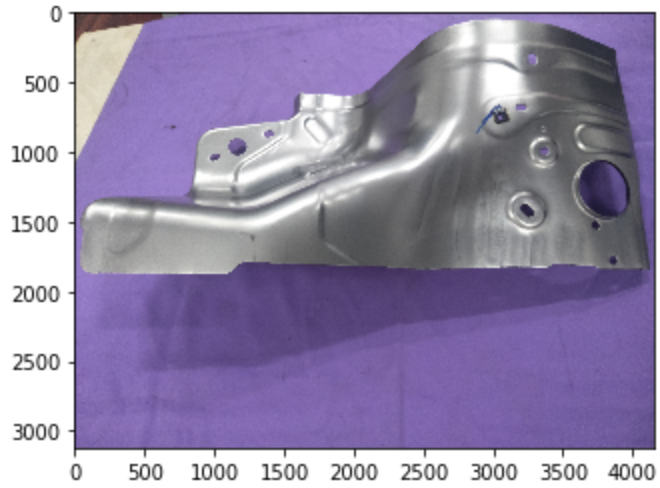
Step1: Import image dataset and EDA

Each dataset is of size (3120*4160*3). Here 3 means Red, green and blue in sequence
plt.imshow(healthy)



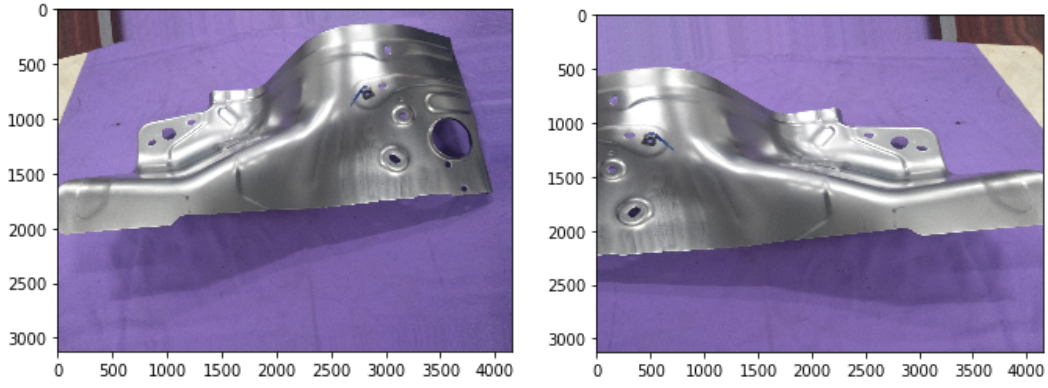
Step2: Preparing the Data for the model

There is too much data for us to read all at once in memory. We can use some built in functions in Keras to automatically process the data, generate a flow of batches from a directory, and also manipulate the images.

Image manipulation

Its usually a good idea to manipulate the images with rotation, resizing, and scaling so the model becomes more robust to different images that our data set doesn't have. We can use the ImageDataGenerator to do this automatically for us.

After image manipulation we will get more data which help in training



They are the same image but (rotated, shifted and flipped) which help in training the model

Step3: Creating the Model

Following parameter used in this process:

A) Filter size of kernel_size=(3,3)with 32 number of filter at the input layer. Generally (3,3) prefered in 2-D convolution layer with 2^n number of filter where n is an integer.
B) Since dataset was taking a very long time, due to limited RAM and processor, i consider strides=(2,2)which help to decrease the number of parameters by half.
C) Max pooling used after Convolution layer of size (2,2) which help to store maximum property of image pixel.
D) Dropout of 0.25 used to avoid overfitting. Means at one point of time it deactivate 25% neurons in one layer.
E) Similarly there are 3 convolution layers used followed by flatten layer.
F) After that two fully connected layer used(one is hidden and the other is output layer)
G) The output layer used **sigmoid** as activation function because it used at output layer for binary classification.
H) All other layer used **ReLu** as activation function because now a days , Binary_crossentropy used as loss function because this problem is binary classification problem.
I) Final optimizer **Adam** used. it used widely which have both concept of momentum as well as RMSProp activation.
J) Total parameter after above processing is 25,655,362.

Training the Model

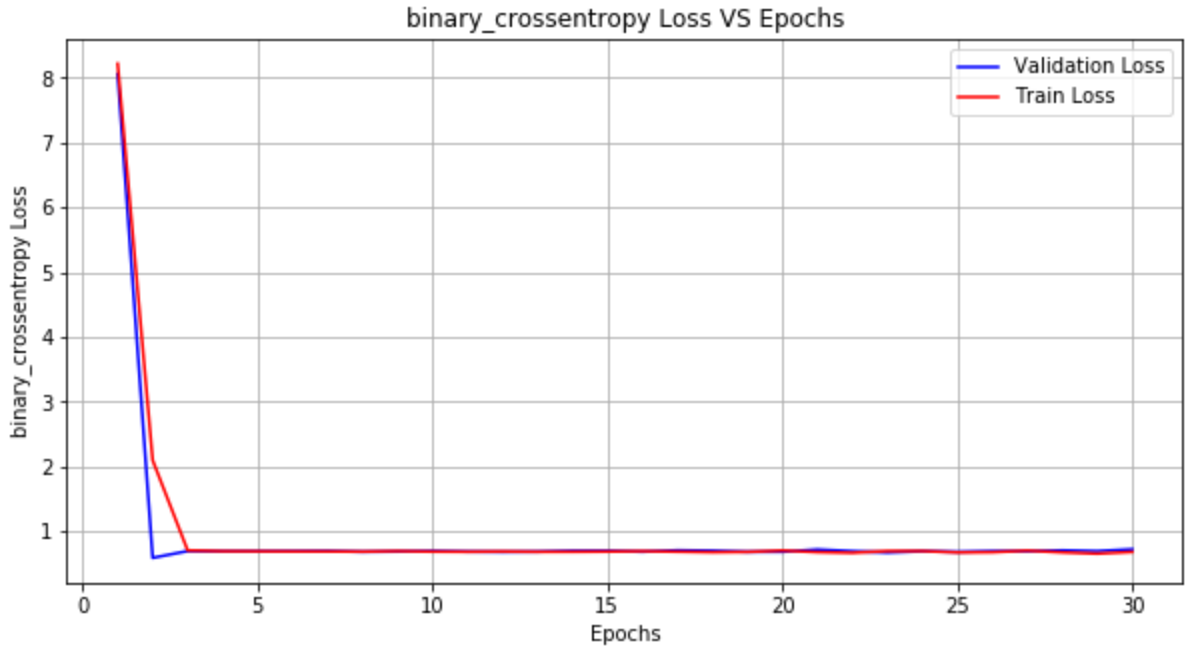
Model training done on 180 images belonging to 2 classes. And test it o 70 images belong to 2 class directly from directory.

Training done on 30 epochs value with batch size of 1 because in 30 epochs the loss value got saturated. And due to my system limitation (processor, RAM), I used only 1 batch size.

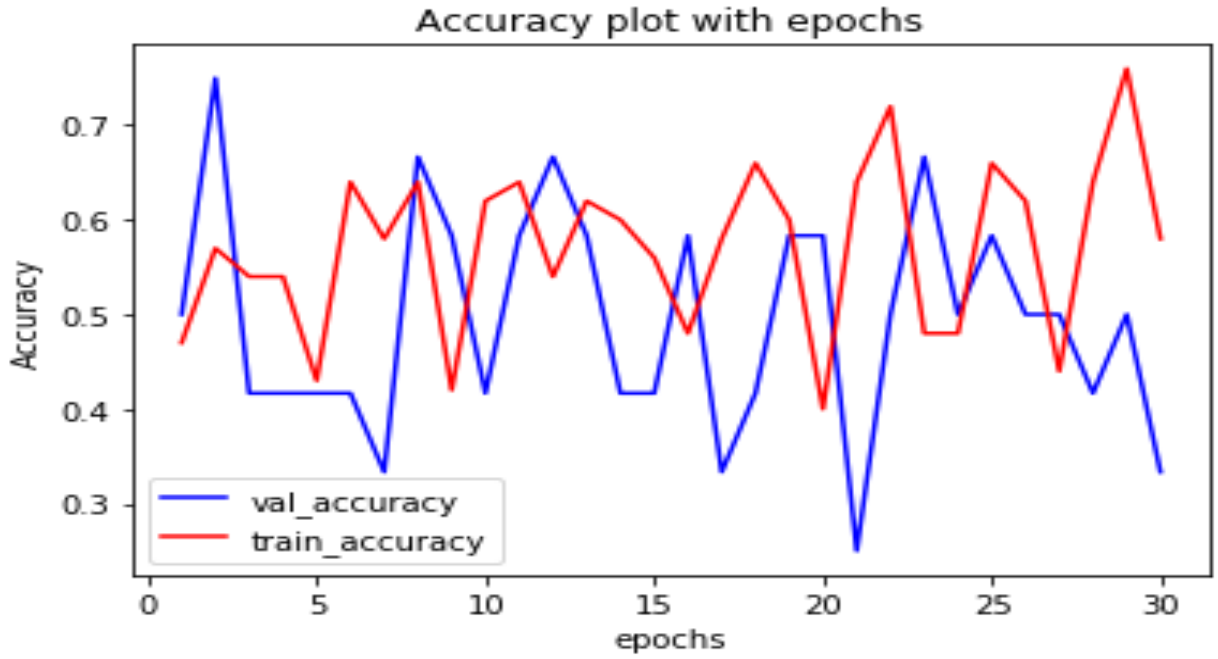
After completing training, model saved to **model1.h5** file.

Result and observation:

Loss vs Epochs plot



Here training loss and test loss value decrease upto 3 epochs than become constant. So at optimal epochs we will get minimum loss value.



Accuracy here vary randomly but bias and variance tradeoff can be seen between 8-13 epochs value and 22-26 epochs values.

Predicting on new images:

```
import numpy as np
from keras.preprocessing import image

file =
'../tmp/YE358311_Fender_apron/test/defect_test/IMG20180905150321.jpg'

img = image.load_img(file, target_size=(3120, 4160, 3))

img = image.img_to_array(img)

img = np.expand_dims(img, axis=0)
img = img/255
```

```
# Output prediction
print(f'Probability that image is a healthy is: {prediction_prob} ')
```

```
Probability that image is a healthy is: [[0.45860156 0.5421296 ]]
```

```
from sklearn.metrics import classification_report
predictions = model1.predict_classes(img)
predictions
```

```
array([1])
```

Here after puting image value from directory, we are getting class value 1. Means the image shows from healthy folders.