Deep Learning for the Classification of Building Facades

DOMINIK LAUPHEIMER[1] & NORBERT HAALA[1]

Abstract: In recent years, 3D models containing both geometric and semantic information have become of great public interest. The geometric information of 3D models can be provided by (a combination of) photogrammetric methods, laser scanning and traditional surveying. For providing the semantic information for 3D urban models in an automated way, we established an end-to-end approach for classifying images of building facades into five different utility classes (commercial, hybrid, residential, specialUse, underConstruction) by using Convolutional Neural Networks (CNNs). We did several experiments on different data sets with various CNNs for evaluating the performance of this approach. Using Class Activation Maps (CAMs), we examined which features are learned during the training process in order to sort the facades into the considered classes.

1 Introduction

In the last few years Artificial Neural Networks (ANNs) have become state of the art in various research areas like image recognition, object recognition and speech recognition. ANNs are weighted graphs whose parameters $\boldsymbol{\theta}$ are trainable. Using ANNs is said to be Deep Learning. Particular ANNs that have been originally inspired by the human visual cortex are called Convolutional Neural Networks (CNNs). They proved to be very useful in the visual domain due to their properties like locality, parameter sharing and dimension reduction (RUSSAKOVSKY et al. 2015). The architecture of a CNN can be seen in Fig. 1.

Our aim is to establish an end-to-end approach for classifying street-level images of building facades into five different utility classes (see Tab. 1) by using CNNs. This is a highly complex real-world task struggling with various problems depending on image properties, object properties and environmental factors (variation of object scale and orientation, changing amount of buildings per image, cropped facades in images, occlusions, changing illumination, …). Moreover, the appearance of a facade is not necessarily in accordance with the actual usage of the building, e.g. there can be a medical practice in a former purely residential building without a constructional change of the facade. We assume, this discrepancy cannot be solved within the visual space.

For training the CNNs we use labeled Google Street View (GSV) images. The labels are acquired by the linking between labeled LoD3 building models and GSV images provided by TUTZAUER & HAALA (2017). The subsequent aim - using the trained CNN - is to enrich any 3D urban model where the semantic information is not already known.

We experiment on different data sets (see 2.2 Data Preparation) with various CNNs (VGG16, VGG19 (SIMONYAN & ZISSERMAN 2014), Resnet50 (HE et al. 2016), InceptionV3 (SZEGEDY et al. 2016), self-designed networks) in order to investigate the performance of our

[1] Universität Stuttgart, Institut für Photogrammetrie, Geschwister-Scholl-Str. 24D, D-70174 Stuttgart, E-Mail: [dominik.laupheimer, norbert.haala]@ifp.uni-stuttgart.de

end-to-end approach where no hand-crafted features are used. In order to examine which features are learned during the training process, we make use of Class Activation Maps (CAMs) visualizing areas being important for the network's decision (SELVARAJU et al. 2016). A human interpreter can derive the learned features by looking at those important areas. Furthermore, with the help of CAMs we investigate how our trained CNNs perform on data that has different properties than the used training data. Classification results and CAMs are shown in 2.4 Results.

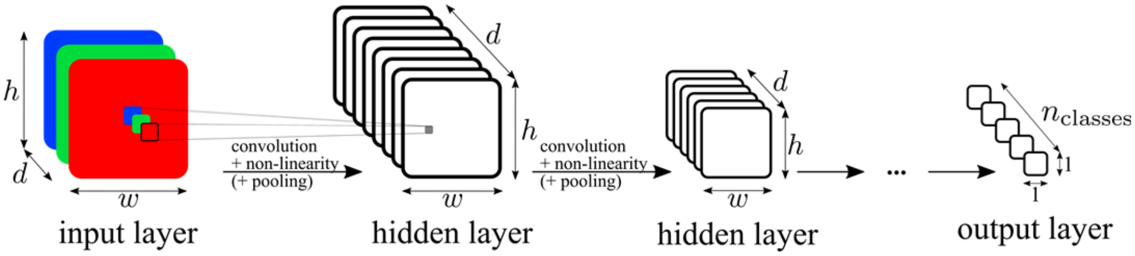


Fig. 1: Architecture of a CNN for multiclass classification with an input layer (here: RGB image), multiple hidden layers and an output layer. The amount of classes $n_{classes}$ defines the size of the output layer representing the estimated label vector $\hat{\boldsymbol{y}}$. Spatial dimensions w and h depend on the input image and the used convolutions (and poolings). The depth d of hidden convolutional layers depends on the amount of used convolutional kernels applied to the previous layer. Each depth slice is a so-called activation map.

Recent years have shown a great effort in scene understanding and consequently, extending 3D urban models with semantic information. Similar to our task, MOVSHOVITZ-ATTIAS et al. (2015) perform a fine-grained multi label classification of store fronts exhausting GSV data. It can be seen as a more detailed classification of our class *commercial*. MARTINOVIC et al. (2015) do segmentation and labeling of facades in 3D space. Unlike us, they classify only components of facades whereas we classify the facades as a whole. Therefore our classification is on a higher abstraction level.

Tab. 1: Considered classes of building facades for our end-to-end approach.

name	**definition**
commercial	purely commercial use
hybrid	mixture of commercial and residential use
residential	purely residential use
specialUse	anything else not matching the other four classes, e.g. gyms and churches
underConstruction	sites being under construction independently on their actual building class

2 Setup and Training of the CNNs

The aim of our end-to-end approach is to provide an automated pipeline for extracting semantic information in images showing building facades. As a first step, we perform image classification considering only five different classes (see Tab. 1) as we want to investigate the feasibility of such a pipeline when using CNNs. In order to learn features properly, a huge amount of labeled data for the training process has to be provided. As there doesn't exist any benchmark for our purposes, we have to provide a labeled data set on our own (see 2.2 Data Preparation). A labeled data set consists of images $\boldsymbol{x}$ labeled with a correspondent label $\boldsymbol{y}$, i.e. tuples $(\boldsymbol{x}, \boldsymbol{y})$. We provide different data sets with images for training (*train set*), for validating the network's performance during the training process (*validation set*) and for evaluating the network's performance after the training process (*test set*). Results are shown in 2.4 Results.

2.1 Pipeline

In order to use CNNs for image classification you have to train them by using ground truth data (see 2.3 Training). A labeled input image $(\boldsymbol{x}, \boldsymbol{y})$ is fed to the CNN, processed by its current parameters $\boldsymbol{\theta}$ which produce a predicted label vector $\hat{\boldsymbol{y}}$. The label vector can be interpreted as a probability density function over the considered classes. This output is compared to the ground truth $\boldsymbol{y}$ (one-hot label vector). The discrepancy between $\hat{\boldsymbol{y}}$ and $\boldsymbol{y}$ is backpropagated during the network wherefore the parameters $\boldsymbol{\theta}$ are updated (LECUN et al. 1998). The more labeled data are available for the training, the better the parameters can anneal to their ground truth values and the better is the achieved classification. Once training is done one can process any new unseen data considering the CNN as a black box accomplishing the classification (see Figure 2).

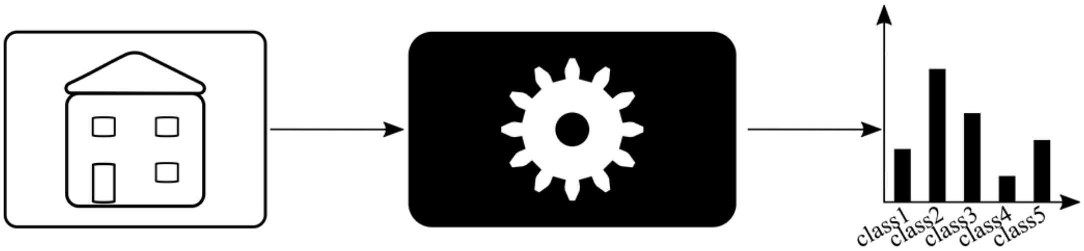


Fig. 2: Concept of the classification pipeline using CNNs. An input image showing a facade is processed by the CNN producing a probability density function over the considered classes

2.2 Data Preparation

Providing a big enough labeled data set in order to deal with the problem's complexity is a big challenge and a very tedious task. We provide different data sets for binary and multiclass classification tasks in order to investigate the impact of the separability of classes. All images are RGB images from GSV and labeled based on the fine-grained labeling given by TUTZAUER & HAALA (2017). These labels have to be sorted into our coarse classification given in Tab. 1. Obviously, in the real world there are more facades of residential buildings than facades belonging to other classes causing imbalanced data sets, initially. With the help of a priori data augmentation (horizontal flipping, warping, cropping, jittering and modification of saturation) we created equally distributed training data sets. I.e. every considered class has the same amount of samples.

2.3 Training

Generally, ANNs can be seen as function approximation machines consisting of a huge amount of parameters $\boldsymbol{\theta}$ which are learned during a training process by using a labeled *train set* (supervised learning). The schematic representation of an ANN is given in Figure 2. Equation (1) shows the mathematical representation of the function approximation machine (*left side*) and its ground truth relation (*right side*) with perfectly known parameters $\boldsymbol{T}$ resulting in perfect labels $\boldsymbol{y}$.

$$\hat{\boldsymbol{y}} = f(\boldsymbol{x}, \boldsymbol{\theta}) \qquad \boldsymbol{y} = f(\boldsymbol{x}, \boldsymbol{T}) \qquad (1)$$

The learning process adapts the originally randomly initialized parameters $\boldsymbol{\theta}$ in an optimal sense. Mathematically speaking, the learning process is an optimization problem where the trainable parameters have to minimize a cost function $C(\boldsymbol{y}, \hat{\boldsymbol{y}})$ measuring the discrepancy between $\hat{\boldsymbol{y}}$ and $\boldsymbol{y}$. C is a proxy measure for the classification error. Due to the architecture of CNNs the learned parameters are stored in convolutional masks. Hence, they can be interpreted as feature detectors. We used RGB images of size 400px × 400px. Our *train set* consists in total of 75.000 images (15.000 images per class) due to data augmentation. *Validation set* and *test set* consist of 350 images each (70 images per class).

2.4 Results

Our work shows, that an overall accuracy of approximately 64% can be achieved with current state-of-the art models (VGG16, VGG19, ResNet50, InceptionV3) when being evaluated on data sets provided by us. As expected, classes *specialUse* (accuracy: 25.71%) and *underConstruction* (accuracy: 48.57%) perform worse than the other classes due to their definitions (high intra-class variance, bad separability) and lower the overall accuracy by approximately 18%. Furthermore, *residential* performs best (accuracy: 98.57%), which is pleasant as the majority of real-world buildings belongs to this class.

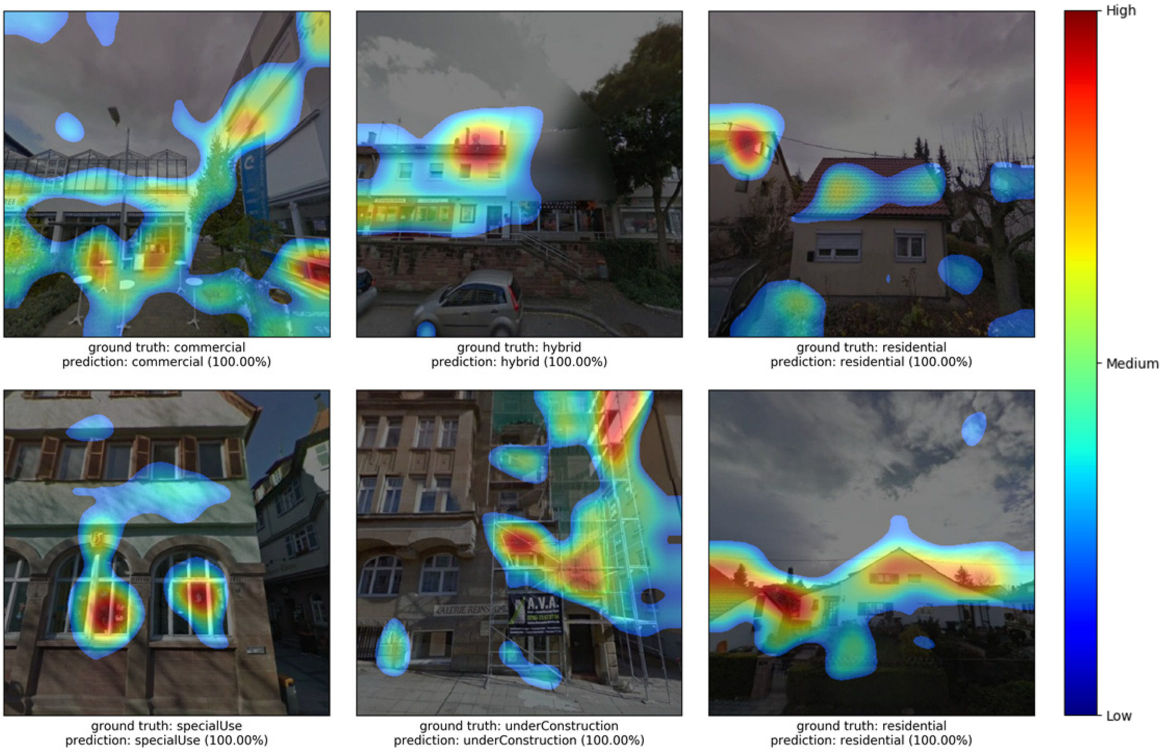

ground truth: commercial
prediction: commercial (100.00%)
ground truth: hybrid
prediction: hybrid (100.00%)
ground truth: residential
prediction: residential (100.00%)
ground truth: specialUse
prediction: specialUse (100.00%)
ground truth: underConstruction
prediction: underConstruction (100.00%)
ground truth: residential
prediction: residential (100.00%)
High
Medium
Low

Fig. 3: Some examples of correctly classified images overlaid with CAMs showing the class-specific important areas. The image of class *specialUse* shows a kindergarten (*bottom left*) with bricolages at the windows. The respective color of the CAM represents the importance regarding the network's prediction

By looking at images of correctly classified images overlaid with CAMs (see Figure 3) we can collect a subset of learned features for each class (see Table 2). Interestingly, they are similar to features a human operator would use instinctively. CAMs base on the activation maps of the last convolutional layer of a trained CNN. These activation maps contain high-level features while conserving the spatial information (which is discarded by fully connected layers). The gradient of the predicted class with respect to the activation map shows the importance regarding the network's decision. I.e. the CAM is class-specific.

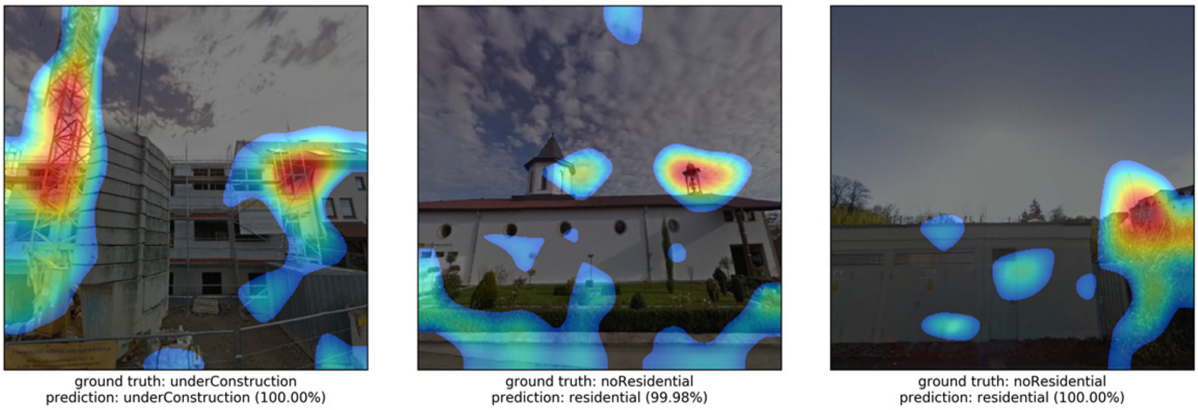
Tab. 2: Extracted learned features (interpreted by using CAMs)

class	**features**
commercial	windows (arrangement, size, amount), advertising panels/signal colors, shop windows, big doorways, roof shape (flat roof), huge chimneys
hybrid	windows (arrangement, size, amount), advertising panels/signal colors, shop windows, roof shape (saddle roof), roof extensions (e.g. dormers, chimneys or antennas), environmental objects (sky, vegetation, cars, humans), big doorways
residential	windows (arrangement, size, amount), roof shape (saddle roof), roof extensions (e.g. dormers, chimneys or antennas), environmental objects (sky, vegetation, cars, humans), arrangement/skyline of adjacent buildings
specialUse	building shape, decorated windows, special doorways
underConstruction	scaffolds, cranes

(a shows a correctly predicted construction site overlaid with its CAM. By evaluating CAMs we could locate the problems for the image classification task. There are misclassifications due to not detected features and misclassifications due to misinterpreted features (see (b). Apart from these misclassifications, there are classifications that contribute to the classification error as predictions and ground truth labels differ due to different focusing. From a human point of view the building covering the majority is decisive for labeling, but CNNs only take care of features with the most impact on predictions - independently of location and size within the image. Basically, these predictions are not necessarily false predictions, but they still are lowering the overall accuracy if they do not match the human-given label. As a consequence, the classification of building facades, currently realized in form of image classification, underestimates the real power. (c shows exemplarily an image of such a “false” prediction overlaid by its corresponding CAM.

Results of our binary and ternary classifiers show that classification results can be improved by improving class separability, i.e. by reducing the intra-class variance (overall accuracies for the binary/ternary case: approximately 88% / approximately 73%). Our pragmatic approach was to reduce the amount of classes in order to avoid a new labeling.

In addition to that, CAMs are used to investigate how trained CNNs perform on different data representations. For that purpose, CAD models in LoD3, textured meshes (textured with images of Google Earth) and cut out buildings of GSV images are considered (see Figure 5). In the following, we refer to them as non-image representations as the first two are 3D models and the latter are cut out images only. These representations have also been used by a user study on the human ability to perceive building classes from geometric representations (TUTZAUER & HAALA, 2017).


ground truth: underConstruction
prediction: underConstruction (100.00%)
ground truth: noResidential
prediction: residential (99.98%)
ground truth: noResidential
prediction: residential (100.00%)

(a): Correct prediction (b): False prediction (c): “False” prediction

Fig. 4: Example images overlaid by corresponding CAMs of correct and false predictions for binary classification. The small bell tower in Figure 4b is mistaken as a chimney causing a false prediction. The “false” prediction in Figure 4c is a result of the discrepancy between human labeling and automatic labeling. The transformer house occupying the major area is decisive for the human-given label. The CNN focuses on the marginal residential building

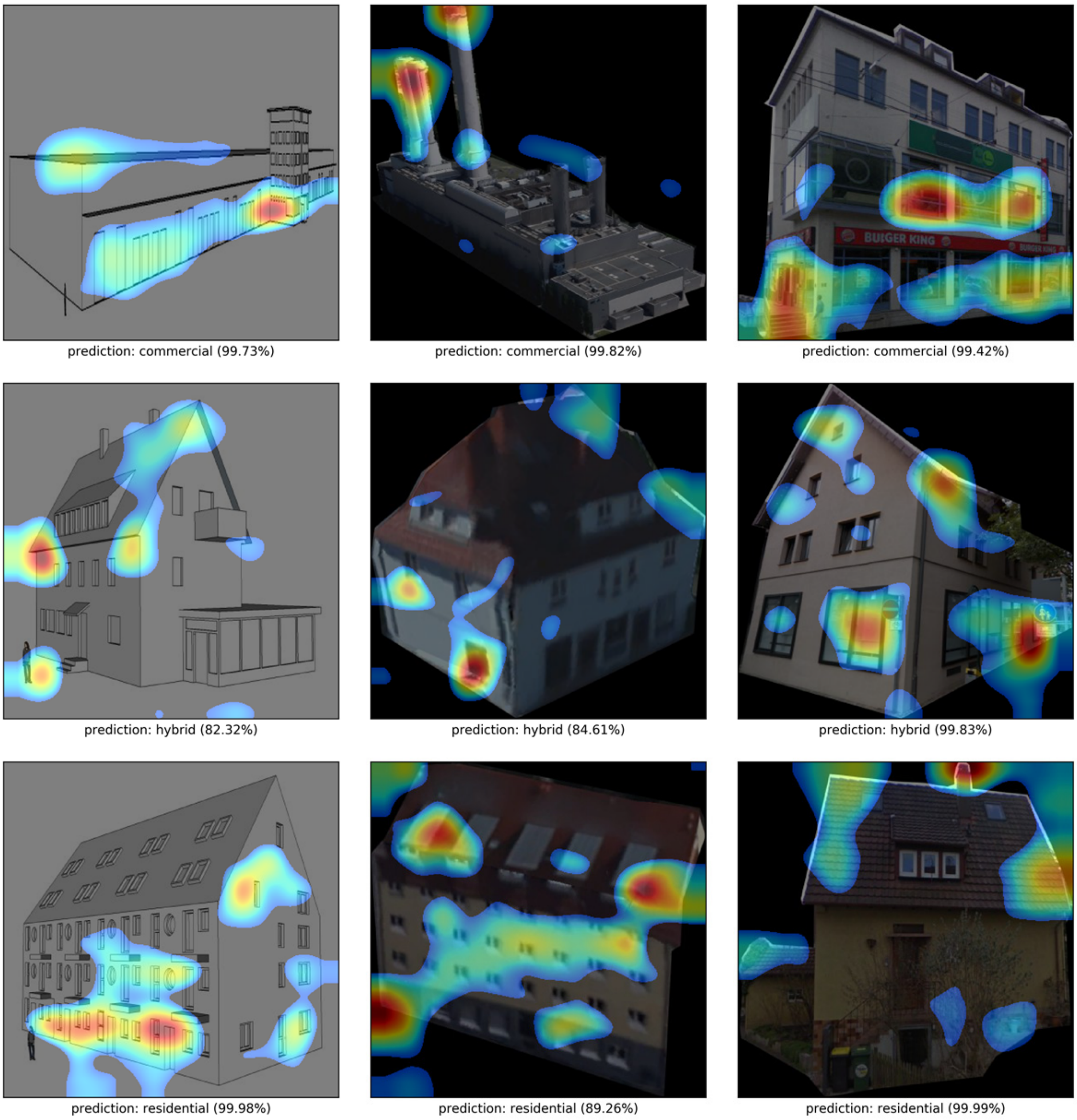
Since data has to be provided to the network in the same manner like during training, trained CNNs cannot be applied instantaneously to the considered non-image representations. As we trained our CNNs on 400×400×3 sized images we have to provide RGB images of the same size to the networks. Therefore, snapshots of those non-image representations are taken.

Figure 5 shows snapshots of different non-image representations overlaid by corresponding CAMs. All representations have in common that there is no environmental information. A small exception is the LoD3 model generated by SketchUp. A human is placed automatically next to every 3D model by the program. This is worth mentioning, as *human* is an important feature for classes *hybrid* and *residential* (see Table 2). Please note, snapshots of textured meshes are taken from a bird’s eye view which differs significantly from the exposure direction of our training images (street-level images).

Figure 6 exemplarily shows a correctly predicted building throughout all representations overlaid by corresponding CAMs. The first row shows snapshots of LoD3 models whereby in the left image the human is removed. It can be seen that the same areas activate for ground truth class *hybrid* (neglecting the human part in the right image). Removing the human means removing the most important feature (see the change of CAM’s coloring and the worsening of the prediction confidence). This shows the importance of environmental objects for the classification.

The lower left shows the snapshot of the textured mesh representation. Its exposure direction differs from street-level images used for training the network. The lower right image shows the cut out GSV image (overlaid by its CAM) which achieves the best result.

To summarize, the more information is carried by the representation (e.g. color, environmental objects, etc.) and the more the representation converges to the representation used during training, the better becomes the classification result. Snapshots of Figure 6 show that CNNs are a manifestation of *weak Artificial Intelligence*. CNNs can solve tasks reliably they are familiar with. The more the task's preconditions or the task itself differs from the task during training, the more limited is the classification quality.



prediction: commercial (99.73%)

prediction: commercial (99.82%)

prediction: commercial (99.42%)

prediction: hybrid (82.32%)

prediction: hybrid (84.61%)

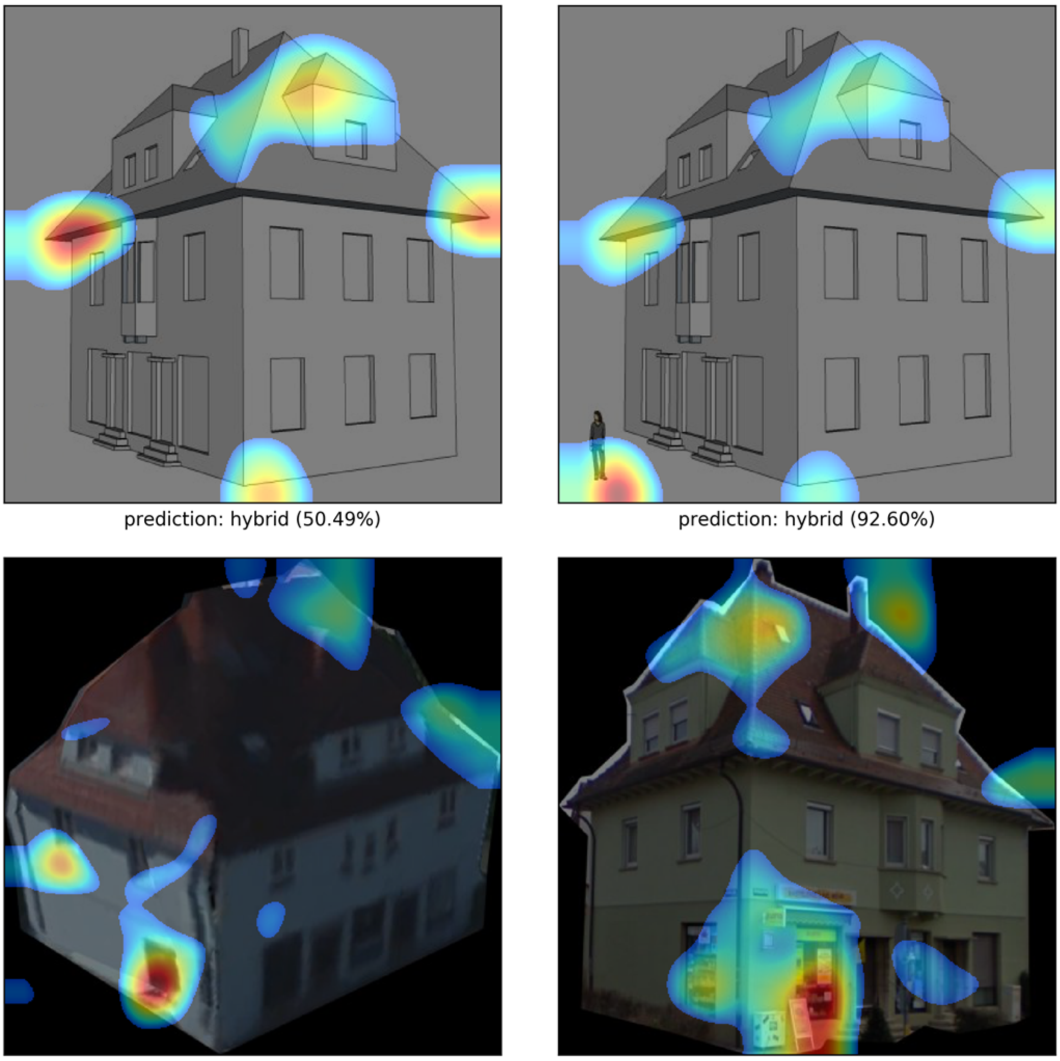
prediction: hybrid (99.83%)

prediction: residential (99.98%)

prediction: residential (89.26%)

prediction: residential (99.99%)

Fig. 5: Snapshots of different non-image representations that have been classified correctly overlaid by corresponding CAMs (ground truth in first row: *commercial*, ground truth in second row: *hybrid*, ground truth in last row: *residential*). The first column shows snapshots of LoD3 models. The second column shows snapshots of textured meshes (textured with Google Earth images). The last column shows cut out Google Street View images



prediction: hybrid (50.49%)

prediction: hybrid (92.60%)

prediction: hybrid (84.61%)

prediction: hybrid (99.98%)

Fig. 6: Snapshots of different non-image representations of the same building and the corresponding CAMs. Please note, the building is oriented differently in the snapshots. The ground truth label is class *hybrid*. The upper row shows a snapshot of the LoD3 model without a human (*left*) and one snapshot with a human (*right*). The second row shows a snapshot of the textured mesh (textured with Google Earth images) and a cut out Google Street View image (*from left to right*).

3 Conclusions and Outlook

The presented end-to-end approach achieves promising results for enriching 3D urban models automatically with semantic information. The overall error of 36% reflects the complexity of this real-world classification task and immediately suggests to reduce the classification error in future work. This can be achieved by gathering more training data. Avoiding "false" predictions might be insoluble in the context of image classification (see (c). In future work we suggest to go from image classification to object detection and object classification where this kind of misclassifications do not apply. Alternatively, a multi-label data set should be provided. In such a

data set, every image is tagged with all classes whose instances are visible in the image - sorted by covering area in descending order.
We have to note that reality is far too complex in order to label all images properly into the considered classes. It would be interesting to see how and in how many classes images are clustered by an unsupervised learning method. Based on these results one could rethink the current classes.
In order to form a judgment of the CNNs' performance, we want to conduct a user study on our data sets and determine human performance for comparison. We know there are images where humans have problems to classify the facades correctly.

4 Acknowledgment

The author thanks PATRICK TUTZAUER for his support during the period of the master thesis. Furthermore, we acknowledge the support of NVIDIA Corporation by donation of a Titan X Pascal GPU used to generate the presented results.

5 Bibliography

HE, K., ZHANG, X., REN, S. & SUN, J., 2016: Deep Residual Learning for Image Recognition. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 770-778.

LECUN, Y., BOTTOU, L., ORR, G.B. & MÜLLER, K.-R., 1998: Efficient BackProp. Neural Networks: Tricks of the Trade, 9-50.

MARTINOVIC, A., KNOPP, J., RIEMENSCHNEIDER, H. & VAN GOOL, L., 2015: 3D All The Way: Semantic Segmentation of Urban Scenes From Start to End in 3D. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 4456-4465.

MOVSHOVITZ-ATTIAS, Y, YU, Q., STUMPE, M. C., SHET, V., ARNOUD, S. & YATZIV, L., 2015: Ontological Supervision for Fine Grained Classification of Street View Storefronts. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1693-1702.

RUSSAKOVSKY, O., DENG, J., SU, H., KRAUSE, J., SATHEESH, S., MA, S., HUANG, Z., KARPATHY, A., KHOSLA, A., BERNSTEIN, M. S., BERG, A. C. & LI, F., 2015: ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision **115**, 211-252.

SELVARAJU, R. R., COGSWELL, M., DAS, A., VEDANTAM, R., PARIKH, D. & BATRA, D., 2016: Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization. See https://arxiv. org/abs/1610.02391 v3.

SIMONYAN, K. & ZISSERMAN, A., 2014: Very Deep Convolutional Networks for Large-Scale Image Recognition. Computing Research Repository (CoRR abs/1409.1556).

SZEGEDY, C., VANHOUCKE, V., IOFFE, S., SHLENS, J. & WOJNA, Z., 2016: Rethinking the Inception Architecture for Computer Vision. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2818-2826.

TUTZAUER, P. & HAALA, N., 2017: Processing of Crawled Urban Imagery for Building Use. Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci., XLII-1/W1, 143-149.