

Façade Segmentation in the Wild

John Femiani¹ Wamiq Reyaz Para² Niloy Mitra³ Peter Wonka²

Miami University¹ KAUST² UCL³

Abstract. Urban façade segmentation from automatically acquired imagery, in contrast to traditional image segmentation, poses several unique challenges. 360° photospheres captured from vehicles are an effective way to capture a large number of images, but this data presents difficult-to-model warping and stitching artifacts. In addition, each pixel can belong to multiple façade elements, and different facade elements (e.g., window, balcony, sill, etc.) are correlated and vary wildly in their characteristics. In this paper, we propose three network architectures of varying complexity to achieve multilabel semantic segmentation of façade images while exploiting their unique characteristics. Specifically, we propose a **MULTIFACSEGNET** architecture to assign multiple labels to each pixel, a **SEPARABLE** architecture as a low-rank formulation that encourages extraction of rectangular elements, and a **COMPATIBILITY** network that simultaneously seeks segmentation across facade element types allowing the network to ‘see’ intermediate output probabilities of the various façade element classes. Our results on benchmark datasets show significant improvements over existing façade segmentation approaches for the typical façade elements. For example, on one commonly used dataset the accuracy scores for `window` (the most important architectural element) increases from 0.91 to 0.97 percent compared to the best competing method, and comparable improvements on other element types.

Keywords: façade segmentation, learning architecture, semantic segmentation

1 Introduction

We propose a deep learning based solution for the per-pixel semantic segmentation and classification of façade images. Although many methods, both hand-crafted and deep learning based, exist for semantic image segmentation, façade images have certain special characteristics that prevent state-of-the-art semantic image segmentation methods from being directly used in our setting. Note that unlike typical semantic segmentation, façade per-pixel segmentation is special as a pixel can be *simultaneously* assigned to multiple labels (e.g., `window` and `balcony`). Further, many façade labels are very thin (e.g., `sill` or `pillar`) and some features such as the partitions between adjacent buildings require special handling for reliable detection.

Typically, façade images broadly come in three different flavors with increasing complexity (see Fig. 1). First, pre-rectified and cropped facade images, e.g., [1], which have been traditionally studied in façade segmentation and parsing. Second, images that are not pre-rectified and cropped but that are usually acquired with special care to have limited distortions. For example, images taken approximately from the front and containing single facades in the center covering most of the image. Third are input



Fig. 1. Different flavors of facade images considered in this paper; (left) a pre-rectified and cropped façade from the CMP dataset; (center-left) a carefully acquired image from eTRIMS, (center-right) a ‘wild’ image automatically acquired from a street-view panorama image, including blooming and stitching artifacts; (right) photosphere panoramic image.

images acquired in the ‘wild,’ particularly panoramic street-view images. These are captured automatically, at scale, and then aligned and stitched together to form 360° photospheres. We focus on the second and third types in this paper.

The advantage of ‘in-the-wild’ images is that they are widely available and can be easily acquired automatically from a vehicle. However, the simplicity of data acquisition comes at the cost of increased challenges. For example, during capture no special attention is put on photographic details like viewpoint selection to avoid occlusions due to vegetation or passing cars, or unfavorable lighting conditions such as direct sun exposure. Further, such panoramic street-view images contain many additional details besides buildings (e.g., foliage, scaffolding, vehicles, etc.). The focus of our work is to get the best possible facade segmentation and to extract architectural details from such raw images.

In this paper, we present a deep learning based solution that is inspired by recent success of semantic segmentation [2,3,4]. We propose three new network architectures particularly focusing on façade images. By explicitly accounting for the typical types of noise in façade datasets, enabling pixels to be assigned multiple labels, and exploiting the correlation among different façade elements, we demonstrate significant improvement over existing state-of-the-art façade segmentation methods. We evaluate our proposed method against a range of competing alternatives, and demonstrate significant improvements in terms of F_1 scores on multiple façade benchmark datasets.

2 Related Work

2.1 Traditional façade parsing methods

Façade parsing or façade segmentation was typically a mixture between traditional segmentation algorithms and finding ways to encode architectural priors. Architectural priors can be encoded using grammars [1,5,6], symmetry [7], matrix rank [8], MRFs [9,10], CRFs [11], element templates [12,13], rules [9], hard constraints [14], and more general energy functions [15,11,16]. The architectural information is typically encoded for architectural elements, e.g., rectangular regions in the segmentation

with the same label. Some examples for architectural priors are element sizes and aspect ratios, allowable neighborhood relationships (e.g., chimney has to be on top of the roof), spacing between elements, constraints of alignment and size between elements (e.g., all windows in a floor need to be aligned and of the same height). The low-level information can come from per-pixel classification algorithms, e.g., a boosted decision tree classifier [14], random forests [15], or mean-shift combined with recursive neural networks [9]. Multiple low level classifiers were evaluated in the ATLAS framework [11], but modern deep learning methods were not included. Alternately, it is possible to extract boxes of labeled regions using object detection algorithms [11,17,18].

One limitation of many traditional facade parsing methods is that they assume facade images that have been ortho-rectified and cropped (see Fig. 1-left). This allows to use much stronger architectural priors than facade parsing in the wild. For example, it is possible to make assumptions about shops being near the bottom of the image or windows being arranged in individual floors. Further, some data sets do not exhibit a strong variation in element arrangement and element size. For example the ECP dataset features many façades of the same (Haussmanian) architectural style.

2.2 Segmentation using CNNs

Semantic segmentation is a classic topic in computer vision and has been heavily researched. In recent years, with the amazing success of deep learning, the state-of-the-art methods have produced large improvements. We refer the readers to a recent survey [19] for a summary of the current methods and details about typical accuracy measures. In the following, we particularly focus on learning methods specialized to façade data.

Schmitz and Mayer [20] used a Convolutional Neural Network (CNN) to segment façade images. They use ‘de-convolution’ (also called transpose convolution) in order to up-sample the a CNN based on AlexNet, and they evaluated their results on the eTRIMS dataset.

The ‘DeepFacade’[18] approach to facade segmentation used a fully convolutional net with a special loss function in order to segment façade images. Their loss function penalized segmentation regions that were not horizontally and vertically symmetric.

However, our results indicate that a basic adaptation of general-purpose segmentation using SEGNET [2,3,4] is already better than the current state of the art deep learning methods. For example, Kelly et al. [21] used SEGNET to determine the locations of windows, balconies, and doors in large scale procedural models of urban areas based on streetview imagery. We compare our results to retrained SEGNET and DeepFacade as the currently best available solutions for façade image segmentation and report significant quantitative improvements.

2.3 Façade datasets

The ECP dataset [22] contains 104 images of rectified and cropped facades with the label set window, wall, balcony, door, roof, sky, shop. The eTRIMS dataset [23] contains 60 images with the label set building, car, door, pavement, road,

sky, vegetation, window. These images are not rectified and not cropped, however, the images stem from a very careful viewpoint selection. All images have an almost frontal view of a single facade that fills most of the image. The CMP dataset [24] contains 606 annotated images with the label set facade, molding, cornice, pillar, window, door, sill, blind, balcony, shop, deco, background.

3 Method

Our goal is to classify façade images into semantic pixel-level classes. Although this is a special case of semantic segmentation, certain aspects of the target dataset make the problem unique. First, the input images are often only partially rectified and suffer from various (unknown) camera and post-processing (e.g., stitching) artifacts. This makes it difficult to train for invariance under the difficult-to-model warping effects. Second, the desired features have vastly different proportions. For example, the windows and doors versus the ledges and window sills have very different aspect ratios. Finally, the typical façade features share strong inter- and intra-label relations, which can easily get lost if their labeling tasks are considered in isolation.

In our early experiments, we found that applying direct semantic segmentation pipelines result in fairly low F_1 scores especially on images such as Google Streetview (GSV) imagery. In our first attempt to build a façade segmentation classifier, we trained on CMP imagery only; but the classifier appeared to generalize poorly to GSV imagery (based on initial qualitative evaluation). Therefore, we doubled the size of our training data with images we annotated from GSV and observed that SEGNET is capable of giving competitive results when these images were included in our training and test sets, as indicated in row one of Table 5. However, we still identified multiple modifications that can significantly improve upon the baseline SEGNET approach for façade segmentation [25]. All of our proposed modifications could, in principle, be applied to other architectures such as fully convolutional networks [26], dilated nets [27,28], or U-nets [29] as well, however we limit the scope of this work to SEGNET.

3.1 Data Augmentation

Our goal is to segment images captured automatically, such as GSV imagery. This data poses several challenges because the images are the result of a number of processing and stitching steps that leave distortion artifacts in the images. Furthermore, the Geographic Positioning System (GPS) information associated with the images is imperfect, so façades are often not rectified or centered in the imagery; in fact there are often facades behind or around a central façade. We address this as follows:

- rectification using [17] to correct for errors in camera orientation;
- *data augmentation* to force the classifier to be robust to errors in rectification by applying a random perspective warp to each image; and
- random sampling and manual labeling GSV imagery in the datasets used for training and testing.



Fig. 2. An example illustrating the way GIS data is used to extract GSV imagery; shown (left) are building outlines from Amsterdam, Netherlands (violet, dotted lines), and their simplified and merged outlines (light green, solid lines). The location of GSV photospheres are indicated by green dots, and a selected wall and photo-sphere are indicated in red. The photosphere (right) is ray-cast onto a quad to form the façade image.

In order to collect façade images we project GSV photosphere images onto planes derived from the linear segments of a GIS polygonal building footprints layer¹. Building footprint datasets are quite common and can be obtained from public sites such as OpenStreetMap; the training data we consider ‘from the wild’ was extracted from OpenStreetMap building footprints boundaries of 20 large metropolitan areas around the world (Amsterdam, Antwerp, Athens, Atlanta, Auckland, Austin, Berlin, Bern, Bordeaux, Bucharest, Brisbane, Brussels, Cape Town, Chicago, Cleveland, Copenhagen, Dallas, Honolulu, and Hong Kong). However, the building footprints are digitized at a variety of levels of precision; with some examples capturing sub-façade details such as awnings or bay windows (see e.g., the upper left corner of Fig. 2). We simplify and merge building footprints with a tolerance of 2m in order to capture the dominant plane of each façade (or each group of collinear façades that form a wall of the building). We found GIS height data for buildings to be inconsistent, and treat each wall as though it were 40m tall. The spatial resolution is also effected by the horizontal angles between points on the façade and rays towards the photosphere, so we subdivide walls so that they are each approximately 40m long. Each linear segment is extruded by 40m upwards in order to form a 3D quadrilateral in which GSV photospheres will be sampled. The quad is subdivided to form a grid of samples at a resolution of 0.025m between samples. Finally, façade images were generated by ray-casting from the photosphere center of each to each grid point, and using the colors where the ray intersects the photosphere.

Based on the photosphere’s odometry information, we find that our images are approximately rectified (e.g., to within about 15°). However, the discrepancy is significant enough to obscure important horizontal and vertical alignments between features on the façades. In order to account for errors in the orientation of each photosphere, we extend each façade by 2m before projecting. This also results in overlap between images on long walls, increasing the odds that each image contains a complete façade. Then we use the single-image rectification approach of Affara *et al.* [17] in order to find a homography which increases the dominance of horizontal and vertical edges in each image.

¹ The source code to download footprints will be made available online after the blind-review process is complete.

For rectification, we use a classifier trained on Camvid data in order to identify pixels which are likely not part of a façade (sky, pedestrians, vehicles, or vegetation) and we remove those edges from consideration for rectification. During training, each image is warped by a uniform random perspective transformation which displaces the corners of the image by up to 20 percent of the image width.

The segmentation approaches described in the following sections all share the same corpus of training data, which includes imagery from the CMP dataset and also labeled images captured from GSV. New data that was acquired was rectified, and then the boundaries of façades were manually marked in each image. In a second phase, each individual façade was extracted to form a single-façade image, which was then completely labeled. During labeling, we encountered many out-of-model elements that were common in façade images. We list *window-AC units*, *awnings*, *fire escapes*, and *bay-windows* as examples. Our labeling process resulted in 22 common features, along with an ‘outlier’ label and an ‘undeterminable’ label for objects that we could not resolve. These include the 11 labels used for CMP and the 8 labels used in eTRIMS data which we use for comparisons. Of these labels, only the 11 CMP-labels were used for the model presented in this work; additional labels (*sky*, *roof*, and *chimney*) were added for cross-validation models presented in Tables 2 and 3.

3.2 Baseline: FACSEGNET

The SEGNET [25] assigns labels to each pixel using an auto-encoding approach; the VGG16 [30] convnet architecture is used as *encoding* layers to form a deep representation of an input image and then a mirrored series of *decoding* layers that reverse the VGG max-pooling operations are used to reconstruct a dense output label image. SEGNET originally presented results on driving scenes from the Camvid [31] data-set, but we re-targeted it to segment façade images using refinement learning.

The SEGNET architecture was modified slightly to allow the input layer to accommodate 512×512 images, which was a compromise keeping the spatial resolution of the images large enough to resolve façade elements, and keeping the images small enough to fit within the available GPU RAM. During training and inference, the input images were scaled so that their height was 512 pixels, and after scaling they were partitioned into horizontally overlapping tiles that were each 512 pixels wide. Narrow images were padded with mirrored copies; if it was wider than 512 pixels then it was tiled into the smallest number of 512-wide tiles that overlap by at least 16 pixels. After segmentation, the softmax scores for overlapping pixels were averaged and then re-scaled before taking their argmax in order to determine the final label. The loss function used by SEGNET is a weighted cross entropy; the original weights were used to deal with class-imbalance issues so that the weight of each label is inversely proportional to its frequency and the median frequency has a weight of one. We computed new frequencies based our combined CMP+GSV dataset and used median-frequency class balancing when refining SEGNET.

In this paper, we refer to SEGNET refined for façade segmentation as FACSEGNET. Starting with Camvid model weights, we trained on 80% of the labeled data (1200 tiles) using Stochastic Gradient Descent (SGD) with weight decay and a low learning

rate ($1e-6$) until the training loss plateaued (around 200 epochs). We continued to train the network until we reached 300 epochs.

3.3 Network Architecture 1: MULTIFACSEGNET

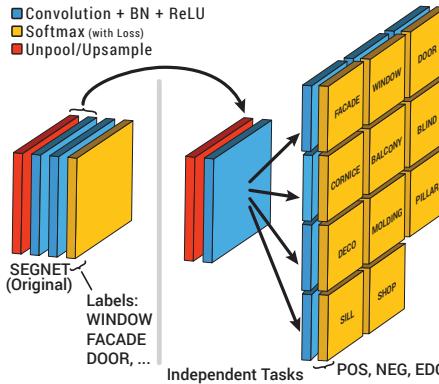


Fig. 3. The new output layers for independent labeling. The last convolutional layer and softmax layer of SEGNET are repeated 11 times; once per each type of object we aim to segment from façade images. For each type of object we label pixels as NEG, POS, or EDG, as well as an additional UNK label used to indicate lack of information during training (not used during inference).

Façade segmentation is a multi-label problem where each pixel can have multiple labels, i.e., the regions that are assigned to each label are not disjoint. For example, the regions of the image that are assigned the labels `window` and `balcony` often overlap. In both the CMP dataset, as well as the data we created from GSV imagery, ground-truth annotations were provided as polygonal shapes that extend behind occluding features in the imagery so that we had access to multiple labels at each pixel for training. We conjectured that (i) the task of labeling complete objects may be simpler than forcing the classifier to decide between two plausible labels when pixels are partially covered, and (ii) the large receptive fields of each output of the net would allow it to recognize partially occluded objects if we did not force the output labels to be disjoint. Hence, we replaced the single softmax operation of the SEGNET classifier with 11 separate softmax operations (we do assign a label to `background` as it is considered a lack of any other label); treating each feature as a separate classification problem (See Fig. 3).

In principle, each feature could be treated as a separate *binary* labeling problem, however, we use one additional label per feature. We observed that our baseline FAC-SEGNET segmentation had a tendency to produce smooth (or ‘blobby’) outputs, and many facade features are thin, or are separated by thin regions in the image. It is important that the segmentation does not merge nearby objects, and it is also difficult for annotators to precisely mark the boundaries between objects. Furthermore, the *boundaries*, or *edges* of objects seem to take on different characteristics than their interiors;

for example, the connected window regions might be bounded by window frames. We posit that object edges can be treated as a distinct class (based on their appearance) than the interiors of objects, and that treating edges as a separate target for each feature would drive our classifier to make more accurate predictions.

The EDG label was assigned to pixels within 10cm of the edge of the target feature, before the images were scaled down to 512 pixels in height. For certain features we decided that the edges should be handled differently; for the façade itself we wanted to be able to determine the dividing line between adjacent façades so we used one foot for the vertical edges. The tops and bottom of façades were difficult to reliably label, so we did not mark horizontal edges for the facade (a.k.a. wall) element.

The last decoding layer of SEGNET was replaced by 11 different 3×3 convolutional layers corresponding to the 11 CMP labels (excluding background). Each convolutional layer had 4 outputs, indicating whether the output pixel is NEG, UNK, POS or an EDG of the feature. The class-imbalance issue among the four outputs is much more extreme than it was for training FACSEGNET because the NEG label is far more frequent when a single element is considered in isolation. Instead of median-frequency balancing we opted to use assign a loss-weight of 1 to false POS labels, 0.5 to false NEG labels and a loss of 6 to false EDG labels. These numbers reflect our best estimate of how important each type of error is.

We initialized MULTIFACSEGNET classifier using the baseline FACSEGNET weights, and trained using SGD with weight decay and a low learning rate (1e-6) for 300 epochs.

3.4 Network Architecture 2: SEPARABLE

We observe that façade layouts are often approximately arranged on a flexible grid, with some exceptions. Furthermore, we operate on façade images that are approximately rectified using an automatic method [17] so that most façade elements occupy rectangular, nearly horizontally and vertically aligned regions. Therefore, we posit that we should be able to generate façade labels using convolutions by sequence of horizontal and vertical filters, which we expect to encourage synthesis of similarly aligned elements in the output. Furthermore, since these filters require less parameters we can increase the lateral propagation of information between labels in the output image by using much larger horizontal and vertical filters. This could, for example, allow the decoding layer to reconstruct details in the output labels that were occluded in the input.

The MULTIFACSEGNET classifier used 3×3 convolutions during the decoding portion. We aim to replace them by a *horizontal* convolution followed by a *vertical* convolution, however there would be little effect if the filters were on 3 pixels long. Instead we replaced each 3 filter with a 1×9 convolution, followed by a batch-normalization layer and 9×1 filter (See Fig. 4). This increased the number of trainable parameters in the decoding portion of the net, but we believe it also increased the spread of information as the output is hierarchically produced, so that long range relationships between output labels could be encouraged during the decoding process.

The classifier was initialized using the MULTIFACSEGNET weights, with the new convolutional layers initialized using random values from a truncated standard normal distribution. We trained this network using SGD with weight decay and a low learning rate (1e-6) for 300 epochs.

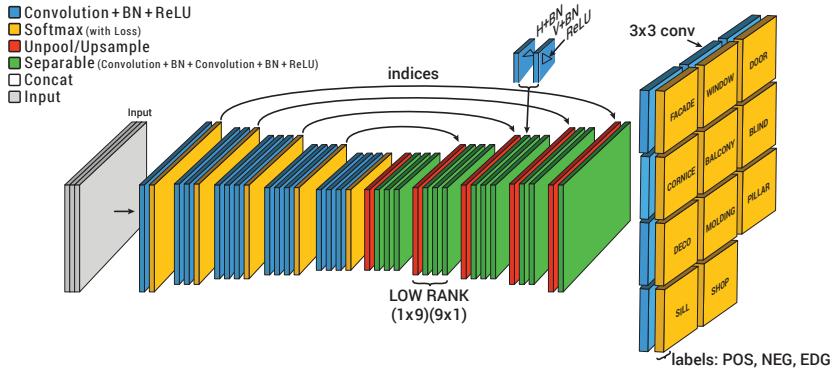


Fig. 4. The SEPARABLE network architecture is formed by modifying the decoding layers of the MULTIFACSEGNET architecture and replacing each of the 3 convolutional layers (blue) with a pair of 1×9 horizontal and 9×1 vertical convolutions, each followed by batch normalization (pairs are indicated in green).

3.5 Network Architecture 3: COMPATIBILITY

The SEPARABLE network for façade segmentation decoupled the labels for the 11 different objects we aimed to identify in façade images. However, for objects that are compatible with each other (e.g., shop and window or door), whereas others are not. In addition, some objects (e.g., sill or cornice) are more likely to occur in the vicinity of other classes of object such as window. Much of this coupling is inherently captured by the large receptive fields and information sharing that happens as SEG-NET encodes and then decodes an image, however we suspect that certain errors that are indicated by incompatible labels being used together could best be identified from the outputs of a segmentation approach. In order to address this possibility we created a recurrent block (see Fig. 5) of output layers that follow the output of SEPARABLE. Each block starts with a concatenation of the softmax outputs of SEPARABLE, followed by 11 different 3×3 convolutional layers corresponding to the 11 output labels, each taking the entire concatenated layer as input. Each convolutional layer is followed by another softmax operation (which adds a non-linearity to the process). The entire block produces output that is the same shape and semantics as its input, so one could repeat the block any number of times in a recurrent fashion.

Specifically, we added two blocks (concatenation, 11 convolutions, and softmax) to the trained SEPARABLE network, essentially unrolling the loop in the network architecture twice for training. Since the recurrent convolutional layers in Fig. 5 occur twice in the unrolled network, we ensured that the weights were shared between convolutional layers each time they were repeated. During training, we added together the weighted cross-entropy losses associated with each softmax unit, so if the compatibility block is repeated twice there are a total of $11 \times 3 = 33$ loss terms associated with the output. In order to prevent random initialization of weights in the new convolutional layers from adding noise to the earlier layers of the net, we first froze all of the weights in SEPARABLE and trained the network for 100K iterations (it converged much more quickly) with

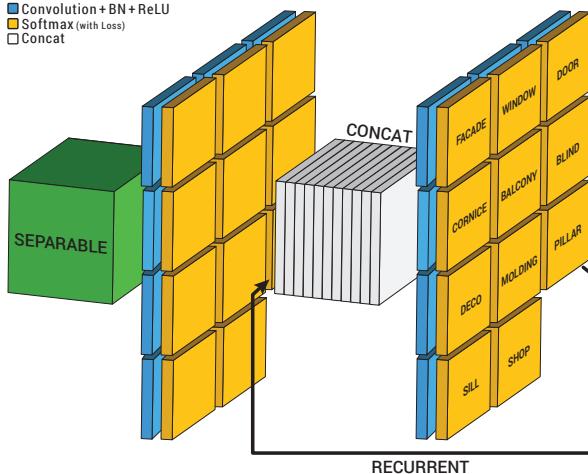


Fig. 5. The changes made to SEPARABLE in order to become the COMPATIBILITY architecture. The output of SEPARABLE is shown on the left, with softmax outputs (yellow) for each of the 11 labels. A *recurrent block* of layers consisting of concatenation (white), 3×3 convolution (blue) and softmax nonlinearities (yellow) is appended to the network. Because the output and input are the same shape, this block can be repeated multiple times (e.g., twice in our experiments).

Table 1. Comparison on 60 images from eTRIMS for window based on our SEPARABLE model.

Approach	Acc	P	R	F_1
Yang and Förstner 2011 [34]	0.75	0.75	0.60	0.67
ATLAS [35]	0.73	-.-	-.-	-.-
Cohen <i>et al.</i> 2014 [36]	0.71	-.-	-.-	-.-
Schmits and Mayer 2016 [20]	0.86	0.67	0.71	0.69
DeepFacade [18]	0.91	-.-	-.-	-.-
Ours	0.971	0.89	0.64	0.74

a learning rate of $1e - 4$. Then we restored the learning rate for the initial SEPARABLE layers of the net and resumed training for another 100K iterations in order to produce a final COMPATIBILITY network. Training this network proved to be difficult; we had to increase the learning rate by a factor of 100 for the recurrent layers and set a negative slope to all (leaky) ReLU activation to prevent the network training from stalling.

4 Results

We evaluate our approaches using a holdout set of 293 images, with 175 holdout images from the CMP [32,33] dataset of around 600 rectified facade images, the remaining 118 where randomly selected GSV images that we had annotated for this project. In order to quantify and compare our results we use the accuracy (*Acc*) as well as the precision (*P*), recall (*R*), and F_1 -measure (F_1). Although accuracy is often reported, we found that F_1 -score seems to correlate with our visual impression of the quality of the result.

We introduced the class EDG into our labeling scheme during training, but during inference and testing we exclude the EDG label by normalizing the POS and NEG probability outputs of our classifier to sum to one. Pixels marked as UNK or EDG in our ground-truth data are ignored; we consider ignoring EDG to be a reasonable decision as annotators are often uncertain about the precise locations of object-boundaries [37], however, ignoring edges seems to have little effect on the numbers. When we compare against other methods in Table 2 we do *not* ignore EDG labels in order to ensure that comparisons are fair to prior art.

For object based scores, we find the bounding boxes of each connected component in our argmax outputs, and in the ground truth. The object score, especially precision, can be heavily influenced by small spurious components for regions of pixels near the classifier’s decision boundary. This could be addressed using Conditional Random Field (CRF) optimization but we instead use a small 3×3 morphological opening operation prior to connected component labeling. We consider two objects to be a potential match if their Intersection over Union (IoU) is more than 0.5, and we find a maximum weighted bipartite matching between objects detected as positive by our system, and objects from the ground truth. We consider matching objects to be true-positives (TP_{ob}) and the unmatched objects are false alarms (FP_{ob}) and misses (FN_{ob}). We also report object-based recall (R_{ob}), precision (P_{ob}) and F_1 -scores ($F_{1\,ob}$).

In our evaluations, we expect annotators to be precise to within a 5-pixel (10cm) boundary around the edges of the labeled regions. Labels within this boundary region were excluded from our evaluation as they are unreliable annotations; this is the same approach taken for example in VOC challenge data [37]. Table 4 shows our results for window on the CMP data, and Table 5 shows our results for window on our street view dataset. We can observe that our new additions to the SEGNET architecture provide significant improvements over a baseline method FACSEGNET. We can observe that both MULTIFACSEGNET, as well as SEPARABLE lead to better results; COMPATIBILITY is sometimes best. In Fig. 6 we show a visual comparison between different variants and in Fig. 7 we show visual results for different labels.

Table 2. Quantitative comparison on ECP data using our SEPARABLE model. We used five-fold cross validation on ECP and we show the mean and variance across folds, the top scores are indicated in bold. Following [18], we show results compared to several approaches; (1) is Yang and Förstner [34], (2.1, 2.2) are two variants of Mathias *et al.*[11], (3.1, 3.2, 3.3) are three variants of Cohen *et al.*[36], and (4) is the best of three variants presented by Liu *et al.*[18].

Class	(1)	(2.1)	(2.2)	(3.1)	(3.2)	(3.3)	(4)	Ours
window	62	76	78	68	87	85	93.04	95.6 ± 0.23
facade(wall)	82	90	89	92	88	90	96.14	91.70 ± 0.39
balcony	58	81	87	82	92	91	95.07	96.0 ± 0.25
door	47	58	71	42	82	79	90.95	98.8 ± 0.09
roof	66	87	79	85	92	91	94.02	97.7 ± 0.10
chimney	-	-	-	54	90	85	91.30	98.9 ± 0.10
sky	95	94	96	93	93	94	97.72	98.4 ± 0.12
shop	88	97	95	94	96	94	95.68	96.9 ± 0.26
total acc.	74.71	88.07	88.02	86.71	89.90	90.34	95.40	96.74

Table 3. Accuracy, Precision, Recall, and F_1 for SEPARABLE on ECP data based on 5-fold cross validation. We suggest that accuracy, which includes the true-negatives, is a poor way to evaluate façade segmentation as it rewards rare objects, for example compare it to the F_1 scores for `door` and `chimney`.

label	Accuracy (A)	Precision (P)	Recall (R)	F_1
window	95.6 ± 0.23	81.5 ± 2.03	79.1 ± 1.58	80.4 ± 0.90
facade (wall)	91.7 ± 0.39	88.6 ± 1.17	93.4 ± 0.70	90.9 ± 0.40
balcony	96.0 ± 0.25	86.8 ± 1.06	83.2 ± 3.11	84.9 ± 1.61
door	98.8 ± 0.09	49.8 ± 4.91	53.5 ± 4.76	49.9 ± 2.68
roof	97.7 ± 0.10	84.6 ± 0.75	78.2 ± 3.10	81.1 ± 1.45
chimney	98.9 ± 0.10	67.6 ± 3.41	61.4 ± 4.43	64.0 ± 2.87
sky	98.4 ± 0.12	83.6 ± 2.61	94.7 ± 0.84	88.8 ± 1.28
shop	96.9 ± 0.26	94.1 ± 2.55	83.2 ± 3.23	88.1 ± 0.97

Table 4. Quantitative results for `window`, based on 175 CMP holdout images (of 606 total).

Approach	Acc	P	R	F_1	P_{ob}	R_{ob}	F_{1ob}
SEGNET	0.93	0.74	0.70	0.72	0.72	0.67	0.69
MULTIFACSEGNET	0.94	0.98	0.57	0.72	0.82	0.66	0.73
SEPARABLE	0.96	0.96	0.71	0.81	0.81	0.71	0.76
COMPATIBILITY	0.95	0.80	0.86	0.83	0.81	0.74	0.77

Table 5. Quantitative results for the `window` class on GSV data with focus on the F_1 -scores as a predictor of performance; each modification has led to a substantial increase in F_1 .

Variant	Acc	P	R	F_1	P_{ob}	R_{ob}	F_{1ob}
SEGNET	0.93	0.55	0.62	0.58	0.76	0.57	0.65
MULTIFACSEGNET	0.96	0.92	0.56	0.70	0.88	0.61	0.72
SEPARABLE	0.97	0.81	0.70	0.75	0.86	0.70	0.77
COMPATIBILITY	0.95	0.85	0.79	0.82	0.81	0.72	0.76

Table 6. Quantitative results for all labels on CMP + GSV (combined) data using SEPARABLE.

Target	Acc	P	R	F_1	P_{ob}	R_{ob}	F_{1ob}
balcony	0.97	0.79	0.51	0.62	0.34	0.45	0.39
blind	0.98	0.63	0.22	0.33	0.35	0.21	0.26
cornice	0.98	0.73	0.55	0.63	0.43	0.33	0.38
deco	0.98	0.43	0.15	0.23	0.30	0.08	0.12
door	0.99	0.39	0.49	0.43	0.15	0.41	0.22
molding	0.94	0.90	0.53	0.67	0.21	0.42	0.28
pillar	0.99	0.75	0.00	0.01	0.60	0.00	0.01
shop	0.97	0.46	0.69	0.55	0.11	0.15	0.13
sill	0.98	0.72	0.21	0.32	0.22	0.11	0.15
window	0.96	0.93	0.74	0.82	0.79	0.75	0.77



Fig. 6. Examples of estimated probabilities assigned to the `window` class for a variety of images in our evaluation sets.

Comparison on eTRIMS. In addition, we compare against a number of other façade segmentation approaches using the eTRIMS data set. Table 1 shows results on eTRIMS data, however their labels are not in perfect semantic agreement with the labels used to train our network; in particular partially occluded windows are considered negative examples, and our `shop` label is considered to be `window` in eTRIMS. Nevertheless our accuracy is higher than other reported accuracies, as is our F_1 -score on this dataset. Unlike approaches that used cross-validation on eTRIMS to generate their results, our results are based on a network that was trained of none of the eTRIMS data which makes our result significant; although our evaluation is limited to only windows. The main point of this evaluation is to show that our results are still very good and that even our baseline FACSEGNET algorithm is already better than other state of the art approaches for facade segmentation. In particular, baseline FACSEGNET outperforms other deep learning based approaches.

Comparison against ECP. A number of authors report accuracy on the ECP dataset, so we refine our network to do 5-fold cross validation on ECP imagery. We added new outputs (increasing the number from 11 to 15) for ECP elements not in our orig-



Fig. 7. Example estimated probabilities for different façade elements using SEPARABLE network.

inal training set. ECP data did not include labels for overlapping objects, so in our multi-label representation of ground-truth we marked occluded regions as UNK during training. For evaluation, we composite each of 8 ECP labels to create an image that matches the ECP ground-truth format. Quantitative results are presented in Table 2 and compared with a variety of approaches; for most features (with the exception of `facade`) our SEPARABLE approach outperforms other methods according to accuracy. We also report precision, recall, and F_1 -score per-label in Table 3.

Comparison between approaches. Our aim is to find an approach with a high F_1 -score in relatively unconstrained images (e.g. GSV imagery), and in particular with high per-object recall R_{ob} and F_{1ob} because we are inspired by inverse procedural modeling. In Table 4 we demonstrate that each approach significantly increased the F_1 and F_{1ob} on our most important class of object (`window`), and similarly when we limit the results to only GSV imagery in Table 5 with the exception that the COMPATIBILITY variant hurt our object-based scores. In order to understand how these results break down between labels, we report per-element pixel and object-based metrics for all hold-outs in Table 6 using the SEPARABLE approach. On a per-pixel basis we achieve satisfying results, however our simplistic approach identifying objects based on those pixels only leads to satisfying results on windows.

5 Conclusion

We presented a deep-learning based facade segmentation approach that works for a variety of data sets from pre-rectified and cropped facade images to automatically captured panoramic street view images. Starting from the established SEGNET architecture, our main ideas are to add separate edge labels, overlapping labels, separable filters, and an iterative optimization for label smoothing. We also provide a larger dataset consisting of labeled street level photospheres. Our results demonstrate a significant improvement over the state of the art. In future work, we would like to extend the work to improve object detection especially for thin objects, to investigate an end-to-end network trained for rectification and segmentation using transformer networks as well as employing separable filters for region completion (e.g., due to occlusion from trees).

References

1. Müller, P., Zeng, G., Wonka, P., Van Gool, L.: Image-based procedural modeling of facades. ACM SIGGRAPH **26**(3) (2007) 85
2. Badrinarayanan, V., Kendall, A., Cipolla, R.: Segnet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence (2017)
3. Badrinarayanan, V., Handa, A., Cipolla, R.: Segnet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling. arXiv preprint arXiv:1505.07293 (2015)
4. Kendall, A., Badrinarayanan, V., , Cipolla, R.: Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. arXiv preprint arXiv:1511.02680 (2015)
5. Riemenschneider, H., Krispel, U., Thaller, W., Donoser, M., Havemann, S., Fellner, D., Bischof, H.: Irregular lattices for complex shape grammar facade parsing. IEEE CVPR (2012) 1640–1647
6. Teboul, O., Kokkinos, I., Simon, L., Koutsourakis, P., Paragios, N.: Parsing facades with shape grammars and reinforcement learning. IEEE TPAMI **35**(7) (2013) 1744–1756
7. Shen, C.H., Huang, S.S., Fu, H., Hu, S.M.: Adaptive partitioning of urban facades. ACM SIGGRAPH Asia **30**(6) (2011) 184
8. Yang, C., Han, T., Quan, L., Tai, C.L.: Parsing façade with rank-one approximation. IEEE CVPR (2012) 1720–1727
9. Martinović, A., Mathias, M., Weissenberg, J., Van Gool, L.: A three-layered approach to facade parsing. ECCV (2012) 416–429
10. Kozinski, M., Gadde, R., Zagoruyko, S., Obozinski, G., Marlet, R.: A mrf shape prior for facade parsing with occlusions. IEEE CVPR (2015) 2820–2828
11. Mathias, M., Martinović, A., Van Gool, L.: Atlas: A three-layered approach to facade parsing. International Journal of Computer Vision **118**(1) (2016) 22–48
12. Nan, L., Jiang, C., Ghanem, B., Wonka, P.: Template assembly for detailed urban reconstruction. CGF Eurographics **34**(2) (2015) 217–228
13. Ceylan, D., Dang, M., J. Mitra, N., Neubert, B., Pauly, M.: Discovering structured variations via template matching. CGF (01 2016)
14. Cohen, A., Schwing, A.G., Pollefeys, M.: Efficient structured parsing of facades using dynamic programming. IEEE CVPR (2014) 3206–3213
15. Dai, D., Prasad, M., Schmitt, G., Van Gool, L.: Learning domain knowledge for facade labelling. ECCV (2012) 710–723
16. Jiang, H., Nan, L., Yan, D.M., Dong, W., Zhang, X., Wonka, P.: Automatic constraint detection for 2d layout regularization. IEEE TVCG **22**(8) (2016) 1933–1944
17. Affara, L., Nan, L., Ghanem, B., Wonka, P.: Large scale asset extraction for urban images. European Conference on Computer (2016) 437–452
18. Hantang Liu, Jialiang Zhang, J.Z.S.C.H.H.: Deepfacade: A deep learning approach to facade parsing. In: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17. (2017) 2301–2307
19. Thoma, M.: A survey of semantic segmentation. CoRR **abs/1602.06541** (2016)
20. Schmitz, M., Mayer, H.: A Convolutional Network for Semantic Facade Segmentation and Interpretation. ISPRS-International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences (2016) 709–715
21. Kelly, T., Femiani, J., Wonka, P., Mitra, N.J.: Bigsur: Large-scale structured urban reconstruction. ACM Transactions on Graphics **36**(6) (November 2017)

22. Teboul, O., Kokkinos, I., Simon, L., Koutsourakis, P., Paragios, N.: Shape grammar parsing via reinforcement learning. In: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, ieeexplore.ieee.org (June 2011) 2273–2280
23. Korč, F., Förstner, W.: eTRIMS Image Database for interpreting images of man-made scenes. Technical Report TR-IGG-P-2009-01, Dept. of Photogrammetry, University of Bonn (April 2009)
24. Radim Tyleček, R.Š.: Spatial pattern templates for recognition of objects with regular structure. In: Proc. GCPR, Saarbrücken, Germany (2013)
25. Badrinarayanan, V., Kendall, A., Cipolla, R.: Segnet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence (2017)
26. Shelhamer, E., Long, J., Darrell, T.: Fully convolutional networks for semantic segmentation. IEEE transactions on pattern analysis and machine intelligence **39**(4) (April 2017) 640–651
27. Yu, F., Koltun, V.: Multi-Scale Context Aggregation by Dilated Convolutions. ArXiv e-prints (November 2015)
28. Yu, F., Koltun, V., Funkhouser, T.: Dilated residual networks. In: Computer Vision and Pattern Recognition. Volume 1. (2017)
29. Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional networks for biomedical image segmentation. ArXiv e-prints (May 2015)
30. Simonyan, K., Zisserman, A.: Very Deep Convolutional Networks for Large-Scale Image Recognition. ArXiv e-prints (September 2014)
31. Brostow, G.J., Fauqueur, J., Cipolla, R.: Semantic object classes in video: A high-definition ground truth database. Pattern Recognition Letters **xx**(x) (2008) xx–xx
32. Tylecek, R.: The cmp facade database. Technical report, Tech. rep., CTU–CMP–2012–24, Czech Technical University (2012)
33. Tyleček, R., Šára, R.: Spatial pattern templates for recognition of objects with regular structure. German Conference on Pattern Recognition (2013) 364–374
34. Yang, M., Förstner Photogrammetric image analysis, W., 2011: Regionwise classification of building facade images. Springer (2011)
35. Mathias, M., Martinović, A., Van Gool, L.: ATLAS: A Three-Layered approach to facade parsing. Int. J. Comput. Vis. **118**(1) (May 2016) 22–48
36. Cohen, A., Schwing, A.G., Pollefeys, M.: Efficient structured parsing of facades using dynamic programming. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition, ieeexplore.ieee.org (June 2014) 3206–3213
37. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The pascal visual object classes (VOC) challenge. International journal of computer vision **88**(2) (June 2010) 303–338