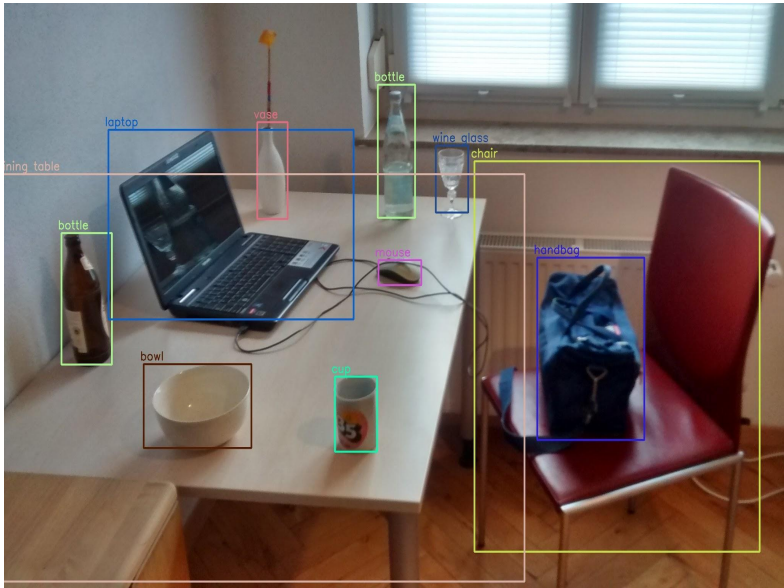
Creation of 2D Synthetic dataset and Object Detection Training

For Data Mining and Machine learning subject By Gowtham B C

Introduction to Object detection field

Object detection is a sub field of Computer Vision where in provided the image the model should be able to recognise the **Class** and **Object** in the image at **highest accuracy** and **least latency**.

- Class: Say the object in image is a Milk carton
- Object: The object is in Xmin, Ymin, Xmax, Ymax coordinate.
- Highest accuracy: Can be co related to confidence score. (For Academia, this criteria is important.)
- Latency: At higher speed above task needs to be done. (For Industry, this criteria is important.)



Problem Statement:

Object detection for Custom Objects

- Lack of specific niche product image data. Ex:Mars chocolate
- Lack of corresponding annotated dataset.
- For Corporate production release, time to add new product is high.
    - Creation of new dataset.
        - The process of data annotation was done manually. For every 100 images for 2 hours.
    - Training the model on this new dataset.
        - Thumb rule: training in Object detection field is of 30000 epochs.

Solution:

- Creation of Dataset using **Synthetic process**.
- Using the power of **Transfer learning** to train and deploy model quickly.

Implementation steps:

1. Creation of synthetic data and corresponding annotated values.
2. Break the created synthetic data to format required for the Deep Neural Network model i.e. JSON to Pascal VOC XML
3. Train the Deep Neural Network model.
4. Inference the model (as a Company requirement to build product) or Benchmark the accuracy and speed of model (a research paper requirement.)

Why Synthetic data is required?

A lot of models or datasets are available now, why do we need to spend time on dataset creation? (Object detection domain)

- The pretrained available models are **trained on freely available datasets** for research purpose like coco dataset.
- But these datasets have **license restrictions** on use for Professional environment such as companies for profit. (BSD, JRL)
- Professional environment **needs more data** than just Proof Of Concept to solve problems in real world.
- Especially creation of these **customised datasets for clients** are still a concern in research as well as industry.
- **Using synthetic dataset, huge number of dataset, as realistic to real life examples can be obtained in an instant.**

How to create synthetic data?

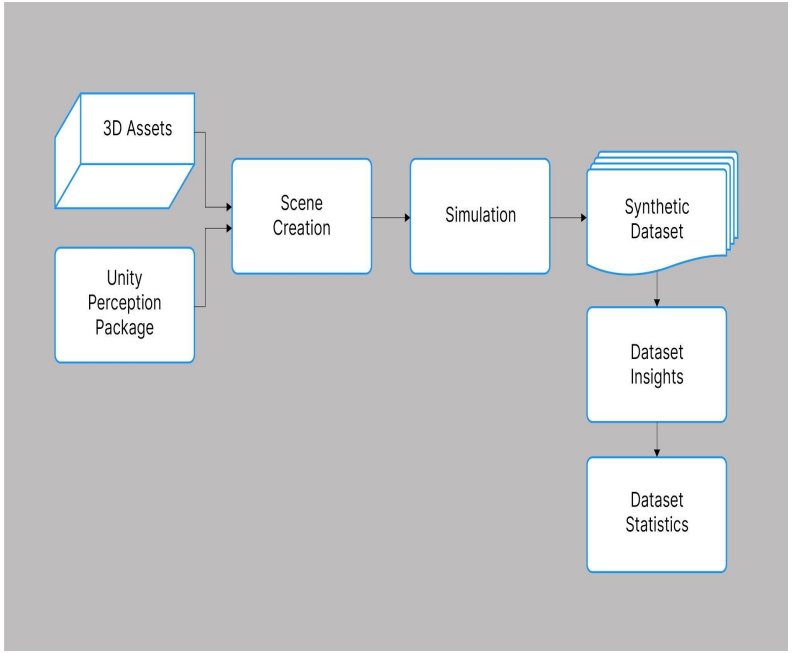
First, buy the product from physical store and scan the wrappers.

Create the 3D cube model in blender and wrap it around it using scanned images from “Blender” software. (prefab format)

Using unity 3D, import then in virtual environment.

Jumble the products, find and store ground truth values and images.

Voila, Synthetic dataset is created.




CAP'N CRUNCH
CAP'N CRUNCH
Polygons' UV coordinates are unwrapped
Textures are correctly aligned to objects

Trivia:

Well funded companies in Automatic retail checkout, Smart solutions such as Aifi, Standard Ai use this mechanism to train there model.

The Google germany on Feb 2019, presented a paper on synthetic dataset: “An Annotation Saved is an Annotation Earned: Using Fully Synthetic Training for Object Instance Detection”.

How does Sample object detection data looks like?

"filename": "RGBf9199b2f-0000-41b4-a01a-4b690831f6c1/rgb_94.png", "format": "PNG", "annotations": [
{
"id": "35108dcd-54b8-4567-9f57-168ac0293e40",
"annotation_definition":"f9f22e05-443f-4602-a422-ebe4ea9b55cb",
"values": [
{"label_id": 9,
"label_name": "snack_biscotti_ghiott",
"instance_id": 1,
"x": 399.0,
"y": 189.0,
"width": 45.0,
"height": 40.0
},

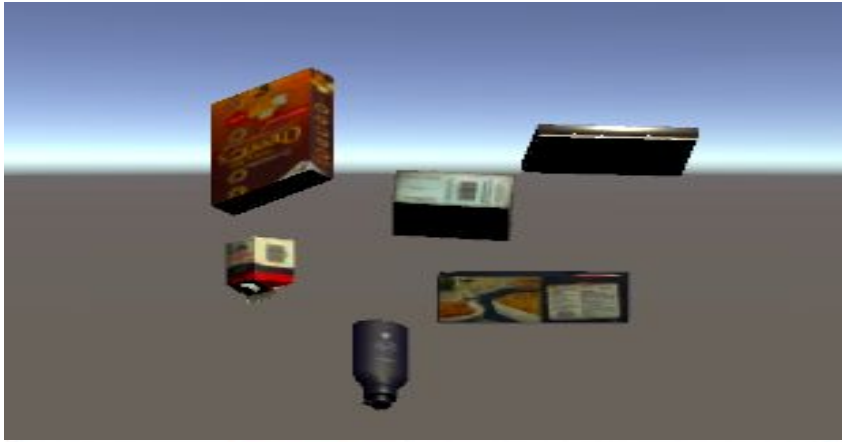


Image Source: Right bottom[2]

Synthetic dataset creation:

- Using Unity perception package, synthetic dataset is created.
- This simulation is replica of Google Cloud AI research group.

Similarities to original paper:

- Random Rotation, Hue variation in X,Y,Z axis are considered.
- Object in Dataset is equally distributed.

Our improvements:

- Object placement is varied in X and Y axis only.
- Object size is increased for better feature extraction.
- Shadows are turned off.
- Directional light is made ambient so all area is illuminated.

Results:

- This method could generate 1,00,000 images and annotated dataset within 5 minutes.
- For manual annotation, more than a month is required.

Conversion of JSON to Pascal VOC

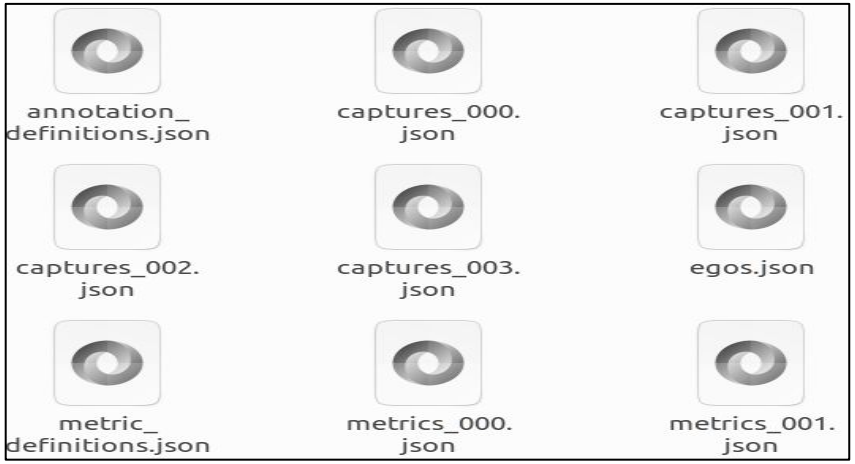
- The previous obtained JSON images are needed to be converted to below format.
- Using NodeJs, a custom solution was developed to solve this problem.

```
<annotation><folder>labelled_data</folder><filename>rgb_3.png</filename><path>/home/gowtham/Desktop/Unity3d/ConvertJSONtoXML/labelled_data/rgb_3.png</-
path><size><width>640</width><height>640</height><depth>3</depth></size><object><name>snack_granolabar_naturevalley</name><bndbox><xmin>273</xmin><ymin>281</-
ymin><xmax>438</xmax><ymax>440</ymax></bndbox></object><object><name>drink_whippingcream_lucerne</name><bndbox><xmin>471</xmin><ymin>246</ymin><xmax>592</-
xmax><ymax>374</ymax></bndbox></object><object><name>craft_yarn_caron</name><bndbox><xmin>449</xmin><ymin>84</ymin><xmax>572</xmax><ymax>209</ymax></bndbox></-
object><object><name>snack_granolabar_naturevalley</name><bndbox><xmin>243</xmin><ymin>85</ymin><xmax>432</xmax><ymax>277</ymax></bndbox></-
object><object><name>drink_greentea_itoen</name><bndbox><xmin>100</xmin><ymin>76</ymin><xmax>234</xmax><ymax>248</ymax></bndbox></-
object><object><name>snack_biscotti_ghiott</name><bndbox><xmin>0</xmin><ymin>261</ymin><xmax>149</xmax><ymax>451</ymax></bndbox></-
object><object><name>cleaning_snuggle_henkel</name><bndbox><xmin>128</xmin><ymin>344</ymin><xmax>297</xmax><ymax>496</ymax></bndbox></-
object><object><name>craft_yarn_caron</name><bndbox><xmin>295</xmin><ymin>495</ymin><xmax>481</xmax><ymax>592</ymax></bndbox></-
object><object><name>craft_yarn_caron</name><bndbox><xmin>470</xmin><ymin>436</ymin><xmax>607</xmax><ymax>542</ymax></bndbox></-
object><object><name>craft_yarn_caron</name><bndbox><xmin>0</xmin><ymin>507</ymin><xmax>110</xmax><ymax>585</ymax></bndbox></-
object><object><name>cleaning_snuggle_henkel</name><bndbox><xmin>138</xmin><ymin>554</ymin><xmax>325</xmax><ymax>640</ymax></bndbox></-
object><object><name>craft_yarn_caron</name><bndbox><xmin>0</xmin><ymin>1</ymin><xmax>94</xmax><ymax>152</ymax></bndbox></object></annotation>
```

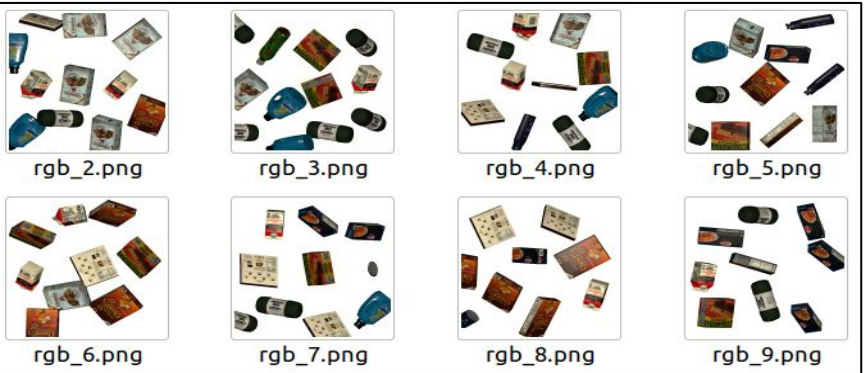
Is this step required?

- Yes and No, Yes because, It is always the best to practise industry format. No because, If we change our training model to read input in JSON format, This step could be avoided.
- In this case, i wanted to follow industry format.

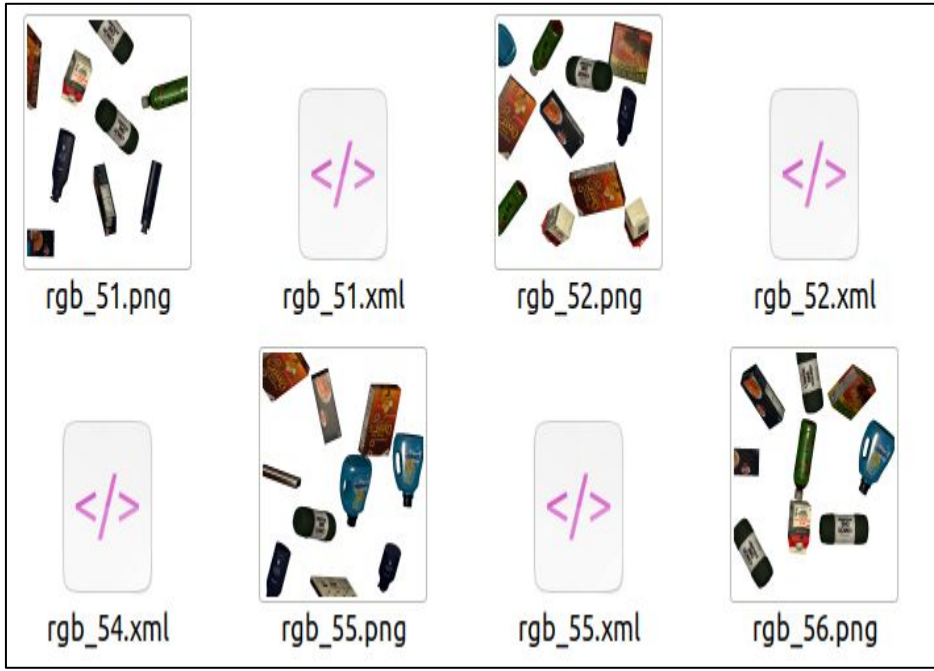
Results:


annotation_definitions.json
captures_000.json
captures_001.json
captures_002.json
captures_003.json
egos.json
metric_definitions.json
metrics_000.json
metrics_001.json

+


rgb_2.png
rgb_3.png
rgb_4.png
rgb_5.png
rgb_6.png
rgb_7.png
rgb_8.png
rgb_9.png

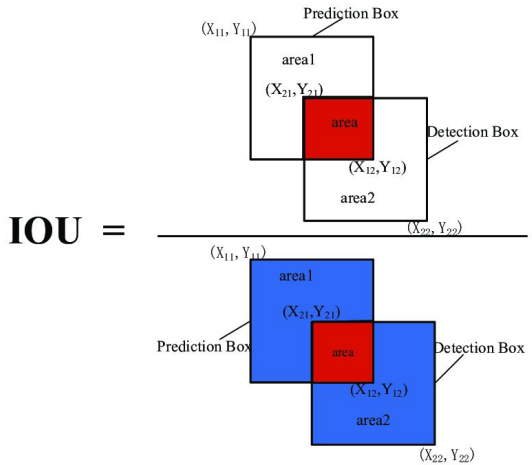
=


rgb_51.png
rgb_51.xml
rgb_52.png
rgb_52.xml
rgb_54.xml
rgb_55.png
rgb_55.xml
rgb_56.png

Understanding Object detection Model lingo

- Our Model Name: SSD_Resnet50_v1_fpn_640x640_coco17_tpu-8
- SSD: Single shot detection [3]
- Resnet 50 is an architecture backbone for feature extraction and overall accuracy of the model.[4]
- V1 is version 1
- FPN is feature pyramid networks for feature extraction [5]
- 640x640 is the input image size for default RGB depth.
- Coco17 is the database name the model has trained upon.
- Tpu-8 is tensor processor unit with 8 bit matrix multiplication.

Evaluation is by mAP@IOU50



Tip:

- Tensorflow object detection model zoo consists many models evaluated with respect their FPS (Frames per second) and mAP (Mean Average Precision).
- Object detection is always a trade off between Accuracy and Speed.
    - Thumb rule: Accuracy and Speed are inversely proportional. (exception cases do exist)

Transfer learning on generated dataset

(Background: The model is trained on COCO dataset of 80 classes i.e. Our model has never seen our custom dataset and has no idea that it exist.)

- The images (300 images) are read and converted to numpy array and then to tensors.
- The category_index file containing class names(10 classes) are read.
- The XMl files consisting annotations are converted to tensors and analytics is obtained.
- The model is created and weights pretrained are loaded.
- A random batch of 8 Tensors is created and fed to predict the output, The gradient is adjusted based on the loss obtained, learning rate.
- The weights are updated are performed to 30,000 epochs.
- The model is a able to predict the image class and bounding box accurately.
- To evaluate mAP, mAP@IOU50, mAP per class metrics are obtained and compared with published paper.

Results:

Fig: Classes Frequency distribution

```
Started reading images.
Finished reading 100 images.
Finished reading 200 images.
Finished reading 300 images.
Finished reading images.
Started reading xml annotations and images as numpy array.
```

Class Name	Frequency count	'%' out of 100
candy_minipralines_lindt	322	10.4342
cereal_cheerios_honeynut	299	9.68892
cleaning_snuggle_henkel	298	9.65651
craft_yarn_caron	309	10.013
drink_greentea_itoen	304	9.85094
drink_whippingcream_lucerne	300	9.72132
lotion_essentially_nivea	322	10.4342
pasta_lasagne_barilla	315	10.2074
snack_granolabar_naturevalley	339	10.9851
snack_biscotti_ghiott	278	9.00843

```
Total Tensors count:3086
Finished reading xml annotations.
```

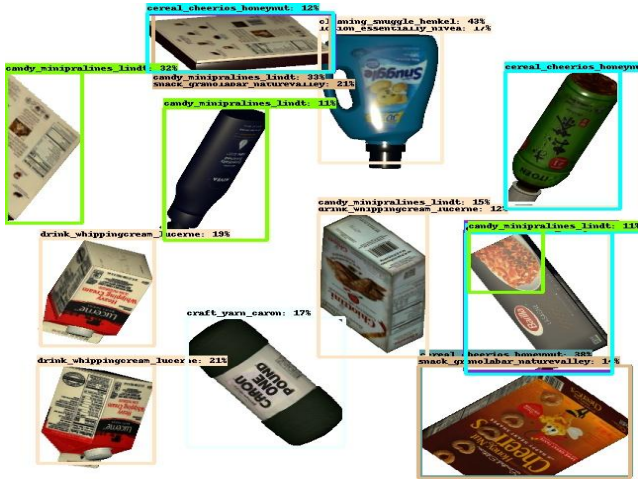
Results

```
batch 0 of 30000,localisation loss=0.05511796474456787,Classification loss=1.2170995473861694 Time=0:00:18.850585
batch 100 of 30000,localisation loss=0.024452072056010365,Classification loss=0.6322769328951836 Time=0:01:03.477460
batch 200 of 30000,localisation loss=0.022426242744550108,Classification loss=0.5583216986060142 Time=0:01:04.571001
batch 300 of 30000,localisation loss=0.0177627265593037,Classification loss=0.5360408356785774 Time=0:01:04.262537
batch 400 of 30000,localisation loss=0.016118463929742574,Classification loss=0.5204154360294342 Time=0:01:04.132708
batch 500 of 30000,localisation loss=0.013328252979554235,Classification loss=0.5287477904558182 Time=0:01:03.741932
batch 600 of 30000,localisation loss=0.012766260006465018,Classification loss=0.4902830070257187 Time=0:01:03.809802
batch 700 of 30000,localisation loss=0.012893615341745317,Classification loss=0.49783828735351565 Time=0:01:03.738663
batch 800 of 30000,localisation loss=0.011847332045435905,Classification loss=0.5089026337862015 Time=0:01:03.857958
batch 900 of 30000,localisation loss=0.010718445279635488,Classification loss=0.46156094878911974 Time=0:01:03.578623
batch 1000 of 30000,localisation loss=0.010891341315582394,Classification loss=0.51002146422863 Time=0:01:03.801699
```

Fg: Training screenshot of 1st 1000 and last 1000 epochs.

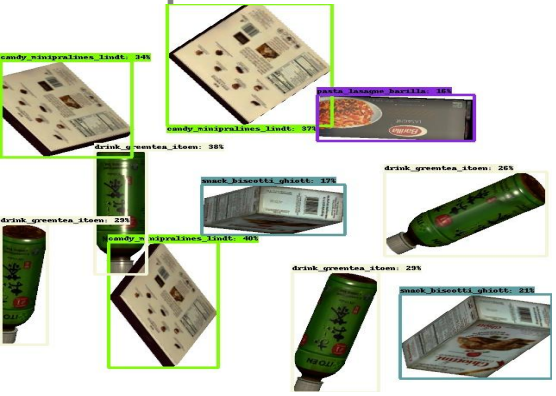
```
batch 29000 of 30000,localisation loss=0.0006888412157422863,Classification loss=0.4760645201802254 Time=0:01:03.856339
batch 29100 of 30000,localisation loss=0.0006717621334246359,Classification loss=0.4890224027633667 Time=0:01:03.774818
batch 29200 of 30000,localisation loss=0.0006819230536348187,Classification loss=0.5200309637188911 Time=0:01:03.757670
batch 29300 of 30000,localisation loss=0.0006819733511656523,Classification loss=0.46236383110284807 Time=0:01:03.789229
batch 29400 of 30000,localisation loss=0.0007108638316276483,Classification loss=0.4630893433094025 Time=0:01:03.748313
batch 29500 of 30000,localisation loss=0.0006665968557354062,Classification loss=0.5152908250689506 Time=0:01:03.786662
batch 29600 of 30000,localisation loss=0.0006399191281525418,Classification loss=0.48789058685302733 Time=0:01:03.812459
batch 29700 of 30000,localisation loss=0.0006808498143800534,Classification loss=0.4545522198081017 Time=0:01:03.832113
batch 29800 of 30000,localisation loss=0.0006149103178177029,Classification loss=0.48272709906101224 Time=0:01:03.817104
batch 29900 of 30000,localisation loss=0.0006486875520204194,Classification loss=0.46494717568159105 Time=0:01:03.727302
Done fine-tuning!
Finished training model
Done.
```

0th epoch



100th epoch

1100th epoch



18500th epoch

24400th epoch

29900th epoch

Results

mAP@IOU90 is at 0.387

mAP@IOU50 is at 0.67

Pasta_lasagne_barilla: 0.46

Cleaning_snuggle_henkel: 0.94

```
Average precision values per class:
candy_minipralines_lindt: 0.649908
cereal_cheerios_honeynut: 0.480704
cleaning_snuggle_henkel: 0.947008
craft_yarn_caron: 0.725478
drink_greentea_itoen: 0.851264
drink_whippingcream_lucerne: 0.860232
lotion_essentially_nivea: 0.645522
pasta_lasagne_barilla: 0.465118
snack_biscotti_ghiott: 0.656190
snack_granolabar_naturevalley: 0.459366
Mean Average Precision @50 is:0.6740790438601992
```

```
Average precision values per class:
candy_minipralines_lindt: 0.321209
cereal_cheerios_honeynut: 0.257714
cleaning_snuggle_henkel: 0.695668
craft_yarn_caron: 0.500510
drink_greentea_itoen: 0.465501
drink_whippingcream_lucerne: 0.371073
lotion_essentially_nivea: 0.338346
pasta_lasagne_barilla: 0.231803
snack_biscotti_ghiott: 0.418296
snack_granolabar_naturevalley: 0.270110
Mean Average Precision @90 is:0.3870230199852739
```

IOU Threshold	Mean Average Precision
25	0.67668
50	0.674079
75	0.66931
90	0.387023

Interesting findings in our case:

- It just takes 10,000 epochs at 8 batches to identify all objects correctly ie equivalent less than 15 mins of training.
- Further training, decreased the localization loss but not classification loss.
- Addition of Hue, distraction Background objects and distraction background textures did affect the accuracy of the model but not too much.

- Ours evaluation stood at .67 and original paper at .75 iou.
- Effect of the **foreground object density** is a real point of consideration as more the object dense lesser the accuracy.
- **Effect of Hue and light** places a significant role in object recognition. Better the variation, better the training.
- **Object size** in individual frame is important as better the object size better the feature extraction
- Elimination of depth in the image resulted in better accuracy.

Disclaimer: Ofcourse 60 classes of original paper and our was 10 classes. They cannot be tangibly compared.

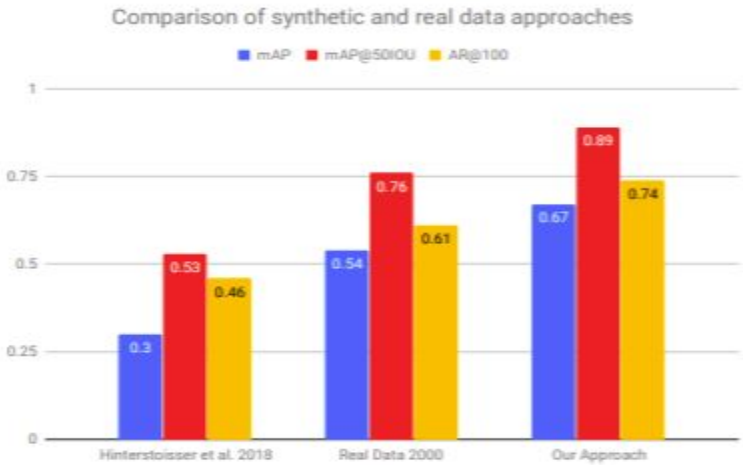


Image source: mAP@IOU50 [2]

Scope to improve

- Limiting Number of tensors repetition.
- Have more classes and more images.
- Try to SSD Mobilnet or YOLO which is better than Faster RCNN.
- Try with background distractions

Thank you.

Any Questions?

Github links

Synthetic data:
https://github.com/Unity-Technologies/com.unity.perception/blob/master/com.unity.perception/Documentation~/Tutorial/Phase1.md

Conversion of annotated data to pascal voc:
https://github.com/Gowtham171996/ConvertUnityPerceptionJSONtoPascalPOCFormat

Training the model:

https://github.com/Gowtham171996/Tensorflow-SSD-Resnet50-Object-Detection

References

1. https://blogs.unity3d.com/2020/06/10/use-unitys-perception-tools-to-generate-and-analyze-synthetic-data-at-scale-to-train-your-ml-models/
2. Hinterstoisser, Stefan & Pauly, Olivier & Heibel, Hauke & Marek, Martina & Bokeloh, Martin. (2019). An Annotation Saved is an Annotation Earned: Using Fully Synthetic Training for Object Instance Detection.
3. SSD: Single Shot MultiBox Detector
4. Convolutional Neural Networks Backbones for Object Detection
5. Feature Pyramid Networks for Object Detection

Demo