Generalising Land Use and Land Cover Mapping Across Geographical Regions for Rapid Infrastructure Planning

Mike Boss (mboss), Dario Zehnder (dzehnder), Nando Galliard (gnando),
Jonathan Ehrat (jehrat), Aleksandar Milojevic (amilojevic)
Department of Computer Science, ETH Zurich, Switzerland

Abstract

Having access to a wide variety of labeled data is not always granted and often boosts the generalisation ability of a supervised learning model. To overcome this issue of label-unavailability in the context of a segmentation task for generating land use maps of unlabeled geographical regions, this paper presents a CycleGAN related style-transfer approach to generate synthetic image-label pairs for an unlabeled target domain by leveraging data from a labeled source domain. The synthetic image-label pairs are then used as training data for a segmentation network that is tested on the unlabeled target domain. Our approach was able to boost the visual accuracy of a segmentation algorithm that was applied on an unseen geographical region. Additional time was spent in developing a processing pipeline for high-resolution remote sensing imagery.

1. Introduction

Land use maps are an integral part of infrastructure planning. The process of creating such maps is usually tied to human labour and can involve hand labeling copious amounts of buildings and streets from remote sensing imagery. Due to the lack of immediate financial potential, labeling work is rarely done for cities in developing countries. An alternative approach for generating such land use maps is by using algorithmic, learning-based methods. The ideal outcome of this approach is an algorithm that takes satellite images as input and returns the respective land use maps as an output. A first step towards such a solution could be a supervised segmentation algorithm that learns to recognize buildings and streets from the given images and encodes the results as segmentation masks. To teach such a segmentation algorithm the correct mapping function, it first needs to be trained by providing it with existing pairings of satellite images and their corresponding segmentation masks with pixel-wise label information, i.e. whether the pixel belongs to a building, a street or the background. Furthermore, an ideally trained version of this algorithm would be able to generalize-well across various geographical domains, meaning that it would be able to generate accurate segmentation masks of previously unseen regions.

The so-called generalization ability (generalizability) of a supervised segmentation algorithm is closely tied to the variance in the image-data it was trained on. In the context of remote sensing imagery, variance entails images of different geographical regions (domains), which have distinct characteristics. Conclusively, a segmentation algorithm will perform well on unseen regions if it is able to generalize well, and it is more likely to generalize well, if it was trained on image-data, containing examples of various domains. While free geographical databases, like for example OpenStreetMap [5], provide map references that can be processed into building-and-street segmentation maps, they lack accurate map information for cities in developing countries, as mentioned above. The lack of labeled data for certain cities is thus still an issue, even for supervised, algorithmic approaches, such as image segmentation.

In this paper, we present a cycleGAN [7] based method for generating synthetic labeled data in form of image-label pairs in the context of satellite imagery and corresponding building-and-street-segmentation masks. Furthermore, we show that enriching the training data set of a segmentation algorithm, results in visually more accurate segmentation mask on unseen regions.

We start in section 2 by examining general land use map generation methods, specific methods that can be used for synthetic label generation, as well as the general cycleGAN model. Afterwards, in section 3, we illustrate the preprocessing steps for the satellite image input data. Then, we introduce our cycleGAN, style-transfer based method to generate synthetic image-label pairs for the above mentioned use-case. Section 4, presents the results of our approach as well as the experimental set-up and the evaluation method. Finally, we conclude with section 5, where we give a final summary according to the results and mention potential future work.

2. Related works

In the following paragraphs, we first examine a land use map generation approach that is not concerned with generating synthetic image-label pairs. Then, we present the models, which combined together, constitute our final model. Namely, Unet [6], the model for the segmentation algorithm, the general CycleGAN model, and a more specialized CycleGAN variation, called cyCADA [3].

Topological map extraction. [4] We started out querying general approaches for land use map generation, such as the so-called *PolyMapper* approach for topological map extraction. This approach does not focus on predicting pixel-wise segmentation. Instead, it tries to achieve decent generalizability by predicting buildings and streets directly as polygons. Receiving the satellite image as input, the model first uses a convolutional neural network to extract key points from the image, which are then connected and processed into vector representations for each object. The entire pipeline of this approach is rather involved and eventually achieves the polygon mapping by unifying object detection, instance segmentation and vectorization. After further inspection of this approach, we decided to go into a completely different direction and a more lightweight context, by focusing on a more general solution to the underlying problem. Namely, find a way to boost a basic segmentation algorithm's generalizability, instead of pursuing more complex segmentation approaches.

Unet. [6] *Unet* is a plain, rather lightweight but effective segmentation model. The core of this convolutional network is an encoder-decoder architecture. The contracting path of the model, i.e. the encoder, follows a typical convolutional architecture by applying a series of convolution and max-pooling layers to the input image. While the image dimensions are down-sampled, the number of feature channels increases. This way, the model tries to capture the context of the image, like for example spatial information.
The expanding path of the model, i.e. the decoder, is symmetric to the contracting path. It uses up-sampling layers or transposed convolution layers to expand the spatial dimensions of the down-sampled input and reduce the number of feature channels. Working with a more low-level representation of the initial image, the decoder part focuses more on the low-level features. Furthermore, at each up-sampling step, the output of the lower level is concatenated with the corresponding feature map from the contracting path, consequently combining spatial information of the image with the low-level feature information. Having up-sampled to the final layer, the model uses its final convolution layer to output a segmentation map with the same spatial resolution as the initial input image.

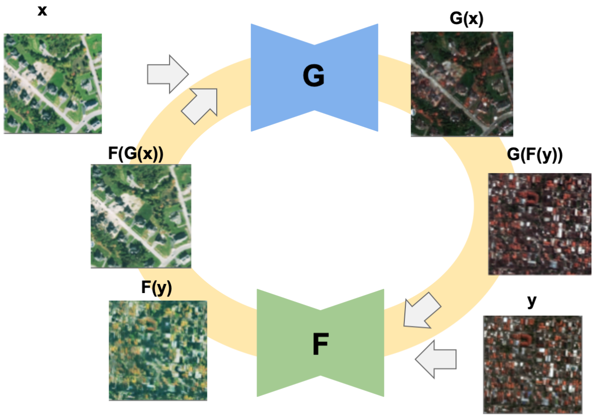

Figure 1. The cycle of a CycleGAN with generators F and G.

There are various versions of the Unet model architecture, very lightweight ones, as well as ones that are more refined. We have used a custom lightweight architecture for the initial experiments and ended up using a more complex version for the final outputs.

CycleGAN. [7] The CycleGAN is used to apply a style-transfer to a labeled source domain image $x \in X$, such that the transferred image keeps the structure of the input image but is stylized in a way that it matches the target domain Y. The transferred image can then be used together with the segmentation mask of the input image to serve as a synthetic image-label-pair belonging to the unlabeled target domain. The CycleGAN architecture consists of two generators, G and F, and two discriminators, D_x and D_y. The generators transform source domain images to target domain images (G) and vice versa (F) while the discriminators try to distinguish generated images from real ones in a min-max-game. Additionally, a so-called cycle-consistency is enforced, such that $x = F(G(x))$ and $y = G(F(y))$. The full objective is explained in [7] and summarized as follows:

$$\begin{aligned}\mathcal{L}(G,F,D_x,D_y) = \\ &\mathcal{L}_{GAN}(G,D_y,F,D_x) \\ &+ \lambda\mathcal{L}_{cycle}(G,F)\end{aligned} \tag{1}$$

where $\mathcal{L}_{GAN}$ is the adversarial loss introduced in [1] and $\mathcal{L}_{cycle}$ ensures the cycle-consistency mentioned above.

CyCADA. [3] CyCADA is a CycleGAN-based model developed by Hoffman et al. It augments the CycleGAN with the addition of unsupervised adversarial adaptation methods. The goal of the model is to predict the label for some target data. Ie. it is able to adapt in the absence of target labels, and is applicable in pixel-level and feature space. A

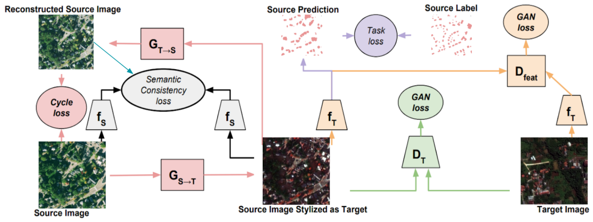

Figure 2. CyCADA extends the CycleGAN architecture for domain translation. The GAN and Cycle loss, indicated in red and green, are the same as in CycleGAN. CyCADA adds a semantic consistency loss, visible in gray, based on a pre-trained segmentation model as well as training losses, visible in purple and orange, for the target segmentation model. We extended the semantic consistency loss with segmentation information of the re-constructed source image, shown in light blue.

shortcoming of such techniques in the past was, that there was no guarantee that the source image stylised as the target preserves the structure or content of the original sample. To encourage the source content to be preserved, CyCADA imposes a cycle-consistency constraint. They introduce another mapping back to reproduce the original sample, thereby enforcing cycle-consistency. For our use case this additional constraint serves to be able to differentiate generally hindering features like added trees on roads or the recoloring of roofs.

3. Method

To present our method, we first elaborate on the data collection a pre-processing step, since it consumed a big part of our time and is an essential first step for our final experiments. Subsequently, we present our final method for generating synthetic image-label pairs for an unlabeled geographical domain.

3.1. Data Collection and Processing

To test the results of our work, we used satellite images of African cities that were provided without any corresponding segmentation masks. To prevent the manual labeling of this data for training purposes, we decided to start with US satellite imagery instead, since it is available in high resolution, includes infrared channels the same way as the given data from less developed cities, and most importantly comes pre-labled through free geographical databases. The decision to use well-labeled, American city images for training purposes helped in testing the generalization ability of our model, since they aren't designed and constructed in the same manner as the African cities we used for our final evaluation.

We generated a random list, containing a variety of east coast cities and loaded the ground truth from Open Street Maps [5] (OSM) while getting the satellite images from Earth Engine [2]. For each city we gathered the satellite images in the bounding area of the ground truth data and split it into 3 km chunks, since it is easier to work with multiple smaller files than one large one. We visualised the OSM data and separated it into a building and a street data set. OSM data does not always contain street width, but always their position, so we manually selected an appropriate general width for the obtained satellite imagery. When visualizing the data we had to make sure to work in the same coordinate reference system (CRS) in all the cases. To improve the performance of the final model, we added variance and permutations to the data. We resized, flipped and normalized the data before we used it in our pipeline. The data frequently includes dead data from outside of the city limits. To remove these chunks with numerous empty spaces we had to add additional scripts to the pre-processing pipeline. Lastly, in edge cases, one of the data sets was incomplete so we removed every chunk without intersections in both sets.

3.2. Training the Generators

Our approach for generating synthetic image-label pairs in the context of land use maps is rather straightforward. We first decide on a geographical region, which has enough and accurate hand-labeled map data available. This is usually the case for wealthy and well-populated regions, like for example the east-coast in the USA, and such map data is freely available in geographical databases like *OpenStreetMap* [5]. After we process the image and label pairs from the geographical database according to the process discussed in the section above, we have a consistent image-label pairs available, for a well-labeled domain, which we call domain X. Now we are able to freely choose another geographical region for our domain Y, independent of available map data for said region, even if there is no map data available at all. We process the satellite imagery of domain Y in such a way that it is consistent in format with the processed color images of domain X. The colored images from both domains are then arbitrarily paired together to form a data set that is used as input to a CycleGAN model. What the CycleGAN model learns, is to apply style transfer to image x from the labeled domain X, such that it visually looks like it belongs to the unlabeled domain Y. Additionally, it does learns the same thing in the other direction. Namely, it learns a mapping of image $y \in Y$ in such a way, that it looks like it belongs to domain X. The result of this procedure are two trained generators, G and F, which are able to apply style-transfer from and to the respective domains, as can be seen in figure 3. For this step any CycleGAN variation can be used, such as the CyCADA model mentioned in section 2.

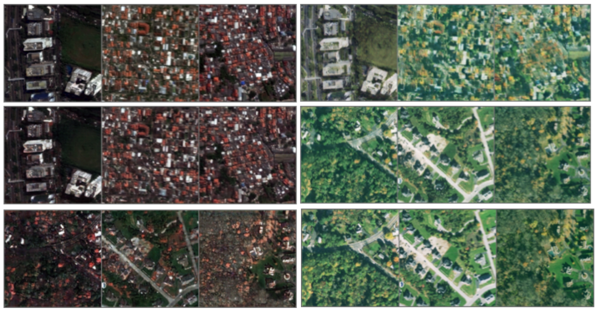

Figure 3. CycleGAN mapping. Domain X is east coast of USA and domain Y is Jakarta. On the left from top to bottom: y, G(F(y)), G(x). On the right from top to bottom: F(y), F(G(x)), x

3.3. Boosting Segmentation Generalizability

The boosting of the segmentation generalizability, i.e. the segmentation performance on an unseen geographical region, of any preferred segmentation algorithm can now be achieved in two different ways. Namely, by generating synthetic image-label pairs or an approach we call *pseudo segmentation*.

3.3.1 Generating synthetic image-label pairs

As the name suggests, we try to boost the generalization ability of a segmentation algorithm by generating synthetic image-label pairs. For this, we take generator G, which maps x to the style of domain Y, and apply style-transfer to a set of images from X, resulting in a image set $G(X)$. Since the style-transfer keeps the structural information of the input image during the mapping, the segmentation mask of the input image x can now be paired together with the resulting image $G(x)$, which looks like it belongs to the unlabeled domain Y. Now we have synthetic image-label pairs $(G(x), seg(x))$, which represent the unlabeled target domain Y, and where $seg(x)$ is the segmentation mask of input image x. These synthetic image-label pairs can now be used to augment the train-data set of any segmentation algorithm, adding more variance to the training procedure. Our assumption is, that having examples in the training set, that resemble more closely to the unseen region, our segmentation algorithm will perform better on said unseen region compared to not having been trained with such a train set augmentation.

3.3.2 Pseudo Segmentation

Another way to make use of the CycleGAN training result, and try to boost the generalizability of a segmentation algorithm, is to take generator F, which maps y to the style of domain X, and involve it directly in the segmentation process. For this, we create a segmentation pipeline, where we take as input an image y from an unlabeled domain Y, and first transfer the style of y to the style of a labeled domain X, resulting in the image $F(y)$. Subsequently, $F(y)$ is used as an input to a segmentation algorithm S_x, that was trained on the labeled domain X. This way, we hope to boost the generalizability of S_x in an implicit manner, providing it with a synthetic image that looks like one from its expert domain.

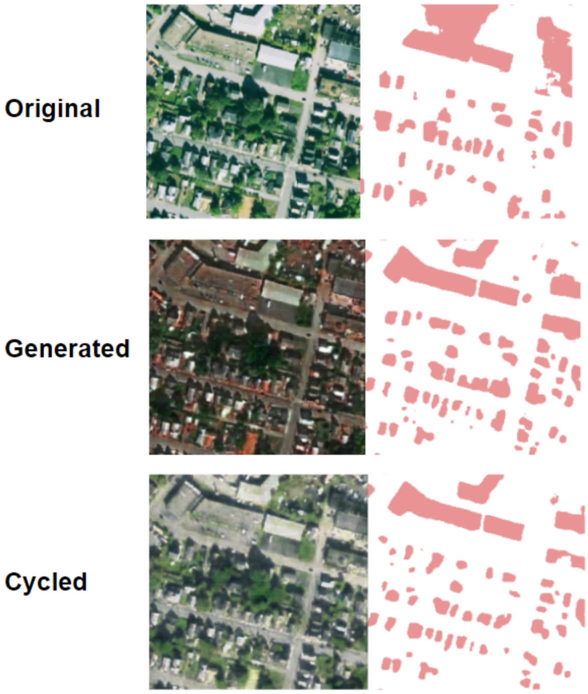

Figure 4. Style transfer and cycling back to source style with CyCADA for building recognition.

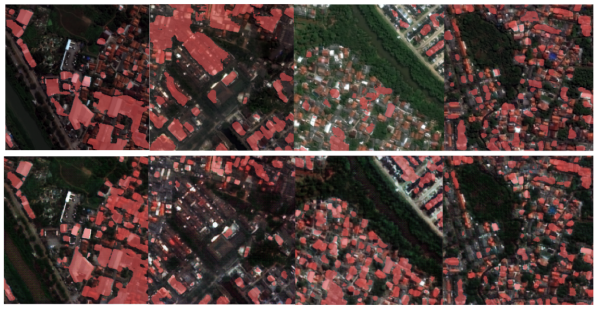

Figure 5. Predictions of the UNet trained on fake Jakarta images *top* and predictions on the re-constructed Jakarta images by CyCADA trained to translate from and to fake Jakarta from real Jakarta images *bottom*

.

3.3.3 Replacing CycleGAN with CyCADA

As mentioned in section 2, CyCADA is an augmentation of CycleGAN for domain translation. Instead of solely relying on the CycleGAN architecture, it uses additional loss information of a pre-trained segmentation model to preserve semantic consistency before and after translation to the target domain, shown in gray in figure 2. We additionally add the re-constructed image to this semantic consistency loss to preserve segmentation after re-construction, indicated in light blue in figure 2. To prevent [**?**] we inject minimal Gaussian noise after before each generator step to mitigate the usage stored information in the translated image by the model. The original model did not produce certain colours from the target images, for example the yellow roofs that are typical in Jakarta. Therefor, we added a jitter element to the generator step $G_{T\to S}$ of the stylized source images to assist it with handling more colour variations. For the same reason, we also trained the UNet with jittered east coast images. The result of this change can be seen in figure 4. After we trained and predicted the fake Jakarta images, we now re-train the UNet on these translated images. We additionally re-train the CyCADA model using the newly trained UNet on to translate from and to Jakarta from the fake Jakarta images. Figure 5 shows in the top row the on fake Jakarta re-trained UNet's predictions and below the predictions by the re-trained CyCADA model. This re-training greatly increases the building recognition visible as the red overlay. The re-trained UNet model over-predicts buildings quite visibly, as shown in figure 7. Re-training the CyCADA model using this UNet removes these artifacts in the reconstructed images, as seen in figure 8. However, certain dense areas also lose predictions, as seen in figure 5 in the second image. The reason for this loss in prediction is not clear, even so predictions that are removed by re-construction are predominantly coarse over-predictions with no individual buildings predicted.

4. Results

In the following paragraphs we present the results of our method. It is important to note that the *Pseudo Segmentation* approach ended up producing unusable results. Therefore, we only show the results of synthetic image-label pair generation, which was able to boost the performance of a segmentation algorithm, by visual evaluation, on an unseen region.

4.1. Experimental Setup

Labeled images from the east coast of the USA will serve as our labeled domain X. This data was downloaded from $OpenStreetMap$ and then processed according to our preprocessing step. As our unlabeled domain Y, we picked the city Jakarta, which is sufficiently distinct from our X domain. The remote sensory images from Jakarta were provided by GIZ and were also processed according to our preprocessing step. Furthermore, we used a Unet segmenation algorithm as our segmentation network.

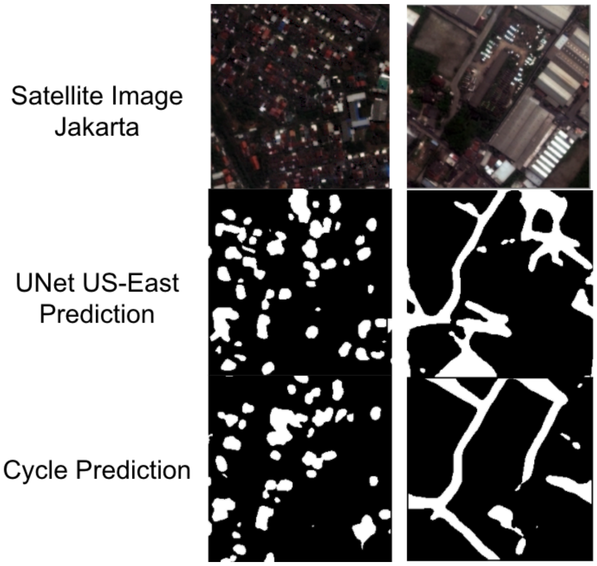

Figure 6. The synthetic label approach via CycleGAN only showed visual improvements for the street segmentation.

4.2. Evaluation Method

In order to evaluate how well each approach is performing in terms of providing synthetic labeled data for an unlabeled target domain, a consistent procedure must be used. Therefore, we trained two instances of the same segmentation network on two different data sets. One instance of our segmentation network was trained on a data set containing only image-label pairs from domain X. The other instance was trained on synthetic image-label pairs form our synthetic label method mentioned in section 3. As our final evaluation, we applied both segmentation instances on the unseen, unlabeled domain Y, i.e. Jakarta. We evaluated the performance according to visual accuracy of the segmentation since we do not have ground truth labels of the unlabeled domain to be used for metric calculations.

4.3. Performance

As can be seen in figure 9, the Unet that was trained on the synthetic images managed to produce visually more accurate road segmentation. The segmentation for buildings did not show any improvements. The building detection was improved with CyCADA and false positives were limited that were a problem with the original prediction as one can see in figure 7. In the end result with CyCADA there an increase of false negatives was noticeable in figure 8 yet the edges proof usable for further utilisation in future work.

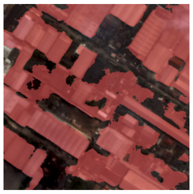

Figure 7. Original prediction. Visible overprediction.

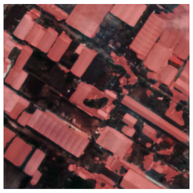

Figure 8. Improvement with CyCADA

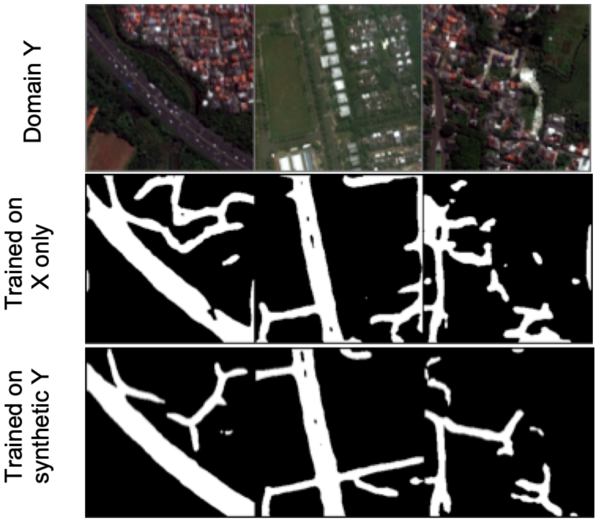

Figure 9. Street segmentation on the unlabeled region Jakarta. Once with a Unet that was trained on the East-Coast region (X), and once with a Unet that was trained on synthetic labels for Jakarta (Y).

5. Conclusions

The main conclusion of our work is that we were able to boost the generalization ability of a segmentation model for road segmentation by construction synthetic image-label pairs of an unlabeled domain. Due to time constraints, as well as a rather time-consuming data-collection step. In the last paragraphs below, we discuss the limitations and future work of both the main approach as well as the data processing.

5.1. Limitations and Future Work

Our approach to boost the generalizability of a segmentation model in the context of land use maps and satellite imagery is rather simple and straight forward. The results of the buildings segmentation were a bit disappointing and not usable at all. We believe that this is due to general poor performance of the segmentation model itself, when it comes to identifying buildings from satellite imagery. Nevertheless, the results from road segmentation on the unseen, unlabeled region were satisfying and show that there is some potential in pursuing this approach further.

All in all, the fact that our approach is not specific on a certain domain, it can be used and combine across various other domains or with more specialized segmentation algorithms, like for example the $PolyMapper$ approach mentioned in section 2. Furthermore, other CycleGAN variations could be used for training the generators. The loss of the CycleGAN model could be manipulated in a way, such that the semantic consistency of the labeled domain image x is considered during training.

5.2. Data gathering

While we were able to boost the generalization ability of a segmentation algorithm when it comes to road segmentation, the big disparity in source and target domain can still lead to more flawed predictions, than a if there were less disparities. If one specific topography is of special interest, one could use a neighbouring country or city with known ground truth as source domain which would lead to better performance for that specific area.

Lastly, our visualization technique used for the ground truth is satisfactory but not flawless. An automated approach for finding street widths would increase reliability and is needed to pursuit this approach further.

References

[1] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. *Generative Adversarial Networks*, volume 27. 06 2014. 2

[2] Noel Gorelick, Matt Hancher, Mike Dixon, Simon Ilyushchenko, David Thau, and Rebecca Moore. Google earth engine: Planetary-scale geospatial analysis for everyone. *Remote Sensing of Environment*, 2017. 3

[3] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei A. Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. *CoRR*, abs/1711.03213, 2017. 2

[4] Zuoyue Li, Jan Dirk Wegner, and Aurélien Lucchi. Topological map extraction from overhead images, 2019. 2

[5] OpenStreetMap contributors. Planet dump retrieved from https://planet.osm.org . https://www.openstreetmap.org, 2017. 1, 3

[6] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015. 2

[7] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2223–2232, 03 2017. 1, 2