Basic Quality inspection using image processing

Arunkumar B

Balachandra P Shetty

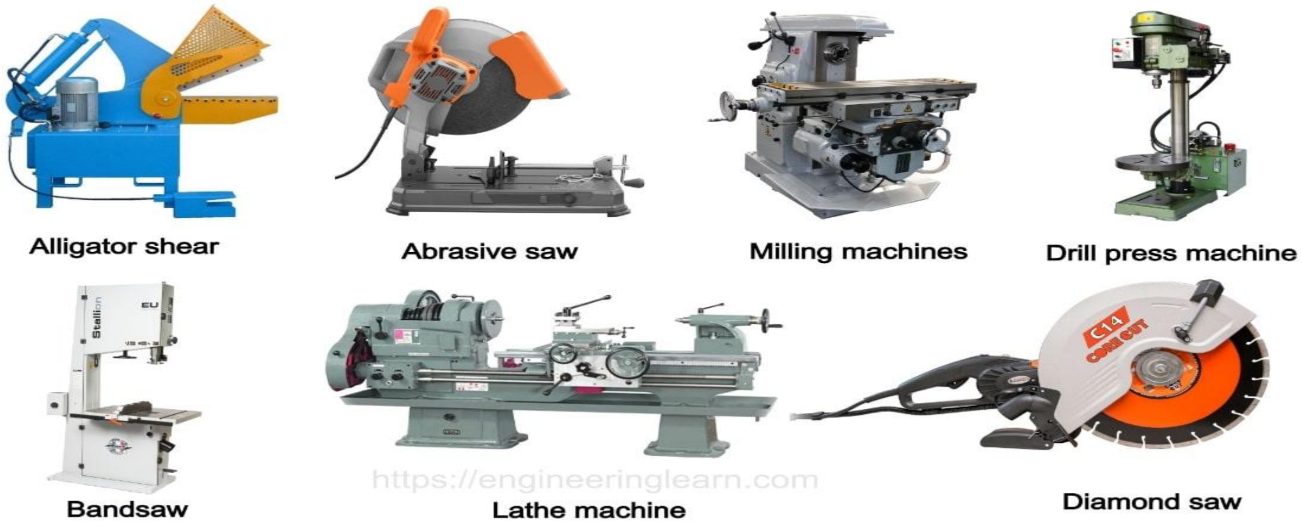
11/30/2022

OBJECTIVE

- Build a Quality system which is capable of doing some of the quality inspection using image process techniques which are currently done manually by human

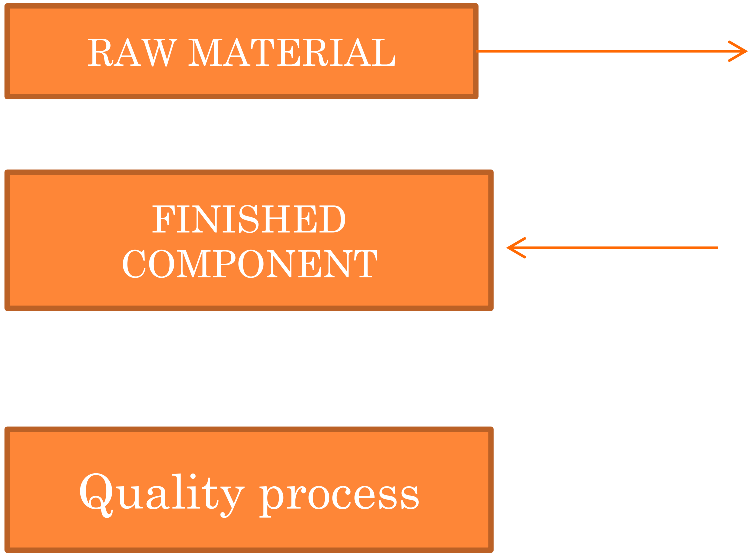
MACHINE TOOL

- A machine tool is a machine for machining metal or other rigid materials, usually by cutting, boring, grinding, shearing, or other forms of deformations.



MACHINING PROCESS

- Machining is process where metal will be removed from the input material to get desired shape



RAW MATERIAL

FINISHED COMPONENT

Quality process

QUALITY PROCESS

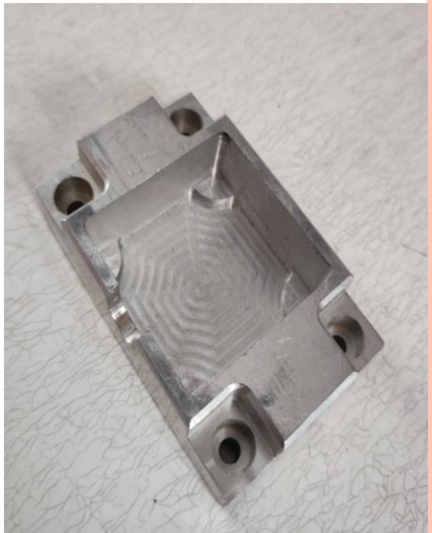
- Once machining process complete next is Quality process
- Quality ensure the correct finished product reaches the customer
- If Quality fails component needs to be send for for rework
    - Examining
    - Measuring
    - Testing

VISUAL INSPECTION

- Identify if all machining operation competed
- Checking for any physical damage to the component
- Checking for improper machining
- Checking for any corrosion /rust
- Checking for cutter mark

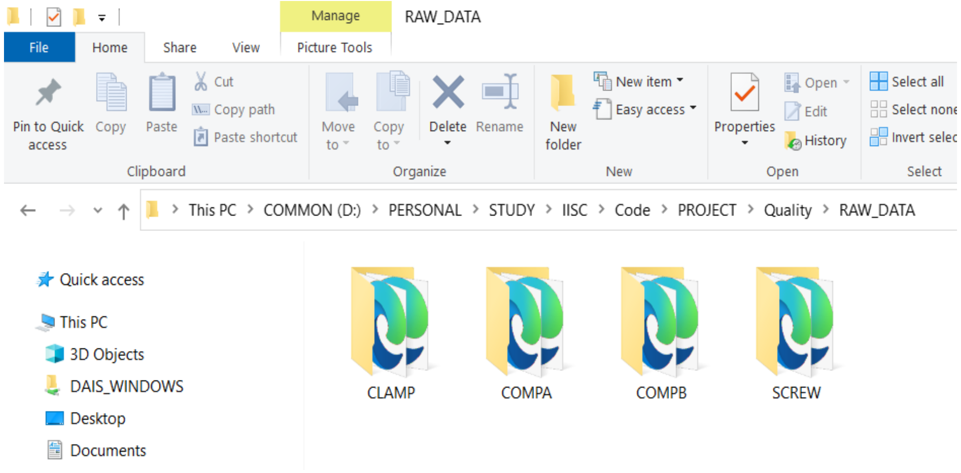
Why AI based visual inspection

https://www.ibm.com/in-en/topics/visual-inspection

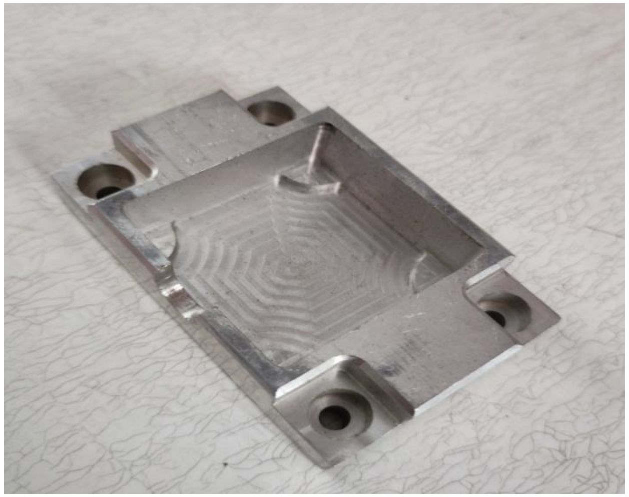


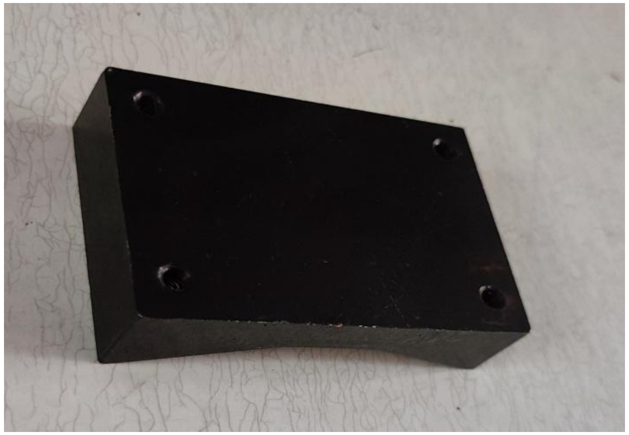
DATA COLLECTION

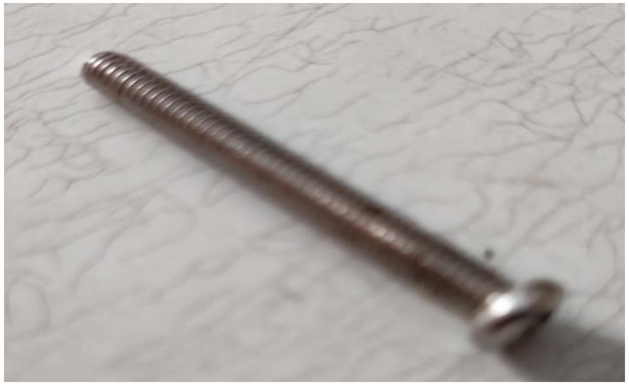
- USB based camera is connected to Laptop which can capture
- Python code is made which request user to select location, number of images to acquire and select name for each class
- Based on the input data will be saved in the location



DATA COLLECTION

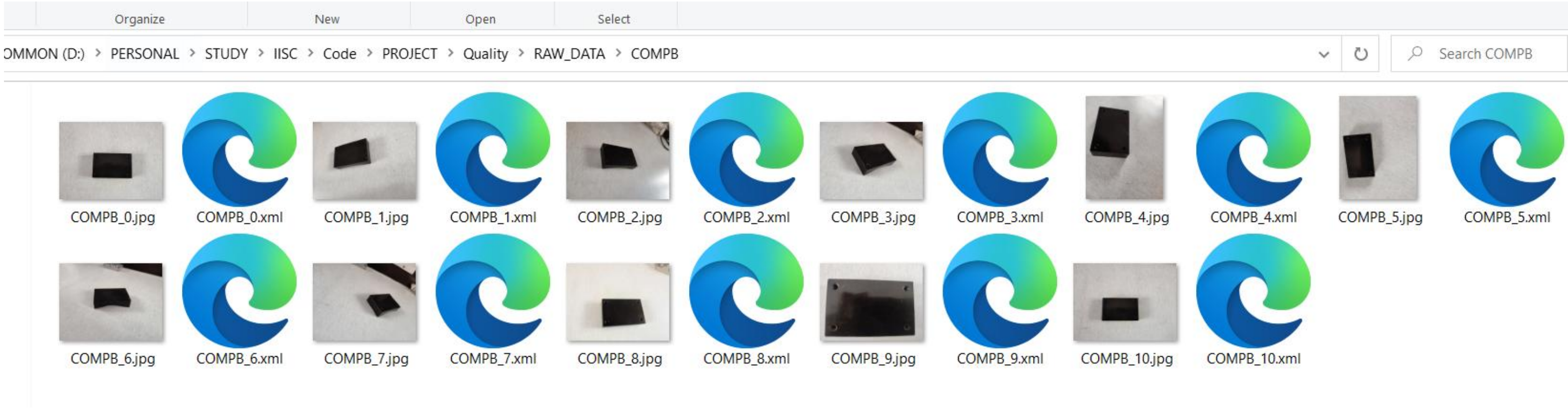






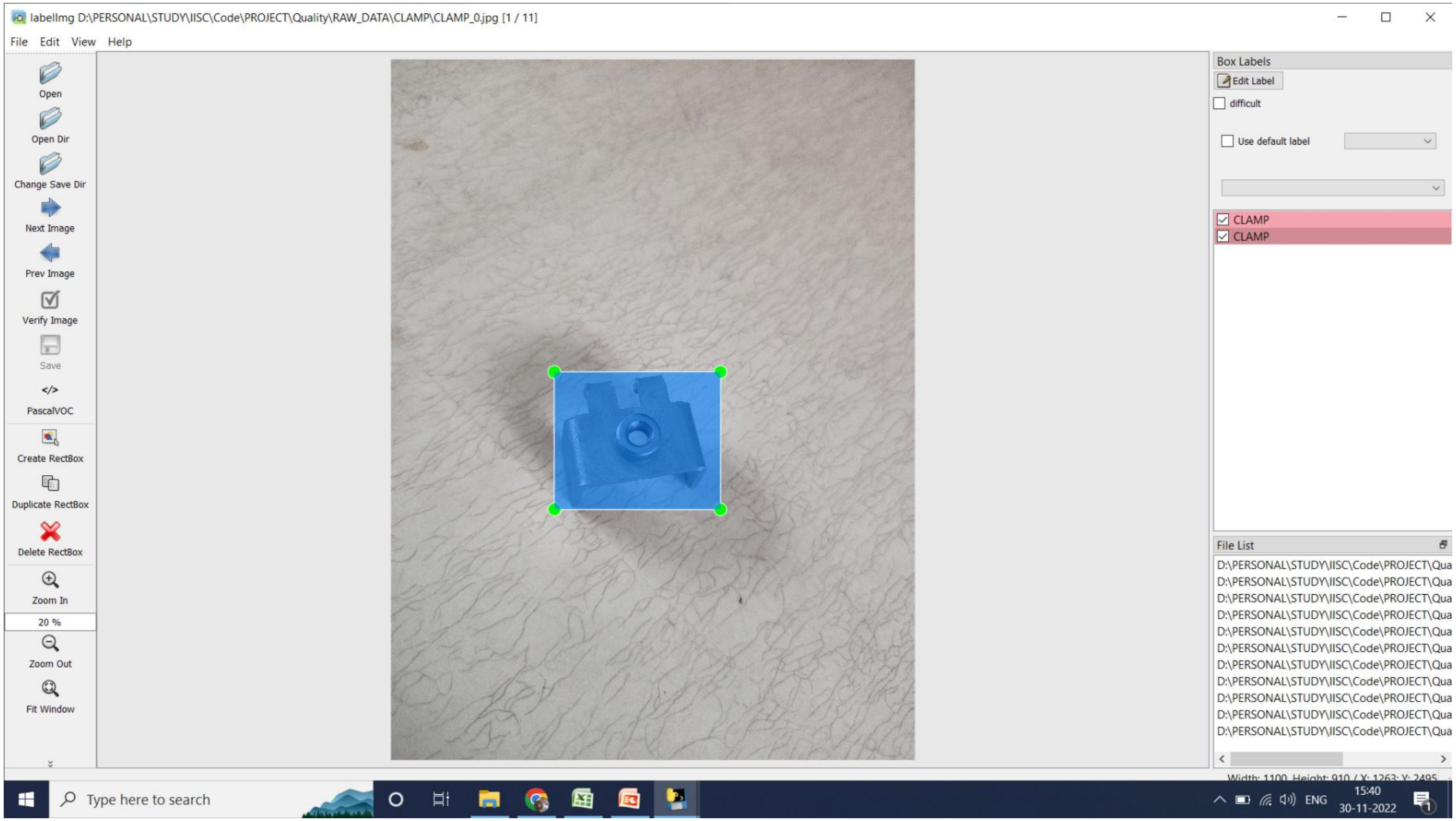
LABELLING

- Downloaded open source labelling tool "LABELIMG"
- Compiled and used for Labelling all the images
- Its created xml which have selected box in it



11/30/2022

LABELLING



labelImg D:\PERSONAL\STUDY\IISC\Code\PROJECT\Quality\RAW_DATA\CLAMP\CLAMP_0.jpg [1 / 11]

□ difficult

□ Use default label

☑ CLAMP
☑ CLAMP

File List

D:\PERSONAL\STUDY\IISC\Code\PROJECT\Qua
D:\PERSONAL\STUDY\IISC\Code\PROJECT\Qua
D:\PERSONAL\STUDY\IISC\Code\PROJECT\Qua
D:\PERSONAL\STUDY\IISC\Code\PROJECT\Qua
D:\PERSONAL\STUDY\IISC\Code\PROJECT\Qua
D:\PERSONAL\STUDY\IISC\Code\PROJECT\Qua
D:\PERSONAL\STUDY\IISC\Code\PROJECT\Qua
D:\PERSONAL\STUDY\IISC\Code\PROJECT\Qua
D:\PERSONAL\STUDY\IISC\Code\PROJECT\Qua
D:\PERSONAL\STUDY\IISC\Code\PROJECT\Qua
D:\PERSONAL\STUDY\IISC\Code\PROJECT\Qua

Width: 1100, Height: 910 / X: 1263; Y: 2495

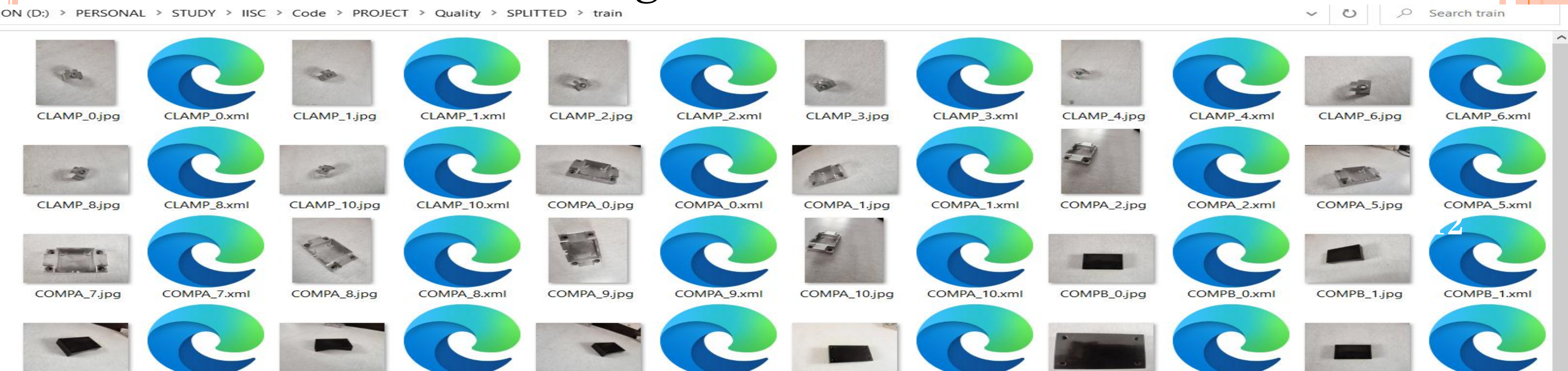
ENG 15:40 30-11-2022

LABELLING

- Totally 11 images collected for each segment
- Named under four classes
  - COMPA
  - COMPB
  - CLAMP
  - SCREW
- Created label file also with same name which will be used in further work
- labels = [{'name':'CLAMP', 'id':1}, {'name':'COMPA', 'id':2}, {'name':'COMPB', 'id':3}, {'name':'SCREW', 'id':4}]

SPLITTING DATA

- Made a small python code which ask for location where data will be available, Number of data required for training and number of data required for testing
- Based on the data it will randomly split data for testing and training
- In this case I have selected 8 data fro training and 3 data for testing



ON (D:) > PERSONAL > STUDY > IISC > Code > PROJECT > Quality > SPLITTED > train

Search train

CLAMP_0.jpg CLAMP_0.xml CLAMP_1.jpg CLAMP_1.xml CLAMP_2.jpg CLAMP_2.xml CLAMP_3.jpg CLAMP_3.xml CLAMP_4.jpg CLAMP_4.xml CLAMP_6.jpg CLAMP_6.xml

CLAMP_8.jpg CLAMP_8.xml CLAMP_10.jpg CLAMP_10.xml COMPA_0.jpg COMPA_0.xml COMPA_1.jpg COMPA_1.xml COMPA_2.jpg COMPA_2.xml COMPA_5.jpg COMPA_5.xml

COMPA_7.jpg COMPA_7.xml COMPA_8.jpg COMPA_8.xml COMPA_9.jpg COMPA_9.xml COMPA_10.jpg COMPA_10.xml COMPB_0.jpg COMPB_0.xml COMPB_1.jpg COMPB_1.xml

ENVIRONMENT

- Following library/models is installed
  - Tensor flow
  - Object detection
  - Xml
  - Matplotlib

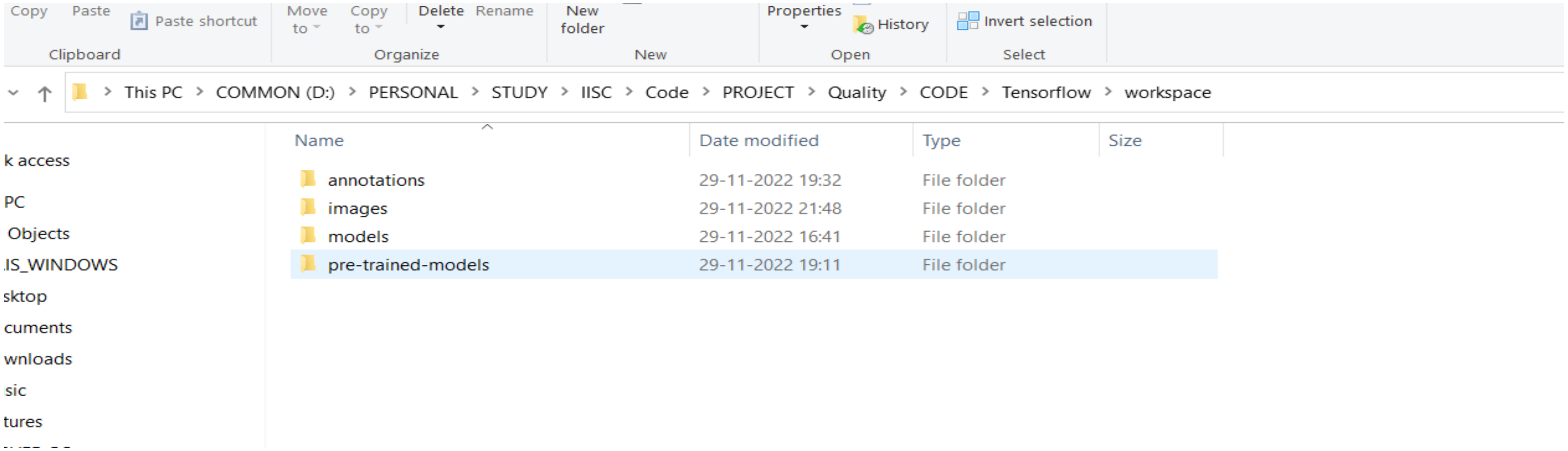
  (Procedure for installation is added in python script)
- SSD MobileNet V2 FPNLite model is selected for this application
- There will be model_builder_tf2_test.py which will help to check all the needed dependency is available or not

MODEL SELECTION

- Got Numerous number of Model form https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/tf2_detection_zoo.md
- I have Selected SSD MobileNet V2
- SSD Stand for Single shot detection, Single image is required to detect image
- This model is very lite which can be deployed even in the low compute devices

TRAINING MODEL

- Training image location, Label map file and converted TF record file is provided as input for training
- Pipe line config file is edited for all the configuration and path related

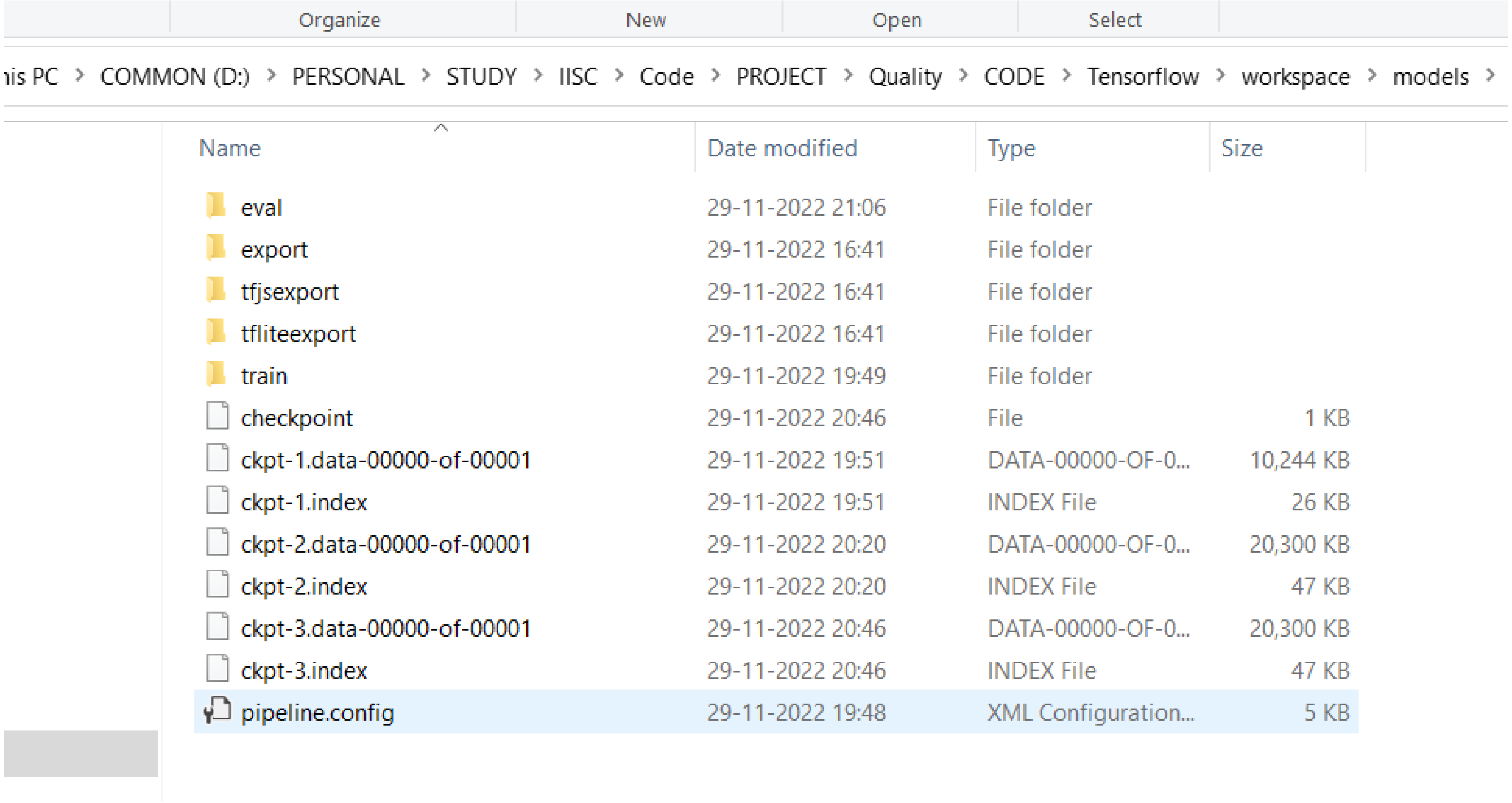


This PC > COMMON (D:) > PERSONAL > STUDY > IISC > Code > PROJECT > Quality > CODE > Tensorflow > workspace

Name	Date modified	Type	Size
annotations	29-11-2022 19:32	File folder	
images	29-11-2022 21:48	File folder	
models	29-11-2022 16:41	File folder	
pre-trained-models	29-11-2022 19:11	File folder	

TRAINING MODEL

- Total time of 55 min taken for training
- Training has been done using only CPU

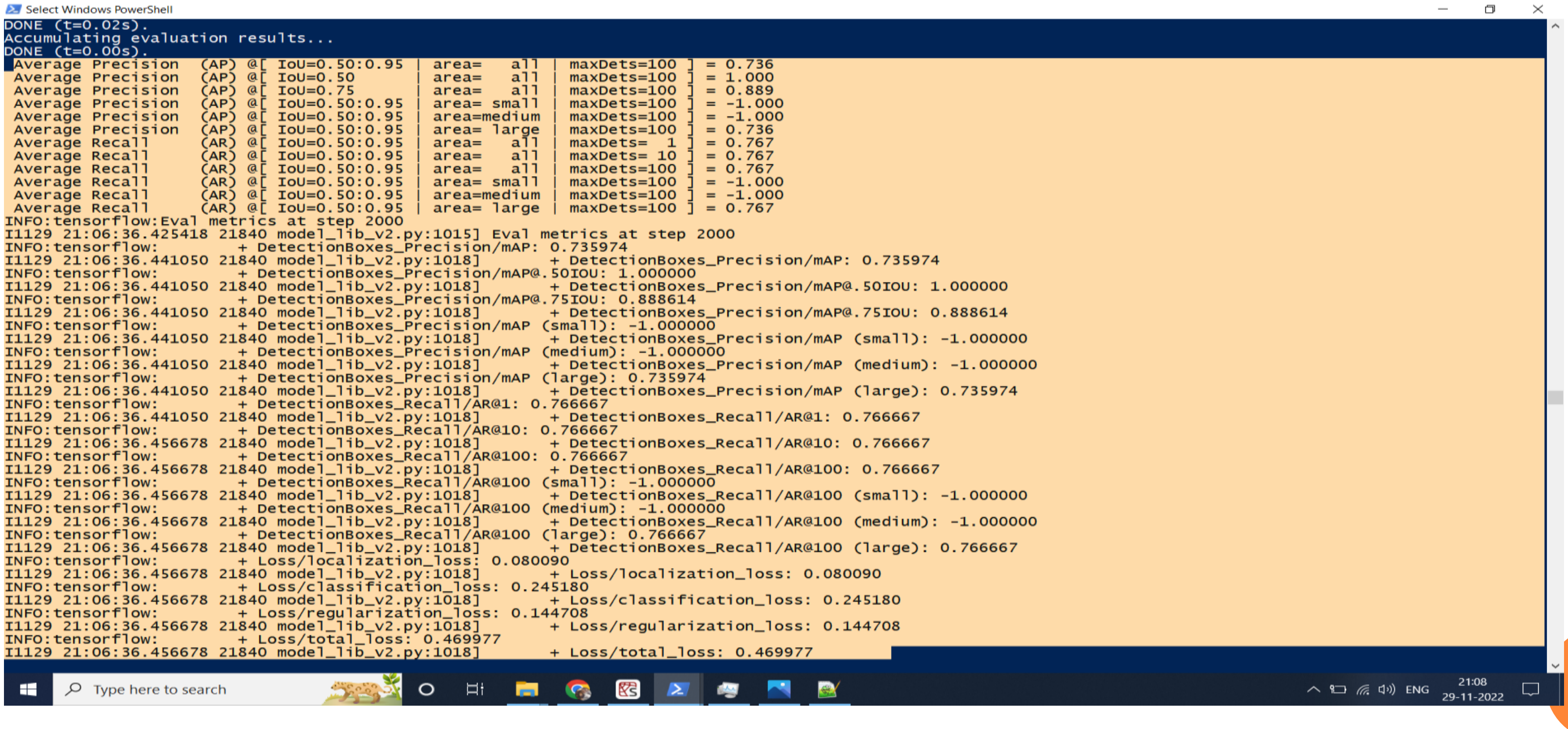


iis PC › COMMON (D:) › PERSONAL › STUDY › IISC › Code › PROJECT › Quality › CODE › Tensorflow › workspace › models ›

Name	Date modified	Type	Size
eval	29-11-2022 21:06	File folder	
export	29-11-2022 16:41	File folder	
tfjsexport	29-11-2022 16:41	File folder	
tfliteexport	29-11-2022 16:41	File folder	
train	29-11-2022 19:49	File folder	
checkpoint	29-11-2022 20:46	File	1 KB
ckpt-1.data-00000-of-00001	29-11-2022 19:51	DATA-00000-OF-0...	10,244 KB
ckpt-1.index	29-11-2022 19:51	INDEX File	26 KB
ckpt-2.data-00000-of-00001	29-11-2022 20:20	DATA-00000-OF-0...	20,300 KB
ckpt-2.index	29-11-2022 20:20	INDEX File	47 KB
ckpt-3.data-00000-of-00001	29-11-2022 20:46	DATA-00000-OF-0...	20,300 KB
ckpt-3.index	29-11-2022 20:46	INDEX File	47 KB
pipeline.config	29-11-2022 19:48	XML Configuration...	5 KB

TESTING MODEL

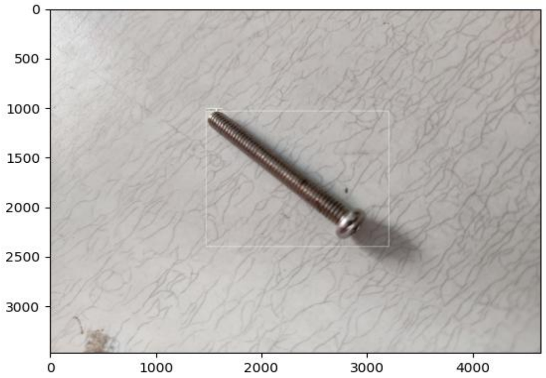
- Test data folder, Label map is provided as input
- Got Average Precision of about 73%

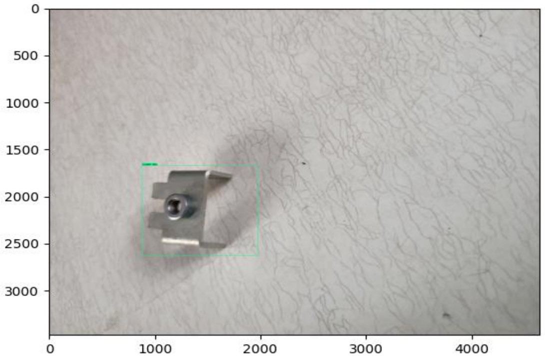


```
DONE (t=0.02s).
Accumulating evaluation results...
DONE (t=0.00s).
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.736
 Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 1.000
 Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.889
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = -1.000
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = -1.000
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.736
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.767
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.767
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.767
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = -1.000
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = -1.000
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.767
INFO:tensorflow:Eval metrics at step 2000
I1129 21:06:36.425418 21840 model_lib_v2.py:1015] Eval metrics at step 2000
INFO:tensorflow:        + DetectionBoxes_Precision/mAP: 0.735974
I1129 21:06:36.441050 21840 model_lib_v2.py:1018]       + DetectionBoxes_Precision/mAP: 0.735974
INFO:tensorflow:        + DetectionBoxes_Precision/mAP@.50IOU: 1.000000
I1129 21:06:36.441050 21840 model_lib_v2.py:1018]       + DetectionBoxes_Precision/mAP@.50IOU: 1.000000
INFO:tensorflow:        + DetectionBoxes_Precision/mAP@.75IOU: 0.888614
I1129 21:06:36.441050 21840 model_lib_v2.py:1018]       + DetectionBoxes_Precision/mAP@.75IOU: 0.888614
INFO:tensorflow:        + DetectionBoxes_Precision/mAP (small): -1.000000
I1129 21:06:36.441050 21840 model_lib_v2.py:1018]       + DetectionBoxes_Precision/mAP (small): -1.000000
INFO:tensorflow:        + DetectionBoxes_Precision/mAP (medium): -1.000000
I1129 21:06:36.441050 21840 model_lib_v2.py:1018]       + DetectionBoxes_Precision/mAP (medium): -1.000000
INFO:tensorflow:        + DetectionBoxes_Precision/mAP (large): 0.735974
I1129 21:06:36.441050 21840 model_lib_v2.py:1018]       + DetectionBoxes_Precision/mAP (large): 0.735974
INFO:tensorflow:        + DetectionBoxes_Recall/AR@1: 0.766667
I1129 21:06:36.441050 21840 model_lib_v2.py:1018]       + DetectionBoxes_Recall/AR@1: 0.766667
INFO:tensorflow:        + DetectionBoxes_Recall/AR@10: 0.766667
I1129 21:06:36.456678 21840 model_lib_v2.py:1018]       + DetectionBoxes_Recall/AR@10: 0.766667
INFO:tensorflow:        + DetectionBoxes_Recall/AR@100: 0.766667
I1129 21:06:36.456678 21840 model_lib_v2.py:1018]       + DetectionBoxes_Recall/AR@100: 0.766667
INFO:tensorflow:        + DetectionBoxes_Recall/AR@100 (small): -1.000000
I1129 21:06:36.456678 21840 model_lib_v2.py:1018]       + DetectionBoxes_Recall/AR@100 (small): -1.000000
INFO:tensorflow:        + DetectionBoxes_Recall/AR@100 (medium): -1.000000
I1129 21:06:36.456678 21840 model_lib_v2.py:1018]       + DetectionBoxes_Recall/AR@100 (medium): -1.000000
INFO:tensorflow:        + DetectionBoxes_Recall/AR@100 (large): 0.766667
I1129 21:06:36.456678 21840 model_lib_v2.py:1018]       + DetectionBoxes_Recall/AR@100 (large): 0.766667
INFO:tensorflow:        + Loss/localization_loss: 0.080090
I1129 21:06:36.456678 21840 model_lib_v2.py:1018]       + Loss/localization_loss: 0.080090
INFO:tensorflow:        + Loss/classification_loss: 0.245180
I1129 21:06:36.456678 21840 model_lib_v2.py:1018]       + Loss/classification_loss: 0.245180
INFO:tensorflow:        + Loss/regularization_loss: 0.144708
I1129 21:06:36.456678 21840 model_lib_v2.py:1018]       + Loss/regularization_loss: 0.144708
INFO:tensorflow:        + Loss/total_loss: 0.469977
I1129 21:06:36.456678 21840 model_lib_v2.py:1018]       + Loss/total_loss: 0.469977
```

EVALUATING MODEL

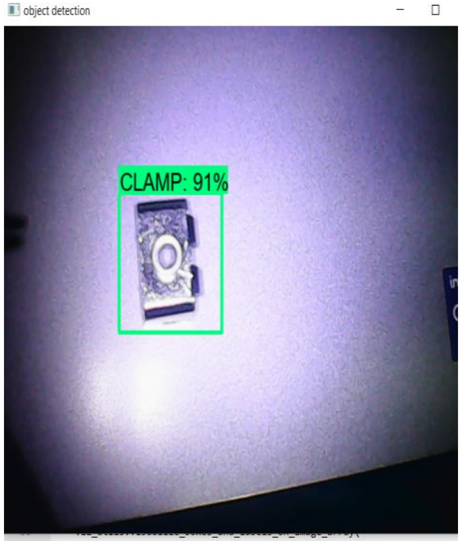
- Small python code is made to manually pass image to trained model
- Object in the model has been correctly identified got boxes and label around it

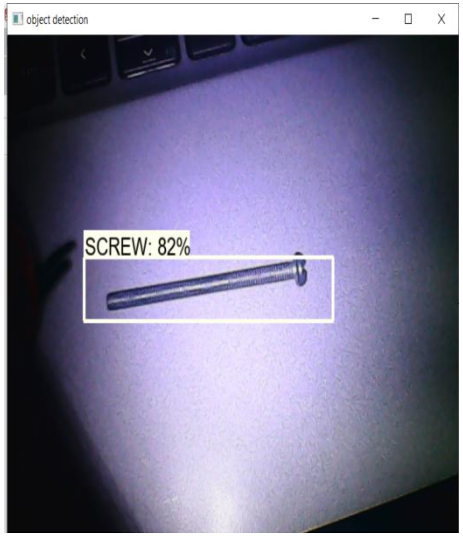


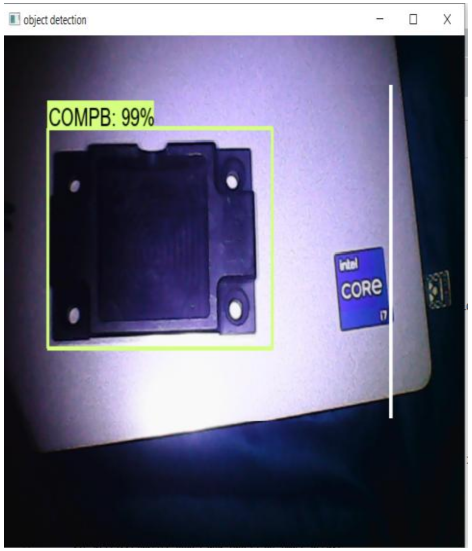


EVALUATING MODEL

- Small test python code is made which is acquire image from the camera and pass to model to identify if trained object is detected

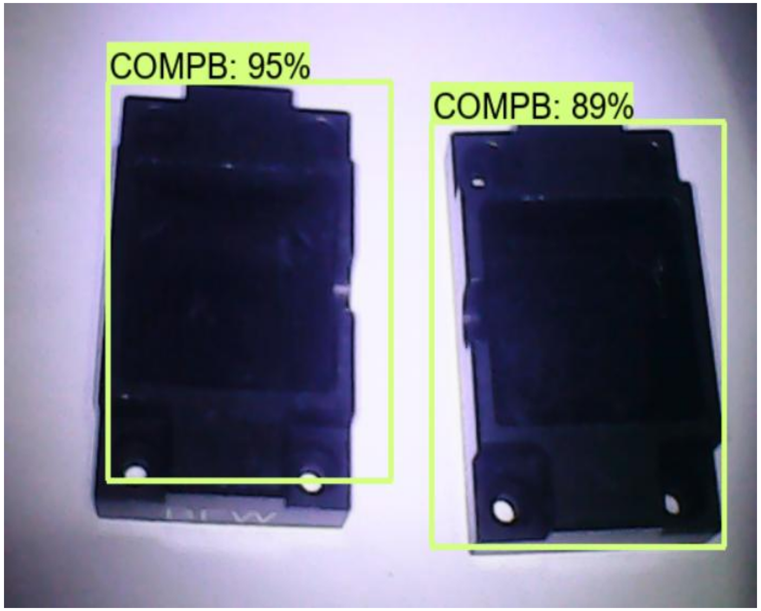


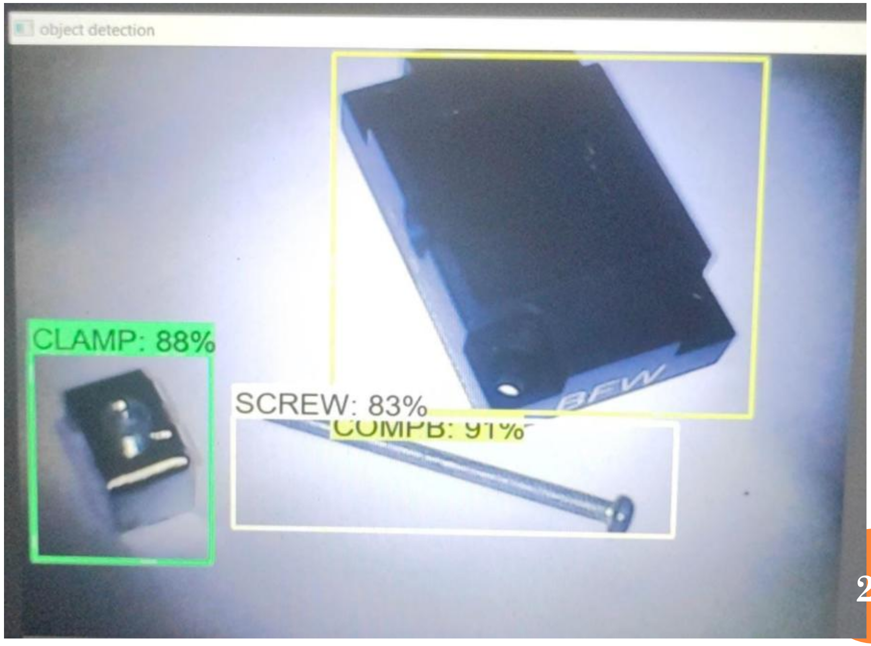




Evaluating model

- Image will Multiple variety of component is passed and tested
- Image will multiple number of component is passed and tested





CONCLUSION

- If the required component is available
- How many number of components available
- Various variety of component available

FUTURE

- Check if machining process completed or not
- Check if there is physical damage in the component

REFERANCES

- https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/tf2_detection_zoo.md
- https://www.tensorflow.org/guide/model_garden
- https://www.youtube.com/watch?v=yqkISICHH-U
- https://vidishmehta204.medium.com/object-detection-using-ssd-mobilenet-v2-7ff3543d738d

Thank you ....!