Holistic Multi-View Building Analysis in the Wild with Projection Pooling

Zbigniew Wojna¹, Krzysztof Maziarz²*, Łukasz Jocz¹, Robert Paluba¹, Robert Kozikowski¹, and Iasonas Kokkinos³

¹ Tensorflight
{zbigniew,lukasz,robert,kozikow}@tensorflight.com
² Jagiellonian University
krzysztof.maziarz@microsoft.com
³ University College London
i.kokkinos@ucl.ac.uk

Abstract. We address six different classification tasks related to fine-grained building attributes: construction type, number of floors, pitch, and geometry of the roof, facade material, and occupancy class. Tackling such a problem of remote building analysis became possible only recently due to growing large-scale datasets of urban scenes. To this end, we introduce a new benchmarking dataset, consisting of 49426 top-view and street-view images of 9674 buildings. These photos are further assembled, together with the geometric metadata. The dataset showcases a variety of real-world challenges, such as occlusions, blur, partially visible objects, and a broad spectrum of buildings. We propose a new projection pooling layer, creating a unified, top-view representation of the top-view and the side views in a high-dimensional space. It allows us to utilize the building and imagery metadata seamlessly. Introducing this layer improves classification accuracy – compared to highly tuned baseline models – indicating its suitability for building analysis.

1 Introduction

This work aims to develop a deeper understanding of scenes containing buildings, based on both aerial (top-view) and multiple side view (street-view) images. This problem is both technically challenging and of great practical importance. It allows for automatic pricing of an insurance policy, mechanical claims analysis, risk detection, understanding of the environment for self-driving cars, or extraction of socioeconomic statistics. The onset of new computer vision techniques and growing large-scale datasets of urban scenes naturally lead to a variety of new scientific challenges related to holistic building understanding, much deeper than simple single-task classification. We investigate a multi-task classification problem of automated building analysis. The goal is to determine the following set of building attributes: construction type, i.e., the material the building is

* Work done during an internship at Tensorflight. The author is now at Microsoft Research.
Fig. 1. An example of a challenging building scene in our dataset. We aim to answer multiple questions about the building, and the answer to each lies in a different input image. The green mark represents the source location of the bottom right image number 4, while the green rays represent its field of view. The building of interest is marked with a green pointer on the top-view image, and it is the white one on the side view. The shading on the top-view image reveals that the roof is not flat, and the geometry is gable. The front wall occludes most of the roof, but we can still infer its low pitch. The first image is not precisely calibrated, and could suggest wrong answers, but combining information from all the sources should prevent it.

made of, number of floors, roof pitch, roof geometry, facade material, and occupancy type. Those features are crucial for catastrophic risk estimation [44]. The real-world unconstrained environment is often poorly recognizable based on a single image. Rather, it requires more complex analyses of multiple images taken from different angles. In our model, we use both the top-view image and numerous street-view photographs of the same building to understand its fine-grained details.

In this work, we propose a new fusion technique, leveraging both the geometric structure of the scene and the high-dimensional features extracted from the top and street-view images. As a result, we obtain a single unified top-view representation of the scene, including information from side views, building outline, and imagery metadata. Based on the street-view photo location, its direction, and the field of view, i.e., angle representing the visible range, we construct a projection of the street-view features onto the building walls outlined in the top-view image.

Understanding of physical structures, such as buildings, may require integrating information from all possible input sources. An example of a building scene from our dataset is presented in Figure 1. The four camera marks on the top-view image represent four different locations where the street-view images were taken. The green mark and the blue boundary on the top-view image point to the building of interest. The violet boundaries represent other visible buildings, which can be a source of confusion for the model. We can easily see how different images complement each other. The top-view image allows us to see that the roof is not flat as there is a ridge joining the two opposite sides and
Holistic Multi-View Building Analysis in the Wild with Projection Pooling

a skewed shadow. The roof geometry is only visible from the top-view image. At the same time, the construction type, number of floors, facade material, and occupancy class can be determined only based on the street-view images.

Some decisions may require more sophisticated reasoning. A possible overlap of the features from different street-view photos might mitigate the errors caused by occlusions and inaccuracies present in a single image. In the example in Figure 1, the first street-view image points towards the center of the building, but is occluded by an adjacent building and could lead to classification errors. However, when looking at all four images, it is possible to correctly classify the building’s attributes by combining several clues. The side-attached structure made of bricks may correctly suggest that the construction type of all buildings along the street is masonry. We are not able to estimate the exact roof pitch by looking exclusively at the street-view images. We can only assume that the roof is flat or has a low slope as it does not stick out from the front.

To summarize, our contributions are the following:

(1) We propose a new layer called projection pooling, which takes into account the building scene geometry and the relationship between images from different views and perspectives. We then integrate it into a deep learning architecture. It results in a new, unified, high-dimensional representation and suits well the classification task.

(2) We develop a new deep convolutional model, fusing multiple inputs for a better understanding of building characteristics. It includes the projection pooling layer and achieves results that are superior to highly tuned baseline models. These results indicate that one can design substantially more accurate models by incorporating information from multiple images.

(3) We build a new real-world multi-view multi-task dataset of building images, annotations, and attributes. Buildings are heterogeneous in architectural style, size, age, and come from all around the world. It sets a new benchmark of detailed building understanding to spark further research in this area. We will release the dataset upon publication.

2 Related work

In this section, we discuss the ideas which inspired this work. These include (a) the use of street-view images, (b) building modeling, (c) the fusion of street-view and top-view imagery, and (d) building attribute classification.

(a) As street-view imagery is becoming more ubiquitous, it brings new research opportunities. One can assess socioeconomic statistics [15,16], evaluate the safety, beauty and popularity of the neighborhood [1,12,5], estimate the road safety [40], perform road scene segmentation [10], and determine precise geolocalization of the car [2,19,34,3]. Combining house rental ads and street-view imagery allows for performing 3D building reconstruction [8]. In our work, we show that street-view images can also help with understanding the state of urban structures.
(b) Building facade segmentation is a well-studied problem on 2D images [51,46,37,45,30,25,24,31,28], through combination of 2D and 3D [14,36] and directly using 3D data from scanners [39,26]. Many methods rely on assuming symmetry of the building facades [9,32,33,50,54]. Facade datasets are usually collected at a specific location, consist only of a few hundred images, are homogeneous in style and well rectified. Therefore, models can fail when tested on different architectural styles [29]. Our work introduces a diverse and large-scale dataset, with buildings coming from all over the world, along with potential noise, and without any assumption about rectification, symmetry or style.

(c) A fusion of street-view and top-view imagery may further improve performance. One can fuse top-view and ground-level imagery for detailed city reconstruction [4], use cross-view matching with top-view images to improve street-view geolocalization [48,21], or use cross-view matching with street-view to retrieve latent representation of top-view images [49,6]. In [53], the authors transform a semantic top-view scene into a semantic street-view scene. Detecting trees can be achieved through merging both of these sources [47]. Other works fuse LIDAR sources to perform 3D object detection [27,7]. Using the geometry by projecting the street-view latent representation onto an orthographic projection allows for 3D object detection [38]. Our solution also takes advantage of both input modalities but focuses on detailed building understanding. As opposed to the previous works, our fusion strategy directly utilizes the building geometry to create a 2D latent representation through the projection pooling layer.

(d) Closely related to our study is the classification of building age, condition, and land use from a single street-view image [52,23,22,55]. Urban zone classification was also explored in multi-view settings [42,20,43,13]. We are the first one to study buildings from all around the world, where multiple recognition tasks are trained together. Most importantly, we propose a new fusion strategy based on building geometry, which boosts the classification accuracy of our models.

3 Dataset

One of the goals of this study is to create the first large-scale benchmark of buildings which represents a diverse spectrum of architectural styles, locations, and building attributes. Building sizes vary: from small wooden houses, through churches made of brick, to extensive concrete manufacturing facilities. Thus, models trained on such a dataset should be applicable to any building in the world. We present the world heatmap of building sites in the Appendix. Our dataset of top-view images was collected from Google Maps and Bing Maps, while the street-view photos come from Google Street View and Bing Street-Side. The typical top-view image resolution is 30 cm per pixel; see the Appendix for more information on the resolution of aerial images. We built a custom tool to annotate the georeferenced top-view images with the following set of classes: buildings, temporary structures, trees. We selected the street-view photos based on tree locations and other buildings visible on the top-view image to account
for potential occlusions. The images were selected so that they have the best building visibility and cover all sides of the building. For every building, we have one top-view image and between 1 and 9 street-view photos. Knowing the locations of the street-view images, we compute the field of view required for every image to capture the entire building. Since these locations can be imprecise, we set the actual field of view to be 20% broader to make sure the whole building is present in the photo. Top and street-view images usually come from different dates, but our annotators verified that both images refer to the same structure. For every building of interest, we annotated its construction type, number of floors, roof pitch, roof geometry, facade material, and occupancy type. Hired contractors double-checked all the annotations and building characteristics. Additionally, construction types were checked by a professional architect.

We classified the building attributes as follows [44]:

- the construction type describes the predominant material used for the building construction, and it consists of four classes: masonry, metal, reinforced concrete, and wood;
- the number of floors includes five categories: one, two, three, four and five or more;
- the roof pitch was divided into four categories: flat (0° - 9.5°), low (9.5° - 22.5°), medium (22.5° - 37°) and steep (> 37°);
- the roof geometry classes are: flat, gable, hip and shed;
- facade material was grouped into brick, cement block, concrete, glass, metal, plaster, plastic, stone, and wood;
- the occupancy classes are the following: agriculture, commercial, industrial, mercantile, public, and residential.

We present an example of each instance for each attribute in the Appendix.

Our dataset consists of 6477 training and 3197 testing building scenes, split by stratified sampling. The total number of street-view images is 29350 in the training dataset, and 10402 in the testing dataset. Most of the street-view photos are of size 640 × 640, but around 25% of them are higher. This is because we extended some of the images upwards – by stitching together multiple photos – to obtain a better view of taller structures. The size of the top-view image (one per building) depends on the building size.

The dataset includes the following:

- top-view images
- street-view images
- the location of the street-view images
- direction of the street-view images
- field of view for the street-view images
- the georeferenced footprint of the building.

In Figure 1, we present a full example, showing all images and metadata available for a single building, along with the correct output classes.
4 Multi-view Multi-task approach

We start by introducing the basic definitions and then present increasingly strong models for building attribute classification.

We denote the top-view image by $I_m$, and the $n$-th street-view input image by $I_m$, for $n \in \{1, \ldots, N\}$, where $N$ is the total number of street-view images for the given building. We transform the input images to obtain spatial feature maps $f_0, f_1, \ldots, f_N$ using feature extractor networks $\text{CNN}_{TV}$ (top-view) and $\text{CNN}_{SV}$ (street-view), with $d$ output channels for street-view features and $D$ output channels for top-view features:

$$f_0 = \text{CNN}_{TV}(I_m) \quad f_n = \text{CNN}_{SV}(I_m_n).$$

Feature vectors obtained from these feature maps by mean-pooling along spatial dimensions are denoted as $v_0, \ldots, v_N$. We use six linear heads with softmax activation to generate the class probabilities for each of the tasks. We train the models to minimize the total cross-entropy loss summed over all of the classification tasks. We consider the following baselines.

**Top-view (TV)** When only the top-view input image is used, the input to the six task-specific classification layers is $v_0$.

**Street-view (SV)** To fuse the street-view input images, we compute feature vectors $v_1, \ldots, v_N$, and use their average as input to the task-specific classification layers.

**Street-view + Top-view separately (SV+TV separately)** The different tasks achieve varying results depending on which kinds of images are given as input: street-view or top-view. Based on the results of the baselines that use a single input type (either street-view or top-view), we choose the best baseline model for each task separately. We train the top-view model to classify the roof slope and roof geometry, while the street-view model to classify other attributes. The two networks are completely separate and are trained separately.

**Street-view + Top-view (SV+TV)** We use two independent feature extractors: one for street-view and one for top-view images. The vectors $v_1, \ldots, v_N$ are averaged as in the street-view baseline, and then this average is concatenated with $v_0$ to form the input to the task-specific classification layers. This fusion is shown in Figure 2. We also consider a modification where the same feature extractor network is used both for the top-view and street-view images (i.e. $\text{CNN}_{TV} = \text{CNN}_{SV}$).

5 Our model

The previous models take into account neither the relatively simple cuboid-like geometry of a typical building nor the source of the street-view image. These factors, especially the distance between the building and the source of the street-view photo, may impact what is visible on the input image. We design a fusion strategy to leverage the building geometry, street-view photo location, and its direction. It results in a single, unified, top-view representation of the building.
We average the street-view features and then concatenate them with top-view features for the final class prediction.

We extract the feature maps from all street-view photos as defined in Equation (1), and map (called projection in this work) these features onto relevant parts of the building polygon, as seen on the top-view image. Those projected features are overlapped and concatenated with the top-view aerial representation, to let the model infer about the whole visible scene.

The final representation encompasses the following information: all street-view images, top-view image, building footprint, street-view image positions, directions, and fields of view.
5.1 Projection Pooling

We describe the construction of the unified building representation following the explanatory Figure 3. For each street-view image $I_m$ of size $[H_i, W_i, 3]$, we extract the feature map $f_i$ of shape $[h_i, w_i, d]$, and average out the vertical dimension, while keeping the horizontal dimension intact. It results in a feature stripe $s_i$ of shape $[w_i, d]$, which will be projected onto the building outline. Since all the street-view images in our dataset are of the same width, we can replace the $w_i$’s by one value $w$, so all the stripes $s_i$ are in fact of shape $[w, d]$. We initialize a zero tensor $T$ of shape $[h_0, w_0, d]$, where $h_0$ and $w_0$ are the height and width of the feature map $f_0$. $T$ represents top-view 2D-grid of $d$-dimensional neurons. Neurons in $T$ are projected from each stripe $s_i$. By concatenating $T$ of shape $[h_0, w_0, d]$ with the top-view feature map $f_0$ of shape $[h_0, w_0, D]$, we construct the final unified representation.

In this paragraph, we explain the details of how we project a stripe $s_i$ onto the tensor $T$. Using the building outline, source location, and direction of street-view image $I_m$, we compute the visible parts of the building polygon (represented by colored segments on tensor $T$ in Figure 3). An efficient way of computing them is described in the Appendix. These segments correspond to parts of the edges of the building polygon in $I_m$. They can be discretized and approximated by a set $P_i$ of $d$-dimensional neurons located in $[h_0, w_0]$ 2D-grid of $T$. For each neuron $p = (x_p, y_p) \in P_i$, we compute the angle at which it can be seen from the location of $I_m$. This forms its cone of visibility, as presented in Figure 4. This cone lets us compute the visible part of the building corresponding to $p$ in the image $I_m$. This translates to an angle range $(a_l, a_r)$ along the width dimension of $s_i$. We project the features from this range in $s_i$ onto the $(x_p, y_p)$ position in $T$. In Subsection 5.2, we explore various ways of sampling the features from a single stripe $s_i$. Some of the neurons in $T$ might not be visible on any of the street-view images, or do not belong to the building polygon - the corresponding feature vectors in $T$ will be zero vectors. If a single neuron in $T$ gets features projected from multiple stripes, these feature vectors are max-pooled.

This description can be formulated as follows:

\[
s_i[ww, dd] = \sum_{hh \in [0...h_i-1]} f_i[hh, ww, dd] \quad (2)
\]

\[
T[hh, ww] = \max_{i \in [1...N]} \text{ProjectionPooling}(hh, ww, prt, s_i, src_i, dir_i, FoV_i) \quad (3)
\]

Where $hh, ww, dd$ are indexes of height, width and depth dimensions, $prt$ is the building footprint, $N$ is number of street views images, $src_i$ is source location of i-th street view image, $dir_i$ is direction of i-th street view image and $FoV_i$ is its field of view.

5.2 Stripe sampling (SS)

A single neuron in the top-view feature map can correspond to a relatively large area in the original input image. The angle of visibility for a neuron on a street-view image can cover multiple pixels in $s_i$, especially for the close-by photos (see
Algorithm 1 ProjectionPooling(hh, ww, prt, si, srci, diri, FoVi)

if (hh, ww) not in ConeOfVisibility(srci, diri, FoVi) then
    return 0
end if
if (hh, ww) not in Boundary(prt) then
    return 0
end if
if (hh, ww) in OccludedByOtherWallSide(prt, srci, diri, FoVi) then
    return 0
end if
ai, ac, ar = MapPixelToStripeWidthRange(hh, ww, prt, srci, diri, FoVi)
return StripeSample(ai, ac, ar, wi, si, FoVi)

the red area in Figure 4). For example, a $1 \times 1$ pixel in the output feature map of ResNet-50 [18] with a total stride of 32 corresponds to an area of $32 \times 32$ pixels in the input image. The details on how to calculate the neuron value can be crucial for the overall performance of the model [17]. We define three different strategies for deriving the neuron value from the $s_i$ stripes. While we describe these strategies below, we also presented them visually in Figure 4.

Nearest (N) uses the angle, measured clockwise, between the beginning of the field of view of given street-view image and the center of the neuron. We
denote this angle by $a_c$. FoV represents the field of view angle for the entire street-view image, and $w$ is the width of the associated stripe $s$ (as defined in Subsection 5.1). In this strategy, we use a single feature vector from $s$ to calculate value of neuron $p$:

$$p_{\text{nearest}} = s \left( \frac{w}{FoV} a_c \right).$$

In other words, we draw a ray from the location of the street-view photo towards the neuron’s center, calculate the position $p \in \{0, \ldots, w - 1\}$ of its intersection with the stripe $s$, and take the $d$-dimensional feature vector at this position (recall that $s$ is of shape $[w, d]$).

**Sum (S)** uses two rays pointing towards the pixel’s ends instead of one ray pointing towards the pixel’s center. Denote the angles of these rays by $a_l$ and $a_r$ for the left and right one, respectively. Then, the value of the projected feature vector is defined below, where $\lfloor \cdot \rfloor$ represents the floor operation, $\lceil \cdot \rceil$ the ceil operation and $\{\cdot\}$ takes the fractional part of a number.

$$p_{\text{sum}} = s \left( \frac{w}{FoV} a_l \right) \cdot \{w \frac{FoV - a_l}{FoV}\} + s \left( \frac{w}{FoV} a_r \right) \cdot \{w \frac{a_r}{FoV}\} + \sum_{i=\lfloor w \frac{a_l}{FoV} \rfloor}^{\lfloor w \frac{a_r}{FoV} \rfloor - 1} s[i]$$

**Average (A)** follows a similar strategy to Sum – i.e. it takes into account feature vectors from multiple positions along the feature stripe $s$. However, a potential problem with the Sum strategy is that the magnitude of values in the resulting vector may vary significantly with the distance to the building wall: more feature vectors are summed if the building wall is in proximity. Therefore, we propose a variant of the Sum strategy, where the output is additionally normalized. We choose to simply divide the resulting feature vector by the width of the part of $s$ corresponding to the neuron $p$:

$$p_{\text{average}} = \frac{p_{\text{sum}}}{w \frac{a_r - a_l}{FoV}}$$

### 5.3 Visualizing projected features

We visualize the features extracted from the street-view photos and how they are projected to create the final unified building representation. We obtain features from the test set pictures corresponding to multiple buildings and project them to three dimensions using principal component analysis, resulting in RGB coordinates. It allows us to visualize a single feature vector as a color.

We visualize the street-view feature maps both before, and after their height dimension is averaged out. Then, we look at the feature vectors that were projected onto the building polygon from each street-view photo separately. We show this in Figure 5. Note that for the selected building, the south wall looks substantially different from the others, and this difference can be seen in the visualizations. Note that the feature maps in positions where no building is visible get projected to the same uniform color (pink), showing that the feature extractor ignores irrelevant information. Finally, we examine the vectors of the
Fig. 5. Left: We show street-view photos corresponding to a single building (top row), visualization of the extracted feature maps $f_i$ before averaging out the height dimension (second row), and after (third row). In the last row, we show features contributed to specific neurons of the unified representation. Right: Visualization of the unified representation pooled from all the images. The neurons shown as white did not get any features projected onto them, i.e., their values are 0.

unified representation, i.e., after max-pooling the features projected from different views. We see that, again, the side of the building providing different visual information has a substantially different projection.

6 Experiments and results

6.1 Experimental setup

We benchmark our approach on the newly collected dataset of building attributes, as defined in Section 3. Street-view images are resized to size 500 × 500. The top-view image is cropped to the bounding box containing the building of interest and resized to keep the aspect ratio so that the length of the longer side is 500. As the feature extractor network, we use the ResNet-50 model pre-trained on the ImageNet dataset, as available in PyTorch [35]. We discard the final fully connected output layer. We freeze the parameters of the stem and the first block in the pre-trained ResNet-50 feature extractors and do not fine-tune the batch normalization layers. We train the network using stochastic gradient descent with momentum, on a single GPU, and with a batch size of one building. The effective number of images in a single batch is equal to 1 for the top-view branch, while for the street-view branch, it varies from 1 to 9. We use the learning rate of 0.0001 for ten epochs and then 0.00001 for one more epoch. We set the momentum to 0.9. We apply L2 regularization with a weight decay of 0.001 and augment the dataset with random color jittering. We compare the models using average classification accuracy over the six tasks.

Image dropout (ID) Inspired by dropout [41], we regularize the training by randomly dropping entire street-view images, which we call image dropout. During training, every street-view image in the batch of the given building is omitted with probability $p$. We make sure that there is always at least one street-view image in a batch. We do not rescale the values of the features as with the
original dropout. With image dropout, we force the model to use a different set of images in each iteration. We do not apply image dropout at test time, allowing the model to use all of the street-view images available for the given building.

**Projection thickness (TH)** When rounding up the top-view building polygon to a set of neurons in $T$, we obtain a thin building outline, where each side is discretized as a one-pixel wide line. We consider using an outline wider than one. Projection thickness of $k$ means that the building outline is discretized to line segments, which are $k$ neurons wide. We assume that pixels belonging to the same side do not occlude each other. Widening the polygon allows the convolutional operation to rely on multiple pixels when multiplying the input tensor by the kernel.

**Cutout (CU)** To further regularize the training over multiple potentially redundant street-view images, we apply Cutout [11]. For every street-view image, with probability $q$, we blackout 40% of pixels, by covering the image with a randomly placed black rectangle. Using this approach, we force the model to rely on multiple different parts of an image when performing classification.

**Splitting the street-view images (SS)** To perform projection pooling, we average the street-view feature maps $f_i$ along the height dimension, which gives a very rough feature stripe $s_i$. We also investigate a different strategy, which allows the model to take into account the differences between lower, middle, and upper parts of the street-view images. We split the feature representation into $k$ different tensors along the height dimension, and concatenate them along the depth dimension. In other words, we shift the height dimension into the depth dimension. From a feature map of size $[h, w, d]$, we obtain one of size $[\frac{h}{k}, w, d \cdot k]$ and treat it as $f_i$ in the projection pooling layer.

### 6.2 Results

In this subsection, we discuss the experimental results.

| Model                        | Constr | #Floors | Roof | Roof | Facade | Occup | Avg  |
|------------------------------|--------|---------|------|------|--------|-------|------|
| TV - only top-view           | 72.2   | 57.2    | 79.2 | 90.9 | 51.1   | 70.0  | **70.10** |
| SV - only street-view        | 73.4   | 72.2    | 76.7 | 87.0 | 61.5   | 74.2  | **74.14** |
| SV+TV, separately            | 73.3   | 72.5    | 77.7 | 89.8 | 60.3   | 74.2  | **74.64** |
| SV+TV, CNN$_{TV}=CNN_{SV}$   | 73.1   | 72.0    | 79.7 | 91.1 | 58.5   | 72.5  | **74.49** |
| SV+TV                        | 73.0   | 72.7    | 79.4 | 91.4 | 57.9   | 73.2  | **74.59** |
| Projection pooling           | 76.0   | 75.6    | 81.3 | 91.9 | 62.4   | 76.6  | **77.28** |
Holistic Multi-View Building Analysis in the Wild with Projection Pooling

In Table 1, we present a comparison between competitive methods and the new proposed model with the projection pooling layer. Using only the top-view images gives worse results than using only the street-view photos. It is still better than one would expect, given that most parts of the building are not visible on the top-view image. Unsurprisingly, using the street-view rather than the top-view gives the most substantial gains when counting the number of floors (+15%) and predicting facade material (+10.4%). On the other hand, top-view images are more informative when it comes to roof geometry (+3.9%) and roof pitch (+2.5%).

Combining both top-view and street-view images gives better results than using only one of the sources, but only by 0.35% than when using street-view alone. Separate networks for the top and street-view images yield a small improvement. We obtain the best baseline results when training the top-view and street-view networks independently on different tasks.

Incorporating the projection pooling layer results in about 3% accuracy improvement over the best baseline.

Table 2. Ablation experiments for the model with projection pooling layer. SS - stripe sampling; ID - image dropout; TH - projection thickness; CU - cutout probability; SS - number of vertical splits for street-view images.

| Constr | #Floors | Roof Pitch | Roof Geom | Facade Material | Occup | Avg |
|--------|---------|------------|-----------|-----------------|-------|-----|
| N 0 1 0 0 1 | 76 | 73.6 | 78.9 | 91.3 | 59.5 | 75.2 | 75.76 |
| S 0 1 0 1 | 75.7 | 73.3 | 79 | 91.5 | 59.5 | 75.7 | 75.78 |
| A 0 1 0 1 | 76.1 | 73.5 | 78.7 | 91.6 | 60 | 75.7 | 75.93 |
| A 0.5 1 0 1 | 75.9 | 73.1 | 80 | 91.8 | 60.2 | 75.8 | 76.16 |
| A 0.5 3 0 1 | 76.4 | 74 | 80.2 | 91.5 | 60.8 | 76.4 | 76.54 |
| A 0.5 3 0.5 1 | 76.3 | 73.5 | 80.5 | 92.1 | 62.3 | 76.5 | 76.87 |
| A 0.5 3 0.5 3 | 76 | 75.6 | 81.3 | 91.9 | 62.4 | 76.6 | 77.28 |

We also performed an ablation study presented in Table 2. First, we compare different stripe sampling strategies, and find that using the averaging strategy gives the best results. Adding image dropout with probability \( p = 50\% \) further regularizes the model, and increases average accuracy. We tested values higher than 1 for projection thickness, and found that using 3 improves performance, while increasing beyond this number did not bring further gains. Applying cutout with probability \( q = 50\% \) improves our model even further, which shows the importance of using a wide variety of regularization techniques. The final improvement comes from splitting the street-view feature maps into three tensors, and concatenating them along the depth dimension, as described in Subsection 6.1. In
this way, lower, middle, and upper parts of the building are separately projected, which is especially helpful for discrimination of the number of floors.

Table 3. Impact of the maximum number of street-view images per example on model performance.

| Constr | #Floors | Roof Pitch | Roof Geom | Facade Material | Occup | Avg  |
|--------|---------|------------|-----------|-----------------|--------|------|
| 1xSV + TV | 75.7 | 72.0 | 80.3 | 91.0 | 60.3 | 74.4 | 75.60 |
| Up to 2xSV + TV | 76.3 | 74.4 | 81.1 | 92.1 | 61.9 | 75.9 | 76.96 |
| Up to 3xSV + TV | 76.1 | 75.2 | 81.4 | 91.9 | 62.3 | 76.4 | 77.22 |
| Up to 4xSV + TV | 76.1 | 75.5 | 81.1 | 92.1 | 62.5 | 76.3 | 77.26 |
| All SV + TV    | 76.0 | 75.6 | 81.3 | 91.9 | 62.4 | 76.6 | 77.28 |

Impact of multiple images We investigate the impact of varying the number of street-view images per building. For \( k \in \{1, 2, 3, 4\} \), we test our best model with a restriction to use at most \( k \) street-view images.

The results of this comparison are shown in Table 3. We see a substantial gain from using more than one street-view image, suggesting that photos from multiple angles are often necessary for correct classification. On the other hand, the benefits of adding more street-view images quickly plateau, as there is a considerable overlap of information provided by the different street-view photos.

7 Conclusions

Our study presents a novel solution to a practical problem of building understanding. For the first time, this problem is approached by using the building geometry inside a deep neural architecture to create a unified high-dimensional representation. We propose a new way to integrate the features from multiple views called projection pooling. It is a general method for creating a unified representation of 3D objects from orthogonal projections and is particularly well suited for building analysis. We propose a model for building feature recognition, which incorporates the projection pooling layer, and its results are superior to the highly tuned baseline models.

We build a new dataset focused on fine-grained building attributes, and analyze fusion techniques for integrating information from multiple views. The dataset establishes a demanding benchmark for the state-of-the-art deep learning methods, with challenges far beyond those of ImageNet. It requires reasoning about numerous images at once to give accurate results. In the future, we plan to expand the dataset for the detection of objects, such as doors and windows.
References

1. Andersson, V.O., Birck, M.A., Araujo, R.M.: Investigating crime rate prediction using street-level images and siamese convolutional neural networks. In: LatinAmerican Workshop on Computational Neuroscience. pp. 81–93. Springer (2017)
2. Armagan, A., Hirzer, M., Roth, P.M., Lepetit, V.: Accurate camera registration in urban environments using high-level feature matching. In: Proceedings of the British Machine Vision Conference (2017)
3. Armagan, A., Hirzer, M., Roth, P.M., Lepetit, V.: Learning to align semantic segmentation and 2.5 d maps for geolocalization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3425–3432 (2017)
4. Bódis-Szomorú, A., Riemensneider, H., Van Gool, L.: Efficient volumetric fusion of airborne and street-side data for urban reconstruction. In: International Conference on Pattern Recognition. pp. 3204–3209. IEEE (2016)
5. Bulbul, A., Dahyot, R.: Social media based 3d visual popularity. Computers & Graphics 63, 28–36 (2017)
6. Cao, R., Zhu, J., Tu, W., Li, Q., Cao, J., Liu, B., Zhang, Q., Qiu, G.: Integrating aerial and street view images for urban land use classification. Remote Sensing 10(10), 1553 (2018)
7. Chen, X., Ma, H., Wan, J., Li, B., Xia, T.: Multi-view 3d object detection network for autonomous driving. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1907–1915 (2017)
8. Chu, H., Wang, S., Urtasun, R., Fidler, S.: Housecraft: Building houses from rental ads and street views. In: Proceedings of the European Conference on Computer Vision. pp. 500–516. Springer (2016)
9. Cohen, A., Oswald, M.R., Liu, Y., Pollefeys, M.: Symmetry-aware façade parsing with occlusions. In: 2017 International Conference on 3D Vision (3DV). pp. 393–401. IEEE (2017)
10. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3213–3223 (2016)
11. DeVries, T., Taylor, G.W.: Improved regularization of convolutional neural networks with cutout. arXiv preprint arXiv:1708.04552 (2017)
12. Dubey, A., Naik, N., Parikh, D., Raskar, R., Hidalgo, C.A.: Deep learning the city: Quantifying urban perception at a global scale. In: Proceedings of the European Conference on Computer Vision. pp. 196–212. Springer (2016)
13. Feng, T., Truong, Q.T., Thanh Nguyen, D., Yu Koh, J., Yu, L.F., Binder, A., Yeung, S.K.: Urban zoning using higher-order markov random fields on multi-view imagery data. In: Proceedings of the European Conference on Computer Vision. pp. 614–630 (2018)
14. Gadde, R., Jampani, V., Marlet, R., Gehler, P.V.: Efficient 2d and 3d facade segmentation using auto-context. IEEE transactions on pattern analysis and machine intelligence 40(5), 1273–1280 (2017)
15. Gebru, T., Krause, J., Wang, Y., Chen, D., Deng, J., Aiden, E.L., Fei-Fei, L.: Using deep learning and google street view to estimate the demographic makeup of neighborhoods across the united states. Proceedings of the National Academy of Sciences 114(50), 13108–13113 (2017)
16. Gebru, T., Krause, J., Wang, Y., Chen, D., Deng, J., Fei-Fei, L.: Fine-grained car detection for visual census estimation. In: Thirty-First AAAI Conference on Artificial Intelligence (2017)
17. He, K., Gkioxari, G., Dollár, P., Girshick, R.B.: Mask R-CNN. In: Proceedings of the IEEE international conference on computer vision. pp. 2961–2969 (2017)
18. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016). https://doi.org/10.1109/CVPR.2016.90
19. Hirzer, M., Arth, C., Roth, P.M., Lepetit, V.: Efficient 3d tracking in urban environments with semantic segmentation. In: Proceedings of the British Machine Vision Conference (2017)
20. Hoffmann, E.J., Wang, Y., Werner, M., Kang, J., Zhu, X.X.: Model fusion for building type classification from aerial and street view images. Remote Sensing 11(11), 1259 (2019)
21. Hu, S., Feng, M., Nguyen, R.M., Hee Lee, G.: Cvm-net: Cross-view matching network for image-based ground-to-aerial geo-localization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7258–7267 (2018)
22. Kang, J., Körner, M., Wang, Y., Taubenböck, H., Zhu, X.X.: Building instance classification using street view images. ISPRS journal of photogrammetry and remote sensing 145, 44–59 (2018)
23. Koch, D., Despotovic, M., Sakeena, M., Düller, M., Zeppelzauer, M.: Visual estimation of building condition with patch-level convnets. In: Proceedings of the 2018 ACM Workshop on Multimedia for Real Estate Tech. pp. 12–17. ACM (2018)
24. Kozinski, M., Gadde, R., Zagoruyko, S., Obozinski, G., Marlet, R.: A mrf shape prior for facade parsing with occlusions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2820–2828 (2015)
25. Kozinski, M., Obozinski, G., Marlet, R.: Beyond procedural facade parsing: Bidirectional alignment via linear programming. In: Asian Conference on Computer Vision. pp. 79–94. Springer (2014)
26. Li, Y., Hu, Q., Wu, M., Liu, J., Wu, X.: Extraction and simplification of building façade pieces from mobile laser scanner point clouds for 3d street view services. ISPRS International Journal of Geo-Information 5(12), 231 (2016)
27. Liang, M., Yang, B., Wang, S., Urtasun, R.: Deep continuous fusion for multi-sensor 3d object detection. In: Proceedings of the European Conference on Computer Vision. pp. 641–656 (2018)
28. Liu, H., Zhang, J., Zhu, J., Hoi, S.C.H.: Deepfaçade: A deep learning approach to façade parsing. In: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017. pp. 2301–2307 (2017). https://doi.org/10.24963/ijcai.2017/320
29. Lotte, R., Haala, N., Karpina, M., Aragão, L., Shimabukuro, Y., et al.: 3d façade labeling over complex scenarios: A case study using convolutional neural network and structure-from-motion. Remote Sensing 10(9), 1435 (2018)
30. Martinovic, A., Van Gool, L.: Bayesian grammar learning for inverse procedural modeling. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 201–208 (2013)
31. Mathias, M., Martinovic, A., Van Gool, L.: Atlas: A three-layered approach to façade parsing. International Journal of Computer Vision 118(1), 22–48 (2016)
32. Mitra, N.J., Pauly, M., Wand, M., Ceylan, D.: Symmetry in 3d geometry: Extraction and applications. In: Eurographics. pp. 29–51 (2012). https://doi.org/10.2312/conf/EG2012/stars/029-051
33. Musialski, P., Wonka, P., Recheis, M., Maihofer, S., Purgathofer, W.: Symmetry-based façade repair. In: Proceedings of the Vision, Modeling, and Visualization Workshop 2009, November 16-18, 2009, Braunschweig, Germany. pp. 3-10 (2009)
34. Panphattarasap, P., Calway, A.: Automated map reading: Image based localisation in 2-d maps using binary semantic descriptors. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 6341–6348. IEEE (2018)
35. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in pytorch. In: NIPS-W (2017)
36. Riemenschneider, H., Bödis-Szomorú, A., Weissenberg, J., Van Gool, L.: Learning where to classify in multi-view semantic segmentation. In: Proceedings of the European Conference on Computer Vision. pp. 516–532. Springer (2014)
37. Riemenschneider, H., Krispel, U., Thaller, W., Donoser, M., Havemann, S., Fellner, D., Bischof, H.: Irregular lattices for complex shape grammar facade parsing. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1640–1647. IEEE (2012)
38. Roddick, T., Kendall, A., Cipolla, R.: Orthographic feature transform for monocular 3d object detection. arXiv preprint arXiv:1811.08188 (2018)
39. Serna, A., Marcotegui, B., Hernández, J.: Segmentation of façades from urban 3d point clouds using geometrical and morphological attribute-based operators. ISPRS International Journal of Geo-Information 5(1), 6 (2016)
40. Song, W., Workman, S., Hadzic, A., Zhang, X., Green, E., Chen, M., Souleyrette, R., Jacobs, N.: Farsa: Fully automated roadway safety assessment. In: Winter Conference on Applications of Computer Vision (WACV). pp. 521–529. IEEE (2018)
41. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research 15(1), 1929–1958 (2014)
42. Srivastava, S., Vargas Muñoz, J.E., Lobry, S., Tuia, D.: Fine-grained landuse characterization using ground-based pictures: a deep learning solution based on globally available data. International Journal of Geographical Information Science pp. 1–20 (2018)
43. Srivastava, S., Vargas-Muñoz, J.E., Tuia, D.: Understanding urban landuse from the above and ground perspectives: A deep learning, multimodal solution. Remote Sensing of Environment 228, 129–143 (2019)
44. Stone, H.: Exposure and vulnerability for seismic risk evaluations. Ph.D. thesis, UCL (University College London) (2018)
45. Teboul, O., Kokkinos, I., Simon, L., Koutsourakis, P., Paragios, N.: Shape grammar parsing via reinforcement learning. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2273–2280. IEEE (2011)
46. Tyleček, R., Sára, R.: Spatial pattern templates for recognition of objects with regular structure. In: German Conference on Pattern Recognition. pp. 364–374. Springer (2013)
47. Wegner, J.D., Branson, S., Hall, D., Schindler, K., Perona, P.: Cataloging public objects using aerial and street-level images-urban trees. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6014–6023 (2016)
48. Workman, S., Souvenir, R., Jacobs, N.: Wide-area image geolocalization with aerial reference imagery. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 3961–3969 (2015)
49. Workman, S., Zhai, M., Crandall, D.J., Jacobs, N.: A unified model for near and remote sensing. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2688–2697 (2017)
50. Wu, C., Frahm, J.M., Pollefeys, M.: Detecting large repetitive structures with salient boundaries. In: Proceedings of the European Conference on Computer Vision. pp. 142–155. Springer (2010)
51. Yang, M.Y., Förstner, W.: A hierarchical conditional random field model for labeling and classifying images of man-made scenes. In: 2011 IEEE international conference on computer vision workshops (ICCV Workshops). pp. 196–203. IEEE (2011)
52. Zeppelzauer, M., Despotovic, M., Sakeena, M., Koch, D., Döller, M.: Automatic prediction of building age from photographs. In: Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval. pp. 126–134. ACM (2018)
53. Zhai, M., Bessinger, Z., Workman, S., Jacobs, N.: Predicting ground-level scene layout from aerial imagery. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 867–875 (2017)
54. Zhang, H., Xu, K., Jiang, W., Lin, J., Cohen-Or, D., Chen, B.: Layered analysis of irregular facades via symmetry maximization. ACM Transactions on Graphics (TOG) 32(4), 121–1 (2013)
55. Zhu, Y., Deng, X., Newsam, S.: Fine-grained land use classification at the city scale using ground-level images. IEEE Transactions on Multimedia (2019)
1 Appendix

1.1 Resolution of the top-view imagery

Aerial photos’ spatial resolution depends on the latitude and can be expressed as zoom 19 using the slippy map definition. The exact spatial resolution $r$, measured in meters per pixel, is expressed by the following formula:

$$ r = \text{equator length} \cdot \cos(\text{latitude}) \cdot \frac{1}{2^{19}} \cdot \frac{1}{2^{8}}. $$

The factor of $2^{-19}$ corresponds to the zoom, whereas the factor of $2^{-8}$ matches with the square tile size of 256 pixels.

1.2 Additional statistics about the dataset

![Fig. 1. A world heatmap of the building locations for our dataset. Our dataset represents a broad variety of buildings and architectural styles.](image)
Fig. 2. Examples of street-view images for every class in the dataset together with the building counter for that class in the training dataset. The dataset presents many challenges, such as occlusions and small object recognition from afar. Some images contain multiple buildings, confusing the model, and a few of them even miss the correct building.
1.3 Dataset examples

Fig. 3. Photos of a shopping mall from completely different perspectives. The model sees three different sides of the building from different angles and must be able to filter out the cars appearing in images.

Fig. 4. A typical example from an urban area, where more than one building is visible on the photo. Moreover, it is not ideally cropped in the second image. Observe that the roof pitch could be correctly estimated only from the third photo.
Fig. 5. Different photos of the same building highlighting various difficulties of the dataset. The first image is from a very close distance. The second image is partially occluded; the third one is also occluded by a visible car. The last one is from afar. Note that we can infer the construction type only from the first photo, where the bricks can be seen, whereas the last photo clearly shows the number of stories.

Fig. 6. Four different photos of the same building. They all look like different buildings. The last image is partially occluded. The facade on the second and third photo might suggest the building is made of masonry, when the correct answer is reinforced concrete.
Fig. 7. A building which is almost completely occluded but we are still able to infer some of its features. It has one floor, low roof pitch and is made of wood.

Fig. 8. A challenging example, where the building is occluded on the first photo, and partially occluded on the second one with cars present. Moreover, we can infer the number of floors only from the last photo, since the building is not fully visible in the third one. The photos are from different sides of the building, making the task even more difficult.
1.4 Dataset comparison

Table 1. Comparison of available to download datasets related to building analysis. 
#SV - number of street-view images, #TV - number of top-view images.

| Dataset                  | #SV  | #TV  | Location          | Dataset Type     | Multi-view |
|--------------------------|------|------|-------------------|------------------|------------|
| ICG Graz50 [10]         | 50   |      | Graz (Austria)    | 2D Segmentation  | ✓          |
| eTRIMS [7]               | 60   |      | Multiple          | 2D Segmentation  | ✓          |
| ENPC ArtDeco [5]         | 79   |      | Paris (France)    | 2D Segmentation  | ✓          |
| ECP Hausmannian [12]     | 104  |      | Paris (France)    | 2D Segmentation  | ✓          |
| CMP [13]                 | 378  |      | Multiple          | 2D Segmentation  | ✓          |
| LabelMeFacade [4]        | 945  |      | Multiple          | 2D Segmentation  | ✓          |
| ZuBuD [11]               | 1 005|      | Zurich (Switzerland) | 2D Segmentation | ✓          |
| RueMonge2014 [9]         | 428  |      | France            | 2D and 3D Segmentation | ✓          |
| SJC [8]                  | 175  |      | Brazil            | 2D and 3D Segmentation | ✓          |
| Limmatquai and Munsterhof [3] | 1 476 | 23 | Zurich (Switzerland) | 3D Reconstruction | ✓          |
| CVUSA [14]               | 1 588 655 | 879 318 | USA              | Geolocalization  | ✓          |
| Kang et al. [6]          | 196 58 |      | USA, Canada       | Classification (occupancy type) | ✓          |
| UC Merced [16]           | 2 100 |      | USA              | Classification (land use) | ✓          |
| DeepSat [2]              | 950 000 |    | Global            | Classification (land use) | ✓          |
| Albert et al [1]         | 140 000 |   | Europe            | Classification (land use) | ✓          |
| Brooklyn and Queens [15] | 38 603 | 10 044 | NYC (USA)        | Classification (3 attributes) | ✓          |
| Ours                     | 39 752 | 9 674 | Global            | Classification (6 attributes) | ✓          |
1.5 Projection pooling visualizations

Fig. 9. Larger version of Projection Pooling Visualization. Each row corresponds to one street-view image.
1.6 Efficiently computing the visible parts of the building

For each street-view image we compute which part of the top-view building polygon is visible. Note that many real-life building outlines are non-convex, and so the visible part might be a disconnected set of pieces of sides of the building polygon. We also take into account the occlusions caused by adjacent buildings. Here, we describe the approaches we investigated, comparing their time complexities in terms of the number of sides of the polygon $n$.

First, let us describe a naive approach that works in $O(n^2)$. We begin by fixing a single side, and computing what part of it is visible (possibly the entire side is occluded, in which case it will be ignored). To check visibility, we simply iterate through all the remaining sides, checking for occlusions.

In practice, we noticed that using a naive $O(n^2)$ algorithm was considerably slowing down both training and inference. This is because the number of polygon vertices – and thus sides – in our dataset can be quite large. In particular, one can often find multiple vertices near the building corners, due to post-processing of human annotations for loop closing. Therefore, we also developed a faster sweep line-based algorithm, which works in $O(n \log n)$ time.

![Fig. 10. Visualization of the radial sweep algorithm. We denote a single position of the ray with a solid line, where the arrow shows the direction of movement. The last and next events are denoted with dotted lines. The current set $S$ contains the two bolded sides of the building polygon, and the bottom one is visible. In red we show the visible side limited to the current range of ray angles, which will be included in the final result. When we process the next event, two more sides will be added to $S$, and the visible building side will change.](image)
The sweep line algorithm works as follows. We start with a ray originating at the street-view image location and pointing in an arbitrary direction, and rotate the ray until it makes a complete turn. Intuitively, any moment when the ray intersects the building polygon corresponds to some part of the building being visible, and the visible part is the closest intersection point of the ray with the building polygon. At any point during the sweep, we maintain the set $S$ of all polygon sides intersected by the ray. Each side needs to be inserted into $S$ exactly once during the sweep, and then (at a later point) deleted. Therefore, even though the radial sweep is a continuous process, there are only $2n$ events (each corresponding to a specific angle of the ray) that change the contents of $S$. Note that the polygon part visible at a given moment corresponds to a side in $S$ which intersects the ray closest to the origin, and this does not change as long as the set $S$ does not change. Thus, between every two consecutive events, we need to look up the closest segment in $S$, and add a part of this segment to the final result. The segment of interest can be extracted from $S$ in $O(\log n)$ time, assuming $S$ is implemented as a balanced Binary Search Tree, where the elements (polygon sides) are sorted according to their intersection with the ray. Thus, the complexity of the entire algorithm is indeed $O(n \log n)$. In Figure 10 we show a single moment during the radial sweep.
[1] Albert, A., Kaur, J., Gonzalez, M.C.: Using convolutional networks and satellite imagery to identify patterns in urban environments at a large scale. In: Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining. pp. 1357–1366 (2017)

[2] Basu, S., Ganguly, S., Mukhopadhyay, S., DiBiano, R., Karki, M., Nemani, R.: Deepsat: a learning framework for satellite imagery. In: Proceedings of the 23rd SIGSPATIAL international conference on advances in geographic information systems. pp. 1–10 (2015)

[3] Bódis-Szomorú, A., Riemenschneider, H., Van Gool, L.: Efficient volumetric fusion of airborne and street-side data for urban reconstruction. In: International Conference on Pattern Recognition. pp. 3204–3209. IEEE (2016)

[4] Frohlich, B., Rodner, E., Denzler, J.: A fast approach for pixelwise labeling of facade images. In: 2010 20th International Conference on Pattern Recognition. pp. 3029–3032. IEEE (2010)

[5] Gadde, R., Marlet, R., Paragios, N.: Learning grammars for architecture-specific facade parsing. International Journal of Computer Vision 117(3), 290–316 (2016)

[6] Kang, J., Körner, M., Wang, Y., Taubenböck, H., Zhu, X.X.: Building instance classification using street view images. ISPRS journal of photogrammetry and remote sensing 145, 44–59 (2018)

[7] Korč, F., Förstner, W.: eTRIMS Image Database for interpreting images of man-made scenes. Tech. Rep. TR-IGG-P-2009-01, Dept. of Photogrammetry, University of Bonn (April 2009)

[8] Lotte, R., Haala, N., Karpina, M., Aragão, L., Shimabukuro, Y., et al.: 3d façade labeling over complex scenarios: A case study using convolutional neural network and structure-from-motion. Remote Sensing 10(9), 1435 (2018)

[9] Riemenschneider, H., Bódis-Szomorú, A., Weissenberg, J., Van Gool, L.: Learning where to classify in multi-view semantic segmentation. In: Proceedings of the European Conference on Computer Vision. pp. 516–532. Springer (2014)

[10] Riemenschneider, H., Krispel, U., Thaller, W., Donoser, M., Havemann, S., Fellner, D., Bischof, H.: Irregular lattices for complex shape grammar facade parsing. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1640–1647. IEEE (2012)

[11] Shao, H., Svoboda, T., Van Gool, L.: Zubud-zurich buildings database for image based recognition. Computer Vision Lab, Swiss Federal Institute of Technology, Switzerland, Tech. Rep 260(20), 6 (2003)

[12] Teboul, O., Kokkinos, I., Simon, L., Koutsourakis, P., Paragios, N.: Shape grammar parsing via reinforcement learning. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2273–2280. IEEE (2011)
[13] Tyleček, R., Šára, R.: Spatial pattern templates for recognition of objects with regular structure. In: German Conference on Pattern Recognition. pp. 364–374. Springer (2013)

[14] Workman, S., Souvenir, R., Jacobs, N.: Wide-area image geolocalization with aerial reference imagery. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 3961–3969 (2015)

[15] Workman, S., Zhai, M., Crandall, D.J., Jacobs, N.: A unified model for near and remote sensing. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2688–2697 (2017)

[16] Yang, Y., Newsam, S.: Bag-of-visual-words and spatial extensions for land-use classification. In: Proceedings of the 18th SIGSPATIAL international conference on advances in geographic information systems. pp. 270–279 (2010)