Machine Learning Internship Questionnaire

Objective:
Develop a machine learning model that can classify images of industrial equipment into two categories: 'defective' and 'non-defective'

Methodology

I. Data Accusation and pre-processing

The Metal Nut Data had labelled images of nut bolt and their condition i.e. Bent, colour, flip, scratch, good. The data was taken from a website online. And it was a quite small dataset with around 40 images in each category. So data augmentation was required to increase the data for the model. Then the data was split into train, test and validation set for the model.

II. Model Architecture

The core of classification is series of convolution layers of kernel size (3, 3) with ReLU activation functions. Followed by MaxPool2D with kernel size (2, 2)
After that we flatten the image then classify them in five categories using SoftMax function.

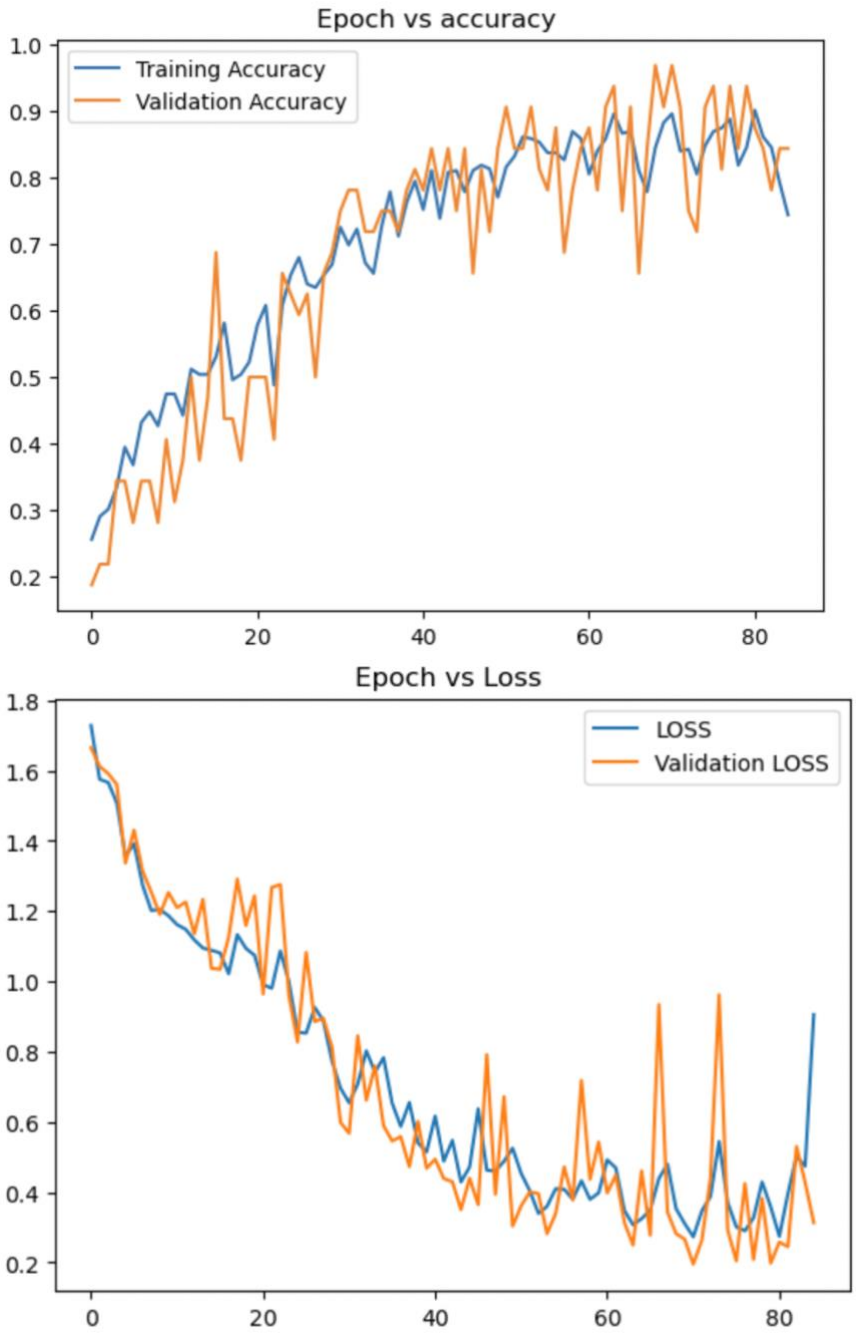
```
Model: "sequential_5"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 sequential_3 (Sequential)   (32, 256, 256, 3)         0

 sequential_4 (Sequential)   (32, 256, 256, 3)         0

 conv2d_6 (Conv2D)           (32, 254, 254, 32)        896

 max_pooling2d_6 (MaxPoolin  (32, 127, 127, 32)        0
 g2D)

 conv2d_7 (Conv2D)           (32, 125, 125, 64)        18496

 max_pooling2d_7 (MaxPoolin  (32, 62, 62, 64)          0
 g2D)

 conv2d_8 (Conv2D)           (32, 60, 60, 64)          36928

 max_pooling2d_8 (MaxPoolin  (32, 30, 30, 64)          0
 g2D)

 conv2d_9 (Conv2D)           (32, 28, 28, 64)          36928

 max_pooling2d_9 (MaxPoolin  (32, 14, 14, 64)          0
 g2D)

 conv2d_10 (Conv2D)          (32, 12, 12, 64)          36928

 max_pooling2d_10 (MaxPooli  (32, 6, 6, 64)            0
 ng2D)

 conv2d_11 (Conv2D)          (32, 4, 4, 64)            36928

 max_pooling2d_11 (MaxPooli  (32, 2, 2, 64)            0
 ng2D)

 flatten_1 (Flatten)         (32, 256)                 0

 dense_2 (Dense)             (32, 64)                  16448

 dense_3 (Dense)             (32, 7)                   455

=================================================================
Total params: 184007 (718.78 KB)
Trainable params: 184007 (718.78 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

III. Training process

The training process involved feeding the images to the model for 85 epochs and batch size of 32. Adam optimizer was used with loss function being Sparse Categorical Cross entropy loss with metric being accuracy.

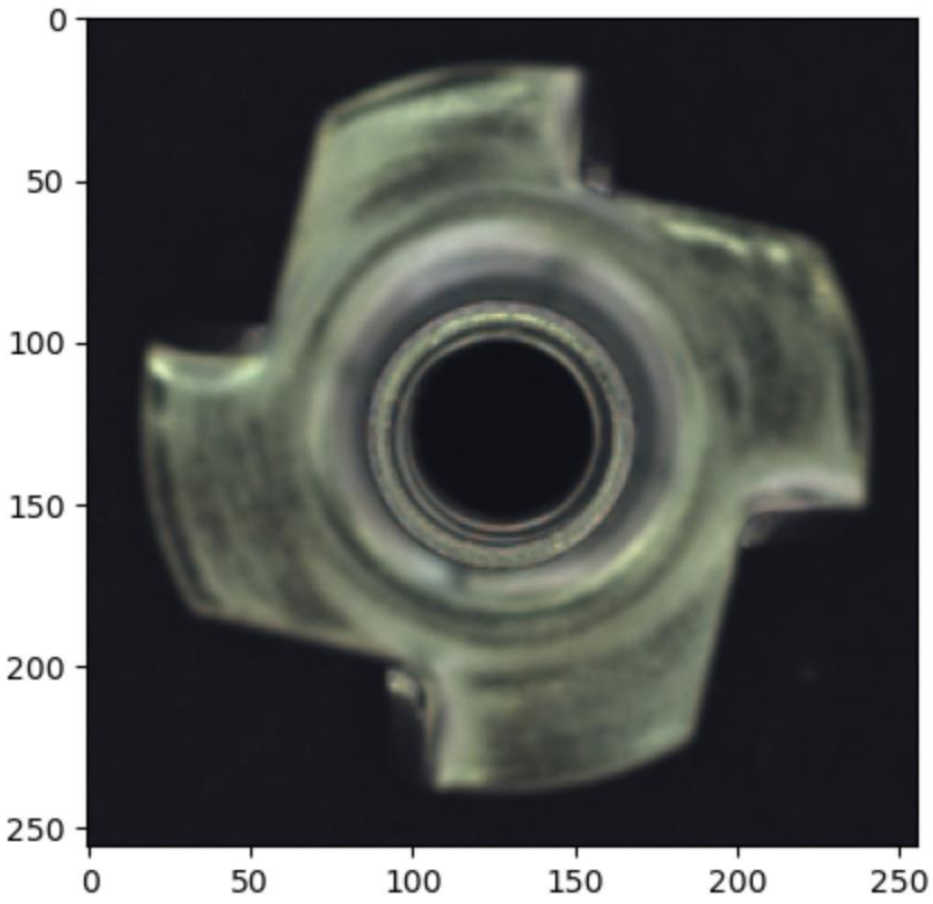
Result

The training result were varying with each epoch. But the test set got a decent result with accuracy of **0.8594.**



[49]:
```
for images_batch, label_batch in test_ds.take(1):

    first_image = images_batch[0].numpy().astype('uint8')
    first_label = label_batch[0].numpy()


    print("First Image")
    plt.imshow(first_image)
    print("Actual Label: ", class_names[first_label])

    batch_pred = model.predict(images_batch)
    print(class_names[np.argmax(batch_pred[0])])
```

```
First Image
Actual Label:  flip
1/1 [==============================] - 0s 29ms/step
flip
```



Insights

The learning and insights I got from the project was that finding industrial data is really difficult. And the even with little data we could augment the data so that it increases the data size to train the model on. And the model performed really well for that. Maybe making a more complex model could be used. Even transfer learning could have been utilised for this purpose. With main model that could have used is UNet as segments the image which would have been useful in this case.