PROCESSING OF CRAWLED URBAN IMAGERY FOR BUILDING USE CLASSIFICATION

P. Tutzauer[a] , N.Haala[a]

[a] Institute for Photogrammetry, University of Stuttgart, Germany - (Patrick.Tutzauer, Norbert.Haala)@ifp.uni-stuttgart.de

Commission II, WG II/6

KEY WORDS: Crawling, Scene Understanding, Data Registration, Deep Learning, Building Classification

ABSTRACT:

Recent years have shown a shift from pure geometric 3D city models to data with semantics. This is induced by new applications (e.g. Virtual/Augmented Reality) and also a requirement for concepts like Smart Cities. However, essential urban semantic data like building use categories is often not available. We present a first step in bridging this gap by proposing a pipeline to use crawled urban imagery and link it with ground truth cadastral data as an input for automatic building use classification. We aim to extract this city-relevant semantic information automatically from Street View (SV) imagery. Convolutional Neural Networks (CNNs) proved to be extremely successful for image interpretation, however, require a huge amount of training data. Main contribution of the paper is the automatic provision of such training datasets by linking semantic information as already available from databases provided from national mapping agencies or city administrations to the corresponding façade images extracted from SV. Finally, we present first investigations with a CNN and an alternative classifier as a proof of concept.

1. INTRODUCTION

Over the last few years, there has been a shift in photogrammetry and geoinformation applications from pure geometric reconstruction of virtual cities to 'intelligent' data, models with semantics. Building Information Modeling (BIM) and Smart Cities currently are hot topics. These applications feed on a multitude of data sources. However, this reveals a discrepancy at the same time - semantic information as required for a multitude of applications like urban planning and infrastructure management, includes building use, number of dwelling units and more (Hecht, 2014). A key information, from which several other metrics can be derived or at least be approximated, is the aforementioned building use. Therefore, we see a need for large-scale automatic building category classification. The following paper proposes an approach to leverage Google's region wide available Street View data and link the inherent buildings with data from the digital city base map provided by the City Survey Office Stuttgart. To extract only building-relevant parts from the Street View data we pre-process the images. Therefore, we utilize metadata provided by the Street View API (Google Developers, 2017) and take advantage of a Deep Learning framework for semantic image segmentation (Long et al., 2015) to analyze our data for relevant content. Based on the information obtained in the crawling process we try to link image content with building polygons in the ground truth. The outcome is a tuple of building images and its corresponding building category. This data is then used to train a classifier. With the trained classifier it will be possible to predict building categories for new input images. First experiments are focused on investigating the potential of a Bag-of-Words (BoW) approach and a pre-trained CNN.

For now, we want to distinguish between four different building use types: *residential* (purely residential use), *commercial* (purely commercial use), *hybrid* (mixture of commercial and residential use) and *special use* (which can be a building use of anything else, for example: churches, hospitals, museums, but also construction sites). The remainder of this paper is structured as follows: in section 2 we give a brief review on urban classification using semantic segmentation and deep learning, section 3 describes our approach for the generation of training data to perform building use classification, section 4 shows some first results and in section 5 we discuss and draw some conclusions.

2. RELATED WORK

Within this section, several topics of related work are discussed. Section 2.1 briefly gives an overview of the subject of Urban Classification as a whole. In section 2.2 we more specifically address Semantic Segmentation for Urban Scenes. Finally, section 2.3 investigates recent related work in the field of Deep Learning.

2.1 Urban Classification

Urban classification can be hierarchically divided regarding the type of data acquisition the classification is based on. Satellite data provides information to perform classification with respect to different land use, based on hyperspectral analyses. (Hoberg et al., 2015) present a multitemporal and multiscale classification based on Conditional Random Fields (CRF). As well as there are several approaches to perform building outline detection from satellite imagery (Niemeyer et al., 2014). With aerial data acquisition, urban classification typically further diversifies – not only building outlines are extracted (Ortner et al., 2007), but typically the scenery is divided into vegetation, ground and buildings. Besides pure 2D image segmentation, state-of-the-art is to use 3D point cloud information obtained from dense image matching (Haala and Rothermel, 2015) or LiDAR (Guo et al., 2011). Data obtained by LiDAR systems can either stem from airborne laser scanning (ALS) or terrestrial – either static (TLS) or mobile (MLS). Particularly MLS data is in the focus of urban classification and will be discussed in the next section.

2.2 Semantic Segmentation for Urban Scenes

When dealing with terrestrial urban data a great number of tasks is tackled in literature. In (Weinmann et al., 2015) several

approaches (e.g. Nearest Neighbor, Decision Tree, SVM, Random Forest, Multilayer Perceptron) are investigated to classify MLS point clouds into semantic urban classes like façade, ground, cars, motorcycles, traffic signs and pedestrians. They report that Random Forests provide the best trade-off between accuracy and efficiency. Wang et al. (2015) presented an approach for holistic scene understanding, which reasons jointly about 3D object detection, pose estimation, semantic segmentation and depth reconstruction from a single geo-tagged image by using a holistic CRF. Similarly, (Xiao and Quan, 2009) use pairwise Markov Random Fields across multiple views to perform semantic segmentation for Street View images. We are aware of the large body of literature concerning building façade segmentation and interpretation. However, since we do not aim on extracting individual façade parts such as windows and doors in the presented work, but rather want to determine a specific building use category, we are not covering this topic here. An extensive overview on urban reconstruction, including façade interpretation can be found in (Musialski et al., 2013).

2.3 Deep Learning

Recent years have shown rapid development in CNN designs, performances and applications. Deep Learning is not only successfully applied in speech recognition (Hinton et al., 2012) and natural language processing (Collobert and Weston, 2008) tasks but also state-of-the-art for image classification and segmentation nowadays (Russakovsky et al., 2015, Everingham et al., 2012). Recent work proposed an approach to generate full sentences that describe image content (Karpathy and Fei-Fei, 2015). With regards to urban data, (Weyand et al., 2016) presented an approach that treats the photo geo-location problem as classification problem, in contrast to the more popular strategy of framing it as an image retrieval problem. They subdivide the earth into thousands of multiscale, geographical cells and train a deep network (PlaNet) using millions of geotagged images. For a query image, PlaNet outputs the probability distribution over the surface of the earth. The same task is addressed by (Hershey and Wulfe, 2016). They use a GoogLeNet model, pre-trained on a scene classification data set, to geo-locate images taken from GSV from 10 different cities. They report human exceeding accuracy of 75%. The work of (Movshovitz-Attias et al., 2015) uses SV images for the classification of storefronts, more specifically the classification into business categories. They create a large training data set by propagating business category information with help of an ontology that uses geographical concepts. For learning, they also use a network based on GoogLeNet. With a top1 accuracy of 69%, they are approximately at human level.

3. REGISTRATION OF IMAGE DATA WITH BUILDING USE CATEGORY

This part is structured as follows: in section 3.1 we describe the crawling process to extract georeferenced façade images from SV data. Selection and preprocessing of images to provide suitable image patches for classifier training is covered in section 3.2. Finally, in section 3.3 we elaborate on linking image patches to existing semantic information using coarse georeferencing information from Street View.

3.1 Urban Image Crawling

A crucial element in performing classification tasks is to obtain an appropriate number of training samples. Frequently, these are available from datasets and benchmarks within the fields of Computer Vision and Machine Learning. The SUN database (Xiao et al., 2010) consists of almost 4000 object categories but there are only slightly over 1000 images containing buildings. ImageNet (Deng et al., 2009) provides over 20,000 indexed synsets (synonymous word fields) and over 14 million images in total.

There are also several benchmarks for urban scenes – (Geiger et al., 2013) developed a mobile mapping platform and host KITTI, a benchmark with data for a variety of vision tasks from stereo matching, over scene flow to semantic segmentation. Likewise, the CITYSCAPES dataset provided by (Cordts et al., 2016) contains scenes from 50 cities with corresponding semantic pixelwise annotations for each frame, obtained by a windshield-mounted stereo camera system. For these datasets, GPS information of the car's trajectory is available. However, for our task these datasets are not suitable since we aim on assigning specific usage categories to buildings. We take another path and make use of municipal surveying data in combination with a publicly available image source. This way we can narrow down amd merge the variety of building categories, and enforce correctness of ground truth. There are several reasons why we pursue the proposed framework at all, when there are already huge CNNs that classify hundreds of categories with a reasonable level of correctness, including classes like *apartment building* or *office building*. First, those very deep CNNs developed by companies are fed with massive amounts of training data – not everybody can provide or produce those huge collections of training examples. Moreover, large CNNs have a broad range of category types they cover, while our work aims on a small subset of those classes. We are not interested in classifying a plethora of different categories, but rather very few, with potentially high intra-class variance. The evaluation of state-of-the-art approaches with a multitude of classes is frequently based on the top5 error, however, since we aim on the determination of a rather limited number of classes at a rather high reliability, the top1 error is our main interest.

The actual crawling is implemented in Java Script based on (Ashwell, 2015) modified for our use. As output from the crawling process, we obtain a list of positions P_i(longitude λ_i, latitude ϕ_i) and headings κ_i, where $i = 1, \ldots, N$, with N as the total number of crawl positions. By dragging the Google Maps marker one can define the initial crawling position. Using the Street View API the crawler searches for the next available panorama based on the current position. Figure 1 shows the crawling interface with the initial Street View on the left and all crawled panoramas on the right. We use two different modes of crawling: panorama-link based and random sampling. The first method successively visits the link nodes stored in the current panorama until a predefined total number of panoramas is

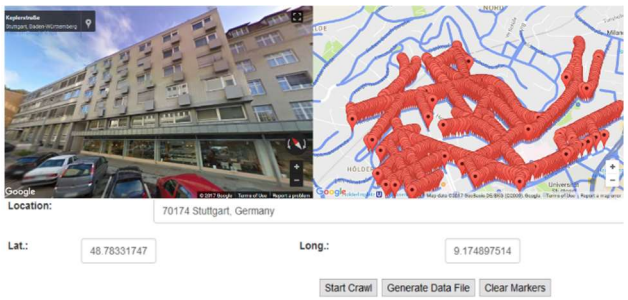

Figure 1. Left: Initial crawling position. Right: Markers depicting each crawled position after process has finished.

fetched. However, this method only returns the center heading κ_{c_i} of the street view car for this position. Therefore, when using

panorama-link based method we add 90° to κ_{c_i} – thereby we obtain frontal views of the buildings. When using the random sampling technique, we generate random offsets for latitude and longitude, thereby performing a random walk of the geographical position. To prevent from excessive divergence we reset to the initial position in predefined intervals. Based on the randomly sampled positions we then search for the nearest panorama and calculate the heading. Outcome of both crawling processes is a list of 2D geographic coordinates and a corresponding heading κ_i. We use this data together with the parameters pitch Φ and field of view (FOV) to query an image I_i as part of the panorama via the Street View API. Φ is measured positively looking upwards with respect to the camera's initial horizontal position. We chose $\Phi = 15°$ and $FOV_{\text{horizontal}} = FOV_{\text{vertical}} = 90°$ to ensure that also larger buildings are covered.

3.2 Extraction of building-relevant images

We aim on the extraction of good training data, which are images with clear view onto only one single building in center. However, many of the initial crawled images do not meet those requirements (see also section 3.2.1 and section 3.2.2).

Figure 2. Input image and corresponding output from the FCN evaluation. The semantic class *building* is depicted in blue, *sky* in red, *road* in yellow, *plant* in green and *car* in bright blue, respectively.

Thus, after fetching the Street View data we preprocess all images $I_{1\ldots N}$ to extract only samples with relevant content. One tool we use to analyze the images is a reimplementation of a Fully Convolutional Network (FCN) (Long et al., 2015) provided by (Caesar and Uijlings, 2016). This end-to-end/pixel-to-pixel trained network uses "raw" images as input and produces a semantic pixelwise labelling. We use the FCN-16s SIFT Flow model, which is based on the SIFT Flow dataset with roughly 3000 images and their corresponding pixel labels. In total, there are 33 semantic categories like *awning*, *balcony*, *bird*, over *mountain*, *person* to *tree* and *window*. However, there are not only semantic, but also geometric labels – the FCN can learn a joint representation and predict both. We are not interested in all of those classes. Effectively, we only want to detect whether or not a building is the actual main content of the current image. Hence, we merge several classes – for example, we merge *awning*, *balcony* and *window* to the *building* class. Similarly, we merge *grass* and *tree* to the *plant* class.

3.2.1 Occlusions: As stated in the previous section, we have to ensure, that the main image content is the building of interest. Thus, as a first step of processing the crawled urban imagery, we use the described FCN to perform a pixelwise segmentation. By using the merged classes introduced in the previous section we obtain results like depicted in Figure 2 on the right. If the main content of our segmented image consists of *plant* or *car* pixels, we discard this image.

Figure 3. Left: Building Polygons $BP_{1\ldots k}$ of Neighbourhood NH_i, based on crawling position P_i (depicted with a red cross, see also Figure 6); Right: Corresponding SV image I_i.

3.2.2 Blurred Images: Each building owner has the legal right to demand Google to make his or her private property unrecognizable within the Street View data. Google approaches this the same way they anonymize persons – by blurring the affected buildings. Obviously, we want to discard those images since there is no actual content provided. There has been a lot of work on edge-based blur detection (Ong et al., 2003; Narvekar et al., 2011). In fact, edge detection delivers quite consistent results in our case, as shown in Figure 4. However, as we incorporate the aforementioned FCN, we can make use of a particular property when evaluating images. In that framework, blurred regions are typically classified as *sky* or *sea* pixels and can thus be detected easily.

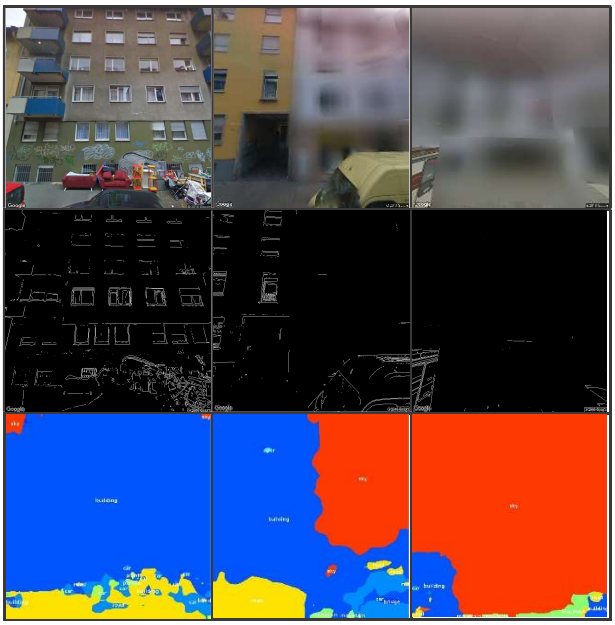

Figure 4. From left to right: SV images with ascending level of blurriness. From top to bottom: SV input data, edge images, output of the FCN evaluation. The colour coding in the last row is the same as in Figure 2.

3.3 Linkage of images with correct ground truth

Our ground truth data consists of a 2D shape file with ground plan polygons for each building, enriched with several aspects of semantic information like address, communal district, building block number and, especially of our interest, building use. For each building polygon BP_j we calculate its centroid c_j, where $j = 1, \ldots, M$, with M as the total number of buildings in the data set.

Figure 5. Columns: Our four classes (f.l.t.r): *commercial*, *hybrid*, *residential*, *special use*. The first two rows depict samples considered to be good, whereas the last row shows bad examples.

Once it is ensured, that there is actual building content contained in I_i, we have to link it to the correct corresponding ground truth. Here, we make use of the previously gathered data from the crawling process - we know the actual positions P_i, for each obtained SV image. However, these positions are in geographic coordinates. Ground truth data is located in the Gauß-Krüger coordinate system. Therefore, we perform a datum transformation between geographic coordinates and the reference coordinate systems from the national mapping agency. Subsequently, for each P_i we carry out a nearest neighbour (NN) search in the ground truth dataset based on the centroids c_j for each building polygon and extract k candidates $BP_{1\dots k}$ Those buildings depict our neighbourhood NH_i, in which we have to find the actual building displayed in the image, denoted as Γ_i. To obtain the correct Γ_i we have to address several issues, covered in the following now.

3.3.1 Interiors: In the crawling process, especially the random sampling approach is not limited to the required street level imagery but potentially also provides images from interior panoramas. To eliminate such data, typically covering shops, public institutions and suchlike, we take P_i and perform a point-in-polygon for each $BP_{1\dots k}$ in NH_i. If the test returns true for one of the polygons, I_i contains indoor scenery and is discarded. However, too limited geolocation accuracy of these interior panoramas might lead to an actual position outside the building. In future work we have to counteract this problem since the semantic segmentation FCN is trained for outdoor scenes and hence does not provide useful information in this case. Once interiors are handled we make use of the heading information κ_i to construct a line of sight ψ_i with the corresponding predefined $FOV_{h/v}$. We limit the length of ψ_i to 20 meters, to ensure Γ_i is the central content of I_i. In the next step, we determine whether ψ_i hits any of the polygons NH_i.

3.3.2 Multiple Hits and Viewing Angle Dependency: To verify whether or not there exists a suitable Γ_i, we use the line of sight ψ_i and perform a test for intersection with $BP_{1\dots k}$. If there are intersections, we call this a hit $H_{(1\dots h)}$. However, it is possible that we obtain multiple hits. The second hit is likely to be the intersection of the same BP on its rear or side part. For multiple buildings in close proximity, there can be more than two hits.
If this occurs, we simply sort $H_{1\dots h}$ by distance to P_i and take the candidate with the shortest Euclidean distance as our correct hit H_s. Multiple hits are more likely if the viewing angle onto Γ_i is very flat. Not only therefore we want to avoid flat viewing angles but mainly due to the reason, that we do not consider those samples as good training input. Ideally, we aim on quasi-frontal shots of the building façades. Thus, we proceed as follows. First, we determine our hit H_s and detect the edge where Γ_i is intersected. This edge is considered our façade plane. On the location H_s we construct the façade normal N_f and determine the angle α between N_f and ψ_i, representing our viewing angle (Figure 6). Ideally, α would be close to zero. The viewing angle depicted in Figure 3 is still in order, however if α exceeds a certain threshold we discard this image candidate. In the future, we plan on not only considering the central line of sight ψ_i but also the bounding rays for our $FOV_{h/v}$, in cases where the hit of ψ_i might not represent the actual central building content but rather a different building polygon within the bounds of the $FOV_{h/v}$. Figure 5 depicts crawled imagery for all four classes. The first two rows show examples we consider as good, whereas the last row demonstrates some negative examples.

4. STREET-VIEW BASED IMAGE CLASSIFICATION

At the moment, we limit our classification problem in terms of the number of classes. Thus, one might argue about the classifier of choice. From our point of view it is worthwhile not to restrict ourselves to handcrafted features like HOG, SIFT or SURF but also investigate in learned features from CNNs. Several works show, that on small-scale datasets with homogenous distribution, performance of handcrafted features can be considered on a par with learned ones. Whereas increased and more heterogeneous

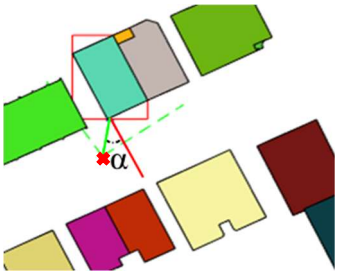

Figure 6. Viewing angle dependency. The red bounding box depicts the detected Γ_i. The straight line emerging from Γ_i is the façade normal, whereas ψ_i is depicted in green. α is the enclosed angle between those lines.

datasets lead to superiority of CNNs (Antipov et al., 2015; Fischer et al., 2014). Since we are crawling Street View images, we effectively have a vast amount of training data available – our limiting factor is the availability of correct ground truth for the building use.

4.1 Bag-of-Words Classification

For comparison, we applied an already existing implementation of a Bag-of-Words classifier, based on SURF features and a multiclass linear SVM. The underlying training and test database is described in section 4.2 in more detail. The original training set is randomly split in 80% actual training and 20% validation set. SURF features for each image are extracted and subsequently clustered using K-Means to create the visual vocabulary. Based on this vocabulary a multiclass linear SVM is trained on the training set and evaluated on the validation set. Average accuracy on the validation set is only 62%, same holds for average accuracy on the training set, which is at 63%. This classifier is now applied to a test set with available ground truth (the same as in section 4.2). The average accuracy here is at 41%. Obviously, those results are not really useful, thus an alternative approach is required.

4.2 Pre-trained Convolutional Neural Network

The data we use for training and testing the CNN is the same as in section 4.1, therefore we further elaborate on it here.
Our training set consists of 8000 images (4 classes, each 2000 images) and the validation set contains at least 70 images per class. However, the original training set is smaller – roughly 2200 images with a distribution of 19% *commercial*, 22% *hybrid*, 43% *residential* and 16% *special use*. Thus, we use data augmentation to provide an equal number of training samples for each class. Therefore we randomly pick images and randomly perform one of these three manipulations: 1.) flip image on its vertical axis, 2.) crop and resize to original dimension, 3.) define random 2D affine transformation (in certain range), warp the image and resize to original dimension.
For our first proof of concept we use transfer learning on the imagenet-vgg-f model from (Chatfield et al., 2014). For further information about the architecture, we would like to refer to the reference. To adapt this network to our needs we remove the last two layers (the fully connected fc8 layer and the softmax layer) and add a custom fc8 layer, which only has an output data depth of 4 as opposed to the original output depth of 1000. As final layer we add cross-entropy because we want to determine loss. Additionally, we add two dropout layers between fc6 and fc7, as well as between fc7 and fc8, with a dropout rate of 0.5 each – since they were probably removed in the testing phase of the original network. During training phase, we use jittering to reduce overfitting. Within each training batch we randomly flip and crop images. On top of that, we apply an alternation of the RGB channel intensities using PCA, as reported in (Krizhevsky et al., 2014). We use a batch size of 40 images and a fixed learning rate $\eta = 0.0001$. After 96 epochs, the top1 training error

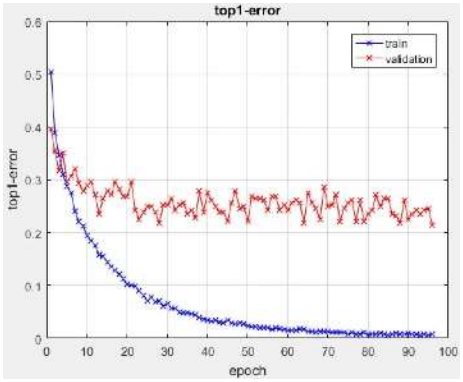

Figure 7. Top1 error after 96 epochs. Training is depicted in blue, validation in red.

Figure 8. Predictions of the approach described in section 4.2, depicted in the shape of a classification matrix. The main diagonal entries are correct predictions. Please note how some of the actual ground truth labels themselves are sometimes ambiguous or the correct class is even for humans hard to identify. Example 1: row 2, column 4 was classified as *hybrid* but has the ground truth label *special use* – actually this is a care facility and we class the entirety of care facilities as *special use*. Example 2: row 4, column 3 is clearly a building under construction, though the *residential* label is obviously correct – but we trained the network on several construction sites with the label *special use*, therefore the respective prediction. (Note: *special use* class is labelled with *unknown* in the images here.)

is at 0.725% and the top1 validation error is at 21.4% (Figure 7). We run this on a test set (the same as for the BoW classifier), which however also contains images from the evaluation set. Here, we obtain an average accuracy of 75.9%. In Table 1, the results for precision and recall are depicted. With 85%, the precision for *residential* is best, whereas the *special use* category is with 63.3% at the lower end. This is most likely due to the high intra-class variance of the *special use* category, whereas the *residential* class is more homogenous in terms of visual similarity. In Figure 8, some examples of the classification are provided. We depict correct and wrong examples in terms of a confusion matrix. Columns represent ground truth, rows are predictions from the CNN, correspondingly. Correctly classified images are therefore displayed on the main diagonal, all remaining images are wrong classifications.

	Recall	Precision
Commercial	0.7162	0.7260
Hybrid	0.7680	0.8067
Residential	0.7589	0.8500
Special Use	0.7848	0.6327

Table 1. Precision and recall after evaluation on our test set (a value of 1.0 equals 100%).

Transfer to unknown data representation type: For comparison purposes we additionally applied our trained net to data we used in a previous test, where humans should classify input images into respective building categories (Tutzauer et al., 2016). This database additionally provided two alternative representations for building objects – firstly screenshots of textured meshes from Google Earth and secondly screenshots of manually modelled untextured LOD3 building models. We picked the untextured LOD3 models for input to the CNN, since they only have an abstract resemblance with the original training data. In total we evaluated almost 80 images and achieve an average accuracy of 63.6%. There are two important issues: a) the CNN has not seen this representation type at all during training phase and b) the LOD3 models additionally contain several samples with class-specific geometric properties on which the network was not trained. However, this shows the transferability of the network to even a completely different representation type in the input data. Some examples are depicted in Figure 9.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we successfully linked Google Street View imagery to a database that contains semantic information for every contained building polygon. Such databases are available for a number of cities. Hence, it is potentially possible to generate large amounts of training data, which is a prerequisite for the successful application of Deep Learning frameworks for classification. In a first test, it was verified, that this approach can be promising, however future work will aim exactly on that very topic. In order to do so, some additional work has to be done in the processing step. Indoor scenes with limited geo-location accuracy have to be detected and eliminated. The incorporation of the bounding $FOV_{h/v}$ rays might help in cases where the hit of ψ_i is not representing the actual central building content. Moreover, the FCN used for image analyses could be replaced by an object detector framework like Faster R-CNN (Ren et al., 2015), since we are ultimately only interested in the bounding boxes of buildings. However, pre-trained models do not contain a building class yet. Therefore, such a network has to be trained from scratch. In our investigations, we found that semantic data

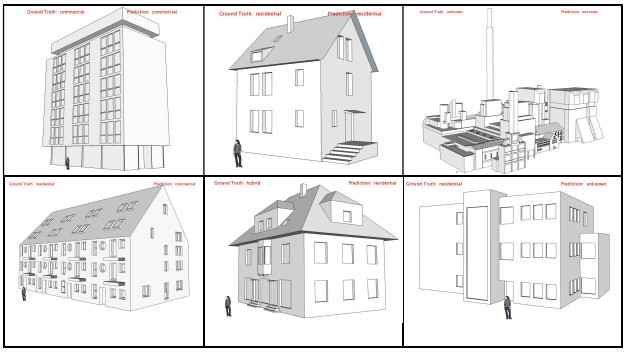

Figure 9. Results from prediction of pre-trained CNN. The first row shows some correct predictions, f.l.t.r: *commercial*, *residential*, *special use*. The second row depicts wrong classifications, f.l.t.r. denoted in *ground truth* vs *predicted*: *residential* vs *commercial*, *hybrid* vs *residential*, *residential* vs *special use* (note: *special use* is labelled with *unknown* here).

provided by the city administration can be ambiguous or even erroneous. This is an issue, which at the same time shows the necessity of the proposed approach of automatic building use classification. For now, obviously wrong or ambiguous samples were discarded in an interactive post-processing step to provide a reasonable training input. In the future, we aim on training a variety of architectural styles as well as performing the training phase in one city and testing in a different one to investigate transferability. For that purpose we want to train our own CNN architecture from scratch. Since we ultimately want to further diversify from the current four classes, it is conceivable to leverage the original building-related segmentation classes from the FCN (*awning*, *balcony*, *door*, *window*) as a meta-classifier. As an application for our approach, we think of area-wide enrichment of crowd-source data like OSM building polygons.

ACKNOWLEDGEMENTS

We would like to thank the German Research Foundation (DFG) for financial support within the project D01 of SFB/Transregio 161.

REFERENCES

Antipov, G., Berrani, S.-A., Ruchaud, N., Dugelay, J.-L., 2015. Learned vs. Hand-Crafted Features for Pedestrian Gender Recognition, In: *Proceedings of the 23rd ACM International Conference on Multimedia*, MM '15. ACM, New York, NY, USA, pp. 1263–1266. doi:10.1145/2733373.2806332

Ashwell, P. 2015. Street View Crawler, https://github.com/peterashwell/streetview-crawler, 2015.

Caesar, H. and Uijlings, J., 2016. Matconvnet-Calvin https://github.com/nightrome/matconvnet-calvin

Chatfield, K., Simonyan, K., Vedaldi, A., Zisserman, A., 2014. Return of the Devil in the Details: Delving Deep into Convolutional Nets. arXiv:1405.3531 [cs].

Collobert, R., Weston, J., 2008. A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning, In: *Proceedings of the 25th International Conference on Machine Learning*, ICML '08. ACM, New York, NY, USA, pp. 160–167. doi:10.1145/1390156.1390177

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B., 2016. The Cityscapes Dataset for Semantic Urban Scene Understanding, In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* pp. 3213-3223. arXiv:1604.01685 [cs.CV]

Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L., 2009. ImageNet: A large-scale hierarchical image database, In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255. doi:10.1109/CVPR.2009.5206848

Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J. & Zisserman, A. 2012. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results, http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html

Fischer, P., Dosovitskiy, A., Brox, T., 2014. Descriptor Matching with Convolutional Neural Networks: a Comparison to SIFT. arXiv:1405.5769 [cs].

Geiger, A., Lenz, P., Stiller, C. and Urtasun, R., 2013. Vision meets robotics: The KITTI dataset. *The International Journal of Robotics Research*, 32(11), pp.1231-1237. doi:10.1177/0278364913491297

Google Developers. Street View Service | Google Maps JavaScript API | Google Developers, 2017. https://developers.google.com/maps/documentation/javascript/streetview.

Guo, L., Chehata, N., Mallet, C. and Boukir, S., 2011. Relevance of airborne lidar and multispectral image data for urban scene classification using Random Forests. *ISPRS Journal of Photogrammetry and Remote Sensing*, 66(1), pp.56-66. doi:10.1016/j.isprsjprs.2010.08.007

Haala, N. and Rothermel, M., 2015. Image-based 3D Data Capture in Urban Scenarios. In: *Proc. Photogrammetric Week 2015.* Wichmann, Berlin/Offenbach, pp. 119-130.

Hecht, Robert, 2014. Automatische Klassifizierung von Gebäudegrundrissen. Dissertation, Leibniz-Institut für ökologische Raumentwicklung. Rhombos-Verlag.

Hershey, D. and Wulfe, B., 2016 Recognizing Cities from Street View Images.

Hinton, G., Deng, L., Yu, D., Dahl, G.E., Mohamed, A.R., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T.N. and Kingsbury, B., 2012. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6), pp.82-97. doi:10.1109/MSP.2012.2205597

Hoberg, T., Rottensteiner, F., Feitosa, R.Q. and Heipke, C., 2015. Conditional random fields for multitemporal and multiscale classification of optical satellite imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 53(2), pp.659-673. doi:10.1109/TGRS.2014.2326886

Karpathy, A. and Fei-Fei, L., 2015. Deep visual-semantic alignments for generating image descriptions. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3128–3137.

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. In: *Advances in Neural Information Processing Systems*, pp. 1097–1105.

Long, J., Shelhamer, E. and Darrell, T., 2015. Fully convolutional networks for semantic segmentation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3431-3440. arXiv:1411.4038 [cs.CV]

Movshovitz-Attias, Y., Yu, Q., Stumpe, M.C., Shet, V., Arnoud, S. and Yatziv, L., 2015. Ontological supervision for fine grained classification of street view storefronts. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1693-1702.

Musialski, P., Wonka, P., Aliaga, D.G., Wimmer, M., van Gool, L. and Purgathofer, W., 2013. A Survey of Urban Reconstruction. *Computer Graphics Forum* 32, pp. 146–177. doi:10.1111/cgf.12077

Narvekar, N.D. and Karam, L.J., 2011. A no-reference image blur metric based on the cumulative probability of blur detection (CPBD). *IEEE Transactions on Image Processing*, *20*(9), pp.2678-2683. doi:10.1109/TIP.2011.2131660

Niemeyer, J., Rottensteiner, F. and Soergel, U., 2014. Contextual classification of lidar data and building object detection in urban areas. *ISPRS journal of photogrammetry and remote sensing*, *87*, pp.152-165. doi:10.1016/j.isprsjprs.2013.11.001

Ong, E., Lin, W., Lu, Z., Yang, X., Yao, S., Pan, F., Jiang, L. and Moschetti, F., 2003. A no-reference quality metric for measuring image blur. In: *Signal Processing and Its Applications, 2003. Proceedings. Seventh International Symposium on,* pp. 469-472. doi:10.1109/ISSPA.2003.1224741

Ortner, M., Descombes, X. and Zerubia, J., 2007. Building outline extraction from digital elevation models using marked point processes. *International Journal of Computer Vision*, *72*(2), pp.107-132. doi:10.1007/s11263-005-5033-7

Ren, S., He, K., Girshick, R. and Sun, J., 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In: *Advances in neural information processing systems*, pp. 91-99. arXiv:1506.01497 [cs].

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M. and Berg, A.C., 2015. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3), pp.211-252. arXiv:1409.0575 [cs].

Tutzauer, P., Becker, S., Fritsch, D., Niese, T., and Deussen, O., 2016. A Study of the Human Comprehension of Building Categories Based on Different 3D Building Representations. *Photogrammetrie-Fernerkundung-Geoinformation*, 2016(5-6), pp. 319-333. doi:10.1127/pfg/2016/0302

Wang, S., Fidler, S. and Urtasun, R., 2015. Holistic 3d scene understanding from a single geo-tagged image. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3964-3972. doi:10.1109/CVPR.2015.7299022

Weinmann, M., Jutzi, B., Hinz, S. and Mallet, C., 2015. Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers. *ISPRS Journal of Photogrammetry and Remote Sensing,* 105, pp.286-304. doi:10.1016/j.isprsjprs.2015.01.016

Weyand, T., Kostrikov, I. and Philbin, J., 2016. PlaNet - Photo Geolocation with Convolutional Neural Networks. In: *European Conference on Computer Vision,* pp. 37-55. Springer International Publishing. doi:10.1007/978-3-319-46484-8_3

Xiao, J. and Quan, L., 2009. Multiple view semantic segmentation for street view images. In *Computer Vision, 2009 IEEE 12th International Conference on,* pp. 686-693. doi:10.1109/ICCV.2009.5459249

Xiao, J., Hays, J., Ehinger, K.A., Oliva, A. and Torralba, A., 2010. Sun database: Large-scale scene recognition from abbey to zoo. In: *Computer vision and pattern recognition (CVPR), 2010 IEEE conference on,* pp. 3485-3492. doi:10.1109/CVPR.2010.5539970