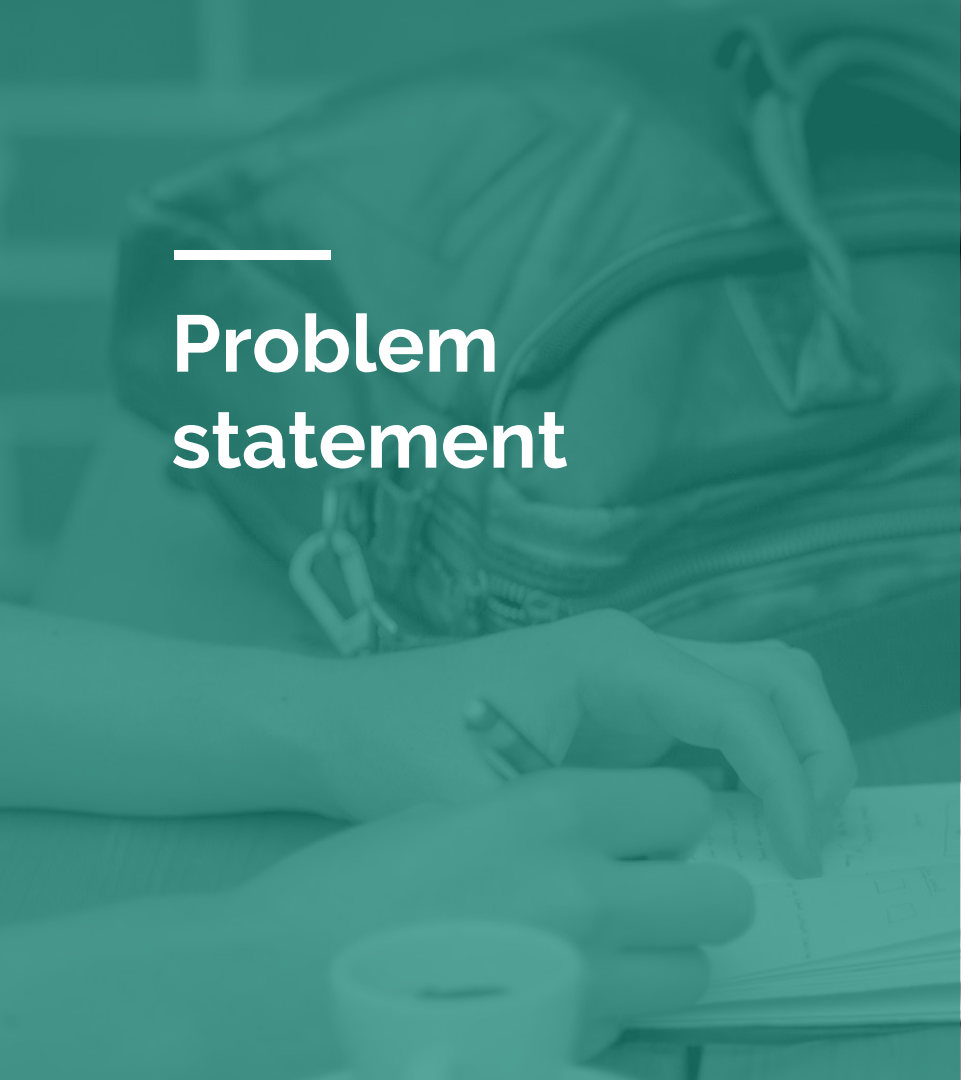
Outline



Problem statement

The mechanical equipment needs to be free from visual defects such as surface defects, nicks, dents, black patches, uncleaned or un-honed surfaces, laser marking missing, character or faint marking etc. This needs to be prevented from being passed on. An automated visual inspection system is required to detect these defects under actual production environment & oily conditions. Robot system with correct algorithms & intelligence built-in(for self-learning) also need to be built in.

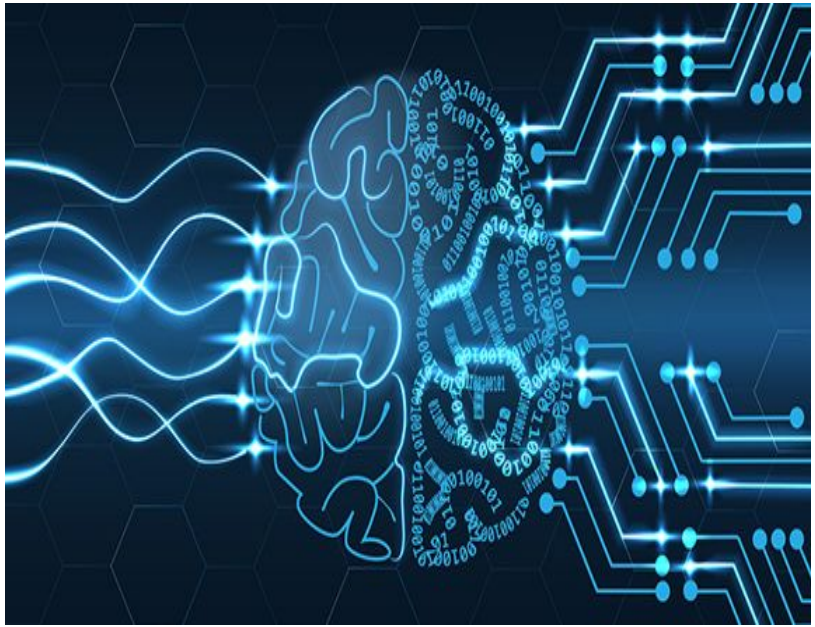
Proposed Solution

Our solution comprises of a deep learning model comprising of multiple Convolutional Neural Networks (CNNs) which will act as an ensemble of predictors. Each CNN will see a different angle of the Bearing from the input images taken from a setup of cameras and make a prediction. The model will -

- preprocess the inputs using Opencv Inpainting Algorithm to interpolate the image defects caused due to oil.
- The images will be evaluated by a CNN model known as **Xception** and a label will be given to the Bearing.
- The output of whether the bearing is defective or not will be displayed to the user by mobile app (Android app).
- The model will use this input to learn from the data.

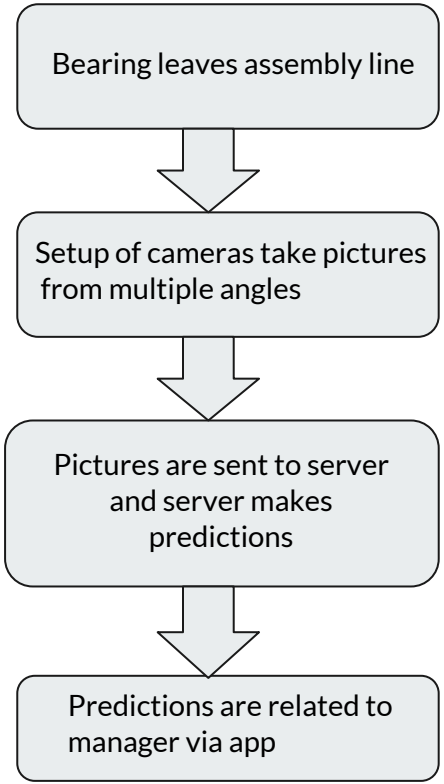
Technology Used

- Programmed in Python
- OpenCV
- Keras
- Xception Model
- Android Studio
- Javascript



Flow of Process

Pictures captured through various angles.



Choices Made to Reach Desired Software Solution

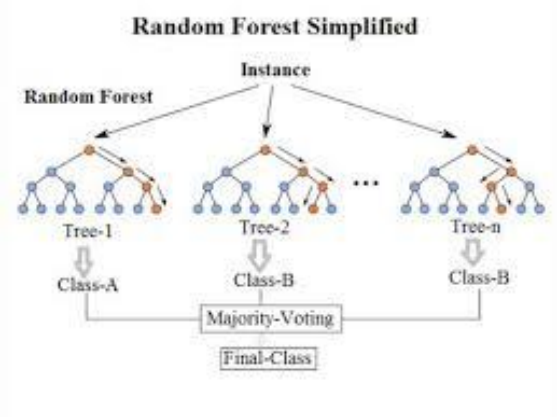
Initially, we had to choose from the following 2 Choices:

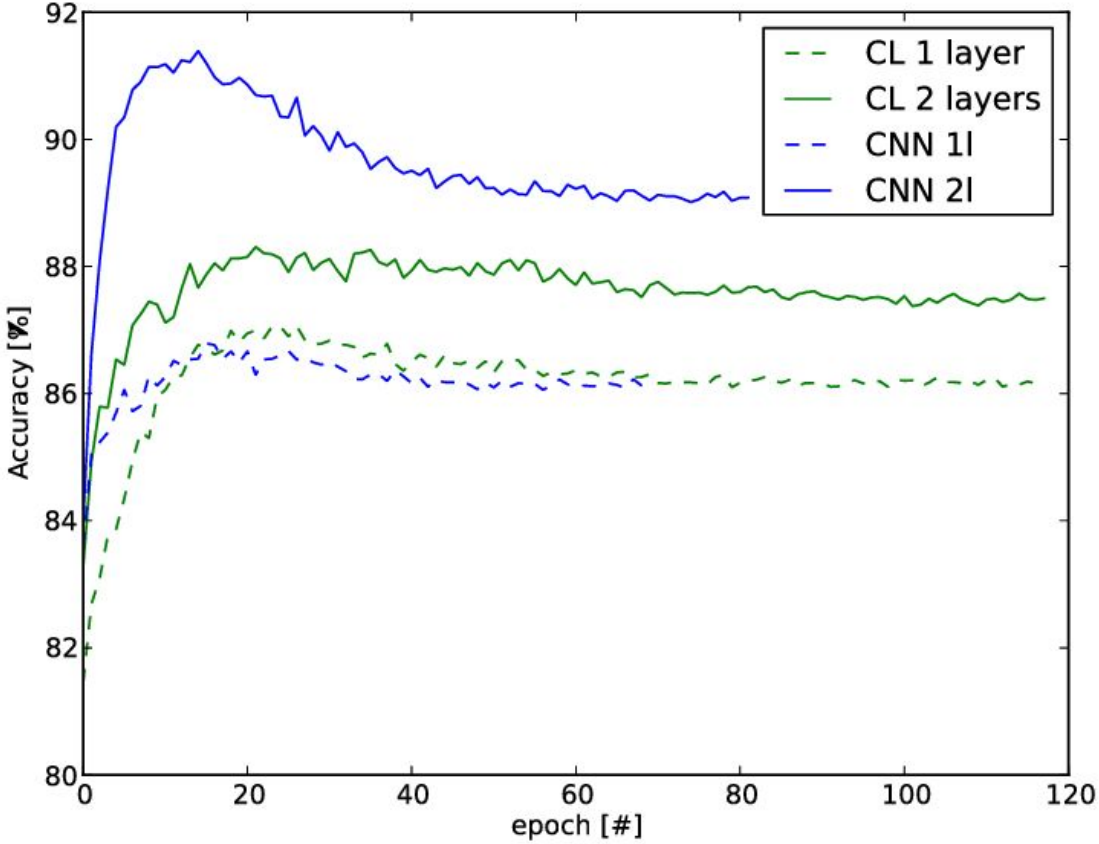
Hardware+Software Solution VS Only Software Solution:

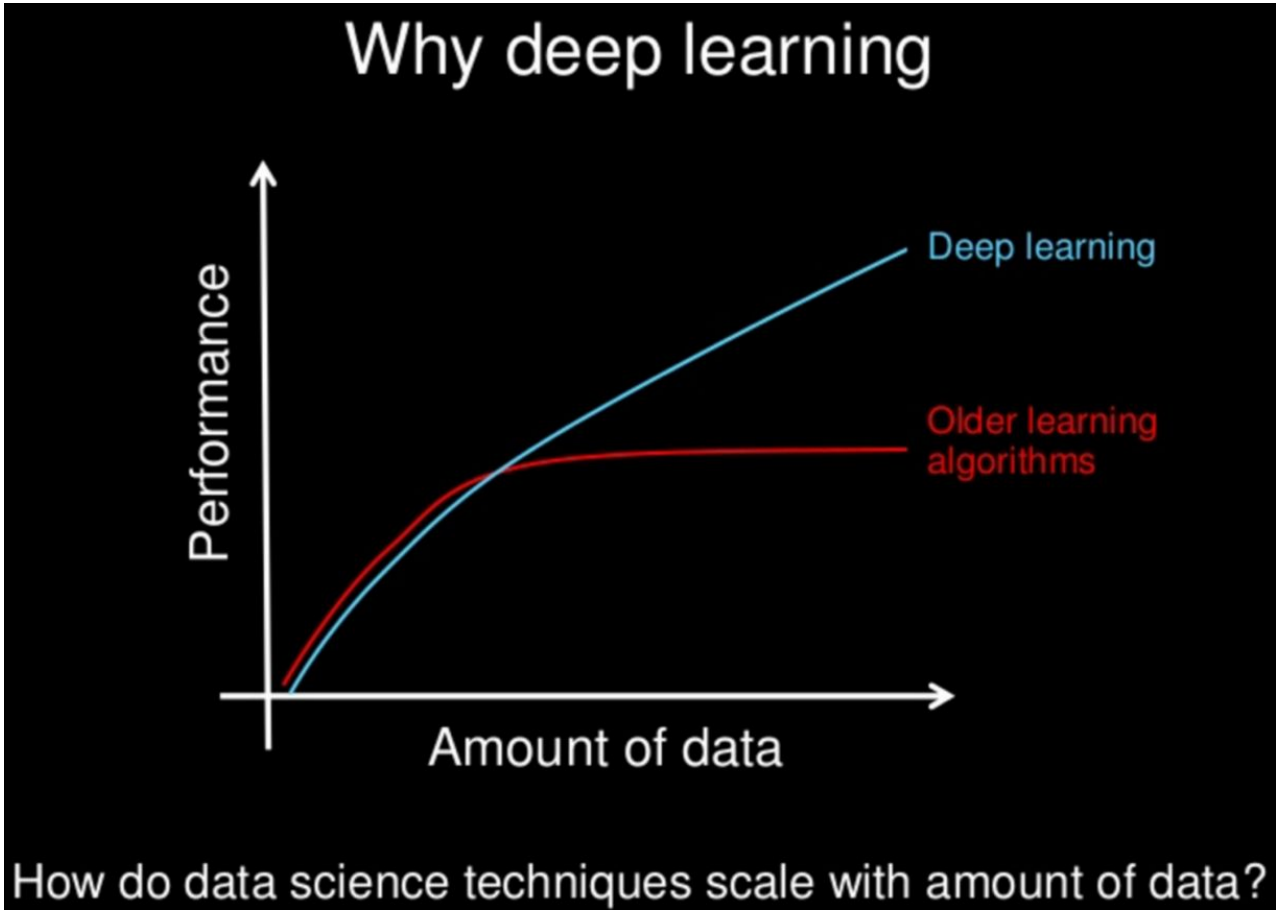
1. Any hardware would need specific ambient conditions to operate.
2. Any hardware would need integration into existing hardware
3. Hardware creates maintenance and installation costs.
4. Different hardware design needed for different components whereas a single software component works for all components without creating any additional costs.

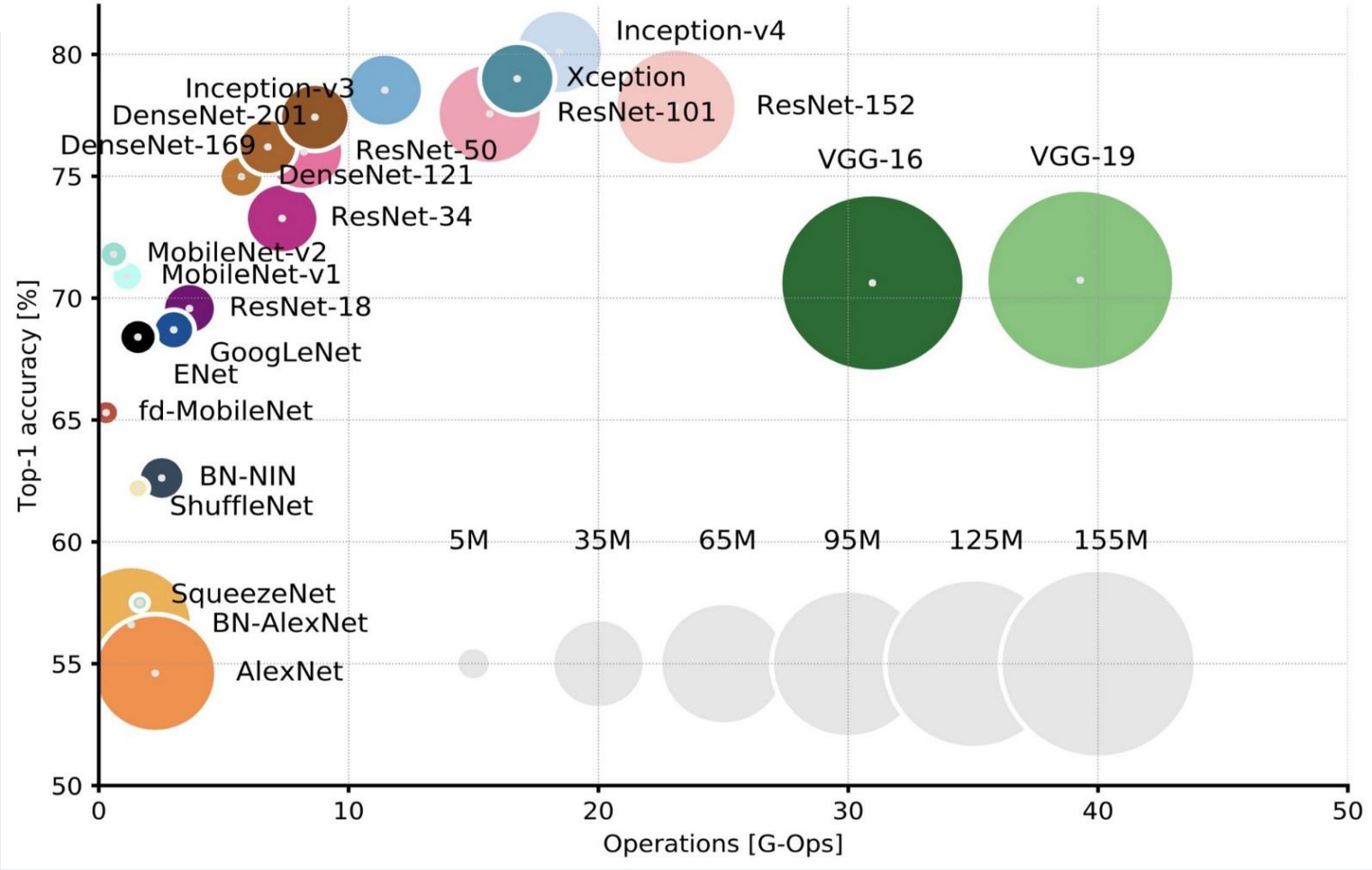
Choices Made to Reach Desired Software Solution

2. Support Vector Machines - The issue with this solution was that performance stagnates after a certain point.
3. Random Forest - Makes multiple decision trees which vote on the classification . Can't learn incrementally for new information. If we have to add new data to dataset, then we need to train it from scratch.




92
90
88
86
84
82
80
Accuracy [%]
0
20
40
60
80
100
120
epoch [#]
CL 1 layer
CL 2 layers
CNN 1l
CNN 2l


Why deep learning
Performance
Deep learning
Older learning algorithms
Amount of data
How do data science techniques scale with amount of data?


Inception-v4
Inception-v3
Xception
DenseNet-201
ResNet-101
ResNet-152
DenseNet-169
ResNet-50
VGG-16
VGG-19
DenseNet-121
ResNet-34
MobileNet-v2
MobileNet-v1
ResNet-18
GoogLeNet
ENet
fd-MobileNet
BN-NIN
ShuffleNet
5M
35M
65M
95M
125M
155M
SqueezeNet
BN-AlexNet
AlexNet
Top-1 accuracy [%]
Operations [G-Ops]
80
75
70
65
60
55
50
0
10
20
30
40
50

Model / Characteristics	Alex Net	VGG-16	ResNet-18	Xception	Inception-V4
Layers	8	16	18	36	**96**
Evaluation Time	2	3	4	**1**	5
Model size	62M	125M	**10M**	15M	35M
Performance/ Accuracy	54%	72%	69%	79%	**81%**
Complexity	**Low**	**Low**	Medium	High	Very High
Parallel Stages	1	1	2	**3-7**	**3-7**
Total Score =	17	20	18	**14**	20
Final Rank =	2	5	3	**1**	4

Camera Specifications and Threshold Values

We can select the camera specification according to the amount of clarity we want

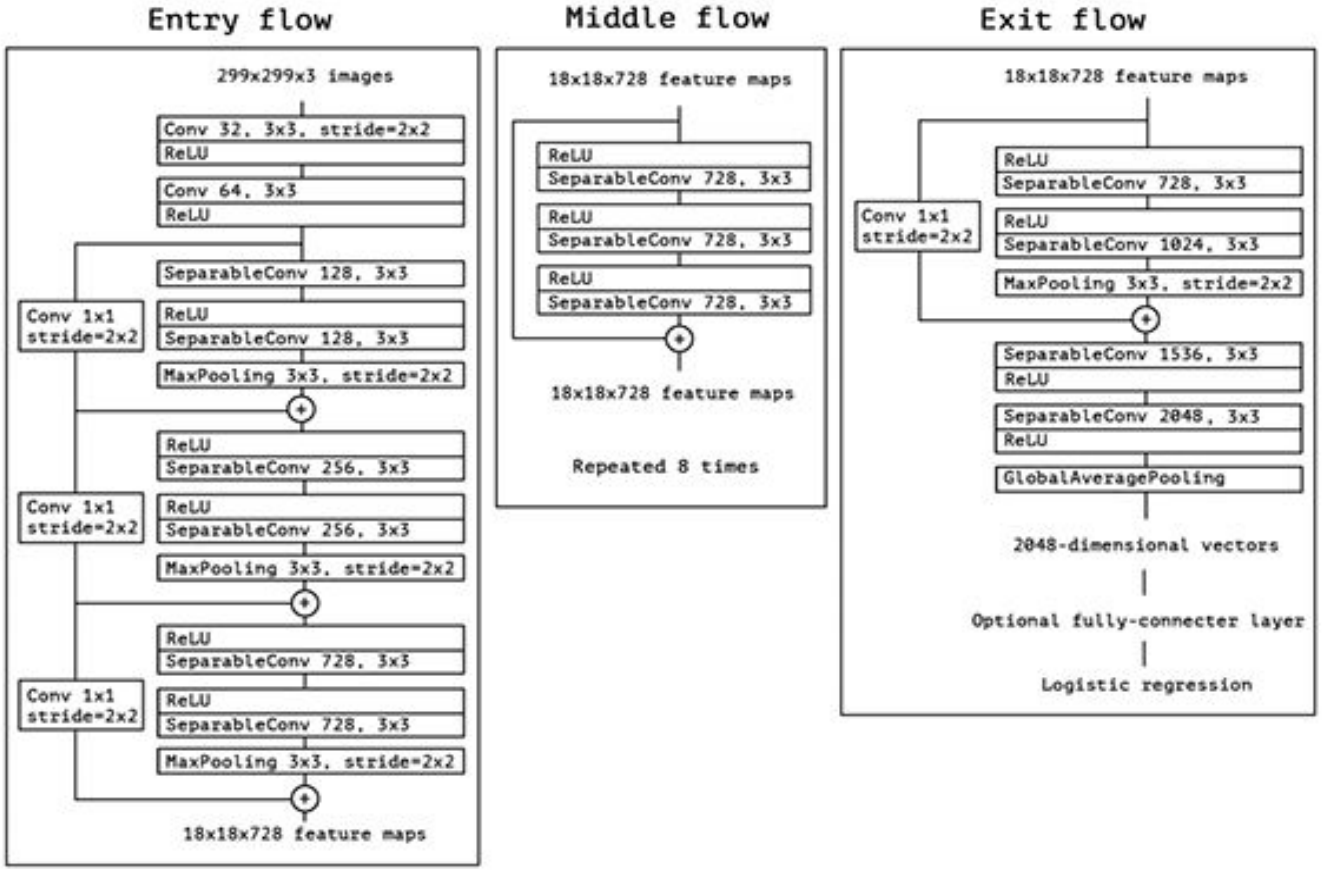
- For a clarity level of 5 pixels/micron= 1.27 Mp camera
- For a clarity level of 10 pixels/micron= 2.54 Mp camera
- For a clarity level of 20 pixels/micron= 5.08 Mp camera

For a 100 micron defect in our ball bearing, if we choose a 2.54 MP camera, the defect in the image will be represented by 1000 pixels.

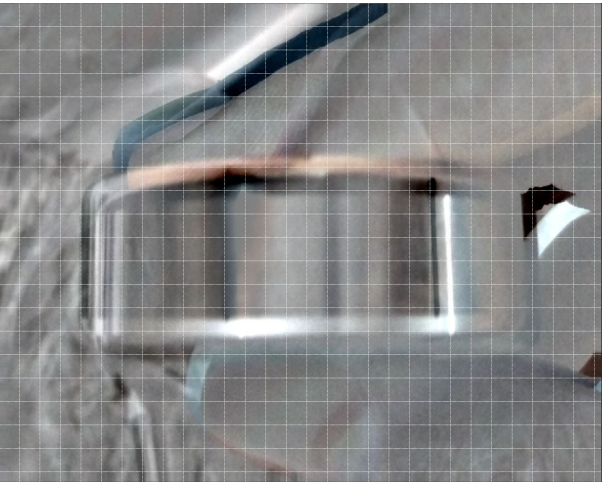
Therefore, we recommend using a camera of at least 5 MP so that even a 100 micron defect is represented clearly in the image.

Model Components

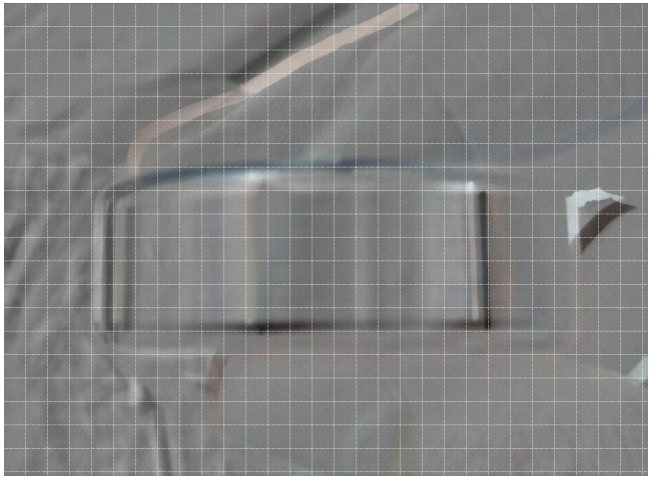
Xception CNN Model


Entry flow
299x299x3 images
Conv 32, 3x3, stride=2x2
ReLU
Conv 64, 3x3
ReLU
SeparableConv 128, 3x3
Conv 1x1 stride=2x2
ReLU
SeparableConv 128, 3x3
MaxPooling 3x3, stride=2x2
ReLU
SeparableConv 256, 3x3
Conv 1x1 stride=2x2
ReLU
SeparableConv 256, 3x3
MaxPooling 3x3, stride=2x2
ReLU
SeparableConv 728, 3x3
Conv 1x1 stride=2x2
ReLU
SeparableConv 728, 3x3
MaxPooling 3x3, stride=2x2
18x18x728 feature maps
Middle flow
18x18x728 feature maps
ReLU
SeparableConv 728, 3x3
ReLU
SeparableConv 728, 3x3
ReLU
SeparableConv 728, 3x3
18x18x728 feature maps
Repeated 8 times
Exit flow
18x18x728 feature maps
Conv 1x1 stride=2x2
ReLU
SeparableConv 728, 3x3
ReLU
SeparableConv 1024, 3x3
MaxPooling 3x3, stride=2x2
SeparableConv 1536, 3x3
ReLU
SeparableConv 2048, 3x3
ReLU
GlobalAveragePooling
2048-dimensional vectors
Optional fully-connecter layer
Logistic regression

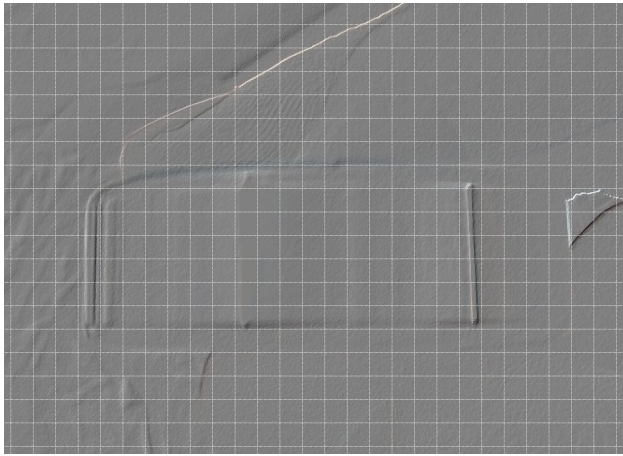
Evolution of image through the Xception Model



Initial Stage



Intermediate Stage



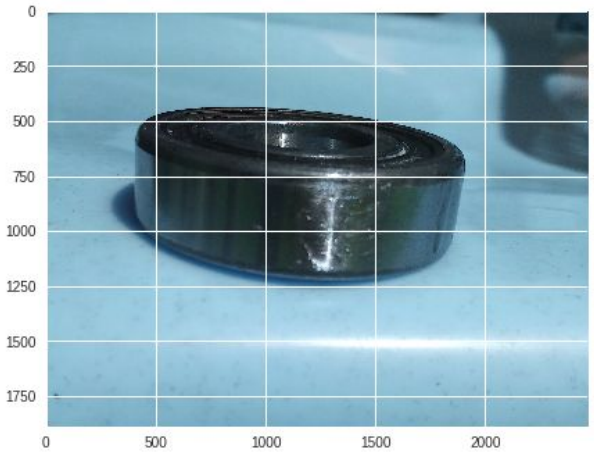
Final Stage

Effect of Pre processing on defective bearings.

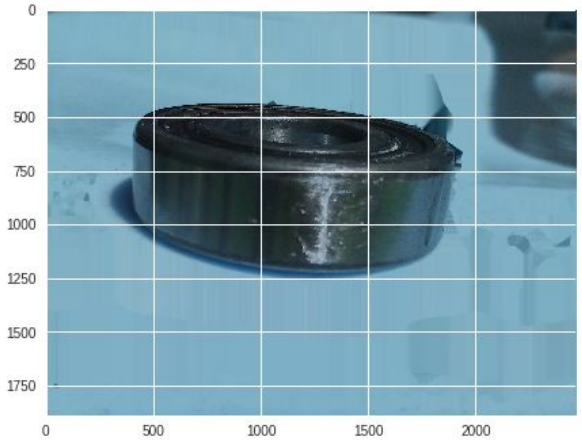
The glare due to the oil or stensol is removed but we can still see the defects below the glare.

This is implemented using OpenCv's inpainting algorithm.

Given Image

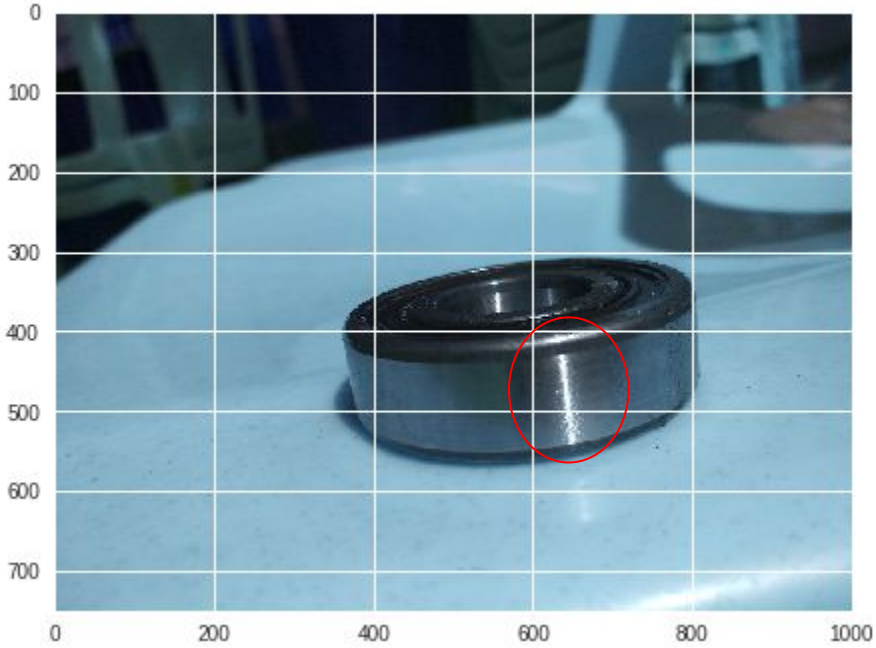


Before Glare removal

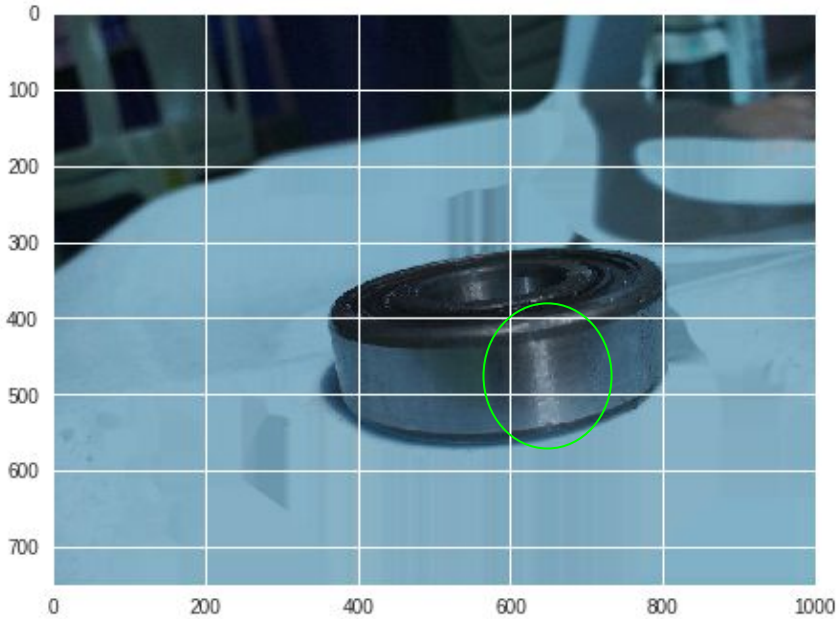


After Glare removal

Effect of Preprocessing on non-defective bearings

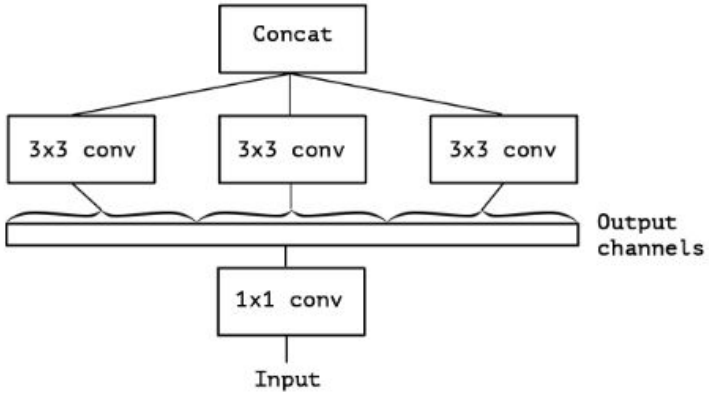


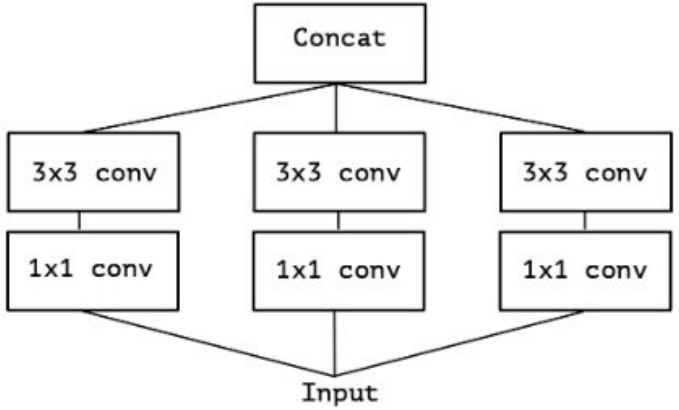
Before

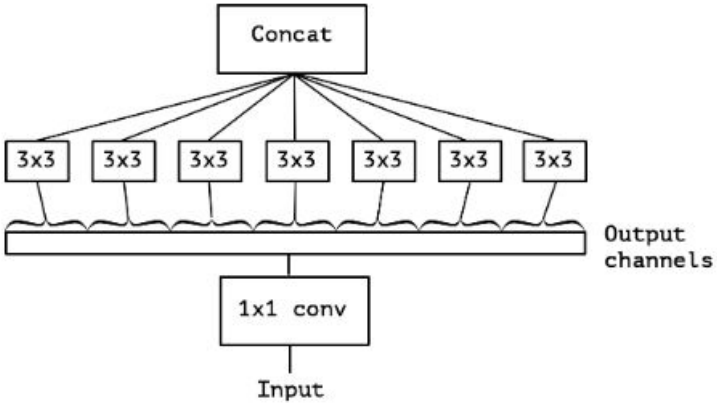


After

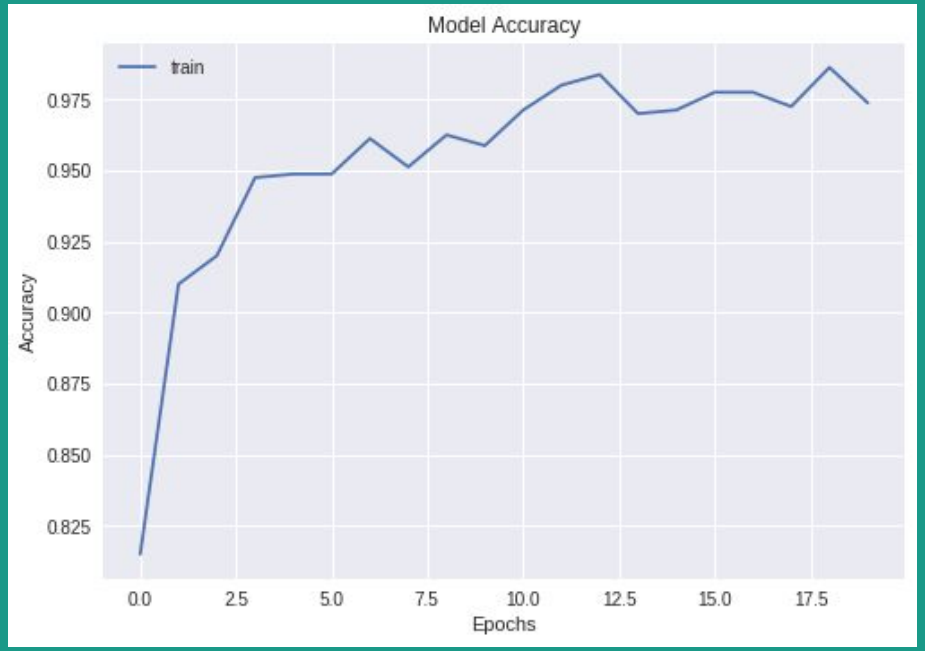
The research of Xception and its different models.

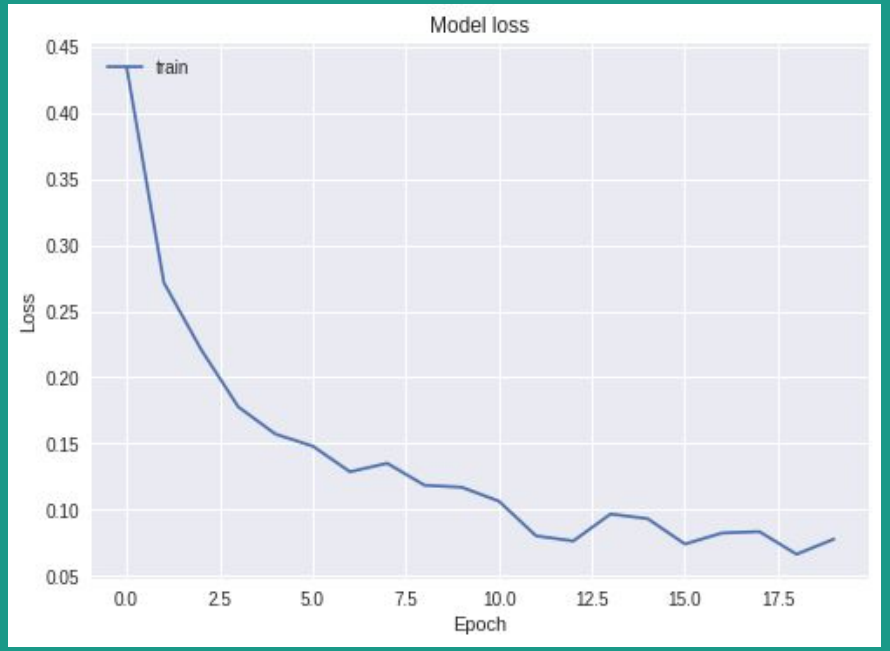






Training the system.





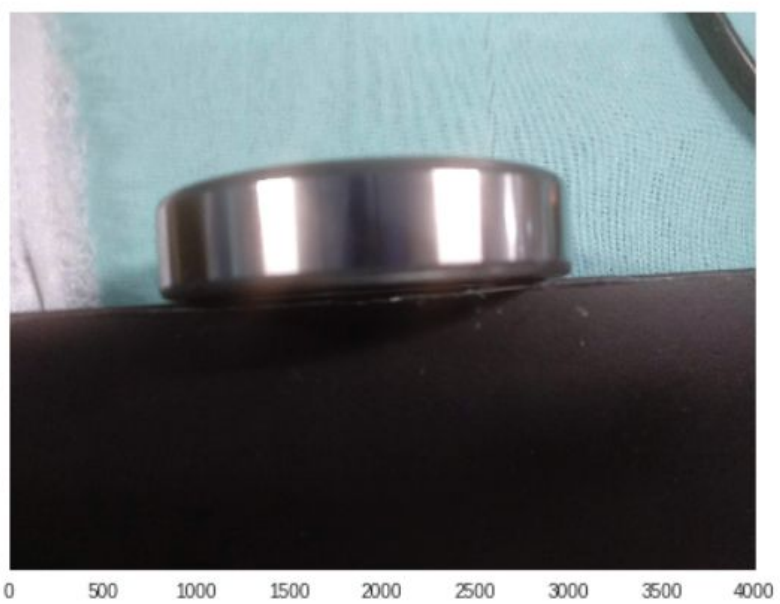
After processing and training the examples, we give various input images like:

1. Non-oily, Non-Defective bearing
2. Non-oily, Defective bearing
3. Oily, Non-Defective bearing
4. Oily, Defective bearing

Non-oily , Non-defective bearing

```
[188] #TNO
      test_image_proc('none.jpg')
```

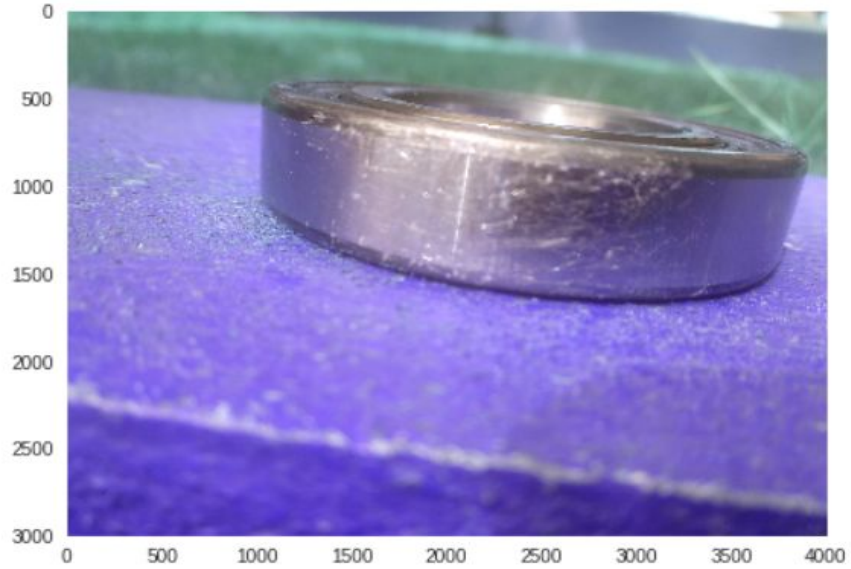
```
Generated img-pre.jpg
Result for Image:
Non-Defective Bearing
```



Non-oily Defective bearing

```
[189] #FNO
      test_image_proc('def.jpg',  false_pos = False)
```

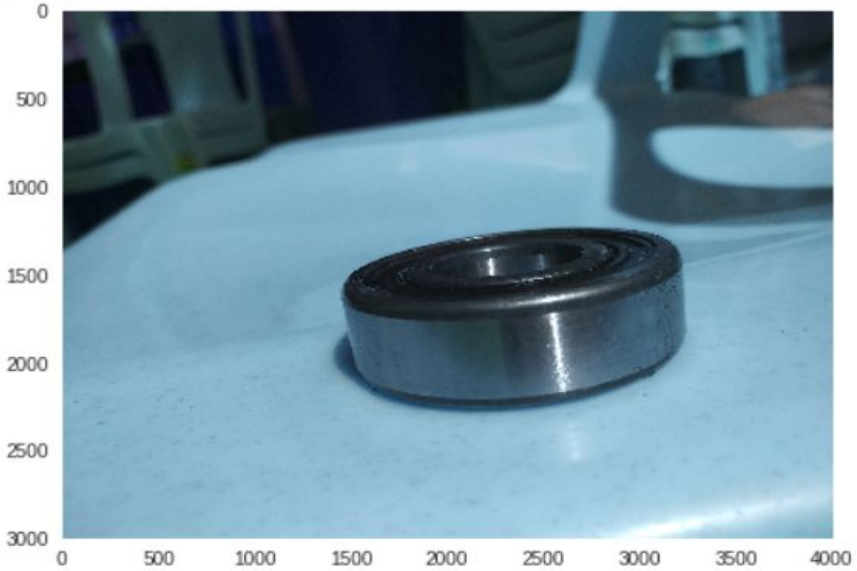
```
Generated img-pre.jpg
Result for Image:
Defective bearing
```



Oily Non-defective bearing

```
[193] #TO
      test_image_proc('Oil-NonDefect.jpg')
```

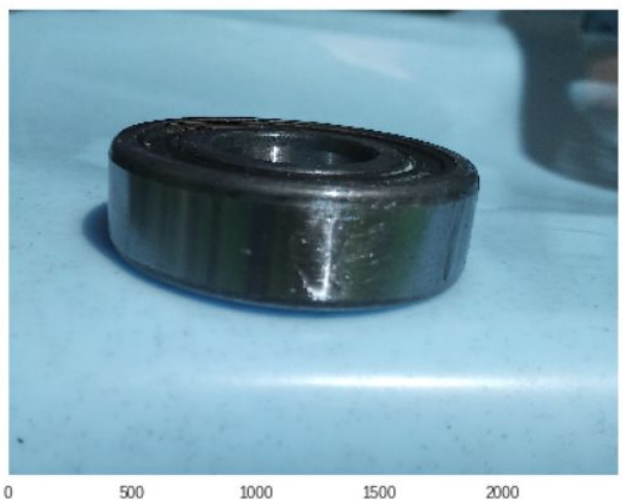
```
Generated img-pre.jpg
Result for Image:
Non-Defective Bearing
```



Oily Defective bearing

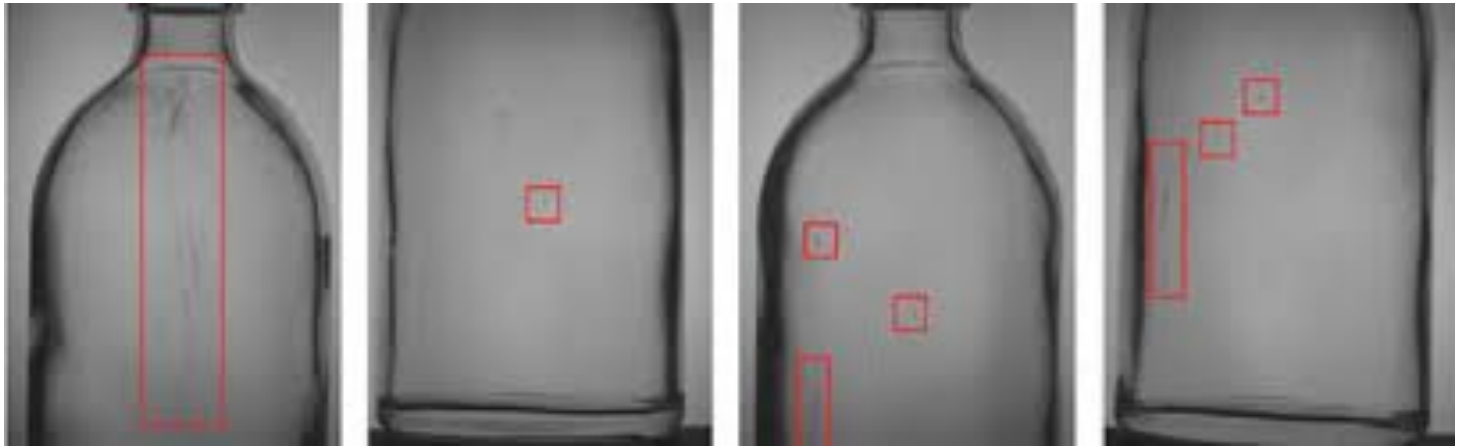
```
#FO
test_image_proc('Oil-Defect.jpg',  false_pos = False)
```

```
Generated img-pre.jpg
Result for Image:
Defective bearing
```



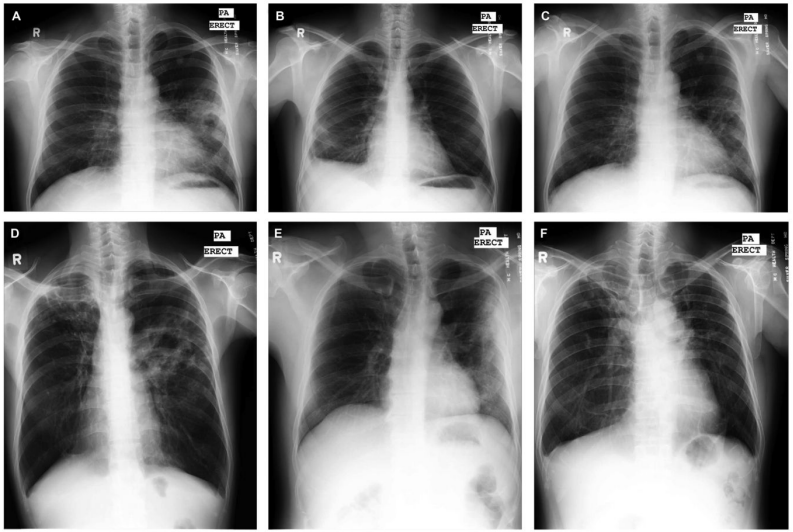
Multiple usage of this software.

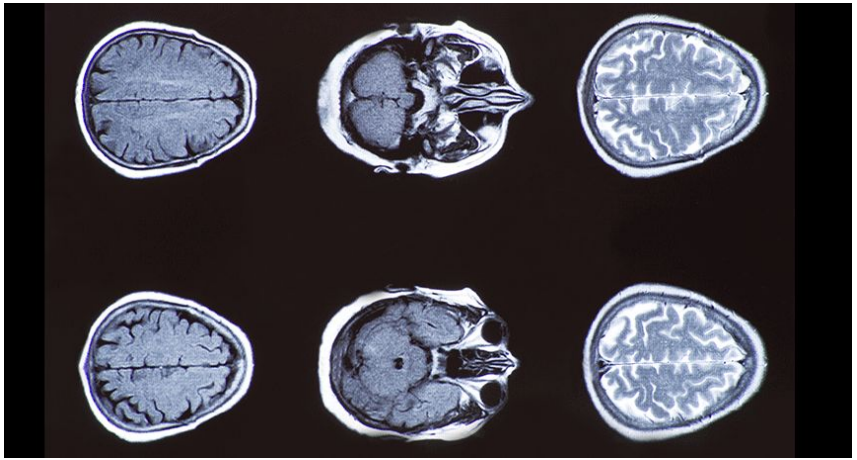
- This model can be used to classify and find errors in any equipment. The only hurdle will be to train it on a dataset of the required equipment.
- For example, it can be used to check if a glass equipment has defect or not.



Usage in medical field:

- This model can be used to improve the test results of X-rays, CT-scans, MRI etc. as the accuracy increases with the size of dataset which can be easily provided by the hospitals.
- This model can be trained on medical dataset by giving the dataset of healthy person and unhealthy patients.





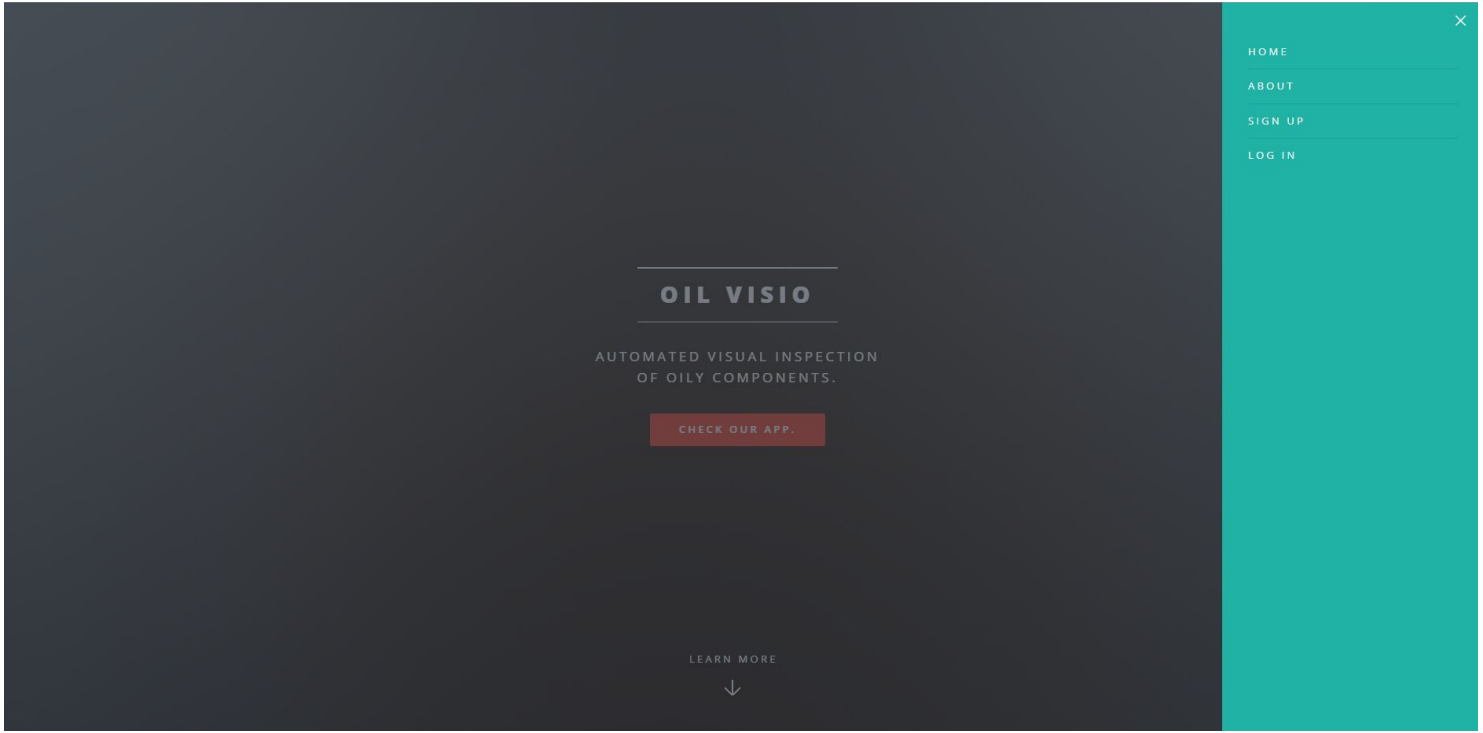
Future Prospects

- ➔ As we increase the test data, the accuracy of the system will increase.
- ➔ On taking samples from different angles the results will be more accurate.
- ➔ This system is implemented in such a way that it improves on its own on daily basis.
- ➔ The manager will be able to check the real time accuracy of the model via Mobile and Web App.

USER INTERFACE

Web Application
Mobile Application

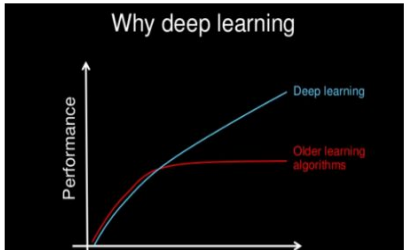
Web Application



What is Oil Visio?

A soultion to detect defective oily components by **You Know Who.** Implemented using Deep Learning, that is a subfield of machine learning concerned with algorithms inspired by the structure and function of the brain called artificial neural networks.

Why Deep Learning?



The researchers found that "almost all the value today of deep learning is through supervised learning or learning from labeled data". This is the concept of our

GET IN TOUCH WITH YOU KNOW WHO.

We are happy to design the solution to every problem that your business deals with.

Name

Email

Message

SEND MESSAGE

Netaji Subhas University of Technology

+91-7888763776

Youknowwho@gmail.com

Mobile Application


11:32
Parse Starter Project
Assembly Line 1
35
Defected
65
Non Defected
Assembly Line 2


11:32
Parse Starter Project
Daily Stats
Description
Monthly Stats

Thankyou.