Centre for Digital Built Britain

2018/19 General Project

Open ML Training Data For Visual Tagging Of Construction-specific Objects (ConTag)

Final Reporting

10/07/2019

Dr J Boehm	University College London (UCL)
	j.boehm@ucl.ac.uk

Abstract

ConTag has generated open datasets for visual machine learning (ML) specific to the construction industry. ML technology has enabled a revolutionary leap in many digital economies generating growth in activity and business mainly for the ITC sector. Part of the growth is generated through sharing of IP, knowledge, tools and datasets. We want to adopt this approach for the digital construction sector. ConTag provides visual and 3D training datasets for training deep neural networks (DNNs) and provides weights for pre-trained networks. The research output is to support visual tagging of assets from reality capture data. Such automatically generated semantic information can be used to generate or populate digital twins in the example scenarios. The first dataset is a collection of fire safety equipment typically found in indoor environments. The dataset contains the classified images, per-pixel label images and bounding box data for object detection. The second dataset is a synthetic 3D point cloud of an outdoor urban street scenario. The dataset contains the point cloud data and per-point label data. We expect this shared and open datasets to kick-start further ML developments in both academia and industry. It is intended as a seed point for collaborative research.

Main Text

One of the fundamental requirements for supervised deep learning are large, accurately labelled datasets. For this reason progress in two-dimensional (2D) image processing is often largely accredited to the wealth of very large, high quality datasets such as ImageNet [1] (classification), COCO [2] (object detection) and Pascal VOC [3] (segmentation). It is now common practice to pre-train Convolutional Neural Networks (CNN) on large datasets before fine-tuning on smaller domain specific datasets. The outcome of this project are two such domain specific datasets relevant to the construction sector. As distinguishing feature our indoor dataset directly links the objects to their respective Uniclass, the UK construction sector's classification scheme to name objects. This is in line with BS 1192-4 (and follow-up ISO standards) which are part of the BIM level 2 suite of documents. Our indoor 2D dataset will be available for download at http://www.firenet.xyz.

Despite the large success of deep learning for 2D image processing, it is evident that automatic understanding for three-dimensional (3D) point cloud data is not as mature. We argue one of the reasons for this is the lack of training data at the scale of that available for 2D data. A key reason for the lack of 3D training data is that naturally the amount of data decreases as the complexity of labelling increases. For example, in 2D, single image classification (i.e. dog, car, cup etc.) is generally trivial and can therefore be carried out by large communities of untrained workers. Object detection requires more skill and has an added level of subjectivity. Segmentation again requires further precision, delicacy and involved more subjectivity. Per-point 3D segmentation requires highly skilled users and generating perfect labels for even the most advanced users is non-trivial. A potential solution to account for this is to synthetically generate training data (i.e. ShapeNet [4]). The primary purpose of our dataset is therefore to offer an open dataset to aid further research assessing the potential of synthetic datasets for pre-training Deep Neural Networks (DNNs) for automatic point cloud labelling. We believe successful progression in this area could have potentially huge implications of the future of automatic point cloud labelling. Our 3D outdoor dataset will be available for download at http://www.synthcity.xyz.

Indoor Scenario

The intention for generating the indoor dataset is to produce a visual recognition training dataset that is relevant to applications in the construction domain. We have chosen fire safety equipment as an example scenario. The dataset provides training data for deep neural networks (DNN) for the three typical tasks of classification, detection and segmentation. The three tasks have increasing complexity. For classification a label (or tag) is applied to the whole image. For object detection the location of the object is indicated by a bounding box. For segmentation the exact mask of the object is given in a label image. See Figure 1 for an illustration of the three tasks.

Figure 1: Image (with label), object with bounding box and segmented object (left to right).

From the potential object categories, or classes, that the Uniclass classification scheme [5] contains for fire safety equipment, we chose a sub-set of classes, which is shown in Table 1. The criteria for selecting classes are mostly in their visible distinction. Installed equipment needs to be visible in the public domain thus only objects in the 'product' branch of Uniclass are chosen. Classes form the 'Systems' branch of Uniclass for example are typically collections of physical objects grouped together by a certain function they provide. This is not suitable for visual recognition. We favor number of examples over number of classes as the parameters of modern DNN architectures are growing. This means that many examples are need for an individual class to be learned properly. Where classes are visually very similar they are grouped into one class. This is for example the case for Pr_40_50_28_64 (Portable fire extinguishers). Other types in the same sub-category include foam fire extinguishers (Pr_40_50_28_30), dry powder fire extinguishers (Pr_40_50_28_24), etc. The actual sub-type of fire extinguisher cannot be determined visually. A similar situation exists for smoke detectors.

Table 1: Selected object classes for FireNet.

Uniclass	Technical Description	Non-technical Description
Pr_40_50_28_64	Portable fire extinguishers (and similar)	Fire Extinguisher
Pr_75_75_30_50	Manual call points	Alarm Activator
Pr_40_50_28_29	Fire protective blankets	Fire Blanket
Pr_40_10_77_32	Fire escape route signs	Fire Exit Sign
Pr_40_10_77_31	Fire equipment signs	Fire Suppression Sign
Pr_75_75_30_97	Visual alarm signal devices	Flashing Light Orb
Pr_75_75_30_30	Fire alarm sounders	Sounders
Pr_75_75_30_65	Point smoke detectors (and similar)	White Domes

For data collection in the case of images ‘web scraping’ or ‘web harvesting’ is a popular approach. Web search engines such as Google image search are used to find images that meet the desired object category. The advantage of web harvesting is that one can get a global distribution of images and is thus not restricted to one’s locality. In practice however the dominant image source is North America. Furthermore, using image search to retrieve examples in fact relies on the images being ‘tagged’, or in other words the classification problem has to have been solved by the provider already. This severely limits the number of available useful images as soon as the categories become more specialized. In this case useful means images that clearly show the object, have an appropriate license for reuse and are not catalogue images, i.e. the foreground object is not extracted and set in front of a blank background. As an example for the category ‘fire extinguisher’ we are immediately able to retrieve 178 useful examples from the web, whereas for ‘fire blanket’ only 21 images are retrieved.

For this project we have therefore collected most data by acquiring new pictures with mobile phone cameras. Images were collected and assigned to the respective classes. Images which contained objects of multiple categories were assigned to multiple categories. With respect to the selected classes we were able to collect 243 example images for fire extinguishers, 294 for alarm activators, 215 for fire blankets, 271 for fire exit signs, 247 for fire suppression signs, 33 for flashing light orbs, 268 for sounders and 281 for white domes. This shows that most categories are sampled with well over 200 example images. However, flashing light orbs (visual alarm signal devices) are underrepresented. This creates a class imbalance that must be taken into account during training.

The labelling of the images (delineating the objects in the images) was crowd sourced. We used the dominating Amazon Mechanical Turk (MTurk) platform for this task [6]. MTurk provides templates for a web based user interface to complete the task. An example of this user interface is given in Figure 2. With the large user base of MTurk the task usually completes with a very short period of time. For batches of about 200 images the task completed in less than two hours. In our case we set a price for the HITs to $0.1.

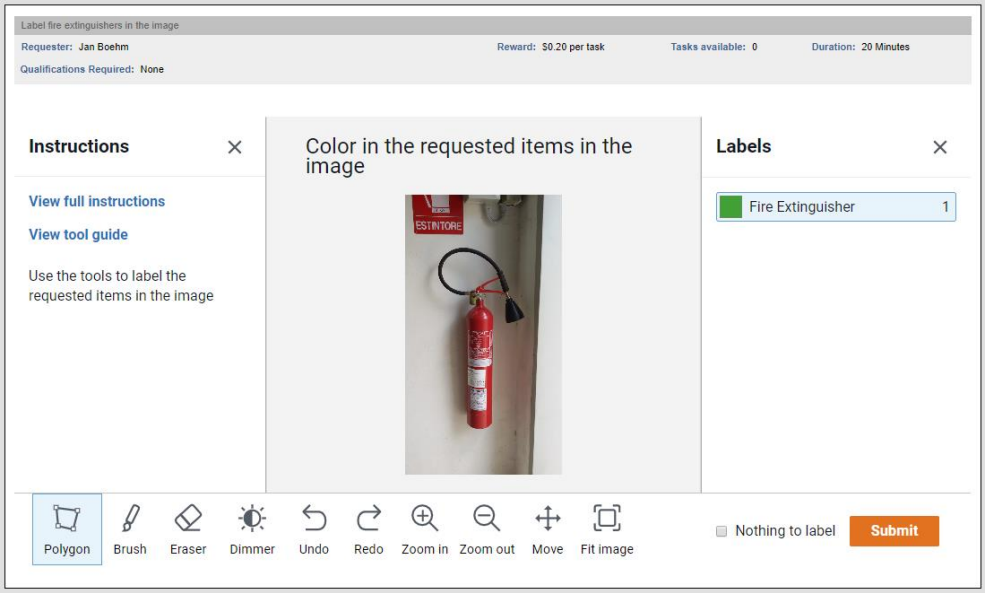

Figure 2: Example Human Intelligence Task (HIT) for segmenting an object of a given class in an image.

Table 2: Statistical summary for crowd sourced labels.

Class	Images	Semantic Segmentation	Object Bounding Box	Unused
Fire Extinguisher	243	160 (66%)	210 (86%)	30 (12%)
Alarm Activator	294	96 (33%)	248 (84%)	46 (16%)
Fire Blanket	215	135 (63%)	181 (84%)	34 (16%)
Fire Exit Sign	271	163 (60%)	247 (91%)	24 (9%)
Fire Sup-pression Sign	247	185 (75%)	207 (84%)	40 (16%)
Flashing Light Orb	33	32 (97%)	32 (97%)	1 (3%)
Sounders	268	190 (71%)	262 (98%)	6 (2%)
White Domes	281	211 (75%)	275 (98%)	6 (2%)
Total	**1852**	**1172**	**1662**	

However, as the labels are crowd-sourced they need to be verified. We developed Python scripts to directly display and visually verify the results from Amazon MTurk. Verifying a correct labelling usually takes less than 10 seconds per image. This is significantly less than marking the objects in the image interactively. Furthermore, as mentioned above due to the large user base in MTurk the labelling task is massively parallelized. We could not have achieved this with a small team in-house. Analysis clearly shows that crowd-sourced labelling was not successful for all images. We typically went through two iterations for crowd sourcing. Tasks that were rejected in the first iteration are re-submitted for a second iteration. Thus, individual tasks can be submitted twice and the number of HITs issued is therefore larger

than the number of images. The remaining rejected tasks are discarded. Often these are difficult images where the object is hard to identify or only partially visible. We show a summary of the results for crowd-sourcing in Table 1. Some of the labelling is useless for our tasks and the results need to be rejected. This is represented in the last column. Fully successful labelling is shown in the third column. Some labelling result that were not accurate enough for semantic segmentation can still be useful to generate a bounding box, which is reflected in the fourth column.

FireNet has been designed as a ML training dataset for experimentation and therefore fulfills multiple machine learning scenarios (classification, object detection, sematic segmentation). The dataset itself is not big enough to train a modern DNN from scratch. It is intended as a domain specific dataset to refine pre-trained standard architectures. There is a remaining class imbalance and a small number of images that were not successfully labelled. As part of a continuous maintenance to the dataset we are exploring options to revisit the remaining images.

Outdoor Scenario

The primary aim in generation our outdoor dataset is to produce a globally registered point cloud where each point $\mathbf{P} \in R^{n\times 3}$. Additionally, each point **P** contains a feature vector $F \in R^{n\times d}$ where n is the number of points such that $n = 367.9M$ and d is red, green, blue, time, end of line, and label l where $l \in L$ such that $|L| = 9$. The SynthCity data was modelled inside the open-source Blender 3D graphics software. The initial model was downloaded from an online model database. The model was subsequently duplicated with the object undergoing shuffling to ensure the two areas were not identical to one another. Road segments were duplicated to connect the two urban environments leaving large areas of unoccupied space around the roads. To populate these areas additional typical suburban building models were downloaded and placed along the road. With respect to model numbers the dataset contains: 130 buildings, 196 cars, 21 natural ground planes, 12 ground planes, 272 pole-like objects, 172 road objects, 1095 street furniture objects and 217 trees (Table 3). The total disk size of the model was 16.9GB. The primary restriction for the size of the dataset was the availability of random access memory (RAM) required on the interactive gaming workstation. This was limited to 32GB in our case, however, with a larger RAM the model size could have easily been extended.

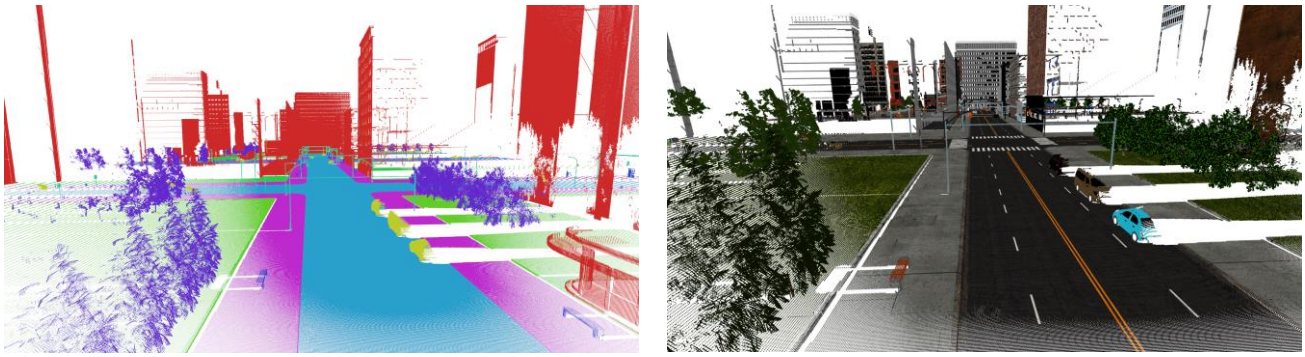

Figure 3: Example of the SynthCity dataset displaying class labels (left) and RGB values (right).

The open-source Blender SensOr Simulation plugin BlenSor [17] was used for simulation of the MLS and thus point cloud generation. We use the following setup for scanning:

Scan type	Generic LIDAR
Max distance	100 m
Angle resolution	0.05 m
Start angle	−180 degree
End angle	180 degree
Frame time	1/24 s

A typical scan took ∼330 s to render and a total of 75,000 key frames were rendered from a pre-defined trajectory. To increase realism and generate more variability in point density the trajectory spline was moved by a random permutation at random intervals in all *x,y,z* directions. The final rendering required (330 × 75000) / 86400 = 286.46 days CPU compute time. This was processed using AWS cloud computing service. We launched 22 type r4.2xlarge Ubuntu 18.04 EC2 spot instances, each containing 8 virtual CPUs and 61GB RAM. These were selected as rendering typically required <50GB RAM. All data was read and written to a EFS file storage system to allow for joint access of a single model instance. The total rendering time took ∼13 days on 22 EC2 instances (4,500 hours CPU time).

We choose to store our data in the parquet data format. The parquet format is very efficient with respect to memory storage but is also very suitable for out-of-memory processing. The parquet format is created by Apache designed to integrate with the Apache Spark ecosystem. It can be directly read into python Pandas dataframes but also Python Dask data frames which allow for easy out-of-memory processing directly in the python ecosystem.

The dataset is modelled from a completely fictional typical urban environment. In reality the environment would be most similar to that of downtown and suburban New York City, USA. This was due to the initial starting model. Other buildings and street infrastructure are typical of mainland Europe. We classify each point into one category from: road, pavement, ground, natural ground, tree, building, pole-like, street furniture or car. To address the class imbalance issue, during construction of the model we aimed to bias the placement of small less dominant features in an attempt to reduce this as much as possible. As point cloud DNNs typically work on small subsets of the dataset we argue that this approach should not introduce any unfavourable bias, but instead help physically reduce the class imbalance.

Category	No. Models	No. Points
Road	172	215,870,472
Pavement	172	21,986,017
Ground	12	6,206,312
Natural ground	21	4,788,775
Tree	217	12,088,077
Building	130	97,973,820
Pole-like	272	1,636,443
Street furniture	1095	1,469,766
Car	196	5,907,347
Total	2287	367,927,029

Table 3: Label categories and number of points per category.

The total number of points generated is shown in Table 3. The total area of the site is ~0.3 km^2. The disk space of the complete parquet file is 27.5GB, as a typical work station would not be able to load this model into memory, we split this scan into 9 sub areas. Each sub area is split solely on horizontal coordinates and can therefore contain points from any scan at any key frame.

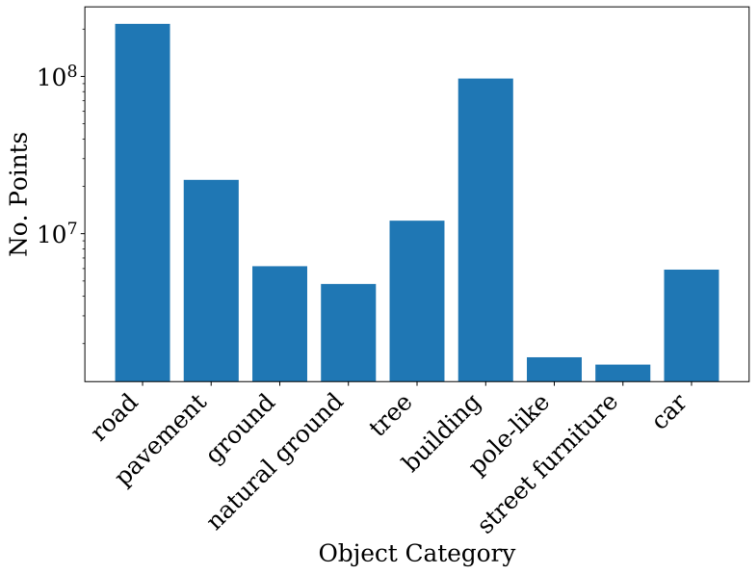

Figure 4: Total point counts for each label category. Note the log y-axis scale.

Although SynthCity was modelled to be biased toward poorly represented categories (i.e. street furniture and pole-like objects), it is evident that a significant class imbalance still exists (Figure 4). The reasons for this is two-fold. Firstly, continuous features such as road and pavement cover significantly larger areas than smaller discrete features. Secondly, due to the nature of MLS, objects closer to the scanner are sampled with a higher point density. As MLS are typically car mounted, road and pavement naturally have very high point densities. A sensible pre-processing approach to account for this issue to first voxel downsample the point cloud to a regular point density. This technique has been shown to considerably improve classification accuracy for both outdoor and indoor point clouds [7]. As one of the primary benefits of a self-constructed synthetic model is the ability to choose the object placement distribution, it is evident from our dataset that this should be further exaggerated still.

SynthCity has been designed primarily to be used for semantic per-point classification. As such each point contains a feature vector and a classification label. Whilst this is useful for a range of applications, currently the dataset does not contain instance IDs for individual object extraction. As each object is a discrete object within the Blender environment extraction of instance id's would be reasonably trivial to extract. Moreover, a simple post processing script could be employed to convert instance IDs to 3D instance bounding boxes which would enable the dataset to be used for 3D object localisation algorithms as well as per-point classification. With SynthCity being an ongoing project we plan to implement this in future releases.

Conclusions

We have generated SynthCity an open, large-scale synthetic point cloud. We release this dataset to help aid research in the potential use for pre-training of segmentation/classification models on synthetic datasets. Impact of such research outcomes are in the automated tagging of urban assets. Owners or stake holders in urban infrastructure can take stock, monitor change and generate digital twins through automatically classified reality capture data. We argue an ability to generalise from synthetic data to real world data would be immensely beneficial to the community as such a wealth of existing synthetic 3D environments exist. Most notably those generated from the gaming, virtual environment and simulated training industries. Our model contains 367.9M perfect labelled points with 5 additional features; red, green, blue, time, eol. In addition, we also present an identical point cloud with the permutation of Gaussian sampled noise, giving the point cloud a more realistic appearance.

As well as the outdoor scenario we have generated FireNet an open ML training dataset for visual recognition of fire safety equipment. We release this dataset to kick-start further ML developments in both academia and industry and as a seed point for collaborative research. Impact of such research outputs is in the automatic asset tagging for fire safety equipment form (mobile phone) imagery. Building owners or asset managers can populate digital twins with this automatically generated tagging information. We used the available funds to explore crowd-sourcing of semantic labels for technical equipment. We have shown that even relatively small teams can generate relevant sized datasets with a quick turn-around. Our tool chain has proven successful and we are able to generate further domain specific datasets with future collaboration partners.

References

[1] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, 2009, pp. 248–255.

[2] T.-Y. Lin *et al.*, “Microsoft COCO: Common Objects in Context,” in *Computer Vision – ECCV 2014*, 2014, pp. 740–755.

[3] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The Pascal Visual Object Classes (VOC) Challenge,” *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Jun. 2010.

[4] A. X. Chang *et al.*, “ShapeNet: An Information-Rich 3D Model Repository,” *ArXiv151203012 Cs*, Dec. 2015.

[5] M. J. Crawford, J. Cann, and R. O’Leary, *Uniclass: unified classification for the construction industry*. RIBA London, 1997.

[6] M. Buhrmester, T. Kwang, and S. D. Gosling, “Amazon’s Mechanical Turk: A New Source of Inexpensive, Yet High-Quality, Data?,” *Perspect. Psychol. Sci.*, vol. 6, no. 1, pp. 3–5, Jan. 2011.

[7] D. Griffiths and J. Boehm, “Weighted Point Cloud Augmentation for Neural Network Training Data Class-Imbalance,” *ISPRS - Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.*, vol. XLII-2/W13, pp. 981–987, Jun. 2019.