan besi yang digunakan untuk pembuatan baja atau baja paduan. Selama pemakaian, baja tersebut akan bereaksi dengan lingkungan yang menyebabkan korosi (kembali menjadi senyawa besi oksida) [3].

2.3. Augmentasi

Dalam konteks pembelajaran mesin, augmentasi data adalah teknik yang digunakan untuk meningkatkan jumlah dan variasi data pelatihan melalui transformasi seperti rotasi, pemotongan, penskalaan, dan perubahan warna. Teknik augmentasi data membantu model pembelajaran mesin untuk menjadi lebih tahan terhadap overfitting dan meningkatkan kemampuan generalisasi, terutama dalam tugas-tugas pengenalan citra dan segmentasi objek. Dengan augmentasi data, model dapat belajar dari lebih banyak variasi data, yang pada akhirnya meningkatkan akurasi dan kinerja model dalam pengenalan pola [4].

2.4. Convolutional Neural Networks

Convolutional Neural Networks (CNN) adalah arsitektur deep learning yang sangat efektif dalam analisis citra. CNN bekerja dengan menerapkan konvolusi pada input untuk mengekstraksi fitur, yang kemudian digunakan untuk klasifikasi atau segmentasi. Arsitektur CNN terdiri dari beberapa lapisan konvolusi yang diikuti oleh lapisan pooling dan fully connected. CNN telah digunakan secara luas dalam berbagai aplikasi seperti pengenalan objek, deteksi wajah, dan segmentasi citra medis. Keunggulan CNN terletak pada kemampuannya untuk secara otomatis mengekstraksi fitur penting dari data gambar tanpa memerlukan teknik ekstraksi fitur manual [5].

2.5. Deep Convolutional Neural Networks

Deep Convolutional Neural Networks (DCNN) adalah arsitektur deep learning yang sangat efektif dan banyak diterapkan dalam berbagai aplikasi computer vision, seperti klasifikasi gambar, deteksi objek, dan segmentasi gambar. DCNN mampu mengekstraksi fitur dari gambar dengan menggunakan lapisan konvolusi, yang membantu model untuk mengidentifikasi pola dan struktur dalam data visual. DCNN terdiri dari beberapa komponen utama, yaitu lapisan konvolusi, lapisan pooling, lapisan fully connected, dan fungsi aktivasi. Lapisan konvolusi

bertanggung jawab untuk mengekstraksi fitur dari input gambar melalui penggunaan filter yang beroperasi secara lokal pada gambar. Lapisan pooling digunakan untuk mengurangi dimensi spasial dari fitur peta, mengurangi jumlah parameter dan komputasi dalam jaringan, serta membantu dalam generalisasi model. Setelah beberapa lapisan konvolusi dan pooling, lapisan fully connected digunakan untuk menggabungkan semua fitur yang diekstraksi dan membuat keputusan akhir untuk klasifikasi atau deteksi objek. Fungsi non-linear seperti ReLU (Rectified Linear Unit) digunakan setelah lapisan konvolusi dan fully connected untuk memperkenalkan non-linearitas ke dalam model, memungkinkan jaringan untuk belajar dari data yang kompleks [6].

2.6. DEEP RESIDUAL NETWORK (Resnet)

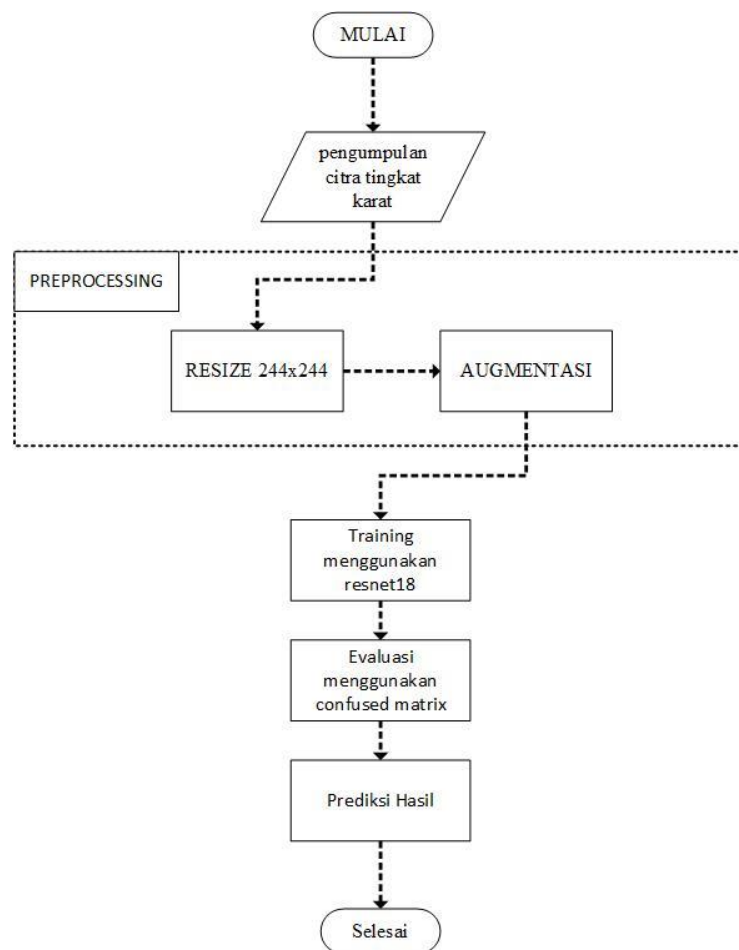
ResNet atau Residual Network adalah jenis arsitektur deep learning yang diperkenalkan untuk mengatasi masalah vanishing gradient yang umum terjadi pada jaringan yang sangat dalam. ResNet menggunakan pendekatan residual learning dengan memperkenalkan shortcut connections yang memungkinkan informasi dapat mengalir langsung melintasi beberapa lapisan tanpa terhalang. ResNet18, salah satu varian dari ResNet, terdiri dari 18 lapisan dan telah terbukti sangat efektif dalam berbagai tugas pengenalan dan klasifikasi citra. Model ini memungkinkan pelatihan jaringan yang sangat dalam tanpa kehilangan akurasi karena masalah degradasi, sehingga untuk meningkatkan kinerja model dalam tugas-tugas yang kompleks [7].

BAB III METODE PENELITIAN

Prosedur penelitian dimulai dari perumusan masalah kemudian pengumpulan data dan informasi melalui studi literatur review melalui jurnal, skripsi, tesis, dan ebook. Kemudian analisis data yang diperoleh berdasarkan analisis tersebut kemudian membuat inovasi system klasifikasi tingkat karat. Langkah selanjutnya perancangan model dengan menggunakan dataset, kemudian system akan dilakukan uji coba dilanjutkan pada pembahasan.

3.1. Alur Penelitian

Adapun berikut merupakan alur penelitian mengenai klasifikasi tingkat karat pada baut menggunakan *deep convolutional neural network* dengan arsitektur resnet18 dapat dilihat pada Gambar 3.1.

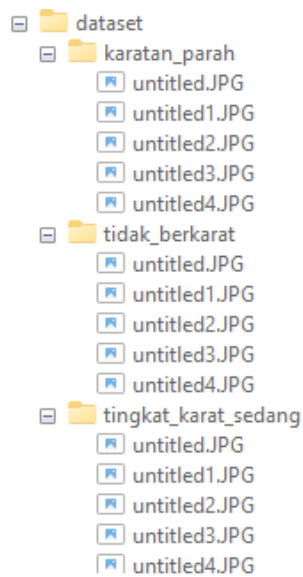


Gambar 3.1. Alur Penelitian

Dari alur penelitian yang telah dibuat maka setiap tahap dijelaskan lebih lengkap seperti dibawah ini:

a. Pengumpulan Data

Pengumpulan data dilakukan dengan studi pustaka, ataupun studi literatur mengenai tingkat karat yang dapat diklasifikasi. Dataset dibuat menggunakan folder dan setiap sub folder dibuat menjadi label, yang berisi gambar sesuai dengan karakteristik tingkat karat. Adapun beberapa data Tingkat karat dapat dilihat pada Gambar 3.2.



Gambar 3.2. Dataset Tingkat Karat

b. Tahap Pre-Processing

Tahap preprocessing dimulai dengan melakukan *crop* pada gambar input menjadi format yang dapat di proses oleh Resnet18, yang berukuran 224x224 pixel, berikutnya dilakukan augmentasi pelatihan menerapkan augmentasi gambar secara dinamis atau hanya mengubah ukuran gambar selama pelatihan menggunakan *syntax* “*augmentedImageDatastore*”. Data citra berikutnya dibagi menjadi 80% untuk *data training* dan 20% untuk *data testing*.

```
% 5. Ubah Ukuran Gambar Menggunakan augmentedImageDatastore
inputSize = [224 224 3]; % Ukuran input yang diharapkan oleh ResNet18
augimdsTrain = augmentedImageDatastore(inputSize(1:2), imdsTrain);
augimdsTest = augmentedImageDatastore(inputSize(1:2), imdsTest);
```

c. Training menggunakan Resnet18

Resnet18 memiliki struktur terdiri dari 72 layers diantaranya 1 layer data, 1 *preprocessing layer*, 1 maxpooling layer, 20 convolutional layers, 20 *batch normalization layers*, 17 activation layer ReLU, 8 addition layer, 1 average pooling layer, 1 fully connected layers, 1 softmax layers, 1 classification layers. Adapun untuk arsitektur Resnet18 dapat dilihat pada Gambar 3.3.

```
>> resnet18

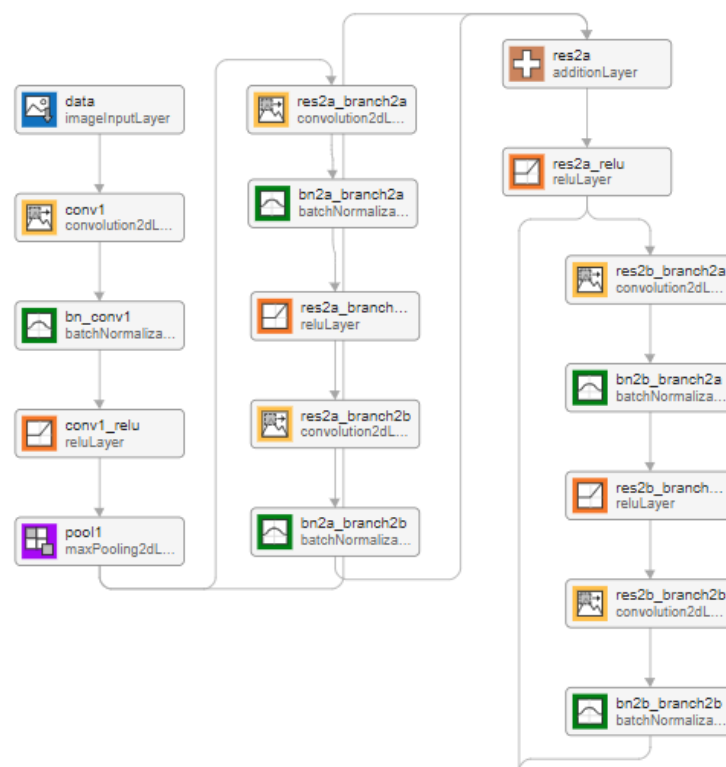
ans =

DAGNetwork with properties:

    Layers: [71x1 nnet.cnn.layer.Layer]
Connections: [78x2 table]
InputNames: {'data'}
OutputNames: {'ClassificationLayer predictions'}
```

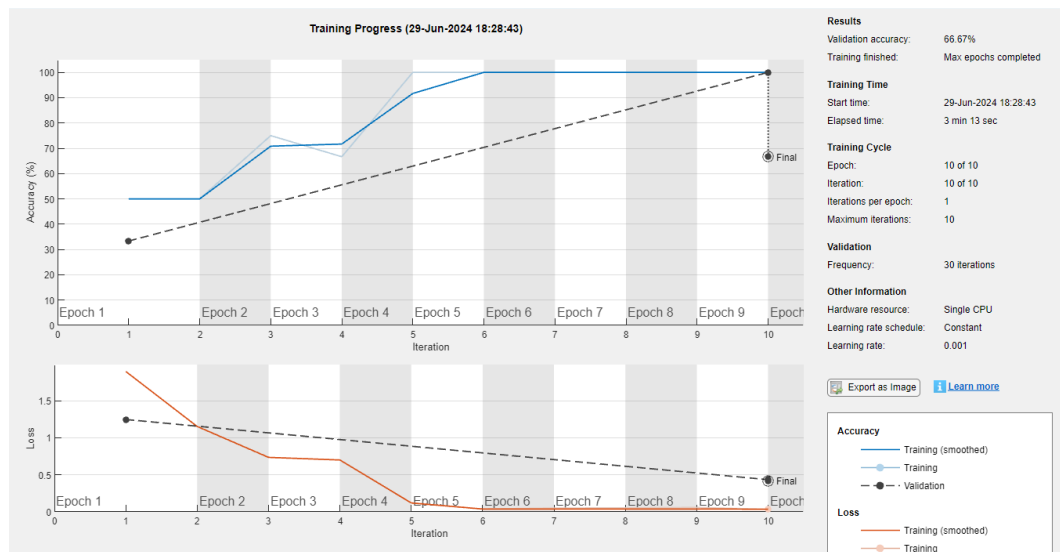
Gambar 3.3. Resnet18 Pada Matlab

Adapun penggambaran visual dari Resnet18 pada MATLAB dapat dilihat sebagai berikut pada Gambar 3.4.



Gambar 3.4. Arsitektur Resnet18

Adapun berikut merupakan hasil training Dataset menggunakan Resnet18 dapat dilihat sebagai berikut.



Gambar 3.5. Training Dataset

d. Evaluasi Model

Evaluasi Model menggunakan confused matrix(Matrix kebingungan), menggunakan *syntax confusionmat Confusion matrix untuk masalah klasifikasi dengan N kelas adalah matriks $N \times N$ yang isinya sebagai berikut:*

1. *True Positives (TP):* Jumlah sampel yang benar-benar diklasifikasikan sebagai kelas positif.
2. *True Negatives (TN):* Jumlah sampel yang benar-benar diklasifikasikan sebagai kelas negatif.
3. *False Positives (FP):* Jumlah sampel yang salah diklasifikasikan sebagai kelas positif (Type I error).
4. *False Negatives (FN):* Jumlah sampel yang salah diklasifikasikan sebagai kelas negatif (Type II error).

Adapun berikut merupakan penggunaan confolution matrix pada MATLAB dapat dilihat pada Gambar 3.6.

```
% Matriks Kebingungan
confMat = confusionmat(actualLabels, predictedLabels);
confMatChart = confusionchart(confMat, categories(actualLabels));
```

True Class	karatan_parah	1		
	tidak_berkarat		1	
	tingkat_karat_sedang	1		
		karatan_parah	tidak_berkarat	tingkat_karat_sedang

Gambar 3.6. Confusion Matrix

Contoh perhitungan keakurasian dengan menggunakan data dari *confusion matrix* :

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

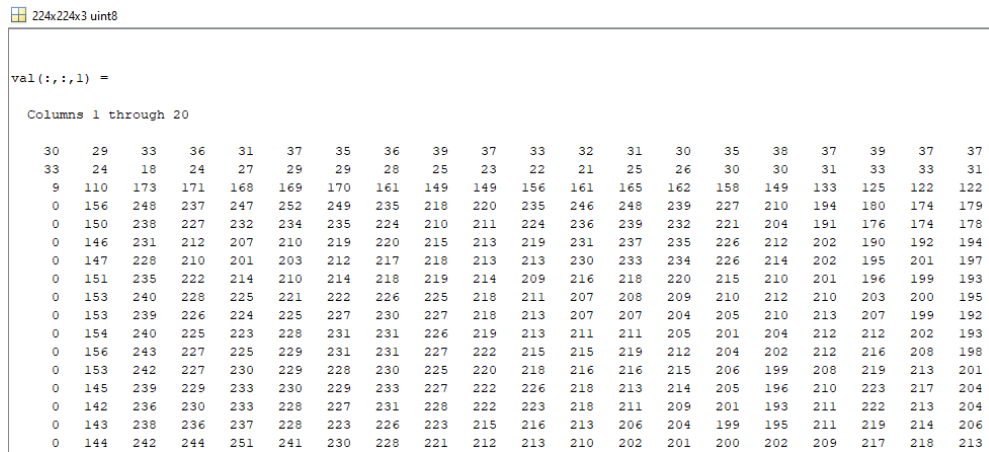
Test Accuracy: 0.66667

BAB IV HASIL DAN PEMBAHASAN

Adapun hasil dan kesimpulan didapatkan dari hasil test keakurasian menggunakan model yang telah dibuat menggunakan arsitektur resnet18.

4.1. Preprocessing

Penelitian yang dilakukan dengan menggunakan dataset yang terdiri dari 3 kategori tingkat karat, yang berasal dari studi pustaka literatur dengan tujuan agar data lebih fleksibel dilakukan penyesuaian. Data yang digunakan dilakukan resize menjadi 244*244 dengan tujuan agar dapat diolah menggunakan model Resnet18. Pelabelan data dibuat menjadi 3 kategori adapun kategori pertama merupakan “tidak_berkarat”, kedua “karatan_tingkat_sedang”, dan ketiga “karatan_full”. Adapun hasil resize dapat dilihat pada Gambar 4.1.



```
224x224x3 uint8

val(:,:,1) =

Columns 1 through 20

    30    29    33    36    31    37    35    36    39    37    33    32    31    30    35    38    37    39    37    37
    33    24    18    24    27    29    29    28    25    23    22    21    25    26    30    30    31    33    33    31
     9   110   173   171   168   169   170   161   149   149   156   161   165   162   158   149   133   125   122   122
     0   156   248   237   247   252   249   235   218   220   235   246   248   239   227   210   194   180   174   179
     0   150   238   227   232   234   235   224   210   211   224   236   239   232   221   204   191   176   174   178
     0   146   231   212   207   210   219   220   215   213   219   231   237   235   226   212   202   190   192   194
     0   147   228   210   201   203   212   217   218   213   213   230   233   234   226   214   202   195   201   197
     0   151   235   222   214   210   214   218   219   214   209   216   218   220   215   210   201   196   199   193
     0   153   240   228   225   221   222   226   225   218   211   207   208   209   210   212   210   203   200   195
     0   153   239   226   224   225   227   230   227   218   213   207   207   204   205   210   213   207   199   192
     0   154   240   225   223   228   231   231   226   219   213   211   211   205   201   204   212   212   202   193
     0   156   243   227   225   229   231   231   227   222   215   215   219   212   204   202   212   216   208   198
     0   153   242   227   230   229   228   230   225   220   218   216   216   215   206   199   208   219   213   201
     0   145   239   229   233   230   229   233   227   222   226   218   213   214   205   196   210   223   217   204
     0   142   236   230   233   228   227   231   228   222   223   218   211   209   201   193   211   222   213   204
     0   143   238   236   237   228   223   226   223   215   216   213   206   214   199   195   211   219   214   206
     0   144   242   244   251   241   230   228   221   212   213   210   202   201   200   202   209   217   218   213
```

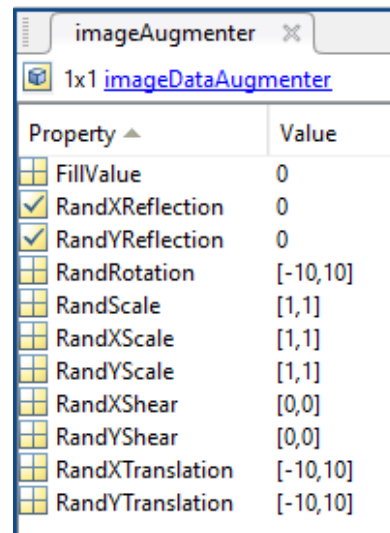
Gambar 4.1. Data Hasil Resize

4.2. Augmentasi Data

Tujuan dilakukannya augmentasi data sebagai opsi untuk memperbanyak pilihan variasi data. Pengaruh positif dari augmentasi data dapat mengurangi *overfitting* saat system bekerja. Augmentasi data efektif meningkatkan ukuran dataset pelatihan tanpa harus mengumpulkan gambar baru, yang dapat mahal dan memakan waktu. Parameter augmentasi yang digunakan antara dengan menggunakan beberapa fitur syntax MATLAB, dengan function Rand rotation, RandXTranslation, dan RandYTranslation. Adapun penjabaran lebih lengkap dapat dilihat pada Gambar 4.2.


```
% Augmentasi data pelatihan
imageAugmenter = imageDataAugmenter( ...
    'RandRotation', [-10, 10], ...
    'RandXTranslation', [-10, 10], ...
    'RandYTranslation', [-10, 10]);
```

Gambar 4.2. Fitur Augmentasi data



Gambar 4.3. Penggunaan fitur imageDataAugmenter

Adapun penjelasan lebih lanjut mengenai augmentasi data yang digunakan dapat dilihat sebagai berikut:

a. RandRotation

Parameter ini mengindikasikan bahwa augmentasi data akan mencakup rotasi acak dari gambar input. Rentang rotasi yang diberikan adalah antara -10 derajat hingga +10 derajat. Setiap gambar dalam dataset pelatihan dapat dirotasi secara acak dalam rentang ini, yang membantu model belajar dari gambar yang mungkin sedikit miring atau berbeda sudut.

b. RandXTranslation

Parameter ini mengindikasikan bahwa augmentasi data akan mencakup translasi acak (pergeseran) gambar pada sumbu X. Rentang translasi yang diberikan adalah antara -10 piksel hingga +10 piksel. Setiap gambar dalam dataset pelatihan dapat digeser ke kiri atau ke kanan secara acak dalam rentang ini, yang membantu model belajar dari gambar yang mungkin tidak terpusat sempurna.

c. RandYTranslation

Parameter ini mengindikasikan bahwa augmentasi data akan mencakup translasi acak (pergeseran) gambar pada sumbu Y. Rentang translasi yang diberikan adalah antara -10 piksel hingga +10 piksel. Setiap gambar dalam dataset pelatihan dapat digeser ke atas atau ke bawah secara acak dalam rentang ini, yang membantu model belajar dari gambar yang mungkin tidak terpusat sempurna pada sumbu vertikal.

4.3. Pelatihan data

Pelatihan data dengan menggunakan beberapa opsi seperti Kode ini mendefinisikan opsi-opsi untuk melatih sebuah jaringan saraf tiruan (neural network) menggunakan fungsi `trainingOptions` di MATLAB. Berikut adalah penjelasan masing-masing parameter yang digunakan:

1. `sgdm` (Stochastic Gradient Descent with Momentum):

Metode Optimisasi: Metode SGDM digunakan untuk mempercepat konvergensi model. Gradient descent stochastic menghitung gradien menggunakan subset acak (mini-batch) dari dataset, yang mengurangi variabilitas dalam estimasi gradien dibandingkan dengan metode batch penuh. Momentum mempercepat pembelajaran di arah gradien yang relevan dan mengurangi osilasi. Ini dilakukan dengan menambahkan fraksi dari pembaruan gradien sebelumnya ke pembaruan gradien saat ini.

2. `MiniBatchSize`, 32:

Ukuran Mini-Batch: Ukuran mini-batch yang digunakan adalah 32. Ini berarti gradien dan pembaruan parameter dihitung menggunakan 32 contoh data pada setiap iterasi. Penggunaan mini-batch mengurangi variansi dalam estimasi gradien, memberikan keseimbangan antara pembaruan parameter yang cepat (stochastic) dan stabilitas (batch).

3. `MaxEpochs`, 10:

Jumlah Epoch Maksimal: Epoch adalah satu siklus penuh melalui dataset pelatihan. `MaxEpochs` menentukan bahwa pelatihan akan dilakukan maksimal selama 10 epoch, menetapkan batas atas berapa kali jaringan akan

melihat setiap contoh dalam dataset selama pelatihan, berfungsi sebagai mekanisme untuk mencegah overfitting.

4. `InitialLearnRate', 0.001:`

Laju Pembelajaran Awal: Laju pembelajaran (learning rate) mengontrol ukuran langkah yang diambil dalam ruang parameter pada setiap pembaruan. Laju pembelajaran yang terlalu tinggi dapat menyebabkan pelatihan yang tidak stabil dan gagal konvergen, sedangkan laju pembelajaran yang terlalu rendah akan membuat proses pelatihan sangat lambat dan mungkin terjebak dalam minima lokal.

5. `Shuffle, every-epoch:`

Pengacakan Data: Data akan diacak ulang pada setiap epoch. Pengacakan data mencegah jaringan dari belajar urutan spesifik dari data, yang dapat membantu mencegah overfitting dan meningkatkan generalisasi model.

6. `ValidationData, augimdsValidation:`

Data Validasi: Dataset validasi digunakan untuk mengevaluasi kinerja model selama pelatihan tanpa mempengaruhi parameter model. Validasi berkala memungkinkan pelatih untuk memantau kinerja model dan mendeteksi overfitting atau underfitting pada tahap awal.

7. `ValidationFrequency, 30:`

Frekuensi Validasi: Model akan divalidasi setiap 30 iterasi. Validasi yang sering memungkinkan pelatih untuk memonitor kinerja model dengan baik dan membuat keputusan yang tepat mengenai penyesuaian hyperparameter.

8. `Verbose, false:`

Verbosity: Menetapkan apakah output rinci akan ditampilkan selama pelatihan. Dengan Verbose diatur ke false, hanya informasi penting yang akan ditampilkan, mengurangi kebisingan (noise) dalam output.

9. `Plots, training-progress:`

Plot Kemajuan Pelatihan: Mengaktifkan tampilan grafik yang menunjukkan metrik pelatihan seperti akurasi dan kerugian (loss) selama pelatihan. Visualisasi kemajuan pelatihan membantu dalam memahami

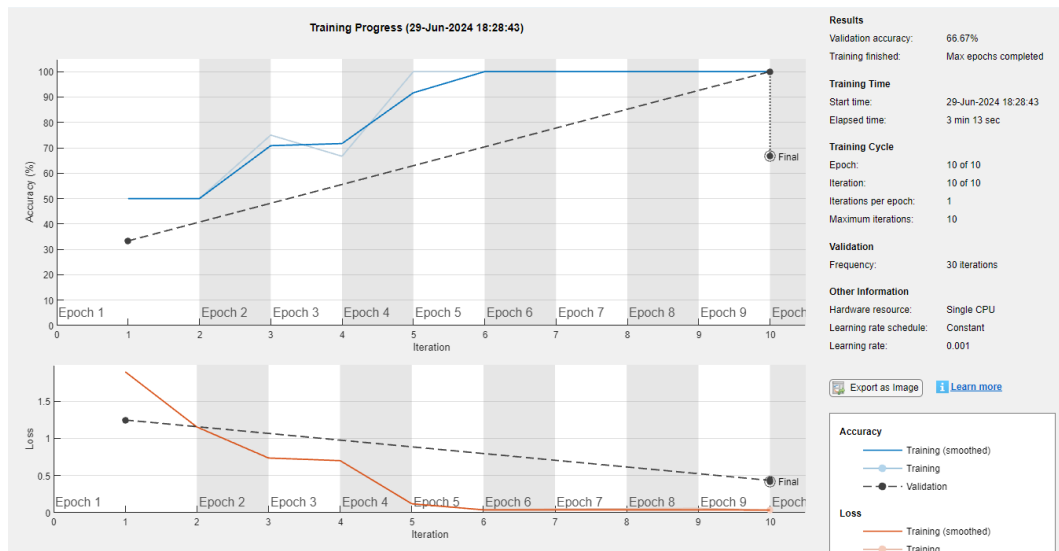
bagaimana model belajar dan kapan mungkin diperlukan penyesuaian hyperparameter atau intervensi lainnya.

Adapun berikut merupakan penggunaan beberapa parameter pelatihan dapat dilihat pada Gambar 4.4.

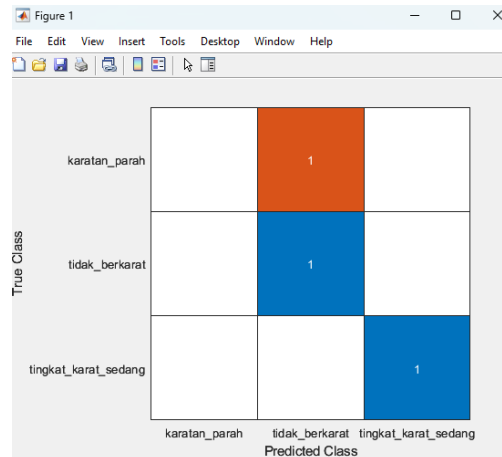
```
% Menentukan opsi pelatihan
options = trainingOptions('sgdm', ...
    'MiniBatchSize', 32, ...
    'MaxEpochs', 10, ...
    'InitialLearnRate', 0.001, ...
    'Shuffle', 'every-epoch', ...
    'ValidationData', augimdsValidation, ...
    'ValidationFrequency', 30, ...
    'Verbose', false, ...
    'Plots', 'training-progress');
```

Gambar 4.4. Opsi Pelatihan

Adapun hasil training yangn telah dilakukan dengan menggunakan MATLAB dapat dilihat pada Gambar 4.5.



Gambar 4.5. Hasil Training



Gambar 4.6. Confussion Matrix

4.4. Klasifikasi Tingkat Karat Menggunakan DCNN model RESNET18

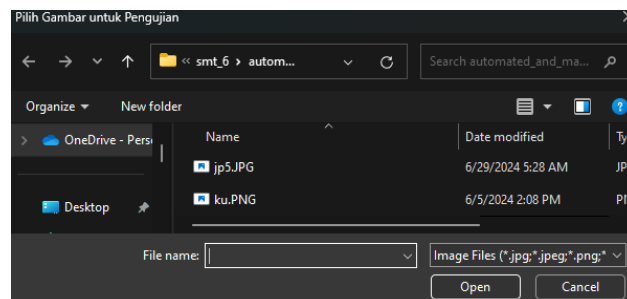
Klasifikasi dilakukan dengan menggunakan model ResNet18 yang telah dimodifikasi dengan menambahkan layer upsampling pada bagian akhir. Untuk menyesuaikan dengan jumlah kelas pada dataset baru, layer keluaran (classification layer) pada ResNet18 diganti dengan jumlah kelas yang sesuai. Layer fully connected dan classification layer diubah menggunakan replaceLayer.

```
net = resnet18;

% Mengganti lapisan terakhir untuk klasifikasi baru
lgraph = layerGraph(net);

% Mengganti lapisan klasifikasi terakhir
numClasses = numel(categories(imdsTrain.Labels));
newLearnableLayer = fullyConnectedLayer(numClasses, ...
    'Name', 'new_fc', ...
    'WeightLearnRateFactor', 10, ...
    'BiasLearnRateFactor', 10);
lgraph = replaceLayer(lgraph, 'fc1000', newLearnableLayer);
```

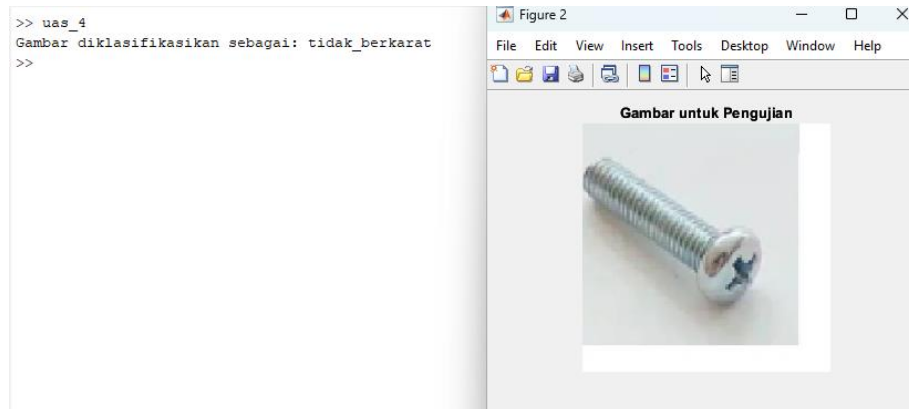
Gambar 4.7. Modifikasi Layer Akhir Upsampling



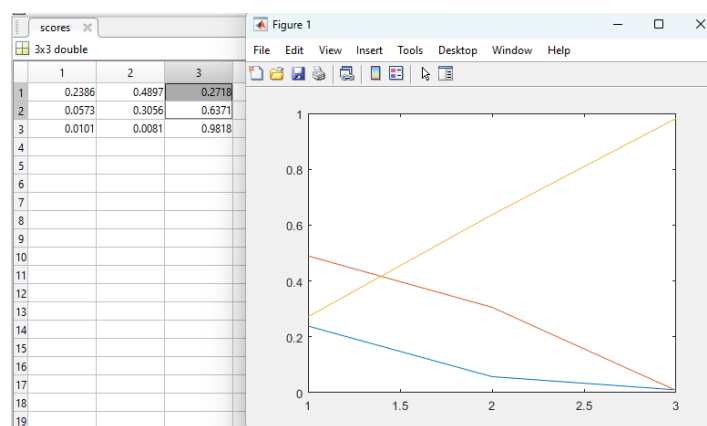
Gambar 4.8. Input Contoh Gambar dari file

Pengujian dilakukan dengan memprediksi 3 gambar citra inputan dari file, Adapun pengujian dapat dilihat sebagai berikut:

1. Pengujian menggunakan gambar baut bersih

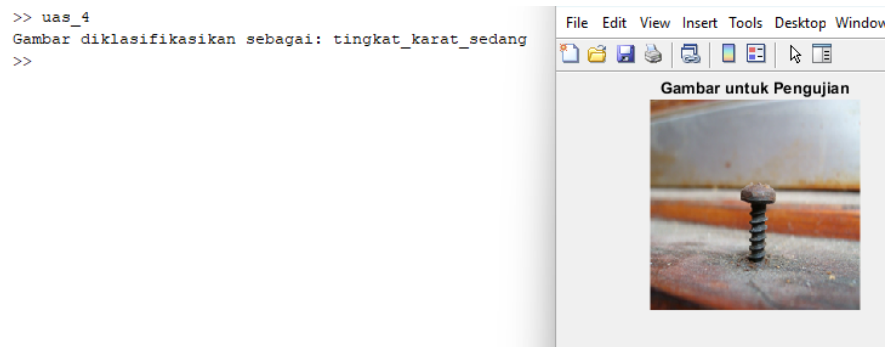


Gambar 4.9. Baut Bersih

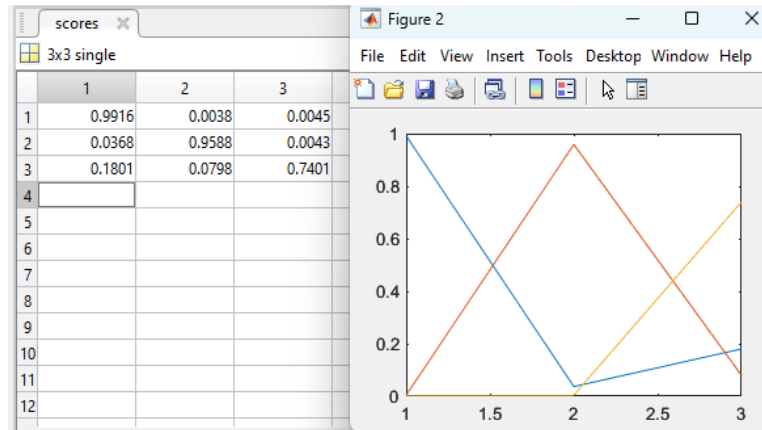


Gambar 4.10. Hasil Plot scores

2. Pengujian dengan baut agak berkarat

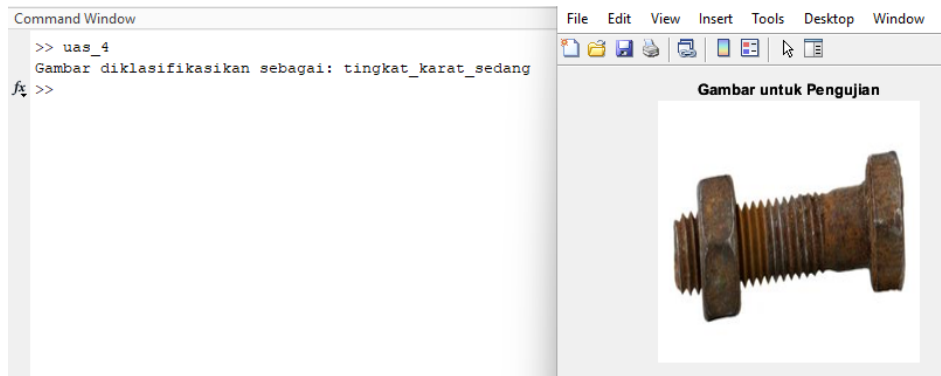


Gambar 4.11. Baut Agak Berkarat

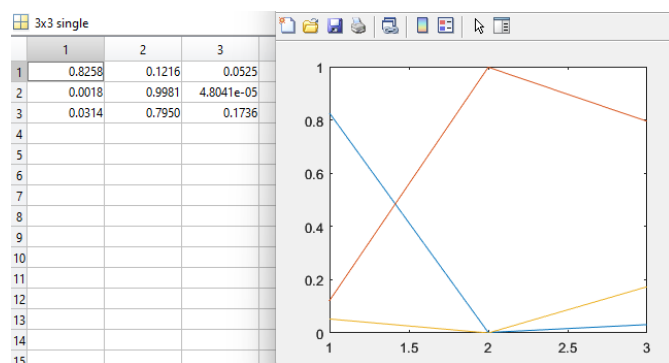


Gambar 4.12. Hasil Plot Baut Agak Berkarat

3. Pengujian dengan menggunakan baut pada karat parah



Gambar 4.13. Baut Karat Parah



Gambar 4.14. Hasil Plot Baut Karat Parah

DAFTAR PUSTAKA

- [1] Elshawesh, F., Abusowa, K., Mahfud, H., et al. (2008). Stress-Corrosion Cracking and Galvanic Corrosion of Internal Bolts from a Multistage Water Injection Pump. *Journal of Failure Analysis and Prevention*, 8, 48–53.
- [2] Wijaya, R., & Azis, M. (2017). Buku Pintar Teknologi Pemeliharaan dan Perawatan Fasilitas Industri. Penerbit Andi.
- [3] Zhang, W., Li, C., & Zhang, X. (2020). A Survey of Intelligent Corrosion Detection Methods. *Journal of Intelligent Manufacturing*, 31(3), 567-582.
- [4] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770-778.
- [5] Zhang, Z., & Hu, X. (2019). Improving the Robustness of Deep Convolutional Neural Networks with Residual Learning. *IEEE Transactions on Neural Networks and Learning Systems*, 30(2), 482-494.
- [6] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems (NIPS)*, 1097-1105.
- [7] Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely Connected Convolutional Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4700-4708.
- [8] Li, Y., Dong, W., & Zhang, W. (2019). A Deep Learning Approach for Corrosion Detection and Classification. *Journal of Manufacturing Processes*, 38, 170-178.
- [9] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *Advances in Neural Information Processing Systems (NIPS)*, 91-99.