UNIVERSITY OF MALAYA

Corrosion Classification

WQD7001 Principles of Data Science

Table of Contents

1.1 Introduction

This paper present a classification of corrosion for material that are subject to the effects of corrosion using machine learning technique and features trained from a model. A model will be trained to ‘learn’ the differentiation between corroded surface and non-corroded surface. Corrosion management can be very expensive but can be reduced by going through appropriate corrosion management. Objective of this product is to provide solution to industrial organization in terms of maintenance, lifetime estimation of the equipment.

1.1.1 Target customer to use the data product

The main targeted user for this product are organization that use metal in their plant or as part of their daily runs for machines and equipment. For example;

- Oil and gas: pipeline, compressor, heat exchanger, piping, drill pipe, drill bit, etc.
- Machinery: broaching machine, screw machine, milling machine, lathe, etc.
- Transportation: trains, ships, etc.
- Plant: transmission towers, storage tanks, boilers, reactors, etc.

1.1.2 Data collection method

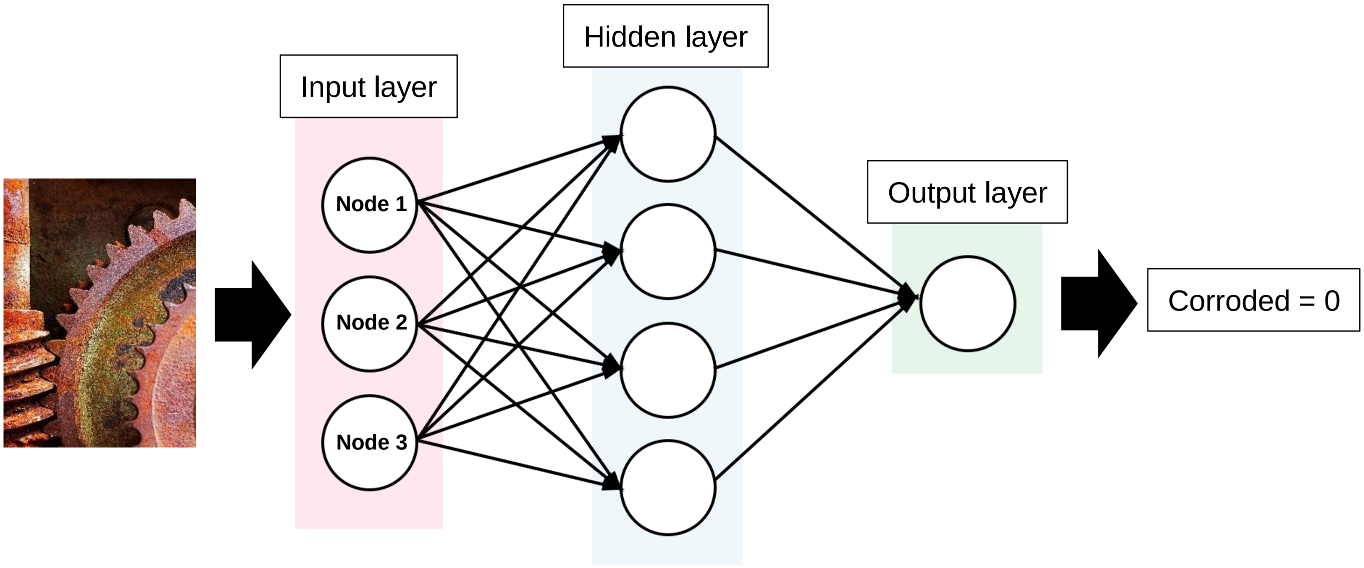
Primarily, for collecting data various types of random corroded metal images are collected from multiple online sources in order to train the model. Comparatively less noisy and zoomed in images are given priority in choosing the images for input.

Both corroded and not corroded steel images are collected in order to compare between them and measure the amount of corrosion.

The customers can give in their own images as input and the pre-trained model will compare the input images and calculate the amount of corrosion in the given data.

1.2 Project Workflow

In order to come out with an intelligent model to recognize the image and classify it, we use simple neural network approach which using nodes and layers to train the input data.



1.3 Data Story Telling

You may refer to this link for animated data story telling:

https://create.piktochart.com/output/39166420-corrsave

1.4 R packages

R-Packages and libraries:

I. **shiny:** This package is used in order to make interactive web application. It allowed the project to be shared straight from the Rstudio.

II. **Keras :** The package used is an API developed for neural network, that allow the program to both run in CPU or GPU (graphical processing unit) ,also adaptable in tensorflow . In our program we used neural network and tensor flow so, keras is useful in this.

III. **Imager:** This package is implemented here to provide basic image processing tools like filtering and transformation, it processes up to 4 dimensions : two spatial ,one color and one depth dimension.

IV. **Shinyjs:** In order to utilize java script features in shiny applications shinyjs is used. It can also be used to call custom javascript functions from R.

V. **ggplot2:** It deals with the data provided and map variables to aesthetics according to the given command. It creates graphics based on the grammar of graphics.

VI. **DT:** DT stands for DataTables, Data objects in R is organized using this java script featured library ,this library has been included in R package.

VII. **dplyr:** dplyr is used to abstract over how the data is stored. It helped to go over basic changes in data like select(), summaries(), arrange() etc.

VIII. **readr:** essential package to access and read rectangular data that are in csv, tsv, fwf format.

IX. **EBImage:** This package is useful toolbox for image processing and analysis, it works on the high throughput images and can break it down to segment cells and extract quantitative cellular descriptors.

X. **scales:** Applied for mapping data to aesthetics and also for automatically determining breaks and labels for axes and legends.

XI. **glcm:** It calculates the texture features of the image that originates from the grey level co-occurrence matrices in R. Texture statistics like mean, variance, homogeneity, contrast, dissimilarity , correlation is measured by it.

XII. **wvtool:** This tool facilitates pre-processing and analysing 2- dimensional wood images toward automated recognition. It helps to measure Haralick parameters, local binary pattern (LBP), higher order local autocorrelation (HLAC), Gabor filtering. The functions in this packages considers the grayscale images as input data.

XIII. **pRoc:** pRoc tool is implemented for visualising , comparing or smoothing the receiver operating characteristic curves (ROC) and also AUC or area under the curves. Confidence intervals can be computed for AUC and ROC curves.

XIV. **caret:** stands for characteristic and regression training , is a set of function that assisted with pre-processing, data splitting, training/testing functions and model comparison like confusion matrix.

XV. **tensorflow:** It is an open source software library for computation in data flow graphs. The flexible architecture allows to deploy computation to one or more 'CPUs' or 'GPUs' in a desktop, server, or mobile device with a single 'API'.

1.3 Expected Outcome

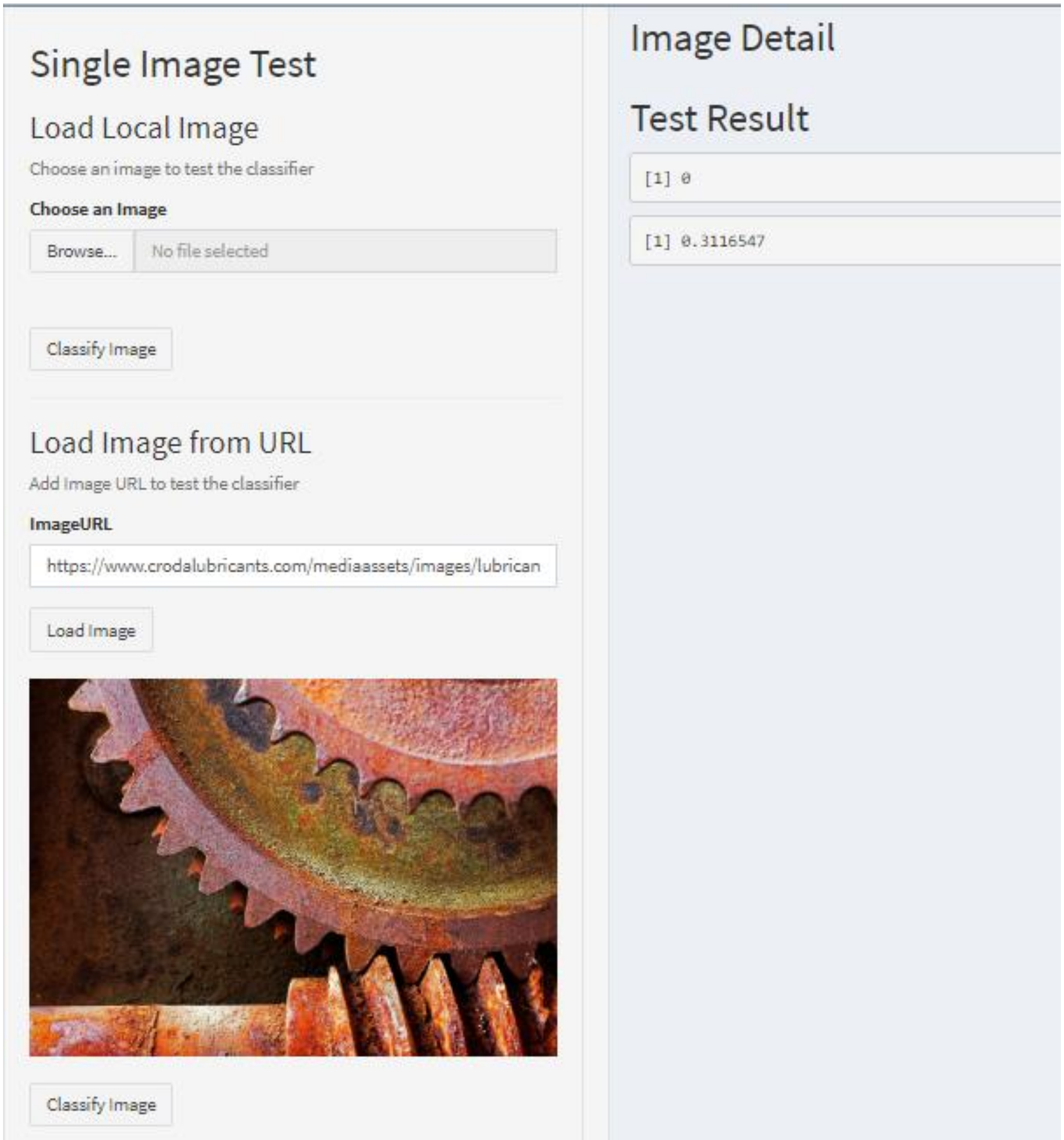
There are 3 options to test the data by:

i. Uploading and test 1 image

ii. Test image from URL (must be .png)

iii. Uploading a folder with multiple image

Below screenshot are the expected outcome of this project by using shinyapps.io:



Single Image Test

Load Local Image

Choose an image to test the classifier

Choose an Image

Browse... No file selected

Load Image from URL

Add Image URL to test the classifier

ImageURL

https://www.crodalubricants.com/mediaassets/images/lubrican

Image Detail

Test Result

```
[1] 0
```

```
[1] 0.3116547
```

1.4 User Manual

Option 1: Use our pre-trained model to test your data.

1.1 Download pre-trained model from https://github.com/nurafiqah78/CorrSave/blob/master/pre-trained_model%202019-05-16%2010.37.57

1.2 Skip Tab 1 as you already has the scale attributes ready from the trained model.

1.3 Download scale attributes file from https://github.com/nurafiqah78/CorrSave/blob/master/Scale%20Attribute.csv

1.4 Tab 2 (Pre-Train Model): upload (1.) Model File and (2.) Scale Attribute File

1.5 Tab 3 (Test Model): Upload your test data (1 image, URL or multiple images in older)

Option 2: Train your own datasets.

1.1 Separate your datasets into 2 folders (0: corroded, 1: non-corroded)

1.2 Tab 1 (Model Initialization): set your own parameters

1.3 Tab 2 (Train Model): upload class 0 and class 1 folders

1.4 Click 'Calculate features' and 'start training'

1.5 Tab 3 (Test Model): Upload your test data (1 image, URL or multiple images in older)