**Industrial anomaly detection of wood texture using unsupervised learning for quality assessment process automation**

Chaitanya

Department of Psychology, The George Washington University

PSYC 3170: Clinical Psychology

Dr. Tia M. Benedetto

April 19, 2023

Contents

[1. Project Background & Overview 4](#_Toc133477373)

[2. Problem Statement 4](#_Toc133477374)

[4. Literature Review 5](#_Toc133477375)

[4. Machine Learning Techniques used for Current Work 6](#_Toc133477376)

[Isolation Forest [8,9] 6](#_Toc133477377)

[Auto Encoders10 6](#_Toc133477378)

[5. Methodology 7](#_Toc133477379)

[5.1 Overview 7](#_Toc133477380)

[5.2 Data Source 7](#_Toc133477381)

[5.3 Data Cleaning and Transformation 8](#_Toc133477382)

[Fixing image size inconsistency [12, 13] 8](#_Toc133477383)

[Image to Pixel Array Conversion 9](#_Toc133477384)

[Addressing data quantity/size issue 10](#_Toc133477385)

[Feature Selection and Engineering 11](#_Toc133477386)

[Object Edge Extraction 11](#_Toc133477387)

[Image Pixel Scaling 13](#_Toc133477388)

[7. Model Selection and Evaluation 13](#_Toc133477389)

[8. Model Results and Discussion 14](#_Toc133477390)

[Descriptive Analysis 14](#_Toc133477391)

[Analyzing Sample Anomalous vs. Non-Anomalous Images 14](#_Toc133477392)

[Understanding an Anomalous Object 15](#_Toc133477393)

[Do we observe any significant differences in an anomaly vs non-anomaly object? 15](#_Toc133477394)

[Model Evaluation 16](#_Toc133477395)

[Isolation Forest – Parameter Tuning and Evaluation 16](#_Toc133477396)

[Auto Encoders – Parameter Tuning and Evaluation 18](#_Toc133477397)

[Autoencoder – Final Model Deep dive 21](#_Toc133477398)

[9. Discussion on Key Findings 22](#_Toc133477399)

[10. Conclusion 23](#_Toc133477400)

[11 Code Exhibit 23](#_Toc133477401)

[EDA & Model Development 23](#_Toc133477402)

[Resizing Images for Consistency (Wood Anomaly Detection\_Rename\_Data Prep and Initial Modeling\_64\_64.ipynb): 24](#_Toc133477403)

[Code for Average Image (Wood Anomaly Detection\_Rename\_Data Prep and Initial Modeling\_64\_64.ipynb): 25](#_Toc133477404)

[Image Rotation (Wood Anomaly Detection\_Rename\_Data Prep and Initial Modeling\_64\_64.ipynb): 25](#_Toc133477405)

[Edge Detection (Wood Anomaly Detection\_Rename\_Data Prep and Initial Modeling\_64\_64.ipynb): 26](#_Toc133477406)

[Isolation Forest Modeling (Wood Anomaly Detection\_Rename\_Data Prep and Initial Modeling\_64\_64.ipynb): 26](#_Toc133477407)

[Autoencoder Modeling (Modeling with 128\_128\_Auto Encoders.ipynb): 27](#_Toc133477408)

[Web Application Development 27](#_Toc133477409)

[Code for Autoencoder Prediction (supportFunction.py): 28](#_Toc133477410)

[Code for Main Application (woodAnomaly.py): 28](#_Toc133477411)

[Code for Styling (custom.py): 29](#_Toc133477412)

[12. References 30](#_Toc133477413)

# 1. Project Background & Overview

Accurate and swift quality inspection is very essential for meeting planned demand as well as maintaining good customer satisfaction scores in any manufacturing industry [1, 2]. Majority of the industries employee workers for manual identification of product defects. Though manual QA is works well for minor industrial units, heavy manufacturing industries which mass produces physical goods needs autonomous system for manufacturing as well as quality inspection. Manual inspection will not only slow down the manufacturing process but also has serious impact business operating cost. Growing human resource cost is also one of the major factors which signals business to move out of traditional manual QA to automated one. As occurrence of defects/anomalies is a rare instance hence less data to train models, this makes modeling anomalies very difficult. This business problem of anomaly detection under the presence of low or no-data needs unsupervised machine learning and deep learning techniques instead of traditional classification-based models.

# 2. Problem Statement

Quality inspection is crucial for manufacturing industries to meet demand and maintain customer satisfaction3. Manual inspection works for small units, but heavy industries need an autonomous system for efficiency. Manual inspection slows down the manufacturing process and increases operating costs. Rising human resource costs prompt businesses to move to automated QA. Modeling anomalies is challenging due to the rarity of defects and lack of data.

Anomaly detection requires unsupervised machine learning and deep learning techniques [4, 5]. Traditional classification-based models are not suitable for this problem. The project focuses on using images of wooden objects to train unsupervised models. The models will identify possible defects in the object's texture. The ultimate goal current project is to improve quality of inspection and increase production efficiency.

# 4. Literature Review

There are limited number of research papers available in the web for identifying anomalies in wooden patterns/textures. For instances “An Improved Wood Recognition Method Based on the One-Class Algorithm”6 by Jie He & Co. used VGG16 pre-trained model for identifying anomalies in wooden patterns and “Detecting Faulty Piles of Wood using Anomaly Detection Techniques”7 by Jonathan Olsson used f-AnoGAN and GAN for identifying anomalies in wooden patterns. As per the above papers average prediction accuracy for anomalous class was around 65%. Which shows the relative ability of machine learning models for detecting anomalous wooden objects.

In our current proposal we suggest the use of unsupervised technique for anomaly identification. Apart of the same we have developed Isolation Forest and CNN Autoencoders for unsupervised anomaly detection.

# 4. Machine Learning Techniques used for Current Work

## Isolation Forest [8,9]

Isolation forest is an unsupervised machine learning technique for anomaly detection. It is a tree-based anomaly detection technique and works similar to that of a decision tree. The algorithm starts with the training of the data, by generating Isolation Trees. Each tree in an Isolation Forest is called an Isolation Tree. The algorithm randomly selects features and thresholds for each Tree and assigns a score based on the depth of the tree required to arrive at a given data point. An anomaly score of -1 is assigned to anomalies and 1 to normal points based on the contamination parameter. Isolation Forest is being used across industries for identifying anomalies/frauds in across industries like medial, manufacturing, cybersecurity and finance.

## Auto Encoders10

Autoencoders are form of unsupervised deep learning models where model training happens on the independent variables. Traditionally autoencoders has two major layer section, encoding layers and decoding layers. Encoding layer of the autoencoders create an feature compression effect similar to Principle Component Analysis while decoder layer tries to predict the features back from the compressed output of encoder layer. Basically, an autoencoder model tries to mimic the data and if any datapoint deviates a lot from already learnt distribution/patterns it gets treated as anomaly.

One of the major applications of autoencoder is anomaly detection where an autoencoder model is trained on completely normal datapoints, post training loss is calculated for each datapoint in trainset from which a train loss threshold is identifying (by observing loss distribution). During testing if any datapoint has loss greater than threshold defined (expected loss) it is flagged as anomaly. For encoding and decoding layers of autoencoder any type of layers can be utilized like dense, convolutional etc.,

# 5. Methodology

## 5.1 Overview

Since the data used for model training is set of images, initially images are preprocessed for size etc., before converting images into vector format for model development. Though the data source has tagged images on defect vs good objects we won’t be using defect images in model train as we are using unsupervised model to so the problem. Tagged test data with defects will only be used for model evaluation which data with only good objects images (one class) will be used for model training. For the current problem at hand, we are proposing the use of both ML and DL techniques for anomaly detection and final model producing better MAPE on test data will be selected for web application development. Following are the models proposed

* Isolation forest
* Autoencoders

Model performing best on train data will be saved as pickle or .h5 file which later is used for serving web application developed using streamlit 11

## 5.2 Data Source

Data for current project purpose has been extracted from Kaggle.com. Following is the source description as per Wiki “*Kaggle, a subsidiary of Google LLC, is an online community of data scientists and machine learning practitioners. Kaggle allows users to find and publish data sets, explore and build models*”.

**Method of Procurement:** Direct download from Kaggle.com

**Data Context:** Data consists of set images belonging to an wooden objected manufactured postindustrial operations. Images belong to a single product with looks like a wheel. Data provided has images for both good and faulty wooden objects under different folders. As the idea of the current project is to us unsupervised techniques for anomaly detection, we will be using only images under “train” and “validation” folders (which only contains images of object without defects) and “test” (which only contains images of object with defects) for evaluation of unsupervised model developed.

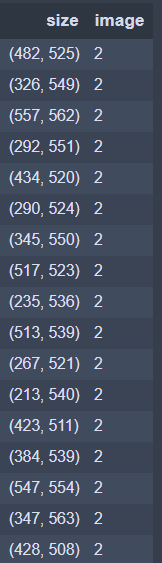
## 5.3 Data Cleaning and Transformation

Following data cleaning steps have been performed based on initial Exploration of the dataset.

### Fixing image size inconsistency [12, 13]

Image in the data source procured has varied dimensions [Refer to table 1 below]. Image size consistency is very crucial for modeling. Because of the following reasons

* Each pixel of image will be treated as feature/column in a table for modeling. It is essential to have all the images in training and test dataset to have same size for easy modeling
* Image rescaling to small size also helps in reducing the number feature there by reducing model runtime. In current project we have resized all image to 60x60 initially



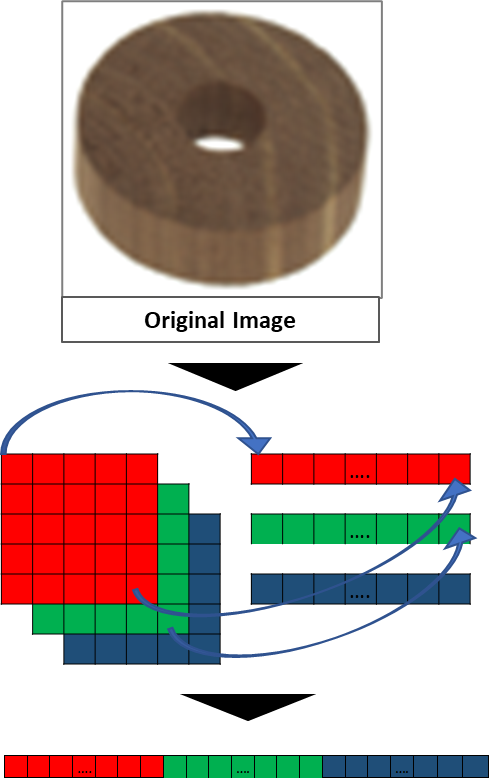
**Table 1:** Table showing a sample of images size and corresponding count of images

### Image to Pixel Array Conversion

For any machine learning model to be developed data should always be present in numerical as well as tabular format. Since we are using image for modeling training, images first have to be converted into numerical/model understandable format. Following steps have been performed for image conversion to numerical format [Refer to Figure 1 below]

* RGB values are extracted from colored resized images
* Extracted RGB values are stacked together to from a single 1D array of RGB values

Above process is followed for each image and final base table for model processing has been created



**Figure 1:** Image to RGB conversion. Then to 1D Array

### Addressing data quantity/size issue

Training data extracted from the source only has ~950 images for training which is relatively lower for training a good machine learning model. To address this problem, we will be using image rotation technique14 where an image will be rotated around the median by a specified angle. In the current project we are apply three rotations to original image there by generating 3x additional data for modeling [Refer to figure 2 below].

We have applied rotation by **90°, 180°** and **270°** for each original image which helped us in increasing the training data size from ~950 images to ~3800 images.

**

**Figure 2:** Rotation of Original Image by 90 90°, 180° and 270°

# Feature Selection and Engineering

Following feature engineering steps have been performed on the data based on insights from initial exploratory data analysis.

## Object Edge Extraction

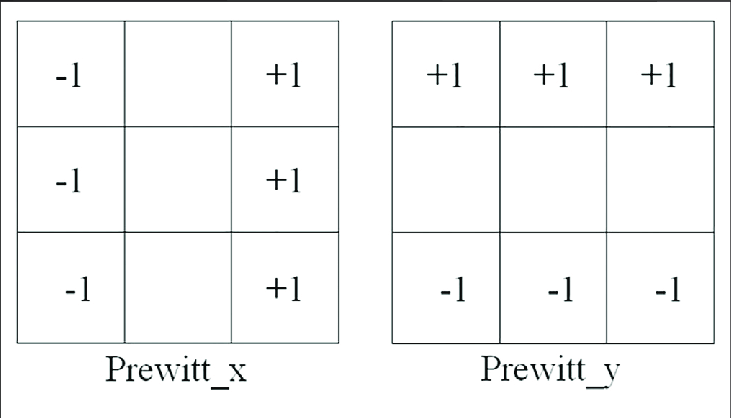
Complex texture and patterns on wooden surfaces might confuse the model for surface irregularities with natural patterns. This uncertainty can be reduced to some extent we could elevate the deviation from the entire objected and edge detection techniques helps in achieving the same. Following are the few of the advantages of edge detection

Building model on edge extracted images helps in

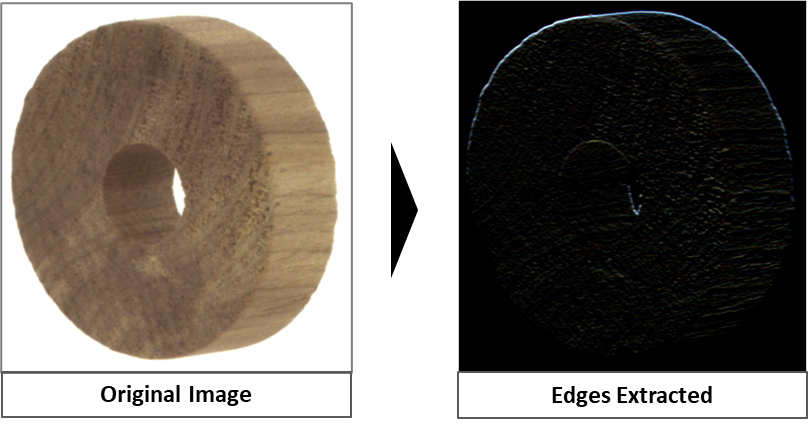
* + Reducing the noise in images there by improving modelling accuracies
  + Edge extracted images also helps in dimensionality reduction as image gets converted into grey scale

Prewitt Kernel 15 has been used for edge extraction and this edge extracted image are used for model development. Prewitt edge detection is a widely used method for detecting edges in digital images. It works by calculating the gradient of an image using a set of convolution masks or filters. These filters are designed to detect edges by emphasizing the differences in intensity values between adjacent pixels.

The Prewitt operator consists of two 3x3 convolution masks, one for detecting horizontal edges and the other for vertical edges. The horizontal mask is [-1 -1 -1; 0 0 0; 1 1 1] and the vertical mask is [-1 0 1; -1 0 1; -1 0 1] [Refer to figure 3 below]. These masks are convolved with the image to calculate the gradient in the horizontal and vertical directions.



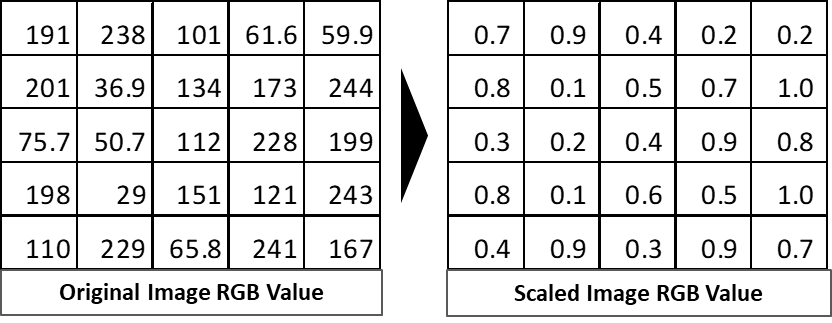
**Figure 3:** Prewitt Kernel



**Figure 4:** Image before and After Edge Extraction

## Image Pixel Scaling

Large feature values will result in unstable deep learning models due to large weights. To avoid this problem of unstable models and to make model process faster we are scaling input features using min max scaler i.e., dividing by 255 [Refer to figure 5]. Scaled image pixel are used for final model development



**Figure 5:** Image Before and After Scaling

# 7. Model Selection and Evaluation

As unsupervised anomaly detection models selected (Isolation Forest and Autoencoders) are trained to identify abnormal images from rest of the images, evaluation metrics used for traditional classification models should work well. Precision and Recall metrics are given higher focus due to lower magnitude of anomalous images in comparison with normal images

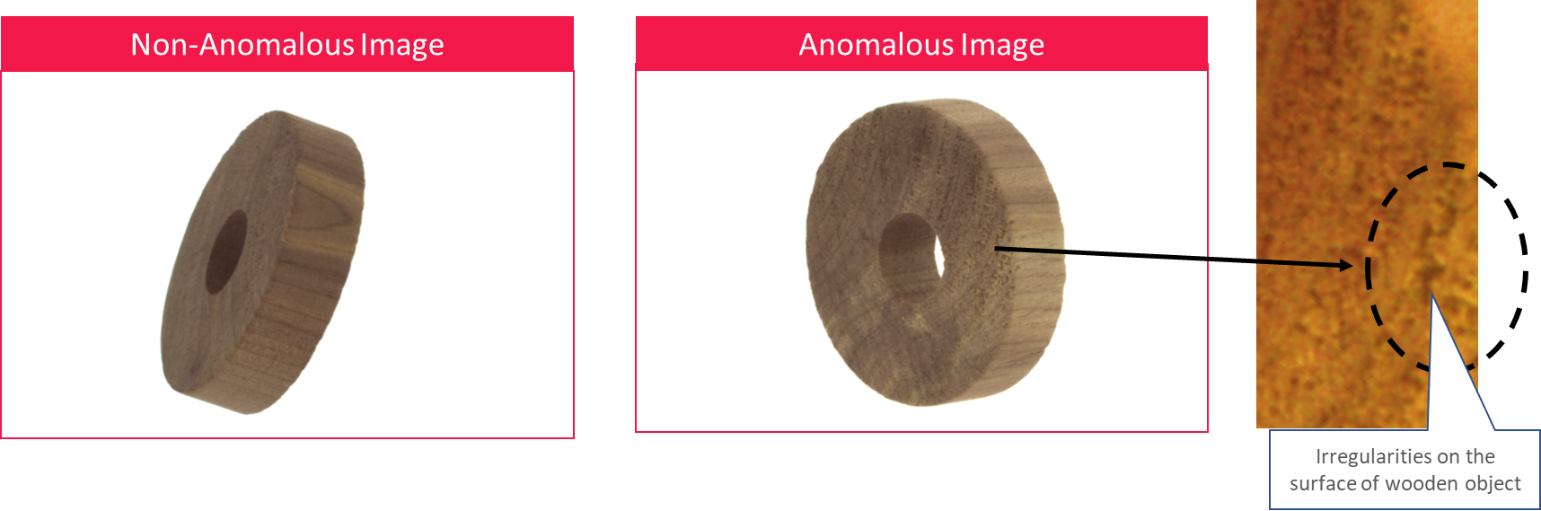
# 8. Model Results and Discussion

## Descriptive Analysis

* As its observed from the initial EDA that data extracted has only 953 image which is a bit less from ML/DL models to learn
  + To address this data quantity issue image rotation technique can be used for increasing data volumes for training
* We have 1,257 unique image dimensions out of 1,280 images this shows that as part of pre-processing we need to resize the images before we start model development

### Analyzing Sample Anomalous vs. Non-Anomalous Images

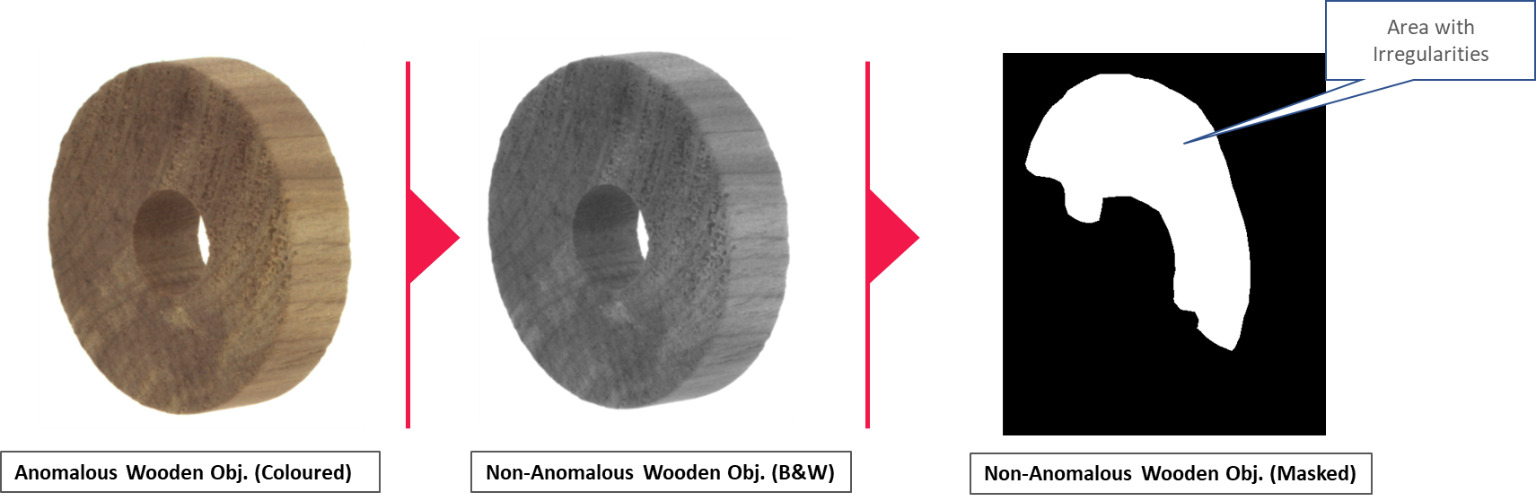
Based on initial observation of the images, anomalies on the wooden object are majorly observed due to irregularities on the image while a proper object is expected to be smooth [Refer to Figure 6 below]



**Figure 6:** Anomalous vs Non-Anomalous Images

### Understanding an Anomalous Object

* Anomalies on the wooden object are majorly observed due to irregularities on the image [Refer to Figure 7 below]
* Rough surfaces/Irregularities on the wooden object have to be identified by machine learning models. This can be challenging as irregularities are appearing similar to the colour patterns on the surface

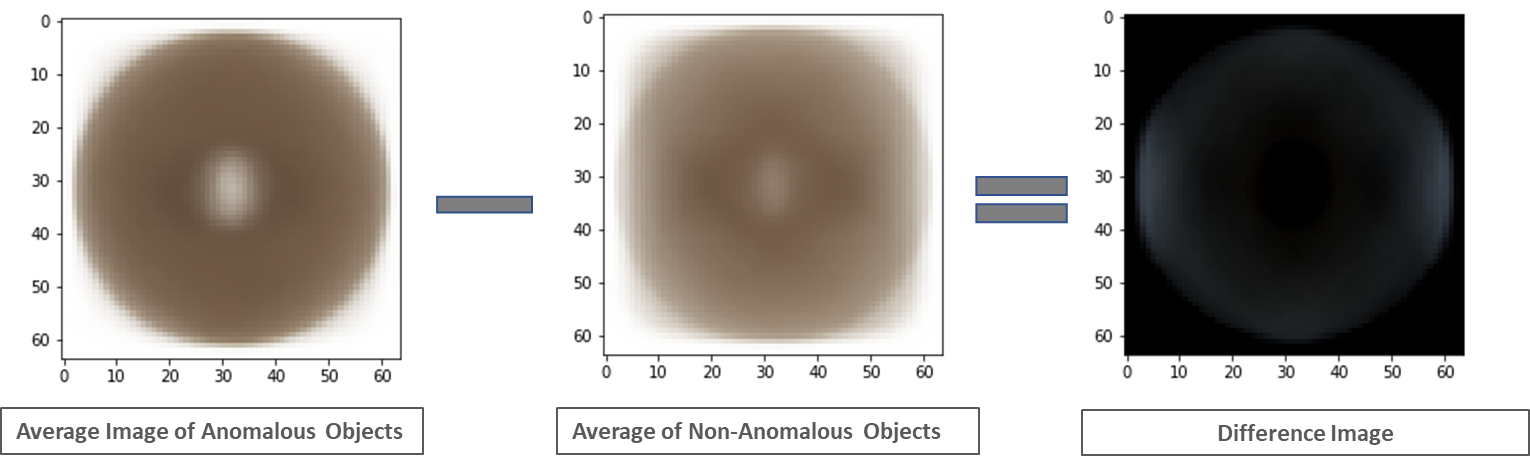


**Figure 7:** Surface Irregularities on Anomalous Images

### Do we observe any significant differences in an anomaly vs non-anomaly object?

A general idea differences between on all anomalous vs non-anomalous images can be identified by observed an “Average Image” or “Variance Image”. An average image is an image obtained by averaging pixel values of individual images similar a variance image is an image obtained by calculating standard deviation of pixel value of individual images.

Average image didn’t provide much information on anomaly vs non-anomaly due the data issue i.e., we have image with different dimension as well as rotations which in turn is causing issues for comparison [Refer to Figure 8 below].



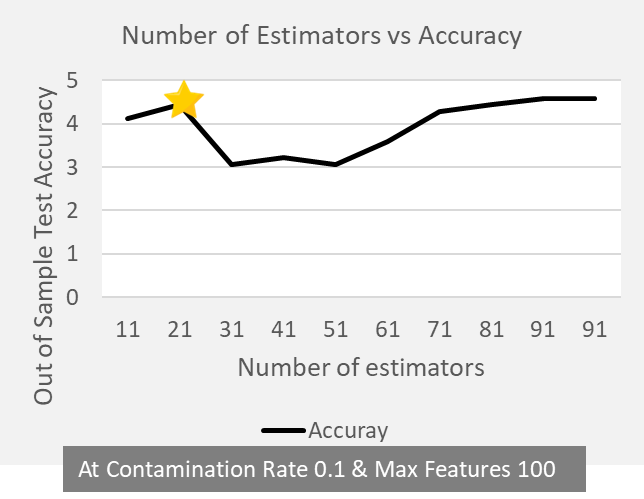
**Figure 8:** Average Image of Anomalous vs Non-Anomalous Objects

## Model Evaluation

### Isolation Forest – Parameter Tuning and Evaluation

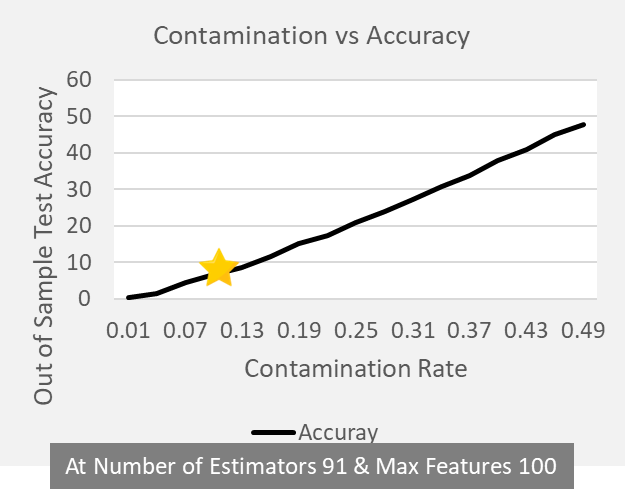
As per initial propose architecture, unsupervised isolation forest model has been developed and tuned on ***Number of Estimators***, ***Contamination rate*** and ***Max Features*** parameters. Following are the high-level results and observations

* **Number of Estimators:** Model Exhibited best accuracy at low (<20) and higher number (>71) of estimators. Number of estimators = 91 is considered as best value due high accuracy (~4.5%) and low complexity (fewer model iterations) [Figure 9 below]



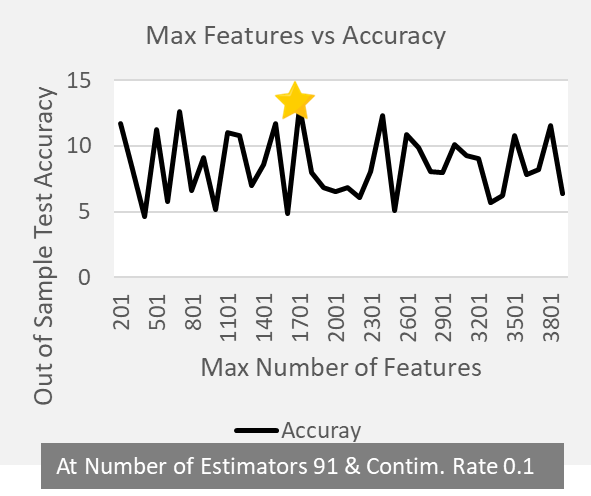
**Figure 9:** Isolation Forest Test Accuracy vs Number of Estimators

* **Contamination Rate:** Model’s capability to increase anomalies on out sample test data is increasing with increase in contamination rate. But this is not desirable as the increase in contamination rate will increase False positive drastically. Due to this reason Contamination rate of 0.1 is selected as balance threshold [Figure 10 below]



**Figure 10:** Isolation Forest Test Accuracy vs Contamination Rate

* **Max Features:** Model exhibited mixed performance with change in number of features. At number of features 1700 (with best # Estimators & Contamination Rate) best model accuracy has been observed with a value of ~13% [Figure 11 below]



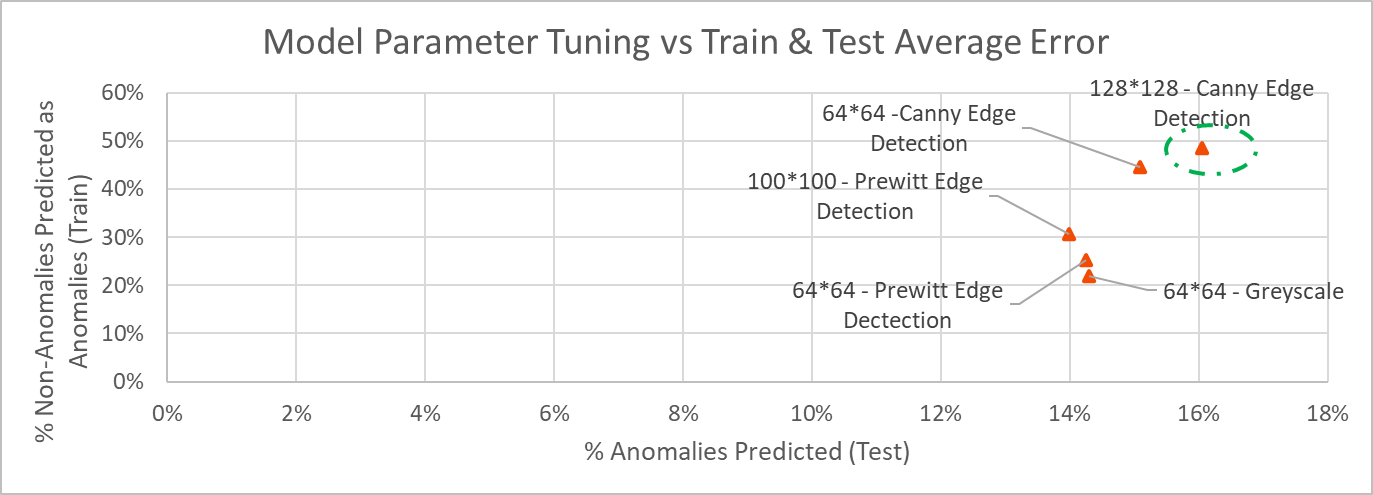
**Figure 11:** Isolation Forest Test Accuracy vs Max Features

### Auto Encoders – Parameter Tuning and Evaluation

Developed CNN Autoencoder are trained and test for

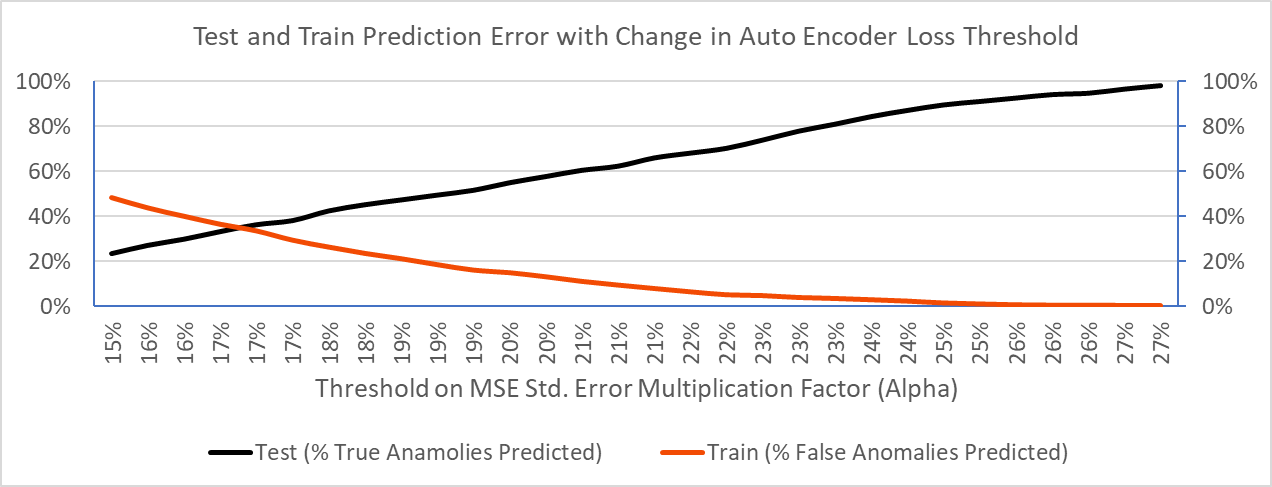
* + By varying model architecture (units and depth)
  + Data Treatment
    - Varying images sizes (64\*64 and 128\*128)
    - Varying Edge Detection techniques (Prewitt and Canny Edge Detection)
  + Varying learning rate and Epochs

Fined tuned **128\*128 resized image** data with **Canny edge detection [16, 17}** exhibited best performance in comparison with others. This model has been selected as final model for anomaly detection. Refer to Figure 12 below showing model test vs train error across iterations



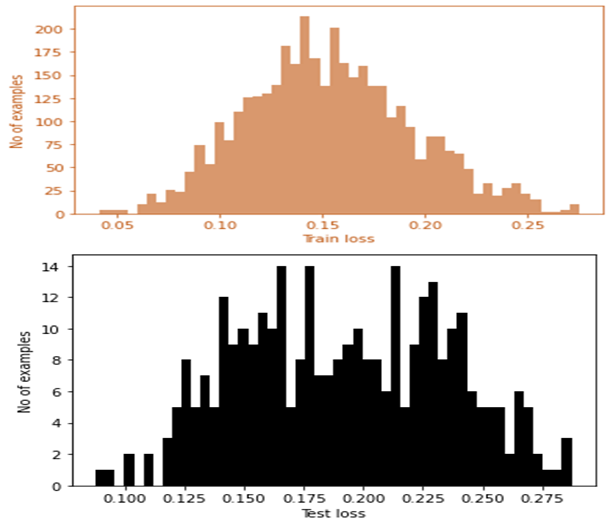
**Figure 11:** Autoencoder Test vs Train Error across Iterations

* As we increase threshold value on train loss for classifying anomaly vs non-anomaly test accuracy is increasing while train accuracy is decreasing (refer to chart above
  + Based on afore mentioned observation threshold on train error is set in such a way that we don’t observe a lot of false positive predictions [Refer to Figure 12]



**Figure 12:** Train and Test Error vs Loss Threshold

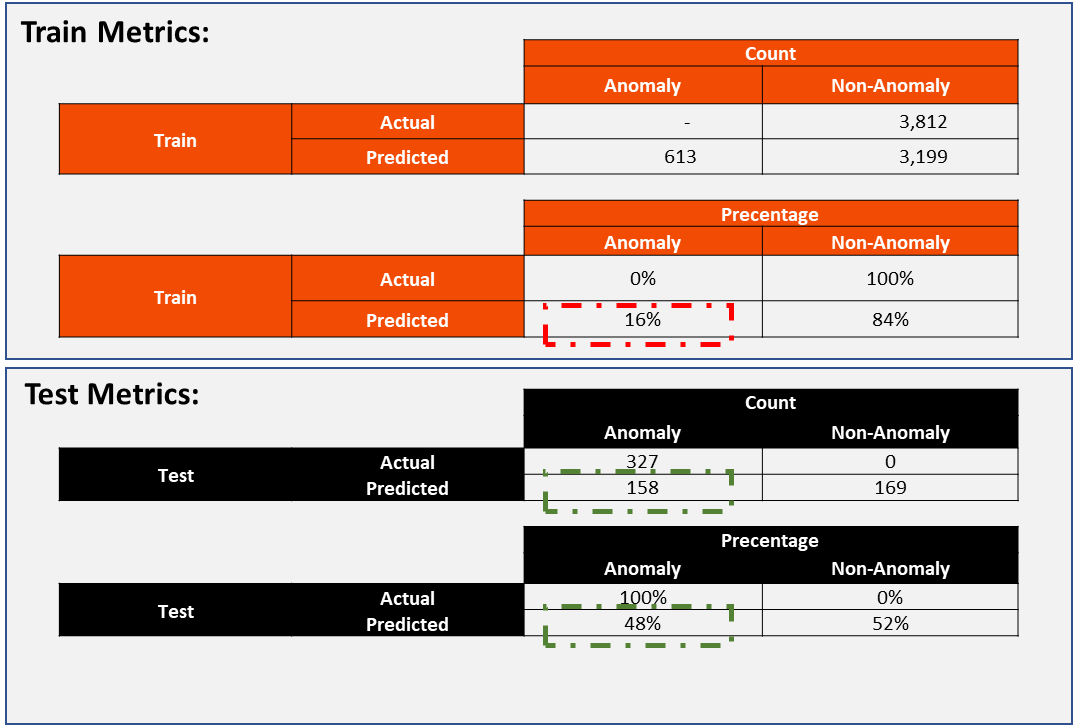
* Best model, with variance bias trade-off, is observed at an alpha value of 34% (i.e., an observation is anomaly if prediction loss is > Mean Loss + 0.34 (Standard deviation of Loss) [Refer to Figure 13]



**Figure 13:** Train and Test Loss Distribution

### Autoencoder – Final Model Deep dive

Autoencoders outperformed isolation forest in identifying anomalous images. Best autoencoder model could identify ~50% of anomalous images with 16% False Positive (FP) rate while isolation forest could identify only 13% of anomalous images at 10% Train FP rate. Based on this auto-encoder have been selected as final model for predicting anomalous objects. Refer to Figure 14 showing test and train confusion matrix



**Figure 14:** Autoencoder – Train and Test Confusion Matrix

# 9. Discussion on Key Findings

Contamination rate of random forest has been explored randomly due to the lack of estimate on the extent of anomalies in the data. Although the model performs well in predicting the positive class at higher contamination rates, increasing the contamination rate is not desirable due to high false positive rates. Isolation forest has a best accuracy of 12.99% on test data and might not be performing well due to the high dimensionality.

Autoencoders outperformed isolation forest in identifying anomalous images, with the best autoencoder model identifying around 50% of anomalous images with a 16% false positive rate. Current model's recall rate is around 50%, it suggests that the model is doing satisfactorily well. In the current project we couldn’t use the complete capabilities of deep learning due to lower data size. Further experimentation on larger dataset could provide improved results.

# 10. Conclusion

There are limited number of research papers available in the web for identifying anomalies in wooden patterns/textures. For instances “***An Improved Wood Recognition Method Based on the One-Class Algorithm***” by Jie He & Co. used VGG16 pre-trained model for identifying anomalies in wooden patterns and “***Detecting Faulty Piles of Wood using Anomaly Detection Techniques***” by Jonathan Olsson used f-AnoGAN and GAN for identifying anomalies in wooden patterns.

As per the above papers average prediction accuracy for anomalous class was around 65% and in our current problem settings, we could achieve an accuracy of ~50% which suggests that our model is doing satisfactorily doing well. Also, we can compare these papers to our model exactly as dataset used for analysis are different. Since we had very few datapoints for model train we could not fully utilized the capabilities of deep learning.

# 11 Code Exhibit

## EDA & Model Development

**File Names:** Wood Anomaly Detection\_Rename\_Data Prep and Initial Modeling\_64\_64.ipynb and Modeling with 128\_128\_Auto Encoders.ipynb

**Folder:** code

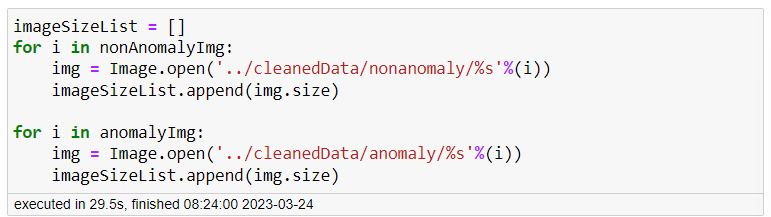
**Description:** Wood Anomaly Detection\_Rename\_Data Prep and Initial Modeling\_64\_64.ipynb file has the code for data exploration and initial Isolation forest model on 64\*64 resized image. While the file Modeling with 128\_128\_Auto Encoders.ipynb has modeling code autoencoders on 128\*128 resized data

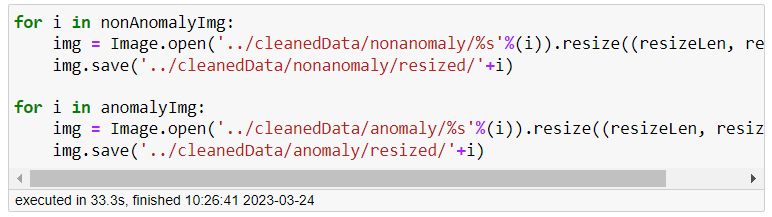
**Programming Language**: Python

**IDE:** Jupyter Notebook

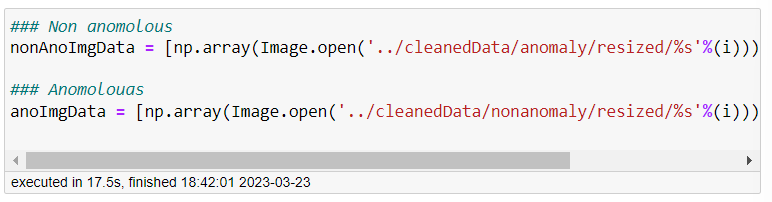
**Packages:** Pandas, Numpys, Matplotlib, PIL, sklearn, keras and tensorflow

### Resizing Images for Consistency (Wood Anomaly Detection\_Rename\_Data Prep and Initial Modeling\_64\_64.ipynb):

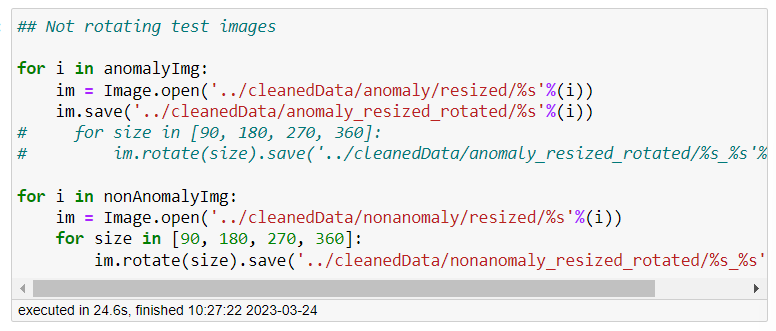




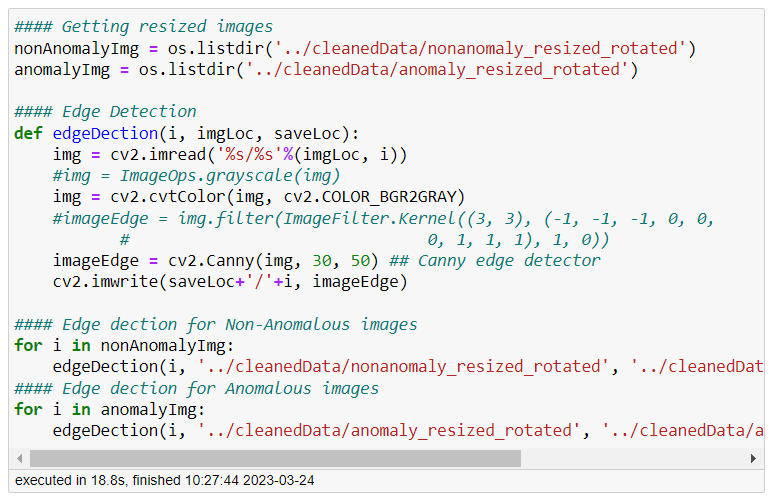
### Code for Average Image (Wood Anomaly Detection\_Rename\_Data Prep and Initial Modeling\_64\_64.ipynb):



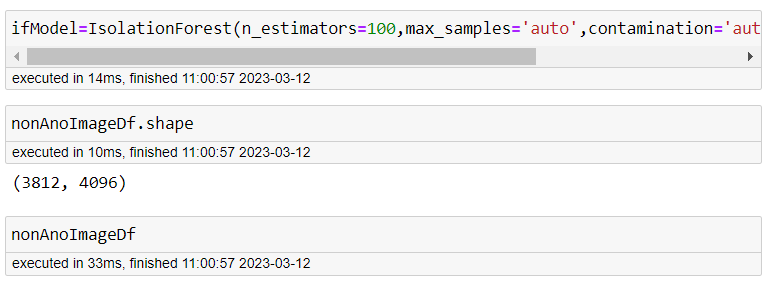
### Image Rotation (Wood Anomaly Detection\_Rename\_Data Prep and Initial Modeling\_64\_64.ipynb):



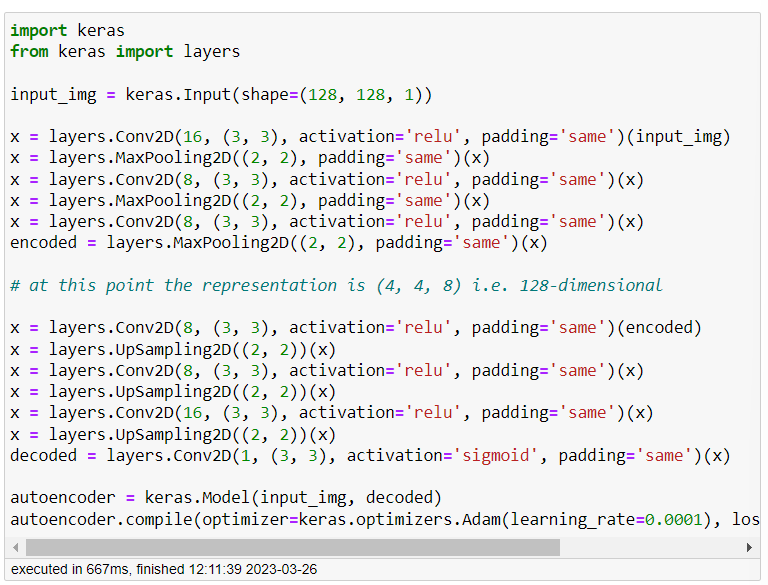
### Edge Detection (Wood Anomaly Detection\_Rename\_Data Prep and Initial Modeling\_64\_64.ipynb):



### Isolation Forest Modeling (Wood Anomaly Detection\_Rename\_Data Prep and Initial Modeling\_64\_64.ipynb):



### Autoencoder Modeling (Modeling with 128\_128\_Auto Encoders.ipynb):



## Web Application Development

**File Names:** woodAnomaly.py, supportFunction.py and custom.css

**Folder:** app\woodAnomaly

**Description:**

woodAnomaly.py – File has code for application UI elements definition

supportFunction.py – File has code for function for prediction using autoencoders

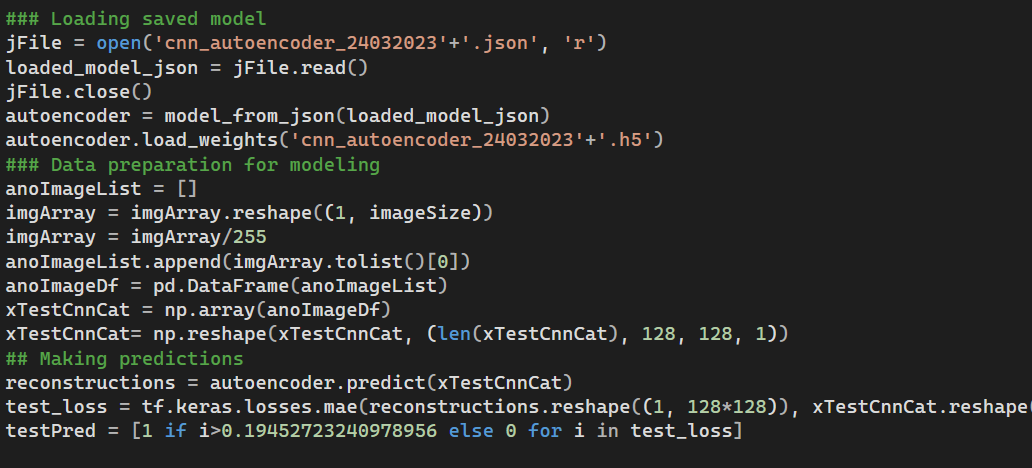
custom.css – File has CSS code for application styling

**Programming Language**: Python

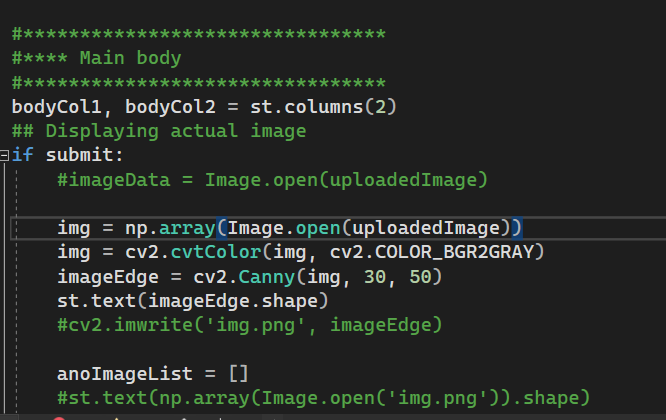
**IDE:** Jupyter Notebook

**Packages:** Pandas, Numpys, Matplotlib, PIL, sklearn, keras and tensorflow

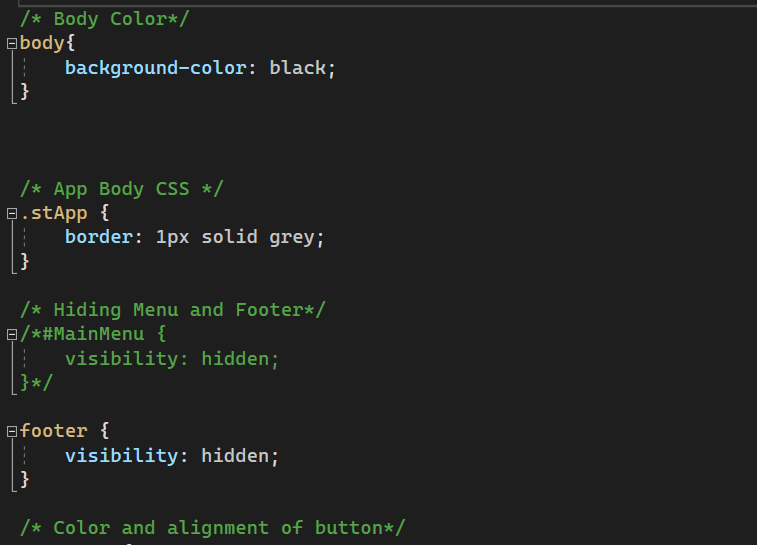
### Code for Autoencoder Prediction (supportFunction.py):



### Code for Main Application (woodAnomaly.py):



### Code for Styling (custom.py):



# 12. References

[1] Smith, J. (2021). The importance of industrial automation in the modern manufacturing industry. Journal of Manufacturing Technology Management, 32(4), 712-725. https://doi.org/10.1108/JMTM-01-2020-0011

[2] Brown, T., & Williams, S. (2019). Industrial automation and the future of work. International Journal of Industrial Ergonomics, 72, 118-127. <https://doi.org/10.1016/j.ergon.2019.04.001>

[3] Zhang, Y., & Han, Y. (2018). The importance of quality inspection in manufacturing industry. MATEC Web of Conferences, 246, 01035. <https://doi.org/10.1051/matecconf/201824601035>

[4] Zhang, J., Wu, Y., & Huang, H. (2019). Anomaly detection under low data using machine learning. Journal of Intelligent Manufacturing, 30(4), 1337-1348. https://doi.org/10.1007/s10845-017-1392-5

[5] Ma, X., & Cui, Y. (2020). Anomaly detection for low data using machine learning. Future Generation Computer Systems, 103, 260-267. <https://doi.org/10.1016/j.future.2019.08.058>

[6] He, J., Zhang, J., Yang, Y., & Wu, X. (2021). An Improved Wood Recognition Method Based on the One-Class Algorithm. IEEE Access, 9, 96215-96224. <https://doi.org/10.1109/access.2021.3094279>

[7] Olsson, J. (2021). Detecting Faulty Piles of Wood using Anomaly Detection Techniques. Master's thesis, Luleå University of Technology, Sweden. Retrieved from http://urn.kb.se/resolve?urn=urn:nbn:se:ltu:diva-81557

[8] Liu, F. T., Ting, K. M., & Zhou, Z. H. (2008). Isolation-based anomaly detection. In Proceedings of the 23rd International Conference on Machine Learning (ICML) (pp. 379-386). ACM. https://doi.org/10.1145/1390156.1390208

[9] Liu, F. T., Ting, K. M., & Zhou, Z. H. (2012). Isolation forest. In Proceedings of the 2012 11th International Conference on Machine Learning and Applications (ICMLA) (pp. 450-456). IEEE. <https://doi.org/10.1109/ICMLA.2012.62>

[10] Wang, W., Li, W., Ding, H., Li, P., & Li, Y. (2019). Anomaly detection in mechanical systems based on deep autoencoder neural network. Complexity, 2019, 1-10. <https://doi.org/10.1155/2019/6791252>

[11] Allaire, J., Cheng, J., Xie, Y., McPherson, J., Chang, W., Allen, J., ... & Waskom, M. (2020). Streamlit: A Python framework for building and sharing web apps for machine learning. Journal of Open Source Software, 5(51), 2144. <https://doi.org/10.21105/joss.02144>

[12] "An Introduction to Image Processing with Python" by Michael Galarnyk: https://towardsdatascience.com/an-introduction-to-image-processing-with-python-af8d0db854b7

[13] "Image Processing in Python: Algorithms, Tools, and Methods" by Stan Tyan: <https://www.altoros.com/blog/image-processing-in-python-algorithms-tools-and-methods/>

[14] Shorten, C., & Khoshgoftaar, T. M. (2019). Data augmentation for deep learning: A survey. IEEE access, 7, 36398-36415.

[15] Sinha, N., Rajbongshi, A., & Dandapat, S. (2015). Edge detection techniques for image segmentation-A survey of soft computing approaches. International Journal of Computer Applications, 118(19), 23-33.

[16] Canny, J. (1986). A computational approach to edge detection. IEEE Transactions on pattern analysis and machine intelligence, (6), 679-698.

[17] Rahmati, M., & Sadri, S. (2019). Comparative study of canny and sobel edge detection methods for image analysis. Journal of Computational and Applied Research in Mechanical Engineering (JCARME), 9(1), 1-8.