3D Object Detection in the Wild

Matt Clifford
Supervised by Dr R. Santos-Rodriguez

November 30, 2018

1 Introduction

Object detection is important for computers to understand the real world. For example, helping a self driving car know what objects are in it's planned path. 2D object detection is where objects are classified and located in 2D space, an RGB image. However, since the world lies in 3 dimensions, this 2D simplification of the world isn't enough; if there is a cyclist on the road, the self driving car needs to know accurately where they are, to avoid a collision when overtaking, therefore there is a high demand for accurate 3D detectors. A lot of focus with 3D object detection has been with respect to self driving cars. Although they are in a 'wild' and unpredictable environment, they are confined to roads. This makes 3D object detection systems for them very specific, as they need to follow strict rules, detect specific objects and are trained on days of driving footage captured from expensive and detailed LIDAR sensors.

With recent advancements in mixed reality technology from Google glass and Microsoft HoloLens, there is an interest in the use of mixed reality to help create 'smart spaces'. Where mixed reality users are informed and guided through environments unknown to them. This addresses circulation issues, or aids the visually and/or navigationally impaired.

Fracture Reality [21], have expressed specific interest in 3D object detection. They specialise in creating bespoke mixed reality software for both private and government sectors. They mostly work with mixed reality visualisations of maps of an environment, for example to aid the control centres in airports [22]. Although they have many projects that would benefit from 3D object detection, an ongoing project which is investigating how effective the use of mixed reality is in tackling navigational issues, where fire fighters are aided by a mixed reality headset when entering and exiting new buildings. The use of 3D object detection would identify and locate objects such as stairs, doors and elevators. This would aid the fire fighter in identifying if these objects are correct pathway for them, without them having detailed prior knowledge of the building, since a map can be stored within the mixed reality device.

Fracture Reality are able to create some object specific use case data, but in the region of hundreds of examples due to the expensive and time consuming nature of hand labelling 3D data. This is the main bottleneck in creating effective object detection systems. As well, due

to the specificity of each task, finding existing object datasets in 3D is unlikely. This makes training a detector on 3D data alone for every new object infeasible, especially considering the size of the datasets required in deep learning. 2D data however is more widely abundant and relatively cheap and quicker to obtain, which has potential to aid 3D tasks. The need for a system that can detect objects given as little training examples as possible which also makes use of 2D data is therefore needed. Also, since the only certainty is that the objects are going to be in a real life setting, the 3D object detection must be robust and able to cope in the wild, where object labels might not be clear and can change over time. This can be achieved by utilising general knowledge learnt from similar or relevant tasks and applying it to a new specific task of interest, known as transfer learning [4]. This is where deep neural networks are pre-trained on general tasks where data is sufficiently available to train a robust and general system are then re-purposed and re-trained using data specific for the new task. This works due to earlier layers in networks extracting more general features from the data, which are applicable to many similar tasks.

2 Literature review

2D Image Classification

In 2012, Alex Krizhevsky et al. revolutionised computer vision with a convolution neural network (CNN), inspired by [16]. It performed image classification on the ImageNet dataset [26]. The CNN, named AlexNet, consists of 5 convolution layers followed by 3 fully-connected layers. It won ImageNet's ILSVRC-2010 and ILSVRC-2012 image classification contests [14]. In [14], they claim 'All of our experiments suggest that our results can be improved simply by waiting for faster GPUs and bigger datasets to become available', and since AlexNet's success there have been consistent advancements in CNN state of the art. In 2014, [28] propose a CNN named VGG16, consisting of 16 convolution layers followed by 3 fully-connected layers. VGG16 achieves a top-5 error rate of 7.4% on ILSVRC-2014 compared to AlexNet's 17.0%. [28] achieves this performance boost by using smaller convolution filters, 3x3 compared to AlexNet's 11x11 which decreases the number of weights to train at each convolution layer, alongside a deeper convolution architecture, which can extract deeper relevant semantic meaning from the images. Further performance improvements were made by the use of 'inception models' [32][33], which stack the outputs of several convolutions of the same input, followed by filter concatenation. As well as 'residual learning' [13], which connects the outputs of multiple convolution layer. [31] formulates a combination of 'inception' and 'residual' models. Further improvements to 2D image classification include [17] [41].

2D Object Detection

To identify the location of objects in images, [9] uses a regional CNN (RCNN) model. Regional proposals of the image, which are run though a modified AlexNet. The positive classification results from the regional proposals are then adjusted using a linear regression model to obtain better object bounding boxes. Computational speed ups are proposed in Fast-RCNN [8], which

pools the regional proposals. Further computational speed ups are proposed in Faster-RCNN [24], which combines the selective search regional proposals into the CNN. [23] proposes a grid search grid method rather than the more expensive regional proposals approach, known as 'you only look once' (YOLO). YOLO is faster than Faster-RCNN, making it more suitable for real time application, but it comes at the cost of slightly worst performance.

3D Object Detection

[27] proposes a method of using the YOLO object detection approach in 3D space named Complex-YOLO, using only point cloud data from LIDAR depth sensors. Complex-YOLO uses a Euler-Region Proposal Network which estimates the orientation of objects by adding an imaginary and real part for each proposal box box. This results in 5 times speed up in object detection from the previous state of the art, with on par or better accuracy evaluated on data from KITTI benchmark suit [7]. Other state of the art methods based the KITTI benchmark suite include [38][37][18][40].

Some methods focus on using detailed, accurate point cloud objects as input [3][1][35] from datasets such as [29]. Using this type of model directly is unsuitable for mixed reality, as there is little background variation which makes them not robust enough for detection in the wild.

[20] and [15] use 2D object detectors to aid the regional proposal of the 3D object, by searching only the 3D space in the point cloud occupied from the projected frustum obtained from the 2D object detector. This vastly reduces the search space. Resulting in reduced misclassification, due to the higher accuracy of 2D detection, especially if the 3D object suffers from occlusions or has a sparse representation. Speed is also improved when compared to using point cloud data alone, due to the pre determination of the object class and reduced search space for the 3D detector. An alternative approach could combine 2D object mask detectors[2][12] with the 3D projection of the mask to help further refine the 3D object search space. [25] uses latent support surfaces for 3D object detection on the SUNRGBD dataset. Another notable 3D object detector that use the SUNRGBD dataset is [10].

Transfer Learning

Transfer learning is a well studied area of deep learning [4][19][39], where a network trained for a specific task is re-purposed for a similar task. This is often achieved by truncating the last few layers of a pre-trained network where the network is specific to the trained task, and keeping the starting layers that have more general representations. Since the start of the network is already trained for general tasks relevant to both the original training task and the new desired task, only the last few layers need to be re-trained for the new task. This can be done using considerably less training data than the original network was trained with. Transfer learning could be used to help solve the problem of using as little 3D data as possible to train a 3D object detector. However, when using as little data as possible during the retraining part of transfer learning, the original network needs to be trained for a task as similar as possible as the re-purposed task.

Synthetic Data

[5] proposes a 'cut and paste' style approach to synthesising 2D object detection datasets. First an object mask is predicted for the object, which is then applied to the image to 'cut and paste' the object into background scenes. Occlusions, truncations and blends are then applied to the object, helping it fit more naturally into the scene. This address' the problem of not being able to annotate or collect enough data by hand. [11] shows that training scene detectors on synthetic data produces comparable results on real life tests, with object detectors trained on the SUNRGBD state of the art dataset.

3 Project plan

Week	Task	Relevance to project
3	Formalise project problem	Important to have solid project foundations
4/5/6	Review literature	Will use anything relevant to help with the problem and get a feel for what is realistic
7	Collect/ process dataset(s)	Essential for training any system. Making a good pipeline for data enables fast and affective research
8	CNN as baseline for image classification	Indication of how good the dataset is and how well object detection can work
9	Write interim report	
10/11	fast-RCNN or YOLO for 2D object detection	Will be used to help 3D detection in frustum method and transfer learning from 2D to 3D
12/13	Baseline 3D object detector using point cloud data/ start making project presentation	Indication of how well 3D object detection can work with full amounts of data
14/15	Encorporate 2D data using frustum method	Make 3D detector better with 2D data
15	Explore whether masks of 2D object further improve frustum method	Make 3D detector better with 2D data
16	Explore how many objects are needed for the transfer learning of new 3D objects	Use as little 3D data as possible to detect new objects. This is what will happen when Fracture is presented with a new project
17/18	Transfer learning of 2D detection to 3D space/ find common representation/ spend some time writing draft section	Use as little 3D data as possible, and utilise available 2D data
19	'Cut and Paste' data generation in 3D	Use as little real 3D data as possible
20/21/22/EV1	Finalise report/ poster and allow for some contingency time	

4 Progress

Datasets

The KITTI benchmark suit [7] is an autonomous driving dataset with 200,000 3D object annotations captured in cluttered scenarios, with up to 15 cars and 30 pedestrians in each image. The data is obtained from a stereo camera and LIDAR sensor mounted on top of a car that is driven in the real world. Although the KITTI benchmark suit is a rich 3D object dataset, it is not as directly applicable to mixed reality application due to the sensing quality differences between LIDAR and the portable depth camera used in mixed reality. As well as KITTI only

focusing on 8 autonomous driving classes such as pedestrians, cars and bicycles.

[29] is a large-scale 3D object dataset with 32040 object poses and 45 different objects. The point cloud data is triangulated from 11 different views, making highly detailed scenes. The scenes are controlled and do not represent what would be captured from mixed reality depth sensors due to the triangulated different views.

SUNRGBD benchmark suite [30] is a 3D object dataset consisting of 10,335 images with 64,595 3D object bounding boxes. The data is collected on various portable RGBD cameras such as the Kinect device, with indoor scenes focusing on objects such as doors, tables and chairs. A similar quality popular dataset is the Pascal Visual Object Classes (VOC)2012 [6], which consists of 11,530 annotated object images, indoors and outdoors with 20 classes such as chairs, cars, dogs. However, PASCAL VOC 2012 only consists of 2D data. [36] extends the PASCAL VOC 2012 dataset with proposed 3D CAD style projections of the 2D objects. Although rich, this 3D data of the object is dissimilar to that of a depth sensor, leaving SUNRGBD the most suitable starting dataset for this project.

To easily access relevant parts of the dataset a pipeline is made; RGD image, depth image, point cloud, 2D labels and bounding box and 3D labels and bounding box. This data pipeline also includes cropping images to the objects within them, for training an image classifier, and also includes taking a subset of object labels and a subset of these images. This is important, as it simplifies the problem when testing which methods are feasible, before training on the whole dataset. Figure 1 shows a couple of example images with 2D bounding box annotations from the SUNRGBD dataset.

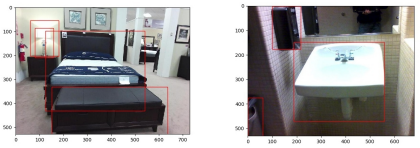

Figure 1: Examples of SUNRGBD datasets RGB images with 2D annotations in red.

Figure 2 shows some examples of objects from the SUNRGBD data set where the bounding box labels are inaccurate or only partially cover the object. This could be a weakness of any system trained using bad data and there could be potential into cleaning the dataset to address this.

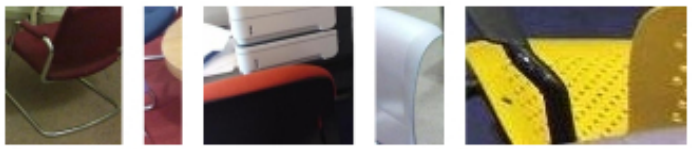

Figure 2: Cropped examples of 'chairs' in the SUNRGBD dataset. Showcasing where bad labelling can occur using the 2D annotations.

Image Classification of SUNRGBD objects

An image classifier is trained on a subset of object classes. As the easiest task, image classification gives an indication of the quality of the dataset and it's annotations, as well as giving an upper bound on what accuracy any object detectors can achieve. Images in figure 2 and similar are included in the training and validation.

The state of the art InceptionV3 [34] CNN architecture is used to demonstrate image classification on a subset of the SUNRGB dataset. 500 examples from the classes: chair, door and table are used. In the SUNRGBD dataset, images contain multiple of the same class objects, for example a room full of the same chair. To give a more general representation of objects and the whole dataset, only one object from each image is sampled in a random order of the images. To reduce training time and improve accuracy, weights pre-trained on ImageNet are used. Since ImageNet has 1000 classes, the feature extraction from the network is very general and should apply well to this similar classification task. Transfer learning of the pre-trained InceptionV3 is achieved by truncating the final classification layer and replacing it with the new desired classes. Re-training the weights is achieved by back propagation of the whole network, with batch normalisation and using the cross-entropy loss function. Figure 3 shows the accuracy of training and validation sets during transfer learning. The model achieves around 94% accuracy on the validation set. This is very high, and is as expected since the task is very similar to the ImageNet classification task the model was pre-trained on.

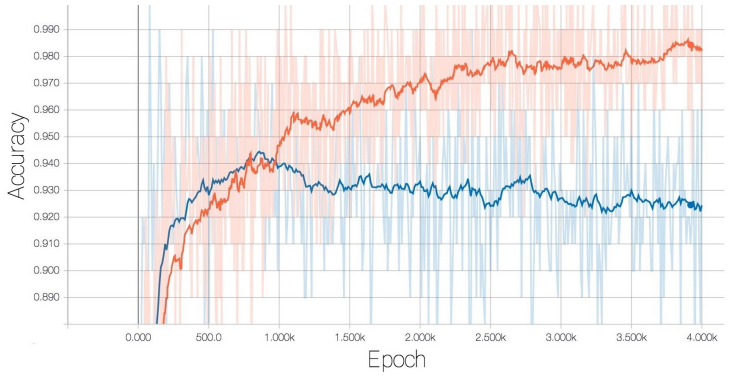

Figure 3: Accuracy for train set (orange) and validation set (blue) for transfer learning of InceptionV3 with weights pre trained on ImageNet, using 500 examples of chair, door and table each from SUNRGBD dataset to re-train. Both curves are smoothed, with the lighter background colour representing the actual data. Training is very quick. Training and validation accuracies start at around 30% (cropped from graph for scale issues). The model starts to overfit after around 800 epochs, where the validation accuracy is seen to decrease from it's peak just above 94% accuracy.

References

[1] V. Balntas, A. Doumanoglou, C. Sahin, J. Sock, R. Kouskouridas, and T.-K. Kim. Pose Guided RGBD Feature Learning for 3D Object Pose Estimation. *ICCV*, 2017.

[2] G. Bertasius, H. S. Park, S. X. Yu, and J. Shi. Unsupervised Learning of Important Objects from First-Person Videos. *ICCV*, 2017.

[3] A. G. Buch, L. Kiforenko, and D. Kraft. Rotational Subgroup Voting and Pose Clustering for Robust 3D Object Recognition. *ICCV*, 2017.

[4] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition. *PMLR*, 2013.

[5] D. Dwibedi, I. Misra, and M. Hebert. Cut, Paste and Learn: Surprisingly Easy Synthesis for Instance Detection. *ICCV*, 2017.

[6] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

[7] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? The KITTI vision benchmark suite, 2012.

[8] R. Girshick. Fast R-CNN. *ICCV*, 2015.

[9] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *CVPR*, 2014.

[10] J. Guerry, A. Boulch, B. Le Saux, J. Moras, A. Plyer, and D. Filliat. SnapNet-R: Consistent 3D Multi-View Semantic Labeling for Robotics. *ICCV*, 2017.

[11] A. Handa, V. PatrAucean, V. Badrinarayanan, S. Stent, and R. Cipolla. Understanding Real World Indoor Scenes With Synthetic Data. *CVPR*, 2016.

[12] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN. *ICCV*, 2017.

[13] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. *CVPR*, 2015.

[14] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. *NIPS*, 2012.

[15] J. Lahoud and B. Ghanem. 2D-Driven 3D Object Detection in RGB-D Images. *ICCV*, 2017.

[16] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proc. IEEE*, 1998.

[17] C. Liu, B. Zoph, M. Neumann, J. Shlens, W. Hua, L.-J. Li, L. Fei-Fei, A. Yuille, J. Huang, and K. Murphy. Progressive Neural Architecture Search. *CoRR*, 2017.

[18] W. Luo and R. Urtasun. Fast and Furious: Real Time End-to-End 3D Detection, Tracking and Motion Forecasting with a Single Convolutional Net. *CVPR*, 2018.

[19] S. J. Pan and Q. Yang. A Survey on Transfer Learning. *TKDE*, 2010.

[20] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas. Frustum PointNets for 3D Object Detection from RGB-D Data. *CVPR*, 2018.

[21] F. Reality. Company website. `fracturereality.io`.

[22] F. Reality. Hololens — airport command and control centre. `https://tinyurl.com/`

yau4cj88.

[23] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You Only Look Once: Unified, Real-Time Object Detection. *CVPR*, 2016.

[24] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *NIPS*, 2015.

[25] Z. Ren and E. B. Sudderth. 3D Object Detection with Latent Support Surfaces. *CVPR*, 2018.

[26] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 2015.

[27] M. Simon, S. Milz, K. Amende, and H.-M. Gross. Complex-YOLO: An Euler-Region-Proposal for Real-time 3D Object Detection on Point Clouds. *CoRR*, 2018.

[28] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR*, 2014.

[29] T. Solund, A. G. Buch, N. Kruger, and H. Aanas. A Large-Scale 3D Object Recognition Dataset, 2016.

[30] S. Song, S. P. Lichtenberg, and J. Xiao. SUN RGB-D: A RGB-D Scene Understanding Benchmark Suite. *CVPR*, 2015.

[31] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. *AAAI*, 2016.

[32] C. Szegedy, W. Liu, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *CVPR*, 2015.

[33] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the Inception Architecture for Computer Vision. *CVPR*, 2015.

[34] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the Inception Architecture for Computer Vision. *CVPR*, 2016.

[35] B. Tekin, S. N. Sinha, and P. Fua. Real-Time Seamless Single Shot 6D Object Pose Prediction. *CVPR*, 2018.

[36] Y. Xiang, R. Mottaghi, and S. Savarese. Beyond PASCAL: A benchmark for 3D object detection in the wild. CVPR, 2014.

[37] B. Xu and Z. Chen. Multi-Level Fusion based 3D Object Detection from Monocular Images. *CVPR*, 2018.

[38] D. Xu, D. Anguelov, and A. Jain. PointFusion: Deep Sensor Fusion for 3D Bounding Box Estimation. *CVPR*, 2018.

[39] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? *NIPS*, 2014.

[40] Y. Zhou and O. Tuzel. VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection. *CVPR*, 2018.

[41] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le. Learning Transferable Architectures for Scalable Image Recognition. *CVPR*, 2018.