The Singapore Cycling Path Dataset

Suen Wai Lun
Stanford University
suenwl@stanford.edu

Abstract

The cycling path network in Singapore is expanding, but little research has focused on scene understanding in the context of such paths. Improved semantic understanding of cycling paths in Singapore can improve safety and pave the way for autonomous usage. We present a newly collected and annotated dataset, the Singapore Cycling Path Dataset, which could aid in efforts to improve our semantic understanding of the cycling network in Singapore. We describe the data collection and processing methodology for our dataset, which total 983 images and 2718 distinct polygon annotations. Subsequently, we apply a baseline Fully Convolutional Network (FCN) model, along with other models based on a more modern transformer architecture, to illustrate the ability of semantic segmentation algorithms to semantically parse a complex urban cycling network. Our best transformer-based model achieves an mIoU of 85.7% , with an IoU of 97.8% on the park connector class.

1. Introduction

Singapore is expanding its cycling path network, with 1300 kilometers of paths to be built by 2030 [2]. These paths are built to encourage outdoor physical activities, to reduce the reliance on vehicles by providing an alternative transportation option, and to improve safety for commuters. By design, these paths are meant to be shared by pedestrians, cyclists, and more, and their use is expected to increase as more paths are built. Due to urban constraints, these paths are not all identical in design and function. Some are built as dedicated, inter-town connections (known as park connectors). Others are built in parallel with pedestrian paths throughout public housing estates in Singapore (known as cycling paths). See Figure 1 for an example of how some of these paths look. Ultimately, these paths are part of the complex urban fabric, and users of these paths need to negotiate with vehicular and pedestrian traffic. To enhance safety, these paths often come with various markings to denote their purpose and type of traffic which is prioritised on it.

While there exists a significant body of computer vision research as well as numerous datasets focusing on road use [6, 4], the study of bicycle paths and other such multi-use pathways has been sparse. In the case of Singapore, researchers have thus far focused on quantifying their utilization [25], as well as crowd control [1]. Stronger semantic understanding of the cycling network can unlock opportunities for greater commuter safety, and potential autonomous usages. For instance, the appropriate application of semantic segmentation of the cycling network to the logistics field could make sidewalk delivery robots feasible and economically viable [11, 16].

Figure 1. A typical park connector (left) and cycling path (right, in red). A regular pedestrian path runs alongside the cycling path.

In this paper, we propose the Singapore Cycling Paths dataset and experiment with a range of semantic segmentation algorithms as a step towards improving scene understanding in Singapore's cycling network. To our knowledge, this is the first ever dataset focusing on semantic segmentation of the bicycle network in Singapore. The input to our algorithm is a set of newly collected and annotated images of the cycling network in Singapore. We then use semantic segmentation algorithms to output predicted segmentation masks denoting where different aspects of the cycling network exist on the image. We use a FCN as a performance baseline, and experiment with the ViT, Swin Transformer, and SegFormer models as ways to improve upon those baseline results. In subsequent sections, we detail the attributes of the new dataset, the segmentation algorithms we use, and ultimately show that we are able to achieve good segmentation results on Singapore's cycling network, with the ViT model achieving a mIoU of 85.7%.

Figure 2. Examples of ground truth annotations from the Singapore Cycling Path Dataset. Notice the merging of cycling (green) and pedestrian paths (blue) into shared paths at road junctions (red), and the segregated nature of park connectors (purple).

2. Related Work

2.1. Semantic segmentation methods

Fully Convolutional Networks (FCNs) were a groundbreaking development when they were first proposed because they enabled end-to-end training for semantic segmentation networks through deconvolutions [14]. Over the years, innovations such as atrous convolutions [5] and pyramid scene parsing networks [27] were introduced, improving upon various aspects of the FCN.

In parallel, convolutional neural network architectures have also advanced. Evolving beyond the VGG backbone [20] used in the original FCN paper, architectures such as ResNet [10] addressed issues with vanishing gradients. U-Nets were the first to popularise the encoder-decoder structure that is commonplace in architectures today [19]. The 2-stage R-CNN family of models introduced the idea of region proposals, and became the state of the art in object detection [8, 18]. Mask R-CNN subsequently adapted the faster R-CNN for image segmentation [9], similarly attaining excellent results.

More recently, due to the high-profile success of transformers in natural language processing, interest has shifted to the application of transformers in computer vision tasks such as semantic segmentation. This has particularly been the case after it was shown that vision transformers can attain state of the art performance in image classification tasks [7]. Subsequently, methods attempted to establish transformers as a general purpose backbone and improve the efficiency and performance of vision transformers [13, 21, 3]. More recently, the SegFormer architecture was proposed as a transformer-based architecture specifically designed for semantic segmentation [24].

2.2. Semantic segmentation of cycling paths

There exists a number of well known datasets that are used in urban scene understanding [6, 4, 15]. In particular, the Cityscapes dataset has been an instrumental benchmark, providing pixel-level annotations for 5000 images of urban environments. These images were captured in diverse conditions and in various cities [6]. Crucially, images from the Cityscapes dataset were collected from a car-mounted camera, which made it an invaluable resource for developing autonomous vehicles. However, that simultaneously restricted the scope of the collected images to streets that permit vehicular traffic. As a result, the dataset is not immediately applicable to learning about cycling paths. This is a common theme for a number of datasets focusing on urban scenes [4, 15].

There exists research focusing explicitly on cycling paths and other urban greenways in Singapore [26]. The objective is typically to understand usage patterns [25] in order to optimise their utilisation. To that end, they focus on using pre-trained models to classify the type of activity users are engaged in. For example, papers might classify the poses of individuals in the scene [25], correlating them with environmental features such as greenery and water bodies. Based on this, they then draw conclusions about the impact that these environmental features have on how the spaces are utilised by the public. These environmental features are detected through semantic segmentation models trained on generic large semantic segmentation datasets such as Ade20k [28].

3. Dataset and Features

To the best of our knowledge, there does not yet exist a dataset that contains annotations suitable for scene understanding in the bicycle path context in Singapore. To provide a foundation for improvements in scene understanding of the bicycle path network, we collected and annotated a dataset that comprises a diverse set of bicycle path images from the central region of Singapore.

3.1. Data Specifications

We attempted to capture images in a variety of conditions in order to reflec the reality on the streets and enable the model to generalise better. The images in our dataset were collected in rainy, sunny, and cloudy conditions, within daylight hours, in the Bishan-Ang Mo Kio area of Singapore. The area is primarily residential in character. Public housing, schools, parks, and canals feature heavily in the background of the images collected. Data recording was done on a bicycle-mounted camera at 1920 x 1080 resolution, at 30 Hz. Approximately 130,000 frames were collected, and of those frames, 1000 were randomly selected. After the removal of blurry and otherwise unusable frames, 983 images remained to form the dataset, which was then split into

training and test sets in a 80/20 split. Within the training set, a validation set was similarly created in a 20/80 split. This results in the following numbers of images in each of the splits as detailed in Table 1.

Split	Images
Train	628
Validation	158
Test	197

Table 1. Number of samples in the train, validation, and test splits.

3.2. Classes and Annotations

Our annotations comprise of layered polygons, and were performed with the CVAT tool. We annotated the images with 5 classes deemed to be the most relevant to a semantic understanding of the cycling network in Singapore. They are bicycle paths (green), pedestrian paths (blue), park connectors (purple), shared paths (orange), and roads (red). Bicycle paths and pedestrian paths often run parallel to each other along roads in Singapore. Bicycle paths, as the name suggests, are meant for dedicated use by cyclists and other riders, whilst pedestrian paths are for pedestrians. These two paths may merge into shared paths where dictated by the constraints of the urban environment. Park connectors are often wider, segregated, purpose-built paths that link various parks and green spaces in Singapore. They are shared by pedestrians, cyclists, and park users. Whilst it is legal for regular bicycles to be used on any of the above paths, there are tighter regulatory restrictions on power-assisted bicycles and electronic scooters, and they are only permitted on cycling paths and park connectors due to their higher speed. In total, our dataset comprises 2718 distinct polygon annotations, with the distribution detailed in Table 2.

Class	Polygon count	% of total
Bicycle paths	529	19.5%
Pedestrian paths	797	29.3%
Park connectors	313	11.5%
Shared paths	463	17.0%
Roads	616	22.7%

Table 2. Number of polygons for each of the annotated classes

4. Methods

We first attempt to establish a baseline using a Fully Convolutional Network (FCN) on the dataset, before experimenting with more recent semantic segmentation methods which employ transformers. Each of the methods we use are described in further detail in the following subsections.

4.1. Cross Entropy Loss

We first define the loss function that we use in all the following models. We are attempting to classify pixels into one of the 5 classes described above, in addition to the background class. For each of the 6 classes and for each image, the cross entropy loss can be defined as:

$$CrossEntropy = -\frac{1}{N}\sum_{i} x_i log\hat{x_i}$$

Where $\hat{x_i}$ is the predicted class for a particular pixel i , and x_i is the actual class for that pixel. To obtain the final cross entropy loss, we take the average of the loss across each class and image.

4.2. Mean Intersection over Union

To compare performance across the models we are experimenting with, we use the mean intersection over union score. This is a commonly-used metric for image segmentation algorithms. The intersection over union (IoU) measures the overlap between the ground truth and the predicted segmentation mask. Specifically, the IoU for a particular class is defined as:

$$IoU_{class} = \frac{|P \cap T|}{|P \cup T|}$$

where P represents the set of pixels predicted for a particular class, and T is the set of pixels that form the ground truth. The mean intersection over union is the average of the IoU scores across all the classes.

4.3. Fully Convolutional Network

The FCN is a pioneering model for dense prediction tasks like image segmentation. In FCNs, the input image is passed through a series of convolutional layers (the backbone) to produce a feature map. This feature map is then upsampled using transposed convolutions to match the original input size. The final segmentation map is obtained by applying a softmax activation function. As mentioned above, the loss function used is the pixel-wise cross-entropy loss.

Instead of the VGGNet backbone used by the authors of the original paper, we use the more contemporary ResNet as our backbone. Specifically, we use a ResNet-50 backbone which comprises a total of 50 convolutional, pooling and fully connected layers. ResNet addresses the vanishing gradient problem that exists in deeper architectures by providing residual connections between layers, enabling gradients to flow directly through the network. The backbone we used was pre-trained on the COCO dataset[12], and the weights were obtained from pytorch hub [17].

4.4. Vision Transformer

The vision transformer (ViT) paper was the first to show that the transformer architecture, which is traditionally used in natural language processing, can attain state-of-the-art performance on image data [7]. In vision transformers, the input image is divided into a sequence of patches, which are linearly embedded. Positional embeddings are added to retain positional information, and the resulting embeddings are fed into a transformer-based encoder as input. The model leverages the self-attention mechanisms of transformers to capture long-range dependencies between patches, producing a feature representation that is then used to predict the segmentation map. The optimization is similarly done using pixel-wise cross-entropy loss.

In the original paper, a multi-layer perceptron (MLP) head takes the encoded input and produces the output of the model, which is of dimension $N \times C$ where N is the number of input images and C is the number of classes. Because we wish to perform segmentation, which involves pixel-wise predictions, we replace this head with a segmentation head. The segmentation head consists of two convolutional layers, with instance normalisation and a GELU activation function sandwiched in between. This enables the segmentation head to produce a $N \times C \times H \times W$ output, where H and W is the height and width of the image.

We obtained pre-trained ViT weights trained on ImageNet-21k from Hugging Face[23].

4.5. Swin Transformer

Unlike ViT, which processes images as a sequence of fixed-size patches, the Swin Transformer introduces hierarchical feature maps. This means the model processes the image at multiple scales, starting with small patches before progressively merging them into larger patches deeper in the network using patch merging layers. The patch merging layers concatenate neighbouring patches, and apply linear layers to downsample the resolution of the concatenated patches. This hierarchical approach enables the model to capture fine details in the initial layers and more abstract, larger-scale features in the deeper layers, similar to CNNs. These features are termed multi-scale feature maps due to their focus on different scales of the images, and their use improves the model's ability to understand complex images.

In addition, the Swin Transformer also improves upon the quadratic time complexity of ViT by computing self-attention locally within windows rather than across the entire image. The number of windows scale linearly with image size, and the windows themselves comprise of a fixed number of patches. Thus, the time complexity for the entire image becomes linear to image size. To ensure that context across windows can be captured, the Swin Transformer incorporates the concept of shifting windows, illustrated in figure 3. This allows connections between windows that would otherwise not be possible if windows remained fixed across layers, thus allowing for sharing of context across the image.

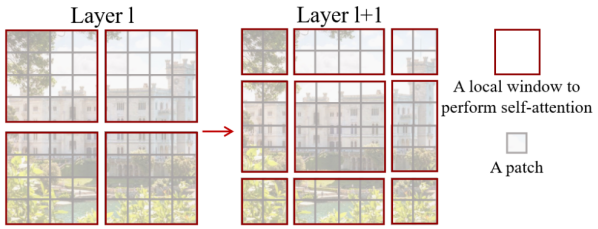

Figure 3. The shifting of windows across layers enables context across layers to be captured despite computing self-attention on smaller, fixed size windows. Reproduced from [13].

We obtained Swin Transformer weights pre-trained on ImageNet-1k from the timm package[22], replacing the segmentation head with a simple 2D convolutional layer that outputs the number of classes required for our dataset.

4.6. SegFormer

The SegFormer architecture also uses hierarchical transformers as an encoder to construct multi-scale feature maps consisting of coarse and fine features at different resolutions [24]. Using the multi-scale feature maps produced by the encoder, a decoder that comprises a series of simple MLP layers is then used to combine these multi-scale features and predict the semantic segmentation mask with dimension $\frac{H}{4} \times \frac{W}{4} \times N_{cls}$, where H, W and N_{cls} are the height and width of the input image, and the number of classes to be predicted respective. In contrast to ViT, SegFormer uses smaller patch sizes of 4×4 which are more adapted to dense prediction tasks. These patches are overlapped using convolutional layers to preserve local continuity at the edges of those patches when generating linear embeddings of those patches. This aims to improve upon ViT, where patches were not overlapped.

SegFormer also aims to improve inference efficiency vis-

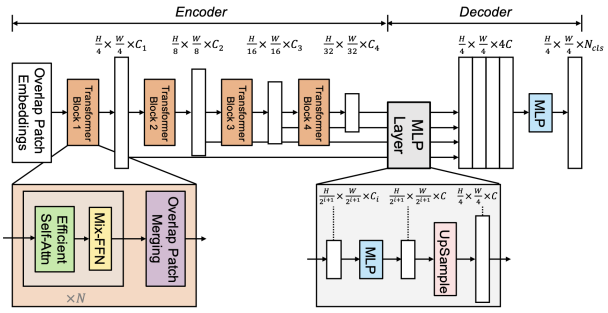

Figure 4. Overview of the SegFormer architecture. Reproduced from [24].

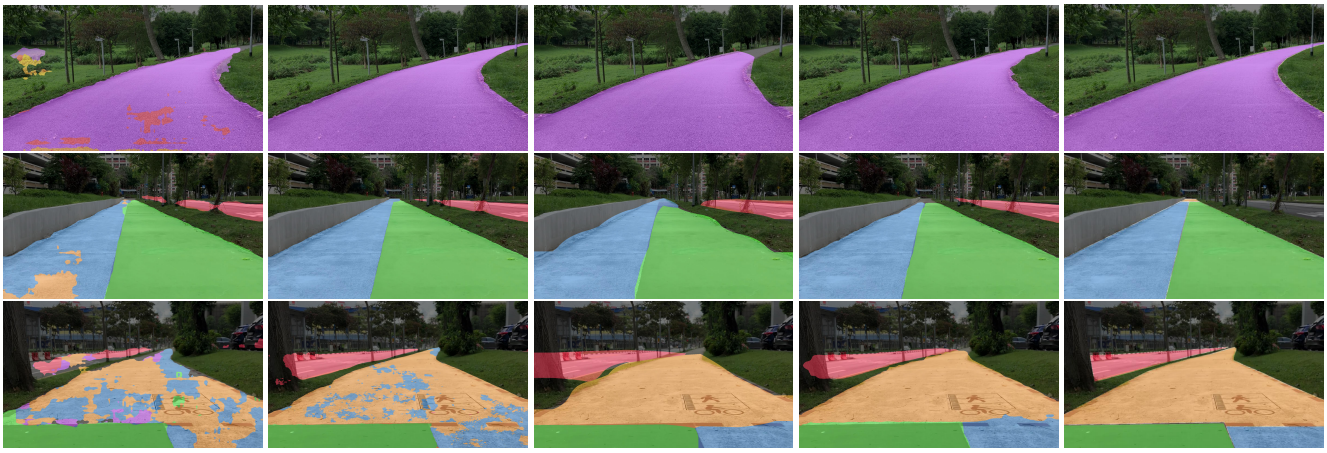

Figure 5. Segmentation predictions across our various models for a select sample of images. From left to right: FCN, SegFormer, Swin, ViT, and the ground truth.

a-vis ViT. Given an input where N is the length of the sequences, the original multi-head self attention mechanism has a quadratic time complexity $O(N^2)$. SegFormer uses a sequence reduction process that reduces the length of the sequence to reduce the time complexity. Specifically, the sequence is first reshaped from $N \times C$ to $\frac{N}{R} \times C \cdot R$, where R is the reduction ratio. The reshaped sequence is then passed through a linear layer that takes a $C \times R$ dimension input and generates a C dimensional output. This process thus yields a $\frac{N}{R} \times C$ dimensional output, and thus a reduced $O(\frac{N^2}{R})$ time complexity.

The authors designed a series of 6 encoders which use the same architecture but have different sizes, with the smallest optimised for fast inference and the largest for the best performance. We experiment with the various encoder sizes to determine which best suits our use case. Pre-trained SegFormer weights trained on the CityScapes dataset were obtained from Hugging Face[23]. Given that the CityScapes dataset also contains urban scenes, and the learned encoder representation should transfer well to our dataset, we fine tune the SegFormer models by freezing the weights on the encoder and optimising only the decode head's weights.

5. Experiments and Results

5.1. Training procedure

When training all of the above models, we first experimented with the learning rate hyperparameter with randomly chosen values between 10^{-3} to 10^{-5}. This was done with a smaller development dataset of 100 images, picked from the training dataset. When a suitable learning rate was found, we then trained on the full training set, validating the loss after each epoch with the validation set. To prevent overfitting, we stopped training when the validation loss plateaued or began to increase. We then used the checkpoint that was saved in the previous epoch.

Model	Learning rate	Batch size	Train epochs
FCN	10^{-3}	20	14
ViT	5×10^{-4}	10	11
Swin	5×10^{-4}	10	7
SegFormer	10^{-3}	1	19

Table 3. Training details for each of the models

Training was done on a M2 series Macbook Pro with 16GB of RAM. We decided to use the Adam optimiser due to its adaptive learning rates and general robustness across different architectures. We generally picked the largest batch size that could be efficiently held in memory for each model type. In some of the transformer based models, batch sizes needed to be reduced to 1, and we correspondingly replaced the batch normalisation layers in the model architectures with instance normalisation layers, since batch normalisation does not work well for small batch sizes, and led to instability in training. The final choices made for each of the methods we used can be found in Table 3. Predictions from the models were standardised to 512×512 using billinear interpolation for comparison of IoU scores.

5.2. SegFormer model size

The authors of the SegFormer model designed 6 SegFormer models from b0 to b5, with all of them using the same architecture but with different encoder sizes. b0 has the smallest encoder size while b5 has the largest. We experimented with the various encoder sizes, and found that the larger SegFormer models had the tendency to rapidly overfit to our relatively small training data set. After training with all the SegFormer model sizes, the b1 model achieved the best performance on the validation set. Ad-

ditionally, training only the decoder weights tended to yield the best performance. For brevity, whereever we refer to the SegFormer model in the subsequent section, we are referring to the SegFormer b1 model.

5.3. Results

The per-class IoU scores for each of our models can be found in Table 4. Compared to the baseline FCN model, the various transformer-based models we experimented with performed better, with higher mIoU scores of between 83.8% and 85.7%, compared to 73.1% for the FCN model.

Across the models, park connectors have the best IoU scores, reflecting the fact that they tend to be built in a manner that is segregated from other paths. As previously discussed, this is an intentional aspect of their design. Due to the relatively clear demarcation of park connectors and the relatively few interaction points they have with other paths, our models perform well on the park connector, with ViT and SegFormer achieving IoU scores in excess of 97.0%. Similarly, bicycle paths also perform well, with their distinctive maroon colour enabling them to be well recognised by the various models (see Figure 1).

5.3.1 Analysis of improvements against baseline

Notably, the improvement in mIoU scores seen in transformer models seem to be at least partially due to an improvement in segmentation ability in the pedestrian path and shared path classes. The two classes see an average improvement of 26.1% and 17.4%, compared to an improvement of less than 10.0% for the other classes. This is likely due to the ability of transformer models to detect longer range context via the self-attention mechanism. The pedestrian path and shared paths are made out of the same concrete material, and often appear in similar contexts. Shared paths are distinguished only by dotted red lines along their borders (see Figure 6). Transformer models were likely able recognise this, and segment pixels which are far away from the dotted red borders correctly. On the other hand, the FCN, which has a relatively limited ability to detect longer range context, mistook two classes for each other more frequently. This is especially so near the middle of each of the paths, where they seem virtually identical if only the immediate surrounding visual context is considered, since the dotted red borders would not be present. Figure 5 illustrates the impact of this by comparing predictions across the various models. The FCN baseline tends to confuse the pedestrian and shared paths more than the transformer models.

5.3.2 SegFormer road class performance

The performance of the SegFormer model on the road class is an outlier (see Table 4). Its road class IoU of 60.2% lags the other transformer models significantly, and is only marginally better than baseline. This is particularly notable because it outperforms the other transformer models on most of the other classes. We analysed the predicted masks to gain a better understanding of why this might be the case.

Figure 6. Closeup images of a typical shared path (left) and pedestrian path. Note the similarities, with the distinguishing feature being dotted red markings along the edges of the shared path.

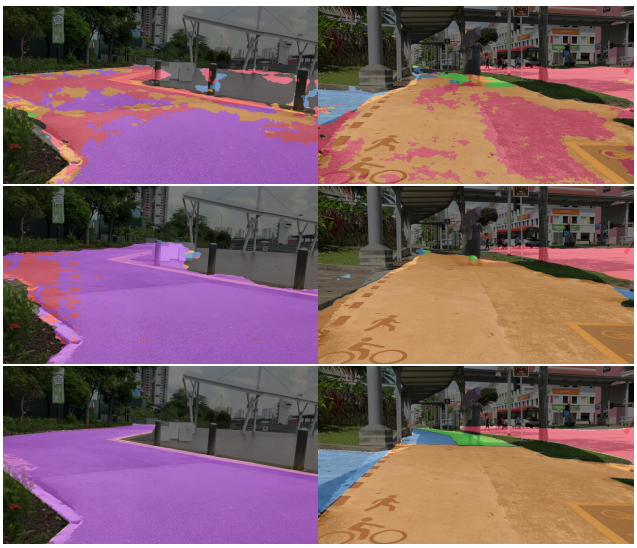

Figure 7. Comparison of two samples where SegFormer performs significantly worse on the road class (in red) than other transformer models. From first row to the third row: SegFormer, ViT, the ground truth

We hypothesise that the main reason for the performance disparity is the limited and coarser training data for roads. During the annotation process, roads were deemed as less important than the other paths. This is because cyclists and other users of cycling paths spend little time on the road. As such, we annotated the class in a coarser manner, such that other objects in the environment such as trees, lampposts, and vehicles were included in their masks. This likely advantaged the other transformer models, which made rougher predictions at 224×224 resolution compared to the 1024×1024 resolution that the SegFormer model makes predictions at. Qualitatively, Figure 7 demonstrates that the SegFormer model is more likely to segment small patches of pixels as being from a particular class when com-

Model	Background	Bicycle Path	Park Connector	Pedestrian Path	Road	Shared Path	mIoU
FCN	95.1	80.2	90.9	49.3	60.2	62.8	73.1
ViT	95.9	88.2	97.8	76.4	74.7	81.1	85.7
Swin	96.7	87.9	95.9	72.1	73.9	75.6	83.7
SegFormer b1	93.2	88.5	97.1	78.6	61.7	83.5	83.8

Table 4. Intersection over union scores for each model

pared to the other transformer models. This results in better edge detection that other methods in general, but also results in a tendency to perform poorer on roughly annotated classes like roads. In our analysis, we found that the SegFormer model tends to over-predict the road class, misidentifying park connectors and shared paths as roads. With finer annotations, newer models such as SegFormer would likely overcome this issue and be able to outperform the other older methods that we have experimented with.

6. Conclusions and future work

In this paper, we introduced a new dataset for semantic understanding of the cycling path network in Singapore. Given the network effect of these new paths, usage is projected to increase. With a better semantic understanding of such paths, mechanisms and tools could be developed for improved safety and autonomous use. We have illustrated that it is possible to gain a good semantic understanding of the paths via current semantic segmentation algorithms, with the ViT model achieving an mIoU of 85.7%. Improvements can be made to the segmentation of roads, which have an IoU of 70.1%. This can largely be attributed to the relatively coarse annotations for the road class.

Future extensions could involve collecting data different parts of Singapore, which may have different geographical and design features for cycling paths. In addition, the set of annotations currently made on the dataset could be expanded. Specifically, other objects commonly seen in the cycling context, such as people, riders, vehicles, lampposts and greenery could be added. The annotations for the road class can also be refined. Finally, since the segmentation of cycling paths is by definition a real-time and mobile endeavour, more research on the use of mobile friendly models such as MobileNetV2 in comparison with the baselines we have established in this paper is also a meaningful next step.

References

[1] Safe Distance @ Parks: how AI replaced eye power for crowd counting — tech.gov.sg. https://www.tech.gov.sg/media/technews/safe-distance-at-nparks, 2021. [Accessed 01-06-2024].

[2] New cycling paths to be built in 7 towns in Singapore — channelnewsasia.com. https://www.channelnewsasia.com/singapore/new-cycling-paths-7-towns-singapore-3200591, 2023. [Accessed 01-06-2024].

[3] H. Bao, L. Dong, S. Piao, and F. Wei. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021.

[4] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020.

[5] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017.

[6] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[7] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[8] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Region-based convolutional networks for accurate object detection and segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 38(1):142–158, 2015.

[9] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[10] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[11] D. Jennings and M. Figliozzi. Study of sidewalk autonomous delivery robots and their potential impacts on freight efficiency and travel. *Transportation Research Record*, 2673(6):317–326, 2019.

[12] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár. Microsoft coco: Common objects in context, 2015.

[13] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo. Swin transformer: Hierarchical vision transformer

using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[14] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[15] G. Neuhold, T. Ollmann, S. Rota Bulo, and P. Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *Proceedings of the IEEE international conference on computer vision*, pages 4990–4999, 2017.

[16] A. Pani, S. Mishra, M. Golias, and M. Figliozzi. Evaluating public acceptance of autonomous delivery robots during covid-19 pandemic. *Transportation research part D: transport and environment*, 89:102600, 2020.

[17] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library, 2019.

[18] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

[19] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015.

[20] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[21] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pages 10347–10357. PMLR, 2021.

[22] R. Wightman. Pytorch image models. `https://github.com/huggingface/pytorch-image-models`, 2019.

[23] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, and A. M. Rush. Huggingface's transformers: State-of-the-art natural language processing, 2020.

[24] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *CoRR*, abs/2105.15203, 2021.

[25] Y. Zhang, Z. Jin, R. Kumari, C. Seah, and T. Chua. Measuring the physical profile and use of park connector network in singapore using deep learning and big data analytics. page 5972, 12 2018.

[26] Y. Zhang, G. X. Ong, Z. Jin, C. M. Seah, and T. S. Chua. The effects of urban greenway environment on recreational activities in tropical high-density singapore: A computer vision approach. *Urban Forestry & Urban Greening*, 75:127678, 2022.

[27] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017.

[28] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 633–641, 2017.