Detecting window-to-wall ratio for urban-scale building simulations using deep learning with street view imagery and an automatic classification algorithm

(Article begins on next page)

Detecting window-to-wall ratio for urban-scale building simulations using deep learning with street view imagery and an automatic classification algorithm.

Anthony Robert Suppa[1,*], Alessandro Aliberti[1], Marta Carla Bottero[1], Vincenzo Corrado[2]

1. Interuniversity Department of Regional and Urban Studies and Planning (DIST), Politecnico di Torino, Viale Mattioli 39, 10125 Turin, Italy

2. Department of Energy, Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Turin, Italy

** anthony.suppa@polito.it*

Abstract

Machine learning techniques can fill data gaps for urban-scale building simulations, particularly gaps around window-to-wall ratio (WWR). This study presents a comprehensive workflow to (1) automatically extract and stitch images from Google Street View (GSV); (2) label images with a custom Rhino-based tool to aid annotation of occluded glazing; (3) detect wall, garage, and glazing objects by training and validating a YOLOv9 deep learning model with three added post-scripts; (4) calculate WWR at façade, building, and district scales; and (5) simulate district energy consumption in an urban building energy model (UBEM). Results include a 96% image-capture rate from GSV, indicating a robust extraction and stitching algorithm. Converting model detections into WWR, 94% and 100% of façades have detected WWRs within ±5% and ±10% of ground truth WWRs, respectively. A novel automatic algorithm upscales façade detections to estimate WWR at non-street-facing sides and rears, resulting in distinct WWRs for each face of each building. For a case study in Turin, Italy, WWR detections are +5.2% and +6.9% greater when upscaling based on OpenStreetMap and municipal GIS data, respectively, compared to TABULA, leading to 1.5% and 35.5% increases in heating and cooling energy need in the UBEM. The workflow is made openly available to support future research in other contexts.

Keywords

Building- and district-scale WWR; machine learning; street view imagery; automatic image extraction and stitching; building façade classification; urban building energy modeling (UBEM).

1 Introduction

Energy performance of buildings is linked to glazing, including window-to-wall ratio (WWR) and glazing performance, as shown worldwide in building energy simulation studies (Cetiner & Ceylan 2013; Samuelson et al. 2016; Liu et al. 2017; Veillette et al. 2021; Ohene et al. 2024). The location of glazing also matters, with east and west orientations having the greatest impact on energy consumption (Feng et al. 2017; Murano et al. 2018). In addition to energy performance, WWR and glazing orientation affect daylighting, thermal comfort, and view quality (Li et al. 2024). Given that WWR in general, and per-orientation WWR in particular, are such significant factors, modeled WWR must accurately reflect real-world conditions, especially when hundreds or thousands of buildings are simulated using urban building energy modeling (UBEM).

UBEM calculates building energy consumption for clusters of buildings at district or city scale, accounting for dynamics of individual buildings, inter-building effects, and microclimate (Reinhart & Davila 2016; Hong et al. 2020). Literature notes that missing data in UBEM studies are significant barriers to model effectiveness and add uncertainty to building energy estimates. (Ali et al. 2021; Ferrando et al. 2020). WWR is one missing data point where UBEM could take advantage of advanced urban sensing technologies and machine learning to fill data gaps (Chen et al. 2019). A recent review reported that incorporating machine learning methods into UBEM was among the most important issues for future research (Dahlström et al. 2022).

Current practices to quantify WWR in UBEM range widely. From coarsest to finest level, UBEM can use (i) a single WWR for an entire district; (ii) a different WWR

Abbreviations

AB	Apartment block	MF	Multi-family house
AP	Average precision	NMS	Non-maximum suppression
API	Application programming interface	OSM	OpenStreetMap
CNN	Convolutional neural network	PD	Public dataset
FoV	Field of view	SF	Single-family house
FP	False positive	SVI	Street view imagery
FN	False negative	TH	Terraced house
GIS	Geographic information systems	TP	True positive
GSV	Google Street View	UBEM	Urban building energy model / modeling
IoU	Intersection over union	WWR	Window-to-wall ratio
mAP	Mean average precision	ΔWWR	Difference in detected and ground truth WWR

for each building archetype; (iii) an average building-level WWR, different for each building; or (iv) a different WWR for each orientation of each building simulated (De Jaeger et al. 2018). Among UBEM studies modeling retrofits, 60% of reviewed studies simulated window replacement (Suppa & Ballarini 2023), though only one used WWRs detected using façade parsing techniques (Szczesniak et al. 2022). Thus, many UBEM studies rely on building archetypes using fixed WWR values (Dahlström et al. 2022). Such reliance is partly due to the vast effort required to determine WWR at scale, since traditional WWR calculation relies on manual measurement of construction drawings. In a district of Ningbo, China, Ma et al. (2022a) sampled 367 residential buildings, collecting blueprints over six months, and calculating WWRs over an unspecified amount of time.

Façade parsing, a field within computer vision that categorizes façade images into elements of "window", "balcony", "wall", and others, has been used to rapidly estimate WWR. These approaches have also been combined with street view imagery (SVI) and deep learning methods using convolutional neural networks. SVI is provided through platforms such as Google Street View (GSV) (Google 2024a) and Mapillary (Meta 2024), and has the advantage of being plentiful and widely geographically distributed. GSV, the most common SVI source in research, operates in 90 countries worldwide (Biljecki & Ito 2021).

In this work, we present a comprehensive workflow to fill data gaps for UBEM using deep-learning-based detection of WWR in GSV images. A novel algorithm estimates WWRs for each building face, including sides and rears not visible in SVI. The workflow is applied to a case district of 1,597 residential buildings in Turin, Italy, first employing a script to automatically determine which GIS line segments are street-facing. Second, 2,393 façade images are extracted and stitched from 7,906 constituent GSV image tiles. Third, 576 façades are annotated using a custom tool in Rhino/Grasshopper software. Fourth, annotated images are used to train and validate a deep learning model, thus detecting walls, garage doors, and glazing objects, which are converted to WWR. Fifth, the developed algorithm classifies building faces and estimates WWR at non-visible sides and rears based on urban planning regulations and assumptions, which are verified using manual measurement of Google Earth images. Finally, UBEM is developed to compare building energy consumption when modeling detected and standard archetype WWRs.

1.1 Literature review

Many façade parsing studies rely on public datasets (PDs), which contain images and annotation files with corresponding labels and are openly available for research. PDs include "CMP", containing 606 rectified images from cities around the world and diverse architectural styles (Tyleček 2013); "ECP", with 104 highly-regular Haussmannian-style buildings in Paris (Teboul et al. 2011); "Graz", containing 50 images of diverse architectural styles from Germany & Austria (Riemenschneider et al. 2012); and "ENPC", with 79 images of Art-Deco buildings in Paris (Gadde et al. 2016).

The architectural styles are relevant since early façade parsing works use style-appropriate shape grammars. Shape grammars use a set of matching and replacement rules and treat façade parsing as an optimization problem, often employing Markov Chain Monte Carlo and related methods (Riemenschneider et al. 2012; Gadde et al. 2016). Among grammar-based parsing studies, Mathias et al. (2016) notably developed "weak architectural principles" to guide their algorithm, including vertical and horizontal alignment, similarity of windows on the same façades, façade symmetry, ground-level door touching the ground, vertical region order (ground, shop, wall, sky), and co-occurring elements.

Besides grammar-based methods, deep-learning-based studies use convolutional neural networks (CNNs) for façade parsing. CNNs, a neural network specializing in pattern recognition and feature extraction from images, train on hundreds or thousands of annotated images before learning to predict objects such as windows or walls. Li et al. (2020) used 3,400 PD and internet images in an object-detection CNN called "Heatmap Fusion", first locating keypoints of windows, then learning keypoint relationships and grouping these into windows, allowing non-rectangular detections in non-rectified façades such as in the PD "eTRIMS" (Korč & Förstner 2009). In "DeepFacade", Liu et al. (2020) detected bounding boxes directly with the "Mask RCNN" algorithm (He et al. 2017), building a semantic segmentation network on top of this, and introduced a symmetric loss function to output symmetric window shapes. Ma et al. (2022b) developed "Progressive Feature Learning", a segmentation architecture effective with occlusions, including street trees existing in the ENPC dataset and random objects superimposed on ECP images.

The above PD-based studies do not report WWR, though one could extrapolate WWR from semantic segmentation studies by dividing total pixels in the window class by the sum of those in the window and wall classes. Still, an examination of the PD labels shows that these are not consistent with the notion of WWR, and thus would require re-annotation to make an equal comparison. Examples are depicted in Figure 1, where balcony guard annotations block the lower extents of glazing. While this is accurate from the perspective of visible pixels, a WWR calculation would account for glazed balcony doors extending to the top of balcony slabs. Furthermore, ground-floor glazing objects classified as "shop" should contribute to WWR.

Beyond PD-based parsing research, studies most related to the present create SVI-based datasets and/or analyze WWR, as summarized in Table 1. Whereas PDs contain clear, focused images, SVI is captured from a street-level vehicle without attention to perspective, faces occlusions from vehicles and other objects, and may include blooming and stitching artifacts (Femiani et al. 2018). Steep pitches are required to capture the height of tall buildings from below, resulting in protruding balcony slabs that occlude upper-level glazing. Few studies comment on balcony occlusion, though Tarkhan et al. (2022) note successful detection results despite balcony occlusion, and Femiani et al. (2018) annotate images with multiple labels so that windows can simultaneously exist behind balcony guards. Lu et al. (2023) use a squeeze-and-excitation module to improve their model's recognition of partially occluded glazing, though they do not use SVI and do not indicate whether balcony occlusion was significant.

Some of the precedents used manual collection of GSV images (Kong & Fan 2021; Nordmark & Ayenew 2021; Sezen et al. 2022), a labor-intensive process difficult to upscale. Others developed API-based workflows to automatically extract hundreds or thousands of buildings with single GSV images (Szczesniak et al. 2022; Tarkhan et al. 2022; Tarkhan et al. 2024) or multiple/stitched GSV images (Femiani et al. 2018). Automatic extraction methods were based on central coordinates of the four longest building edge segments (Szczesniak et al. 2022), or on coordinates of street-facing façade segments obtained through a concave hull algorithm (Tarkhan et al. 2022; Tarkhan et al. 2024).

1.2 Literature gap and research problem

There are gaps in the literature regarding image extraction, image annotation, WWR at non-visible sides, and incorporating WWR into urban energy models.

Figure 1. Images and ground truth annotations from ECP at left (Teboul et al. 2011) and ENPC at right (Gadde et al. 2016), with "window" class in red, "balcony" in purple, and "shop" in green.

Table 1. Façade parsing studies using SVI and/or calculating WWR. Studies use convolutional neural networks (CNNs) except where grammar-based (GB) methods are noted. PD = public dataset. Evaluation metrics: P = precision; R = recall; F1 = F1 score; Acc = pixel accuracy; IoU = intersection over union; mAP = mean Average Precision; ΔWWR = difference in WWR between ground truth and detection.

Reference	**Model**	**Dataset used**	**If SVI: image source**	**If SVI: extraction method**	**If SVI: single or multiple/ stitched images per building**	**No. images annotated and/or manually measured in study**	**Annotation tool used**	**Image location**	WWR **calculated**	**Key evaluation metrics used**
Xu et al. 2025	U-Net (Ronnenberger et al. 2015)	SVI	Baidu Street VIew	Automatic	Single	1,000	-	Wuhan, China	Façade scale	P, R, Acc, mean IoU
Lu et al. 2023	Improved SOLOv2 (Wang et al. 2020b)	Images from Internet and by researchers	-	-	-	1,169	LabelMe	-	Façade scale, stratified by story	ΔWWR, mAP at IoU 0.5, 0.75, and 0.5-0.95
Szczesniak et al. 2022	Sobel filters (GB)	SVI	GSV	Automatic	Single	1,057	LabelMe	New York, USA Chicago, USA	Façade scale	ΔWWR
Tarkhan et al. 2022	Heatmap Fusion (Li et al. 2020)	SVI & PD	GSV	Automatic	Single	537	LabelMe	New York, USA Lisbon, Portugal	Façade scale	ΔWWR
Tarkhan et al. 2024	Sobel filters (GB) & Heatmap Fusion (Li et al. 2020)	SVI & PD	GSV	Automatic	Single	537	LabelMe	New York, USA Lisbon, Portugal	Façade scale	ΔWWR
Sezen et al. 2022	YOLO v3, v4, and v5 (Redmon & Farhadi 2018; Bochkovskiy et al. 2020; Ultralytics 2020)	SVI	GSV	Manual	Single	1000	Not stated	Turkey	-	P, R, mAP at IoU 0.5
Femiani et al. 2018	Based on Segnet (Badrinarayanan et al. 2017)	SVI & PD	GSV	Automatic	Multiple/ stitched	118	Not stated	Auckland, Brisbane, Cape Town, Hong Kong plus 15 North American & European cities	-	P, R, F1, Acc
Wang et al. 2022b	Building Façade Parsing R-CNN	SVI & PD	Oxford Robot Car dataset	(Extraction not part of study)	-	500	LabelMe	Oxford, UK	-	Acc, mean IoU
Kong & Fan 2021	Semantic segmentation network built on YOLOv3 (Redmon & Farhadi 2018)	SVI & PD	Mapillary	Manual	Single	900	VIA	Wuhan, China Paris, France Trondheim, Norway	-	P,R, mAP at IoU 0.5; Acc
Nordmark & Aneyew 2021	Mask R-CNN (He et al. 2017)	SVI & PD	GSV	Manual	Single	100	VIA	Gothenburg, Sweden	-	Acc

First, literature does not adequately detail how to extract and stitch multiple GSV images of each façade, which can capture wider/taller extents than single images. It appears the only study stitching multiple GSV images for façade parsing (Femiani et al. 2018) did not make code available, which could support the development of local datasets for future research.

Second, literature gaps exist around annotating occluded glazing, especially at balconies, a common source of occlusion in SVI since images are captured from below. Studies used the LabelMe (Russell et al. 2008) or VIA (Dutta & Zisserman 2019) annotation tools, which are freely available and easy to use but cannot create gridlines, which can help locate occluded objects and facilitate precise pasting of repeated objects. Precedent studies have also not evaluated annotation tools against automated AI-assisted annotation services such as Roboflow (2024), which uses a pre-trained deep learning model to detect window and wall elements.

Third, literature does not sufficiently deal with building sides and rears that are not visible in SVI. Among precedents, only Szczesniak et al. (2022) and Xu et al. (2025) comment on non-visible sides and rears, to which both studies assign the same value as street-facing fronts when calculating WWR. Only Lu et al. (2023) stratified façade detections into ground-floor and upper-floor results, but to our knowledge, their code was not made publicly available and was not used to estimate WWR at non-visible sides. Shops at ground level may contain extensive storefront glazing, thus separate upper-level WWR detections could provide a better estimate for WWR at non-visible building rears.

Finally, there are gaps in how detected WWRs are incorporated into UBEM, and what impact detected versus archetype-based WWRs have on energy use. Among precedents, only Szczesniak et al. (2022) applied detected WWR values to UBEM, using the application UMI. Thus, only one precedent study compared energy use impacts when using detected versus default WWRs (in this case, using constant WWRs of 40%), and no study compared detected WWRs to archetype-based WWRs such as those in TABULA, a widely-used European archetype project (Loga et al. 2016).

1.3 Research objectives

The objectives of the research are as follows:

1. To automatically extract and stitch multiple GSV images for each building in a district.
2. To improve image annotation tools to better locate glazing that is frequently occluded by balcony slabs, as GSV images are taken from below at steep angles.
3. To predict WWR in façade images and estimate WWR at each side of each building, including non-visible sides and rears.
4. To conduct UBEM using detected WWRs, and compare energy consumption against models using standard archetype WWRs.
5. To use window and wall detections to fill gaps in OpenStreetMap (OSM) data with building height and story count, enabling UBEM using this open data source.

1.4 Research contributions and novelty

The following are developed and made available with this research[1]:

1. Python code to classify street-facing segments in GIS buildings. The novel method based on angle and distance to OSM road segments quickly prepares datasets for SVI extraction.
2. Python code to extract two to six GSV images per façade and stitch these into rectified images, permitting automatic capture of wider/taller façades compared to single images.
3. Annotation tool comprised of Grasshopper script and Rhino template. The tool can create gridlines to locate occluded glazing and precisely copy/paste repeated objects using object snapping, features not present in existing tools. Additionally, we compare the tool to AI-assisted annotation services.
4. Python post-script to split deep learning detections into ground and above-ground levels. A novel application, we use upper-level front WWRs to estimate WWR at building rears, as ground levels often contain shops with extensive glazing.
5. Python code for our automatic classification algorithm. The algorithm classifies building line segments as non-street-facing sides and rears, then assigns WWRs to each segment based on urban planning regulations and assumptions. Such assumptions, which we verify with Google Earth imagery, are particular to the case study but can be modified to suit other contexts. The novelty is to determine distinct WWRs for both visible and non-visible sides, rather than using a constant detected WWR for all sides, as in precedents.
6. Python post-script to estimate building height and floor count in detected images, thus filling a data gap in the OSM dataset and enabling UBEM using open data.
7. Python code to convert detections into per-orientation (north, east, south, west) WWRs, a format required for UBEM applications including CitySim (Robinson et al. 2009), UMI (Reinhart et al. 2013), UBEM.io (Ang et al. 2022), and URBANopt (NREL 2024).
8. Grasshopper script with Dragonfly/URBANopt UBEM workflow to model detected WWRs. Another novelty, we compare detected and archetype-based WWRs from TABULA.

We also train and validate a YOLOv9 model to predict walls, garages, and glazing objects; this is an existing deep learning architecture and is not presented as an innovation of the work. The selection of YOLOv9 was based on successful precedents using earlier YOLO versions, and other deep learning architectures could be substituted into the overall WWR detection workflow. However, we demonstrate the trained model is effective for the present use, with proficiency in a district in Turin, Italy, and on the Graz, CMP, and EPC datasets, representing diverse regions.

1.5 Article structure

The article is structured as follows. Section 2 details the methodology including SVI extraction; the deep learning stage; WWR calculation at façade, building, and district scales; and application of detected WWRs to UBEM. Section 3 describes the case study, a district in Turin, Italy, with 1,597 buildings. Section 4 presents results of each of the six methodological sub-sections. Section 5 compares the trained YOLOv9 model on datasets of other styles and contexts, compares our annotation workflow with AI-assisted annotation services, discusses the results of the WWR upscaling and UBEM, and discusses the validity and limitations of the research. Section 6 provides a conclusion.

2 Methodology

The methodology includes six steps: (1) GIS data preparation; (2) SVI extraction and stitching; (3) image annotation using a custom Rhino-based tool; (4) deep learning to predict

[1] All code is available at: https://github.com/asuptor/WWRdetection

WWR in façade images; (5) upscaling WWR to building and district scales; and (6) urban building energy modeling. An overview of the methodology is depicted in Figure 2, and each of the steps is detailed in turn.

2.1 GIS data preparation

OSM (OSM Contributors 2024) provides GIS data for the first four steps of the methodology, including image extraction, annotation, and deep learning stages, and both OSM and the municipal GIS database (City of Turin 2023) are used for the last two steps. The OSM database contains basic building footprints, whereas the municipal database is more detailed, containing building volumes within footprints and heights obtained using LiDAR.

In ArcGIS Pro (Esri 2024), the OSM building polygons are split into line segments, and the "calculate geometry attributes" tool provides latitude and longitude for end and center points of each line, plus geodesic bearing and line length. Additionally, the "near" tool is used to calculate the angle ("near angle") and distance from each OSM road segment plus a unique identifier for each road segment.

Next, the data are imported into a Python script to automatically identify street-facing segments. A line parallel to a road segment would have a near angle of 90° or 270°, so the script first filters building line segments to these values, plus/minus a threshold (default of ±20°). Then distances from each filtered building line to each near road segment are calculated, and the line segment with the minimum distance is set as a street-facing façade. The code permits a multiplier on the minimum distance (e.g. between 1.1 and 1.5), in case multiple street-facing lines exist along the same near-road segment, as depicted in Figure 3. Finally, the lines are exported

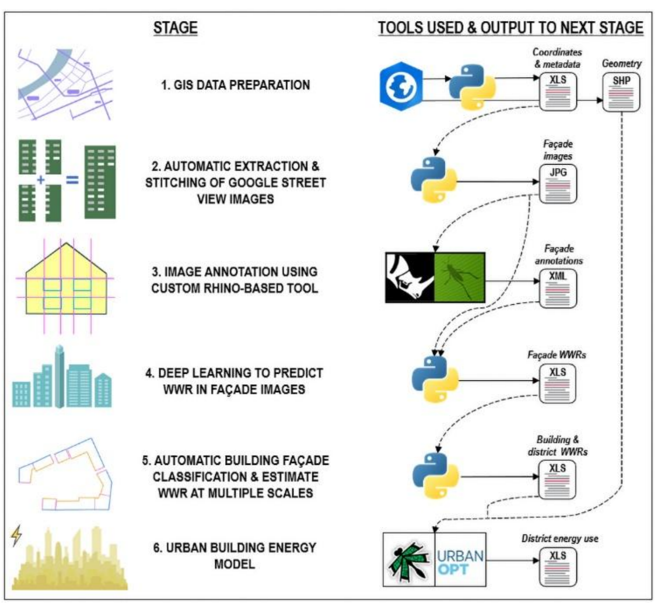


Figure 2. Methodology overview.

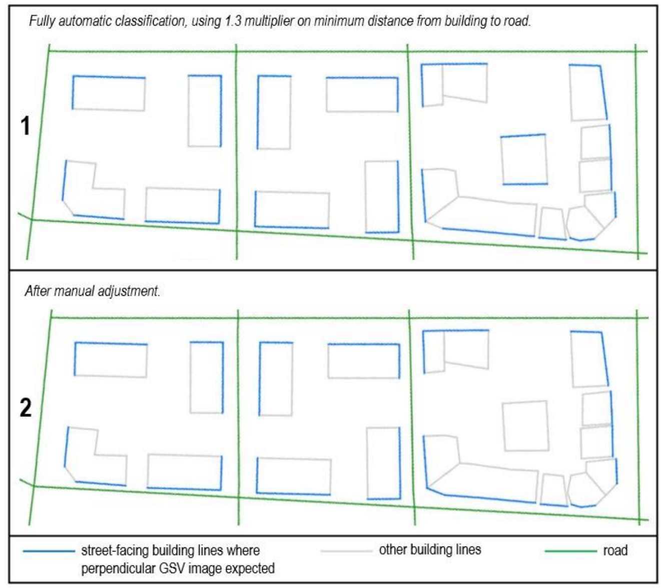


Figure 3. Visualization of code output identifying street-facing building façades where perpendicular GSV images are expected. The angled building fronts at bottom right and left of each image will be classified as additional street-facing fronts in the subsequent classification algorithm.

back to GIS and reviewed using multiple colors, allowing for quick manual adjustments to ensure the layer contains only street-facing segments where perpendicular GSV images are expected. Finally, the coordinates and bearings are converted to Excel format for the subsequent image extraction step.

2.2 SVI extraction and stitching

2.2.1 Automatic algorithm

A custom Python script uses GIS coordinates for an initial call to the GSV API with the library "Google Streetview" (Wen 2019). If a GSV image exists within the specified radius of a building, an image and metadata file containing coordinates of the nearest GSV camera are returned. API calls use the following parameters: radius of 20 meters maximum from camera to façade; field of view (FoV) of 120º, and pitches of 0º and 20º. Headings for image capture are calculated for each building using camera coordinates from the first API call, where the script determines whether target buildings are within the FoV or whether camera rotations are needed to capture the building extent, as depicted by Figure 4.

Next, with up to three headings per building at pitches of 0º and 20º, a second API request captures two, four, or six images, always starting with the perpendicular image at 0º pitch, which is an orthogonal view of the façade. The inclined pitch of 20º captures the upper extent of buildings.

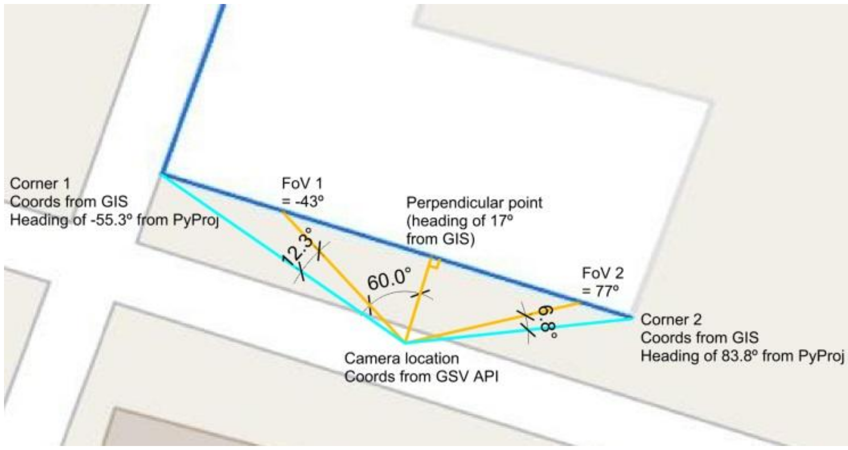


Figure 4. Example façade capture, with rotations left and right calculated using the PyProj library (Whitaker 2023). Basemap source: OSM Contributors (2024).

Third, automatic image stitching and cropping are completed in multiple rounds, as Figure 5 depicts[2]. Starting with an orthogonal source image, keypoints are computed and matched with target images using the SIFT algorithm (Lowe 2004), and functions in the OpenCV library (Bradski 2000) calculate homographies and warp targets to match orthogonal source images. Areas of missing data appear in black, which the code automatically crops based on summed values of pixel rows/columns.

2.2.2 Evaluation and cropping of extracted images

Upon extraction, we randomly sample images for the training and validation ("Train/Val") dataset and inspect these according to quantitative and qualitative criteria. Quantitatively, we compare the alignment of second-level glazing objects with an orthogonal line, as depicted in Figure 7b, rejecting images exceeding ±2.5º. Qualitatively, images are rejected based on criteria in Appendix A1, drawing on quality criteria from Hou and Biljecki (2022) for excessive blurriness, noise, obstruction, and bad capture settings.

After evaluation, images are manually cropped within one meter at building sides. Long buildings with excessive noise or blurriness are cropped at sides to yield a center building view. The image bottom is cropped near the sidewalk curb, as visible driveways could help the deep learning algorithm identify garage doors. The image top is cropped within two meters of the roof. The tops are cropped to exclude step-backs at the uppermost floors, if present, since glazing here cannot be seen from the street.

Beyond the Train/Val set, all other images extracted (the "Inference" dataset) follow the qualitative evaluation and cropping procedures before inference with the deep learning model.

2.3 Image annotation stage

Next, we develop an annotation tool in Rhino3D (McNeel 2024), a software common in architecture and engineering and where UBEM plug-ins UMI and URBANopt operate. Rhino's companion software, Grasshopper, is a graphical algorithm editor permitting automatic creation of annotation XML files.

Since images are captured from below, they may be occluded by balcony slabs and guards, and upper regions of taller buildings contain noise or graininess, as depicted in Figure 6. Thus, we begin annotations by creating orthogonal gridlines to align elements and locate hidden edges, according to these ***annotation architectural principles***:

- Similar-appearing glazing objects are aligned vertically and horizontally.
- Similar-appearing glazing objects repeating vertically/horizontally have identical sizes.
- If balcony doors at the second and/or third levels are visible, and if a matching pattern of balcony slabs continues upward in vertical stacks, there must be balcony doors present, even if occluded.
- Balcony doors extend down to the top of balcony slabs, taken where slabs connect to the building (not where protruding into the foreground).
- Ground-level doors, storefront glazing, and garage doors extend to ground level.

The principles are not coded into the deep learning model but guide the annotation process, and are inspired by the "weak architectural principles" of Mathias et al. (2016). As depicted in Figure 7, implementing the principles begins with gridlines tight to second-level glazing objects, which have the least distortion and occlusion.

[2] Due to GSV copyright, street view images printed in this work are facsimiles of GSV images, using photos taken by the authors.

Figure 5. Automatic image extraction, cropping, and stitching. Top row: original images 0, 1, 2, and 3. Middle row: first round of stitching/cropping. Bottom row: second (final) round of stitching/cropping.

Figure 6. Occlusion from balcony slabs and noise in upper regions of images.

Figure 7. Annotation workflow: (a) image insertion; (b) orthogonality tests; (c) gridlines tight to glazing edges; (d) outlining second-level glazing along gridlines; (e) copying/pasting glazing elements to upper levels; (f) outlining objects for ground-level glazing, garage, and wall (g) the final annotation is checked before XML conversion.

Gridlines also set identical widths/heights of repeating windows and balcony doors, allowing precise pasting of repeated objects using object snapping in Rhino. During annotations, separate layers distinguish building levels and windows versus balcony doors, permitting flexibility for future uses, though these are merged into three classes for the deep learning stage: (1) wall, (2) glazing, and (3) garage door.

Finally, a custom Grasshopper script converts bounding boxes into normalized coordinates, calculates WWR as annotated (i.e. the ground truth), and converts annotations to XML format, as depicted in Appendix A2.

2.4 Deep learning stage

2.4.1 YOLOv9 model

A YOLOv9 model was selected following precedent studies using earlier YOLO versions. Kong & Fan (2021) built a semantic segmentation network on top of a YOLOv3 detector, achieving best performance among peer studies on the ECP dataset, and good performance on non-rectified SVI from Mapillary. Sezen et al. (2022) used YOLO versions 3, 4, and 5 on non-rectified façade images from GSV, resulting in F1 scores of 81% and 78%, and mAP50 of 84% and 79% with v4 and v5, respectively. These predecessor YOLO models are outdated—v3 was introduced in 2018 (Farhadi & Redmon 2018), and v4/v5 in 2020 (Bochkovskiy et al. 2020; Ultralytics 2020). Thus, we hypothesized that YOLOv9, updated in 2024, could be effective for WWR detection in SVI, building on the successes of Kong & Fan (2021) and Sezen et al. (2022), while using rectified and cropped images to avoid the multiple perspectives and/or complex backgrounds in those works.

2.4.2 Evaluation metrics

For object detection tasks in deep learning, performance metrics include intersection over union (IoU), precision, and recall. IoU is calculated according to:

$$IoU = \frac{Area\ of\ Intersection}{Area\ of\ Union} \tag{1}$$

Precision and recall are based on whether a detection result is a true positive (TP), false positive (FP), or false negative (FN). Detections are positive if the IoU with the ground truth exceeds the IoU threshold, set to 0.5 in this work. Detections are true if confidence scores exceed the confidence threshold, indicating the model's confidence that a detected bounding box contains an object of interest and its confidence in predicting the object's class (Redmon et al. 2016); threshold selection is discussed in Section 4.4.

Using TPs, FP, and FNs, precision (P) and recall (R) are calculated according to:

$$P = \frac{TP}{(TP+FP)} \tag{2}$$

$$R = \frac{TP}{(TP+FN)} \tag{3}$$

Thus, precision indicates a model's accuracy upon prediction, and recall indicates when a model fails to make a prediction, and there is a trade-off in optimizing one or the other metric. For the present study, where the dataset widely contains occluded glazing, recall is an important indicator of successful detections despite occlusions. Appendix A3 provides example calculations of evaluation metrics.

The cumulative metrics F1 score and mean Average Precision (mAP) balance trade-offs between precision and recall. F1 score is defined by:

$$F1 = \frac{2PR}{(P+R)} \tag{4}$$

mAP requires ordering detections in order of confidence and plotting cumulative precision-recall (P-R) curves. Average precision (AP) is the area under multiple P-R curves when varying confidence thresholds, and mAP is the mean AP for all images in the dataset given an IoU threshold (Wu et al. 2020). The metric mAP50 uses an IoU threshold of 0.50 and mAP50-95 uses a range of IoUs from 0.50 and 0.95, in increments of 0.05.

2.4.3 Model training and validation

Experimentation and automatic tuning within YOLOv9 (employing genetic algorithm) are used to tune hyperparameters and maximize model performance, with IoU and confidence thresholds of 0.5 held constant. The final hyperparameters used to train the model are as per Appendix A4. Training begins with transfer learning, using weights by the YOLOv9 provider Ultralytics, pre-trained on the MS-COCO dataset containing 330,000 images (Ultralytics 2024). The YOLOv9 model applies default augmentations including jittering of color, brightness, and contrast, plus random rotation and scaling, among others. To each batch, we add blur functions using the Albumentations library (Buslaev et al. 2020), expecting that training with blurred images will lead to better predictions on the dataset where tall and wide buildings frequently contain noise at upper and lateral extents.

2.5 WWR analysis

2.5.1 WWR in detected façades

After training, the YOLOv9 model is run in predict mode to detect glazing, walls, and garage doors. For each image, object areas are calculated as pixel areas, and WWR is calculated according to:

$$WWR = \frac{A_{\mathrm{gl}}}{(A_{\mathrm{w}} - A_{\mathrm{g}})} \tag{5}$$

where A_{gl} is glazing area, A_w is gross area of perimeter walls, and A_g is garage area. Garage doors are deducted because behind these are unconditioned spaces, mere openings with passage to parking areas.

In each façade, the difference in WWR (ΔWWR) between detection results (WWR_d) and ground truth annotations (WWR_{gt}) is determined by:

$$\Delta WWR = WWR_{\mathrm{d}} - WWR_{\mathrm{gt}} \tag{6}$$

where 0% ΔWWR is ideal, and the mean of the absolute value of ΔWWR in each image is the key dataset-level metric.

2.5.2 WWR upscaling

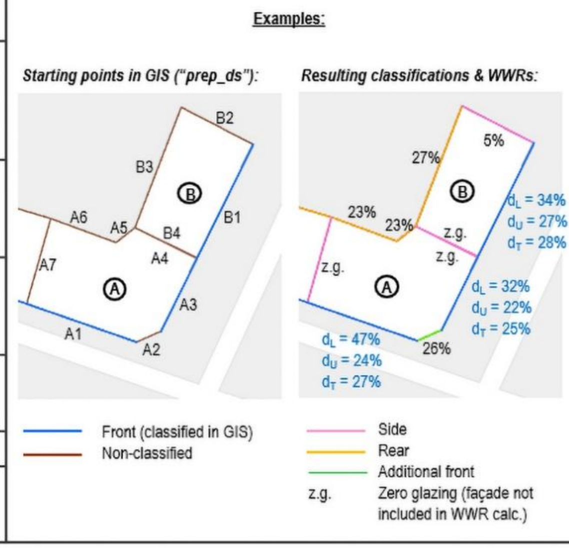
A novel algorithm achieves WWR upscaling in two steps, as detailed in Figure 8, first classifying building faces as additional street-facing fronts or non-street-facing rears and sides, and then using a custom script to assign a distinct WWR for each building line segment.

The first step begins with the "polygon neighbor" tool in ArcGIS, indicating if buildings intersect, and if so the developed script calculates distances between center points of each line in source and neighboring polygons, classifying segments with the minimum distance as attached sides. Next, additional fronts are determined if a line: (i) is a mid-segment between two fronts, or (ii) connects to a front with an angular difference less than a threshold (default set to 60º). Third, among remaining unclassified segments, detached sides are those connecting to a front at one point. Finally, residual segments are classified as rears. If a building has no street-facing fronts, the building is not classified.

Before proceeding to the second step, the script determines if a "primary" front façade was successfully extracted, meaning

Step 1 – Line segment classification:

Function	Conditions	Results Ⓐ	Results Ⓑ
1. is_front	True if the line exists in the original prepared GIS file with street-facing façades ("prep_ds").	A1, A3	B1
2. is_attached_side	Looks up neighbor tables for IDs of attached buildings, compares centroid distance between lines in source and neighbor buildings. True for lines with minimum distance from centroid to line segment with target neighbor ID.	A4, A7	B4
3. is_mid_segment	Only acts if there are two lines identified as fronts in prep_ds. True if a line segment has one common point with each of the two prep_ds lines. This line is considered a front.	A2	-
4. is_addtl_front	True if a line segment shares one common point with a segment previously classified as front, and the angle between the two lines is less than a threshold (default set to 60º).	-	-
5. is_side	True for segments that share one common point with a front, and true for segments previously classified as attached sides.	A4, A7	B2, B4
6. is_rear	True for any line that is not a front or side.	A5, A6	B3
7. no_front	If no fronts were present in prep_ds, all classifications are reset to false, and all building segments are not classified (side status set to "n/a").	-	-



Step 2 – WWR calculations:

A2: set to mean of total detected WWR for A1 & A3, weighted by line length → WWR = 26%
A4 & A7: script uses minimum distance between center lines of each segment to determine which is attached → assigns **zero glazing**
A5 & A6: script separates ground-level and upper-level glazing detections and assigns upper-level WWR (mean d_U of A1 & A3) to rear → WWR = 23%

B2: script looks up era of the building (Era 5) and appropriate Polygon Neighbor table (minimum 3 m distance), to determine the building meets regulations → WWR = 5%
B4: script uses minimum distance between center lines of each segment to determine which is attached → assigns **zero glazing**
B3: script separates ground-level and upper-level glazing detections and assigns upper-level WWR (d_U of B1) to rear → WWR = 27%

Figure 8. Algorithm to classify line segments and calculate building-scale WWR. Ground-level (lower), above ground-level (upper), and total detected WWRs are denoted by d_L, d_U, and d_T. Basemap source: Esri (2024).

a front segment within a squareness threshold (default set to 0.7) of the longest line in the polygon. This avoids WWR calculations for long rectangular buildings if multiple street-facing façades are identified at GIS stage but only short sides are extracted. Such sides might have limited or zero glazing, even if street-facing, and are not representative of the building's overall WWR.

In the second step, WWRs are assigned based on numerous criteria in the code. For street-facing fronts with no detections, detected WWRs are averaged, weighted on line length. For WWRs at building rears, which are not visible in SVI, we assume these are equal to front WWRs above the ground floor, where shops with extensive glazing may be present. Accordingly, we employ our post-detection script to stratify façade detections into ground-level and upper-level WWRs, and set the average of upper-level WWRs as the value for building rears.

Attached sides are excluded when upscaling WWR calculations. These sides have zero glazing, and walls attached to the heated space of a neighboring building do not count as "exterior" or "semi-exterior" building envelope, which constitute the denominator of WWR in ASHRAE 90.1 (ASHRAE 2022).

For detached sides, the code looks up neighbor tables with polygons buffered to distances of 3 or 10 meters, depending on the source building's era of construction, and if minimum distances are present, assigns WWRs of 0% or 5%. These assumptions are based on urban planning regulations in the case study (see Section 3), but could be adapted to other contexts. In the case study, the minimum required distances between building sides are 3 and 10 meters, based on laws from 1942 (Italian Republic 1942) and 1968 (Italian Republic 1968), corresponding to construction eras 4 and 6, respectively. While no regulations govern maximum WWRs at sides, we observe that detached, non-street-facing sides in the case district have a low WWR, approximately 5%. Thus, if a detached side meets the minimum legal distance to its neighbor, a WWR of 5% is applied, otherwise, a WWR of 0% is applied.

Finally, manual measurements are used to verify assumptions for rear and side WWRs, using a sample of 100 buildings selected to match TABULA archetype percentages existing in the district (see Table 2). Images of building rears and sides are extracted from Google Earth Pro (Google 2024c), manually rectified in Adobe Photoshop (Adobe 2024), and measured using the developed annotation workflow. Google Earth images have visibly lower image quality compared to GSV images, and frequent exterior window coverings at building rears occlude glazing. Thus, during measurement we visually reference aerial images from Bing Maps' "Bird's Eye" function (Microsoft 2024), as well as GSV images from side or rear angles, where available. WWRs from manual measurements are then compared against WWR assumptions.

Table 2. Case district buildings, by TABULA archetype. Residential ("R") archetypes: "AB" = apartment block; "MF" = multi family; "SF" = single family; "TH" = terraced house. Numbers 2 to 8 indicate construction era, with era 2 = 1918 and prior; era 3 = 1919-1945; era 4 = 1946-1960; era 5 = 1961-1970; era 6 = 1971-1990; era 7 = 1991-2005; era 8 = 2006 and after. Eras 1 and 2 are combined in this work because the GIS data relies on Italian National Statistics Agency eras that do not distinguish between buildings built before 1918.

Archetype	Building count	% of buildings in district	Façade count	Training set – count	Training set – %	Validation set – count	Validation set – %	Inference set – count	Inference set – %
R2AB	15	0.9%	22	5	1.1%	1	0.8%	12	1.0%
R3AB	105	6.6%	160	33	7.4%	12	9.4%	100	8.6%
R4AB	222	13.9%	394	63	14.1%	26	20.3%	196	16.8%
R5AB	280	17.5%	475	82	18.3%	21	16.4%	205	17.6%
R6AB	116	7.3%	189	31	6.9%	9	7.0%	57	4.9%
R7AB	41	2.6%	59	9	2.0%	3	2.3%	11	0.9%
R8AB	12	0.8%	19	1	0.2%	1	0.8%	4	0.3%
R2MF	26	1.6%	42	12	2.7%	1	0.8%	28	2.4%
R3MF	67	4.2%	96	24	5.4%	6	4.7%	58	5.0%
R4MF	181	11.3%	264	53	11.8%	13	10.2%	159	13.6%
R5MF	75	4.7%	95	26	5.8%	4	3.1%	43	3.7%
R6MF	29	1.8%	40	8	1.8%	0	0.0%	21	1.8%
R7MF	11	0.7%	14	1	0.2%	1	0.8%	4	0.3%
R8MF	5	0.3%	6	0	0.0%	0	0.0%	3	0.3%
R2SF	6	0.4%	6	1	0.2%	1	0.8%	3	0.3%
R3SF	24	1.5%	32	2	0.4%	3	2.3%	24	2.1%
R4SF	90	5.6%	108	25	5.6%	4	3.1%	52	4.5%
R5SF	50	3.1%	61	13	2.9%	0	0.0%	24	2.1%
R6SF	20	1.3%	22	2	0.4%	1	0.8%	14	1.2%
R7SF	7	0.4%	8	2	0.4%	0	0.0%	1	0.1%
R8SF	3	0.2%	4	2	0.4%	1	0.8%	2	0.2%
R2TH	3	0.2%	5	1	0.2%	0	0.0%	4	0.3%
R3TH	14	0.9%	16	3	0.7%	0	0.0%	13	1.1%
R4TH	86	5.4%	115	25	5.6%	7	5.5%	59	5.1%
R5TH	54	3.4%	67	9	2.0%	6	4.7%	40	3.4%
R6TH	25	1.6%	27	5	1.1%	4	3.1%	12	1.0%
R7TH	5	0.3%	6	4	0.9%	1	0.8%	1	0.1%
Not classified	25	1.6%	41	6	1.3%	2	1.6%	18	1.5%
Total	1597	100.0%	2393	448	100.0%	128	100.0%	1168	100.0%

2.6 Urban building energy model

In the final phase, a UBEM is built in Rhino/Grasshopper using the Dragonfly plug-in (Ladybug Tools 2024) to simulate energy consumption in buildings with the detected WWRs. A Dragonfly tool runs URBANopt (NREL 2024) to conduct full-dynamic simulations in 15-minute timesteps using EnergyPlus (DoE 2024). All model parameters and details follow those of a previous UBEM study in the same district case as the present research (Suppa & Corrado 2023).

The Dragonfly/URBANopt UBEM applies WWR values for each orientation (north, east, south, west) of each building. The per-orientation approach is less detailed than obtained in the presented detection workflow, which determines WWR for each line segment. Thus, the WWR data extracted is converted to a per-orientation basis. A developed script converts bearing of each segment to a positive value between 0° and 360°, subtracts 90° to determine the façade orientation (*FO*), and sets orientation based on:

- North: $FO \geq 315°$ or $FO < 45°$
- East: $45° \leq FO < 135°$
- South: $135° \leq FO < 225°$
- West: $225° \leq FO < 315°$

In a rectangular building, the conversion is straightforward, as each side corresponds to one of four directions. However, in a corner building with six or more segments, the conversion is more complex. If the building has no attached sides, we use line segment length to determine a weighted average for each orientation. If the segment is a side attached to a neighboring building, the script sets that side's weight to zero, so that attached walls are excluded from WWR calculations.

The open workflow presented in previous sections uses OSM geometry to obtain WWRs, though the OSM database lacks building heights required for energy modeling, and thus two UBEM versions are created. The first uses the municipal database (City of Turin 2023), containing detailed building volumes within OSM footprints and ground-to-eaves building heights. To the municipal dataset, we join WWRs from the deep learning detections and run the classification algorithm for WWRs at non-visible sides and rears. This UBEM is run twice, allowing a comparison of energy impacts when using detected WWRs at each orientation of each building versus constant per-archetype WWR values in TABULA.

A second UBEM uses OSM data, enabled by our script to extract building height and number of stories from façade detections using binary masks, where detected walls and windows are assigned values of 0 and 1. Non-zero sums of image columns and rows indicate presence of glazing, and permit splitting the mask first into vertical stacks and then into rows of windows, where the mode of rows is the number of stories (see Figure 15a). The script estimates building height using pixel ratio of width to height in the wall detection, multiplied by the line length (in meters) in the OSM database.

This second UBEM uses a subset of the OSM geometry, discarding buildings where lengths of front segments differ by ±10% of corresponding segments in the municipal dataset, to limit errors from discrepancies in OSM data (see discussion of image splitting, Section 4.1). A final check discards results where detected floor-to-floor heights are less than 2.5 meters or greater than 4 meters. Discarded polygons become context/shading objects in the simulation. The second UBEM permits comparison of building height, number of stories, and energy consumption between OSM and municipal datasets.

3 Case study application

The methodology is applied to the case of *Circoscrizione 6* (District 6) in Turin, Italy, home to 104,408 residents and spanning 25.5 km^2 (City of Turin 2022). Within the district, residential buildings fall into archetypes from the Italian implementation of TABULA, a Europe-wide project classifying buildings as single-family (SF), terraced house (TH), multi-family (MF), and apartment block (AB) types, which are subcategorized by construction era (Ballarini et al. 2014). For each archetype, typical characteristics are described, including wall and glazing areas. This work uses classifications of the district buildings from Suppa & Corrado (2023), based on surface/volume ratios, story count, and footprint area from the municipal database.

Table 2 summarizes archetypes for the 1,597 buildings in the case study, separated into training, validation, and inference datasets. The total of 1,597 is after GIS preparation to include only façades where orthogonal GSV images are expected, a subset of the 1,818 residential buildings in the OSM database. The façade count of 2,393 indicates that buildings have multiple street-facing façades to be attempted for extraction. The most prevalent building type in the district is AB (50%), followed by MF (25%), SF (13%), and TH (12%). The most common era of construction is era 4 (1946-1960) accounting for 36% of district buildings, followed by eras 5 (29%), 3 (13%), and 6 (12%).

As a qualitative measure of the district's architecture, a sample of 160 buildings over two stories in the district was reviewed, showing that 86% of these have an identical repeating glazing pattern between second and top floors, and 7% have identical glazing on alternating floors. Such repetition may enhance a deep learning model's ability to successfully detect objects despite occlusion from balconies and window coverings, which are common in the district.

4 Results

4.1 Image extraction

Of the 1,597 buildings in the case study, at least one façade image was extracted and stitched for 1,483 buildings (93%). Of the 2,393 street-facing façades identified at GIS stage, 2,293 were successfully extracted and stitched (96% of attempts), using 7,906 constituent images. The discrepancy of 100 façades not captured represents failed extractions, e.g. if an image did not exist within the 20-meter target radius. The extraction and stitching process took approximately 2 hours using CPU runtime in Google Colab.

Qualitative evaluation resulted in 1,631 images accepted for the deep learning stage (71% of extracted façades, representing 1,251 buildings), as summarized in Table 3. Obstructions due to window coverings (exterior curtains or solarium glazing) and landscaping (vegetation or fencing) were the most frequent reasons to reject images, meaning that these objects occluded glazing at both second and third levels for at least one vertical stack of glazing, and accounting for over 40% of all rejections.

During image review, façades were split (e.g. left or right) where images contained multiple buildings with different building heights or window alignments. This occurred when OSM data show one contiguous building but GSV data show distinct buildings. Since images were extracted on OSM coordinates, such images were manually split to reflect the reality of the separate buildings.

Image splitting added 35 and 78 images to the Train/Val set and Inference sets, respectively. Finally, 40 images were moved from the Inference to the Train/Val set to better align the archetypes of buildings in the dataset with those of the district (per Table 3) while ensuring training

Table 3. Qualitative evaluation results. Image counts are before image splitting or reallocations. Reasons rejected: BCC = bad capture – camera; BCO = bad capture – orthogonality; BCA = bad capture – angle; BCP = bad capture – proximity; BL = blurriness; NUO = not usable – other; OB = obscured at balconies; OL = obscured by landscaping; OS = obscured by scaffold/construction equipment; OWC = obscured by window coverings.

				Reason rejected									
Dataset	Images extracted	Accepted	Rejected	BCC	BCO	BCA	BCP	BL	NUO	OB	OL	OS	OWC
Train/Val	700	501	199	13.1%	19.1%	0.0%	7.0%	7.5%	9.5%	4.5%	14.1%	5.5%	19.6%
Inference	1593	1130	463	13.5%	0.0%	5.7%	2.8%	6.8%	11.1%	6.8%	21.0%	6.8%	25.5%
Total	2293	1631	662	13.4%	5.8%	4.0%	4.1%	7.0%	10.7%	6.1%	18.9%	6.4%	23.7%

and validation sets of batch size 32. Thus, the final number of façade images in the Train/Val set is 576, with 448 for training (78%) and 128 for validation (22%), and 1,168 images in the Inference set.

Before annotations, accepted Train/Val images were measured for orthogonality. At second-level glazing, the mean difference from orthogonal was 0.7 with a standard deviation of 0.6, indicating that most images are orthogonally correct at lower floors. At the uppermost visible glazing, the difference from orthogonal was significantly higher, with a mean of 1.8 and standard deviation of 2.4, which was evident in images with greater warping visible at upper levels.

4.2 Image annotation results

The 576 Train/Val images were annotated using the presented tool, taking approximately 100 hours for one researcher. A total of 11,658 objects were labeled, including 10,898 glazing objects, 578 walls, and 182 garage doors. A large class distribution imbalance was present, averaging 19 glazing, 1 wall, and 0.3 garage labels per image. During annotations, glazing objects were sub-classified; around 29% of these were balcony doors above the second level, indicating potential occlusion by balconies. As noted, windows and glazed doors were merged into one glazing class for the deep learning stage.

4.3 Deep learning results

Training and validation of the YOLOv9 model were completed using the 448 training and 128 validation images. The final model was trained in batches of 32 for 92 epochs, for a total of 1,288 iterations, taking 24 minutes using a 40-gigabyte GPU runtime in Google Colab. The evaluation metrics precision, recall, F1 score, mAP50, and mAP50-95 were calculated on the validation set at the end of each epoch, where IoU and confidence thresholds are held constant at 0.5. Losses decreased smoothly throughout training without overfitting, with proportional increases in the evaluation metrics at each validation step, as depicted in Figure 9. Numerous experiments were conducted with different hyperparameters, though the metrics shown are for the final model, selected based on the greatest F1 and mAP50-95 scores.

4.4 WWR analysis – detected façades

To calculate WWR on detected façades, the model was run in predict mode on the validation dataset, and three post-processing scripts were added. First, non-maximum suppression (NMS) is added to suppress overlaps between garage and glazing objects, since the default YOLO model uses NMS only on items of the same class. Second, a custom post-processing script suppresses glazing detections outside walls. Third, a script splits detections into ground level and above-ground level, as stratified WWRs for ground and upper levels are used in later stages when predicting WWR at the rear of buildings.

To determine the ideal confidence threshold for detections, WWR was calculated for confidence thresholds from 0.1-0.9, in 0.1 increments, dividing pixel areas of detected windows by detected net wall area (i.e. wall area less garage). Table 4 summarizes predictions on the validation set using the added post-processing scripts, indicating a greater number of glazing object detections with decreasing confidence thresholds. Thresholds of 0.2 and 0.3 have the lowest mean absolute value of ΔWWR, at 1.8%. A visual review of the results showed that the lowest confidence thresholds included extraneous detections while higher thresholds missed many predictions.

We thus selected a confidence threshold of 0.3 for detections, conducting several analyses of ΔWWR on the validation dataset, as depicted in Figure 10. Included is the distribution of ΔWWR, indicating that 94% and 100% of images detected within ±5% and ±10% ΔWWR, respectively. The included boxplots do not appear to show significant trends between ΔWWR and building type or construction era. In the plot of building height versus ΔWWR, the linear regression shows a slight trend of taller buildings with increasingly negative ΔWWR values. Such a trend can be partly explained by decreased image quality in taller buildings, with increased graininess and distortion at balconies on upper floors as seen in Figure 11.

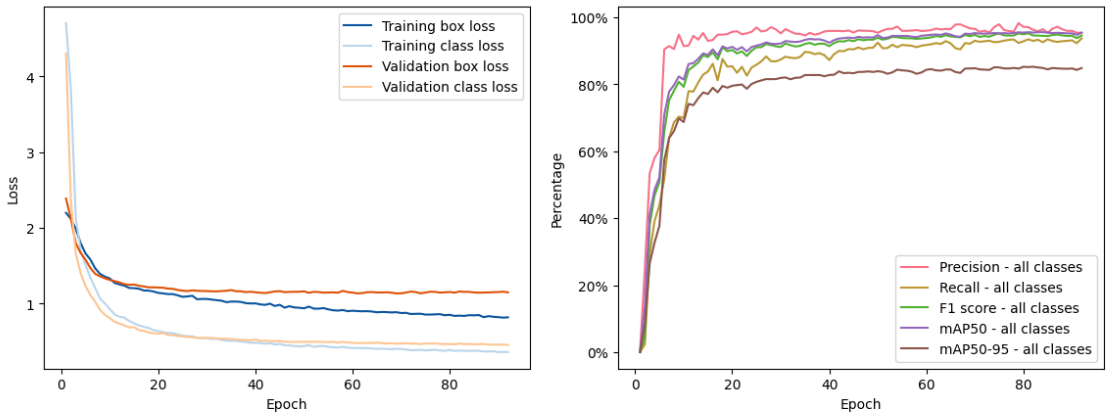


Figure 9. Evaluation metrics on the validation set, by epoch.

Table 4. WWR evaluation of validation set, at confidence thresholds from 0.1-0.9 and constant IoU threshold of 0.5.

	Confidence threshold								
Metric	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**
No. detected glazing objects	2666	2563	2487	2423	2329	2227	2100	1836	556
Mean absolute value of ΔWWR	1.9%	1.8%	1.8%	1.9%	2.2%	2.6%	3.2%	4.5%	12.2%
Mean ΔWWR	0.3%	-0.2%	-0.6%	-0.9%	-1.3%	-1.8%	-2.6%	-4.2%	-12.1%
Min Δ_{WWR}	-7.9%	-7.9%	-9.9%	-13.2%	-13.2%	-14.3%	-16.3%	-21.6%	-27.6%
Max Δ_{WWR}	8.4%	7.5%	7.5%	7.5%	7.5%	7.5%	7.5%	3.7%	1.2%
Detections within ±5% ΔWWR	117 (91%)	121 (95%)	120 (94%)	120 (94%)	116 (91%)	108 (85%)	97 (76%)	83 (66%)	26 (21%)
Detections within ±10% ΔWWR	128 (100%)	128 (100%)	128 (100%)	127 (99%)	124 (98%)	123 (97%)	123 (97%)	108 (85%)	46 (38%)

Figure 11 depicts detections after the three added post-processing scripts and resulting ΔWWRs[3]. The top row of images (a and b) shows results with ΔWWR near zero; even with occlusion from balcony slabs and noise in the GSV image, balcony doors are correctly detected. The second row indicates three positive ΔWWR values: in case (c) this is attributable largely to a difference in the wall area from ground truth to detected; in (d) the balcony door and abutting window are grouped as one larger rectangle, as are two separate windows above the entry door; and (e) has oversized balcony door detections at second and fourth levels where exterior curtains obscure the true size of doors[4]. The fourth third shows small negative variances due to missed detections (f) and due to undersized windows at the upper level (g) where blurriness and noise are present in the upper image extents. Finally, the bottom row shows larger negative variances where upper windows are not detected, with evident blurriness and distorted balcony perspectives in images of these taller buildings.

Finally, before estimating WWR at building and district scales, the model was run in predict mode on the Inference set, yielding predictions for 1,167 of 1,168 images in the dataset. The total of 1,743 façade images from this stage (1,167 Inference + 576 Train/Val) represent 1,251 buildings in the district.

4.5 WWR analysis – building and district scales

The automatic classification algorithm was run on all buildings in the district, resulting in 6,139 line segment classifications in the OSM dataset. 90% of buildings were classified as having two non-street-facing sides, which is the number expected for rectangular buildings or corner L-shaped buildings. A sample of the district is depicted in Figure 12, with results from the Python script imported to GIS and color-coded by classification, with 92% of line segments correct among 649 lines classified.

Among the 1,251 buildings in the Train/Val and Inference sets, 1,064 of these met the squareness condition in Section 2.5. For the subsequent UBEM step, WWR detections for the 1,064 OSM buildings were joined to 1,432 volumes in the municipal dataset, indicating that multiple volumes exist within OSM footprints. The automatic classification algorithm was applied to both datasets, yielding building-scale WWR results as shown in Figure 13, which depicts detected WWRs compared to TABULA values for the predominant archetypes in the district, comprising 97% of gross floor

[3] All detection results show bounding boxes directly from model predictions plus post-processing scripts, plotted without modification onto facsimile images created by the authors.

[4] In the original GSV photo, the curtain at left at second level is drawn closed.

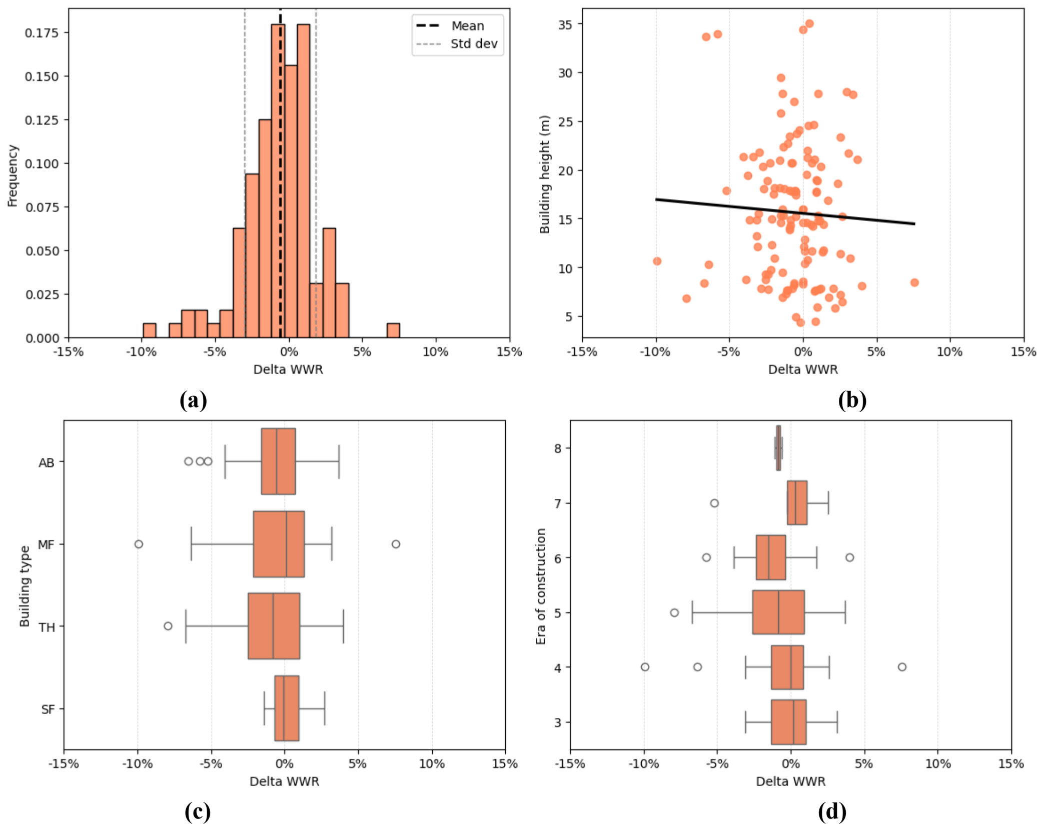


Figure 10. Analysis of ΔWWR in the validation set using confidence threshold of 0.3, including distribution of ΔWWR (a); building height versus ΔWWR (b); boxplot of building type and ΔWWR (c); and boxplot of era of construction and ΔWWR (d).

area. The detected WWRs were higher than TABULA WWRs for most archetypes, with a mean difference of +5.2% and +6.9% in the OSM and municipal datasets, respectively, compared to TABULA values.

To verify assumptions related to WWR upscaling, 100 buildings were manually measured using Google Earth images, at non-street-facing sides and rears. At the rear of buildings, the difference in WWR from rear to front (above ground floor) was a mean of -2.0% with standard deviation of 5.5%, aligning with the assumption that front and rear WWRs are similar. At non-street-facing sides, all except four measured buildings conformed to the regulatory requirements stated in Section 2.5. The 175 non-street-facing sides evaluated included 103 sides attached to neighboring buildings, reflecting the dense urban environment of the case study, where there is zero glazing and the walls are not included in building-scale WWR calculations. Among the remaining 72 detached sides, the mean WWR was 4.6%, with standard deviation of 5.5%, aligning with the WWR assumption of 0% or 5% at non-street-facing sides.

4.6 UBEM results

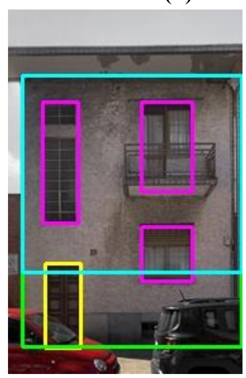
For the municipal-data UBEM, simulations were run twice, adopting TABULA and detected WWRs while holding other values constant, taking approximately 10 hours each. In terms of energy services, only thermal energy needs for heating and cooling varied in the two simulations, as energy use for domestic hot water, lighting, and plug loads are calculated on time- or area-based schedules. Heating energy need constituted 173.9 GWh and 171.3 GWh when simulated with TABULA and detected WWRs, respectively, a 1.5% increase. Cooling energy need showed much smaller initial values totaling 5.4 GWh with TABULA WWRs, increasing to 7.4 GWh with the detected WWRs, for a significantly larger increase of 35.5%. On average, for a 1% increase in WWR, the heating energy need increased by 0.08% and cooling need increased by 8.3%. Analyzing different building types, apartment block (AB) buildings showed the greatest variance in heating energy need (+2.0%) and cooling energy need (+46.0%), but also had a slightly higher difference between detected and TABULA WWR values (+7.8%), whereas single family (SF) buildings showed the smallest change in both heating (0.0%) and cooling (+1.0%) with a difference in detected and TABULA WWRs of +1.3%. Deviations in WWR compared to percentage deviations in space conditioning energy need for each building type are depicted in Figure 14.

Figure 15 depicts results related to the UBEM using OSM data, where a total of 307 buildings were first detected

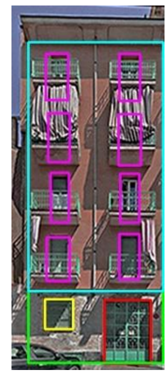
(a) WWR$_d$: **17.8%; ΔWWR: +0.1%**

(b) WWR$_d$: **17.2%; ΔWWR: +0.3%**

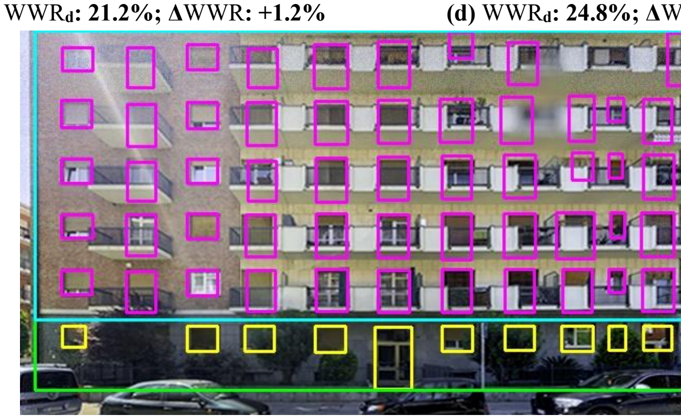
(c) WWR$_d$: **21.2%; ΔWWR: +1.2%**



(d) WWR$_d$: **24.8%; ΔWWR: +2.1%**

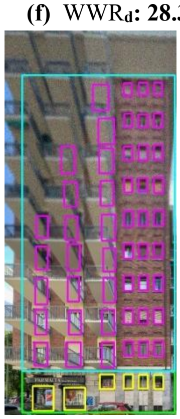


(e) WWR$_d$: **22.0%; ΔWWR: +3.7%**



(f) WWR$_d$: **28.3%; ΔWWR: -1.4%**

(g) WWR$_d$: **24.3%; ΔWWR: -2.0%**



(h) WWR$_d$: **22.7%; ΔWWR: -5.8%**

(i) WWR$_d$: **22.1%; ΔWWR: -6.6%**

Figure 11. Detections on the validation dataset, including ground-floor glazing (yellow), above-ground glazing (magenta), ground-level wall (green), above-ground wall (cyan), and garage (red).

Figure 12. Visualization of automatic building classification. Street-facing façades as identified during GIS stage are shown in blue. Classification algorithm output: green = additional street-facing façade; magenta = non-street-facing side; orange = rear. Buildings not classified (gray) are in the OSM database but had no street-facing front identified at GIS stage. Basemap source: Esri (2024).

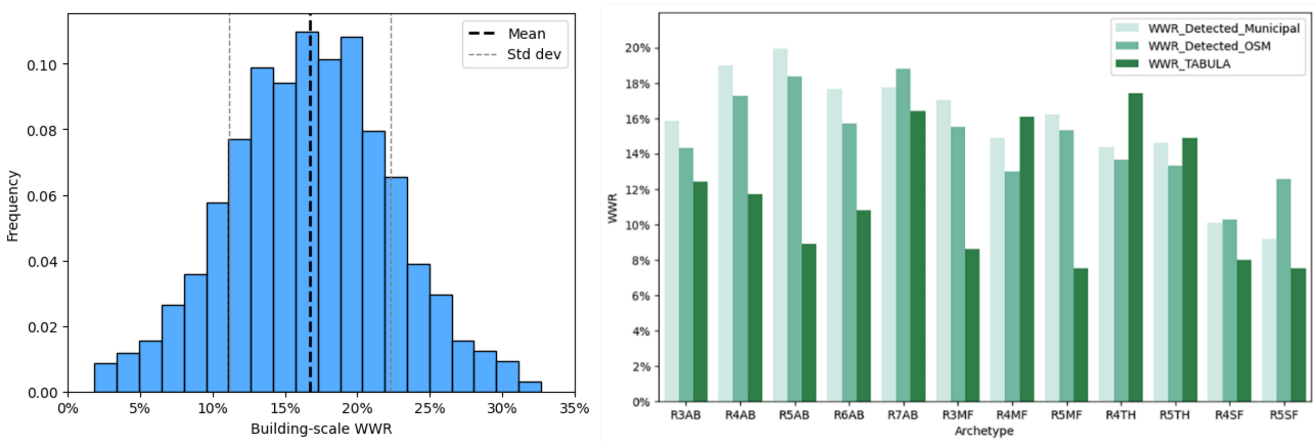


Figure 13. Building- and district-scale WWR results. Left: distribution of building-level WWR when upscaling using OSM geometry. Right: distribution of WWR for the predominant archetypes in the district, aggregated by gross floor area.

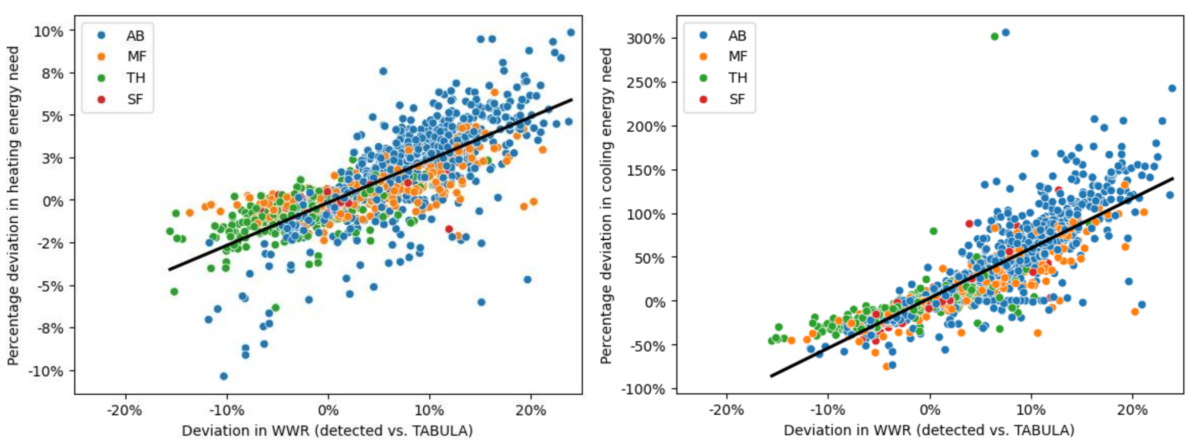


Figure 14. UBEM results with variance in heating (left) and cooling (right) energy need compared to variance in WWR (detected minus TABULA), with building types indicated in colors noted.

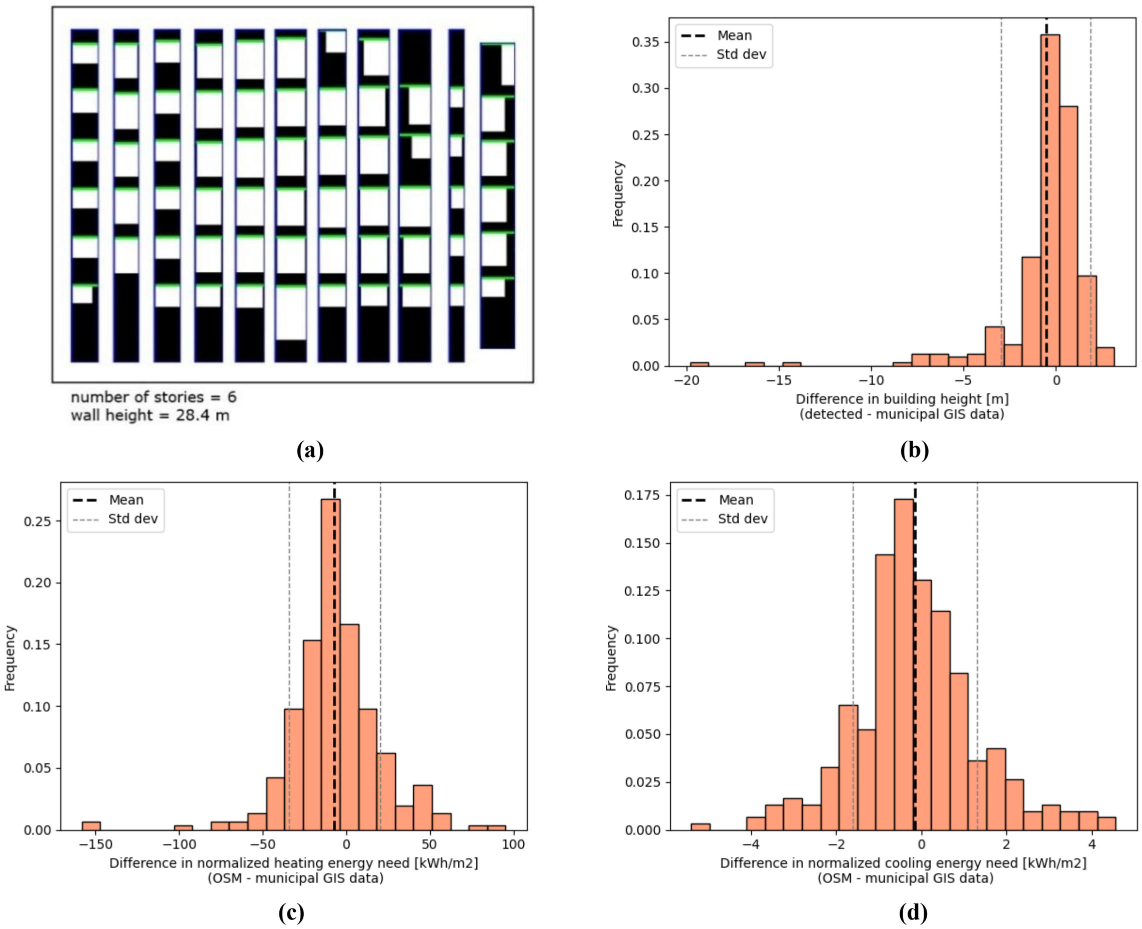


Figure 15. Results related to UBEMs comparing OSM and municipal GIS data. Visualization of post-script counting number of stories, using the building in Figure 11f as an example (a); distribution of difference in building height between detections and in municipal dataset (b); distribution of difference in normalized heating energy demand (c); and distribution of difference in normalized cooling energy demand (d).

for number of stories and building height. For number of stories, the mean difference between detected and municipal data was -0.18 stories, with 82% of cases showing zero difference. Building height differed by -0.54 meters on average. Values for annual normalized energy need differed more significantly. The UBEM on OSM data showed 117.4 kWh/m^2 and 5.0 kWh/m^2 for heating and cooling energy need, compared to 129.0 kWh/m^2 and 5.5 kWh/m^2 in the UBEM based on municipal GIS data. In percent terms, the UBEM based on OSM data showed -9% and -10% for normalized heating and cooling energy need, respectively, compared to the UBEM on municipal data.

5 Discussion

5.1 Deep learning model comparison

To demonstrate the performance of the presented YOLOv9 model with added post-scripts on other building types and contexts, the model was tested on the Graz, CMP, ECP, and ENPC PDs, using 20 images for the first three datasets and 14 from ENPC.[5] As indicated in Table 5, the model shows very good performance on Graz, CMP, and ECP, and acceptable performance on ENPC, compared to previous studies. Notably, another YOLO-based model had the best performance on the ECP dataset (Kong & Fan 2021).

Our model's performance on ENPC is likely affected by street trees in the dataset (see Figure 1), which obscure several stories in some images; while our model saw many examples of glazing obscured by balconies, there were few images with large trees occluding façades in the training set, since images were excluded if vegetation obstructed second and third levels for one or more vertical stacks of glazing.

Three qualifications are required for our test results. First, our model is trained to detect glazed balcony doors extending down to the top of balcony slabs, which differs

[5] For Graz, ECP, and ENPC, the test images were selected match those in the first test folder of the online repository of Ma et al. (2022c); for CMP, the 20 images were randomly selected.

Table 5. Model comparison with metrics for the glazing class. Peer studies: (1) = Femiani et al. 2018; (2) = Liu et al. 2019; (3) = Li et al. 2020; (4) = Kong & Fan 2021; (5.1) = Ma & Ma 2021; (5.2) = Ma et al. 2022b; (6) = Sun et al. (2022), (7) Grounding DINO model using Roboflow annotation software (Roboflow 2025).

	Pixel accuracy								Mean IoU				
Dataset	**(1)**	**(2)**	**(3)**	**(4)**	**(5.2)**	**(6)**	**(7)**	**Ours**	**(1)**	**(5.1)**	**(5.2)**	**(7)**	**Ours**
Graz	-	88.8	90	-	92	95.2	96.7	97.7	71.3	79.1	79.3	78.2	82.9
CMP	96.0	95.0	-	-	96.0	96.5	95.0	96.6	-	-	-	70.0	75.9
ECP	95.6	97.6	95	98.6	88.1	97.2	95.3	95.3	80.3	79.4	79.2	77.8	79.2
ENPC	-	95.4	-	-	89.7	96.1	90.5	93.5	70.7	81	82.3	56.0	71.1

from how PDs are annotated (see Section 1.1), and thus how the peer studies are trained. To solve this, we re-annotated images in the test set using the presented annotation workflow, matching the image pre-processing of Section 2.3.[6] Second, our results above are for above-ground floor only, since the peer studies consider shops at ground floor level, which our model would detect as glazing. Last, all above-noted peer studies except Li et al. (2021) use semantic segmentation models, and the metrics for accuracy and IoU are determined at the pixel level instead of using an IoU threshold (e.g. 0.5) to evaluate bounding box detections (see Section 2.4.2). Similar to Li et al. (2020), we convert our detected bounding boxes to segmentation masks, and evaluate accuracy and IoU according to formulas in Ma et al. (2022b).

The aim of the above comparison is not to indicate competitiveness with peer studies but to demonstrate the trained model's validity as a step within the comprehensive workflow for district-scale WWR.

5.2 Annotation tool comparison

The custom Rhino-based tool was effective in locating occluded glazing using overlaid gridlines, and copy/pasting with object snapping allows quick reproduction of identical bounding boxes. However, the time associated with the manual annotations was significant—around 100 hours, or 10 minutes per image. New commercial services such as Roboflow (Roboflow 2025) provide automatic annotations using the Grounding DINO algorithm, detecting an open set of objects using input terms such as category names (Liu et al. 2024). We compare our custom tool on the test sets described above to automatically generated labels from Roboflow. The automatic method completed each test set of 14-20 images within two minutes total, whereas manual annotations required 6-8 minutes per image (annotating above-ground glazing only). The Roboflow automatic annotations inherently present lower evaluation metrics compared to the manual Rhino-based method, since the ground truth annotations have metrics of 100% by definition. The Roboflow method also has slightly lower accuracy and IoU compared to detections using the trained YOLOv9 model, as shown in Table 5.

5.3 WWR upscaling and UBEM results

Upscaling detected WWRs, the per-archetype results were similar in the two datasets (see Figure 13), with the municipal dataset generally showing 1-2% higher WWRs than the OSM dataset. This difference is partly due to more polygons in the municipal dataset (8,815 vs. 3,679 in OSM), leading to more attached neighbors and attached walls, which are excluded from WWR calculations, thus reducing the denominator of WWR and increasing the building WWR result.

The UBEM comparing detected and TABULA WWRs showed small increases in heating energy need and large increases in cooling energy need, which can be partly explained by high *g*-values (0.75-0.85) in older buildings, as 95% of the district was built before 1977. The literature shows that as solar heat gain coefficient increases, cooling energy needs increase and heating energy needs decrease (Lyu et al. 2025; Lee et al. 2023), so more glazing area with a high *g*-value would reinforce that trend. The results are also consistent with Szczesniak et al. (2022), where two- to six-story apartment buildings over 62 years old showed small changes in heating energy demand (-0.7%) and larger changes in cooling energy demand (-6.7%) when the UBEM was simulated with detected WWRs that were smaller (-7.4%) than the default WWR assumption of 40%.

The detected height and story counts showed only small differences between OSM and municipal data, suggesting the value of our post-script to fill gaps in OSM data, if height/story data are lacking or municipal data are not available. Such enriched data is valuable not only for UBEM but also for other urban-scale simulations, as building typology significantly affects shading, daylighting, and urban

[6] Among the 74 test images, 5 were cropped at top floor step-backs, 5 cropped at sides, and 4 split into left and right buildings; all test images and annotations are made available at: https://github.com/asuptor/WWRdetection

heat islands (Ang et al. 2020). Only 307 of the OSM buildings were detected using this method due to constraints imposed, but this count could be increased using a method similar to that described in Tarkhan et al. (2024) to manually check measurements in Google Earth during image review and cropping stage, thereby ensuring that the line lengths in the OSM data match present in the images.

The UBEM comparing the two datasets showed differences of 9-10% in normalized space conditioning energy need. Such differences may be acceptable and may highlight the value of the OSM-based UBEM workflow for some use cases, if data gaps are present and a minimum amount of detail is sufficient for a working model. One example could be initial insight on carbon emission savings from district-scale building retrofit scenarios (Ang et al. 2020).

5.4 Validity and limitations

The workflow to extract and stitch GSV images is flexible and scalable. Images can be obtained in 90 countries where GSV operates (Biljecki & Ito 2021) using the developed code. Limits to procuring images for deep learning are the time to review the automatic street-facing classifications in GIS, time to review and crop extracted images, and budget for GSV images. In the current work, the GIS file with 1,818 buildings was reviewed in 1.5 hours, and approximately 100 images were reviewed and cropped per hour, both of which should scale linearly. Due to promotions at the time of writing, 7,906 GSV images were extracted at zero cost.

The trained deep learning model with added post-scripts could be applied outside the case study of Turin, given strong performance on PDs from diverse locations. These datasets and the case study all primarily contain masonry buildings with punched windows; thus, the presented model must be limited to this typology, and one cannot assume the model would function on other cladding types without additional testing. Another limitation, while the model performs well despite occlusions from balcony slabs, the test on ENPC shows reduced performance when other occlusions are present (i.e. large street trees). Additional training with tree-obscured images could potentially solve this shortcoming, but this was out of scope for the study. Considering strong performance on the Turin and PDs, and in comparison to the out-of-the-box Grounding DINO detection model, the trained YOLOv9 model is effective as a component of the comprehensive WWR workflow, though it could be substituted as new deep learning architectures emerge.

The building façade classification algorithm could also be applied to other contexts, given its success with diverse building shapes in the case study. The algorithm is also effective with detached and semi-detached geometries in less dense areas, when GSV coverage is continuous at building fronts. However, the automatic classification algorithm does not work well in atypical cases where building fronts face private driveways at block interiors, with public streets only at block perimeters. This was observed in two large blocks in the case district with 29 buildings connected by internal driveways, where images captured included only short building sides or no façades at all.

Another limitation, the calculation of building-level WWR relied on urban planning laws in Italy and observations in the case district, and thus cannot be extended outside this context. The existing code could potentially be modified to suit other contexts if local laws and/or common architectural practices provide indications on WWR at non-street-facing sides and rears.

In any case, testing should be conducted before applying the trained deep learning model or classification algorithm to another context, using methods demonstrated in this work, including testing the trained model on an annotated local dataset and manual measurement with Google Earth imagery.

Last, the comparison between the presented annotation tool and the fully automatic Grounding DINO model shows that our tool has the advantages of greater accuracy and IoU and the disadvantage of significantly slower speed. A limitation, the effect of the increased annotation accuracy on WWR outcomes in the case study could not be evaluated, as uploading the extracted images to a third party would violate the GSV user agreement.

6 Conclusion

This study provides a comprehensive workflow to detect WWR at district scale, including automatic image extraction, annotation using a custom Rhino-based tool, upscaling deep-learning-based WWR detections to each face of each building, and applying detected WWRs in UBEM to determine energy impacts. The key findings can be summarized as follows:

- The extraction and stitching algorithm is robust, with 96% of targeted images extracted and stitched and 71% of images were accepted based on developed criteria.
- The developed annotation tool improves the ability to locate glazing occluded by objects such as balcony slabs, which are common in SVI.
- The trained deep learning model showed 94% of WWR detections were within ±5% of ground truth WWR in the case study, with strong performance on the Graz, CMP, and ECP public datasets.
- The developed classification algorithm effectively estimated WWRs at each face of each building, as verified by measurement using Google Earth data, improving on the status quo of setting non-visible sides and rears as equal to WWR detected in street-facing fronts.

- Detected WWRs were 5.2%-6.9% higher than archetype values in TABULA, leading to small changes in heating demand (+1.5%) and large changes in cooling demand (+35.5%).
- The script to detect façade images for number of stories and building height showed to have close alignment to municipal data, proving the tool's value to fill gaps in OSM data.
- The workflow enabled UBEM using OSM data, showing a 9-10% difference in normalized space conditioning values compared to using municipal data, which may be adequate for some use cases.

Future research could address existing glazing performance. By quantifying *U*-values of existing windows using thermography (Sadhukhan et al. 2020; Geske et al. 2024), and linking these to WWR detections, one could better quantify energy performance in baseline buildings and energy savings potential of window-substitution retrofits. Future research could also address the dynamic nature of urban development and its impact on district-scale WWR. The present workflow would require re-extracting district images and re-running detections to compare changes over time, though future research could better account for façade renovations changing WWRs or new buildings constructed in a district.

The code made openly available in this work can support future urban-scale research, adapted as required to suit other contexts, thus filling data gaps for energy-related studies as well as daylighting, view quality, and thermal comfort simulations. We expect the present research to be valuable to practitioners of urban-scale building modeling and decision-makers guiding retrofit policies in the urgent transition to climate neutrality.

Data use statement

All test images and annotations from the referenced public datasets are made available in the linked data repository. The street view dataset used is confidential. Google Street View and Google Earth data have been used further to European Union Directive 2019/790 on copyright and related rights in the Digital Single Market, and its national implementation in Italian law, the legislative decree of 8 November 2021, n. 177. Due to Google copyright on GSV images, all street view images printed in this work are facsimiles of the GSV images, using photos taken by the authors.

References

Adobe. (2024). Adobe Photoshop, v.25.12.0. [Software]. https://www.adobe.com/products/photoshop.html

Ali, U., Shamsi, M. H., Hoare, C., Mangina, E., & O'Donnell, J. (2021). Review of urban building energy modeling (UBEM) approaches, methods and tools using qualitative and quantitative analysis. *Energy and buildings*, *246*, 111073.

Ang, Y. Q., Berzolla, Z. M., Letellier-Duchesne, S., Jusiega, V., & Reinhart, C. (2022). UBEM. io: A web-based framework to rapidly generate urban building energy models for carbon reduction technology pathways. *Sustainable Cities and Society*, *77*, 103534.

American Society of Heating, Refrigerating and Air-Conditioning Engineers (ASHRAE). (2022). *Energy Standard for Buildings Except Low-Rise Residential Buildings* (ASHRAE Standard 90.1-2022).

Ang, Y. Q., Berzolla, Z. M., & Reinhart, C. F. (2020). From concept to application: A review of use cases in urban building energy modeling. *Applied Energy*, *279*, 115738.

Badrinarayanan, V., Kendall, A., & Cipolla, R. (2017). Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, *39*(12), 2481-2495.

Ballarini, I., Corgnati, S. P., & Corrado, V. (2014). Use of reference buildings to assess the energy saving potentials of the residential building stock: The experience of TABULA project. *Energy policy*, *68*, 273-284.

Battersby, S. E., Finn, M. P., Usery, E. L., & Yamamoto, K. H. (2014). Implications of web Mercator and its use in online mapping. *Cartographica: The International Journal for Geographic Information and Geovisualization*, *49*(2), 85-101.

Big Vision (2022). *Mean Average Precision (mAP) in Object Detection.* [Online]. https://learnopencv.com/mean-average-precision-map-object-detection-model-evaluation-metric/.

Biljecki, F., & Ito, K. (2021). Street view imagery in urban analytics and GIS: A review. *Landscape and Urban Planning*, *215*, 104217.

Bochkovskiy, A. (2020). Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*.

Bradski, G. (2000). The OpenCV library. Dr. Dobb's Journal: Software Tools for the Professional Programmer, 25(11), 120-123.

Buslaev, A., Iglovikov, V. I., Khvedchenya, E., Parinov, A., Druzhinin, M., & Kalinin, A. A. (2020). Albumentations: fast and flexible image augmentations. *Information*, *11*(2), 125.

Cetiner, I., & Ceylan, N. (2013). Environmental consequences of rehabilitation of residential buildings in Turkey: A case study of Istanbul. *Building and Environment*, *69*, 149-159.

Chen, Y., Hong, T., Luo, X., & Hooper, B. (2019). Development of city buildings dataset for urban building energy modeling. *Energy and Buildings*, *183*, 252-265.

City of Turin (2022). Registered population by gender and district, data from 31/12/2022. [Online]. http://www.comune.torino.it/statistica/dati/2022/pdf/A1%20Pop%20per%20Sesso%20e%20Circ.pdf

City of Turin. (2023). *GIS database for City of Turin.* [Electronic database]. Used as per framework agreement between Turin and Politecnico di Torino pursuant to Art. 15 L. 241/1990.

Dahlström, L., Broström, T., & Widén, J. (2022). Advancing urban building energy modelling through new model components and applications: A review. *Energy and Buildings*, *266*, 112099.

De Jaeger, I., Ma, Y., & Saelens, D. (2018). Estimating window dimensions of residential buildings in district energy models. 2018 Building Performance Analysis Conference and SimBuild, September 26-28, 2018, Chicago, USA.

Dutta, A., & Zisserman, A. (2019). The VIA Annotation Software for Images, Audio and Video. In *Proceedings of the 27th ACM International Conference on Multimedia (MM '19)*, October 21–25, 2019, Nice, France.

Esri 2024. ArcGIS Pro 3.0 [Software]. https://pro.arcgis.com/en/pro-app

Everingham, M., Van Gool, L., Williams, C. K., Winn, J., & Zisserman, A. (2010). The pascal visual object classes (VOC) challenge. *International journal of computer vision*, *88*, 303-338.

Farah, A., & Algarni, D. (2014). Positional accuracy assessment of GoogleEarth in Riyadh. *Artificial Satellites*, *49*(2), 101-106.

Farhadi, A., & Redmon, J. (2018, June). Yolov3: An incremental improvement. In *Computer vision and pattern recognition* (Vol. 1804, pp. 1-6). Berlin/Heidelberg, Germany: Springer.

Femiani, J., Para, W. R., Mitra, N., & Wonka, P. (2018). Facade segmentation in the wild. *arXiv preprint arXiv:1805.08634*.

Feng, G., Chi, D., Xu, X., Dou, B., Sun, Y., & Fu, Y. (2017). Study on the influence of window-wall ratio on the energy consumption of nearly zero energy buildings. *Procedia Engineering*, *205*, 730-737.

Ferrando, M., Causone, F., Hong, T., & Chen, Y. (2020). Urban building energy modeling (UBEM) tools: A state-of-the-art review of bottom-up physics-based approaches. *Sustainable Cities and Society*, *62*, 102408.

Gadde, R., Marlet, R., & Paragios, N. (2016). Learning grammars for architecture-specific facade parsing. *International Journal of Computer Vision*, *117*, 290-316.

Geske, M., Benz, A., & Voelker, C. (2024). U-value data on an urban scale: Outlier detection using comparative thermography to improve data quality. *Energy and Buildings*, *309*, 114026.

Google (2024a). Google Maps, Street View. [Online]. https://www.google.com/maps/

Google (2024b). *Classification: Precision and Recall*. [Online]. https://developers.google.com/machine-learning/crash-course/classification/precision-and-recall

Google (2024c). Google Earth Pro [Software]. https://earth.google.com

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask R-CNN. In *Proceedings of the IEEE international conference on computer vision* (pp. 2961-2969).

Hong, T., Chen, Y., Luo, X., Luo, N., & Lee, S. H. (2020). Ten questions on urban building energy modeling. *Building and Environment*, *168*, 106508.

Italian Republic (1942). Italian Civil Code. [Online]. https://www.gazzettaufficiale.it/dettaglio/codici/codiceCivile

Italian Republic (1968). Ministerial decree of 2 April 1968. [Online]. https://www.gazzettaufficiale.it/eli/id/1968/04/16/1288Q004/sg

Kong, G., & Fan, H. (2021). Enhanced facade parsing for street-level images using convolutional neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, *59*(12), 10519-10531.

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

Li, C. K., Zhang, H. X., Liu, J. X., Zhang, Y. Q., Zou, S. C., & Fang, Y. T. (2020). Window detection in facades using heatmap fusion. *Journal of Computer Science and Technology*, *35*, 900-912.

Li, L., Qi, Z., Ma, Q., Gao, W., & Wei, X. (2024). Evolving multi-objective optimization framework for early-stage building design: Improving energy efficiency, daylighting, view quality, and thermal comfort. *Building Simulation*, *17*, 2097-2123).

Liu, L., Wu, D., Li, X., Hou, S., Liu, C., & Jones, P. (2017). Effect of geometric factors on the energy performance of high-rise office towers in Tianjin, China. *Building Simulation*, *10,* 625-641.

Liu, H., Xu, Y., Zhang, J., Zhu, J., Li, Y., & Hoi, S. C. (2020). DeepFacade: A deep learning approach to facade parsing with symmetric loss. *IEEE Transactions on Multimedia*, *22*(12), 3153-3165.

Liu, S., Zeng, Z., Ren, T., Li, F., Zhang, H., Yang, J., Su, H., Zhu, J., & Zhang, L. (2025). Grounding DINO: Marrying DINO with grounded pre-training for open-set object detection. In *European Conference on Computer Vision* (pp. 38-55). Springer, Cham.

Loga, T., Stein, B., & Diefenbach, N. (2016). TABULA building typologies in 20 European countries—Making energy-related features of residential building stocks comparable. *Energy and Buildings*, *132*, 4-12.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, *60*, 91-110.

Lu, Y., Wei, W., Li, P., Zhong, T., Nong, Y., & Shi, X. (2023). A deep learning method for building façade parsing utilizing improved SOLOv2 instance segmentation. *Energy and Buildings*, *295*, 113275.

Ma, Y., Deng, W., Xie, J., Heath, T., Xiang, Y., & Hong, Y. (2022a). Generating prototypical residential building geometry models using a new hybrid approach. *Building Simulation, 15,* 17-28.

Ma, W., Xu, S., Ma, W., Zhang, X., & Zha, H. (2022b). Progressive feature learning for facade parsing with occlusions. *IEEE Transactions on Image Processing*, *31*, 2081-2093.

Ma, W., Xu, S., Ma, W., Zhang, X., & Zha, H. (2022c). Github repository: Progressive feature learning for facade parsing with occlusions. [Online]. https://github.com/wohaiyo/PFLNet/tree/main/saves

Mathias, M., Martinović, A., & Van Gool, L. (2016). ATLAS: A three-layered approach to facade parsing. *International Journal of Computer Vision*, *118*, 22-48.

McNeel (2024). Rhino v8 [Software]. https://www.rhino3d.com/

Meta (2024). Mapillary from Meta, street-level images. [Online]. https://www.mapillary.com/

Microsoft (2024). Bing Maps 'Bird's Eye' aerial imagery for Turin. [Online]. https://www.bing.com/maps/

Murano, G., Primo, E., & Corrado, V. (2018). The effect of glazing on nZEB performance. *Energy Procedia*, *148*, 320-327.

Nordmark, N., & Ayenew, M. (2021). Window detection in facade imagery: a deep learning approach using mask R-CNN. *arXiv preprint arXiv:2107.10006*.

Ohene, E., Krarti, M., Chan, A. P., Hsu, S. C., & Ansah, M. K. (2024). Optimal design guidelines for net zero energy residential buildings in cooling-dominated climates: Case study of Ghana. *Building and Environment*, 111685.

OpenStreetMap (OSM) Contributors (2024). OpenStreetMap [Northwest Italy Dataset]. OpenStreetMap Foundation. Available as open data under the Open Data Commons Open Database License at http://www.openstreetmap.org.

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 779-788).Reinhart, C., Dogan, T., Jakubiec, A., Rakha, T., & Sang, A. (2013). UMI–an urban simulation environment for building energy use, daylighting and walkability. In *Building Simulation 2013* (Vol. 13, pp. 476-483). IBPSA.

Reinhart, C. F., & Davila, C. C. (2016). Urban building energy modeling–A review of a nascent field. *Building and Environment*, *97*, 196-202.

Riemenschneider, H., Krispel, U., Thaller, W., Donoser, M., Havemann, S., Fellner, D., & Bischof, H. (2012, June). Irregular lattices for complex shape grammar facade parsing. In *2012 IEEE conference on computer vision and pattern recognition* (pp. 1640-1647). IEEE.

Robinson, D., Haldi, F., Leroux, P., Perez, D., Rasheed, A., & Wilke, U. (2009). CitySim: Comprehensive micro-simulation of resource flows for sustainable urban planning. In *Proceedings of the Eleventh International IBPSA Conference* (pp. 1083-1090).

Roboflow. (2024). *Roboflow Annotate*. [Online]. https://roboflow.com/annotate

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18 (pp. 234-241). Springer international publishing.

Russell, B. C., Torralba, A., Murphy, K. P., & Freeman, W. T. (2008). LabelMe: a database and web-based tool for image annotation. *International journal of computer vision*, *77*, 157-173.

Sadhukhan, D., Peri, S., Sugunaraj, N., Biswas, A., Selvaraj, D. F., Koiner, K., Rosener, A., Dunlevy, M., Goveas, N., Flynn, D., & Ranganathan, P. (2020). Estimating surface temperature from thermal imagery of buildings for accurate thermal transmittance (U-value): A machine learning perspective. *Journal of Building Engineering*, 32, 101637.

Samuelson, H., Claussnitzer, S., Goyal, A., Chen, Y., & Romo-Castillo, A. (2016). Parametric energy simulation in early design: High-rise residential buildings in urban contexts. *Building and Environment*, *101*, 19-31.

Sezen, G., Cakir, M., Atik, M. E., & Duran, Z. (2022). Deep learning-based door and window detection from building façade. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *43*, 315-320.

Sun, Y., Malihi, S., Li, H., & Maboudi, M. (2022). DeepWindows: Windows instance segmentation through an improved mask R-CNN using spatial attention and relation modules. *ISPRS International Journal of Geo-Information*, *11*(3), 162.

Suppa, A. R., & Ballarini, I. (2023). Supporting climate-neutral cities with urban energy modeling: a review of building retrofit scenarios, focused on decision-making, energy and environmental performance, and cost. *Sustainable Cities and Society*, 104832.

Suppa, A. R., & Corrado, V. (2023). Supporting the EU mission “100-climate neutral cities”: using urban building energy modeling for zero-emission building retrofit scenarios at district scale. In *Proceedings of Building Simulation 2023*, 18th Conference of IBPSA, pp. 2953-2960).

Szcześniak, J. T., Ang, Y. Q., Letellier-Duchesne, S., & Reinhart, C. F. (2022). A method for using street view imagery to auto-extract window-to-wall ratios and its relevance for urban-level daylighting and energy simulations. *Building and Environment*, *207*, 108108.

Tarkhan, N., Letellier-Duchesne, S., & Reinhart, C. (2022). Capturing façade diversity in urban settings using an automated window to wall ratio extraction and detection workflow. In *2022 Annual Modeling and Simulation Conference (ANNSIM)* (pp. 706-717). IEEE.

Tarkhan, N., Szcześniak, J. T., & Reinhart, C. (2024). Façade feature extraction for urban performance assessments: Evaluating algorithm applicability across diverse building morphologies. *Sustainable Cities and Society*, *105*, 105280.

Teboul, O., Kokkinos, I., Simon, L., Koutsourakis, P., & Paragios, N. (2011). Shape grammar parsing via reinforcement learning. In *CVPR 2011* (pp. 2273-2280). IEEE.

Tyleček, R. (2013). The CMP façade database (Version 1.1). Research Reports of CMP, Czech Technical University in Prague, No. 24, 2012.

Ultralytics. (2020). *YOLOv5 Documentation*. [Online]. https://docs.ultralytics.com/models/yolov5/

Ultralytics. (2024). *YOLOv9 Documentation*. [Online]. https://docs.ultralytics.com/

Veillette, D., Rouleau, J., & Gosselin, L. (2021). Impact of window-to-wall ratio on heating demand and thermal comfort when considering a variety of occupant behavior profiles. *Frontiers in Sustainable Cities*, *3*, 700794.

Wang, X., Zhang, R., Kong, T., Li, L., & Shen, C. (2020). Solov2: Dynamic and fast instance segmentation. *Advances in Neural information processing systems*, *33*, 17721-17732.

Wang, S., Kang, Q., She, R., Tay, W. P., Navarro, D. N., & Hartmannsgruber, A. (2022). Building facade parsing R-CNN. *arXiv preprint arXiv:2205.05912*.

Wen, R. (2019). *Google-Streetview* (1.2.9). [Programming library]. https://pypi.org/project/google-streetview/

Whitaker, J. (2023). *Py-proj (3.6.1).* [Programming library]. https://pypi.org/project/pyproj/

Wu, X., Sahoo, D., & Hoi, S. C. (2020). Recent advances in deep learning for object detection. *Neurocomputing*, *396*, 39-64.

Xu, C., Chen, S., Ren, H., Xu, C., Li, G., Li, T., & Sun, Y. (2025). A novel deep learning and GIS integrated method for accurate city-scale assessment of building facade solar energy potential. *Applied Energy*, *387*, 125600.

Appendix

A1 – Qualitative image assessment criteria

Table A1. Assessment criteria for the Train/Val and Inference sets. * = the GSV images could not be recreated; instead, an online example is linked.

Criterion	Description	Example
Bad capture – camera	The camera position that GSV returned was not as expected at GIS stage; instead, a camera closer to the façade provides an image with an incorrect perspective.	
Bad capture – proximity	Images are captured too far away to be clearly visible or too close to capture the building.	Example at GSV online: lat/lon = 45.0987842/7.6986704, heading = 122º, image date = November 2022 *
Bad capture – orthogonality *(Train/Val set only)*	Alignment of 2nd level glazing exceeds ±2.5º from orthogonal, as measured in the annotation stage, typically due to discrepancies between OSM and GSV geometry.	
Bad capture – angle *(Inference set only)*	The façade is clearly not orthogonal. Similar to above, but the Inference set is not annotated/measured. This example was rejected from the Inference set and subsequently measured at 6.1º deviation from orthogonal.	
Blurriness	The image is blurred such that more than 25% of glazing objects are not visible. During annotations, it was verified that images are blurred in the online version of GSV.	
Not usable – other	Other reasons not captured above or below render the image unusable, including distortion from extraction/stitching, if the building is non-residential, etc.	Example at GSV online: lat/lon = 45.0976523,7.7277884, heading = 130º, image date = May 2023 *
Obstructions – per sub-categories below:	*The image is obstructed such that one or more vertical stacks of windows at the 2nd or 3rd level are non-visible.*	
Obstructed by landscaping	Vegetation or fencing.	Example at GSV online: lat/lon = 45.0947041,7.7134523, heading = 283º, image date = June 2022 *

Obstructed by window coverings	External drapery/shades or solarium glazing.	
Obstructed by construction equipment	Scaffolding or other construction equipment.	Example at GSV online: lat/lon = 45.0928449,7.7065962, heading = 122º, image date = June 2022 *
Obstructed at balconies	Where deep inset balconies or opaque balcony guards make glazing completely non-visible.	

A2 – Rhino script for annotation XML files

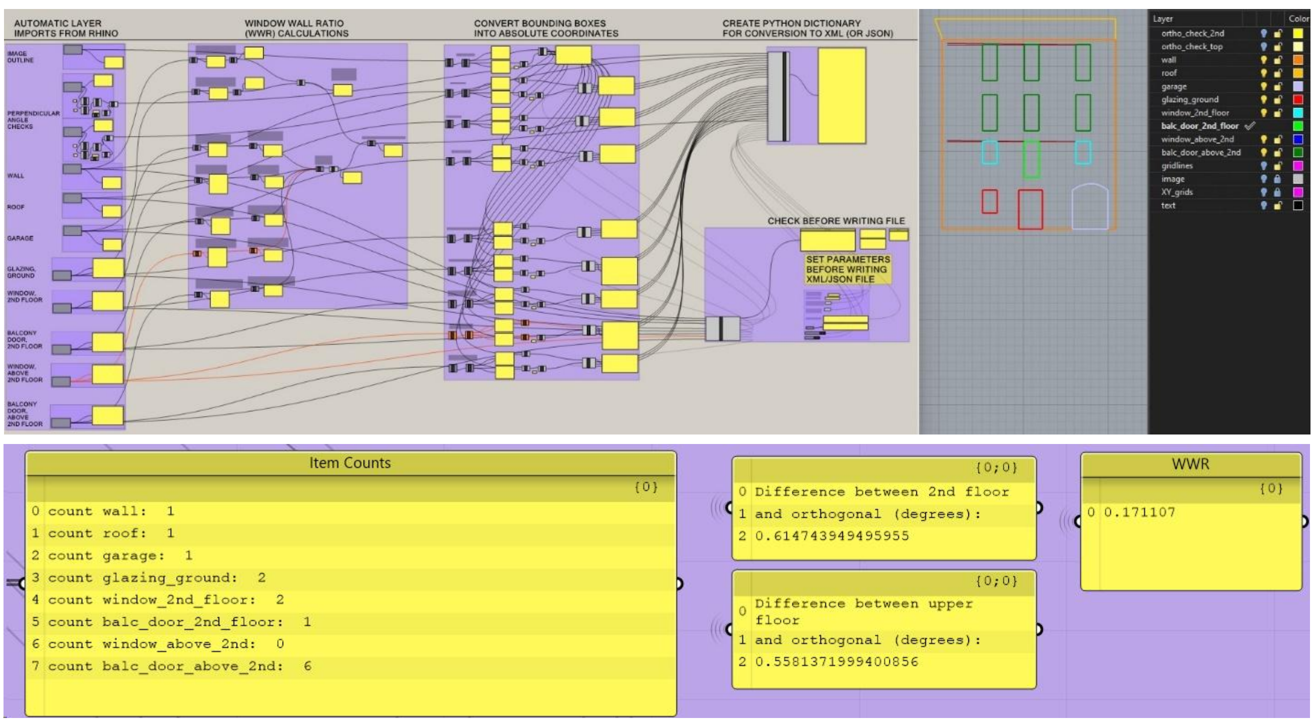


Figure A2. Top: split screen with Grasshopper script (left) which contains visual programming tools to convert Rhino geometry (right) to XML. Bottom: the print-out window for error checking.

A3 – Deep learning metrics

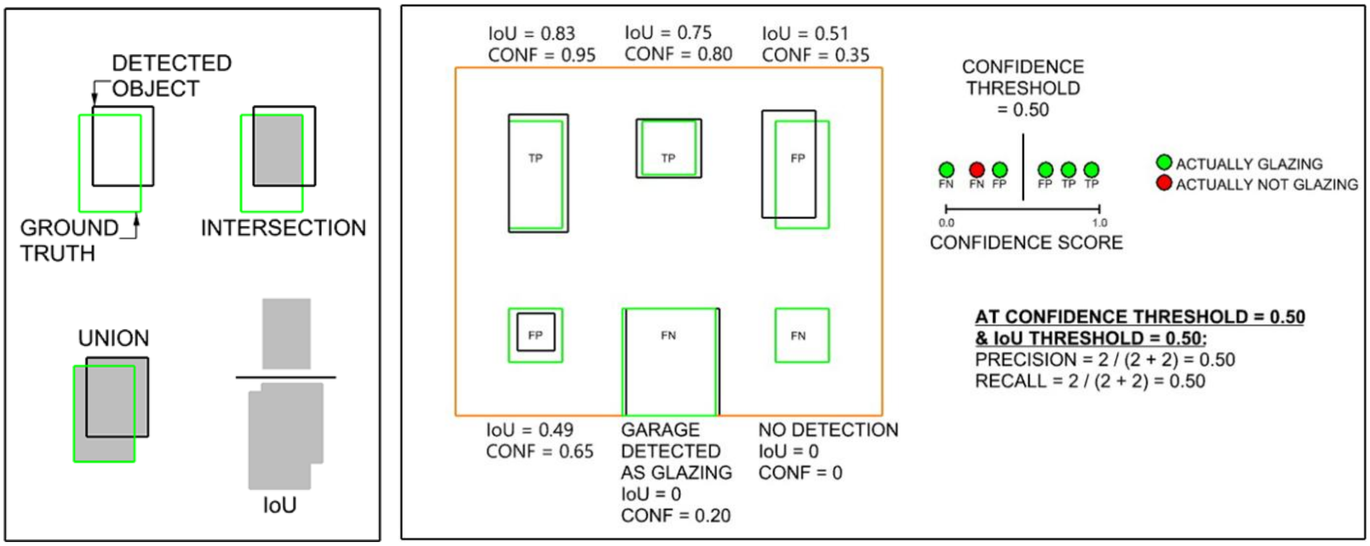


Figure A3. Left: Intersection over union (IoU). Right: Example calculation of IoU, precision, and recall. Ground-truth objects are green and detections are black. Results are arranged in ascending confidence and precision/recall are calculated. Authors' elaboration following Big Vision (2022) and Google (2024b).

A4 – Deep learning hyperparameters

Table A4. Hyperparameters used during model training.

Training hyperparameter	Value
Batch size	32
Image resizing	640x640 pixels
Optimizer	Stochastic Gradient Descent
Momentum	0.937
Weight decay	0.0008
Learning rate – initial	0.0005
Learning rate – final	0.000025
Epochs	200
Early stopping callbacks	If mAP50-95 has not improved for 10 epochs
Augmentations – default in YOLOv9 model	Random brightness contrast; random gamma; image compression; random tone curve; hue saturation value jittering; random rotation; random translation, random scaling, random horizontal flipping; random erasing
Augmentations – custom additions in this work	For each batch, add one of: gaussian blur, CLAHE, defocus, or zoom blur.