# ASSIGNMENT

ISLAM RAIHAN ULL

# TIMELINE :

Assignment assigned : 2024-10-04
Start date : 2024-10-06 (5-6 hours~)
End date : 2024-10-07  (2-2.5 hours~)
Approx time-cost : 8 hours

# ASSIGNMENT REFERENCE :

Assignment Link :
https://drive.google.com/file/d/1RGjU93kkpfemf2aTyr8BPrY2zqVPJpYx/view

# QUESTIONNAIRE PART-1 :

a. How many samples are there?

*Ans : There are total 300 training samples for binary classification. 250 of the samples are OK class and 50 are NG class*

b. Are the samples evenly distributed?

*Ans : No, there is a clear bias towards OK data of 5:1. However, in the world of manufacturing, this sort of bias is expected.*

c. Does the test data appear to be representative of the training data? What would you check to answer “yes” or “no” with certainty?

*Ans : YES. I think the test data is representative of the training data.*

# QUESTIONNAIRE PART-1(cont.) :

d. What kind of anomalies are there?

*Ans : The majority of the anomalies are the incorrect/broken threads. There are some where there are spots in the shank. Please refer to fig-1 for some samples*

• Do you think all of these anomalies should be considered as such specifically for the client's problem, or not, and why?

*Ans : All of the anomalies should be considered for the client. Because the final goal of the model is to pass the screws that are perfect. Any sort of defected screws should be caught and sent back from the assembly line*

• Can you think of any other likely anomalies that do not seem to be represented in the data?

*Ans : Yes. It is unclear if 360 degrees of a single screw is considered or not. For OK screws, there actually might be defects on the opposite side. Also, the head of the screw cannot be assessed from this angle to determine manufacturing defects.*

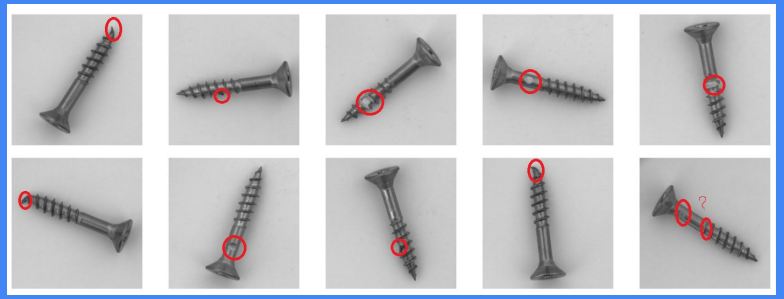

fig-1

# QUESTIONNAIRE PART-1(cont.) :

e. Does the division of the data into good and not good appear to be correct? How do you verify this, and how would you go about it if the dataset was larger?

*Ans : The division of data is 'correct' because in practical life, any mature manufacturing should have less NG samples than OK samples. If the dataset was larger,*

- *Run the model for longer*
- *Make the model more robust. It is unclear the robustness of the predictions*
- *Use less augmentation*

f. Is the data sufficient to train a model?

*Ans : While this is not ideal, the data is sufficient to train a model with above-average performance.*

• If you were to suggest using data augmentation, what image data augmentation procedures are possible, and what are to be avoided, given the nature of the objects in the picture?

*Ans : Brightness, contrast, horizontal and vertical flips are possible. With brightness and contrast adjustment, we have to be careful not to over-do it as to keep the lighting condition as close to baseline as possible. Rotation is also possible but anything above 10 degrees is risky as it cuts parts of the screw. Please refer to fig-2 for example case. Maybe coarse dropout Can also be used but I have not checked that.*

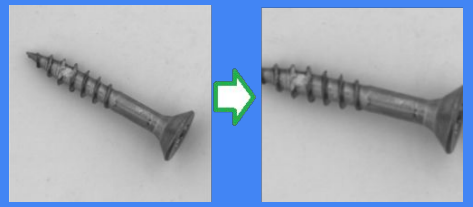

fig-2

# QUESTIONNAIRE PART-1(cont.) :

• If you were to recommend to the client to record additional data, what instructions would you give to them, and why?

*Ans : two approaches can be done.*

- *First is more data, specially for the NG cases.*
- *Second would be a different approach, picture of a single screw from maybe 3 different angles. One picture, rotate 180 degrees, another picture, and a picture of the head. Then combine the 3 images into a single matrix data for training.*

g. What kind of problems can we expect?

*Ans : There is a significant risk of robustness being weak with the small sample size of data.*

h. What can we do to improve the detection of anomalies in this data?

*Ans : Image processing before training for feature enhancement is very important in my opinion. I saw CLAHE algorithm enhancing the overall quality/sharpness of the image significantly. If there was more time I will definitely try some other algorithms.*

i. Does the data contain any patterns or traits that can be useful for detecting anomalies and that can be identified in a given image automatically?

*Ans : The broken/imperfect threads are very obvious and should be relatively easy to detect automatically.*

j. Optional:Anythingelsethatyouthinkisimportant(graphs,visualizations,measurements...)

*Ans : Will be added with Model explanation in Part-2.*

# QUESTIONNAIRE PART-2 :

1. **Data Splitting Strategy:**

There are 250 images of "good" screws and 50 images of "not-good" (NG) screws. This makes the dataset imbalanced, with the "good" class significantly more represented than the "NG" class. I tried to balance it by generating augmented NG samples and added additional 200 images
The 80:10:10 split ensures that we have a training, validation, and test set that represent both classes. I also used CLAHE for each of the image before loading them into the dataloader.
split:
Training Set (80%): 400 images (200 good, 200 NG)
Validation Set (10%): 50 images (25 good, 25 NG)
Test Set (10%): 50 images (25 good, 25 NG)
Stratified sampling helps to avoid bias where only "good" data dominates a particular split.
If the test images had ground truth, further assessments could have been made.

# QUESTIONNAIRE PART-2 (cont.) :

2. **Hyperparameters and Model Choices:**

- ***Model Architecture:*** I chose ResNet18 pretrained on ImageNet due to its effectiveness in transfer learning. Given that the images are single-channeled, I adjusted the model's first layer to accommodate the single-channel input. ResNet is a good candidate for anomaly detection. If I had more data I think I would have used a CAE or VAE.
- ***Image Size:*** I scaled down the images from 1024x1024 to 256x256 to have a good balance between preserving critical features in the images while optimizing for training speed and memory usage.
- ***Freezing Layers:*** Initially, I opted to freeze all layers except the last few. This allows me to leverage the pretrained model's existing knowledge while focusing the training on the final layers. I added a "freeze" flag to experiment with unfreezing the entire model if needed.
- ***Loss Function:*** Since this is a binary classification problem, I chose binary cross-entropy loss, which is standard for such tasks.
- ***Optimizer:*** I selected the Adam optimizer with a default learning rate of 1e-5 as I do it as a rule-of-thumb for varying gradient magnitudes during training.
- ***Weight decay :*** L2 weight decay helps prevent overfitting by penalizing large weights, encouraging the model to learn more generalized patterns. This is especially useful in anomaly detection, where data is often limited, and in transfer learning, where it helps adapt the pretrained ImageNet weights to the new task.
- ***Learning Rate Scheduling:*** To avoid overfitting and to allow smooth convergence, I implemented a ReduceLROnPlateau scheduler. This reduces the learning rate when the validation loss plateaus, which can help maintain learning momentum while preventing overfitting.
- ***Early Stopping:*** I set early stopper. This ensures that training halts if no improvements are observed in validation accuracy over several epochs, preventing unnecessary overtraining.

# QUESTIONNAIRE PART-2 (cont.)

```
Accuracy: 0.9600
Precision: 1.0000
Recall: 0.9200
F1 Score: 0.9583
```

fig-3

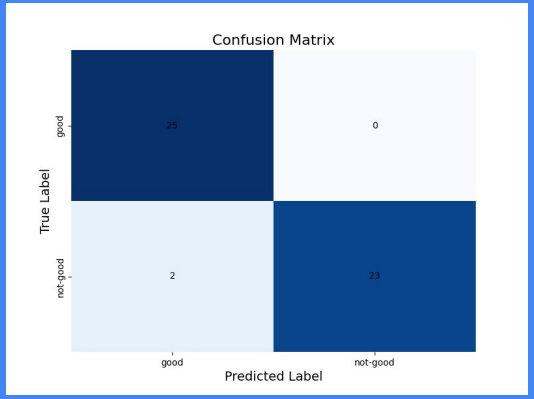


fig-4

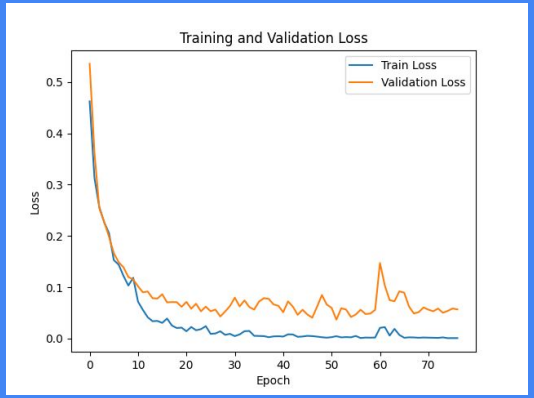


fig-5

## 3. Model Evaluation :

**Accuracy, Precision, Recall, F1 Score (Fig-3)**:

- Accuracy is 96%, indicating the model is generally making correct predictions.
- Precision is 1.0, meaning every "not-good" prediction was correct (no false positives).
- Recall is 0.92, meaning the model missed 8% of actual "not-good" images, reflecting in the 2 misclassifications in the confusion matrix.
- F1 score of 0.9583 shows a good balance between precision and recall, confirming the model's overall robustness.

**Confusion Matrix (Fig-4)**:

- The model correctly classified 25 "good" and 23 "not-good" images.
- Two "not-good" images were misclassified as "good," which contributes to the recall being slightly less than perfect.

**Loss Curve (Fig-5)**:

- Both training and validation loss decreased smoothly, indicating good model training.
- Minor fluctuations in validation loss towards the later epochs suggest possible slight overfitting, but the overall trend is stable.

# QUESTIONNAIRE PART-2 (cont.) :

## 4. Analysis:

**Observations**:

- The original imbalance in the dataset (more "good" images) likely caused the slight drop in recall for the "not-good" class, despite data augmentation.
- The low learning rate likely contributed to the model's stability but may have slowed down convergence.

**Improvements**:

- To improve recall, consider further augmenting the "not-good" class or using techniques like class weighting.
- Trying other loss functions is another avenue to explore for further improvements of the model.
- Visualizing attention maps could help fine-tune model performance by understanding where the model focuses.

# Additional :

I tried to generate attention map of the model on the test data to understand the model. Unfortunately the attention map was not conclusive of the feature extraction of the model. Fig-6 is example of that. Is it due to limited training data or due to my attention-map code, I am unsure.

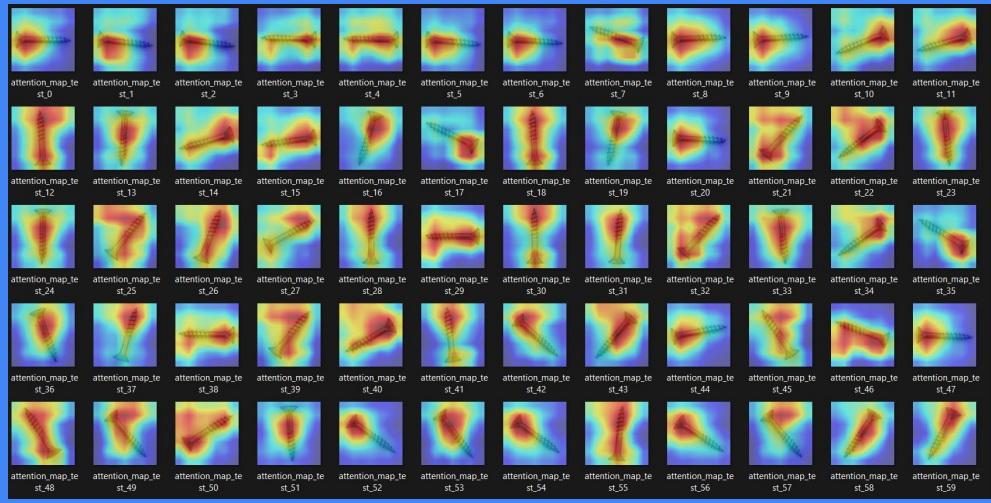

fig-6

# Environment setup :

1. docker build -t raihan-test-base:v0 -f Dockerfile ./
2. docker run -it --rm --name raihan-env-v1 --gpus all --shm-size=100g -v $PWD:/workspace -w /workspace raihan-test-base:v0
3. python main.py