Transfer Learning for Object Detection using Deep Convolutional Neural Networks

Project Report
SRIP 2017

By
Jonti Talukdar

Under the Guidance of
Dr. Ravi Hegde

High Performance Computing Lab,
Indian Institute of Technology, Gandhinagar
July 2017

ABSTRACT

Modern day Computer Vision systems require a large amount of data to perform object detection and classification tasks. Since neural networks constitute a key element of these learning systems, it has become increasingly important to increase the dataset capacity to utilize these networks to their full potential. With the development of more complex and efficient computing resources, this data intensive task has been successfully parallelized to achieve very low training times. In such scenarios, the availability of a big enough dataset with both high diversity as well as generalization capability has become the key bottleneck to the training process. Annotation and retrieval of such a dataset manually is a very time consuming and tedious task. Hence generation of synthetic scenes rendered from 3D shape models offer a promising approach to transfer knowledge from synthetic to real domain. In this project, we aim to tackle this problem by generating synthetic scenes and analyzing their performance on real data, focusing on detection of packaged food products in refrigerator scenes. The synthetic images, generated using Blender-Python API, are clustered in scenes with different packing patterns, stacking, as well as a variety of configurations to cater to the diversity of real scenes. The performance of the synthetic dataset on real scenes is then evaluated using a number of state of the art convolutional neural networks (CNNs) as well as optimization performed to achieve high detection accuracy.

Student Details

Name: Jonti Talukdar

Dob: 23 September 1996

E-mail: jonti.talukdar.05@gmail.com /
14bec057@nirmauni.ac.in

Parent Institute: Institute of Technology, Nirma University, Ahmedabad

Department: Electronics and Communication Engineering

Statement of Experience:

My experience as an intern during SRIP 2017 at IIT Gn was phenomenal! The college is located in a peaceful village near the capital city of Gandhinagar and has a very scenic setting. More importantly, it was the people here who really helped me enjoy my internship. The internship was not only focused on technical knowledge but also focused on interpersonal skills. The facilities at IIT Gn are state of the art. With really spacious labs and a spread out academic area, the campus is highly conducive to research activities. Also, the faculty are extremely friendly and easily approachable. Moreover the students and other interns were extremely interesting to talk to, as well as really helpful. SRIP provided me with a great opportunity to not only work on a scientific project of my undertaking but also provided me with a platform to explore my horizons as well as network with likeminded individuals. For this, I will be forever indebted to this college and the coordinators who made it happen. Finally, I would like to thank my guide Dr. Ravi Hegde for providing me with an opportunity to work under him as part of this cutting edge project.

Table of Contents

LIST OF FIGURES

Chapter 1
Introduction

1.1 INTRODUCTION

Increasing automation has expanded the horizons of modern day computer vision systems. With growing dependence of smarter machines on AI (Artificial Intelligence) and machine learning techniques, there has been a subsequent dependence on advanced computer vision technologies. Computer vision in its present form, is not only limited to niche fields like robotics and manufacturing but also in relevant consumer areas like smart refrigerators, home automation, smart sensing, wearable technologies, medical imaging etc.

This paradigm shift in the field of computer vision can be attributed to a few but revolutionary changes in computing trends in the past two decades. The first being advances in computing architecture as well as increase in the clock speed of several state of the art processor designs. Moreover, parallelization of data processing through advanced and dedicated GPUs has also played a critical role in facilitating the data crunching process with ease. The second being the radical use of deep neural network architectures for recognition and detection tasks in computer vision tasks.

One key reason for the success of neural networks is their generalization capability over large amount of data. Earlier methods [1] developed for object detection and recognition tasks relied heavily on the use of feature engineering which were highly dependent on the dataset in consideration. However the use of neural networks has revolutionized and highly automated the process of feature engineering. This is because neural networks, especially CNNs use multiple layers for detecting a specific set of features which are spatially invariant using convolutions over different feature maps. Their independence on hand crafted features as well as their high generalization have led to their large-scale adoption in recent times.

More than 70% of entries in the ILSVRC 2016 ImageNet object classification challenge used CNNs as their base architecture. Since the performance of CNNs rely on specific layer wise characteristics, hyperparameter optimization of such networks promises key improvements on the accuracy during the training and testing phases.

1.2 TRANSFER LEARNING

Neural networks, especially CNNs rely on a large amount of data to reach to their optimum level of accuracy. Hence a dataset with a large number of diverse images, relating to the classification problem at hand will yield a better accuracy and overall performance as compared to a dataset with small number of images.

In such cases, the requirement of a large and diverse enough set of data is important for the complete and full-scale utilization of the neural networks being used. However the annotation as well as collection and retrieval of such a dataset is a key impediment in the overall train-test object detection pipeline. The critical task of dataset generation is thus extremely important to improve the quality of models being trained using deep CNNs and offers a promising approach in improving the quality of training as well as reducing the amount of time in the process.

The approach of generating 3D scenes synthetically and using them to train deep CNNs for testing and application on real images is known as **transfer learning**. The general approach for transfer learning is show in Fig. 1.

Fig. 1: Overall process flow for Transfer Learning including rendering and visualization using 3D object models [2].

We aim at tackling the problem of object detection in real scenes (packaged food in refrigerators) using synthetically generated images. We use Blender Python-API to generate 3D scenes with different packing styles, stacking, object wrapping and configuration by using more than 100 3D models downloaded from the open-source ShapeNet repository [3]. The resultant transfer learning pipeline for object detection is discussed in further detail in the coming chapters.

***

Chapter 2

Deep CNNs for Transfer Learning

2.1 CONVOLUTIONAL NEURAL NETWORKS

As discussed in the previous chapter, the task of Object Detection can be performed using several methods. Early methods used feature engineering, which included the use of handcrafted feature vectors custom to the object under consideration. However, such methods failed to generalize over a vast number of classes as well were limited to a select variety of objects. Hence modern systems rely on neural networks, mostly CNNs to perform deep vision tasks.

Deep convolutional neural networks are a set of feed forward neural networks that have proven very effective in object detection and classification tasks. Deep CNNs have been highly successful in their application for object detection mainly due to their key properties. Fig. 2 shows the general architecture of a deep CNN with a fully connected network at the end.

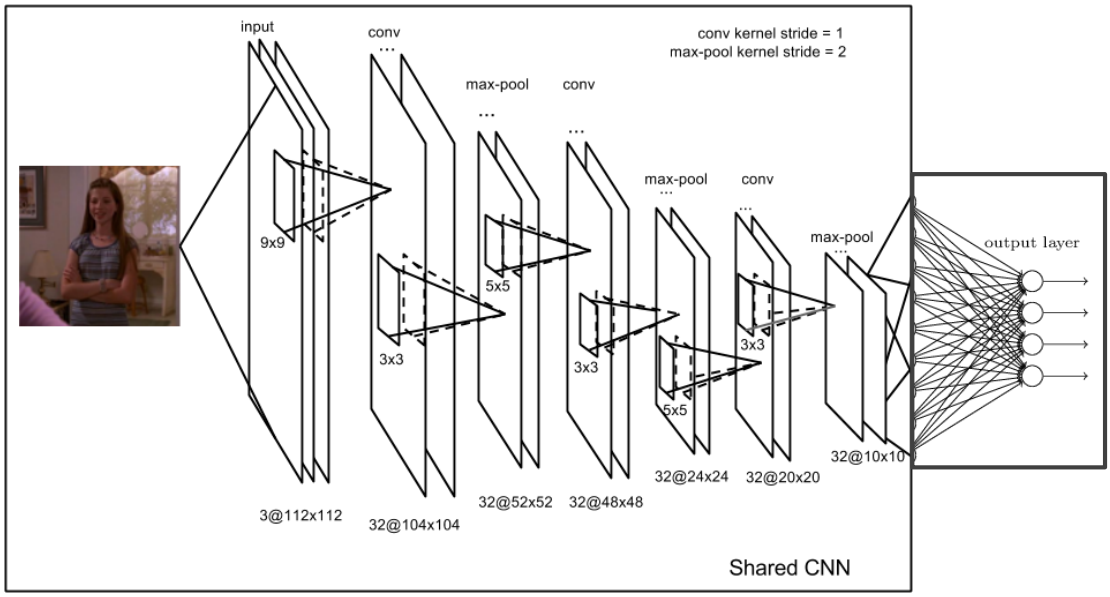

Fig. 2: Overall process flow for Transfer Learning including rendering and visualization using 3D object models [4].

2.1.1 KEY FEATURES OF CNNs

CNNs share some key features which help them in the task of object detection, localization as well as classification. These characteristics are the key differentiating factors which give CNNs a great upper hand in such scenarios. Some of these features are listed below.

1. Sparse Connectivity: Sparse connectivity means that the initial layers of the CNN are not fully connected. In other words, CNNs exploit spatially-local correlation by enforcing a local connectivity pattern between neurons of adjacent layers. This gives rise to receptive fields which are nothing but groups of individual neurons which convolve over the entire image. Sparse connectivity also helps in reducing the overall dimensions of the image. Fig. 3. Shows the size of a receptive field with respect to the other layers deeper in the network.

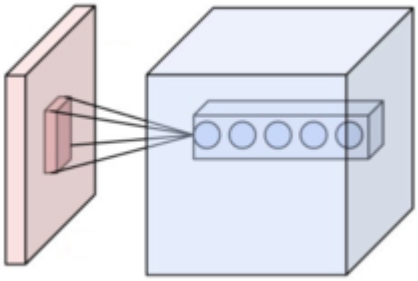

Fig. 3: Neurons of a convolutional layer connected to their receptive fields. [5]

2. Shared Weights: CNNs have a property known as feature maps. Each filter h_i is replicated across the entire visual field. These replicated units share the same parameterization (weight vector and bias) and form a *feature map*. Using shared weights ensures that if a set of features appears across an entire image, then the same can be located in a different region of the overall network. This leads to spatial coherence and invariance.
3. Loss Layers and Pooling: The loss layer specifies how training penalizes the deviation between the predicted and true labels and is normally the final layer. Various loss functions appropriate for different tasks may be used there. Softmax loss is used for predicting a single class of K mutually exclusive classes. Pooling helps in dimensionality reduction and loss layers help in minimizing the detection error.
4. Fully Connected Layer: All CNNs use a single fully connected artificial neural network (MLP) at the end of the final layers for the task of classifying the limited number of classes under consideration. Using the fully connected layer at the end helps in reducing the processing overhead encountered at the beginning of the CNN. It also helps in optimized detection at the end of the network.

The above key features led to the overall success of CNNs as a holistic one shot solution to computer vision tasks like object detection and classification. Further increase in the performance of CNNs is possible using techniques given in the next section.

2.1.2 HYPER-PARAMETER OPTIMIZATION

Hyperparameter optimization is one of the key strategies in enhancing the performance of standard neural networks. A standard CNN has several parameters which can be tweaked to increase their performance as well as generalization capability. Some of the most common hyperparameter settings are given below:

1. Learning Rate: It refers to the rate at which neurons get activated to a certain input of a specific set of features. The learning rate thus helps in deciding the convergence of the overall network. The learning rate should thus be low enough so that the neural network converges to some value but high enough to ensure a reasonable overall training time.
2. Regularization (Weight Decay): In case of small datasets or a large number of iterations, there is a high change of overfitting of data. In those cases, regularization methods help in reducing the chance of overfitting and enhance the generalization capability of such networks.
3. Data Augmentation: Data augmentation is a general method of preprocessing images before training them using any neural network. This step involves performing certain operations on the input image like cropping, contrast variation, resizing, rotation, etc. This method increases the overall variability in the dataset and helps in reducing the redundant instances. The probabilities of several data augmentation tasks can be changed to improve detection results.
4. Annealing Learning Rate: It is the process of choosing the manner in which the learning rate is updated within the overall network as the number of iterations progresses by. The learning rate can be annealed as multistep, inverse exponential, single-step etc.
5. Gamma and Momentum: Since neural nets use gradient descent optimization algorithm to minimize the error function to reach a global minima, the value of gamma and momentum help in deciding on the overall step size of change in the learning rate to ensure that the loss function migrates from a local minimum to a global minimum.
6. Filter Size and Kernel Size: Although this value is fixed based on the architecture in use, the value of stride, also known as kernel size helps in deciding the overall size of features which the CNN detects locally as well as globally over the entire image. Depending on the type of object under consideration, the stride size plays an important role in generalizing and improving the precision of the neural network.

Although all neural networks have a wide variety of parameters, those parameters mentioned above are universal in nature and help in improving detection accuracy.

2.2 STATE OF THE ART CNNs

This project focuses on the overall performance of transferring knowledge from synthetically trained datasets to real scenes. This naturally focusses the choice of neural networks in the direction of those state of the art networks which have been in the winning entries of several object detection and classification challenges like COCO, PASCAL VOC, ILSVRC etc.

Out of the wide ensemble of networks available, this project chose to test its results on a select few, whose description is included in the following sections.

2.2.1 DETECNET: OBJECT DETECTION USING DIGITS

DetectNet [6] is a standard neural network architecture developed by Nvidia for their DIGITS Object detection framework (Deep Learning GPU Training System). It is based on the base GoogLeNet architecture with additional layers for data augmentation as well as evaluation of L1 and L2 regularization losses respectively. The overall network architecture is shown in Fig. 4 below.

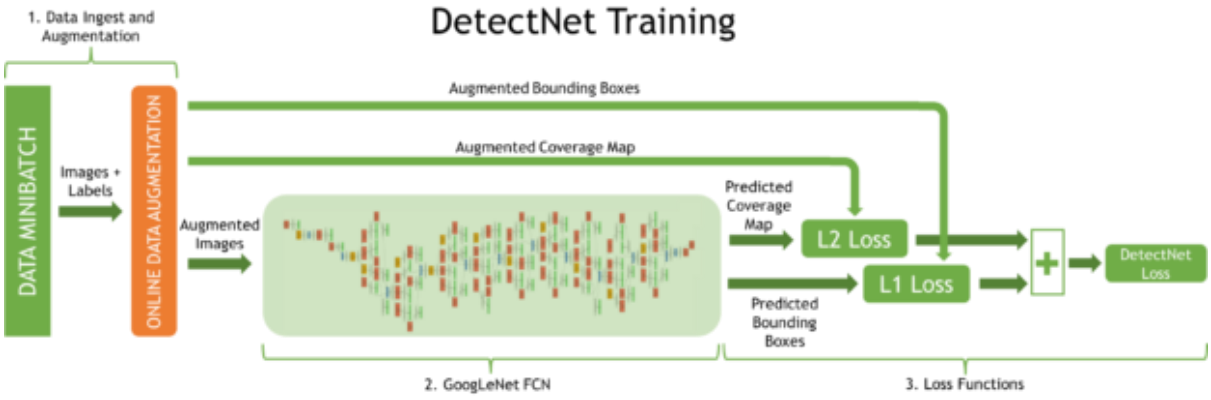

Fig. 4: DetectNet architecture based on GoogLeNet base architecture presented in ILSVRC 2014 [6].

Key features of the DetectNet architecture are shown below:

1. Data layers ingest the training images and labels and a transformer layer applies online data augmentation.
2. A fully-convolutional network (FCN) performs feature extraction and prediction of object classes and bounding boxes per grid square.
3. Loss functions simultaneously measure the error in the two tasks of predicting the object coverage and object bounding box corners per grid square.
4. A clustering function produces the final set of predicted bounding boxes during validation.

5. A simplified version of the mean Average Precision (mAP) metric is computed to measure model performance against the validation dataset.

2.2.2 YOLO

YOLO (You only Learn Once) [7] is a unified model for object detection. YOLO model is simple to construct and can be trained directly on full images. Unlike classifier-based approaches, YOLO is trained on a loss function that directly corresponds to detection performance and the entire model is trained jointly.

The YOLO model focusses on a unified object detection approach which is based on the overall evaluation of bounding boxes for the entire grid of the image and then evaluation of localization as well as confidence. The general approach for YOLO is shown in the figure below.

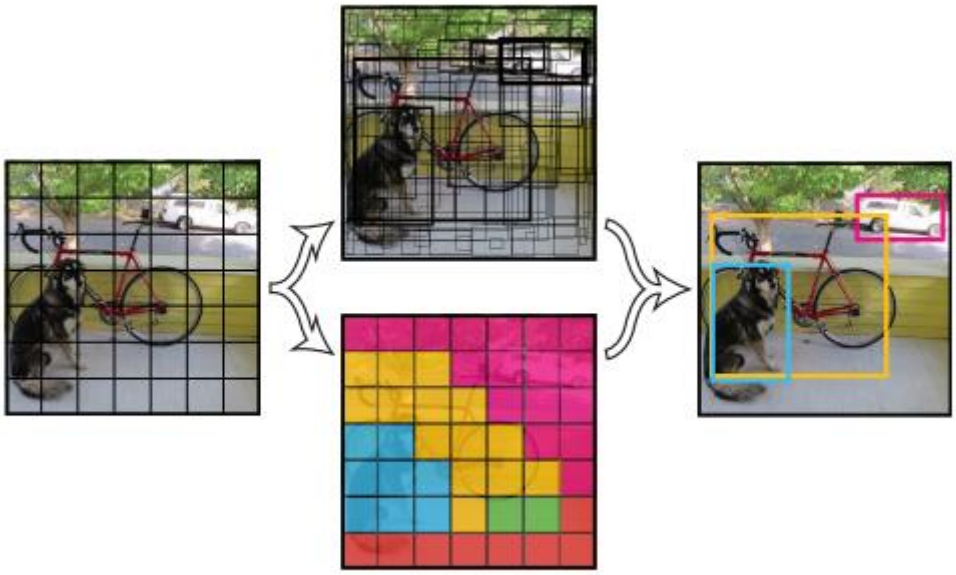

Fig. 5: The entire image is divided into a grid and simultaneously predicts bounding boxes and confidences along with class probabilities, encoded in the form of a tensor [7].

2.2.4 SSD: SINGLE SHOT MULTIBOX DETECTOR

The SSD approach [8] is based on a feed-forward convolutional network that produces a fixed-size collection of bounding boxes and scores for the presence of object class instances in those boxes, followed by a non-maximum suppression step to produce the final detections.

The key difference between SSD and YOLO is that ground truth information needs to be assigned to specific outputs in the fixed set of detector outputs. Once this assignment is determined, the loss function and back propagation are applied endto-end. Training also

involves choosing the set of default boxes and scales for detection as well as the hard negative mining and data augmentation strategies. The overall comparison between the two object detection architectures, SSD and YOLO are given in Fig. 6.

2.2.5 FASTER RCNN

The Faster-RCNN [9] is the final unified object detection system and is composed of two modules. The first module is a deep fully convolutional network that proposes regions, and the second module is the Fast R-CNN detector that uses the proposed regions. A Region Proposal Network (RPN) takes an image (of any size) as input and outputs a set of rectangular object proposals, each with an objectness score. By sharing convolutional features with the down-stream detection network, the region proposal step is nearly cost-free. The use of region proposals allows the network to focus on the necessary scene at hand and improve the accuracy to a high degree.

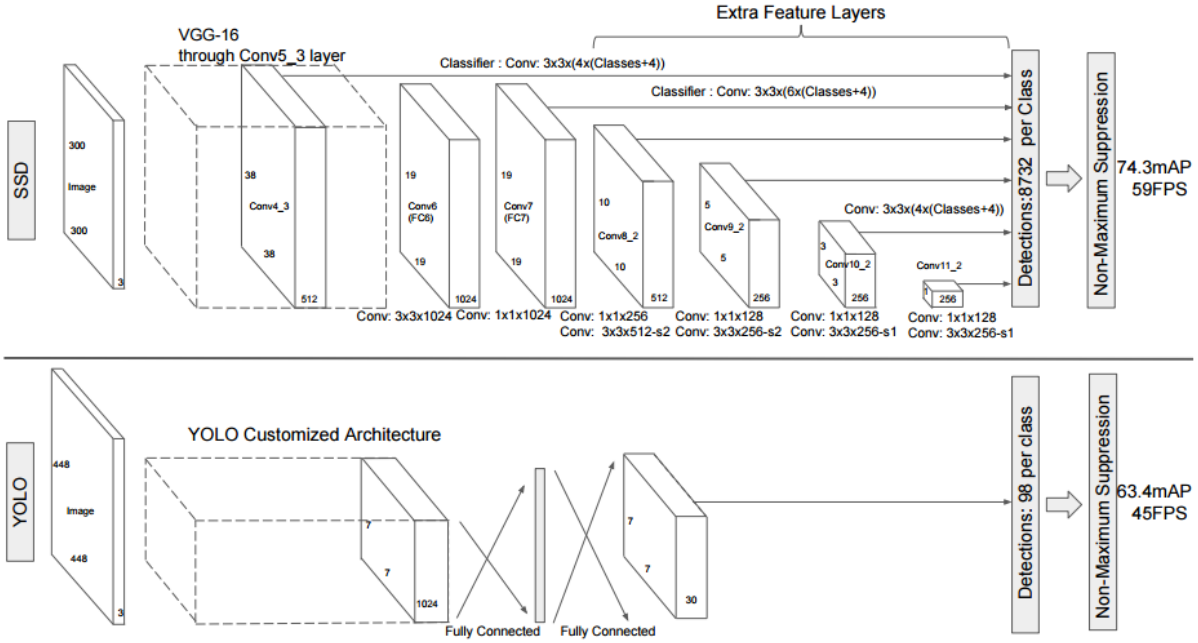

Fig.6: Comparison of SSD and YOLO architectures. SSD model adds several feature layers to the end of a base network, which predict the offsets to default boxes of different scales and aspect ratios and their associated confidences.

***

Chapter 3

Transfer Learning Pipeline

3.1 SYNTHETIC DATA GENERATION

Synthetic images form the key component for training during any transfer learning process. We apply transfer learning strategy to detect packaged food products clustered in refrigerator scenes. We use Blender-Python APIs to load 3D models and automate the process of scene generation. The overall process flow for generation of Synthetic images is given in Fig. 7 below. Fig. 8 shows the wide variety and diversity in scene generation through blender API.

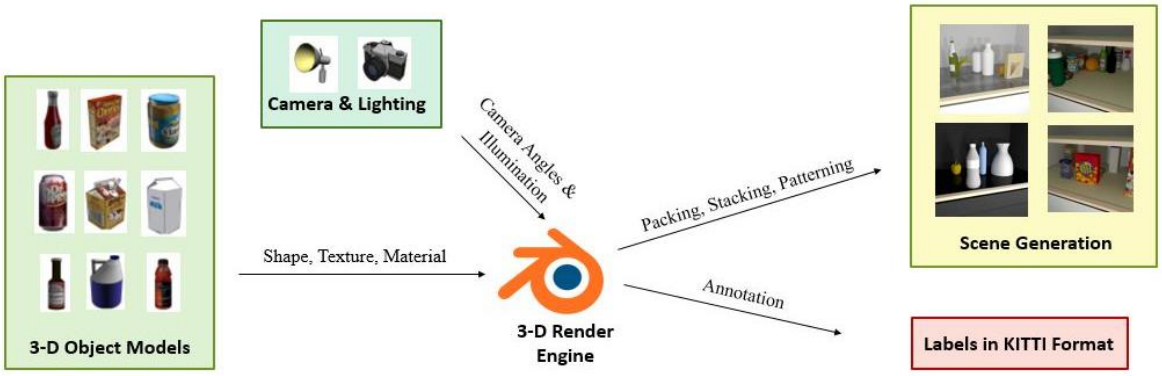

Fig. 7: Synthetic Dataset Generation through Blender, involving Scene Generation through 3D Models and their annotation

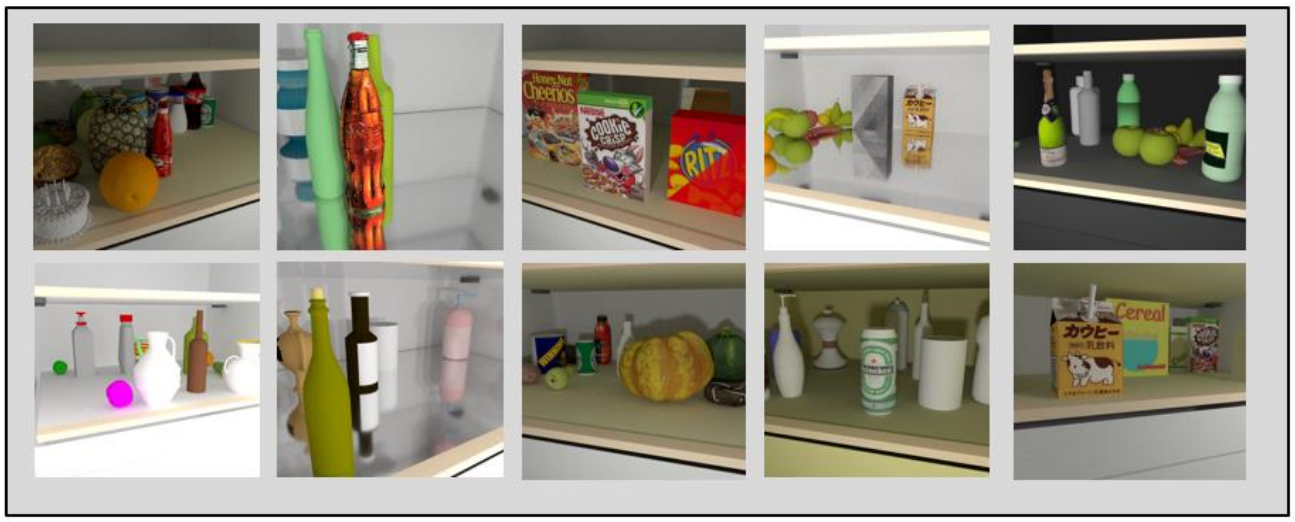

Fig. 8: Synthetic Scenes with different objects, packing, lighting and camera angles, rendered using Blender API for training the transfer learning model.

3.2 SYSTEM ARCHITECTURE

Standard process for transfer learning involves generating a synthetic dataset followed by training it on the desired CNN and ends with validation on a real test set. The overall Object Detection pipeline for transfer learning has been shown in Fig. 9. The following state of the art deep neural network architectures have been evaluated for their performance on Transfer Learning:

1. DetectNet: NVidia DIGITS object detection framework based on GoogLeNet [6].
2. YOLO: You Only Learn Once, Fast Unified Real Time object detection [7].
3. SSD: Single Shot MultiBox Detector [8].
4. Faster-RCNN : Region Proposal based Object Detection [9].

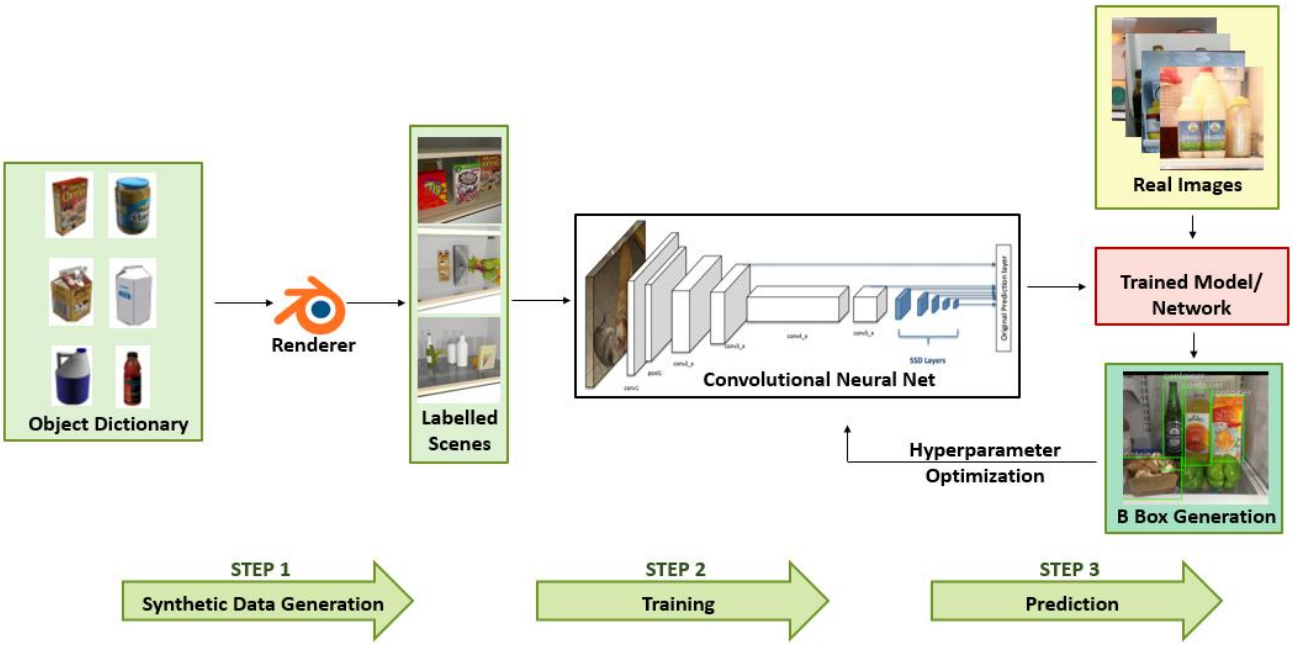

Fig. 9: The standard Object Detection Pipeline for Transfer Learning. Step 1 involves generation of Synthetic Dataset for transfer learning. Step 2 involves the use of a state of the art CNN for training on the synthetically generated dataset. Step 4 involves testing on a real dataset to evaluate network performance and accuracy.

***

Chapter 4

Key Findings & Results

4.1 SYNTHETIC DATASET

It has been observed that a high degree of photorealism is not at all essential for the process of transfer learning. In-fact the time and resources utilized in the generation of photorealistic images can be rather devoted to the manual annotation of a quality dataset. Instead, the key requirements that any synthetic dataset should fulfil include:

- Wide scale diversity to cover all possible scenarios during scene generation.
- Improve the number as well as quality of 3D models imported per scene.
- Dynamic rendering of scenes which cover all possible poses compared to real scenes.
- An equally diverse amount of noise in the form of distractor objects to increase transfer rate of the overall scene.

Every dataset has its own key feature space based on its application domain. Although there is no marked way to define the dataset topology for a certain kind of application, there still exist a few criteria like those mentioned above which increase the likelihood of a good accuracy score due to their 'likeness' to the scenes in real world. Hence it is essential to ensure that the synthetic dataset is **well balanced** to its real life counterpart and is **not necessarily** photorealistic but **diverse** with an equal proportion of noise (distractors).

In our case, other variables which played a decisive factor in the improvement in overall accuracy of detection included:

1. Stacking of objects in synthetic scenes.
2. Addition of distractor objects in synthetic images.
3. Packing of objects in a close knit fashion.
4. Increasing the size of object dictionary to an optimum number.
5. Rotation and translation of objects in 3D space.

4.2 DETECTION ACCURACY

The overall metric which is used for evaluation of objet detection models is known as Mean Average Precision (mAP). It is evaluated as the product of precision as well as recall.

Precision is the ratio of number of true positives to the sum of true positives and false positives. Whereas, Recall is the ratio of true positives to the sum of true positives and false negatives. Hence mAP = Precision * Recall.

The performance analysis of all the 4 state of the art networks, i.e. Nvidia (DIGITS), YOLO and SSD is shown below. Experiments were carried out on workstation with Intel i7-5960X processor accelerated by NVIDIA GEFORCE GTX-1070. Fig. 10 shows the relative performance of the CNN frameworks vs the number of epochs trained.

From Fig. 11 as shown below, we can directly conclude that Faster RCNN has the highest accuracy (mAP) value out of all the other bounding box detectors in consideration. However, SSD is the fastest of all and achieves convergence at a fairly early stage of training.

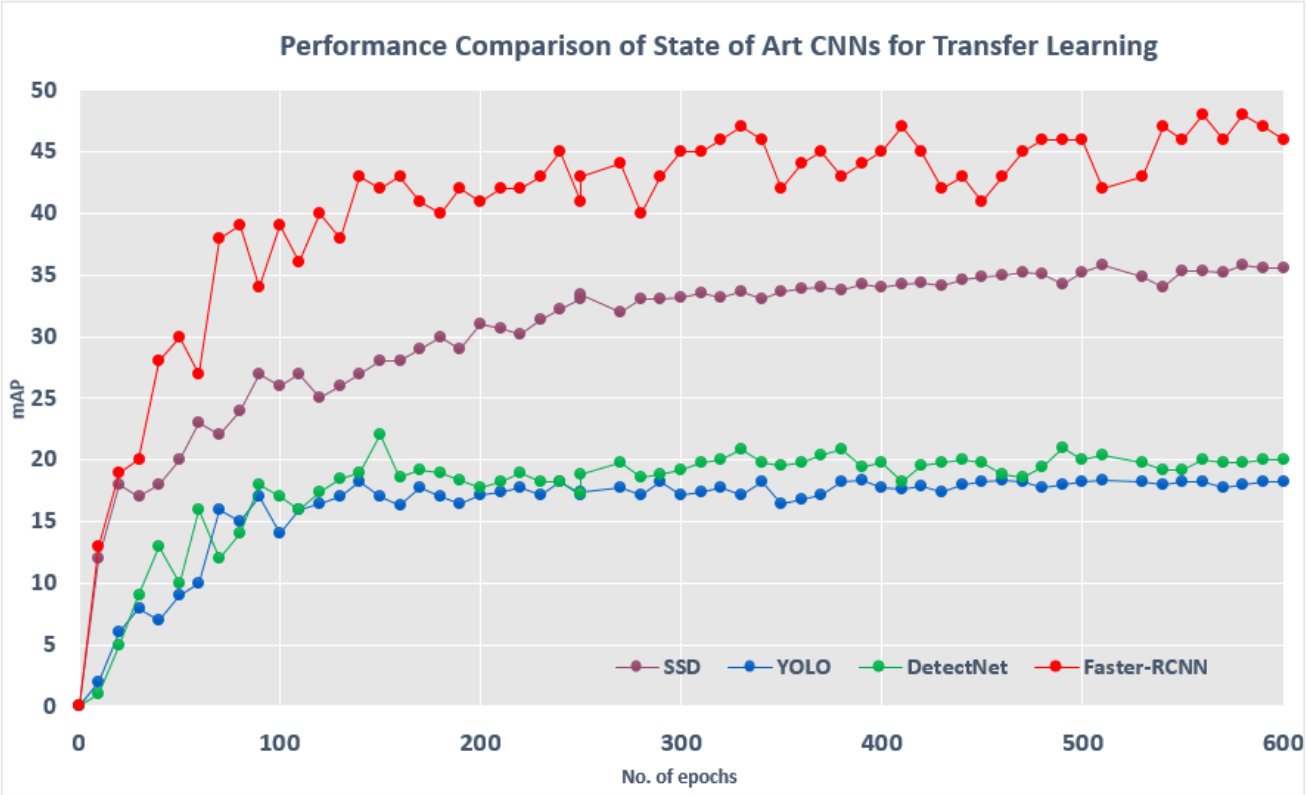

Fig. 10: Performance comparison of SSD, YOLO, DetectNet and F-RCNN showing the mAP vs no. of epochs trained.

mAP Values for different datasets/networks	**DetectNet**	**YOLO**	**SSD**	**F-RCNN**
4k Base Images	17.4	18.3	24.6	37.8
2.5k with Distractor Images	23.6	23.4	34.8	45.6
3.8k Diverse Images	24.8	26.5	47.9	57.8

Fig. 11: Comparative values of mAP for different synthetic data on all the four networks.

4.3 BBOX OUTPUT

The bound box output generated by the SSD network has been shown in Fig. 11 below. The network output consists of a validation set of close to 50 images.

As we can observe, the test results offer a great insight into the importance of using a synthetic dataset which is holistic in nature in terms of both **diversity** as well as **generalization capacity.**

Fig. 11: The output bounding boxes generated by the neural network (SSD) trained on synthetic images.

***

Conclusion

We evaluated the performance of state of the art Convolutional Neural Networks which were trained on synthetic images on real scenes and found that Faster RCNN works with the highest accuracy. However, in terms of speed, both SSD as well as YOLO are comparable with Faster RCNN but have a slight edge in terms of the floating point operations per second. We also narrowed down on the important characteristics which every synthetic dataset should possess in order to optimally transfer knowledge from synthetic to real domain. The overall transfer learning pipeline can further be optimized by narrowing down on the requirement of features specific to the dataset in consideration as well as fine tuning the parameters of the CNN.

***

References

1. S. Khalid et al, “A survey of feature selection and feature extraction techniques in machine learning.” *In Proc. IEEE SAI, 2014.*
2. R. Qui, “Learning 3D Object Orientations From Synthetic Images”, Stanford University, 2015.
3. Shapenet 3D Repository, “https://www.shapenet.org/”, 2017.
4. CS231n, Convolutional Neural Networks, “cs231n.github.io/convolutional-networks/”, 2016.
5. Deep Learning Tutoria, “http://deeplearning.net/tutorial/lenet.html”, 2017.
6. Nvidia DIGITS using DetectNet, “https://devblogs.nvidia.com/parallelforall/deep-learning-object-detection-digits/”, 2017.
7. J. Redmon et al. "You only look once: Unified, real-time object detection." *Proceedings of the IEEE* CVPR (2016).
8. W. Liu et al. "Ssd: Single shot multibox detector." *arXiv preprint arXiv:1512.02325* (2015).
9. Ren, S., He, K., Girshick, R. and Sun, J., 2015. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems* (pp. 91-99).