Facades and Openings Detection Based on Different Deep Learning Models

Xiaorang Guo, Qunyou Liu, Yiwen Ma
Supervisor: Dr.Alireza Khodaverdian, Dr.Saeed Saadatnejad
Hosting lab: Applied Computing and Mechanics Laboratory, EPFL, Switzerland

***Abstract*—It is obvious that the building types in Google Street View always vary with the regions. How to build a general building detection model that is feasible for different testing datasets becomes a topic worth exploring. In this report, we illustrate the data grabbing and processing strategies that transfer the raw images collected from Google Street View to the usable test dataset with detached buildings inside. Also, we compare two pre-trained building detection models and show the approaches we tried to improve the detection performance.**

***Index Terms*—Machine Learning, Deep Learning, SSD, GAN, image processing**

I. Introduction

In this project, we introduced two deep learning models: SSD_FacadeParsing and Pix2pix. SSD_FacadeParsing is a CNN based model that helps in the facade and opening detection, while Pix2pix is a convolutional GAN based model that detects multiple building features through segmentation.

1) SSD_FacadeParsing: SSD is a method that uses a single deep neural network to detect objects in an image. It discretizes the output space of the bounding box into a set of default boxes, which have different aspect ratios and the ratio of each feature map position. During the prediction period, the network generates a score for the existence of each object category in each default box, and adjusts the box to better match the object shape. In addition, the network combines predictions from multiple feature maps with different resolutions. It can naturally handle objects of various sizes. SSD uses VGG16 to extract function maps. Then, it uses the Conv4_3 layer for objects detection. SSD is relatively simple compared to methods that require object proposals, because it completely eliminates the proposal generation and subsequent or function re-sampling stages, and encapsulates all calculations in a network. This makes the SSD easy to train and can be directly integrated into the system that requires testing components[**1**].

2) Pix2pix: The architecture implements conditional GAN. Unlike normal work, for generator "U-Net"-based Architecture was implemented, and for discriminator convolutional "PatchGAN" classifier, which only penalizes structure at the scale of image patches, was used[**2**][**3**]. Both generator and discriminator use modules of the form convolution-BatchNorm-ReLu. In the discriminator, there are 11 layers adopted, starting and ending with convolution layers, with 3 sets of convolution-BatchNorm-ReLu inside. In the generator, a U-Net-shape architecture is used to avoid the gradient vanishing problem.

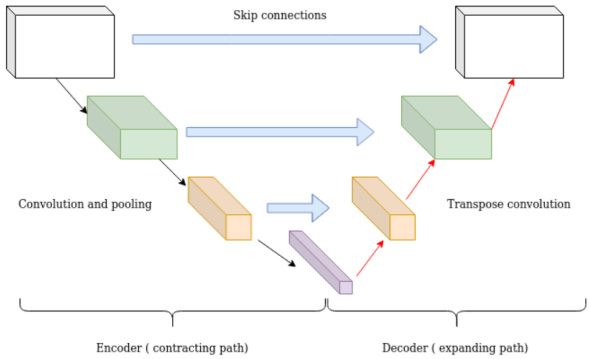

Figure 1. U-Net architectures

Skip connections are added in the general shape of "U-Net"[**4**]. Specifically, the connection is skipped between the i^{th} layer and the n^{th} layer, where n is the total number of layers. Each skip connection only connects all channels of i^{th} layer with channels of n^{th} layer.

Both models are pre-trained with a ground truth dataset which includes mostly perfect detached buildings. The motivation behind the project is to connect the two models with the real street view in Switzerland.

With a raw dataset grabbing directly from Google Street View in Zurich where the building type may be completely different from the training dataset, our goal is to make a dataset that would be suitable for the two models to achieve acceptable performance. We are curious about the difference in the testing results of two models with different deep learning principles. In addition, since we mainly focused on the facade and opening detection, we explored the methods to improve the performance of Pix2pix by merging meaningless features such as doors and balconies.

II. Data Collection

A huge working part of this project is to collect data from Google Street View API, and to make some refinements so that it can be used as valid inputs for the two building-detection models. Firstly, we applied the data downloading function from last year's project 'Building Classification', to grab the images according to the shape-file which contains 10500 locations in Zurich. Since we aim to detect the facades and openings and we do not need the data for the indoor views,

we made some modifications to download only the outdoor views of the building. Also, the locations of some cameras are very close to each other, which may lead to the same building IDs at different longitude and latitude. We made a little change to the code to filter out the same buildings.

Because of the unknowns of the images, we were not sure about the value of the most appropriate heading(the angle of rotation around the vertical axis) and pitch (the angle of rotation around the horizontal axis) that could bring the perfect detached building. In order to collect a properer testing dataset, we need to find a better heading and pitch when grabbing the pictures. To choose the two parameters, we first tried to download 6 pictures per location for several locations, that is, a combination of 3 angles for heading and 2 angles for pitch. Since the shape-file provided by the laboratory includes the coordinates of the center of each building, we were able to get the coordinate to the nearest road by using a snap-to-road function. Then, by calculating the orientation of the required camera perpendicular to the road and the building, we got the heading that makes the images face the building directly. We added to and subtracted from this angle with 30 degrees to get the other two headings and used 0 and 15 degrees respectively as the pitch input.

Figure 2. grabbing 6 images per location and choose the best from it %.

After several experiments, We observed that the pictures with direct heading angle and 0-degree pitch have a better view of the building among all images. With these two value of parameters, the total number of successfully downloaded images is around 6000, 57% of the provided shape-file locations.

III. Data Processing

Since all images are directly collected from google street view, there are a great number of images that do not have any building facades at all or contain buildings surrounded by a mass of noise. We designed three filters to make the testing dataset for our pre-trained models from the raw dataset.

A. Filter by Model Place365

Place365 is a pre-trained CNN model developed by Bolei Zhou and Aditya Khosla from MIT in 2016. It is used for detecting the scene in a picture with more than 400 unique scene categories. We applied Place365 to the raw images from Google Street View and reported a list of all detected classes with their probability. In our preliminary data selection, we set two criteria for detecting if the image contains a building or not.

1) Building Class Detection: We selected all the outdoor building classes as a building list. We sorted the detected classes according to the probability. If none from the top three classes came from the building list, we classified the image as bad image and filter it out.

2) Probability: The probability of the image was set as the highest probability of the detected class that is in the building class list. If the probability was less than a threshold of 0.07, we classified it as a bad image and filter it out.

B. Filter by Green Ratio

After the first filter, although we could filter some non-building images, there were still some pictures that contain the building which is surrounded by a mass of noise, such as the trees or large cars. In the second part, we wrote a function based on CV2, which detecting the green area and filtering the images with a large green ratio.

To make the green detection, we converted the images to NumPy array and checking the color for each pixel. We defined a range of green in HSV from [35,40,40] to [70,255,255] and summed up the number of pixels whose color is in the range. If the ratio that the number of green pixels divided by the number of total pixels is larger than 30%, we deleted this picture from the dataset.

Figure 3. Example of green detection, with a green ratio of 42.2%.

C. Filter by Car Detection

Same as the green filter, we made car detection in order to remove the noisy image with unclear facades. Car Detection was achieved by using the SSD model provided by PyTorch. If the car area ratio was greater than 0.12 and the confidence of detection reported by the SSD model was larger than 70%, we classified it as a bad image because it means the image was covered by a big car.

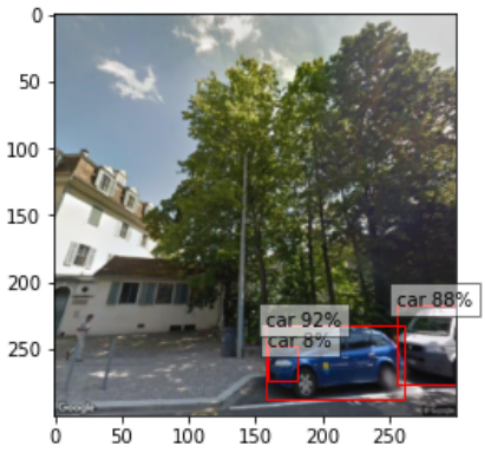

Figure 4. Car detection using SSD model

After adopting this kind of method, about 35.7% data left compared to the original data set. The table below shows the number of images left after each filter and the percentage of the survived images comparing to the number in the last phase.

Filter	None	Place365	Green Ratio	Car
Number	5985	3055	2463	2141
Percentage		0.51	0.81	0.70

IV. Model Implementation and Improvement

For detecting the openings and facades, we applied both deep learning models: SSD_Facade Parsing and Pix2pix on the dataset of street views in Zurich. We aim to compare their performance on the testing dataset in which the building types are most different from those in the training dataset, and to make some adjustment on the model to improve the detection results. In this project, we focused on the improvement of Pix2pix model.

A. SSD_FacadeParsing

SSD_FacadeParsing is the first model that we adopted in our project. We directly applied the model that was pre-trained with a ground truth dataset including 418 images to our google street images. As mentioned, this model can detect different kinds of objects by returning their class probability and the coordinate of detected boxes. The output of the model is the original image with a building box and opening boxes plotted based on the detection results. By using the coordinate of building boxes and window boxes, it is able to compute the area of the facade (yellow box) and the openings (green boxes).

B. Pix2pix

The second model we used is Pix2pix. Same as what we did before, we first tried the pre-trained the model with 400 ground truth images from city center. The images in the training dataset almost contain only the direct view of the facades over the whole image. In contrast, the google street images contain plenty of noise in the background area, which means the buildings always only take partial area of the image. Also, the building types of the training dataset are completely different from the google street view in Zurich (most are not in the city center). Pix2pix is able to detect 12 classes, not only the facades and the openings, but also other features such as balconies, window covers and the doors. All the classes are represented by different colors during segmentation. Thus, the output of the first round test is super noisy because it misclassified background as features and there are a lot of colors shown in one image.

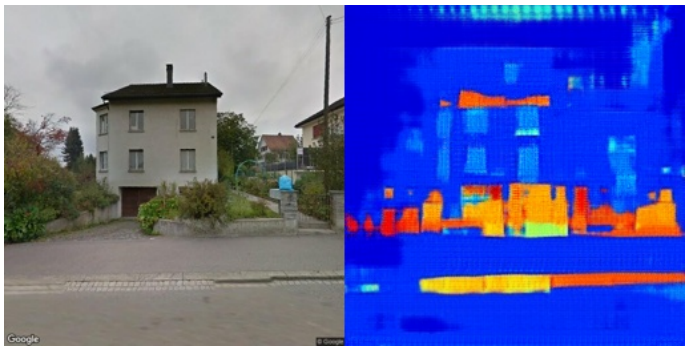

Figure 5. Result with the original pre-trained model: Real Image(left), Output(right)

To solve the problem, we first modified the visualization part of the network to merge the meaningless classes. After analyzing the architecture of the Neural Network, we found that the network divides the test part into three arrays which correspond to RGB colors. The three arrays would be combined together in the end to form the final output image. We modified the visualization part in the last step of combination, to make sure that the network would only output the desired colors.

Since in this phase, we focused on the facade and openings detection, we kept only four colors which are corresponding to 4 classes(background, facades, openings, and covers). The merge process are based on the RGB range: for orange and green detected (facade features), we merged it to the facade; for reddish part (usually roads) and the rest, we merged it to the background; for yellow (stands for cover of openings), we keep it on the image, but in the calculation of the opening/facade ratio later, we counted it into facade.

We re-trained the model with 400 ground-truth images which has been processed by merging the classes for the second time. In this round, the images had less noise, but still had the issue of misclassification. For example, the roads might be recognized as facade because of the lacking of sample images in the training dataset with background. We manually modified the dataset by revising the wrong recognition parts to their corresponding true colors for around 100 pictures and then add them to the training dataset. With around 400 ground truth data and 100 images from google street view, we retrain the model for a third time.

V. Results and Discussion

A. SSD_FacadeParsing

Overall, the output of SSD_FacadeParsing model on the google street view is not ideal. The yellow box could not properly detect the facade area because of the background noise, and the green boxes for the openings were almost completely mis-placed.

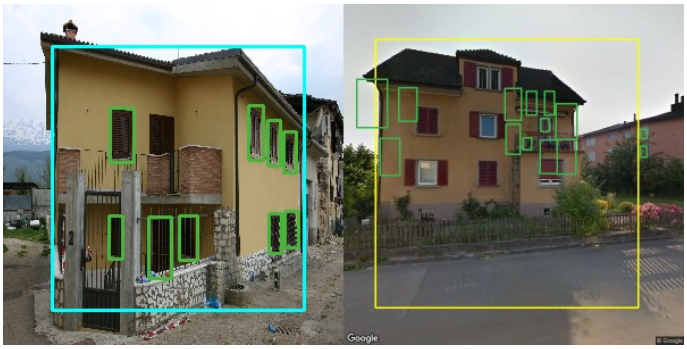

Figure 6. Sample result of the SSD_Facades model: Ground truth image (left) and Output with image from Google Street View(right)

B. Pix2pix

We applied the latest modified model again on the testing dataset, the improvement is obvious compared with the old result. Referring to the color, the dark blue represents the background, blue represents the facades and light blue represents the openings.There is also a yellow color which presents the cover of window, but it is not shown in this picture

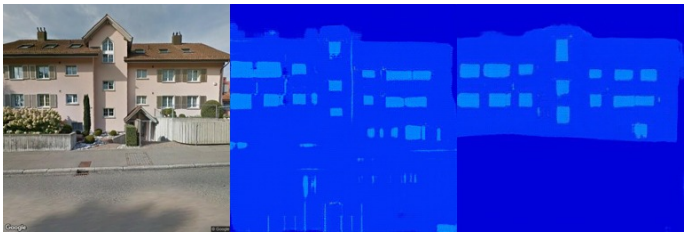

Figure 7. Comparison of old and new results of Pix2pix model: Original image(left), Second round test result(middle), Third round test result(right).

In addition, we recorded the total number of pixels for each class, and reported the Facade/Background ratio and the Openings/Facade ratio. It is obvious that the mean Facade/Background ratio has a significant reduce after the adjustment of the model. Also, there is a slight increase in the Opening/Facades ratio. Both changes symbolize a properer building detection result with the google street view images. We hope these ratio could help with the future building type classification.

	Before	After
Mean Facade/Background Ratio	0.680	0.315
Mean Openings/Facade Ratio	0.159	0.169

VI. Conclusion

To summarize, in this project, we successfully built a data filter system to make a feasible testing dataset from real world Google Street View images. In addition, we made some modification on the model training part of Pix2pix to improve performance. To be more specific, it has a relative clearer detection results for the openings and facades. In contrast, the performance of SSD_FacadeParsing model was not ideal on our dataset. It might be improved by having more fine-tuning on the model, and also by adding more images which include background to the training dataset. For future possible improvements, we suggest to have a look on other building features, such as covers, doors and balconies, and to research in the their influence on the building type detection.

Acknowledgement

We would like to acknowledge the host lab for our project Applied Computing and Mechanics Laboratory (IMAC) and the project mentors Alireza Khodaverdian and Saeed Sa. who provided us the data resources (location shape file in Zurich), and also the advice and guidance through the whole project.

References

[1] Liu, Wei & Anguelov, Dragomir & Erhan, Dumitru & Szegedy, Christian & Reed, Scott & Fu, Cheng-Yang & Berg, Alexander. (2016). SSD: Single Shot MultiBox Detector. 9905. 21-37. 10.1007/978-3-319-46448-0_2.

[2] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016, pp. 770-778, doi: 10.1109/CVPR.2016.90.

[3] P. Isola, J. Zhu, T. Zhou and A. A. Efros, "Image-to-Image Translation with Conditional Adversarial Networks," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, 2017, pp. 5967-5976, doi: 10.1109/CVPR.2017.632.

[4] J. Zhu, T. Park, P. Isola and A. A. Efros, "Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks," 2017 IEEE International Conference on Computer Vision (ICCV), Venice, 2017, pp. 2242-2251, doi: 10.1109/ICCV.2017.244.