AutoLTS: A Computer Vision Approach to Assessing Cycling Stress

Bo Lin, [1] **Timothy C. Y. Chan,** [1] **Shoshanna Saxe** [2]

[1] Department of Mechanical and Industrial Engineering, University of Toronto
[2] Department of Civil and Mineral Engineering, University of Toronto
{blin, tcychan}@mie.utoronto.ca, s.saxe@utoronto.ca

Abstract

Cycling stress assessment, which quantifies cyclists' perceived stress imposed by the built environment and motor traffics, increasingly informs cycling infrastructure planning and cycling route recommendation. However, currently calculating cycling stress is slow and data intensive. We propose a deep learning model to support accurate, fast, and large-scale cycling stress assessments for urban road networks based on street-view images. We apply contrastive learning to learn compact and informative image embeddings that facilitate the prediction performance of the proposed model. On a dataset of 39,153 road links collected in Toronto, Canada, our initial results demonstrate the effectiveness of our model and the value of using image data in the absence of high-quality road geometry and motor traffic data.

1 Introduction

Safety and comfort concerns have been repeatedly identified as major factors that inhibit cycling uptake in cities around the world. A range of metrics (Callister and Lowry 2013; Furth, Mekuria, and Nixon 2016; Huertas et al. 2020) have been proposed to quantify cyclists' perceived stress imposed by the built environment and motor traffics. These metrics are predictive of cycling behaviors (Imani, Miller, and Saxe 2019; Wang et al. 2020) and accidents (Chen et al. 2017), and thus have been applied to support cycling infrastructure planning (Lowry, Furth, and Hadden-Loh 2016; Gehrke et al. 2020; Chan, Lin, and Saxe 2022) and route recommendation (Chen et al. 2017; Castells-Graells, Salahub, and Pournaras 2020). However, calculating these metrics typically requires high-resolution road geometry and motor traffic data, such as motor traffic speed and volume, the locations of on-street parking, and the presence/type of cycling infrastructure on each road link. The practical challenge of collecting accurate and up-to-date data hinders the broader application of cycling stress assessment and other tools building on it.

In this paper, we propose a deep learning model to assess the cycling stress of urban road networks. From a dataset of 39,153 road links collected in Toronto, Canada, our model learns to automate the assessment of the level of traffic stress (LTS) metric (Furth, Mekuria, and Nixon 2016) based on street-view images. As illustrated in Figure 1, road links are classified into four classes, i.e. LTS1–4, corresponding to the cycling suitability of four types of cyclists. LTS1/2 and LTS3/4 are, respectively, referred to as low-stress and high-stress hereon because LTS2 corresponds to the cycling stress tolerance for most adults (Dill and McNeil 2016). LTS has been applied in studies to investigate the connectivity (Lowry, Furth, and Hadden-Loh 2016; Kent and Karner 2019) and equity (Tucker and Manaugh 2018) of urban cycling networks and to evaluate cycling interventions during the COVID-19 pandemic (Lin, Chan, and Saxe 2021). Our model can facilitate timely, accurate, and large-scale assessments of cycling stress because up-to-date street-view images are easy to access via software such as Google Street

(a) LTS1

(b) LTS2

(c) LTS3

(d) LTS4

Figure 1: Example street-view images with the four LTS labels: LTS1 roads are comfortable for all cyclists including children, LTS2 roads are comfortable for most adults, LTS3 roads are for "enthused and confident" cyclists, LTS4 roads are for "strong and fearless" cyclists.

View API, which publishes over 16 million kilometers of street-view images in over 80 countries (Raman 2017). Our model has the flexibility to integrate available link features as model inputs, which helps enhance the model's prediction accuracy. While we focus on LTS for demonstration, our approach can be generalized to any cycling stress metrics.

Our initial results demonstrate the effectiveness of computer vision approaches in cycling stress assessment. Using only street-view images, our model achieves an LTS prediction accuracy of 65.49% and a high/low-stress prediction accuracy of 88.01%. By integrating the street-view images with link features that are relatively easy to access, such as speed limit and the number of lanes, our model achieves an LTS prediction accuracy of 80.90% and a high/low-stress prediction accuracy of 90.43%.

2 Method

2.1 Data Collection

Training and testing our model required: i) road network topology, ii) ground-truth LTS labels associated with the road links, and iii) street view images that clearly present the road links. We collected all the data in Toronto, Canada due to our collaboration with the City of Toronto, which granted us access to detailed and accurate road network and motor traffic data for LTS calculations. Once trained, our model can be used to perform cycling stress assessments for other cities' road networks. Next, we introduce our data sources and the data collection processes in detail.

Road network topology. We retrieved the centerline road network from the Toronto Open Data Portal (City of Toronto 2020). The network was stored as an ArcGIS shape file where a road link was defined as a road segment that connects two adjacent road intersections. Geospatial coordinates of both ends of each link were presented. We excluded links where cycling is legally prohibited, e.g. expressways and railways. The final network consisted of 59,554 road links.

LTS Labels. The LTS calculation required detailed link-level data, such as the number of lanes, the presence/type of cycling infrastructure and on-street parking, and average motor traffic speed and volume during morning rush hours on weekdays. We collected these data from the sources summarized in Lin, Chan, and Saxe (2021) and calculated the LTS for each link following Furth, Mekuria, and Nixon (2016) and Imani, Miller, and Saxe (2019).

Street-view images. We collected street view images using the Google StreetView API. We chose not to collect images for links that were shorter than 50 meters because a significant portion of those images typically presented adjacent road links and or intersections that may have different LTS labels. For each of the remaining links, we collect one image using the geospatial coordinate of its mid-point and a camera angle calculated based on the coordinates of its two ends. We manually examined the collected images to ensure that every image clearly presented the associated link. If a link failed the human screening, the image was re-collected manually when possible. Images were missing for links where driving is prohibited, such as trails and narrow local passageways, resulting in a loss of 1,298 images.

Our image dataset consisted of 39,153 high-quality street view images, with 41.4%, 35.2% 12.6%, and 10.8% of them labeled as LTS1–4, respectively. Images were represented as 320 $\times$ 320 tensors of 3 (RGB) channels. We performed a random 70/15/15 train-validation-test split.

2.2 Model Architecture

We propose a deep learning model that predicts the LTS of a road link based on its street-view image and link features that are available. As illustrated in Figure 2, our model consists of three modules:

- **Image embedding.** This module extracts useful information from the street view image and represents it as a 64-dimensional vector. We implement this module with a ResNet-50 (He et al. 2016) encoder followed by two fully connected layers.
- **Link-feature embedding.** This module allows us to incorporate link features when they are available. Performing link feature embedding prevents the prediction module from being dominated by the image embedding, which is higher dimensional compared to the original link feature vector. We implement this module with a fully connected layer.
- **Prediction.** This module takes as inputs the average of the image embedding and the link-feature embedding and outputs a four-dimensional vector representing the probability of the link being classified as LTS1–4, respectively. We implement this module with a fully connected layer followed by a soft-max layer.

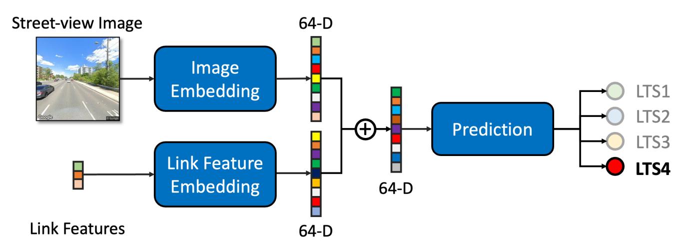

Figure 2: Model architecture.

We provide more details on model architecture in Appendix A.2. Next, we discuss our training methods, focusing on the image embedding module.

2.3 Model Training

Supervised learning. Perhaps the most straightforward way is to train the three modules simultaneously to minimize the cross-entropy loss of the LTS prediction. While the resulting model yields promising prediction performance (Section 4), our model experienced severe overfitting issues due to the challenge of learning a complex mapping from a relatively small dataset. Motivated by recent successes in contrastive learning (He et al. 2020; Chen et al. 2020; Khosla et al. 2020), we hypothesize that training the image embedding module on an auxiliary task beforehand and freezing it during the training of the other two modules would help to reduce the model complexity and thus facilitates learning on the small dataset. Next, we briefly introduce two contrastive learning approaches, one self-supervised and one supervised, that we experimented with in our initial study.

Self-supervised contrastive learning. We use the MoCo framework (He et al. 2020) to train the encoder f on a pretext task where the encoder learns to position "similar" images close to each other in the embedding space. Specifically, we create another encoder g that has the same structure as f and whose parameters are updated using the momentum update function (He et al. 2020) as we train f. Given a batch of N images $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^N$, a stochastic augmentation module is applied twice to each image $\mathbf{x}_i \in \mathcal{X}$ leading to two image views $\bar{\mathbf{x}}_i$ and $\tilde{\mathbf{x}}_i$. Let $\bar{\mathbf{z}}_i = f(\bar{\mathbf{x}}_i)$ and $\tilde{\mathbf{z}}_i = g(\tilde{\mathbf{x}}_i)$ denote the embedding of the two views. The encoder f is trained to minimize the InfoNCE loss (Oord, Li, and Vinyals 2018):

$$L^{\text{self}} = -\frac{1}{N}\sum_{i=1}^{N} \log \frac{\exp(\bar{\mathbf{z}}_i^\intercal \tilde{\mathbf{z}}_i/\tau)}{\sum_{k\in\mathcal{K}} \exp(\bar{\mathbf{z}}_i^\intercal \tilde{\mathbf{z}}_k/\tau)} \quad (1)$$

where τ is a temperature hyper-parameter, $\mathcal{K}$ is a queue of length l that consists of image embeddings generated by g so far. The queue is initialized randomly and updated with the new embeddings $\tilde{\mathbf{z}}_i$ for $i \in [N]$.

Supervised contrastive learning. While using the image embeddings generated by MoCo enhances the prediction performance of our model (Section 4), it does not consider the LTS labels $\{y_i\}_{i=1}^n$ associated with street-view images. Inspired by the supervised contrastive learning loss proposed by Khosla et al. (2020), we adapt the MoCo framework to train the encoder f to minimize the following loss:

$$L^{\text{sup}} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j\in\mathcal{K}_i} \log \frac{\exp(\bar{\mathbf{z}}_i^\intercal \tilde{\mathbf{z}}_j/\tau)}{\sum_{k\in\mathcal{K}} \exp(\bar{\mathbf{z}}_i^\intercal \tilde{\mathbf{z}}_k)/\tau)} \quad (2)$$

where $\mathcal{K}_i = \{k \in \mathcal{K} : y_i = y_k\}$ are indices of views in the queue that have the same LTS label as view $\bar{\mathbf{x}}_i$ for $i \in [N]$.

3 Experiments

3.1 Experiment Setups

We evaluated our model on three scenarios where the deep learning model learns to predict LTS of road links based on

1. Street-view images
2. Street-view images, number of lanes, and speed limit.
3. Street view images, road type (e.g. major/minor arterial, laneway, trail), and presence of cycling infrastructure.

The design of these scenarios were informed by the data collection challenges we encountered in Toronto. The number of lanes and speed limit of each road link were accessed via Open Data Canada (Government of Canada 2020). Road types and the presence of cycling infrastructure on road links were available via Open Data Toronto (City of Toronto 2020). However, since the two data platforms used different base maps, combining features from these two sources took considerable manual effort, echoing the data collection challenges in many other cities.

Computational setups are detailed in Appendix A.1.

3.2 Baselines

To demonstrate the value of using image data, in scenarios where link features were available, we used the Classification and Regression Tree (CART) and Random Forest (RF) that predict LTS only using link features as baselines. Tree models were chosen as most link features were categorical.

To showcase the value of contrastive learning, we compared the predictive power of the learned image embeddings against image embeddings obtained from ResNet-50 (He et al. 2016) pre-trained on the ImageNet (Deng et al. 2009).

3.3 Evaluation Metrics

Given a test set of N_{test} links, let $\hat{y}_i$ denote the predicted LTS of link i. We considered the following evaluation metrics.

- LTS Prediction Accuracy (Acc)

$$\text{Acc} = \frac{1}{N_{\text{test}}}\sum_{i=1}^{N_{\text{test}}} \mathbb{1}[y_i = \hat{y}_i] \quad (3)$$

- High/Low-Stress Prediction Accuracy (H/L Acc)

$$\text{H/L Acc} = \frac{1}{N_{\text{test}}}\sum_{i=1}^{N_{\text{test}}} \mathbb{1}[h(y_i) = h(\hat{y}_i)] \quad (4)$$

  where h is a function that convert an LTS label to high ($= 1$) or low-stress ($= 0$).

- False Low-Stress Rate (FLR)

$$\text{FLR} = \frac{\sum_{i=1}^{N_{\text{test}}} \mathbb{1}[h(\hat{y}_i) = 0]}{\sum_{i=1}^{N_{\text{test}}} \mathbb{1}[h(y_i) = 1]} \quad (5)$$

- False High-Stress Rate (FHR)

$$\text{FHR} = \frac{\sum_{i=1}^{N_{\text{test}}} \mathbb{1}[h(\hat{y}_i) = 1]}{\sum_{i=1}^{N_{\text{test}}} \mathbb{1}[h(y_i) = 0]} \quad (6)$$

- Average False High/Low-Stress Rate (AFR)

$$\text{AFR} = \frac{\text{FLR} + \text{FHR}}{2} \quad (7)$$

Acc and H/L Acc measured the overall prediction performance. FLR, FHR, and AFR considered the fact that the dataset is imbalanced with a higher portion being low-stress.

4 Results

Table 1 compares the prediction performance achieved by our model using different training methods for the image embedding module versus baseline models based solely on link features when available.

Street view images are valuable assets for cycling stress assessments. Our model achieved an LTS prediction accuracy of 65.49% and a high/low-stress accuracy of 88.01% only using street-view images. When link features were available, incorporating street-view images led to an increase of 1.84%–12.61% in overall prediction accuracy with little to no increase in AFR. By combining street-view images with the speed limit and the number of lanes, our

	Feature					Evaluation Metric				
Encoder/Model	Image	Road	Infras.	Speed	#Lanes	Acc (%)	H/L Acc (%)	FLR (%)	FHR (%)	AFR (%)
Res50	✓					**65.49**	**88.01**	**28.87**	6.60	**17.74**
Res50-ImgNet	✓					59.24	85.63	41.62	**5.68**	23.65
MoCo	✓					63.31	86.24	36.83	6.40	21.62
SupMoCo	✓					64.29	86.82	31.13	7.46	19.30
CART		✓	✓			57.50	89.87	8.17	10.76	9.47
RF		✓	✓			57.52	89.89	**8.03**	10.78	**9.41**
Res50	✓	✓	✓			68.45	89.46	27.89	**5.01**	16.43
Res50-ImgNet	✓	✓	✓			68.16	90.55	20.14	6.22	13.18
MoCo	✓	✓	✓			69.66	90.57	19.93	6.09	13.01
SupMoCo	✓	✓	✓			**70.13**	**91.44**	14.37	6.71	10.54
CART				✓	✓	79.02	88.12	41.55	**1.10**	21.33
RF				✓	✓	79.06	89.12	41.55	**1.10**	21.33
Res50	✓			✓	✓	78.24	88.97	**23.31**	7.12	**15.25**
Res50-ImgNet	✓			✓	✓	79.70	89.24	39.51	1.59	20.55
MoCo	✓			✓	✓	80.37	89.94	35.21	2.04	18.63
SupMoCo	✓			✓	✓	**80.90**	**90.43**	29.23	3.30	16.27

Table 1: Prediction performance of the proposed deep learning model versus baseline methods on the test set ($N_{\text{test}} = 5{,}873$). The three blocks (from top to bottom) correspond to scenarios 1, 2, and 3, respectively. Res50 indicates the supervised learning method, and Res50-ImgNet represents using image embeddings obtained from a pre-trained model. MoCo and SupMoCo indicate self-supervised and supervised contrastive learning, respectively.

best-performing model achieved a prediction accuracy of 80.61% and a high/low-stress accuracy of 90.21%. Such a model can be useful for cycling infrastructure planning and route recommendation tools that do not require the granularity of four LTS categories and focus solely on the difference between high- and low-stress links.

Pre-trained image embedding modules help to improve the prediction performance when link features are available. In scenarios two and three, using image embeddings obtained from MoCo and SupMoCo improved overall prediction accuracy by 1.21%-2.13% and 1.68%-2.66%, respectively. Using a general-purpose image embedding module (Res50-ImgNet) did not lead to a consistent improvement in prediction accuracy across the two scenarios. In scenario one where only image data were used as predictive features, supervised learning (Res50) presented the best prediction performance in four out of the five evaluation metrics.

5 Discussions and Future Research

While promising, our results highlight several challenges. First, as presented in Figure 3, our initial models are not effective in distinguishing LTS3 links from LTS2/4 links. This is because, by definition, the main difference between LTS3 and LTS2/4 links lies in motor traffic speed, which is not readily available in street-view images. This issue was rediscovered when visualizing the embeddings obtained from MoCo and SupMoCo (see Appendix B.1) and it persists even after incorporating the link speed limit as a feature. Such confusions have important practical implications. For example, labeling an LTS3 link as LTS2 may lead to a recommended route that exceeds cyclists' stress tolerance and results in increased risks of cycling accidents. This challenge may be tackled by training a separate network to predict the real motor traffic speed/volume using the data collected in Toronto. Once developed, the outputs from this network can be used as inputs to the current prediction module. We are also working on a contrastive learning approach to better separate LTS3 images from the rest in the embedding space.

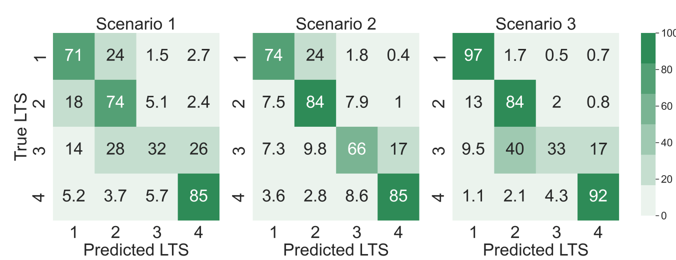

Figure 3: Confusion matrices of the model that achieves the highest prediction accuracy in each scenario. Values are normalized over true LTS labels (rows).

Second, our prediction model does not consider the structure of urban road networks. Road links adjacent to each other generally share similar motor traffic conditions and cycling infrastructure and thus have similar LTS labels. Augmenting the current prediction model with spatial information may help to regularize the predictions, leading to more accurate and interpretable results. Such regularizations can be incorporated via adapting the contrastive learning loss and or performing post-processing for the LTS predictions.

A Experiment Details

A.1 Computational Setups

All deep learning models were implemented using PyTorch 1.7.1 in Python 3.6.0 with 32 GB of CPU RAM. Supervised learning approaches were implemented on a single P100 GPU with 12 GB of memory. Contrastive learning approaches were implemented with a single RTX6000 GPU with 24 GB of memory.

A.2 Model Architecture Details

The image embedding module are implemented with a ResNet-50 encoder. We replace the last fully connected layer with two fully connected layers with output sizes being 128 and 64, respectively. The link feature embedding module was implemented with a fully connected layer whose input size depends on the dimensionality of the feature vector and the output size equals 64. The prediction module is implemented with a fully connected layer whose input and out sizes are set to 64 and 4, respectively. All fully connected layers in the image embedding and link-feature embedding modules use ReLU as the activation function.

A.3 Training Details

For supervised learning approaches, we initialize the encoder with the ImageNet-pretrained weight provided in PyTorch. All images are resized to 224×224 to fit with the input dimensionality of the ResNet-50 encoder. During training, all images undergo random-resized crop, color jittering, random greyscale, random horizontal flip, and normalization before being fed into the encoder. Only resize and normalization are applied to the validation and test data. We use the SGD optimizer with an initial learning rate of 0.01, a weight decay of 0.0001, and a mini-batch size of 128. The model is trained for 100 epochs. The model that achieves the lowest validation loss is selected for evaluation on the test set.

For MoCo, we also initialize the encoder with ImageNet-pretrained weights. In addition to the augmentations applied in supervised learning, training images also undergo the random Gaussian Blur as used by Chen et al. (2020). We use the SGD optimizer with an initial learning rate of 0.03, weight decay of 0.0001, and batch size of 128. The queue size l is set to 6,400 such that the queue fits into the memory of a single RTX6000 GPU. We follow He et al. (2020) to set the temperature hyper-parameter $\tau = 0.07$ and the momentum coefficient to 0.999. When generating image embeddings, each mini-batch is divided into four chunks which are encoded sequentially in order to address the information leakage issue noted by He et al. (2020) on a single GPU. The encoder is trained for 200 epochs. Once trained, we adopt the hyper-parameters used for supervised learning to train the prediction and link-feature embedding modules.

For SupMoCo, we adopt the same hyper-parameters as used for MoCo. According to our preliminary analysis, applying all the augmentations as used for MoCo results in unconverging training loss. We thus only apply the random crop, random greyscale, random horizontal flip, and the Gaussian blur for SupMoCo.

B Additional Results

B.1 Embedding Visualizations

Figure 4 visualizes the image embeddings obtained from MoCo and SupMoCo in a 2-dimensional space. When using MoCo, images with different LTS labels nested with each other in the embedding space since their LTS labels are not considered in the contrastive learning process. When using SupMoCo whose goal is to position images with the same LTS label close to each other in the embedding space, images clearly form two clusters corresponding to LTS1 (square), and LTS2 (triangle). However, LTS3 and LTS4 images are not well separated, reflecting the challenge of distinguishing LTS3 links from others solely based on street-view images.

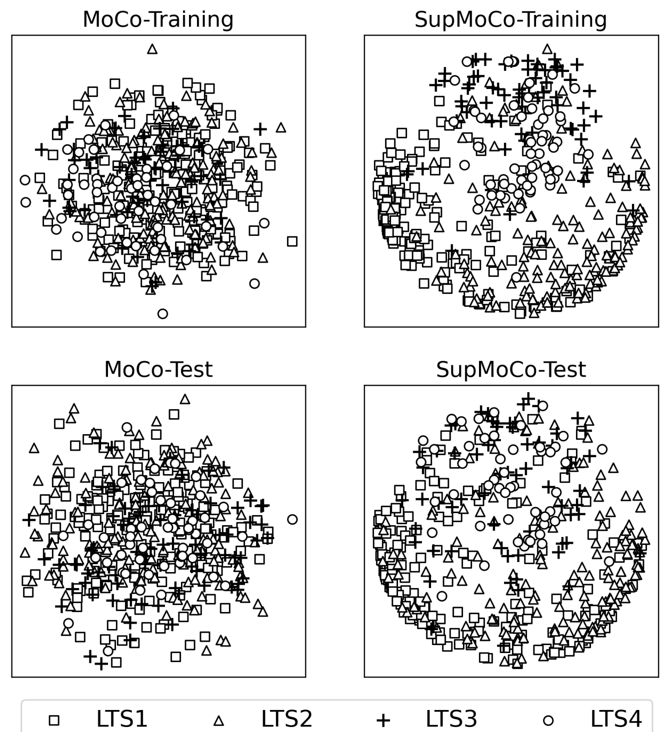

Figure 4: Two-dimensional visualizations of the image embeddings obtained from MoCo and SupMoCo. Image embeddings are projected to the 2-dimensional space using the Principal Component Analysis. We present the visualization of 500 randomly sampled images from the training and test sets, respectively, for clear presentation. Each marker represents one street-view image. Images with LTS1–4 labels are coded with square, triangle, plus sign, and circle, respectively.

Acknowledgments

This research is supported by funding from the City of Toronto. Resources used in preparing this research were provided, in part, by the Province of Ontario, the Government of Canada through CIFAR, and companies sponsoring the Vector Institute.

References

Callister, D.; and Lowry, M. 2013. Tools and strategies for wide-scale bicycle level-of-service analysis. *Journal of Urban Planning and Development*, 139(4): 250–257.

Castells-Graells, D.; Salahub, C.; and Pournaras, E. 2020. On cycling risk and discomfort: urban safety mapping and bike route recommendations. *Computing*, 102(5): 1259–1274.

Chan, T. C. Y.; Lin, B.; and Saxe, S. 2022. A Machine Learning Approach to Solving Large Bilevel and Stochastic Programs: Application to Cycling Network Design. *arXiv preprint arXiv:2209.09404*.

Chen, C.; Anderson, J. C.; Wang, H.; Wang, Y.; Vogt, R.; and Hernandez, S. 2017. How bicycle level of traffic stress correlate with reported cyclist accidents injury severities: A geospatial and mixed logit analysis. *Accident Analysis & Prevention*, 108: 234–244.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, 1597–1607. PMLR.

City of Toronto. 2020. City of Toronto Open Data. https://www.toronto.ca/city-government/data-research-maps/open-data/. Accessed: 2020-09-15.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 248–255. Ieee.

Dill, J.; and McNeil, N. 2016. Revisiting the four types of cyclists: Findings from a national survey. *Transportation Research Record*, 2587(1): 90–99.

Furth, P. G.; Mekuria, M. C.; and Nixon, H. 2016. Network connectivity for low-stress bicycling. *Transportation Research Record*, 2587(1): 41–49.

Gehrke, S. R.; Akhavan, A.; Furth, P. G.; Wang, Q.; and Reardon, T. G. 2020. A cycling-focused accessibility tool to support regional bike network connectivity. *Transportation Research Part D: Transport and Environment*, 85: 102388.

Government of Canada. 2020. Government of Canada Open Data. https://open.canada.ca/en/open-data. Accessed: 2020-09-15.

He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9729–9738.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770–778.

Huertas, J. A.; Palacio, A.; Botero, M.; Carvajal, G. A.; van Laake, T.; Higuera-Mendieta, D.; Cabrales, S. A.; Guzman, L. A.; Sarmiento, O. L.; and Medaglia, A. L. 2020. Level of traffic stress-based classification: A clustering approach for Bogotá, Colombia. *Transportation Research Part D: Transport and Environment*, 85: 102420.

Imani, A. F.; Miller, E. J.; and Saxe, S. 2019. Cycle accessibility and level of traffic stress: A case study of Toronto. *Journal of Transport Geography*, 80: 102496.

Kent, M.; and Karner, A. 2019. Prioritizing low-stress and equitable bicycle networks using neighborhood-based accessibility measures. *International Journal of Sustainable Transportation*, 13(2): 100–110.

Khosla, P.; Teterwak, P.; Wang, C.; Sarna, A.; Tian, Y.; Isola, P.; Maschinot, A.; Liu, C.; and Krishnan, D. 2020. Supervised contrastive learning. *Advances in Neural Information Processing Systems*, 33: 18661–18673.

Lin, B.; Chan, T. C. Y.; and Saxe, S. 2021. The impact of COVID-19 cycling infrastructure on low-stress cycling accessibility: A case study in the City of Toronto. *Findings*, 19069.

Lowry, M. B.; Furth, P.; and Hadden-Loh, T. 2016. Prioritizing new bicycle facilities to improve low-stress network connectivity. *Transportation Research Part A: Policy and Practice*, 86: 124–140.

Oord, A. v. d.; Li, Y.; and Vinyals, O. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Raman, A. 2017. Cheers to Street View's 10th birthday! https://www.blog.google/products/maps/cheers-street-views-10th-birthday/. Accessed: 2022-11-28.

Tucker, B.; and Manaugh, K. 2018. Bicycle equity in Brazil: Access to safe cycling routes across neighborhoods in Rio de Janeiro and Curitiba. *International Journal of Sustainable Transportation*, 12(1): 29–38.

Wang, K.; Akar, G.; Lee, K.; and Sanders, M. 2020. Commuting patterns and bicycle level of traffic stress (LTS): Insights from spatially aggregated data in Franklin County, Ohio. *Journal of Transport Geography*, 86: 102751.