

Urban feature analysis from aerial remote sensing imagery using self-supervised and semi-supervised computer vision

Sachith Seneviratne^a, Jasper S. Wijnands^b, Kerry Nice^a, Haifeng Zhao^a,
Branislava Godic^a, Suzanne Mavoa^{e,f}, Rajith Vidanaarachchi^{a,d}, Mark
Stevenson^{a,d,e}, Leandro Garcia^c, Ruth F. Hunter^c, Jason Thompson^a

^a*Transport, Health and Urban Design Research Lab, Melbourne School of Design, The University of Melbourne, Parkville VIC 3010, Australia*

^b*Royal Netherlands Meteorological Institute (KNMI), Utrechtseweg 297, De Bilt, The Netherlands*

^c*Centre for Public Health, Queen's University Belfast, Belfast, Northern Ireland*

^d*Faculty of Engineering and Information Technology, University of Melbourne, Melbourne, Australia*

^e*Melbourne School of Population and Global Health, University of Melbourne, Melbourne, Australia*

^f*Environmental Public Health Branch, Environment Protection Authority Victoria, Melbourne, Australia.*

Abstract

Analysis of overhead imagery using computer vision is a problem that has received considerable attention in academic literature. Most techniques that operate in this space are both highly specialised and require expensive manual annotation of large datasets. These problems are addressed here through the development of a more generic framework, incorporating advances in representation learning which allows for more flexibility in analysing new categories of imagery with limited labeled data. First, a robust representation of an unlabeled aerial imagery dataset was created based on the momentum contrast mechanism. This was subsequently specialised for different tasks by building accurate classifiers with as few as 200 labeled images. The successful low-level detection of urban infrastructure evolution over a 10-year period from 60 million unlabeled images, exemplifies the substantial potential of our approach to advance quantitative urban research.

Keywords: Computer vision, Urban Analysis, Representation learning, Transport

1. Introduction

Advances in deep learning methods [1] have enabled the analysis of very large datasets, including those containing overhead and satellite imagery, in a fully automated manner. High-definition aerial imagery datasets are becoming

increasingly available as a result of improved capture and storage techniques, as well as advances in processing power. Combined, this is enabling detailed analysis of higher resolution remote sensing scenes. The traditional deep learning process follows the steps of data collection, data labeling, model training and inference on unlabeled data to assign labels automatically to the unlabeled data.

Due to the sheer volume of available data, computer vision techniques are uniquely suited to efficiently process them for different tasks such as classification, object detection and semantic segmentation. In machine learning, supervised learning, which operates on labeled data to build a predictive model, has been extensively used to harness information from aerial imagery. Supervised learning techniques excel when provided with a large volume of labeled data. However, such data needs to be labeled manually which is labour-intensive and therefore expensive and difficult to scale. In contrast, unlabeled data such as satellite imagery is more freely available and exists in greater quantities. Several learning paradigms have investigated how to harness unlabeled data sources more efficiently including self-supervised learning and semi-supervised learning.

Recent advances in aerial imagery techniques have led to a rapid increase in the amount of overhead imagery available. This increase is led primarily by the higher resolution of imagery capture (for example - imagery captured at 10cm resolution would generate 100 times more data compared to imagery captured at 100cm (1m) resolution. However, in order to make use of this data, storage and processing power must also keep up. It is therefore imperative that analytical pipelines are capable of handling such data while maintaining key performance metrics such as analytical accuracy and speed.

High-resolution aerial imagery captures detailed urban characteristics, enabling the potential identification of important urban features [2] such as cycling infrastructure at scale. This work introduces methods for effectively exploring such large volumes of data (scaling up to 60 million images), using a much smaller labeled set of images (as few as 200 images). Methods leveraging self-supervision, semi-supervision are introduced, evaluated and deployed across 15 cities in Australia.

1.1. Advances in neural network training techniques

1.1.1. Self-supervised representation learning

Self-supervised learning extracts knowledge from unlabeled datasets by setting up a pretext task on which the model can be pretrained in a supervised manner [3]. In self-supervised workflows, the focus is on the intermediate representation that is learned by the self-supervision pretext task, rather than maximising prediction accuracy. This intermediate representation is used in downstream tasks such as object detection, with the expectation that the representation learned during the pretext task is robust from a semantic and structural perspective.

There is currently a large body of work focused on learning task-independent representations using these techniques. For example, Noroozi and Favaro [4]

formulated a jigsaw puzzle task by selecting several adjacent blocks of pixels. After shuffling the blocks, the model’s task is to recover the correct spatial order (see Fig. 1a). This task requires high-level reasoning based on the objects and details visible in the image. Therefore, a model that excels in the pre-training task is likely to contain a useful representation of the image. Similarly, Doersch et al. [5] designed the task of retrieving the relative position of a tile compared to a selected image section (see Fig. 1b).

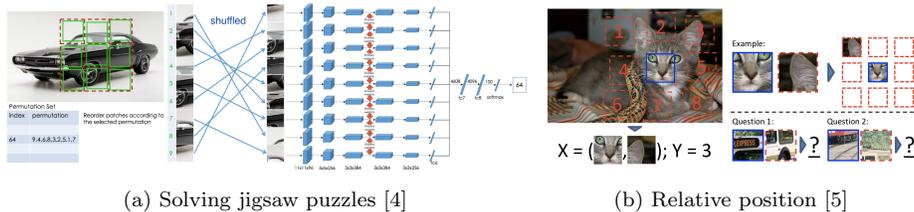


Figure 1: Examples of pretext tasks. By performing such tasks, the neural network develops an initial understanding of the types of tasks it will be trained for in the future. This reduces the difficulty and magnitude of future training.

Importantly, while self-supervised learning tends to reduce the labelling requirements for training neural networks, it does not provide a means for labelling large datasets. This is because self-supervised learning generally provides psuedo-labels for the model to build an initial representation of the world, which helps to reduce the number of labeled data points it needs to see to build a hypothesis about a particular category, but does not necessarily label data points related to those particular categories.

1.1.2. Semi-supervised learning

Semi-supervised learning corresponds to the class of machine learning techniques where a large amount of unlabeled data is available alongside a smaller collection of labeled data. These approaches attempt to use the small volume of labeled data to assign labels to the much larger volume of unlabeled data, in an iterative fashion. Thereby, the set of labeled data grows during analysis, leading to more accurate models.

Prior work has used semi-supervised approaches (also referred to as bootstrapping approaches in some research areas) to improve prediction accuracy of predictive models by generating more training data. However, very few operate in a fully automated manner. An early work in this paradigm of models learning by themselves is by Yarowsky [6], who investigated the possibility of using labeled sentences coupled with unlabeled data to perform word-sense disambiguation. Several works also explored applicability of this technique in computer vision. For example, Cui et al. [7] iteratively grew their dataset by merging in high-confidence predictions from their model. However, a manual vetting process was employed at each step. Huang et al. [8] used morphology and colour-based indices using predefined formulae, as well as openly available

information sources to generate training sets and classify images into the classes of buildings, roads, soil, water, shadow, and vegetation. A key issue with classification approaches is often that classes are assumed to be mutually exclusive. However, in aerial images of urban scenes, it is possible for roads, vegetation, soil, water and buildings to co-exist within the same image.

In general, the key improvements due to semi-supervised learning strategies can be incorporated at two levels:

- **Model level** involves improvements incorporated into model training processes and focus on providing models with a more robust representation from fewer initially labeled image samples.
- **Data level** involves improvements in the semi-supervised labelling process itself, allowing model-independent improvements via techniques such as heuristics and morphological feature extraction.

Model level improvements generally include techniques which are useful even outside of the scope of semi-supervised learning as well. In fact, many of these techniques are used to improve supervised learning models. As examples, Miyato et al. [9] use adversarial training, Siddharth et al. [10] use disentangled feature learning and augmentation strategies such as RandAugment introduced in Cubuk et al. [11] are also commonly used.

Data level techniques operate outside of the scope of the model. These increase the probability of the model correctly labelling unlabeled image samples without human intervention. For example, Kothari and Meher [12] use unlabeled neighbourhood information to improve model performance.

As these two types of techniques apply at different levels, it is additionally possible to overlap them for potential combined improvements as well.

Most work in semi-supervised learning focuses on images captured in the horizontal perspective (images generated by cameras in non-aerial settings), due the abundance of labeled data which enables much easier model evaluation. By treating a large part of the dataset as unlabeled, it is still possible to easily evaluate model behaviour with small labeled datasets while also providing very robust accuracy, precision and recall metrics as required. Using an unlabeled dataset only enables the provision of estimates of such performance metrics, as the ground truth of a large part of the dataset is unknown. However, this type of analysis more accurately matches use of the technique with unlabeled datasets in the wild.

Table 1 contains a comparison of such techniques based on labeled set size, perspective and unlabeled set size. This comparison indicates reported results for the model using the lowest number of labeled images and not the number corresponding to the best results.

Many techniques compare performance based on a percentage of unlabeled data used as labeled data (for example, 1% of data used as labeled data). However, this is not necessarily representative of annotation effort, which is a function of the absolute number of labeled images. As evaluation is primarily carried out using labeled data which is treated as unlabeled data (by hiding the label

Work	Perspective	labeled	unlabeled
Kothari and Meher [12]	Vertical	1200	15000
Yalniz et al. [13]	Horizontal	1M	100M
Zhai et al. [14]	Horizontal	12800	1.2M
Xie et al. [15]	Horizontal	250	25000(S)
This work	Vertical	200	60M

Table 1: Dataset details for semisupervised learning work. (S) indicates synthetic/augmented data generation as the main source of data for semisupervised learning. Perspective refers to the capture angle with vertical perspective corresponding to overhead imagery.

from the model), it is straightforward to do so. However, for use in the wild with a new unlabeled dataset, data annotation effort is often the limiting factor. Additionally, most techniques report performance based on the training set size as a percentage of unlabeled/total set size, disregarding the labelling requirements for validation data. In this article, a major objective is to limit the total labelling requirement, and aim to work with smaller validation sets as well.

1.1.3. Active learning

In machine learning, active learning refers to the class of techniques where the model can iteratively query a human user regarding the ground truth of a subset of input data. Based on the user’s input, the model then performs additional learning to improve its prediction accuracy. This requires manual intervention at each iteration of the learning process. Active learning has been successfully used for a multitude of tasks including crystal structure prediction [16], vehicle detection [17] and facial recognition [18]. These approaches work well in theory, by using an oracle or already annotated dataset for evaluation purposes. However, Settles [19] argues that, when attempting to bootstrap a new dataset in practice, it is generally not time efficient to wait for model training to finish before annotating more images. A key difference between semi-supervised learning and active learning is that the agent doing the annotation in semi-supervised learning is an automated model, whereas in active-learning it is generally a human.

1.2. Applications of overhead imagery

Overhead (satellite and aerial) imagery has been used in previous research for a wide variety of applications. The features of the urban fabric provide important pointers to explore pressing issues in contemporary society. For example, information extracted from high-resolution satellite imagery has been used to estimate poverty in African countries [20] and provide disaster and crisis-management support [21]. Further, it has proven valuable for inferring population size [22], assessing land cover changes [23], and monitoring food security through agricultural crop mapping [24]. Beyond imagery, satellite remote

sensing has enabled global analyses of air pollution [25], vegetation changes [26], and economic activity using night-time lights as a proxy indicator [27].

The studies above provide evidence of the significant potential of space-based observations to explore and understand the effect of contemporary social issues on spatial organisation. While some studies implicitly use features in satellite imagery to find associative evidence, other research has focused purely on feature extraction from imagery. Importantly, the primary task of feature detection could lead to a detailed understanding of environment characteristics and enhance the explainability of research findings. In this case, the task can be formulated as an object detection problem. This research direction has been taken by various studies, generally specialised for the detection of a single object category visible in satellite imagery. For example, Vakalopoulou et al. [28] and Yuan [29] developed algorithms for building detection. Further, many studies have explored the extraction of road networks from satellite imagery [e.g., 30, 31, 32]. Wang et al. [30] achieved this by predicting the road direction in satellite images and constructing the network by analysing imagery at adjacent locations. Zhang et al. [31] created an image segmentation approach based on U-Net [33] to extract road networks. More detailed characteristics of the road network can also be detected, such as specific intersection designs [34]. Cadamuro et al. [35] assessed road quality from satellite imagery, using a combination of an autoencoder [36] and Long Short-Term Memory neural networks [37] to extract and analyse features. Further, Chen et al. [38] designed a methodology that can be used to detect the number of vehicles on roads. An illustration of some of these approaches is provided in Fig. 2.

1.3. Objective

Over the past decades, technical advances in satellite remote sensing have greatly improved the quality of satellite imagery. Further improvements in image resolution have been achieved through aerial photography using airplanes, resulting in an increased availability of very high-resolution overhead imagery datasets. The additional details in high-definition aerial imagery provide opportunities for improving the accuracy of object detection methods. Further, it allows for the detection of new object classes previously undetectable from satellite imagery and difficult to collect otherwise. For example, uncommon types of infrastructure (such as cycling infrastructure) are poorly represented or incomplete in existing datasets, but can be analyzed using aerial imagery. Besides taking advantage of improvements in input data, this article explores new methods for object detection. As described above, current object detection methods are either highly specialised towards extracting a single characteristic from the environment (e.g., buildings or vehicles), or detect many classes at once with extensive manual annotation requirements. Therefore, the gap addressed in our research is the lack of a resource-efficient, generic method that can extract a more complete set of features to describe the environment in a single image. As indicated by Mnih and Hinton [32], pre-training using unsupervised learning methods can improve model accuracy substantially, providing opportunities to develop such a generic approach.

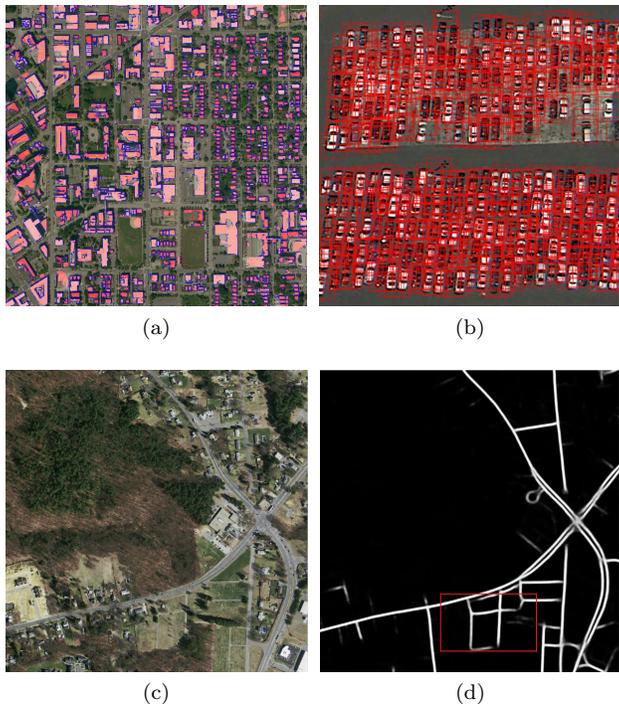


Figure 2: Specialised object detection models: (a) buildings [29], (b) vehicles [38], (c–d) roads [31]

The main motivation of this work is to enable an extensible pipeline that simplifies the collection of data for the purpose of predictive analysis across different infrastructure classes in a scalable manner. Wherever possible, the pipeline was optimized with the following objectives in mind:

- Minimize human annotation effort.
- Flexibility to easily add more classes .

2. Methodology

While existing methods have explicitly explored many road related infrastructure analyses, cycling infrastructure has been poorly explored using aerial imagery. Additionally, cycling infrastructure is often clearly demarcated using specialized symbols and colorful lanes which enables its use as a well-defined type of infrastructure to explore initially using an aerial imagery analysis workflow. Therefore, while initial analysis was carried out on such infrastructure, several additional types of infrastructure and urban features were also explored to highlight the generalisability of the developed pipeline.

City	Images	Area(km ²)
Melbourne	9,018,518	3249
Sydney	6,700,303	2414
Perth	13,205,906	4758
Canberra	4,856,845	1750
Adelaide	1,286,671	464
Brisbane	12,884,242	4642
Geelong	4,601,846	1658
Bendigo	2,112,860	761
Darwin	392,492	141
Ballarat	3,017,364	1087
Hobart	1,043,840	376
Townsville	948,061	342
Cairns	822,028	296
Wollongong	781,521	282
Toowoomba	876,648	316
Total	62,549,145	22,536

Table 2: Imagery details

2.1. Data collection

2.1.1. Aerial imagery

A large image dataset was obtained using an aerial imagery provider. The image collection spanned 15 of the most populated cities across Australia for a total of 62.5 million images of size 256×256 pixels taken at a zoom level of 21 (corresponding to 0.074 metres per pixel at the equator, a tile edge size of roughly 20 metres and an area of roughly 400 square metres covered by each tile). The total area covered by the study was $22,536 \text{ km}^2$. Data collection by city/state is indicated in Table 2.

2.1.2. Cycling infrastructure

For the exploration of cycling infrastructure in urban environments, an initial sample of labeled imagery was obtained through an observational study [39]. A total of 100 participants were recruited between March 2015 and January 2017 near on-road locations in Western Australia where a crash was observed previously. Cyclists were intercepted as they stopped at traffic lights and offered a slap-band for their wrist which had the study website recruitment address printed on it. Cyclists then completed an online questionnaire and were asked to leave their contact details within the questionnaire if they were willing to be contacted to be part of the study. Potential participants were contacted by phone to further explain the study and were sent a consent form by email. Cyclists were eligible to participate if they had not been involved as a cyclist in a bicycle crash requiring hospitalisation in the previous three years, were 18

years or older, lived in the Greater Perth area, spoke English, and cycled at least once per week. If the cyclist agreed to participate, an appointment was made to attach the GPS tracking sensors to their bicycle. Data collection included the recording of up to six hours of cycling video footage and associated GPS data per participant. Participants were asked to record any cycling they participated in and ride exactly as they normally would.

After the conclusion of the observation period, the recorded GPS information was allocated to specific participant trips. The most common routes that each participant travelled on covered a total road network of 1680 kilometres in Western Australia (see Fig. 3). This consisted of 280 kilometres of bicycle paths and 1400 kilometres of on-road routes. The recorded GPS tracks were then used to annotate aerial imagery in Perth with the presence of cycling infrastructure, leading to an initial dataset of labeled imagery.

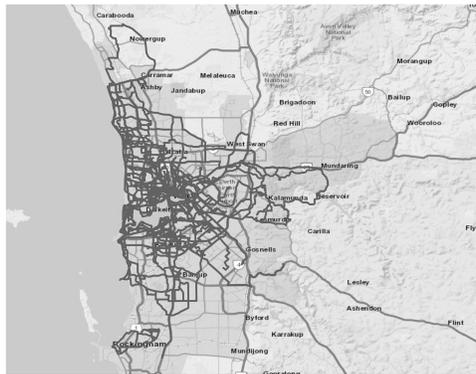


Figure 3: Cycling network extracted from GPS traces near Perth, Western Australia

2.2. Self-supervised representation learning

As discussed in 1.1.1, self-supervised learning techniques allow the use of an unlabeled dataset to build a task-independent representation of the images in the dataset. This representation can then be used for other downstream tasks. In this work, our motivations for the use of self-supervised learning techniques are that they:

- scale well in terms of predictive accuracy with datasets where a large portion of the data is unlabeled.
- allow for the rapid creation of classifiers by either transfer learning or by building a single layer on top of the existing representation.
- allow a single learned representation to be reused across multiple infrastructure identification tasks, allowing a considerable amount of computational work to become front-loaded and one-off.

An experiment was carried out to evaluate the suitability of such techniques, which are traditionally used on images taken from a horizontal perspective, for use with overhead imagery and map imagery.

As an initial selection step, SimCLR [40] and Momentum Contrast (MoCo) [41, 42] were evaluated alongside a convolutional autoencoder (AE). Evaluation was carried out for the city prediction task in [43] using satellite image data for 200 cities. MoCo (95%) had the highest validation accuracy, while SimCLR (24%) and AE (20%) performed significantly worse. Utilising the large batch size for SimCLR reported in the original paper (8192) for building the self-supervised representation was problematic due to computing resource limitations in terms of GPU memory. Instead, a much smaller batch size (64) had to be utilized for evaluation purposes. The original paper discusses the representation learning batch size as an important parameter for learning a general representation, as it impacts the difficulty of the pretext task used for self-supervised learning. Since MoCo provided considerably better results with a manageable batch size (256) and has been previously successfully used with remote sensing imagery [44], MoCo was selected for future experimental work.

For validating the utility of MoCo further for this use case, we refer to [45]. Seneviratne et al. [45] conducted an experiment to verify the applicability of MoCo and to identify the scalability of this method to unseen classes (cities). The city prediction task discussed previously was used, but while representation learning (the pretraining step) was carried out on either 200 or 1667 cities, model training and testing was carried out under two settings: 200 cities and 1667 cities. For the 200 cities, the same 200 cities as in pretraining were used: checking for the ability for the representation to cover tasks or classes captured within the pretraining data itself. With pretraining on 200 cities and training/evaluating on 1667 cities, consistency of the model in representing both previously seen classes and unseen classes was evaluated. This result is important as the class-independent or generic nature of the representation would be crucial for allowing reusability across other problem domains with multiple classes (such as different types of infrastructure). Pretraining and training was carried out on the ResNet50 architecture with a batch size of 256. For training, a high learning rate of 30 was used with stochastic gradient descent since only a single layer needed to be trained (matching a standard workflow employed in previous self-supervision based studies) [41, 44]. Detailed results in Table 3 indicate significant potential for the use of self-supervision to extend to new classes previously unseen by the pre-trained representation.

2.2.1. Ablation on using self-supervision

An ablation test was performed on the above workflow, for validating its usefulness with the aerial imagery. This was achieved by sampling 100 images each for training and 1000 images each for validation for two classes (cycle infrastructure vs other) from the aerial dataset mentioned in Section 2.1.1. A ResNet50 model was then built and trained on this task following three separate configurations. The first was instantiated with the pre-trained weights from ImageNet [46] for ResNet50, which is a commonly used approach in computer vision. An ob-

Imagery	V1/V2	Pretrain cities	Pretrain epochs	test cities	Acc
Satellite	V1	200	200	200	95%
Satellite	V2	200	200	200	99%
Satellite	V1	200	200	1667	81%
Satellite	V2	200	200	1667	95%
Satellite	V2	1667	145	1667	98%
Maps	V1	1667	200	1667	67%
Maps	V2	1667	200	1667	61%

Table 3: Testing on 1667 cities with 1000 images per city with an 80%/20% training/validation split on the city prediction task. V1/V2 refers to the version of MoCo used.

jective of this experiment is to evaluate the suitability of such a technique for use with overhead imagery. A single fully connected layer was trained for the purposes of class prediction, and was placed on top of the final bottleneck layer of the ResNet network (identical to ImageNet training except for the number of classes). The second configuration used a pretrained representation built from 100,000 unlabeled images from the aerial imagery dataset. These pretrained weights were used instead of the weights loaded from the pretrained ImageNet model. For both these configurations, the layers of the ResNet are frozen and the corresponding weights are not updated during training. This ensures that the model is forced to rely on only its pretrained representation as a feature extractor, while learning only very high level abstract concepts relating to the task at hand. A high learning rate of 30 was used with stochastic gradient descent since only a single linear layer was to be trained. The third configuration uses the pretrained representation learned in the second configuration, but uses it for end-to-end transfer learning. In this configuration, all the weights of the ResNet are updated during the training process, which is not the case in the other configurations. A learning rate of 0.001 was used with stochastic gradient descent in order to minimize changes to the pre-trained weights under this configuration. This low weight aims to minimize the destruction of pre-learned features in the model, by only performing small tweaks instead of big shifts in existing features. The neural network was trained for 200 epochs and the checkpoint with the best validation performance was used for reporting performance. The results are in Table 5 under Section 3.1.1.

2.2.2. Characterizing self-supervised performance

As an initial evaluation of transfer learning from the self-supervised representation, an experiment was carried out to evaluate ResNet50 based on finetuning by transfer learning from the frozen MoCo representation. Results are in Table 6. The objective of this experiment was to better characterize performance of the two configurations built on the pretraining workflow. The complete dataset of training set, validation set and test set each representing 2 classes, contained 33,337 images. These aerial images were randomly selected from a large set of

labeled road images sampled from areas known to have cycling infrastructure. Images containing cycling infrastructure were manually filtered such that 18,642 images contained cycling infrastructure, and 14,695 did not contain any cycling infrastructure. The ResNet50 architecture was used in all experimentation and the highest validation accuracy model was picked as the final model. For the “Frozen” configuration, a learning rate of 30 and a batch size of 4 was used with stochastic gradient descent, while for the “Transfer” configuration, a learning rate of 0.001 was used alongside a batch size of 16 with stochastic gradient descent. By testing different configurations of training and validation set sizes, the expectation was that a better understanding of model performance scaling with larger training set sizes could be obtained. This would in turn serve to confirm the results in Table 5 while indicating potential thresholds in terms of manual annotation requirements for solving tasks of this nature. The overall size of the dataset is kept fixed to more accurately mirror the actual situation of using a model to iteratively grow a dataset from a pool of unlabeled images: the size of the unlabeled image set would shrink as more images are moved out of the unlabeled dataset. The results are in Table 6 under Section 3.1.2.

2.3. Semi-supervised learning

Semi-supervised learning was explored as a means of generating more accurate models as well as for creating a workflow capable of utilizing the large dataset available to its maximum potential.

There are two main configurations used in this regard, with training details broadly in line with previous experiments: Frozen and Transfer. The main focus of this section is exploring techniques that allow the training set of the model workflow to continually expand, thereby creating more accurate models. This creates a positive feedback loop that can be used with minimal manual tuning to automatically label and process the entire dataset.

2.3.1. Initial semi-supervised experiment

To evaluate the suitability of semi-supervised learning, an experiment was carried out using the above configurations. These configurations were evaluated on a single task (cycling infrastructure categorization) on the same dataset of 33,337 images as in Section 2.2.2. The results can be found in Table 7 under Section 3.2.1.

2.3.2. Semi-supervised consistency

As a follow-up experiment, the consistency of continued semi-supervised learning was explored as a single-class fixed dataset experiment. A priority queue based implementation was used to track the top 500 highest and lowest cycle symbol confidence predictions from the test set to merge into the training set. The validation set was fixed at 1000 images each. Continuous evaluation of the bootstrapping approach using the transfer learning from the Frozen configuration was carried out, starting with 1000 training and validation images per class with a step size of 500. The results are in Table 8 under Section 3.2.2 .

2.3.3. Analysis of multiple classes using Frozen configuration

While previous experiments were exclusively single class (looking at cycle symbol classification), this experiment aims to evaluate the methodology in a more generic manner. A practical limitation in this regard is the image annotation requirement for attempting many different tasks. To make the most of limited annotator time, a limit of 200 annotations per class per task was imposed, with 100 images each for the training and validation sets respectively. As before, the two classes correspond to a “Task” class vs a ”Background” class. The main reason for this experimental setup is that it is highly likely for multiple infrastructure classes to be present in the same image. Therefore, by creating a binary classification task, we are able to overlap annotations from multiple models on the same image in a similar fashion to object detectors, without needing to generate bounding boxes for the different tasks which would severely limit annotator time availability to explore multiple classes. Within this limitation of 100 training images, prior experiments (Section 2.2.1) indicate that the Frozen configuration performs best and that a validation set of 100 images should be sufficient in this respect. Evaluation of the Frozen configuration trained on 100 images of each class in training and validation is provided to compare the base performance of the methodology on each task. The Frozen configuration is used as it is useful for providing a baseline level of performance to compare against. Further, it has the added benefit of being trained very fast due to the high learning rate used. The percentages reported correspond to the precision of the class under investigation. No false negatives were detected during this set of experiments. Evaluation was carried out on a random sample of 100 images drawn uniformly from the top 1000 predictions at each location ordered by confidence. The results of this experiment can be found under Section 3.2.3.

2.3.4. Automated analysis using archival semi-supervised learning

This experiment explored the development of a workflow centered around using historic images at locations for improving model accuracy. In particular, the main objective was to build upon the results from Section 2.3.3 by using historical imagery as a data augmentation/semi-supervised learning strategy.

To this end, several key semantics of the task at hand are exploited. The key insight for this methodology is that infrastructure is static: if it is available at a location at present, it is likely to have been present at that location in the recent past. It is also reasonable to expect that the probability of finding that infrastructure would decrease if the image was taken at an earlier date than a later date, simply because the infrastructure might have been constructed at an intermediate date. By contrast for the background class: if a particular image does not contain some infrastructure it is highly unlikely to have been there in the past: effective planning schemes mean that cities and other infrastructure are usually planned well ahead of time, and drastic changes are unusual in the short term.

Hence, the following assumptions are made regarding historical images at a location:

- The probability of finding the task class in historical images at a location correctly labeled as the background class is negligible.
- The probability of finding the task class in historical images at a location correctly labeled as the task class is high, with the probability of finding the task class in more recent images being higher than in older images.

Considering the model as a “task” class detector, a false positive confounder would be an image of the “background” class being incorrectly classified as belonging to the “task” class (identical to a false positive). Let Φ be the class of all historical images at all background image locations from the training set. Then, consider the set $\Theta \subset \Phi$ which contains all the confounders from the model performing inference on images contained in Φ . The set Θ is then a very informative dataset for the current model to learn from, as the model has been unable to classify them correctly, despite having seen a preceding image in the training set in the background class. Additionally, more recent confounders would be more useful than older ones, as the more recent images could be expected to look more structurally similar to the image at present and therefore contain more interesting features to include in the background class (as opposed to, for example, an unbuilt area from a long time ago which would likely not add much predictive value to the background class). Note that this logic is not necessarily commutative if the “task” and “background” classes are swapped: the infrastructure under investigation might have been constructed/painted very recently and thus, may not necessarily be misclassified as being “background” class (since if the “task” class is not present in the image, it belongs, by definition, to the “background” class). Conceptually, this is similar to boosting[47] in machine learning, as images misclassified by the model are assigned with an increased weight into the training set thus increasing their importance in terms of contribution to the decision boundary of the model.

Let Φ_T be the set of all historical images corresponding to training set locations and Φ_B be the set of all historical images corresponding to background locations, with $\Phi = \Phi_T \cup \Phi_B$. Note that the latest available images also count as historical images by definition and as such would be included in these sets. Due to being historical images of labeled locations, clearly the sets Φ_T, Φ_B contain images that the model could learn from, in a supervised manner. However, not all images would be equally useful or correct to learn from. Thus, assigning a weight to each individual historical image allows the training process to be controlled (when assigned a weight of zero, an image would essentially have no impact on the training process). Therefore, the problem at hand can be defined as follows:

Let Φ_T^i and Φ_B^j correspond to the above sets with i, j corresponding to arbitrary orderings (indices). Then let the individual loss of each training sample be determined by the function $L(x)$, which would apply the loss function used in the neural network to the corresponding output of x . Then, the overall loss function becomes:

$$Loss = \sum_i \alpha_T^i L(\Phi_T^i) + \sum_j \alpha_B^j L(\Phi_B^j) \quad (1)$$

Where $\alpha_T^i \in \mathbb{N}$ corresponds to individual historical task weights and $\alpha_B^j \in \mathbb{N}$ corresponds to individual historical background weights. Without loss of generality and for simplicity, let the first N elements of both orderings Φ_T^i and Φ_B^j be set to an arbitrary ordering of the initial human labeled training set of N images per class. As the model trains in a semi-supervised manner, the main difference in the data composition is tracked by the different values of α over the entire dataset. Note that images with $\alpha = 0$ have no contribution to model training, and may be omitted during training.

The following operations are defined in order to modularise the workflow for the semi-supervised learning process for improving the performance of the models using archival imagery. It is important to note that confidence metrics are defined with respect to the ‘‘task’’ class. The confidence metric corresponds to the probability of a particular image belonging to the ‘‘task’’ class and is related to the probability of the image belonging to the background class as $P_T = 1 - P_B$ due to the presence of only 2 classes.

- **Train** - Builds a classifier from the currently available training dataset as defined by Equation 1.
- **Predict** - Uses the most recently built classifier to perform prediction on the historical datasets (Φ_T and Φ_B separately) and assigns confidence scores based on the **task** class (not the background class).

The semi-supervised learning process relies only upon training several iterations of computer vision models which have access to different training sets. The **train** and **predict** operations provide interfaces for this functionality. As any modifications to the data/weights only affect the process once a model is trained and prediction is carried out on Φ_T and Φ_B (thus updating the confidence metrics), each step/iteration of the semi-supervised learning process begins with training the model and predicting on Φ_T and Φ_B .

- **Update Task** - Increments α_T^i corresponding to the M_T highest confidence task detections in Φ_T .
- **Update Background** - Increments α_B^j corresponding to the M_B lowest confidence task detections in Φ_B .
- **Update Confounders** - Increments α_B^j corresponding to the M_C highest confidence task detections in Φ_B (hence matching the definition of confounder: high confidence, but assigned to the wrong class).

The **update** operations are used for managing the datasets over iterations of the semi-supervised learning process. By updating the contribution of each individual image to the loss function, the decision boundary of the

model is modified as well, with some images receiving a higher importance than others. It is important to note that order statistics (such as the M_T -th largest confidence value) need to be maintained independently for the two datasets Φ_T and Φ_B as the underlying semantics and class probability distributions for the two datasets are very different.

In combination, these operations define the behaviour of the semi-supervised technique. The temporal control of image availability over time is managed by gradually broadening the time frame over which the model is allowed to update values of α : initially, only values corresponding to more recent images may be updated but in later iterations, α values corresponding to earlier images may be updated as well. This greatly decreases the probability that the model will incorrectly classify an image due to having less of a connection (structurally or otherwise) to the image’s present predecessor image. This behaviour is determined by the parameters D_T and D_B corresponding to the “task” and “background” classes and denotes the maximum duration (in months) from the latest image in the dataset that an image would need to have been captured, in order for the α value to be updatable. In other words, α_T^i may be updated if and only if the image Φ_T^i was captured within D_T months of the last image in Φ_T , and similarly for $\alpha_B^j, D_B, \Phi_T^i$ and Φ_T .

Algorithm 1: Update algorithm

Signature: Update($\Phi[1\dots N]$, $\alpha[1\dots N]$, $conf[1\dots N]$, D , K , top , $date_{ref}$)

Φ - dataset of N images corresponding to α values

α - array of individual image loss contributions

$conf$ - array of confidence values corresponding to dataset

D - duration corresponding to dataset Φ

K - number of α values to be modified

top - Boolean indicating if top or bottom K alpha should be updated

$date_{ref}$ - reference date for duration comparison

Execution:

```
if  $top == True$  then
  # Select k-th largest confidence
   $conf_K = \text{QuickSelect}(conf, N-K)$ 
else
  # Select k-th smallest confidence
   $conf_K = \text{QuickSelect}(conf, K)$ 
end
while  $0 \leq i < N$  do
  if  $top == True$  and  $conf[i] > conf_K$  and
    ( $\Phi[i].date - date_{ref}$ ) <  $D$  then
    |  $alpha[i] = alpha[i] + 1$ 
  if  $top == False$  and  $conf[i] < conf_K$  and
    ( $\Phi[i].date - date_{ref}$ ) <  $D$  then
    |  $alpha[i] = alpha[i] + 1$ 
  end
end
```

As some of the novelty of this work is concentrated in the update operations, we provide an algorithmic implementation of the base update functionality in Algorithm 1. Note that an implementation which uses a heap-based approach for tracking the top and bottom K -th confidence items in a given dataset with complexity $O(N \log K + K \log K)$ for maintaining and iterating over such a list of items. An alternate implementation is to use the QuickSelect[48] algorithm to generate the K -th order statistics for a given unsorted dataset. This allows the creation of a list of the top or bottom K confidence items in $O(N + K)$ time and is presented in Algorithm 1. Since the discussed applications are not time-critical and because K is generally much smaller than N in most situations, both implementations are expected to have similar performance and are listed here for the sake of completion. The update operations previously mentioned

are implemented in Algorithm 2.

Algorithm 2: Updates for Task, Background and Confounder

Signature: UpdateTask($\Phi_T[1\dots N]$,
 $\alpha_T[1\dots N], \text{conf}_T[1\dots N], D_T, M_T, \text{date}_{ref}$)

Execution:
Update($\Phi_T, \alpha_T, \text{conf}_T, D_T, M_T, \text{top}=\text{True}, \text{date}_{ref}$)

Signature: UpdateBackground($\Phi_B[1\dots N]$,
 $\alpha_B[1\dots N], \text{conf}_B[1\dots N], D_B, M_B, \text{date}_{ref}$)

Execution:
Update($\Phi_B, \alpha_B, \text{conf}_B, D_B, M_B, \text{top}=\text{False}, \text{date}_{ref}$)

Signature: UpdateConfounder($\Phi_B[1\dots N]$,
 $\alpha_B[1\dots N], \text{conf}_B[1\dots N], D_B, M_C, \text{date}_{ref}$)

Execution:
Update($\Phi_B, \alpha_B, \text{conf}_B, D_B, M_C, \text{top}=\text{True}, \text{date}_{ref}$)

Algorithm 3 contains the overall procedure (corresponding to a single step from Table 4), building upon Algorithms 1 and 2. Additional subscripts are provided to indicate the recommended parameters for the function based on the previously defined data arrays.

Algorithm 3: Semi-supervised model step(SSMS)

Signature:
SSMS($\Phi[1\dots N], \alpha[1\dots N], \text{conf}[1\dots N], D_T, D_B, M_T, M_B, M_C, \text{date}_{ref}$)

Execution:
#Train the model based on equation 1
model = Train(Φ, α)
#Update the predictions based on the new model
conf = Predict(Φ)
UpdateTask($\Phi_T, \alpha_T, \text{conf}_T, D_T, M_T, \text{date}_{ref}$)
UpdateBackground($\Phi_B, \alpha_B, \text{conf}_B, D_B, M_B, \text{date}_{ref}$)
UpdateConfounder($\Phi_B, \alpha_B, \text{conf}_B, D_B, M_C, \text{date}_{ref}$)

After the steps corresponding to Algorithm 4, all α_B^j are incremented by one to guarantee inclusion in the training set (justified by the assumption that the class of infrastructure being detected was never present at background image locations). The final model used for querying the unlabeled non-archival dataset is then trained using transfer learning as in previous experiments. The results from this experiment are in Section 3.2.4.

2.3.5. Specialized Representation Learning experiment

A natural follow up to the experiment in Section 2.3.4 is to specialize the initial representation learned in a task-independent fashion. Additionally, the results from the ablation study in Section 2.2.1 indicate the benefits of specializing the representation in a more domain specific manner: ImageNet weights

Step	D_T	D_B	M_T	M_B	M_C	$\sum_i \alpha_T^i$	$\sum_i \alpha_B^j$
0	0	0	0	0	0	100	100
1	6	24	0	0	500	100	600
2	6	24	50	0	500	150	1100
3	12	24	100	0	500	250	1600
4	12	24	150	0	500	400	2100
5	24	48	150	250	0	550	2350
6	48	84	350	350	0	900	2700

Table 4: Summary of semi-supervised learning iterations

are often used as a generic task-independent representation, but by specializing the representation for aerial imagery, significant improvements are obtained.

The road images in this experiment were generated as the top 100,000 confidence predictions from the overall dataset by the road model from Table 11 which had a very high precision. This pre-training set was validated by sampling 10,000 images uniformly at random and performing manual annotation. The precision over this sample was 100%. Using this 100,000 image dataset, a further specialized representation was created by using an identical workflow to the initial task-independent representation. With this specialization, the representation is still task-independent, however, it is no longer as generic as the previous pre-trained representation (used in section 2.3.4) and better performance can be expected on tasks specifically associated with roads, and poorer performance otherwise. The results of this experiment are in Section 3.2.5.

2.3.6. Archival Imagery analysis

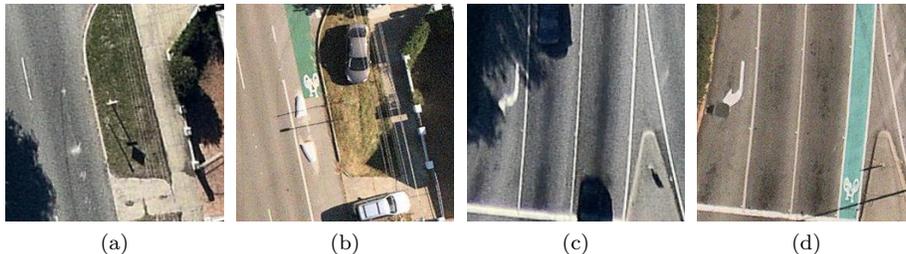


Figure 4: Manual archival imagery evaluation at two sample locations in Melbourne, Australia. Both locations were captured in 2018 (a,c) and 2020 (b,d).

The results of archival analysis are in Fig. 4. From an infrastructure analysis perspective, this allows analysis of the growth of infrastructure at the city level. Since infrastructure forms a key cornerstone of cities that affects all other aspects including transport and health, being able to analyse when specific classes of infrastructure were introduced is very useful. In particular, analysis

across multiple classes of infrastructure is very valuable in understanding the relationship between such classes and other aspects of cities such as inhabitant behaviour and population health outcomes. This can also be used to identify the trajectory of cities with regards to how well they are supporting healthy habits through initiatives such as provision of safe, active transport infrastructure for citizens.

From a computer vision perspective, datasets such as these introduce new ways of utilizing geographical information spanning multiple time-steps. For example, in this work we have exploited the static nature of infrastructure. However, it is possible to take this even further by exploiting the fact that once infrastructure is introduced to a location, it is very likely to stay there. This “expectation of the maintenance of infrastructure” allows the introduction of additional analytical steps that can improve model performance. In this situation, for example, we can expect an example vector over multiple time steps at the same location to look like (False, False, False, True, True) with the infrastructure class not being present in the first three time-steps and being introduced sometime between the third and fourth time-steps. It is reasonable, then, to assume that it will also be present thereafter. Therefore, if we assume that for some time-step t_0 , that the infrastructure was introduced to the location between t_0 and $t_0 + 1$, then the prediction for that location would take the form of a “step” function, with confidence zero upto t_0 and confidence one beyond $t_0 + 1$. This could be used to optimize the infrastructure detection models either by introducing this as a consistency loss which penalizes how different the model’s characteristic function is from a step function, or by enforcing such behaviour by performing smoothing operations on the confidence scores across multiple time-steps at the same location.

3. Results

3.1. Self-supervised learning

3.1.1. Ablation on using self-supervision

The results (Table 5) for the ablation on using self supervision detailed in Section 2.2.1 indicate that the representation learned by MoCo is superior. Interestingly, allowing the model to modify the representation learned by MoCo (Configuration 3) leads to a drop in holdout accuracy from 72% (from Configuration 1) to 61% for the smaller training set size, which is indicative of the issue of overfitting to the data.

3.1.2. Characterizing self-supervised performance

The results in Table 6 correspond to the validation accuracy obtained using the self-supervised representation as part of the experiment described in Section 2.2.2.

Config	Technique	Model	Training	Validation	Accuracy
1	MoCoV2 Frozen	ResNet50	100	1000	72%
2	ImageNet Frozen	ResNet50	100	1000	66%
3	MoCoV2 Transfer	ResNet50	100	1000	61%
1	MoCoV2 Frozen	ResNet50	1000	1000	70%
2	ImageNet Frozen	ResNet50	1000	1000	70%
3	MoCoV2 Transfer	ResNet50	1000	1000	92%

Table 5: Evaluation of the impact of different initializations and training set sizes on performance. **MoCoV2** refers to the self-supervised representation trained as part of this work. **ImageNet** refers to the standard deep learning representation of a model pretrained on the ImageNet dataset. **Frozen** refers to disabling parameter update in most of the neural network, while **Transfer** refers to allowing parameter update in most of the neural network. 1000 images were used for validation and a ResNet50 model was used as the architecture.

Method	Train	Val	Test	TP	TN	FP	FN	Acc(Val)	Acc(Test)
Frozen	100	1000	31137	13892	8181	5334	3730	73%	71%
Frozen	100	100	32937	15071	7718	6697	3451	70%	69%
Transfer	1000	1000	29337	16015	10924	1691	707	93%	92%
Transfer	5000	1000	21337	12223	8301	314	499	96.5%	96.2%

Table 6: Characterizing model behaviour over different training/validation configurations on a fixed size dataset. **Train, Val, Test** - number of training, validation and testing images used, respectively. **TP** = True Positives, **FN** = False Negatives. **Acc(Val)** and **Acc(Test)** correspond to accuracy on the validation and test set respectively.

3.2. Semi-supervised learning

3.2.1. Initial Semi-supervised experiment

Evaluation results of the dataset after evaluating using the pretrained representation are presented in table 7, corresponding to the experiment described in Section 2.3.1. Step size refers to the number of the highest confidence predictions per class which are moved from the test set back into the training set. P_{class} refers to the precision of each class $P_{class} = \frac{correct_{class}}{correct_{class} + incorrect_{class}}$ for the images that are to be moved into the training set for that class (which corresponds to the step size). For example, a $P_{non} = 0.999$ with step size = 1000 would indicate that 1 image belonging to the “non” class has been misclassified.

Method	Train	Val	Test	Step Size	P_{cycle}	P_{non}	Test Acc
Transfer	1000	1000	29337	100	1.0	1.0	92%
				500	1.0	1.0	
				1000	1.0	1.0	
Frozen	100	1000	31137	100	0.8	0.94	71%
				500	0.812	0.918	
				1000	0.796	0.909	
Transfer	5000	1000	21337	100	1.0	1.0	96%
				500	1.0	1.0	
				1000	1.0	1.0	
Frozen	100	100	32937	100	0.85	0.89	69%
				500	0.824	0.89	
				1000	0.829	0.909	
Frozen	500	500	31337	100	0.87	0.99	75%
				500	0.894	0.984	
				1000	0.878	0.98	

Table 7: Results of a single bootstrapping step over different step sizes.

Using this data from Table 7, several conclusions can be drawn:

- The accuracy of the models built using the Frozen configuration are not suitable for bootstrapping at this level
- The size of the validation set only has a minor impact on test set accuracy (2%) based on the results from the two Frozen experiments with 100 training images. Therefore, 100 validation images may function only slightly worse than 1000, thus further reducing annotation requirements.

3.2.2. Semi-Supervised Consistency

This section describes the results of the experiment described in Section 2.3.2. These results indicate that initializing semi-supervised analysis with around 1000 labeled images per class would result in the ability to consistently improve the accuracy of future iterations of models (based on the P_{Class} results).

Method	Train	Val	Test	step size	P_{cycle}	P_{non}	Test Acc
Transfer	1000	1000	29337	500	1.0	1.0	92%
Transfer	1500	1000	28337	500	1.0	1.0	93.6%
Transfer	2000	1000	27337	500	1.0	1.0	93.2%
Transfer	2500	1000	26337	500	1.0	1.0	93%

Table 8: Results of multiple bootstrapping steps over the single chosen step size

Precision	CS	Buildings	GL	Water	Trees	Roads
Canberra	6%	100%	1%	93%	90% *	100%
Ballarat	1%	100%	0%	100%	100%	100%
Bendigo	1%	100%	0%	100%	94% *	100%
Cairns	8%	100%	2%	100%	99%	100%
Darwin	0%	99%	0%	100%	100%	100%
Geelong	2%	99%	3%	100%	100%	100%
Hobart	0%	99%	1%	100%	100%	100%
Melbourne	1%	100%	6%	100%	100%	100%
Brisbane	2%	99%	5%	100%	100%	100%
Adelaide	10%	100%	1%	100%	100%	100%
Toowoomba	0%	100%	1%	N/A	100%	100%
Townsville	5%	100%	3%	100%	100%	100%
Perth	1%	100%	1%	100%	100%	100%
Wollongong	0%	100%	0%	100%	100%	100%

Table 9: Results of evaluation across different cities. CS = Cycle Symbols, GL = Green Cycling Lanes. N/A - Detection not present at location, * - corrupted images detected by model accounted for all erroneous detections.

However, noticing the trend of test set accuracy introduces another issue: the accuracy increases and then starts decreasing despite more images being present in the training set. Due to the limited size of the testing set, the reduction in the overall size of the test set can be seen to affect evaluation in this case. This is because it becomes increasingly harder to correctly predict from a smaller test set of harder examples, as more confident predictions (i.e., samples that are ‘easier to predict’) are moved out of the test set and harder test-cases are left in. For example, in results in Table 8, the test set has shrunk by 3000 images (from 29337 to 26337) corresponding to over 10% of the test set. These results indicate that a larger test set would be beneficial to further analyse this methodology, additionally allowing exploration into more classes.

3.2.3. Analysis of multiple classes using Frozen configuration

This section presents the results of the experiment described under Section 2.3.3. The results in Table 9 indicate that more specialized (hence rarer)

classes are harder to detect. This is because the potential for misclassification with a dataset of this scale increases when the probability of occurrence of a class decreases. Importantly, this is because more common classes are easier to retrieve as the probability of misclassifying them is lower. As an extreme example, even for a model with perfect accuracy, it would be impossible to retrieve a class which does not exist in the imagery dataset, such as trying to retrieve 'desert' in Antarctica. This experiment concluded that some classes are much harder to detect using self-supervised approaches in datasets of this scale. Therefore, further analysis was focused on improving the performance on such classes, which is discussed under Section 2.3.5. Since results are broadly consistent across different cities (performance is consistent across different columns in Table 9, indicating that the impact of location is minimal), further analysis was conducted on the entire dataset.

3.2.4. Automated analysis using archival Semi-supervised learning

Accuracy results using the semi-supervised method described in Section 2.3.4 are given in Table 10. As precision (of the category under exploration) is a useful indicator of performance, precision results on the 60-million image dataset can be found under Table 11. The results are compared against a supervised model trained using the same original training set taken over 5 different runs.

Class	Runs	supervised	semi-supervised	Δ
cycle symbols	5	77.6 \pm 1.32	95.3 \pm 1.33	+17.7
basketball courts	5	84.4 \pm 1.39	99.8 \pm 0.24	+15.4
solar panels	5	76.0 \pm 1.52	99.2 \pm 0.51	+23.2
flat unbuilt	5	99.1 \pm 0.58	99.8 \pm 0.24	+0.7
road writing	5	79.7 \pm 4.86	98.2 \pm 0.93	+18.5
railway lines	5	74.0 \pm 1.87	98.4 \pm 0.66	+24.4
sheep	5	92.3 \pm 4.06	99.3 \pm 0.51	+7.0
cycle lanes	5	87.0 \pm 0.71	99.6 \pm 0.37	+12.6
road arrows	5	81.4 \pm 3.76	96.8 \pm 2.22	+15.4
cars	5	72.7 \pm 2.54	95.4 \pm 0.80	+22.7
green cycle lanes	5	67.6 \pm 2.15	96.2 \pm 1.12	+28.6
footpaths	5	79.7 \pm 1.36	89.7 \pm 1.29	+10.0
buildings	5	81.7 \pm 2.42	98.6 \pm 0.58	+16.9
roads	5	82.1 \pm 0.66	96.83 \pm 1.70	+14.7
trees	5	77.3 \pm 0.51	96.75 \pm 0.75	+19.4
water bodies	5	95.2 \pm 1.6	99.625 \pm 0.65	+4.4
pools	5	96.4 \pm 0.8	99.5 \pm 0.32	+3.1
sports facilities	5	92.3 \pm 2.29	99.9 \pm 0.20	+7.6

Table 10: Average validation accuracy (and standard deviation) results over 5 model training runs indicate that the proposed semi-supervised method is more accurate and more consistent

Class	Precision(Frozen)	Precision(Archival)
Cycle Symbols	0%	85%
Green Lanes	15%	98%
Buildings	100%	100%
Cars	100%	98%
Trees	100%	100%
Water bodies	100%	100%
Solar Panels	100%	100%
Railway Tracks	62%	100%
Footpaths	94%	96%
Lane Arrows on Roads	44%	7%

Table 11: Precision results across multiple tasks on the entire dataset

3.2.5. Specialized representation learning experiment

The results in Table 12 are compared to the baseline results of this technique from Section 2.3.4 corresponding to the use of a more generic representation. The results indicate that specializing the representation using road imagery has improved performance on categories of infrastructure which coincide with roads, whereas some categories (such as water features) become harder to detect using the proposed methodology.

Class	Precision	Difference from baseline
Cycle Symbols	99%	14% ↑
Green Lanes	100%	2% ↑
Buildings	99%	1% ↓
Trees	100%	0%
Water bodies	84%	16% ↓
Solar Panels	100%	0%
Railway Lines	100%	0%
Footpaths	100%	4% ↑
Lane Arrows on Roads	99%	92% ↑

Table 12: Results of specializing the frozen representation using road imagery as a first step. It is important to note that urban features correlated with roads have enjoyed an improvement in accuracy, while water bodies(less common near roads) sees a reduction in accuracy.

4. Discussion

4.1. Speed and scalability

With modelling of this scale, it is important to consider how such analysis could be scaled across computing infrastructure to deliver results at speed. The

proposed method was able to generate results covering 15 cities in Australia spanning 22,000 km² and more than 60 million images in 3 hours. This is a throughput of 20 million images per hour or roughly 7,000 square kilometres per hour. These results were generated leveraging the trivial parallelism due to the inherent independent nature of the inference process in neural networks. Processing was performed on 12 V100 GPUs split across 3 nodes (4 GPUs per node) on the Spartan HPC platform[49]. On a single GPU, the same workload took 24 hours to complete on a single task. This run-time performance evaluation corresponds to the semi-supervised workflows discussed as part of Section 3.2.4.

4.2. Archival imagery analysis

A straightforward use-case of the models on the task of exploring the evolution of infrastructure over time was used to highlight its utility. Analysis is conducted across the city of Melbourne, and cycling infrastructure over time was analyzed. The first instance of identified infrastructure at a particular location was annotated by the year of detection. This information was used to generate a GIS layer loaded into QGIS which was then visualized as in Fig. 5. This highlights the utility of the proposed models in providing accurate and consistent data spanning multiple years over a large geographical area. The same data collected by manual processes is laborious to collect and involves repetitive work for the annotator. Beyond providing additional training data, the exploration of archival imagery provides further insights on the growth and change of infrastructure networks.

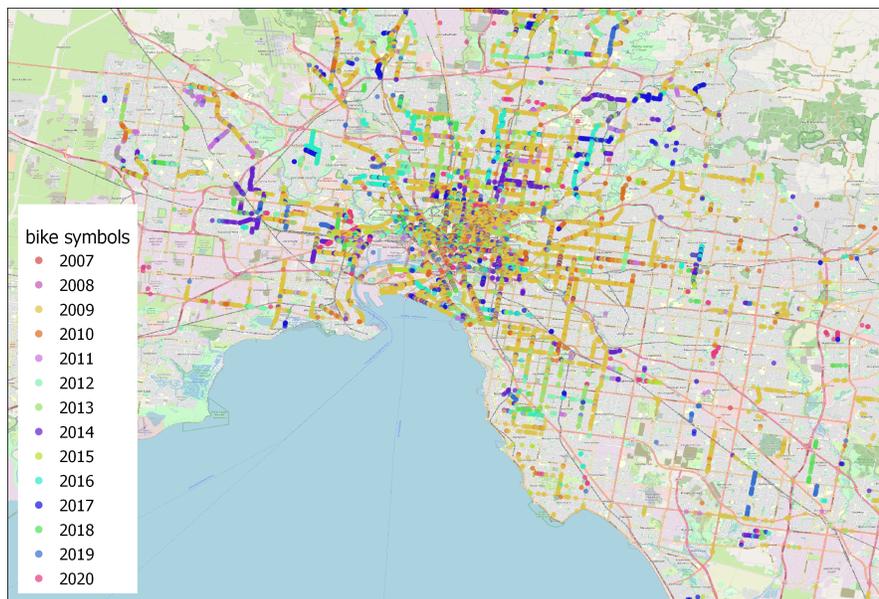


Figure 5: Generated GIS layer of cycling infrastructure over Melbourne

4.2.1. *Semi-supervised learning mixed with active learning/human-in-the-loop verification*

While the methods introduced in this work attempt fully automated analysis using semi-supervision, there are error rates associated with such analysis. It is possible to lower these error rates in between iterations by performing a manual labelling step on the results to prune out any anomalous detections. These erroneous detections can be quite helpful in directing the model away from such mistakes in future iterations by incorporating these samples into the negative class for the problem at hand. Additionally, even “failed” runs where the model cannot provide a high level of accuracy can still be quite useful if the precision is higher than the natural occurrence rate of the detection in the overall dataset. For example, consider a dataset of 100 million images with a detection which occurs in about 0.1% of images. The dataset would have about 10,000 images containing the class under investigation. If the precision of the generated model is at least 20% over the top 1000 confidence images, then by annotating those 1000 detections, at least 200 detections will be obtained and can be used to further expand the labeled training dataset. In contrast, it would require manual annotation of at least 200,000 images, on average, to do the same without using a model or some other method of filtering the data.

4.3. *Interpretability*

A key issue in neural network based methods is the interpretability of the generated model. Since the final predictive function the model commits to is the product of multiple complex layers interacting together, it is important to verify that the decision boundary learned by the model is consistent. There are many works in the area of model explainability and interpretability that relate directly to neural networks. Several of these methods were incorporated to provide further validation of our models, by visualizing the activation of the models on input images which contain the corresponding class.

Two methods (Extremal perturbations [50] and Guided backpropagation [51]) were used in this regard with the results appearing in Fig. 6.

Zhang et al. [52] provide a framework for evaluating attribution techniques by getting the model to “Point” at a single pixel and then scoring based on how far that point is from the given class in the image (15 pixel distance). Points are derived for each technique in a method-dependent fashion.

To generate confidence in the results generated by the neural network models, a similar workflow was implemented using [50]. The single most important image region activated by the neural network was highlighted within the image and manually verified. An example from cycling symbols can be found in Figure 7. Similar results were observed across other classes, however, as this is a class where only a single area within the image corresponds to the task under consideration, this forms one of the harder cases for the model and interpretability technique. Thus, this result was used to highlight and further validate the behaviour of the model.

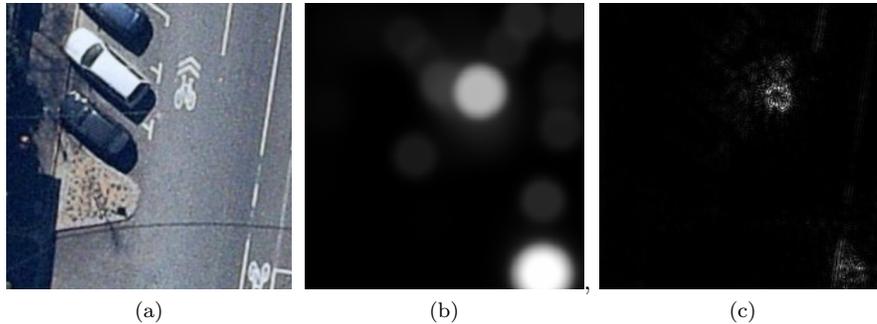


Figure 6: White activations can be seen in the images (b) (Extremal perturbations) and (c) (Guided backpropagation) corresponding to the activation of the neural network within the image for the original image (a)

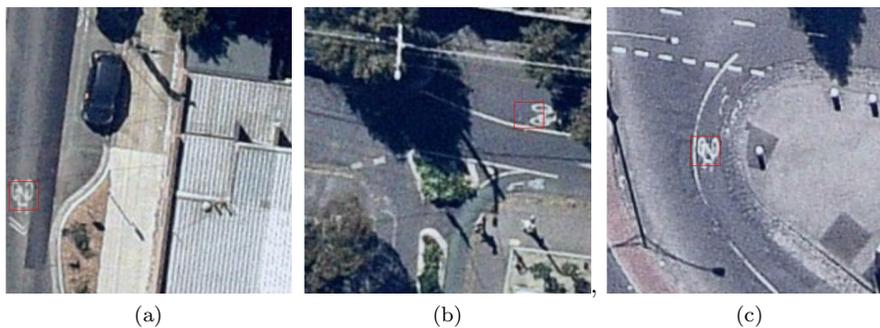


Figure 7: A red box is drawn around the region within the image most associated with the category under consideration (cycling symbols in this case) by the trained model

4.4. Significance of scalable methods in infrastructure analysis

Cycling and active transport can address the increasing congestion on road networks from motorised transport, reduce air pollution, and tackle concerning levels of population inactivity. However, cycling is not without risk of injury [53] and within increasing numbers of cyclists, comes consequent - though not matched - increases in cycling injury if separated infrastructure is not present [54]. Specifically, the number of cyclists suffering life-threatening injuries has increased by an average of 7.5% every year [55]. More recently, social distancing measures related to the COVID-19 pandemic have led to an accelerated increase of cycling activity and strong growth in new bicycle sales, globally [56]. The promotion and increased uptake of cycling requires investigation into features associated with the accompanying increased numbers of injuries. One of these features is the availability of specific cycling infrastructure, such as marked or physically separated lanes. Our study provides an approach to create such a catalogue of cycling infrastructure, which can have many useful downstream

applications such as the development of infrastructure typologies[57]. Importantly, this work showcases how the method can be extended to other types of urban features as well.

5. Conclusion

This article proposes a generic method to extract a broad set of features from aerial imagery, which describe the environment in a single image. Although an image segmentation approach can achieve similar results in a single model, one of the major limitations is the requirement of a large amount of samples for model calibration. For example, Azimi et al. [58] annotated 31 semantic categories, including low vegetation, tree, paved road, non-paved road, paved parking place, non-paved parking place, bike-way, sidewalk, entrance/exit, and 12 lane-marking types. As user requirements vary, multiple datasets were created by merging some of the detailed categories into higher-level classes (e.g., ‘nature’). These image segmentation methods have significant potential for urban infrastructure identification. However, creating annotated training datasets is a highly resource intensive process, with no guarantee that the segmentation categories match the requirements of alternative research questions. In contrast, our method requires only 200 label annotations per category, which is substantially more efficient. Several variations of the methods introduced were also explored, modifying aspects of the self-supervised and semi-supervised learning workflows. Deep learning explainability techniques were applied to verify the hypothesis learned by the model.

This article describes the accuracy of feature detection for various type of infrastructure (e.g., footpaths, cycling lanes), showing commonly encountered infrastructure is easier to detect than rare objects such as cycle symbols. However, the deep learning methods discussed in this article are able to accurately detect any of the investigated types of infrastructure given a sufficient number of training samples. Although the level of initial image annotations can be debated (i.e., set to 200 in this study), a low threshold prevents excessive annotation efforts for features that are easy to distinguish, such as rail tracks. When a higher prediction accuracy is required for certain classes, approaches such as obtaining additional historical imagery at already annotated locations can boost accuracy without the need for further annotation.

Acknowledgments

This project was supported by Australian NHMRC Grant GA80134. This research was undertaken using the LIEF HPC-GPGPU Facility hosted at the University of Melbourne. This Facility was established with the assistance of LIEF Grant LE170100200.

References

- [1] J. Schmidhuber, Deep learning in neural networks: An overview, *Neural Networks* 61 (2015) 85–117. doi:10.1016/j.neunet.2014.09.003.
- [2] B. Neupane, T. Horanont, J. Aryal, Deep learning-based semantic segmentation of urban features in satellite images: A review and meta-analysis, *Remote Sensing* 13 (2021). URL: <https://www.mdpi.com/2072-4292/13/4/808>. doi:10.3390/rs13040808.
- [3] L. Jing, Y. Tian, Self-supervised visual feature learning with deep neural networks: A survey, *IEEE transactions on pattern analysis and machine intelligence* (2020).
- [4] M. Noroozi, P. Favaro, Unsupervised learning of visual representations by solving jigsaw puzzles, in: B. Leibe, J. Matas, N. Sebe, M. Welling (Eds.), *Computer Vision – ECCV 2016*, Springer International Publishing, Amsterdam, 2016, pp. 69–84. doi:10.1007/978-3-319-46466-4_5.
- [5] C. Doersch, A. Gupta, A. A. Efros, Unsupervised visual representation learning by context prediction, in: *2015 IEEE International Conference on Computer Vision (ICCV)*, IEEE, Santiago, 2015, pp. 1422–1430. doi:10.1109/ICCV.2015.167.
- [6] D. Yarowsky, Unsupervised word sense disambiguation rivaling supervised methods, in: *33rd annual meeting of the association for computational linguistics*, 1995, pp. 189–196.
- [7] Y. Cui, F. Zhou, Y. Lin, S. Belongie, Fine-grained categorization and dataset bootstrapping using deep metric learning with humans in the loop, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [8] X. Huang, C. Weng, Q. Lu, T. Feng, L. Zhang, Automatic labelling and selection of training samples for high-resolution remote sensing image classification over urban areas, *Remote Sensing* 7 (2015) 16024–16044.
- [9] T. Miyato, S.-i. Maeda, M. Koyama, S. Ishii, Virtual adversarial training: a regularization method for supervised and semi-supervised learning, *IEEE transactions on pattern analysis and machine intelligence* 41 (2018) 1979–1993.
- [10] N. Siddharth, B. Paige, v. d. J.-W. Meent, A. Desmaison, F. Wood, D. N. Goodman, P. Kohli, H. S. P. Torr, Learning disentangled representations with semi-supervised deep generative models, *NIPS* (2017).
- [11] E. D. Cubuk, B. Zoph, J. Shlens, Q. V. Le, Randaugment: Practical automated data augmentation with a reduced search space, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 702–703.

- [12] N. S. Kothari, S. K. Meher, Semisupervised classification of remote sensing images using efficient neighborhood learning method, *Engineering Applications of Artificial Intelligence* 90 (2020) 103520.
- [13] I. Z. Yalniz, H. Jégou, K. Chen, M. Paluri, D. Mahajan, Billion-scale semi-supervised learning for image classification, 2019. [arXiv:1905.00546](https://arxiv.org/abs/1905.00546).
- [14] X. Zhai, A. Oliver, A. Kolesnikov, L. Beyer, S4L: Self-supervised semi-supervised learning, in: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1476–1485.
- [15] Q. Xie, Z. Dai, E. Hovy, M.-T. Luong, Q. V. Le, Unsupervised data augmentation for consistency training, *arXiv preprint arXiv:1904.12848* (2019).
- [16] E. V. Podryabinkin, E. V. Tikhonov, A. V. Shapeev, A. R. Oganov, Accelerating crystal structure prediction by machine-learning interatomic potentials with active learning, *Physical Review B* 99 (2019) 064114.
- [17] S. Sivaraman, M. M. Trivedi, Active learning for on-road vehicle detection: A comparative study, *Machine vision and applications* 25 (2014) 599–611.
- [18] R. Hewitt, S. Belongie, Active learning in face recognition: Using tracking to build a face model, in: *2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06)*, IEEE, 2006, pp. 157–157.
- [19] B. Settles, From theories to queries: Active learning in practice, in: *Active Learning and Experimental Design workshop In conjunction with AISTATS 2010*, 2011, pp. 1–18.
- [20] N. Jean, M. Burke, M. Xie, W. M. Davis, D. B. Lobell, S. Ermon, Combining satellite imagery and machine learning to predict poverty, *Science* 353 (2016) 790–794. doi:10.1126/science.aaf7894.
- [21] S. Voigt, T. Kemper, T. Riedlinger, R. Kiefl, K. Scholte, H. Mehl, Satellite image analysis for disaster and crisis-management support, *IEEE Transactions on Geoscience and Remote Sensing* 45 (2007) 1520–1528. doi:10.1109/TGRS.2007.895830.
- [22] C. Robinson, F. Hohman, B. Dilkina, A deep learning approach for population estimation from satellite imagery, in: *Proceedings of the 1st ACM SIGSPATIAL Workshop on Geospatial Humanities*, Association for Computing Machinery, Redondo Beach, CA, 2017, pp. 47–54. doi:10.1145/3149858.3149863.
- [23] X. Yang, C. P. Lo, Using a time series of satellite imagery to detect land use and land cover changes in the Atlanta, Georgia metropolitan area, *International Journal of Remote Sensing* 23 (2002) 1775–1798. doi:10.1080/01431160110075802.

- [24] A. Shelestov, M. Lavreniuk, N. Kussul, A. Novikov, S. Skakun, Exploring Google Earth Engine platform for big data processing: Classification of multi-temporal satellite imagery for crop mapping, *Frontiers in Earth Science* 5 (2017) 1–10. doi:10.3389/feart.2017.00017.
- [25] R. V. Martin, Satellite remote sensing of surface air quality, *Atmospheric Environment* 42 (2008) 7823–7843. doi:10.1016/j.atmosenv.2008.07.018.
- [26] G. Meera Gandhi, S. Parthiban, N. Thummalu, A. Christy, NDVI: Vegetation change detection using remote sensing and GIS – A case study of Vellore district, *Procedia Computer Science* 57 (2015) 1199–1210. doi:10.1016/j.procs.2015.07.415.
- [27] C. D. Elvidge, P. Cinzano, D. R. Pettit, J. Arvesen, P. Sutton, C. Small, R. Nemani, T. Longcore, C. Rich, J. Safran, J. Weeks, S. Ebener, The Nightsat mission concept, *International Journal of Remote Sensing* 28 (2007) 2645–2670. doi:10.1080/01431160600981525.
- [28] M. Vakalopoulou, K. Karantzalos, N. Komodakis, N. Paragios, Building detection in very high resolution multispectral data with deep learning features, in: 2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), IEEE, Milan, 2015, pp. 1873–1876. doi:10.1109/IGARSS.2015.7326158.
- [29] J. Yuan, Automatic building extraction in aerial scenes using convolutional networks (2016). arXiv:1602.06564.
- [30] J. Wang, J. Song, M. Chen, Z. Yang, Road network extraction: a neural-dynamic framework based on deep learning and a finite state machine, *International Journal of Remote Sensing* 36 (2015) 3144–3169. doi:10.1080/01431161.2015.1054049.
- [31] Z. Zhang, Q. Liu, Y. Wang, Road extraction by deep residual U-Net, *IEEE Geoscience and Remote Sensing Letters* 15 (2018) 749–753. doi:10.1109/LGRS.2018.2802944.
- [32] V. Mnih, G. E. Hinton, Learning to detect roads in high-resolution aerial images, in: K. Daniilidis, P. Maragos, N. Paragios (Eds.), *Computer Vision – ECCV 2010*, Springer, Berlin, Heidelberg, 2010, pp. 210–223. doi:10.1007/978-3-642-15567-3_16.
- [33] O. Ronneberger, P. Fischer, T. Brox, U-Net: Convolutional networks for biomedical image segmentation, in: N. Navab, J. Hornegger, W. M. Wells, A. F. Frangi (Eds.), *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, Springer International Publishing, Munich, 2015, pp. 234–241. doi:10.1007/978-3-319-24574-4_28.

- [34] J. S. Wijnands, H. Zhao, K. A. Nice, J. Thompson, K. Scully, J. Guo, M. Stevenson, Identifying safe intersection design through unsupervised feature extraction from satellite imagery, *Computer-Aided Civil and Infrastructure Engineering* (2020). doi:10.1111/mice.12623.
- [35] G. Cadamuro, A. Muhebwa, J. Taneja, Street smarts: measuring intercity road quality using deep learning on satellite imagery, in: *Proceedings of the 2nd ACM SIGCAS Conference on Computing and Sustainable Societies*, ACM, Accra, 2019, pp. 145–154. doi:10.1145/3314344.3332493.
- [36] D. H. Ballard, Modular learning in neural networks, in: *Proceedings of the Sixth National Conference on Artificial Intelligence*, AAAI, Seattle, WA, 1987, pp. 279–284.
- [37] S. Hochreiter, J. Schmidhuber, Long short-term memory, *Neural Computation* 9 (1997) 1735–1780. doi:10.1162/neco.1997.9.8.1735.
- [38] X. Chen, S. Xiang, C.-L. Liu, C.-H. Pan, Vehicle detection in satellite images by hybrid deep convolutional neural networks, *IEEE Geoscience and Remote Sensing Letters* 11 (2014) 1797–1801. doi:10.1109/LGRS.2014.2309695.
- [39] L. B. Meuleners, M. Stevenson, M. Fraser, J. Oxley, G. Rose, M. Johnson, Safer cycling and the urban road environment: A case control study, *Accident Analysis & Prevention* 129 (2019) 342–349. doi:10.1016/j.aap.2019.05.032.
- [40] T. Chen, S. Kornblith, M. Norouzi, G. Hinton, A simple framework for contrastive learning of visual representations (2020). arXiv:2002.05709.
- [41] K. He, H. Fan, Y. Wu, S. Xie, R. Girshick, Momentum contrast for unsupervised visual representation learning, in: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Seattle, WA, 2020, pp. 9726–9735. doi:10.1109/CVPR42600.2020.00975.
- [42] X. Chen, H. Fan, R. Girshick, K. He, Improved baselines with momentum contrastive learning (2020). arXiv:2003.04297.
- [43] J. Thompson, M. Stevenson, J. S. Wijnands, K. A. Nice, G. D. Aschwanden, J. Silver, M. Nieuwenhuijsen, P. Rayner, R. Schofield, R. Hariharan, et al., A global analysis of urban design types and road transport injury: an image processing study, *The Lancet Planetary Health* 4 (2020) e32–e42.
- [44] S. Seneviratne, Contrastive representation learning for natural world imagery: Habitat prediction for 30, 000 species., in: *CLEF (Working Notes)*, 2021, pp. 1639–1648.
- [45] S. Seneviratne, K. A. Nice, J. S. Wijnands, M. Stevenson, J. Thompson, Self-supervision. remote sensing and abstraction: Representation learning across 3 million locations, in: *2021 Digital Image Computing: Techniques*

- and Applications (DICTA), 2021, pp. 01–08. doi:10.1109/DICTA52665.2021.9647061.
- [46] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: 2009 IEEE conference on computer vision and pattern recognition, Ieee, 2009, pp. 248–255.
- [47] R. E. Schapire, The boosting approach to machine learning: An overview, *Nonlinear estimation and classification* (2003) 149–171.
- [48] H. M. Mahmoud, R. Modarres, R. T. Smythe, Analysis of quickselect: An algorithm for order statistics, *RAIRO-Theoretical Informatics and Applications* 29 (1995) 255–276.
- [49] L. Lafayette, G. Sauter, L. Vu, B. Meade, Spartan performance and flexibility: An hpc-cloud chimera, *OpenStack Summit, Barcelona 27* (2016).
- [50] R. Fong, M. Patrick, A. Vedaldi, Understanding deep networks via extremal perturbations and smooth masks, in: 2019 IEEE/CVF International Conference on Computer Vision (ICCV), IEEE, Seoul, 2019, pp. 2950–2958. doi:10.1109/ICCV.2019.00304.
- [51] J. T. Springenberg, A. Dosovitskiy, T. Brox, M. Riedmiller, Striving for simplicity: The all convolutional net (2015). [arXiv:1412.6806](https://arxiv.org/abs/1412.6806).
- [52] J. Zhang, S. A. Bargal, Z. Lin, J. Brandt, X. Shen, S. Sclaroff, Top-down neural attention by excitation backprop, *International Journal of Computer Vision* 126 (2018) 1084–1102. doi:10.1007/s11263-017-1059-x.
- [53] G. Henley, J. E. Harrison, Trends in serious injury due to land transport accidents, Australia 2000-01 to 2007-08, *Injury Research and Statistics series no. 66*, Australian Institute of Health and Welfare, 2012.
- [54] J. Thompson, J. S. Wijnands, G. Savino, B. Lawrence, M. Stevenson, Estimating the safety benefit of separated cycling infrastructure adjusted for behavioral adaptation among drivers; an application of agent-based modelling, *Transportation Research Part F: Traffic Psychology and Behaviour* 49 (2017) 18–28. URL: <https://www.sciencedirect.com/science/article/pii/S136984781630239X>. doi:<https://doi.org/10.1016/j.trf.2017.05.006>.
- [55] G. Henley, J. E. Harrison, Trends in serious injury due to road vehicle traffic crashes, Australia 2001 to 2010, *Injury Research and Statistics series no. 89*, Australian Institute of Health and Welfare, 2015.
- [56] G. Fuller, G. Waitt, T. Lea, I. Buchanan, K. McGuinness, The reactivated bike: Cycling activity in the 2020 COVID-19 pandemic, in: *Australian Walking and Cycling Conference, Newcastle, 2020*.

- [57] B. Beck, M. Winters, T. Nelson, C. Pettit, S. Z. Leao, M. Saberi, J. Thompson, S. Seneviratne, K. Nice, M. Stevenson, Developing urban biking typologies: Quantifying the complex interactions of bicycle ridership, bicycle network and built environment characteristics, *Environment and Planning B: Urban Analytics and City Science* 0 (0) 23998083221100827. URL: <https://doi.org/10.1177/23998083221100827>. doi:10.1177/23998083221100827. arXiv:<https://doi.org/10.1177/23998083221100827>.
- [58] S. M. Azimi, C. Henry, L. Sommer, A. Schumann, E. Vig, SkyScapes – fine-grained semantic understanding of aerial scenes, in: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, IEEE, Seoul, 2019, pp. 7392–7402. doi:10.1109/ICCV.2019.00749.