~~1. Sort out folders for Gears~~

2a Generate images for springs

2b1 table and graph

2b Write up: Different image types with different thresholds for different parts of a spring. Variations in angle, distance and lighting may also impact performance. Need sub-classification of the springs dataset. This could be performed manually, or by unsupervised learning techniques. Also need different loss metric.

2c sort out folders for springs

3a Complete pipeline code

3b write up

3c add two pipeline diagrams

3d sort out folders

4a Generated synthetic images: resize, get results

4b write up

4c sort folders

5 future work

6 commit to git

7 email petra

TODO:

Read image and segment (determine threshold for gear and for spring)

Mass generate outputs for test folder.

Get test accuracies

Produce plots

(Was there k-fold applied?)

---

Understand steps and epochs in tensorflow

Train classifier over multiple epochs > Create pipeline for read in image > classify > output segmentation > print to screen and file.

Constraint: no access to a CUDA-enabled GPU. CPU processing is slow. Therefore, the number of tests and training of models is limited. However, the structure and is shown and possibilities for further tests are discussed.

# Segmentation

Generated images

Practically, it may be effective to save the model after each epoch. This will allow for testing of earlier model weights in the event of over training.

Try other segmentation models, eg. …?

GAN for image generation.

Synthetic images with more realistic surface generation, reflections and lighting conditions, including variations of the angle of light.

==

Springs: epochs: 4, due to time constraints

Lower F1 and AUROC than gears. Possibly due to wider range of angles and surface shapes from which images were captured. It may be more effective to train separate segmentation models for each surface type and train a classifier (or use classic image processing) to determine which segmentation model to use.

TODO: threshold?

Train | Loss | F1 | AUROC

Val

Test

Graph

Sample images

Real images

Generated images

Model.load.

Loss function: 3 channels CrossEntropyLoss > better for future, if additional mask colours for different types of faults.

Future work: Better lighting when capturing images. E.g. Different coloured lights at different angles cast different shadows, which may assist with highlighting faults. Alternatively, 3D volumetric images would assist in detecting irregularities in surfaces, but hardware may be expensive for high pixel density over a small area.

Constraints: GPU, loadshedding

# Classification

Used transfer learning on the VGG-16 (Very Deep Convolutional Networks for Large-Scale Image Recognition) convolutional neural network. It’s a commonly used baseline. Small and fast, compared to ResNet and newer classification models, which may be overkill.

Skipped image augmentation for simplicity. Although, augmentation may make classification more robust and accurate, especially on real images or different synthetic images.

Training validation and testing were performed on an approximately 8:1:1 ratio.

loss: 7.6346 - acc: 0.7111 - val\_loss: 1.6257e-11 - val\_acc: 1.0000

Training accuracy: 0.71%

Validation accuracy: 100%

Test accuracy: 100%

This model can now be used to create a pipeline to classify an image as either gear or spring and call the relevant segmentation model.