NATIONAL INSTITUTE OF TECHNOLOGY, TIRUCHIRAPPALLI TAMILNADU 620015

Department of Computer Applications

Project Work-2020

Final Review

Title: Machine Learning Algorithms for Defect Analysis in Thermographic Images

Guide/Mentor

Dr. U. Srinivasulu Reddy

Submitted by

Surendra Lodhi

205117066

Content

- Introduction
- Motivation
- Problem Statement
- Platform
- Literature Survey
- Methodologies
- Design and Development
- Results and Discussions
- Conclusion and Future Work
- References

Introduction

Thermal Imaging:

- Remote sensing technique
- Capable in daytime and night-time

Machine Learning:

- Predictive Analysis
- Classification Analysis
- Clustering Analysis

Motivation

- Potholes on road pose a great danger to lives.
- The root cause of potholes formation is water logging and due to presence of water in cracks and crevices.
- The temperature of potholes is normally expected to be less than the surrounding road.
- Thus it is easy to differentiate between potholes and non potholes using thermal imaging technique.

Problem Statement

Finding the defects in thermographic images and identifying the defect type and predicting the depth of defects using Machine Learning Algorithm.

Platform

Hardware:

- FLIR Camera
- Processor I3 or higher
- 64 bit machine with min 4GB RAM

Software:

- Matlab
- Python(2.7 or 3.7)
- Anaconda(Jupyter Notebook, Tensorflow)

Literature Survey

- Artificial neural networks based quantitative evaluation of subsurface anomalies in quadratic frequency modulated thermal wave imaging

  Link

- Artificial Neural Network Application for Damages Classification in Fibre glass Pre-impregnated Laminated Composites (FGLC) from Ultrasonic Signal

  Link

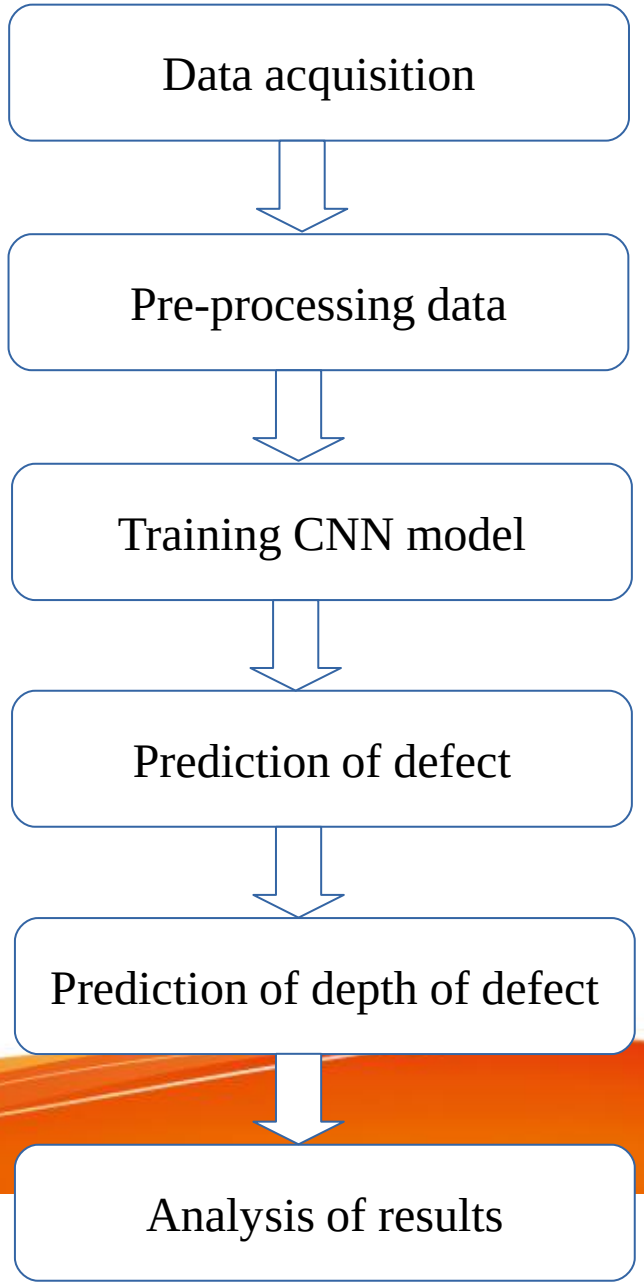
Methodologies

- **Convolution Neural Network(CNN)**
  To classifying the images as defected and non defected

- **Artificial Neural Network(ANN)**
  To predict the depth of defects.

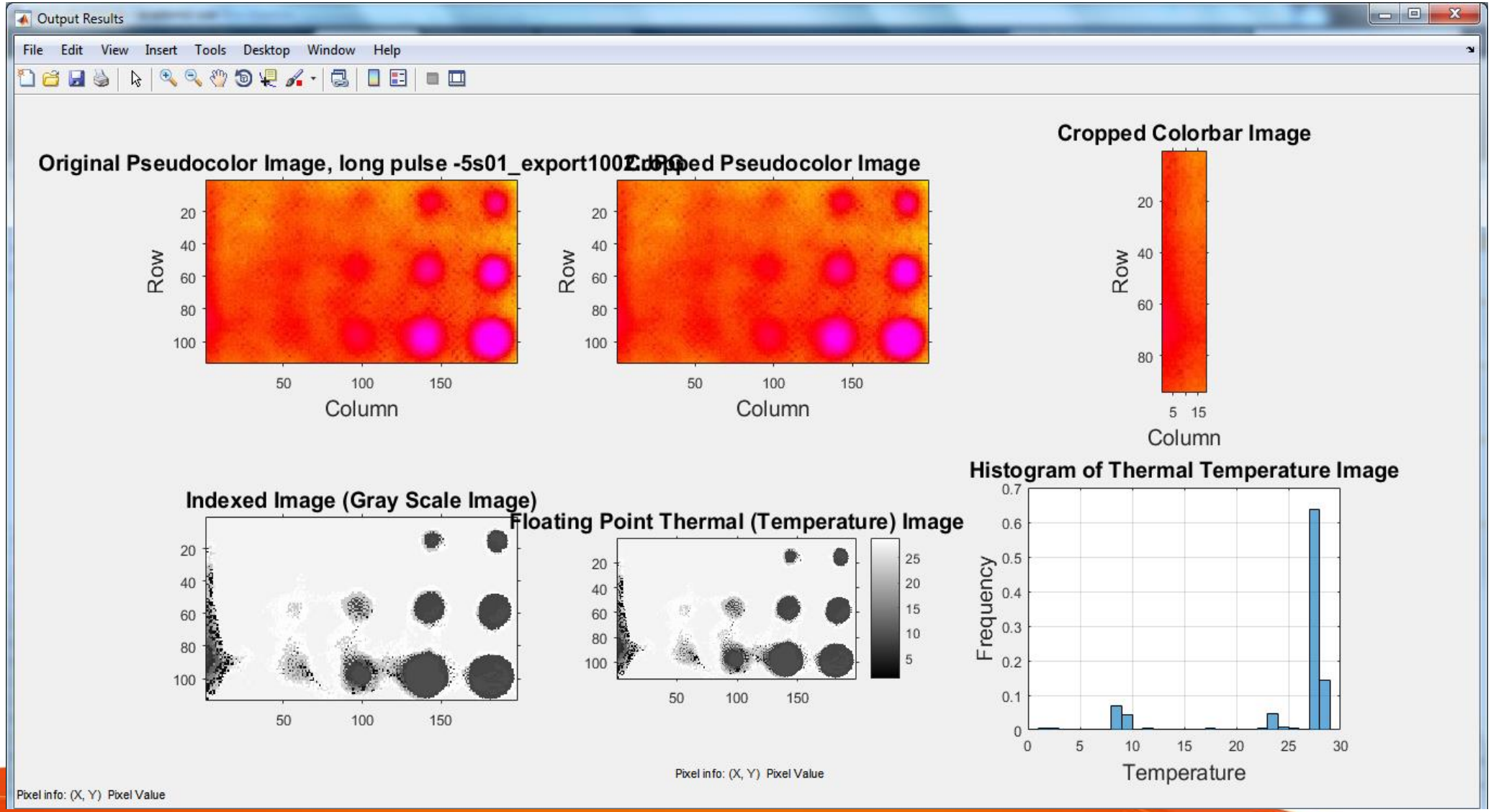
- **Multinomial Logistic Regression**
  To predict the depth of defects.

Design and Development



Results and Discussions

Features analysis of thermographic images



Results and Discussions

Training CNN Model

```
batch_size = 32,
class_mode = 'binary')
classifier.fit_generator(training_set,
steps_per_epoch = 1000,
epochs = 5,
validation_data = test_set,
validation_steps = 2000)
```

```
Using TensorFlow backend.
```

```
Found 2480 images belonging to 2 classes.
Found 16 images belonging to 2 classes.
Epoch 1/5
1000/1000 [==============================] - 408s 408ms/step - loss: 0.0386 - accuracy: 0.9836 - val_loss: 6.1519e-14 - val_acc
uracy: 1.0000
Epoch 2/5
1000/1000 [==============================] - 341s 341ms/step - loss: 0.0203 - accuracy: 0.9929 - val_loss: 3.4594e-17 - val_acc
uracy: 1.0000
Epoch 3/5
1000/1000 [==============================] - 356s 356ms/step - loss: 0.0120 - accuracy: 0.9951 - val_loss: 1.0020e-14 - val_acc
uracy: 1.0000
Epoch 4/5
1000/1000 [==============================] - 346s 346ms/step - loss: 0.0095 - accuracy: 0.9963 - val_loss: 2.0286e-14 - val_acc
uracy: 1.0000
Epoch 5/5
1000/1000 [==============================] - 339s 339ms/step - loss: 0.0105 - accuracy: 0.9958 - val_loss: 4.8909e-30 - val_acc
uracy: 1.0000
```

```
Out[1]: <keras.callbacks.callbacks.History at 0x18f2d9b4648>
```

Results and Discussions continue...

Prediction of defect using CNN

Out[1]: <keras.callbacks.callbacks.History at 0x18f2d9b4648>

In [8]:
```
# Part 3 - Making new predictions
# Prediction of image type
import numpy as np
from keras.preprocessing import image
test_image = image.load_img('thermal-images-dataset/testing/long pulse -5s01_export186.JPG', target_size = (64, 64))
test_image = image.img_to_array(test_image)
test_image = np.expand_dims(test_image, axis = 0)
result = classifier.predict(test_image)
training_set.class_indices
if result[0][0] == 1:
    prediction = 'no defect'
else:
    prediction = 'defect'
print(prediction)
```

```
no defect
```

In [ ]:

Results and Discussions continue...

Prediction of defect using CNN

Out[1]: <keras.callbacks.callbacks.History at 0x18f2d9b4648>

In [9]:
```
# Part 3 - Making new predictions
# Prediction of image type
import numpy as np
from keras.preprocessing import image
test_image = image.load_img('thermal-images-dataset/testing/long pulse -5s01_export1118.JPG', target_size = (64, 64))
test_image = image.img_to_array(test_image)
test_image = np.expand_dims(test_image, axis = 0)
result = classifier.predict(test_image)
training_set.class_indices
if result[0][0] == 1:
    prediction = 'no defect'
else:
    prediction = 'defect'
print(prediction)
```

```
defect
```

In [ ]:

Results and Discussions continue...

Training ANN model

```
np.savetxt('plotoutputAllR01.csv', outputArray, delimiter=',', fmt='%f')
##############################################################################################

from sklearn.metrics import mean_squared_error
print("\n Mean Squared Error:",mean_squared_error(y_test,pred))
```

```
18000/18000 [==============================] - 1s 39us/step - loss: 0.1115 - accuracy: 0.4423
Epoch 93/100
18000/18000 [==============================] - 1s 39us/step - loss: 0.1114 - accuracy: 0.4433
Epoch 94/100
18000/18000 [==============================] - 1s 39us/step - loss: 0.1113 - accuracy: 0.4433
Epoch 95/100
18000/18000 [==============================] - 1s 55us/step - loss: 0.1114 - accuracy: 0.4433
Epoch 96/100
18000/18000 [==============================] - 1s 42us/step - loss: 0.1113 - accuracy: 0.4426
Epoch 97/100
18000/18000 [==============================] - 1s 46us/step - loss: 0.1114 - accuracy: 0.4425
Epoch 98/100
18000/18000 [==============================] - 1s 44us/step - loss: 0.1113 - accuracy: 0.4421
Epoch 99/100
18000/18000 [==============================] - 1s 38us/step - loss: 0.1113 - accuracy: 0.4423
Epoch 100/100
18000/18000 [==============================] - 1s 37us/step - loss: 0.1110 - accuracy: 0.4432

 Mean Squared Error: 0.12177201237579988
```

```
In [ ]:
```

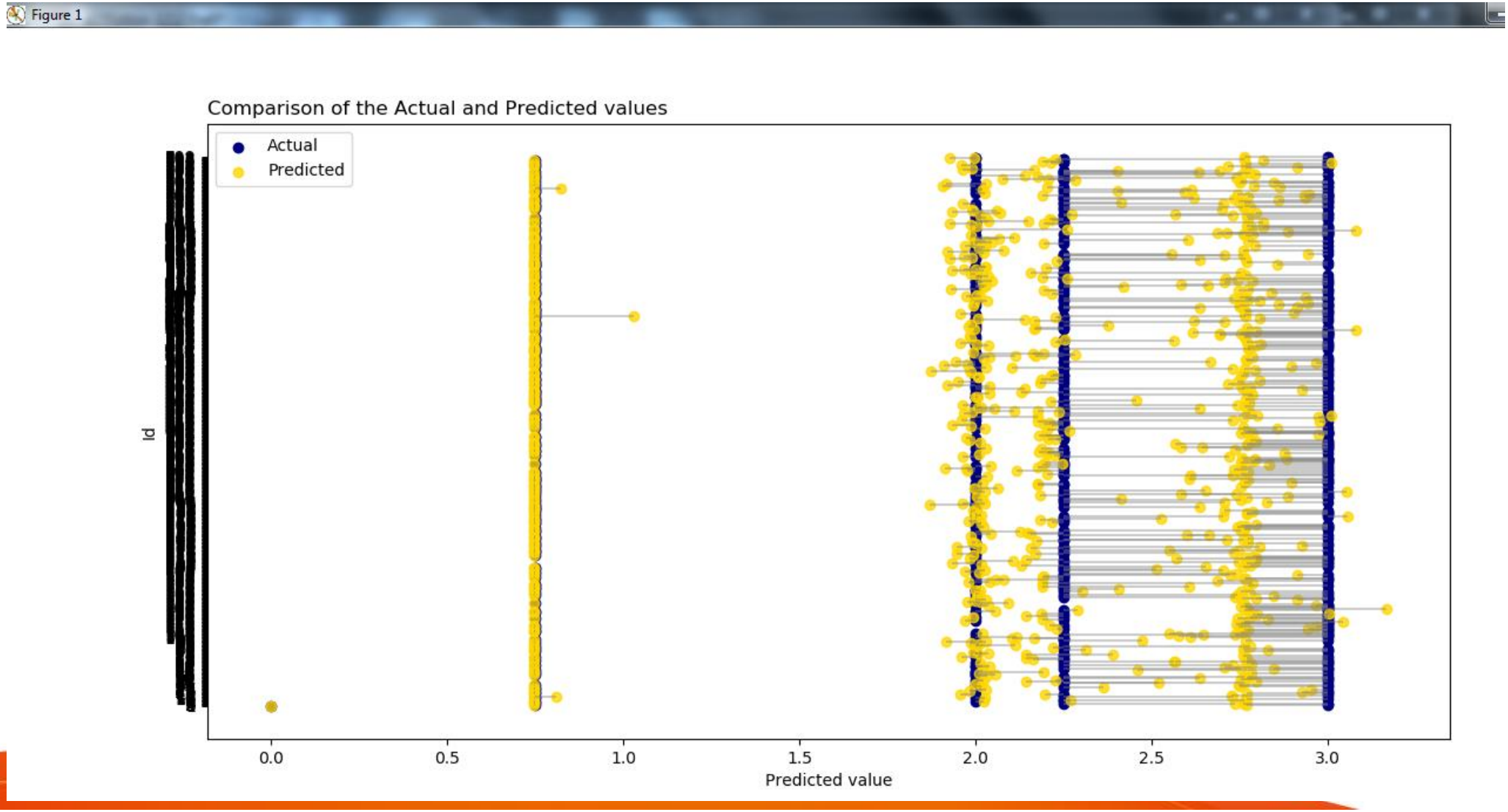
Results and Discussions continue...

Predicted depth of defects using ANN

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Id	Actual	Predicted												
2	1	3	2.8995												
3	2	3	2.875047												
4	3	0.75	0.759383												
5	4	2	2.027622												
6	5	3	2.675244												
7	6	0.75	0.759383												
8	7	3	2.689086												
9	8	3	2.923539												
10	9	2	1.946706												
11	10	2.25	2.095467												
12	11	2	1.992447												
13	12	2	2.066727												
14	13	2.25	2.581211												
15	14	2	1.915715												
16	15	2.25	1.452072												
17	16	3	2.605556												
18	17	2.25	2.383366												
19	18	0.75	0.759383												
20	19	2	2.295075												
21	20	2	1.956765												
22	21	2	2.154702												
23	22	2	1.928988												
24	23	2	1.834614												
25	24	0.75	0.759383												
26	25	0.75	0.759383												
27	26	2	2.06635												
28	27	0.75	0.759383												
29	28	3	2.455251												

Results and Discussions continue...

Comparison graph of actual and predicted depth using(ANN)



Results and Discussions continue...

Training Multinomial Regression Model

```
        outputArray[i][2]="Predicted"
    outputArray[i][0]=i+1;
    outputArray[i][1]=actual[i][0]
    outputArray[i][2]=pred[i]
    #outputArray[i][2]=actual[i][0]-pred[i][0]

np.savetxt('plotoutputlogis.csv', outputArray, delimiter=',', fmt='%f')
##############################################################################################
```

```
       Red  Green  Blue  Gray  Index  Intensity  Temperature
11456  255      1   243     1     12        105      11.7390
16528  255      0   250     0     20        105      18.9650
3253   254      2   187     2      7         98       7.2226
18614  254      5    73     5      2         87       2.7065
1544   255      0   249     0     20        105      18.9650
...    ...    ...   ...   ...    ...        ...          ...
9468   251      1   248     1     12        104      11.7390
3897   255      0   233     0     16        103      15.3520
14402  250      5   255     5     23        107      21.6740
5201   250      4   227     4     17        103      16.2550
14610  248      3   254     3     23        105      21.6740

[8000 rows x 7 columns]
[3 3 2 ... 2 2 3]
Logistic Regression model accuracy(in %): 75.1875
```

In [ ]:

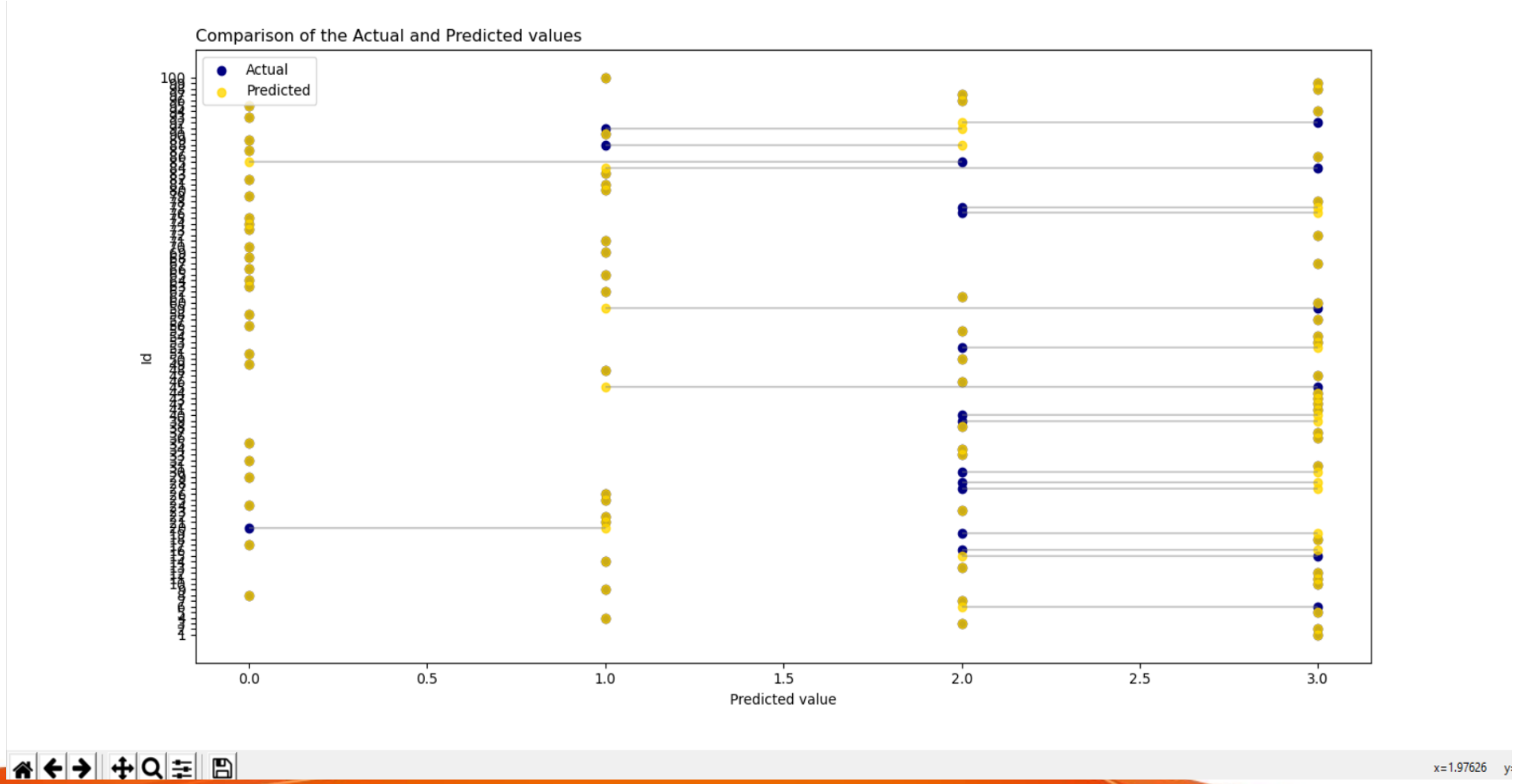
Results and Discussions continue...

Predicted depth of defects using Multinomial Logistic Regression

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Id	Actual	Predicted										
2	1	3	3										
3	2	3	3										
4	3	2	2										
5	4	1	1										
6	5	3	3										
7	6	3	2										
8	7	2	2										
9	8	0	0										
10	9	1	1										
11	10	3	3										
12	11	3	3										
13	12	3	3										
14	13	2	2										
15	14	1	1										
16	15	3	2										
17	16	2	3										
18	17	0	0										
19	18	3	3										
20	19	2	3										
21	20	0	1										
22	21	1	1										
23	22	1	1										
24	23	2	2										
25	24	0	0										
26	25	1	1										
27	26	1	1										
28	27	2	3										
29	28	2	3										
30	29	0	0										

Results and Discussions continue...

Comparison graph of actual and predicted depth(Multinomial Logistic Regression)



Results and Discussions continue...

Table 1 Classification results of thermographic images

Epoch	Loss	Accuracy	Val_loss	Val_accuray
1/1000	0.0386	0.9836	6.1519e-14	1.0000
2/1000	0.0203	0.9929	3.4599e-17	1.0000
3/1000	0.0120	0.9951	1.0020e-14	1.0000
4/1000	0.0095	0.9963	2.0286e-14	1.0000
5/1000	0.0105	0.9958	4.8909e-30	1.0000

Table 2 Comparison of results of different ML Algorithms

Defects	a	b	c	d
Actual size(cm)	0.75	2.0	2.25	3.0
ANN Regression	0.753164	2.08	2.240417	3.003
Logistic Regression	0.75	2.01	2.30	3.0

Conclusion and Future Work

Conclusion:

- Thermal imaging and Machine Learning is used for defect analysis in thermographic images.
- We used CNN for image classification as defected and non defected image.
- ANN and Multinomial Logistic Regression are used for predicting the depth of defects.
- Accuracy of Multinomial Logistic Regression is better than Artificial Neural Network.

Conclusion and Future Work continue…

Future Work:

- We can combine above mentioned ML algorithms in a single program so that it can be used for real life problem.
- This project is general demonstration of how Thermal Imaging and Machine Learning are used for defect analysis.
- Project can be extended and might be used in various fields like Healthcare, Aerospace, Machine Health Monitoring etc.

References

- https://www.sciencedirect.com/science/article
- https://www.researchgate.net/publication
- http://qirt.org/archives/qirt2016/papers/057.pdf
- https://www.ndt.net/article/qirt2014/papers
- https://towardsdatascience.com/
- https://medium.com/

Thank you