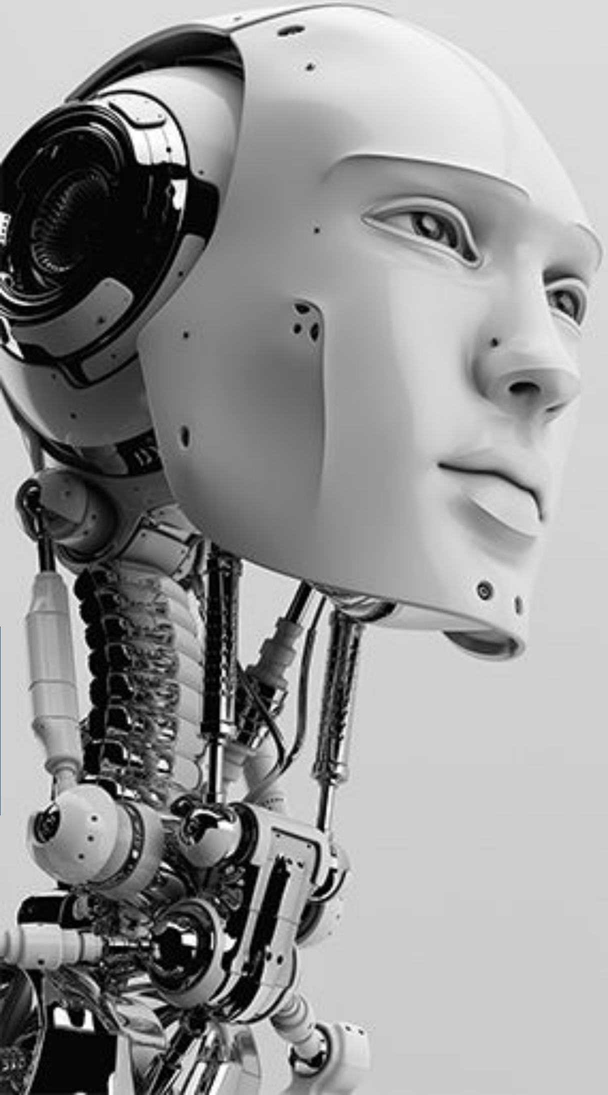
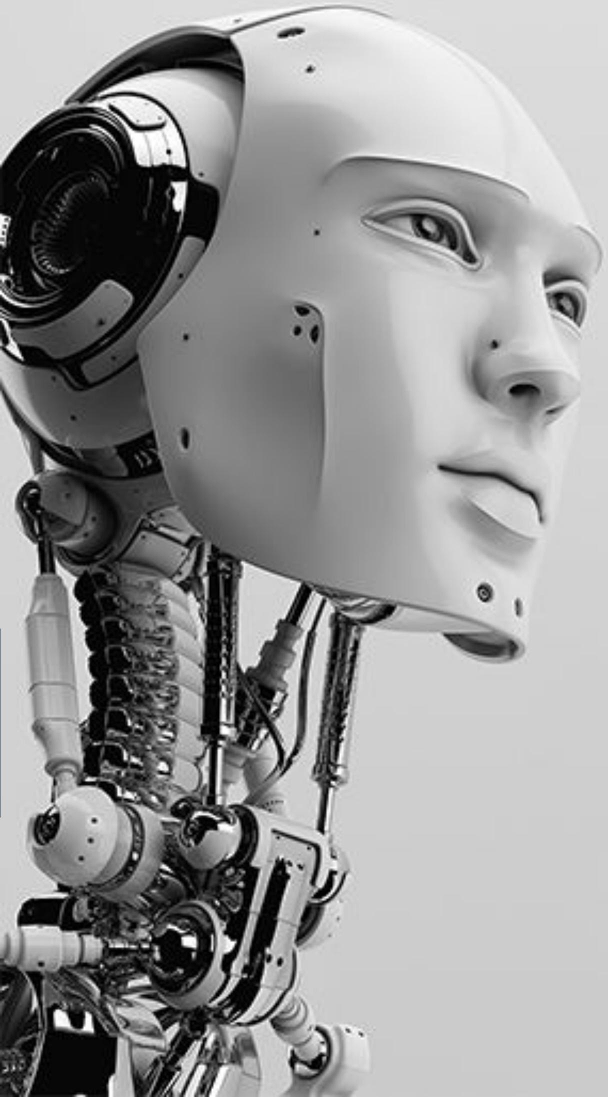


OBJECT DETECTION

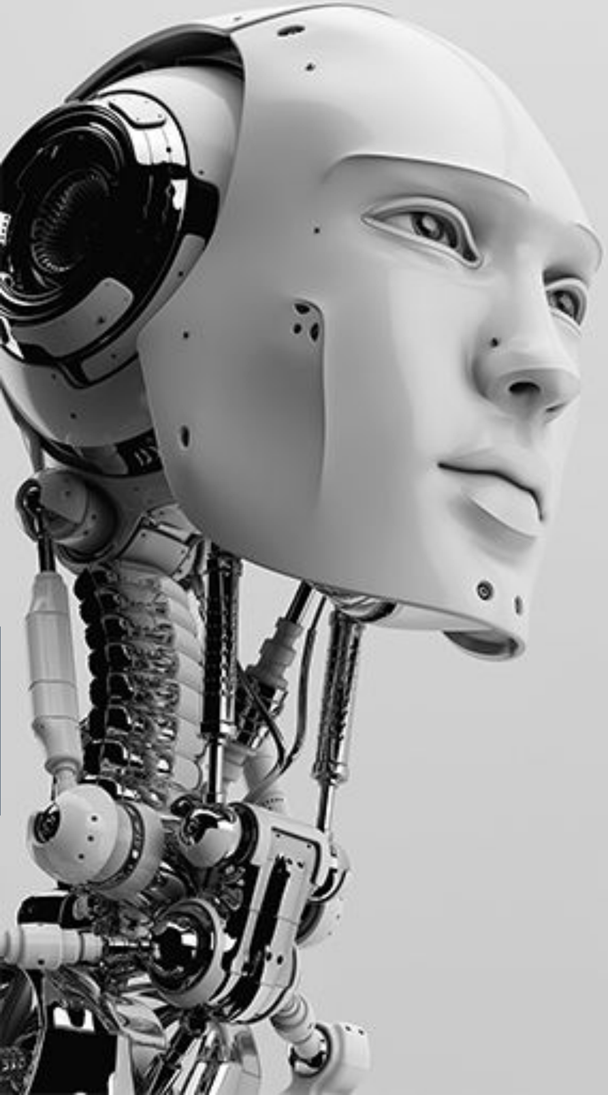
HARI KRISHNAN S
(IV year)

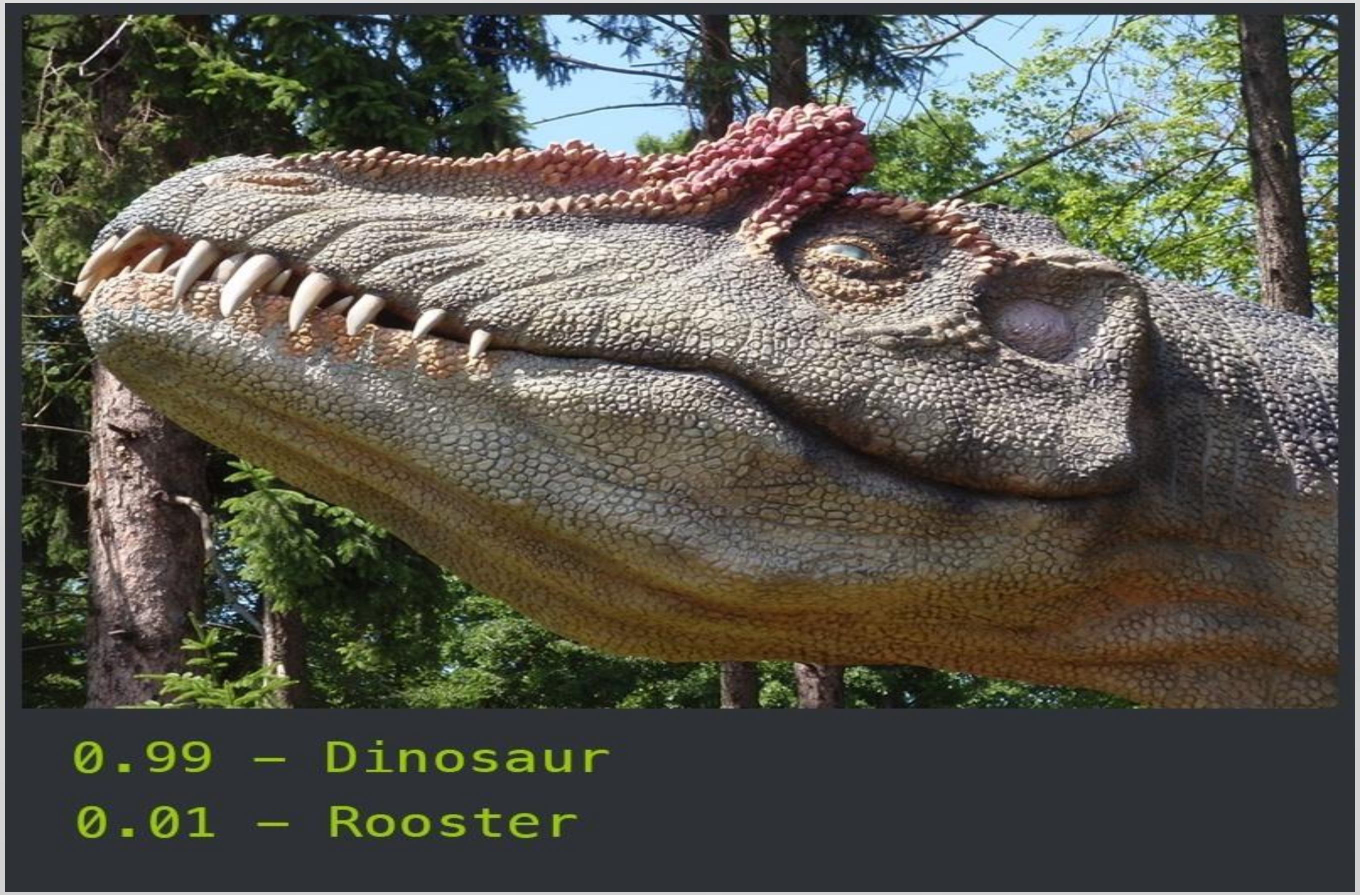
CSE dept,RMKEC

ABSTRACT



- Object Detection is the problem of finding and classifying a variable number of objects on an image.
- We are building the Currency-Detection model that can detect tens,twenties and onehundred.
- It is a DEEP LEARNING technique that uses R-CNN Model(R-region).



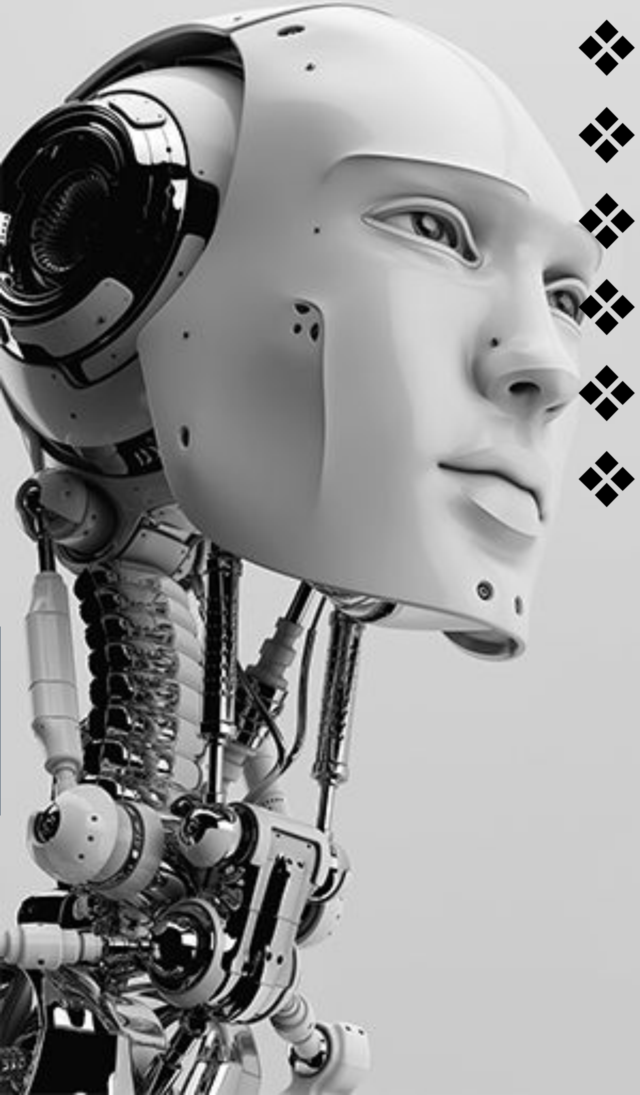


```
0.99 – Dinosaur
0.01 – Rooster
```

TOOLS USED:

Anaconda,Jupyter Notebook, Tensorflow, Label-img,Webcam.

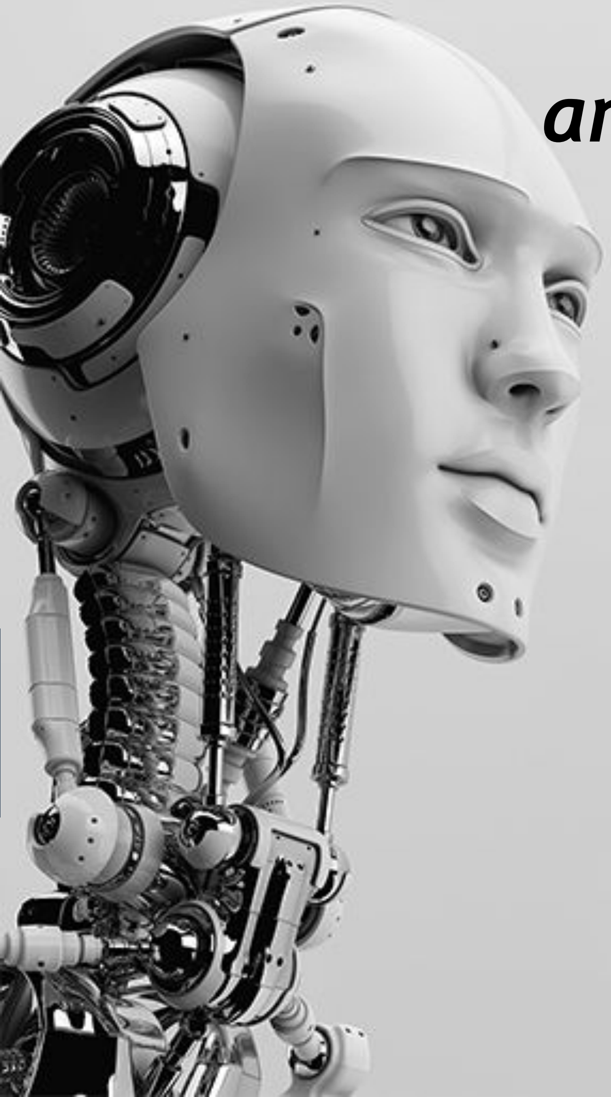
Model Steps



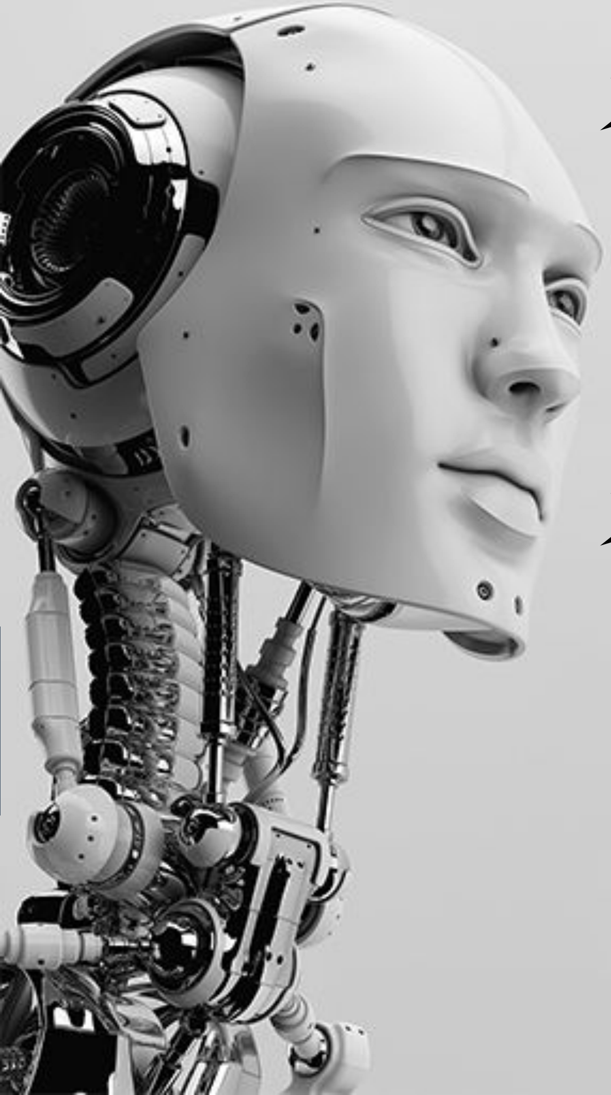
- ❖ Gathering and Labelling pictures
- ❖ Generating Training data
- ❖ Creating a Label map and configuring training
- ❖ Process Training
- ❖ Exporting the Inference graph
- ❖ Testing

Gathering and labelling pictures

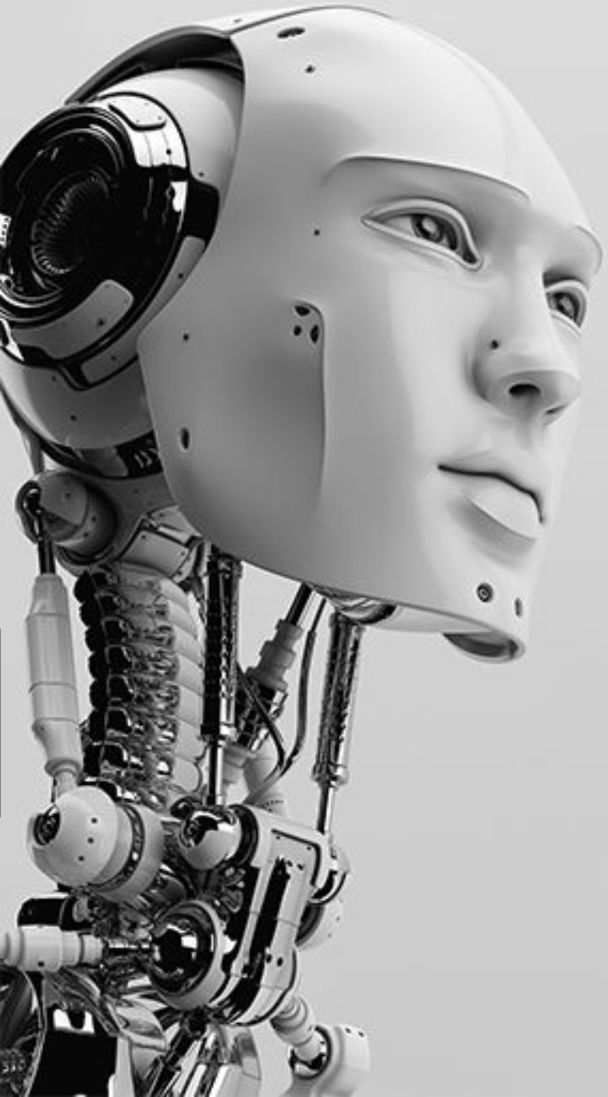
The Tensorflow object Detection API is set up and all the images are provided.



Process of Labelling



- ➢ *Once you have labelled and saved each image,there will be one .xml file for each image in the test and train directories.*
- ➢ *These .xml files used to generate TF records.*



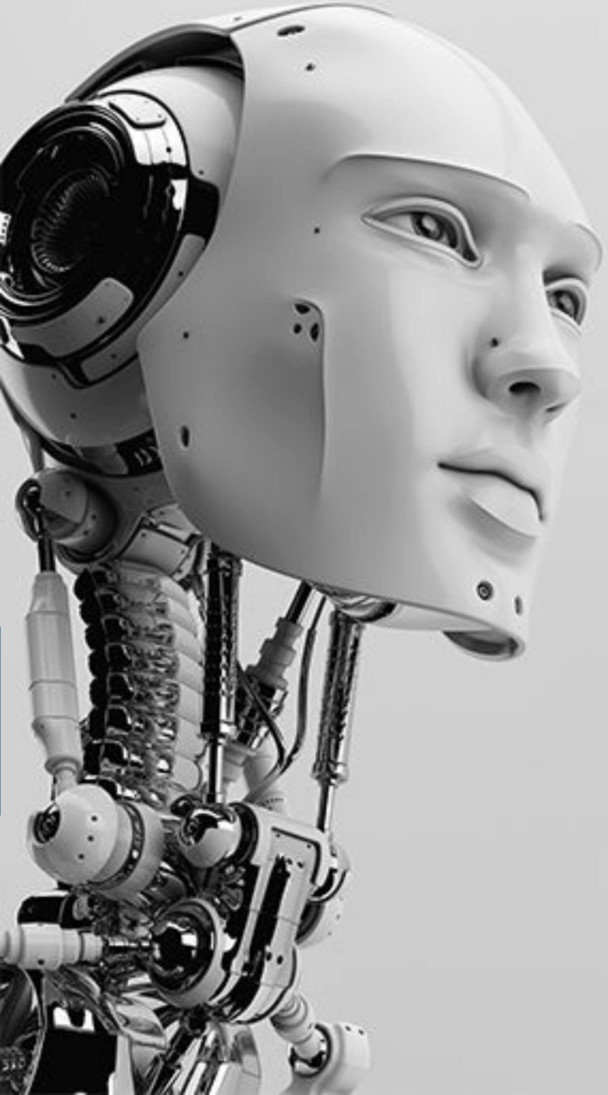
Generating the Training Data

TRAIN LABEL

```
def class_text_to_int(row_label):
    if row_label == 'Ten':
        return 1
    elif row_label == 'Twenty':
        return 2
    elif row_label == 'OneHundred':
        return 3
    else:
        None
```

LABEL MAP

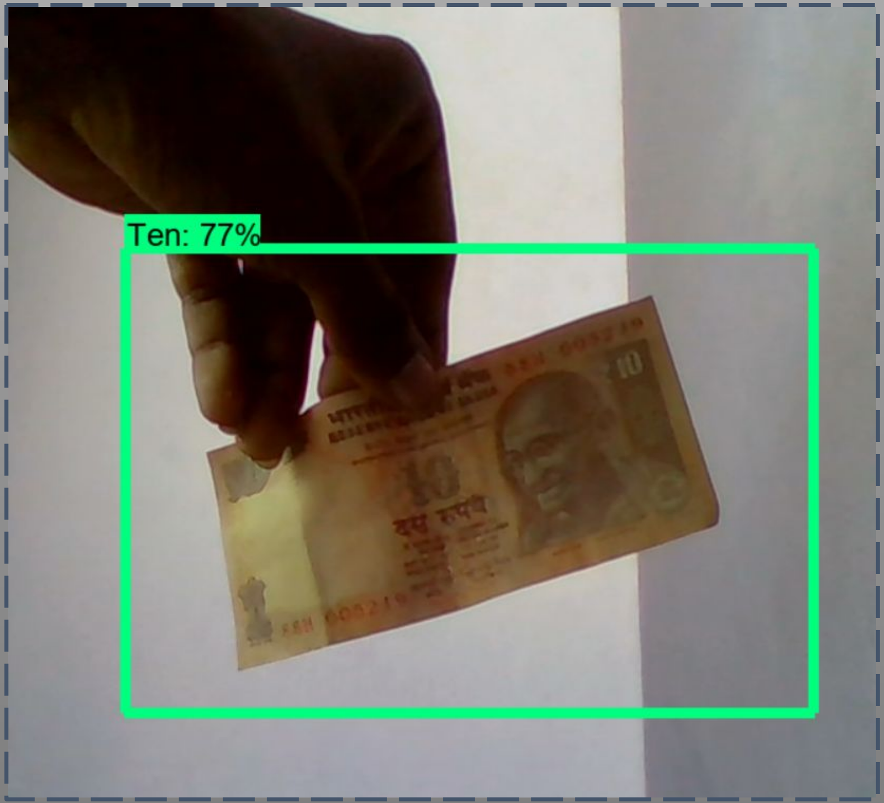
```
item {
  id: 1
  name: 'Ten’'
}
item {
  id: 2
  name: ‘Twenty'
}
item {
  id: 3
  name: 'OneHundred’
}
```

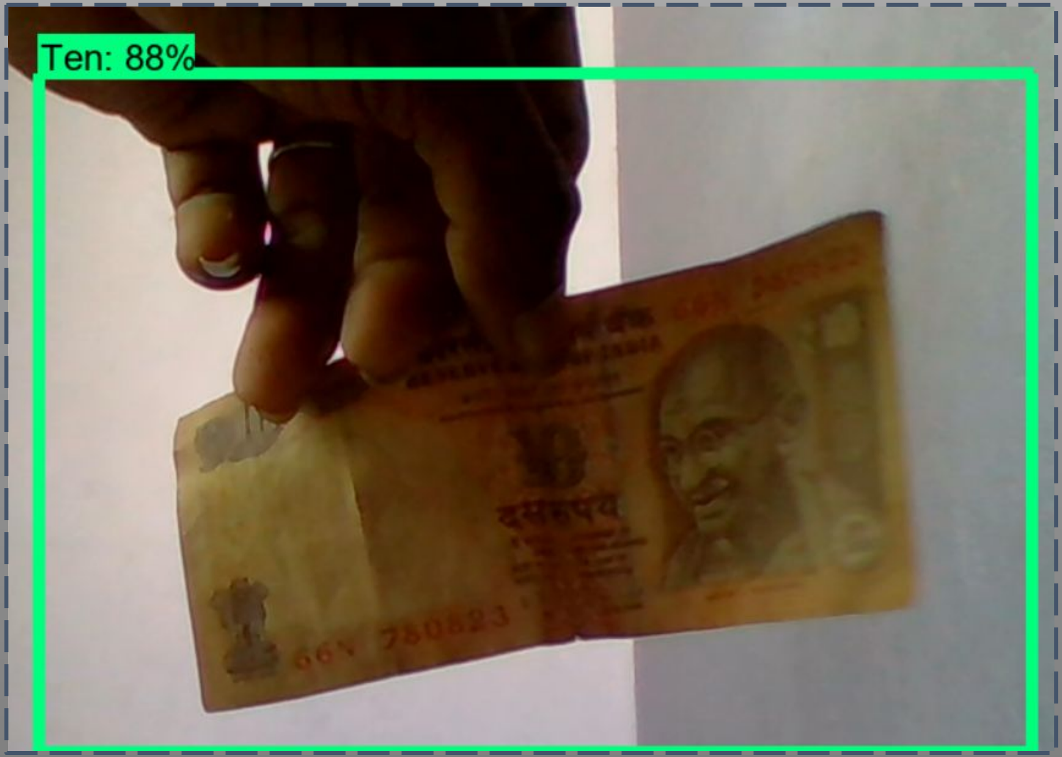


Configure Training

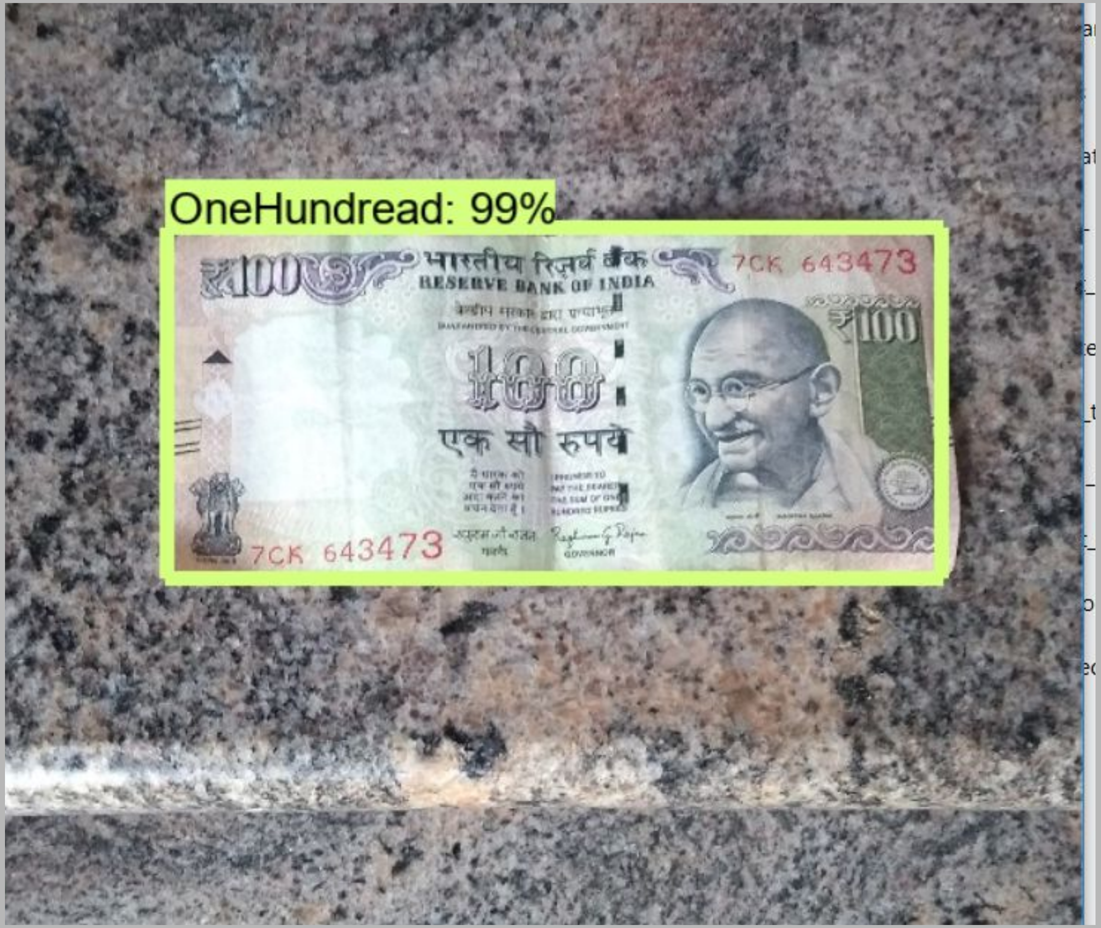
- ➔ *The object detection training must be configured.*
- ➔ *It defines which model and what parameters will be used for training.*

Detection of Rs.10 Note

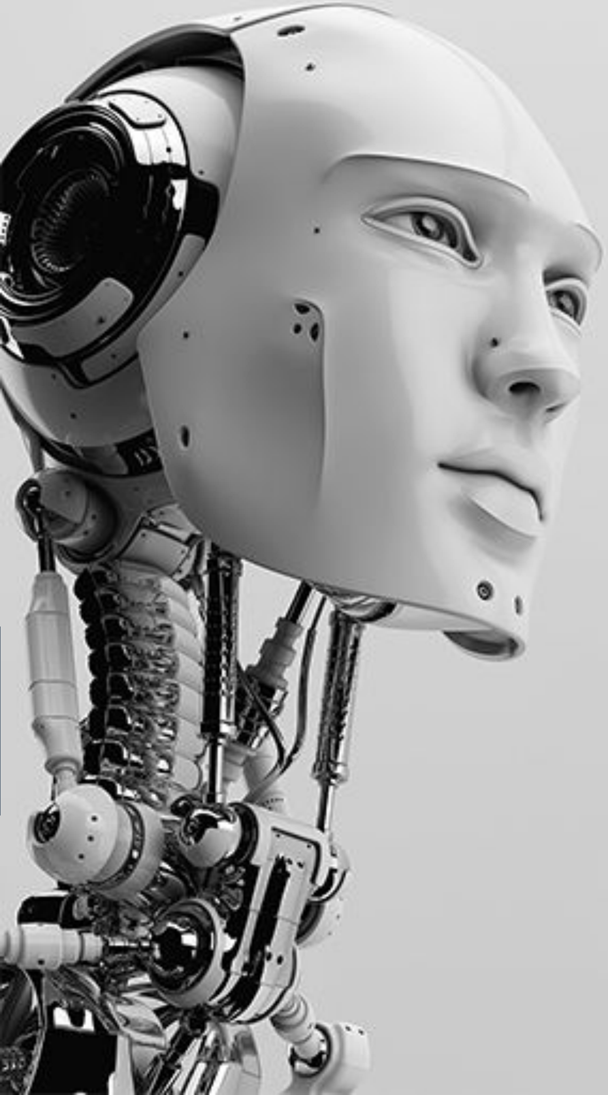




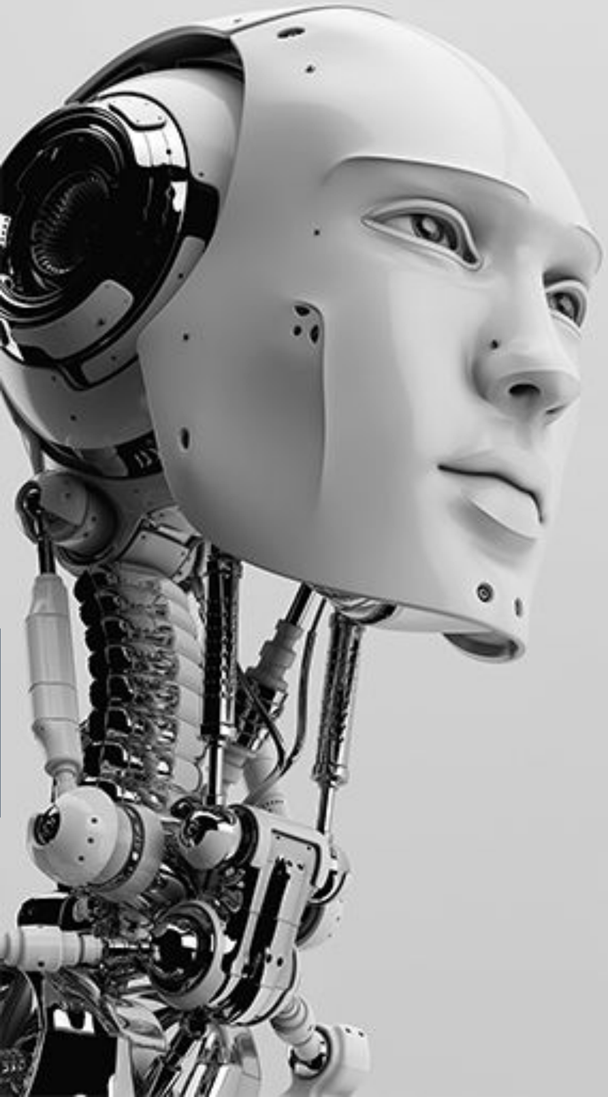
Detection of Rs.100 Note

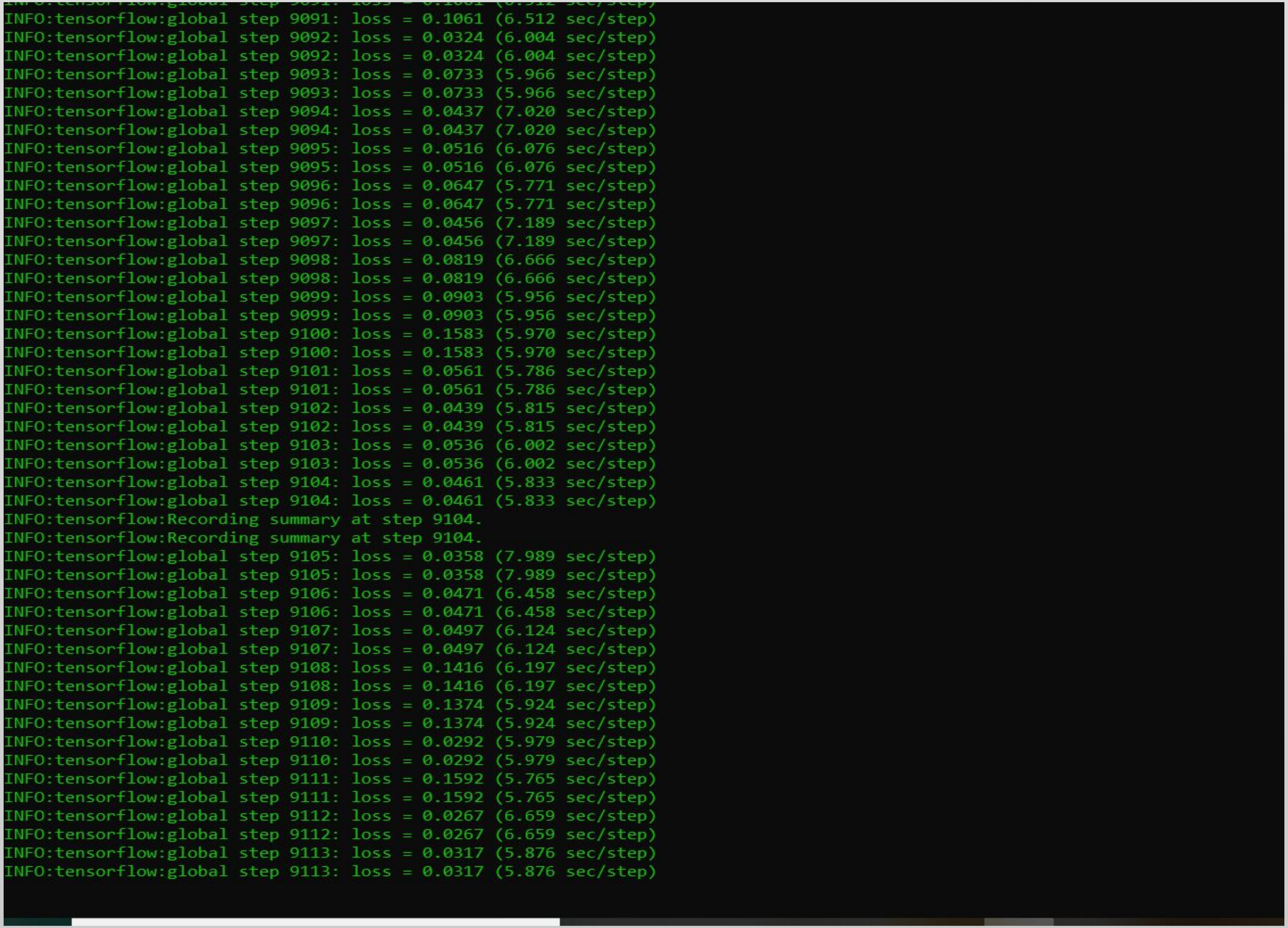


Process Training

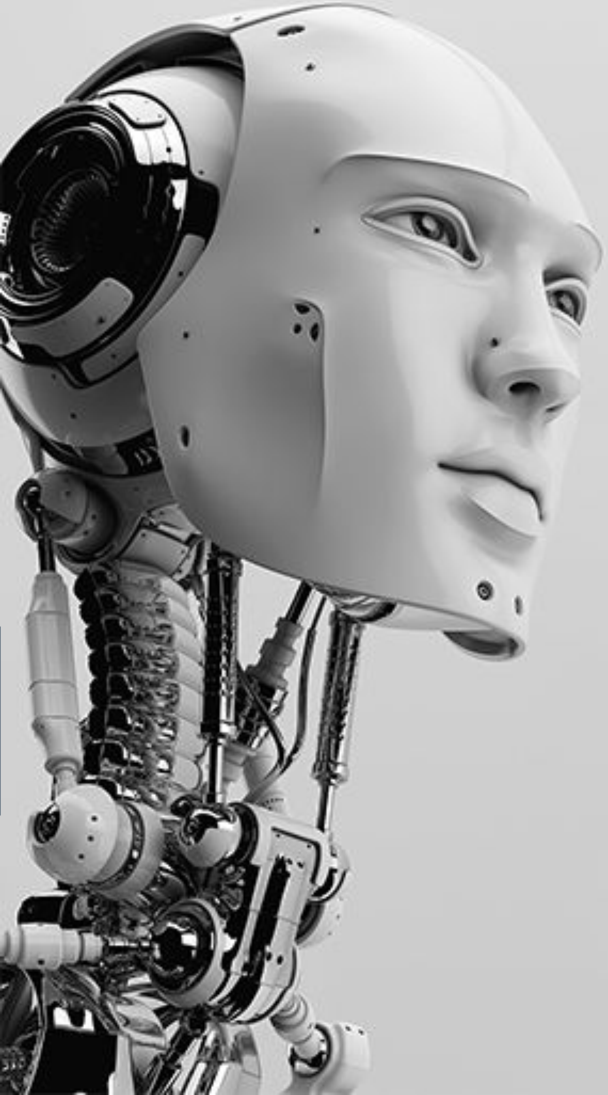


- *Each step of training reports the loss.*
- *It will start high and get lower and lower as training progresses.*
- *Loss=0.005 is a good accuracy for detecting an image.*





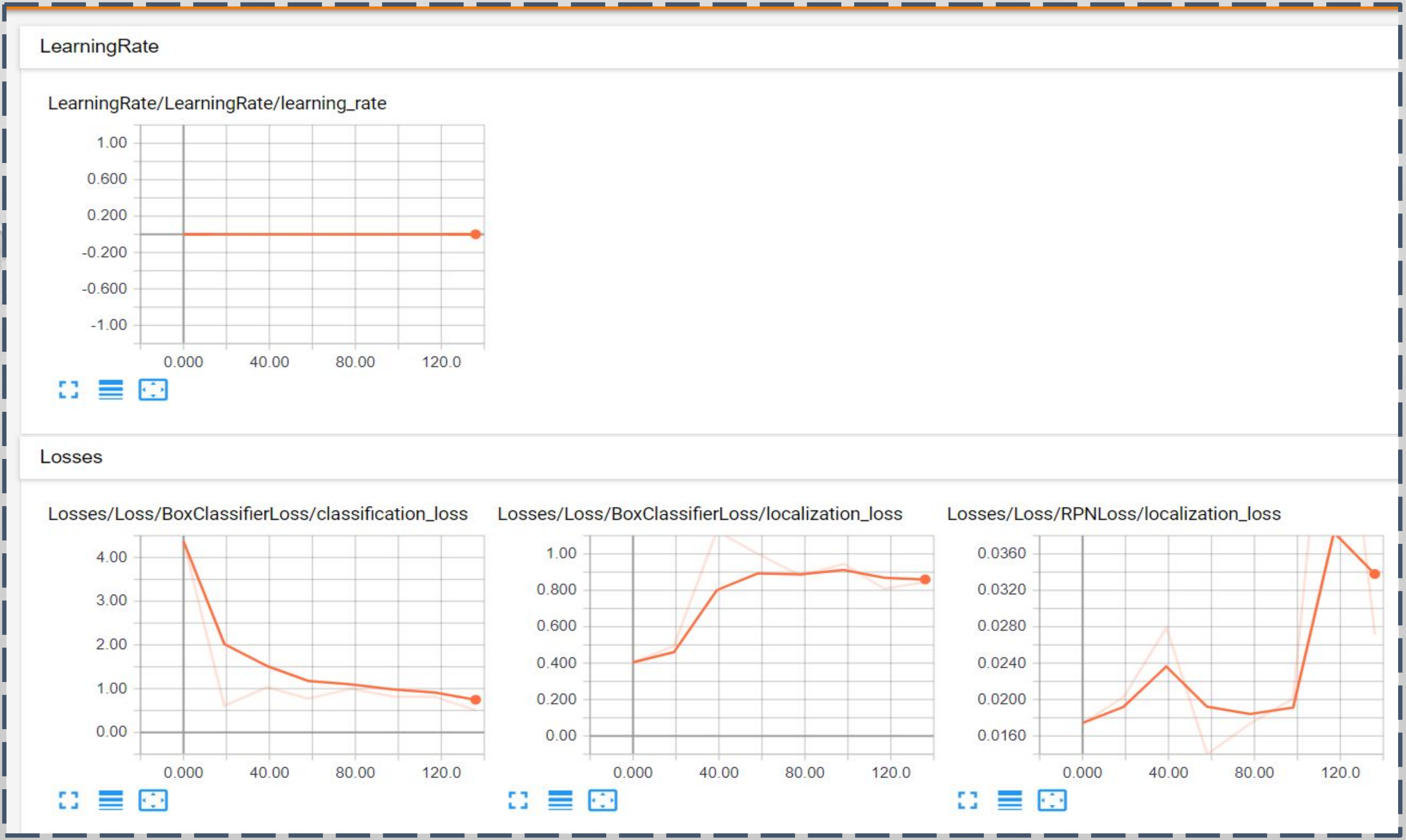
```
INFO:tensorflow:global step 9091: loss = 0.1061 (6.512 sec/step)
INFO:tensorflow:global step 9091: loss = 0.1061 (6.512 sec/step)
INFO:tensorflow:global step 9092: loss = 0.0324 (6.004 sec/step)
INFO:tensorflow:global step 9092: loss = 0.0324 (6.004 sec/step)
INFO:tensorflow:global step 9093: loss = 0.0733 (5.966 sec/step)
INFO:tensorflow:global step 9093: loss = 0.0733 (5.966 sec/step)
INFO:tensorflow:global step 9094: loss = 0.0437 (7.020 sec/step)
INFO:tensorflow:global step 9094: loss = 0.0437 (7.020 sec/step)
INFO:tensorflow:global step 9095: loss = 0.0516 (6.076 sec/step)
INFO:tensorflow:global step 9095: loss = 0.0516 (6.076 sec/step)
INFO:tensorflow:global step 9096: loss = 0.0647 (5.771 sec/step)
INFO:tensorflow:global step 9096: loss = 0.0647 (5.771 sec/step)
INFO:tensorflow:global step 9097: loss = 0.0456 (7.189 sec/step)
INFO:tensorflow:global step 9097: loss = 0.0456 (7.189 sec/step)
INFO:tensorflow:global step 9098: loss = 0.0819 (6.666 sec/step)
INFO:tensorflow:global step 9098: loss = 0.0819 (6.666 sec/step)
INFO:tensorflow:global step 9099: loss = 0.0903 (5.956 sec/step)
INFO:tensorflow:global step 9099: loss = 0.0903 (5.956 sec/step)
INFO:tensorflow:global step 9100: loss = 0.1583 (5.970 sec/step)
INFO:tensorflow:global step 9100: loss = 0.1583 (5.970 sec/step)
INFO:tensorflow:global step 9101: loss = 0.0561 (5.786 sec/step)
INFO:tensorflow:global step 9101: loss = 0.0561 (5.786 sec/step)
INFO:tensorflow:global step 9102: loss = 0.0439 (5.815 sec/step)
INFO:tensorflow:global step 9102: loss = 0.0439 (5.815 sec/step)
INFO:tensorflow:global step 9103: loss = 0.0536 (6.002 sec/step)
INFO:tensorflow:global step 9103: loss = 0.0536 (6.002 sec/step)
INFO:tensorflow:global step 9104: loss = 0.0461 (5.833 sec/step)
INFO:tensorflow:global step 9104: loss = 0.0461 (5.833 sec/step)
INFO:tensorflow:Recording summary at step 9104.
INFO:tensorflow:Recording summary at step 9104.
INFO:tensorflow:global step 9105: loss = 0.0358 (7.989 sec/step)
INFO:tensorflow:global step 9105: loss = 0.0358 (7.989 sec/step)
INFO:tensorflow:global step 9106: loss = 0.0471 (6.458 sec/step)
INFO:tensorflow:global step 9106: loss = 0.0471 (6.458 sec/step)
INFO:tensorflow:global step 9107: loss = 0.0497 (6.124 sec/step)
INFO:tensorflow:global step 9107: loss = 0.0497 (6.124 sec/step)
INFO:tensorflow:global step 9108: loss = 0.1416 (6.197 sec/step)
INFO:tensorflow:global step 9108: loss = 0.1416 (6.197 sec/step)
INFO:tensorflow:global step 9109: loss = 0.1374 (5.924 sec/step)
INFO:tensorflow:global step 9109: loss = 0.1374 (5.924 sec/step)
INFO:tensorflow:global step 9110: loss = 0.0292 (5.979 sec/step)
INFO:tensorflow:global step 9110: loss = 0.0292 (5.979 sec/step)
INFO:tensorflow:global step 9111: loss = 0.1592 (5.765 sec/step)
INFO:tensorflow:global step 9111: loss = 0.1592 (5.765 sec/step)
INFO:tensorflow:global step 9112: loss = 0.0267 (6.659 sec/step)
INFO:tensorflow:global step 9112: loss = 0.0267 (6.659 sec/step)
INFO:tensorflow:global step 9113: loss = 0.0317 (5.876 sec/step)
INFO:tensorflow:global step 9113: loss = 0.0317 (5.876 sec/step)
```



Advantages

- *It is especially designed for the Blind and Low vision community.*
- *It is used for currency conversion.*
- *It is also used in Entertainment domain to train a “PINOCHLE DECK” playing card detector etc,.*

Inference Graph1



LearningRate

LearningRate/LearningRate/learning_rate

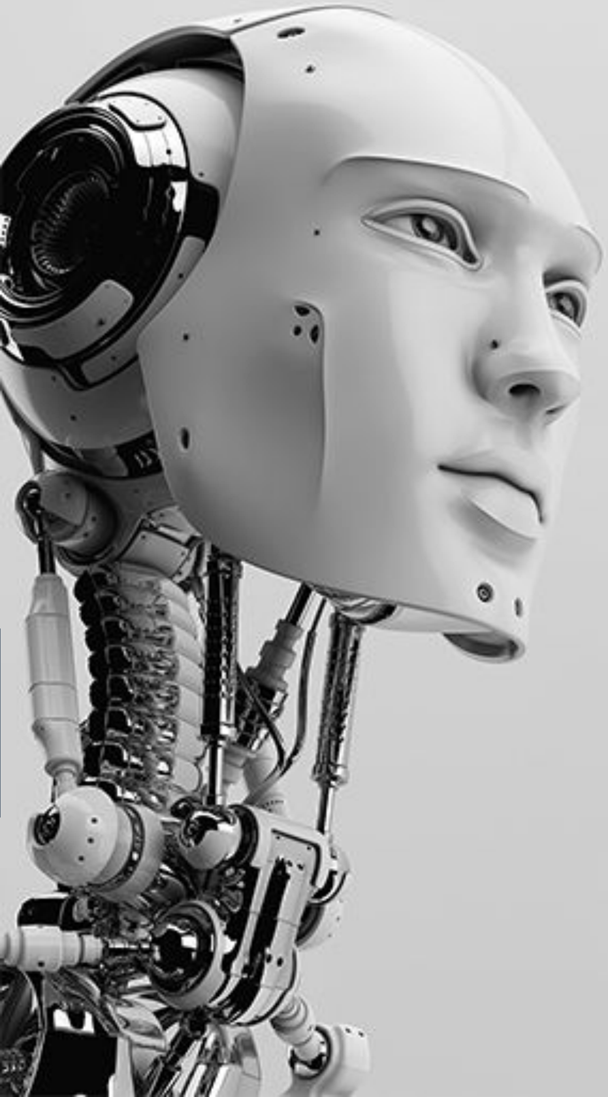
Losses

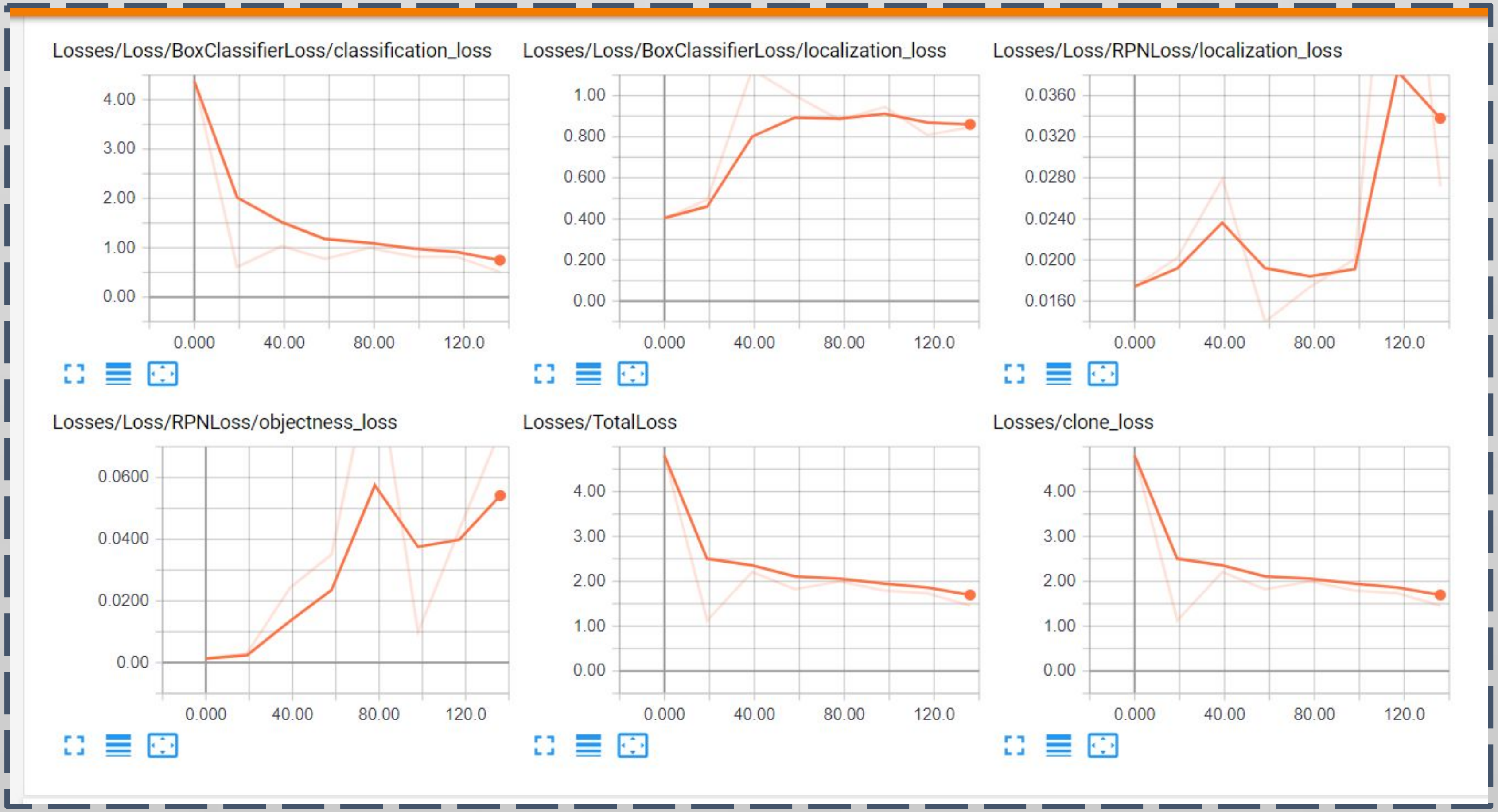
Losses/Loss/BoxClassifierLoss/classification_loss

Losses/Loss/BoxClassifierLoss/localization_loss

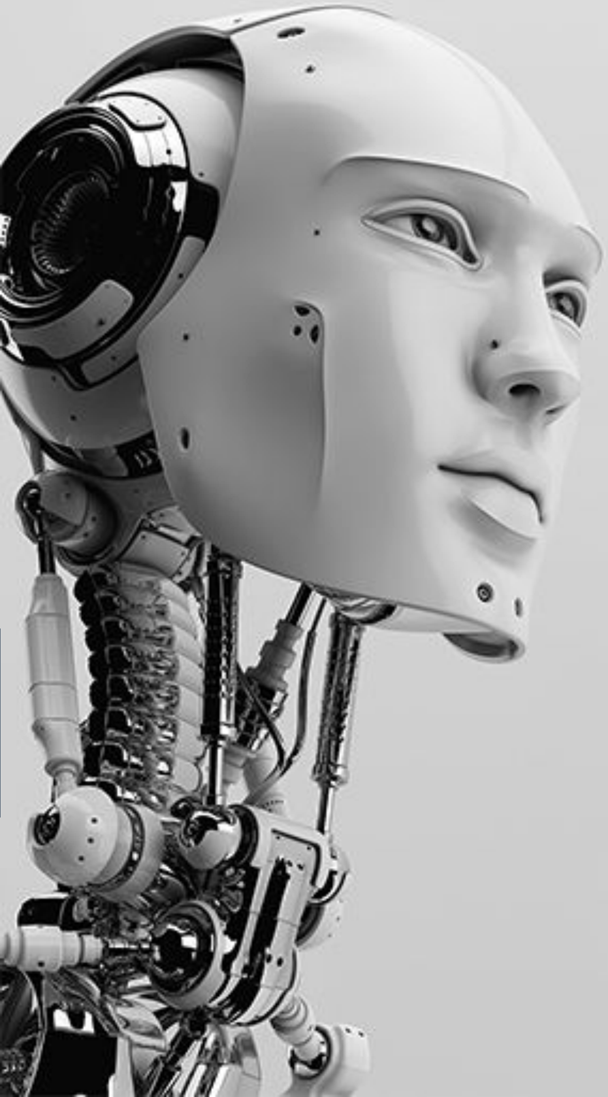
Losses/Loss/RPNLoss/localization_loss

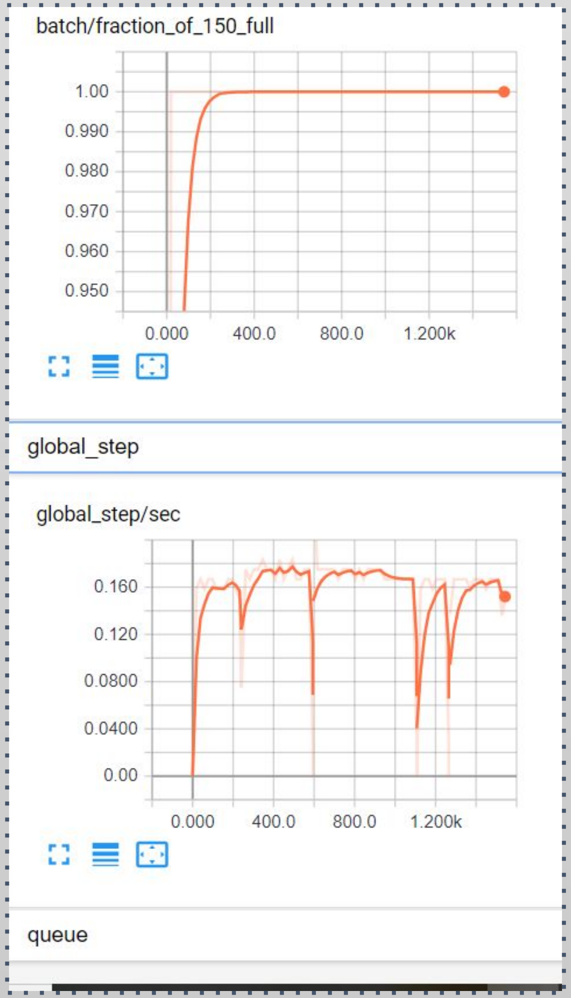
Inference Graph 2





After step 1500



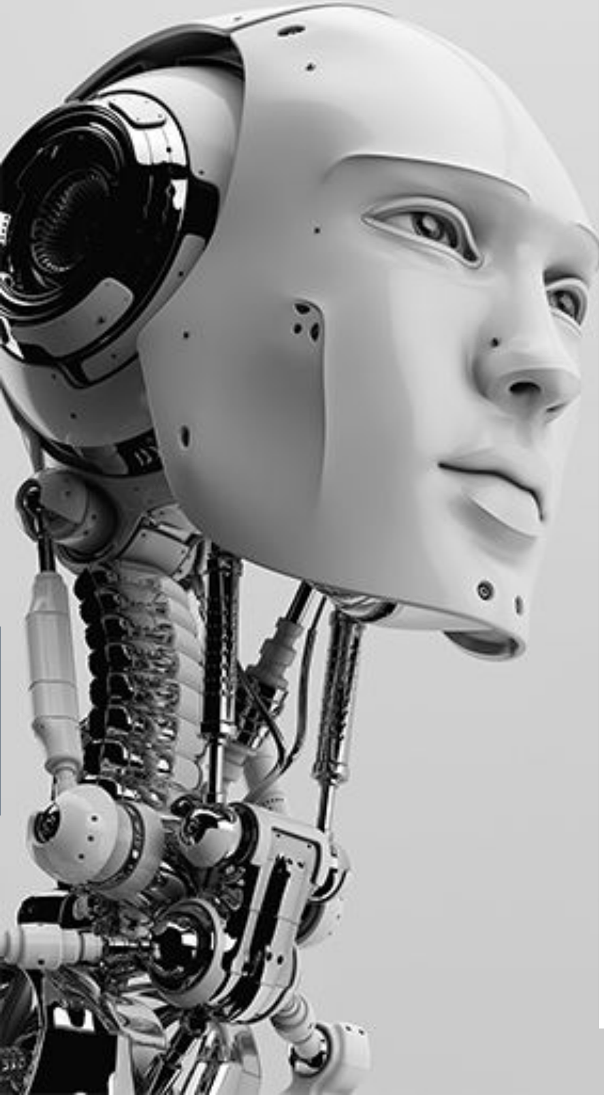


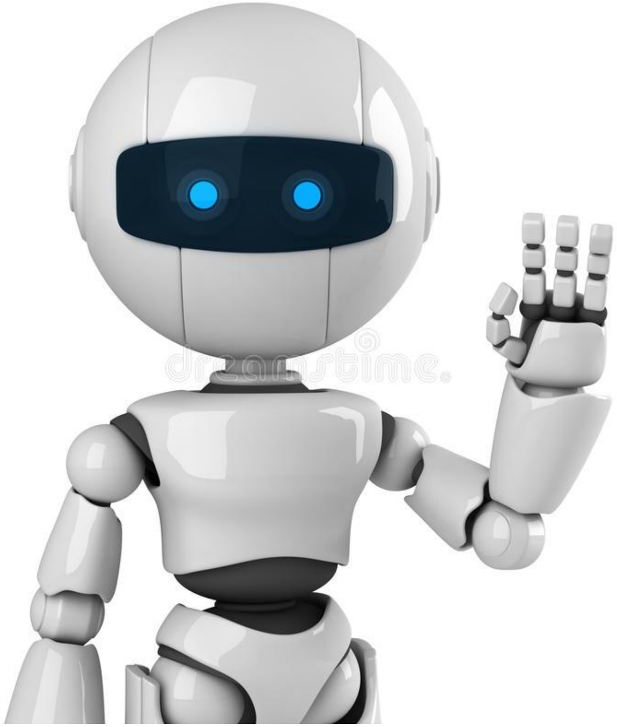
batch/fraction_of_150_full

global_step

global_step/sec

queue





THANK YOU!