Locating Urban Trees near Electric Wires using Google Street View Photos: A New Dataset and A Semi-Supervised Learning Approach in the Wild

Artur Andre A. M. Oliveira
Institute of Mathematics and Statistics
University of São Paulo
Visual Informatics Group
University of Texas at Austin
arturao@ime.usp.br

Zhangyang Wang
Visual Informatics Group
University of Texas at Austin
atlaswang@utexas.edu

Roberto Hirata Jr.
Institute of Mathematics and Statistics
University of São Paulo
hirata@ime.usp.br

Abstract

Vegetation is desirable in most urban spaces, but its management is not easy, mainly the intersection between trees and sidewalks, or trees and electric wires. This work presents a method to automatically detect the latter using ground-level images instead of aerial images. Real-world ground-level urban images are cheap to collect, but they may be hard to label and classify because neural networks tend to be overconfident, and manually labeling thousands of images may be cumbersome and unfeasible. We propose using Focal Loss to calibrate an overconfident neural network and the use of the training protocol Noisy Student to lessen the burden of manually labeling images. Our results show that these methods improve the results over the Cross-Entropy loss, and the confidence levels of the predictions can be used in an Active Learning system to improve the overall accuracy.

Figure 1. Ignition started due to a short-circuit in electrical wires close to trees' branches during a storm. Image from: Robert Lawton (https://commons.wikimedia.org/wiki/File:Crossed_wires.JPG), "Crossed Wires", https://creativecommons.org/licenses/by-sa/2.5/legalcode

1. Introduction

The presence of aerial electrical wires alongside trees on sidewalks may cause several problems, ranging from the disruption of electric distribution, electrocution [17] and even the start of wildfires [15] as shown in Fig. 1. Therefore, it is crucial to properly manage the vegetation to avoid the issues mentioned above.

Usually, the identification of trees near electrical wires is performed manually (e.g., through a citizen report), and an automatic system to detect trees close to electrical wires could decrease the necessity and inaccuracy of person reports. Currently, some proposals that map tall vegetation close to electrical wires (or assess the risk of their interaction) are based on aerial images [3, 10, 21, 23]. For instance, Wanik et al. [21] propose a LIDAR-based approach to directly assess the risk of an outage during a storm in locations where tall vegetation and overhead power lines are close.

We propose a method to automatically detect trees and wires intersection using ground-level images in this work. Several papers assess urban features from ground-level images, i.e., urban imagery as Google Street View (GSV)

ones [1], or user collected images with smartphone cameras [2, 7]. These images allow one to analyze various aspects of an urban environment, for instance, physical aspects like urban vegetation distribution [13], or the relationship between the environment (e.g., greenery, littering) and health or safety outcomes [11, 16]. Another practical application for systems based on such images is the detection of issues in the city like quantification of walls degraded by graffiti [20]. The acquisition of urban images by citizens is cheap, accessible, as demonstrated and made available by crowd-sourced platforms Kartaview and Mapillary [2, 19], and aligned with the citizen science efforts [5].

The detection of intersections of trees and electrical wires is also possible from ground-level images, and to the best of our knowledge, no other work has done that. Figure 2a presents an example of part of a GSV image. The right side of the image clearly shows some electric cables between two large boughs and some electric cables touching one of the boughs from below. However, due to the complexity of urban scenes, sometimes it may be hard to assess the intersection automatically. For instance, a human being can infer that the cables intersect the other tree (Fig. 2a, middle left of the image), but the intersection is not clear anymore. Besides that, several other challenges are present when dealing with GSV images as an imbalance of classes (presence of intersection, or not), different appearance variations (Fig. 2b shows a difficult to deal glare condition), hardness to annotate at scale (annotation is cumbersome and time consuming), and others.

We first collected 50.000 urban ground-level images to build a dataset for training, testing, and validating our method. It is the first public dataset to model trees and wires intersection classification. Using the semi-supervised Noisy Student training protocol [22] we avoid labeling the whole dataset, and we could address the natural imbalance of the dataset using Focal Loss (FL) [14] as the cost function. Our experiments shows an excellent performance of the proposed method. In our experiments training with FL achieved 83.7% and 78.8% recall rates with respect to classes with and without intersections respectively, and an overall test accuracy of 55.3%.

Our contributions are three-fold:

- A public dataset with urban images, eleven thousand labeled for trees, electrical wires, and intersections.
- A method based on FL and Noisy Student training protocol.
- An experimental comparison between FL and the vanilla Cross-Entropy loss (CE) cost functions for this classification problem

2. Method

We review the Noisy Student method and then review the Focal Loss before presenting our approach to classify trees and wires intersection.

The Noisy Student training protocol [22] is a semi-supervised approach created to leverage large amounts of not annotated data. The basic idea is to train a neural network model, or *teacher*, with the labeled data, apply the teacher model to generate *pseudo-labels* for both the labeled and the unlabeled data and then use the whole dataset with the pseudo-labels to train another neural network model, the *student network* model. Besides being trained only with pseudo-labels (rather than ground truth labels), the student network is also required to include input noise, that is, data augmentation over the input images implemented with RandAugment [4] and also model input, that is, Stochastic Depth [9] and Dropout [18]. Figure 3 illustrates the noisy student model/protocol.

The Focal Loss cost function [14] is a generalization of the vanilla Cross-Entropy loss to deal with the class imbalance in object detection due to a significantly large number of easy negative cases in comparison with positive cases. Formally it is defined as:

$$FL(\hat{y}_t) = -\alpha_t(1-\hat{y}_t)^{\gamma} log(\hat{y}_t)$$

where α_t is the weight assigned for the class of sample t, γ is the focusing parameter, y, and $\hat{y}$ are respectively the true and the predicted label by the model, and $\hat{y}_t$ is defined as:

$$\hat{y}_t = \begin{cases} \hat{y} & \text{if } y = 1 \\ 1-\hat{y} & \text{otherwise} \end{cases}$$

The intuition behind the Focal Loss function is that by increasing the focusing parameter, γ, samples correctly classified will have a smaller impact on the cost function (i.e., the factor $(1-p_t)^{\gamma}$ will be small), while samples misclassified will have a larger impact. In this sense, one may interpret correctly classified samples as "easy" samples or, on the other hand, more difficult samples will have a larger weight.

We partially used the Noisy Student protocol, i.e., we trained a teacher model using only the labeled images, then we used the first-generation teacher to generate pseudo-labels for all the unlabeled images. Then we trained a noisy student model using the labeled and the unlabeled images now labeled with pseudo-labels generated by the first-generation teacher. After that, we used the noisy student model to generate a new set of pseudo-labels, replacing the pseudo-labels generated by the previous generation teacher. Thus effectively, the noisy student model becomes a second-generation teacher model. We repeat this iteration of teachers generating new pseudo-labels over the same set

(a) (b)

Figure 2. Images from Google Street View. (a) Trees in contact with electrical wires. (b) Bad visibility due to sun glare.

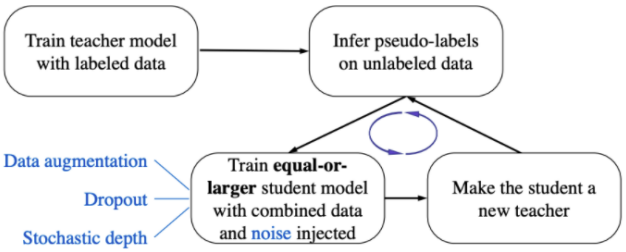


Figure 3. The Noise Student training protocol. Image from [22].

of unlabeled images and combining pseudo-labeled images with the training labeled images to train new students until convergence. We use Early Stopping as our convergence criteria for both the first teacher and every subsequent student network. The Noisy student protocol is not the same as the original proposal because we do not use any form of data augmentation or model noise. Besides that, the model's size is kept constant from one teacher generation to the next.

Algorithm 2 describes formally the procedure used to train a student in pseudo-code. D_t, D_v correspond to the training and validation datasets, respectively. X_t, X_v, X_u are sets of images for the training, validation, and unlabeled datasets. Notice that both X_t and X_v have the corresponding label sets Y_t and Y_v, but the labels for the unlabeled images X_u are pseudo-labels $\hat{Y}_u$ generated by a trained network ϕ_t referred here as the teacher network. We define $maxPatience$ (Alg. 1 and 2, lines four and five, respectively) to 10, which means that the network converged if it can not improve the accuracy over the (labeled) validation dataset for more than ten epochs. The first teacher network ϕ_t is trained using the same procedure as the student (see 1), but using only the training and validation datasets D_t and D_v.

Algorithm 1: The first teacher training procedure

Input: $D_t = \{X_t, Y_t\}, D_v = \{X_v, Y_v\}$
Output: ϕ_t
Initialize ϕ_t;
$Patience = maxPatience$;
$D = D_t$;
Let $lastAcc$ be the accuracy of ϕ_t over D_v;
while $Patience > 0$ **do**
  Optimize ϕ_t using D;
  Let $newAcc$ be the accuracy of ϕ_t over D_v;
  if $lastAcc < newAcc$ **then**
    $lastAcc = newAcc$;
    $Patience = maxPatience$;
  else
    $Patience = Patience - 1$;
  end
end

3. Experiments

The proposed dataset has 50k images, each with a resolution of $640x640$ pixels, and the authors label 11k of them following a simple label protocol described later. We split the 11k labeled images into training, validation, and test datasets, each with 5k, 3k, and 3k images.

3.1. Collecting and labeling the dataset

We first collected the metadata of Google Street View (GSV) images, which contains an identifier that allows the request for an image through the GSV API. This platform allows one to select arbitrary regions in any country around the world and, for each region, collect all the metadata for images made available by GSV. Besides an identifier for each image, the metadata also contains its location, timestamp, and the vehicle's forward direction. A panorama is also associated with each metadata, i.e., a collection of images that compose a panoramic image. Since our objects of interest are the trees and the electric wires, we selected images corresponding to four directions for each panorama: the vehicle's forward and backward directions and the side window view direction.

Algorithm 2: The student training procedure

Input: $D_t = \{X_t, Y_t\}, D_v = \{X_v, Y_v\}, X_u, \phi_t$
Output: ϕ_s
$\hat{Y}_u = \phi_t(X_u)$;
$D_u = \{X_u, \hat{Y}_u\}$;
$Patience = maxPatience$;
$D = D_t \bigcup D_u$;
Let $lastAcc$ be the accuracy of ϕ_s over D_v;
while $Patience > 0$ **do**
  Optimize ϕ_s using D;
  Let $newAcc$ be the accuracy of ϕ_s over D_v;
  if $lastAcc < newAcc$ **then**
    $lastAcc = newAcc$;
    $Patience = maxPatience$;
  else
    $Patience = Patience - 1$;
  end
end

Labeling strategy for challenging images Urban scenery images can be complex, and one of the difficulties is the depth of information lost in 2D images. In our particular case, some intersection instances may be ambiguous to determine if wires appear before or contact the branches of a tree.

Due to this complexity, different human annotators may judge the same picture as having distinct classification labels. To capture possible ambiguities, we propose a new label for challenging images, so an annotator may either consider an image as a *positive* (an intersection is present), negative (no intersection), *challenging* (an intersection may be present, or not), or having no trees. Formally we define four possible labels for an image, these are:

- Trees w/ int.: Trees with an intersection- the images in this class have one or more trees, and the intersection between the branches and the wires is visible;
- Trees maybe w/ int.: In this case, both the trees and the wires are visible, but it is challenging to tell if they are in contact or not;
- Trees w/o int.: Images in this class have trees but no visible wires;
- No trees: There are no visible trees in this class.

3.2. The neural network model

In our experiments, we used the network architecture MobileNetV3 [8] pretrained with the ImageNet dataset [12] both for teacher and student networks. We also experimented with all the family of EfficientNet networks (i.e. B0 to B7) as proposed in [22], both using the same architecture across all generations of teachers and student networks and also using for each new generation an architecture that has the same number or a bigger number of parameters. Surprisingly, the best results in terms of test accuracy were obtained by using the MobileNetV3 in all generations. Furthermore, we also experimented using random initialization for the network weights and observed that for every tried architecture using weights pre-trained with the ImageNet dataset provided better results.

3.3. Computing the confidence of the prediction

A possible approach to estimate the confidence of the network over its predictions is to take the maximum value out of the vector obtained by computing the softmax of the output of the network. Unfortunately, this approach leads to uncalibrated networks that are overconfident in their predictions. We observed that a student network, trained with uncalibrated pseudo-labels, overfits to the training data, and moreover, the pseudo-labels To mitigate this issue, we apply the temperature scaling strategy proposed in [6]. Using this strategy we want to find new confidence values for the predictions of the network such that samples correctly classified have a higher confidence value, and samples misclassified have a smaller confidence value.

For completeness we briefly describe here the strategy to find the optimal temperature scale factor T to calibrate the predictions of a trained network. Let $x \in X_v$ be a sample from the validation dataset. In this section we consider the true label $y_x \in Y_v$ of sample x to be a one-hot encoded vector and the prediction $\hat{y}_x$ a probability vector, computed as the softmax $\sigma(.)$ of the output logits z_x of a trained network ϕ, that is

$$\hat{y}_x = \sigma(z_x) = \sigma(\phi(x))$$

Let the confidence of the network for this prediction be

$$q_x = \max_k \hat{y}_x^{(k)},$$

that is, the confidence is the maximum value in the predicted vector $\hat{y}_x$. Notice that (k), for $k \in K = [0, 1, 2, 3]$, in the exponent indicates a position in the prediction vector corresponding to the predicted class for sample x. Dividing z_x by a temperature scale factor T and then taking its softmax value produces a scaled prediction vector $\hat{y}'_x = \sigma(z_x/T)$. When $T \to \infty$ the values of the vector $\hat{y}'_x$ approach $\frac{1}{|K|}$, that is, every class will have nearly the same probability, thus the confidence of the network for this prediction is the nearly same for every class. In the opposite case, when $T \to 0$, $\hat{y}_x$ will approach a vector where every value is zero, except for $\hat{y}_x^{(k)}$ which will be one, that is, the confidence of the network for this prediction will be 1. Since the temperature scale factor is applied over the logits, before taking the softmax, the final prediction is kept unchanged,

Figure 4. Challenging samples which may be confusing (to an human annotator) to determine if there are intersections between trees and wires or not. Images from GSV.

thus the temperature scaling doesn't change the accuracy of the network.

The optimal temperature scale T_{opt} to calibrate the predictions of the network as proposed by [6] is obtained by minimizing the Cross-Entropy loss between the scaled predictions vector y'_x and the corresponding true label y_x for sample x from the validation dataset, formally:

$$T_{opt} = \min_T \mathbb{E}[\sum_k^K y_x^{(k)} log(\hat{y}_i'^{(k)})]$$

$$= \min_T \mathbb{E}[\sum_k^K y_x^{(k)} log(\sigma(z_x/T)^{(k)})]$$

finally the new calibrated confidence q'_x is defined as:

$$q'_x = \max_k \sigma(z_x/T_{opt})^{(k)}$$

In our experiments we compute T_{opt} only after the training of a network is done. Then we use this trained network as a teacher to compute pseudo-labels (using the unlabeled dataset) for a new student. These pseudo-labels are then calibrated with T_{opt} and only then they are used to train the next student network generation together with the labeled training dataset.

We performed experiments by training a sequence of teachers/student networks using either Cross-Entropy or Focal loss as cost functions. We experimented with different values for the hyperparameters γ and α of the FL (the weighting vector and the focusing parameter, respectively) [14]. We report the results for $\gamma = 2$ and $\alpha = [0.5, 0.1, 0.2, 0.2]$. Note that the weights vector α has an assigned weight for each of the classes described at Section 3.1.

4. Results and Discussion

This section analyzes the accuracy and confidence levels trade-off observed by choosing either FL or CE as cost functions. We characterize the confidence level of a prediction as the highest probability in a prediction vector outputted by the network for a given image. We observed that training with the FL rather than Cross-Entropy provides lower confidence levels for both challenging and incorrectly classified images. In Figs. 5a to 5g we present in the vertical axis of the upper (bottom) half of the graph the number of correctly (incorrectly) classified images. The horizontal axis of these same graphs describes the confidence with which these samples were classified. Note that the vertical axis is log-scaled for better visualization. Each bar corresponds to a confidence bin, including samples predicted with confidence higher or equal to the value of the previous bin and strictly smaller than the current bin.

Figures. 6a and 6b present the confusion matrices for the classification results obtained by a network trained with the Cross-Entropy and with the FL cost functions, respectively. These matrices contain the classifications for the test dataset, and the labels are: (0) Trees w/ an Intersection, (1) Trees maybe w/ Intersection, (2) Trees w/o Intersection, and (3) No trees. Table 4 shows the recall rates, over the test dataset, for the network trained with the Cross-Entropy (CE) and Focal Loss (FL) cost functions. The last row in Table 4 (2+3) is the union of these classes, that is, images without intersections regardless of the presence of trees.

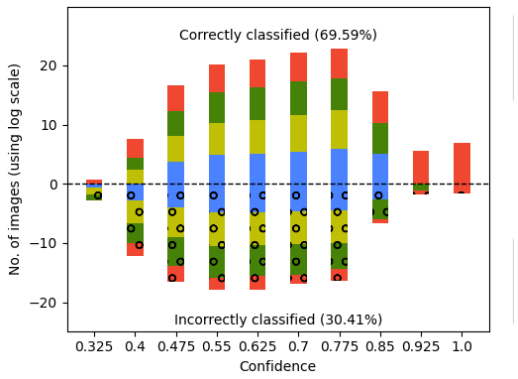
-	Recall CE	Recall FL
(0)	66.5%	83.7%
(1)	64.1%	28.3%
(2)	58.0%	71.6%
(3)	22.7%	22.7%
(2+3)	63.7%	78.8%

As observed in each pair of graphs (in Figs. 5a to 5g), the challenging images (i.e., "trees maybe w. int.") are classified (in)correctly across all levels of confidence when the CE is used. In contrast with FL, confusing images are classified with maximum confidence of 55%. Furthermore, most incorrectly classified images with FL also have a low confidence score.

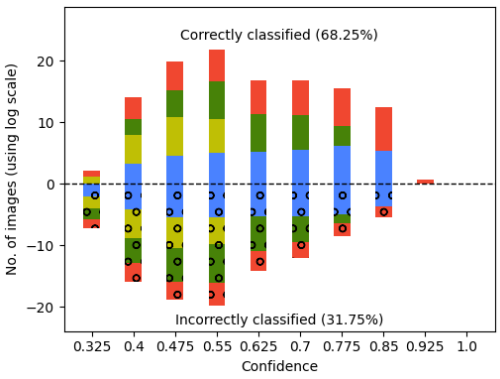
On the other hand, the network trained using the FL achieved a recall of 83.7% for class (0) of positive cases, while the recall for the same class obtained with CE was of 66.5%. For practical applications like the detection of trees entangled with wires the higher recall rate for positive cases is better because an human agent can simple discard false positives. We conjecture that the high recall for the other classes obtained by vanilla CE was due to equal weights assigned to each class. In a future work we will explore the use of active learning based on the confidence of the network for a prediction, that is, given an image, its location and the direction of the camera, if the prediction for this image has a low confidence then another images from a close location where the camera is pointing to the point as before can be collected and used together to improve the final prediction. The combined classification of both images could be used to determine the presence or absence of a tree in contract with electrical wires with higher accuracy and confidence.

5. Conclusion

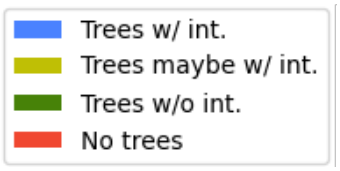
The observed recalls for non-challenging images (i.e., 83.7% for the images with intersections and 71.6% for images without intersections) provide evidence that a system to detect the intersections is feasible and could be implemented with the current technology available. Using the Noisy Student training protocol enables the usage of many images requiring only the manual labeling of a small fraction of them.



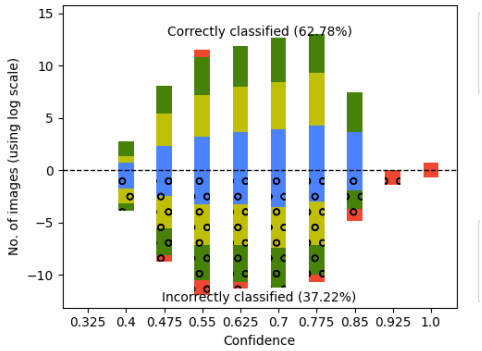
(a) Results for the training partition with CE.



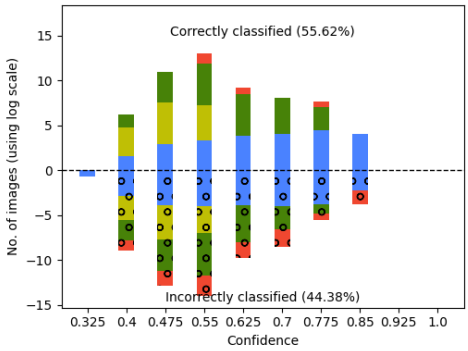
(b) Results for the training partition with FL.



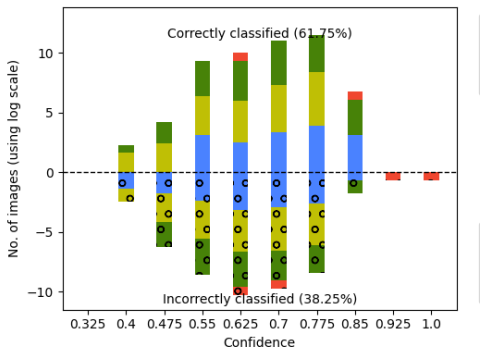
(c) Legend for the colors in the graphs.



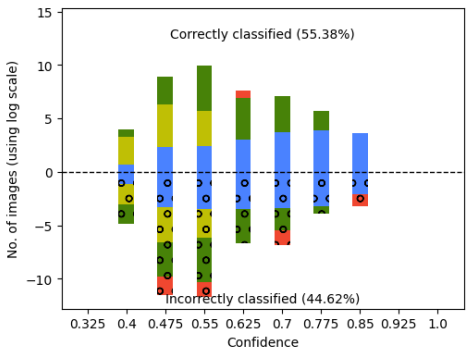
(d) Results for the validation partition with CE.



(e) Results for the validation partition with FL.

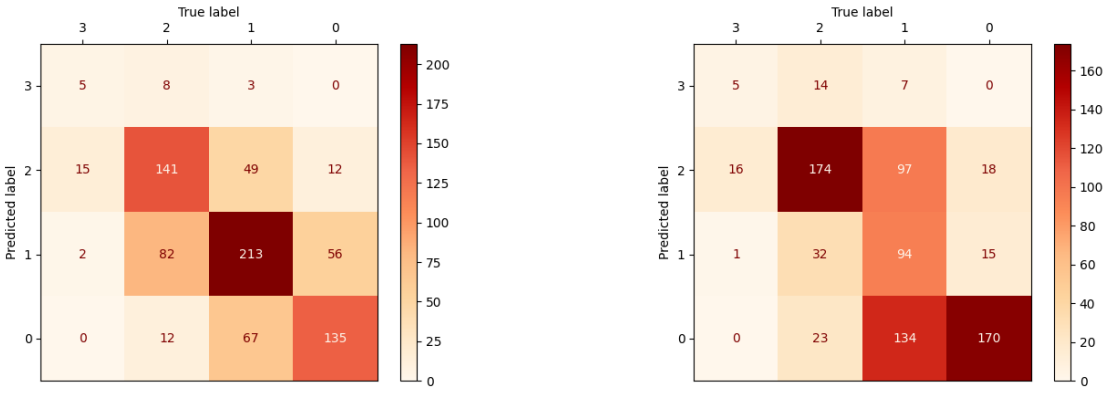


(f) Results for the test partition with CE.



(g) Results for the test partition with FL.

Figure 5. Results for the comparisons between the networks trained with Cross-Entropy or Focal Loss at the training, validation and test partitions.



(a) (b)

Figure 6. (a) Confusion matrix for training with FL over the test dataset. (b) Confusion matrix for training with CE over the test dataset.

(a) (b) (c)

Figure 7. Images from Google Street View. (a) Image misclassified by both networks as 'Trees w/o int'. The confidence for prediction dropped from 59% with CE to 49% with FL. (b) Image misclassified with CE as 'maybe with intersection' and correctly classified with FL as 'without intersection'. (c) Image misclassified with FL as 'without intersection' but with a low confidence of 40%.

Due to the complexity of images in the urban scene, several images may be hard to classify, even for a human annotator. An artificial class can represent such challenging images. Training with the vanilla CE cost function resulted in a network with an overall test accuracy of 60.5%, while the same network architecture trained with FL cost function had an overall test accuracy of 55.3%. Nevertheless, FL is better suited than the CE cost function to keep predicted challenging confidence and incorrectly classified images low. Future works consider the coupling of an active learning system based on the confidence levels of the predictions.

References

[1] Dragomir Anguelov, Carole Dulong, Daniel Filip, Christian Frueh, Stéphane Lafon, Richard Lyon, Abhijit Ogale, Luc Vincent, and Josh Weaver. Google street view: Capturing the world at street level. *Computer*, 43(6):32–38, 2010. 2

[2] Filip Biljecki and Koichi Ito. Street view imagery in urban analytics and gis: A review. *Landscape and Urban Planning*, 215:104217, 2021. 2

[3] Simon Clode and Franz Rottensteiner. Classification of trees and powerlines from medium resolution airborne laserscanner data in urban environments. In *Proceedings of the APRS Workshop on Digital Image Computing (WDIC), Brisbane, Australia*, volume 21, 2005. 1

[4] Ekin D. Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V. Le. Randaugment: Practical data augmentation with no separate search. *CoRR*, abs/1909.13719, 2019. 2

[5] M V et al. Eitzel. Citizen science terminology matters: Exploring key terms. citizen science: Theory and practice. 2(1):1:1–20, 2017. 2

[6] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, pages 1321–1330. PMLR, 2017. 4, 6

[7] Nan He and Guanghao Li. Urban neighbourhood environment assessment based on street view image processing: A review of research trends. *Environmental Challenges*, 4:100090, 2021. 2

[8] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1314–1324, 2019. 4

[9] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q. Weinberger. Deep networks with stochastic depth. *CoRR*, abs/1603.09382, 2016. 2

[10] Y Jwa, G Sohn, and HB Kim. Automatic 3d powerline reconstruction using airborne lidar data. *Int. Arch. Photogramm. Remote Sens*, 38(Part 3):W8, 2009. 1

[11] Basma Korchani and Kaouthar Sethom. Real-time littering detection for smart city using deep learning algorithm. In *2020 International Conference on Communications, Signal Processing, and their Applications (ICCSPA)*, pages 1–5, 2021. 2

[12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012. 4

[13] Xiaojiang Li and Carlo Ratti. Mapping the spatial distribution of shade provision of street trees in boston using google street view panoramas. *Urban Forestry & Urban Greening*, 31:109–119, 2018. 2

[14] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 2, 6

[15] Jun Ma, Jack CP Cheng, Feifeng Jiang, Vincent JL Gan, Mingzhu Wang, and Chong Zhai. Real-time detection of wildfire risk caused by powerline vegetation faults using advanced machine learning techniques. *Advanced Engineering Informatics*, 44:101070, 2020. 1

[16] Quynh C. Nguyen, Sahil Khanna, Pallavi Dwivedi, Dina Huang, Yuru Huang, Tolga Tasdizen, Kimberly D. Brunisholz, Feifei Li, Wyatt Gorman, Thu T. Nguyen, and Chengsheng Jiang. Using google street view to examine associations between built environment characteristics and u.s. health outcomes. *Preventive Medicine Reports*, 14:100859, 2019. 2

[17] Lindsey Purcell. Trees and electric lines. https://extension.purdue.edu/extmedia/FNR/FNR-512-W.pdf, 2015. Access Jan 22nd, 2021. 1

[18] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014. 2

[19] ®Grab and OpenStreetCam Contributors. Hello, kartaview! - openstreetmap @ grab. https://blog.improveosm.org/en/hello-kartaview/, 2020. Access Ago 22nd, 2021. 2

[20] Eric K. Tokuda, Roberto M. Cesar, and Claudio T. Silva. Quantifying the presence of graffiti in urban environments. In *2019 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pages 1–4, 2019. 2

[21] D.W. Wanik, J.R. Parent, E.N. Anagnostou, and B.M. Hartman. Using vegetation management and lidar-derived tree height data to improve outage predictions for electric utilities. *Electric Power Systems Research*, 146:236–245, 2017. 1

[22] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10687–10698, 2020. 2, 3, 4

[23] Wei Yao and H Fan. Automated detection of 3d individual trees along urban road corridors by mobile laser scanning systems. In *Proceedings of the International Symposium on Mobile Mapping Technology, Tainan, Taiwan*, pages 1–3, 2013. 1