scientific reports

OPEN

Domain adaptation using transformer models for automated detection of exterior cladding materials in street view images

Seunghyeon Wang

Recent advances in deep learning have achieved impressive accuracy in various building analysis tasks using street view imagery). However, a major challenge lies in the large-scale, labeled datasets typically required—an obstacle driven by limited raw data access and labor-intensive annotations. To overcome this, the present study introduces a domain adaptation (DA) framework for classifying exterior cladding materials. Six categories are targeted: Brick, Concrete, Glass, Stone, Mixed, and Others. A fully labeled dataset from Scotland and a partially labeled dataset from London form the basis of the approach, which leverages transformer-based architectures, data augmentation, and hyperparameter optimization to boost accuracy. In evaluations on unseen data, an axial transformer trained with augmented data and optimized hyperparameters emerged as most effective, achieving class-specific accuracies of 88.43% (Brick), 73.71% (Concrete), 68.67% (Glass), 91.33% (Stone), 86.65% (Mixed), and 83.46% (Others), culminating in an overall accuracy of 82.04%. These findings illustrate the potential of the DA-based method to maintain strong performance, with further refinements suggested for future work. The paper subsequently explores additional applications of this proposed strategy.

Keywords Exterior cladding material, Street view images, Google street view images, Image augmentation, Domain adaptation, Transformer-based architecture

Exterior cladding materials—such as brick, concrete, and paneling—form the building's outermost protective layer. Their selection often depends on cost, local climate, personal preferences (e.g., texture and color palette), and overall durability with respect to maintenance[1].

Because of these wide-ranging variables, numerous studies have examined the distribution of cladding materials at both urban and national scales[2,3]. Understanding how these materials are employed is vital for multiple applications. For instance, urban heat island analyses depend on recognizing that each cladding type exhibits distinct thermal properties affecting heat retention[4]. Similarly, in post-earthquake scenarios, differences in cladding choices can significantly influence both the extent and nature of observed damage[5].

Among current methods for examining exterior cladding, street view imagery (SVI) stands out as an efficient and cost-effective alternative to on-site inspections, providing a direct view of building façades for focused observations[6]. In particular, google street view (GSV)—a widely adopted SVI platform—spans over 16 million kilometers across 83 countries. By simply entering geographic coordinates, users can readily access these images, making GSV indispensable for large-scale endeavors such as urban planning or nationwide surveys[7].

Despite its accessibility, analyzing SVI to extract exterior cladding materials can be both labor-intensive and time-consuming, especially over large geographical areas. Typical deep learning can automate much of this manual image processing by training a model on large sets of labeled images that capture varied conditions (e.g., differing illumination or scale), thereby exhibiting robust generalization[8]. However, two primary challenges arise when applying exterior cladding material classification to SVI data: (1) acquiring a sufficiently comprehensive set of images (e.g., abundant examples of brick but fewer examples of stone) under diverse real-world conditions, and (2) labeling this large volume of images, which is both tedious and resource-intensive.

A potential solution involves leveraging existing, well-labeled source datasets (i.e., from regions where data is readily available) and applying them to a different target domain (i.e., where labeled data are scarce). However, these source datasets may differ substantially from the target domain in terms of their underlying data

Institute for Environmental Design and Engineering, University College London, London WC1H 0NN, UK. email: seung-hyun.wang@ucl.ac.uk

distributions—a phenomenon known as domain shift. For instance, architectural styles vary among London, Scotland, and South Korea due to distinct cultural contexts and regional building practices. While London and Scotland share certain architectural similarities, South Korea's style diverges more noticeably[6,9]. This contrast becomes evident when visualizing their distributions using t-distributed Stochastic Neighbor Embedding (t-SNE): London and Scotland's data points cluster more closely together, whereas South Korea's points lie farther away, as illustrated in Fig. 1. Such differences—including materials, design principles, and aesthetic nuances—can degrade model performance when a model trained on one region is applied to another.

In this paper, DA techniques are introduced to have the reliable accuracy of deep learning models for extracting the information of exterior cladding materials from SVI. The main contributions are as follows:

1. A DA strategy employing DANN is proposed for robust feature alignment between source and target domains.
2. Five distinct transformer architectures—including Swin Transformer and Axial Transformer—are assessed to determine their suitability as feature extractors.
3. Various hyperparameters are systematically investigated to understand their effect on model performance.
4. An in-depth examination identifies the top-performing model within each architecture–dataset configuration.
5. Both supervised learning and DA methods are evaluated under domain shift conditions.

Literature review

Existing studies that apply supervised learning to building façade image analysis, as well as work on domain adaptation in related contexts, are examined here to clarify the research gap.

Supervised learning

Supervised learning—often termed general deep learning—is widely applied to labeled datasets, enabling models to learn from annotated examples. Within building façade image analysis, numerous studies have utilized SVI to extract the useful information including exterior cladding materials. For instance, Ilic et al.[12] employed Convolutional Neural Networks (CNNs) to detect gentrification-related changes in sequential GSV images, achieving 95.6% accuracy in Ottawa. Hu et al.[13] focused on classifying urban geometry using GSV and Densely Connected Convolutional Networks (DenseNets), reporting 89.3%, 86.6%, and 86.1% accuracies across three Hong Kong regions. Meanwhile, Campbell et al.[14] explored deep learning for autonomous traffic sign detection in GSV images from the City of Greater Geelong, attaining 95.63% accuracy using Single Shot Detection (SSD)

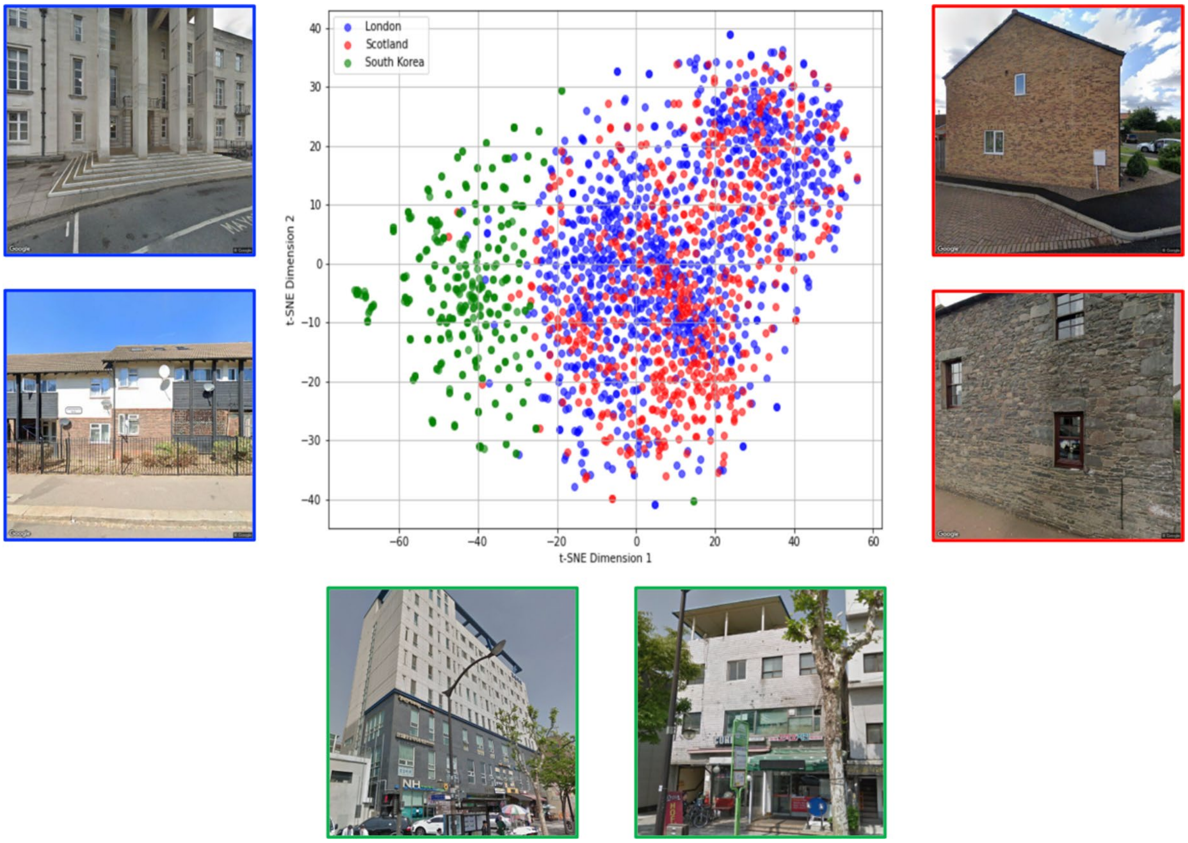


Fig. 1. Visualization of domain shift in three datasets.

with MobileNet. Similarly, Yan and Ryu[15] utilized GSV images and a CNN-based approach to categorize nine crop types (e.g., corn, soybean), reaching 92% accuracy in California's Central Valley and 97% in Illinois.

In another study, Zou and Wang[16] proposed a CNN-VGG16-based technique to identify abandoned houses from GSV data, recording 84% accuracy in five Rust Belt regions. Kim et al.[17] introduced a ResNet-based method to differentiate wooden and metal poles, achieving 97.5% accuracy in Texas. Likewise, Kalfarisi et al.[18] developed a large-scale solution for detecting soft-story buildings via Faster R-CNN with ResNet-50 and Inception-V2, recording 88% accuracy in both San Bruno and Seattle. Finally, Wang and Han[6] proposed an automated system to classify exterior cladding materials using deep learning and GSV images; for London and Scotland, their MobileNetV3 model attained average accuracies of 75.65% and 73.45%, respectively.

Application of domain adaptation in other contexts

The distinction between supervised learning and DA is illustrated in Fig. 2 using an example of two classes—brick and concrete—sourced from Scotland (source domain) and London (target domain). In the left panel, where supervised learning is applied, source and target samples of the same class form separate clusters, and the decision boundary fitted on the source domain does not generalize well to the target domain because of domain shift.

In the right panel, where DA is applied, the framework aligns the feature distributions of the two domains so that brick and concrete samples from Scotland and London become more intermixed within each class, and the decision boundary is adjusted to separate both classes consistently across domains. As noted, DA helps address data-availability challenges for both raw and labeled images when classifying exterior cladding materials across different regions.

For DA to function effectively, several conditions generally need to be fulfilled. First, the source and target domains should share the same task and label space, since the labeled data resides only in the source domain, and both domains' representations must map to these labels. Second, the domains should be sufficiently related—if no direct correspondence exists, knowledge transfer is likely to fail[19], and may even degrade performance compared to training without transfer[20,21]. Finally, both domains should have enough data to produce a robust shared representation; heavily skewed or insufficient datasets in either domain often result in low accuracy in both training and testing[22].

Despite these requirements, DA methods can rival or exceed supervised learning while demanding fewer labeled samples. Hong et al.[23] tackled a face detection challenge—using only one passport photo—by employing facial landmark detection to expand the dataset, achieving 97.91% accuracy. Hu et al.[24] bridged the gap between synthetic point clouds from Building Information Modeling (BIM) and real point clouds for semantic segmentation, achieving an average accuracy of 91.03%. Hong et al.[25] tested data from one construction site (source) against three different sites (targets), with DA boosting accuracy from 40.43 to 82.76%. Duan et al.[26] introduced an unsupervised, feature-level DA approach to enhance a reinforcement learning (RL) strategy for cable-in-duct installation, raising a 98% success rate in simulation to 95.8% in real-world conditions. Similarly, Tran et al.[27] demonstrated that adopting DA in worker detection via object detection models at construction sites led to a 93% success rate in real-world applications.

Such work demonstrates that domain adaptation can substantially enhance performance without relying on large amounts of labeled data, particularly in fields requiring robust transfer from one domain to another.

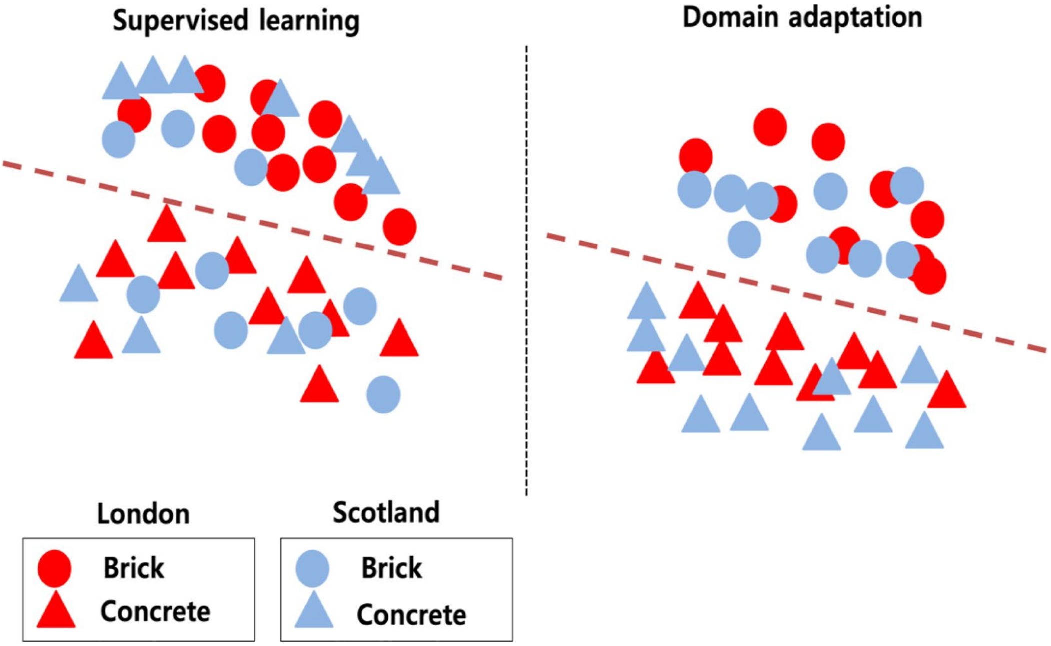


Fig. 2. Visualization of class distributions of source and target domains.

Consequently, if DA techniques are applied to exterior cladding material classification with appropriately designed configurations, high accuracy can be achieved. By leveraging more plentiful datasets from a different source region, domain adaptation boosts model performance while reducing the labeling burden with the collection of raw data for the target domain.

Proposed approach

To accurately classify exterior cladding materials, this study presents a DA) methodology based on DANN, organized into the workflow shown in Fig. 3. The approach comprises six main components—(1) input domain, (2) image augmentation, (3) feature extractor, (4) domain alignment, (5) domain classifier, and (6) label classifier. The subsequent sections provide a detailed explanation of how each component is built and evaluated within the deep learning framework.

Input domain

The input domain consists of all images feeding into the pipeline—whether from the source or target distribution. While both domains may appear generally similar, they can differ significantly in attributes such as visual style and architectural design. The following subsections describe each domain module's function.

Source domain

The source domain module supplies a labeled dataset that underpins supervised training. It provides both images and corresponding class labels (e.g., "brick," "concrete," "stone"), enabling the model to learn how specific features map to each material category. By exposing the feature extractor and label classifier to a sufficiently diverse range of labeled samples, this module ensures a robust foundation for classification. Moreover, the knowledge the model acquires here—such as recognizing common textures or color patterns—is transferred or adapted when encountering the unlabeled target domain.

Target domain

The target domain module contains images that lack reliable labels. Although these images may appear similar to those in the source dataset, they often differ in architecture, context, or lighting, leading to a domain shift[28]. The model, initially trained on the source domain's labeled data, uses DA techniques to accommodate these unlabeled target images. By realigning feature distributions between source and target, DA greatly reduces the need for extensive new labeling in the target environment. Consequently, the model continues to exhibit strong classification performance, even in large-scale or dynamically changing conditions where assembling a fully labeled dataset would be impractical.

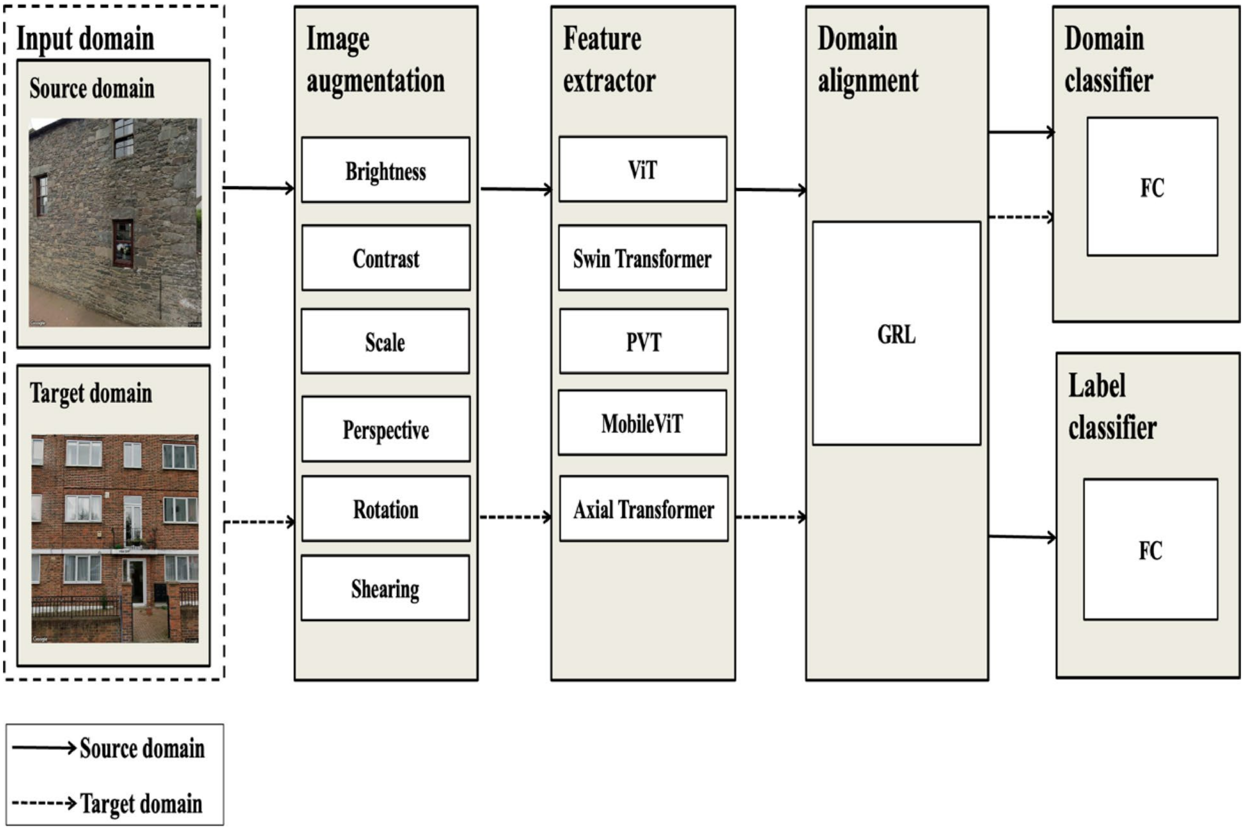


Fig. 3. Overall framework of proposed method based on DA.

Image augmentation

Constructing comprehensive datasets for deep learning typically demands substantial time and effort. To alleviate these challenges, image augmentation is employed to artificially expand the training set. In this study, six different augmentation techniques are utilized, each addressing specific issues as outlined in Table 1. A more detailed, mathematical discussion of these methods is available in prior work[29,30].

Since SVI data are captured outdoors, lighting conditions often fluctuate with factors like weather or time of day. Brightness augmentation replicates such variations by adjusting the light intensity in images, producing outputs that range from brighter to darker[31]. Contrast augmentation, on the other hand, increases luminance contrast, making object features more prominent—particularly useful for emphasizing contours and edges in architectural elements[32].

Since SVI images are taken from diverse angles, perspective transformation modifies the homography matrix so that buildings can be recognized regardless of the viewing angle. Scale augmentation, an affine transformation, replicates varying distances and accommodates actual size differences among buildings. Likewise, building façades may appear misaligned if the camera is not level; rotation augmentation compensates for this, supporting accurate detection of features irrespective of orientation. Finally, shear augmentation addresses distortions caused by an off-perpendicular camera angle, simulating a skew along a chosen axis[33].

Feature extractor

The feature extractor, also known as the encoder or backbone model, converts raw inputs into higher-level, more abstract representations. Rather than relying on raw pixel intensities, the model exploits these extracted features—patterns or attributes that are more stable indicators of content[34–36]. In domain adaptation, a single feature extractor is typically shared by both source and target data, enabling the model to learn a representation space that, ideally, generalizes across domains.

Building on prior work, this study integrates several transformer-based architectures within the domain adaptation framework—namely ViT, Swin Transformer, PVT, MobileViT, and Axial Transformer—chosen for their potential to deliver reliable accuracy in many studies[37–40]. Figures for each architecture are presented to illustrate their unique structures. Readers seeking deeper methodological insights into the components of each model are referred to the comprehensive description in[41]. The subsequent subsections offer a concise overview of the fundamental concepts behind each architecture.

ViT

Vision Transformer (ViT) employs a pure transformer design for image recognition by partitioning images into fixed-size patches and treating each patch as a token (Fig. 4). A key advantage lies in applying self-attention directly to these patch sequences, which allows the model to capture global dependencies across the entire image. This contrasts with traditional CNNs that focus on local receptive fields and often require multiple layers to aggregate global context. However, ViT generally demands large-scale datasets and significant computational resources, owing to the quadratic complexity of self-attention with respect to the number of patches.

Swin transformer

Swin Transformer introduces a hierarchical architecture featuring shifted windows for self-attention (Fig. 5). The image is divided into non-overlapping local windows, and these window partitions shift between layers to capture cross-window interactions. This design yields a linear scaling in attention complexity with increasing image size, making it suitable for high-resolution data. The hierarchical nature also fosters multi-scale feature extraction, proving effective for tasks that require both local details and global context[42].

PVT

PVT (Pyramid Vision Transformer) integrates pyramidal feature extraction—common in CNN-based models—into a transformer framework (Fig. 6). It leverages spatial-reduction attention to downsample image dimensions in the attention layers, thus reducing the cost of processing large inputs[43]. By producing multi-scale feature maps via a pyramid structure, PVT effectively captures features at multiple resolutions, lending itself well to dense prediction tasks such as classification or detection. This structure preserves the global attention benefits of transformers while offering computational efficiency akin to CNNs[44].

Mobilevit

MobileViT merges transformer-based self-attention with lightweight convolutional layers specifically designed for mobile and edge scenarios (Fig. 7). Its defining characteristic is its balanced approach: it harnesses the

Augmentation techniques	Parameters	Ranges of parameter values
Brightness	β	[-30, 30]
Contrast	α	[0.5, 2.0]
Scale	x, y	[0.8, 1.2], [0.8, 1.2]
Perspective	$a_{31}, a_{32},$	[0.01, 0.15]
Rotation	θ	[−25°, 25°]
Shearing	sh_x, sh_y	[−15°, 15°]

Table 1. The used augmentation techniques with parameters and its values.

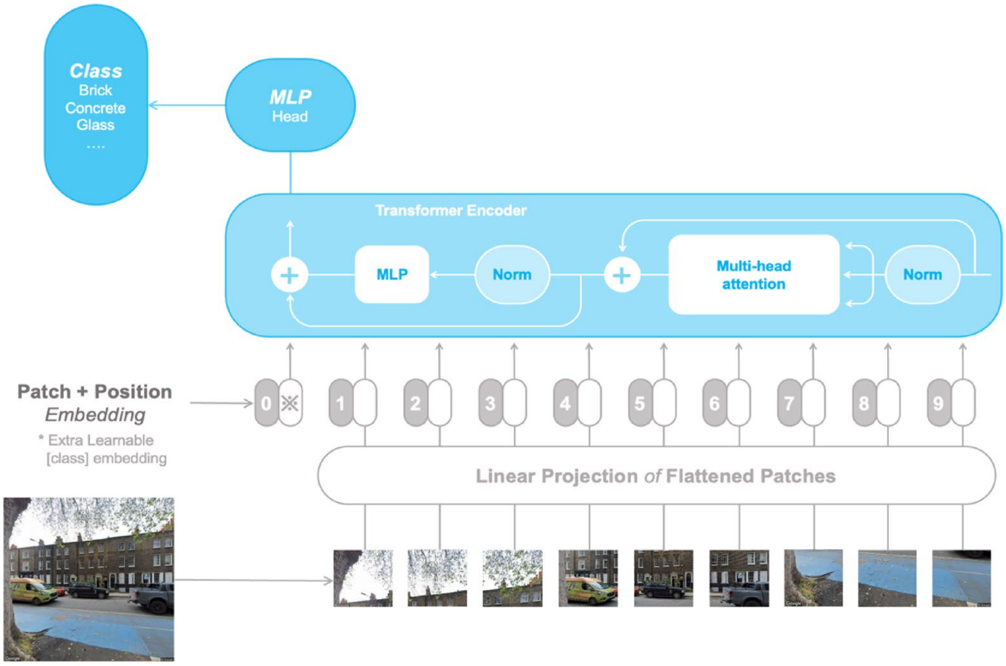

Class
Brick
Concrete
Glass
....
MLP
Head
Transformer Encoder
MLP
Norm
Multi-head attention
Norm
Patch + Position Embedding
* Extra Learnable [class] embedding
0
1
2
3
4
5
6
7
8
9
Linear Projection of Flattened Patches

Fig. 4. The structure of ViT.

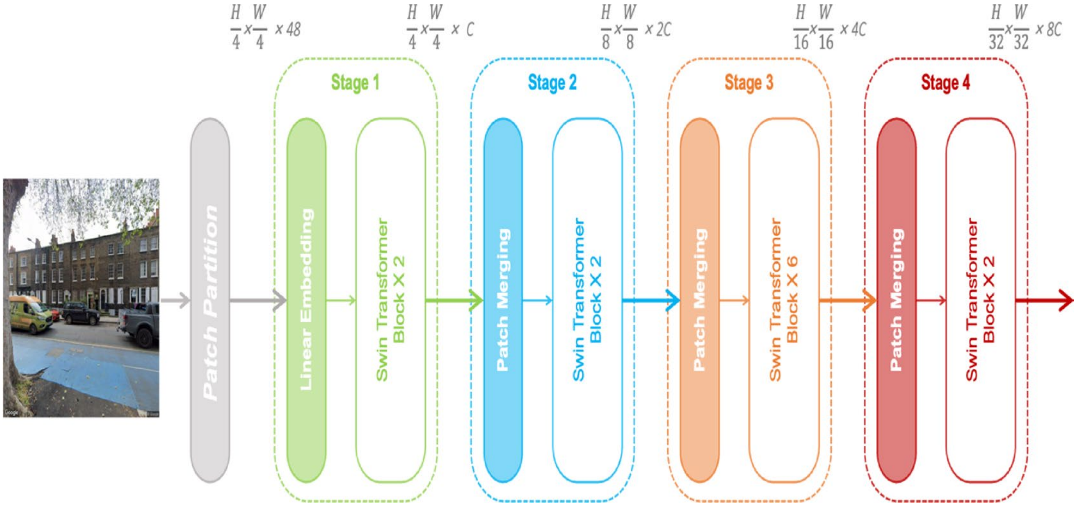

$\frac{H}{4} \times \frac{W}{4} \times 48$
$\frac{H}{4} \times \frac{W}{4} \times C$
$\frac{H}{8} \times \frac{W}{8} \times 2C$
$\frac{H}{16} \times \frac{W}{16} \times 4C$
$\frac{H}{32} \times \frac{W}{32} \times 8C$
Stage 1
Stage 2
Stage 3
Stage 4
Patch Partition
Linear Embedding
Swin Transformer Block X 2
Patch Merging
Swin Transformer Block X 2
Patch Merging
Swin Transformer Block X 6
Patch Merging
Swin Transformer Block X 2

Fig. 5. The structure of Swin Transformer.

power of global context provided by self-attention, while remaining resource-efficient by adopting mobile-friendly architectural blocks[45]. Local features are captured through convolutions, whereas self-attention handles long-range dependencies without substantially increasing parameters or computational load[46]. Consequently, MobileViT suits applications requiring both high performance and limited hardware resources.

Axial transformer

Axial transformer decomposes the standard 2D self-attention operation into two sequential 1D attentions along an image's height and width dimensions (Fig. 8). This axial split reduces the complexity of self-attention from quadratic to linear in terms of spatial size, enabling efficient processing of high-resolution data[47,48]. By applying attention along each axis independently, axial transformer preserves the ability to capture long-range dependencies while maintaining manageable computational demands.

Domain alignment

Even after feature extraction, a domain gap can persist between source and target data distributions. Domain adaptation addresses this issue by introducing alignment mechanisms. In adversarial methods, a domain

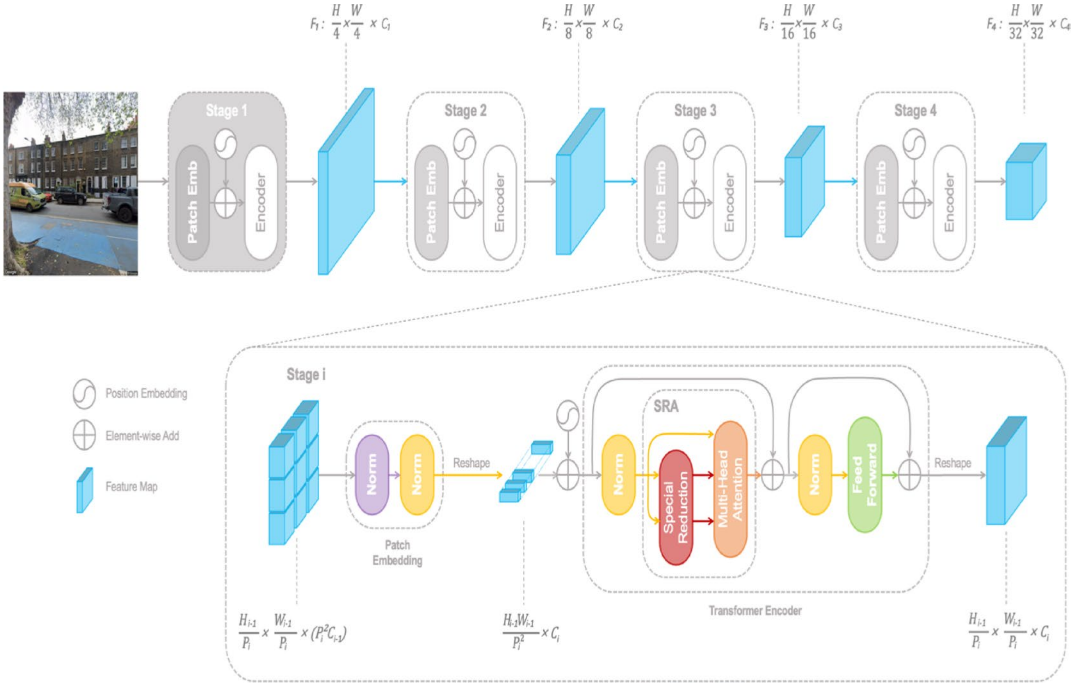

$F_1: \frac{H}{4} \times \frac{W}{4} \times C_1$
$F_2: \frac{H}{8} \times \frac{W}{8} \times C_2$
$F_3: \frac{H}{16} \times \frac{W}{16} \times C_3$
$F_4: \frac{H}{32} \times \frac{W}{32} \times C_4$
Stage 1
Stage 2
Stage 3
Stage 4
Patch Emb
Encoder
Position Embedding
Element-wise Add
Feature Map
Stage i
Norm
Norm
Patch Embedding
Reshape
SRA
Norm
Special Reduction
Multi-Head Attention
Norm
Feed Forward
Reshape
Transformer Encoder
$\frac{H_{i-1}}{P_i} \times \frac{W_{i-1}}{P_i} \times (P_i^2 C_{i-1})$
$\frac{H_{i-1} W_{i-1}}{P_i^2} \times C_i$
$\frac{H_{i-1}}{P_i} \times \frac{W_{i-1}}{P_i} \times C_i$

Fig. 6. The structure of PVT.

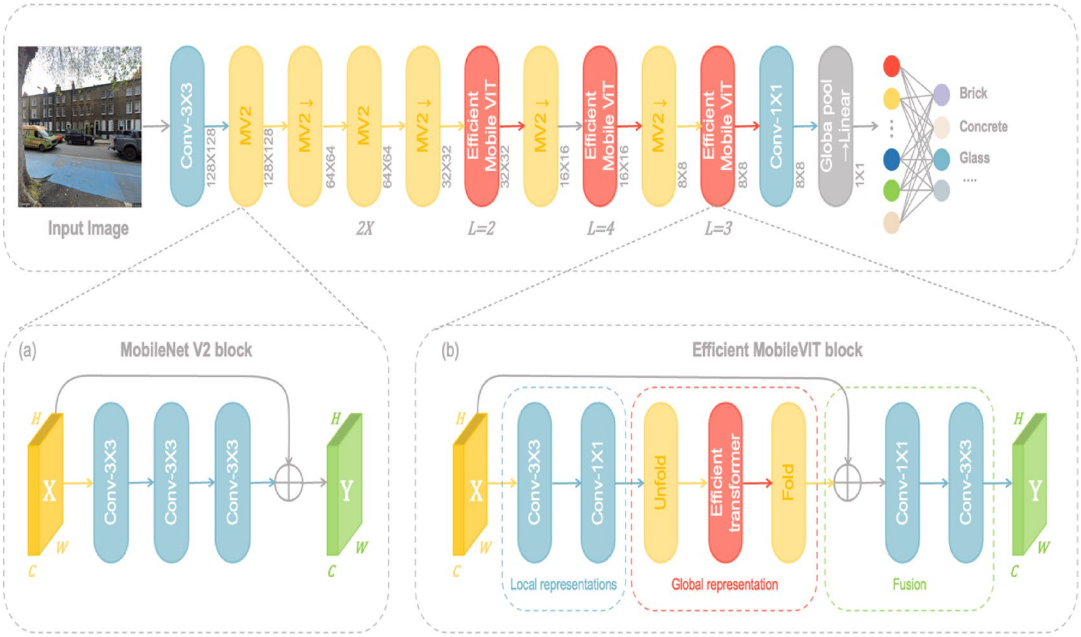


Fig. 7. The structure of MobileViT.

discriminator attempts to distinguish whether extracted features originate from the source or target domain, while the feature extractor aims to confuse the discriminator—forcing the two distributions to appear more similar.

One way to implement adversarial training is the gradient reversal layer (GRL) in DANN[49]. During forward propagation, the GRL passes features to the domain discriminator as usual. However, in backpropagation, it reverses the gradients before they reach the feature extractor. This effectively compels the feature extractor to minimize the discriminator's accuracy, encouraging domain-invariant representations. Over time, this adversarial interplay causes source and target features to converge within the latent space, ultimately enhancing generalization without requiring extensive labeling in the target domain[50].

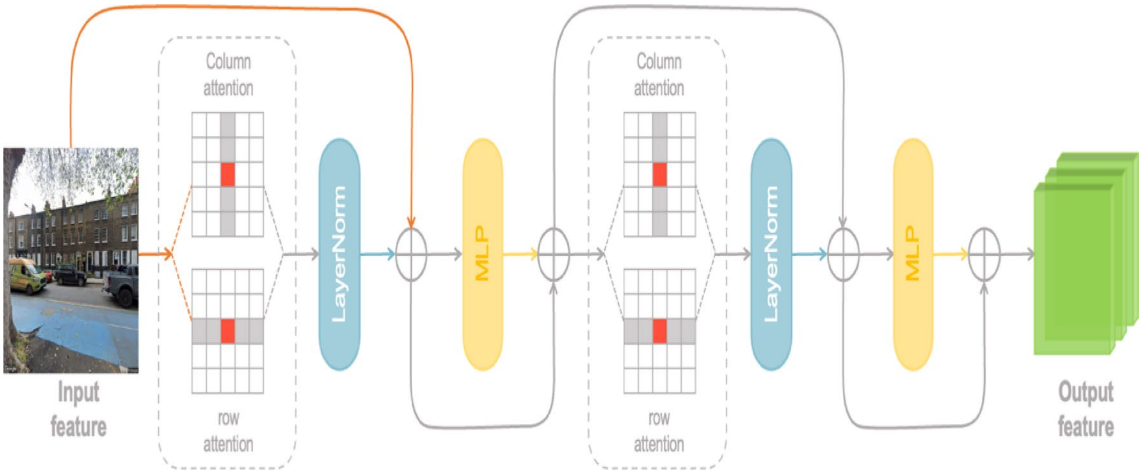


Fig. 8. The structure of axial transformer.

Domain classifier

After feature extraction, a Fully Connected (FC) network is used as the domain classifier. The dimensionality of its input layer matches that of the extracted features to allow seamless data flow. The output layer contains two neurons—corresponding to source and target domains—to reflect the model's confidence in each domain category. The number and size of hidden layers can vary substantially, a choice typically treated as hyperparameter tuning[51]. In this study, the domain classifier is configured with two hidden layers of 32 and 16 neurons each.

Label classifier

In parallel, a label classifier is employed to predict material categories of interest (e.g., "brick," "concrete," "stone") from the aligned features. Like the domain classifier, it also relies on FC layers, whose configuration is determined as part of hyperparameter tuning. In this work, the label classifier has three hidden layers consisting of 32, 16, and 8 neurons, respectively, plus an output layer corresponding to the number of cladding material classes. By making use of the newly aligned features, this classifier more effectively handles unlabeled data from the target domain.

Optimization of the hyperparameters

Although numerous hyperparameters can be tuned to optimize model performance, examining every possible configuration is nearly impossible due to time and computational constraints[52]. Consequently, this study focuses on a select subset of hyperparameters. Stochastic Gradient Descent (SGD) serves as the optimizer, using a batch size of either 1 or 2, which yields highly stochastic (per-sample or near-per-sample) weight updates. Within SGD, the current gradient is combined with scaled gradients from previous iterations using a momentum term, which in this study is set to 0.7 or 0.9.

To mitigate overfitting, a weight decay term (L2 regularization) is added to the loss function for all model weights, tested at 0.0005 or 0.001. The learning rate (step size for weight updates) is assigned either 0.00025 or 0.0001. Because GPU memory limits the batch size, it remains 1 or 2, preserving the intensely stochastic nature of updates. Finally, the number of training iterations—which significantly impacts both precision and training time—is set at 3000 or 5000. Altogether, these five hyperparameters yield 32 unique configurations, as summarized in Table 2.

Evaluation of model performance

Various other metrics can be used to assess a model's performance in image classification. In this study, three specific metrics are selected to evaluate the model.

Underfitting, and overfitting

The evaluation of loss progression throughout iterative training provides ensuring two common issues: underfitting and overfitting. The underfitting is characterized by the persistence of high training and validation loss. Models that underfit, having failed to sufficiently learn from the training data, are likely to deliver high performance on both training and unseen data. In contrast, overfitting manifests as a significant discrepancy between a low training loss and a relatively high validation loss. Models that overfit, despite potentially demonstrating high performance on training data, are prone to failure when exposed to new unseen data[53]. In this study, the models exhibiting signs of underfitting or overfitting were disregarded, as these models are unlikely to yield satisfactory performance on even training or, unseen data, respectively.

F1-score

In image classification, the F1-score is commonly used to evaluate accuracy because it incorporates both precision and recall. Precision represents the proportion of positive predictions that are correct, while recall

	Hyperparameters				
Case	Batch size	Learning rate	Weight decay	Momentum	Iteration
1	1	0.00025	0.0001	0.7	3000
2	1	0.00025	0.0001	0.9	3000
3	1	0.00025	0.0005	0.7	3000
4	1	0.00025	0.0005	0.9	3000
5	1	0.001	0.0001	0.7	3000
6	1	0.001	0.0001	0.9	3000
7	1	0.001	0.0005	0.7	3000
8	1	0.001	0.0005	0.9	3000
9	2	0.00025	0.0001	0.7	3000
10	2	0.00025	0.0001	0.9	3000
11	2	0.00025	0.0005	0.7	3000
12	2	0.00025	0.0005	0.9	3000
13	2	0.001	0.0001	0.7	3000
14	2	0.001	0.0001	0.9	3000
15	2	0.001	0.0005	0.7	3000
16	2	0.001	0.0005	0.9	3000
17	1	0.00025	0.0001	0.7	5000
18	1	0.00025	0.0001	0.9	5000
19	1	0.00025	0.0005	0.7	5000
20	1	0.00025	0.0005	0.9	5000
21	1	0.001	0.0001	0.7	5000
22	1	0.001	0.0001	0.9	5000
23	1	0.001	0.0005	0.7	5000
24	1	0.001	0.0005	0.9	5000
25	2	0.00025	0.0001	0.7	5000
26	2	0.00025	0.0001	0.9	5000
27	2	0.00025	0.0005	0.7	5000
28	2	0.00025	0.0005	0.9	5000
29	2	0.001	0.0001	0.7	5000
30	2	0.001	0.0001	0.9	5000
31	2	0.001	0.0005	0.7	5000
32	2	0.001	0.0005	0.9	5000

Table 2. Combinations of hyperparameters.

reflects the proportion of actual positives that are correctly identified. These metrics are defined mathematically in Eqs. (1) and (2):

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{1}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2}$$

where true positives (TP) denotes the number of correctly identified positive samples, false positives (FP) refers to the number of incorrectly classified positives, and false negatives (FN) represents the number of positive samples the model failed to detect.

The F1-score merges precision and recall using their harmonic mean, making it highly sensitive to low values in either metric. As a result, a model needs sufficiently high precision and recall to achieve an elevated F1-score. Its formulation is provided by Eq. (3):

$$\text{F1-score} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{3}$$

Detection speed

Detection speed represents the time a model needs to process each frame, typically reported as Frames Per Second (FPS). Measuring FPS helps determine how swiftly the classification model can handle incoming images, providing insight into its practical efficiency—especially in scenarios requiring real-time or high-throughput processing.

Experiment

Dataset preparation

In this research, the dataset collected in our previous work[6] using supervised learning was utilized to demonstrate the proposed method. The data collection methodology is briefly described here, with more emphasis placed on showcasing the potential of transformer-based domain adaptation.

Original dataset

For this case study, the United Kingdom was selected because its legal framework permits collecting SVI. In practice, SVI platforms provide large numbers of façade images across many cities, but the distribution of cladding types is uneven and manual annotation is expensive. Rather than repeatedly labeling a full dataset for every new city, an existing labeled dataset from one region can be reused as a source domain and its knowledge transferred to a new, sparsely labeled target region. The experiments therefore employ comparable source and target dataset sizes to control for sample-size effects and to focus on how domain adaptation can mitigate regional shifts in image appearance, while conceptually reflecting the practical need to reduce labeling effort in future deployments.

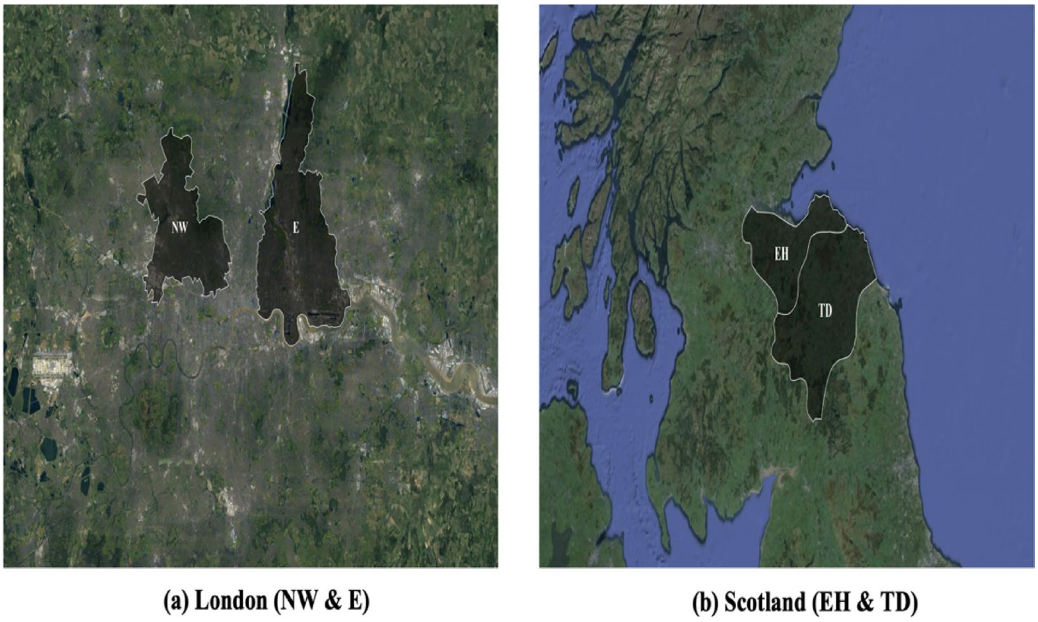
A random sample of 3000 buildings was taken from each of two UK regions—the North-West (NW) and the Eastern (E)—as shown in Fig. 9. The sampling process was carried out via the national mapping agency's website (https://osdatahub.os.uk/). The corresponding addresses were then used to query the GSV Application Programming Interface (API), allowing the retrieval and download of building images.

Various GSV parameters were manually tuned to improve the visual identification of exterior cladding materials. The field of view (FoV), which controls the zoom level or scope of the scene, was set between 10 and 50. The pitch, representing the camera's vertical angle relative to the street-view vehicle, ranged from 25 to 30. Meanwhile, the heading, which specifies the camera's horizontal orientation, varied between 33 and 55. All images were captured at 640 × 640 pixels—the maximum resolution currently provided by GSV. Figures 10 and 11 present representative examples of images taken in London and Scotland, respectively, illustrating typical exterior cladding.

However, many GSV images did not offer a sufficiently clear view for accurate identification. Some showed no building at all, while others were partially obscured by trees or fences, and certain images displayed interiors rather than façades. In some cases, no GSV images were available at the address, only an error message. Figure 12 shows examples of images deemed unusable for this study. These images were discarded through manual review. Of the original 3000 addresses, 1,604 images (53.47%) were deemed usable in London and 1,017 images (53.02%) in Scotland.

Application of image augmentation techniques

Image augmentation parameters directly affect image quality and, consequently, model performance[55,56]. These parameters were selected through a trial-and-error process aimed at generating realistic augmented images, as summarized in Table 1. Brightness was adjusted by randomly shifting pixel intensities (β) between – 30 and + 30, whereas contrast was adjusted by randomly varying α between 0.5 and 2.0. For scale augmentation, images were independently resized along the x- and y-axes to 80–120% of their original size to account for potential differences in building features. Perspective transformations were applied by modifying the homography matrix elements a_{31} and a_{32}, randomly assigning values between 0.01 and 0.15. Rotation was performed within a range



(a) London (NW & E)

(b) Scotland (EH & TD)

Fig. 9. The geographic scope of the dataset.

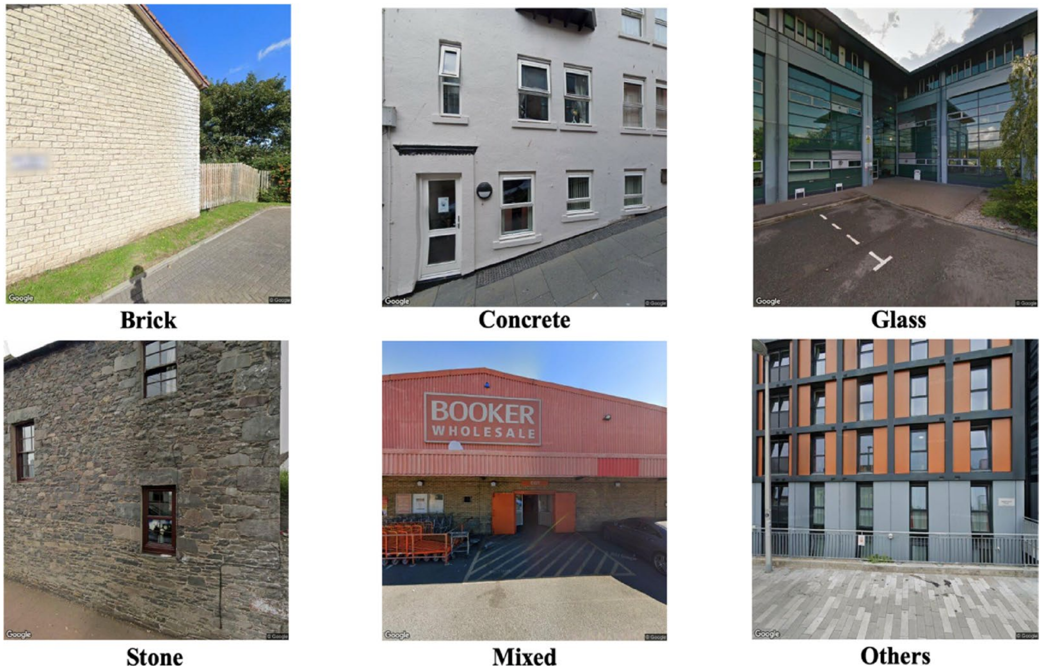


Fig. 10. Examples images in London.

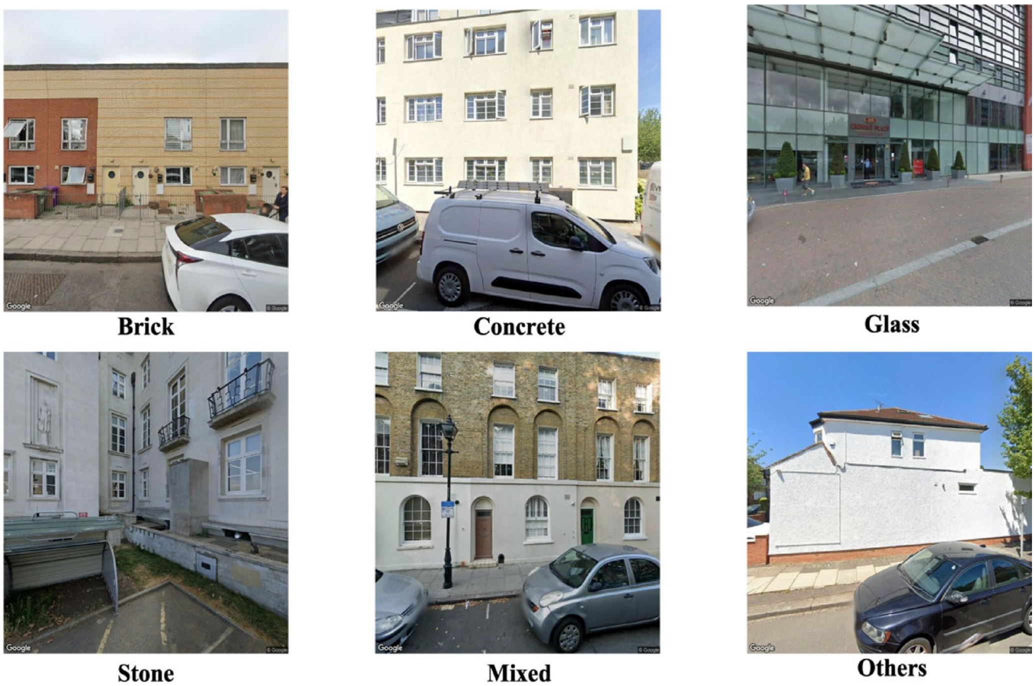


Fig. 11. Examples images in Scotland.

of – 25° to + 25° using the rotation parameter θ, and shearing ranged from – 15° to + 15°, introducing skew via sh_x and sh_y.

These geometric operations—scaling, rotation, perspective shifting, translation, and shearing—can move parts of an image beyond its original boundaries, creating voids or overlapping pixels. Because deep learning models typically require inputs of fixed dimensions, any empty areas were filled with a pixel value of 255 (white), ensuring that all augmented images had uniform size[57]. The original training sets of 928 images for London and 608 images for Scotland were expanded through these augmentation techniques to 5568 and 3648 images, respectively. Including the original (non-augmented) images, the final training sets comprised 6496 images

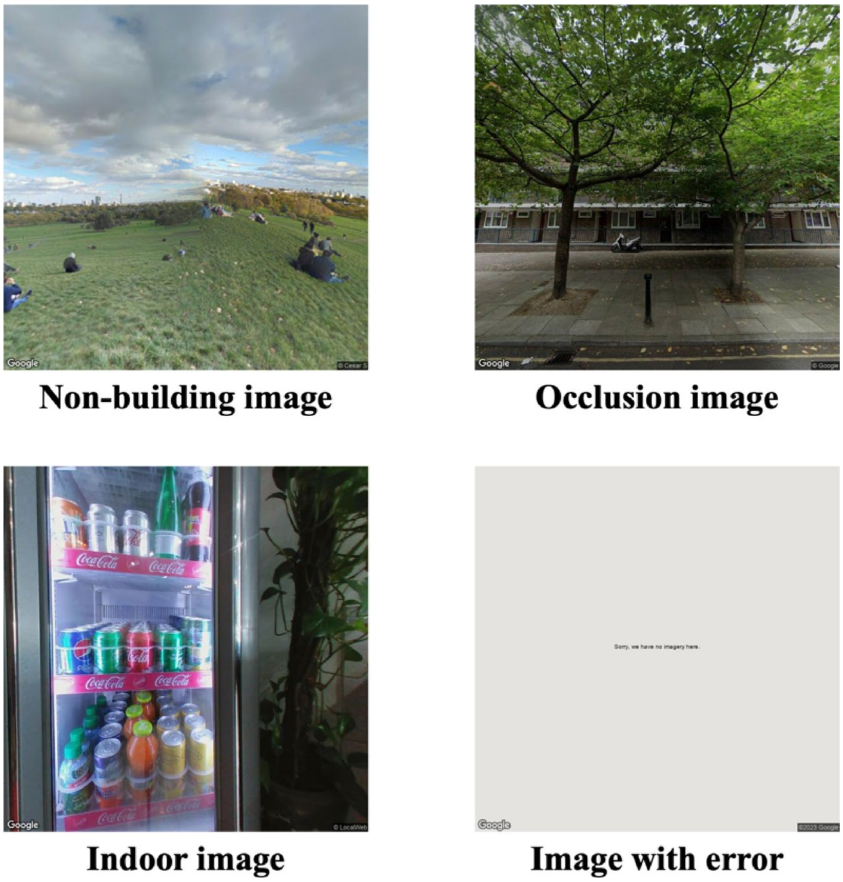


Fig. 12. Examples of unusable images.

for London and 4256 images for Scotland. To demonstrate the effectiveness of the augmented datasets, model accuracy was compared between training with only the original data and training with the augmented data.

Annotation

In image classification, “annotation” (or labeling) refers to assigning each image in the ground truth dataset to a specific category. In this work, images were examined for visual cues indicating the cladding material, and each image was placed into a separate folder—Brick, Concrete, Glass, Stone, Mixed, or Others—corresponding to the relevant cladding class.

Synthesis of final dataset

To validate the proposed methods, four distinct datasets were generated and divided into training, validation, and test sets. Each image was then labeled based on the cladding material category it represented. Table 3 summarizes the annotated distributions for each of these datasets.

Experimental settings

All experiments were performed on a Windows 10 (64-bit) machine equipped with an Intel® Core™ i9-12900 K CPU (5.20 GHz, 16 cores), 64 GB of DDR5 RAM, and an NVIDIA GeForce RTX 3090 GPU featuring 24 GB of GDDR6X VRAM.

Results and discussion

In this study, 640 models were trained, spanning five transformer architectures, four dataset configurations, and 32 hyperparameter settings. For each architecture (ViT, Swin Transformer, PVT, MobileViT, and Axial Transformer) and each dataset configuration (raw source + raw target, augmented source + raw target, raw source + augmented target, and augmented source + augmented Target), all 32 combinations of batch size, learning rate, weight decay, momentum, and number of iterations listed in Table 2 were explored, yielding 5 × 4 × 32 = 640 models.

Underfitting and overfitting

Although the training and validation loss graphs for all these models are omitted here due to space constraints, they can be accessed via a link provided in the “Data Availability” section. Figure 13 shows representative training and validation losses for models trained using augmented source and target datasets. All models

Country	Purpose	Number of images	Class						
			Brick	Concrete	Glass	Stone	Mixed	Others	Total
London	Training	928	265	137	43	24	357	102	928
	Augmentation	5568	1590	822	258	144	2142	612	5568
	Validation	311	91	45	14	8	119	34	311
	Test	311	86	47	15	9	120	34	311
	Total	7118	2032	1051	330	185	2738	782	7118
Scotland	Training	608	46	80	9	174	241	58	608
	Augmentation	3648	276	480	54	1044	1446	348	3648
	Validation	204	15	26	3	61	80	19	204
	Test	205	16	28	4	56	81	20	205
	Total	4665	353	614	70	1335	1848	445	4665

Table 3. Comprehensive dataset distribution in London and Scotland.

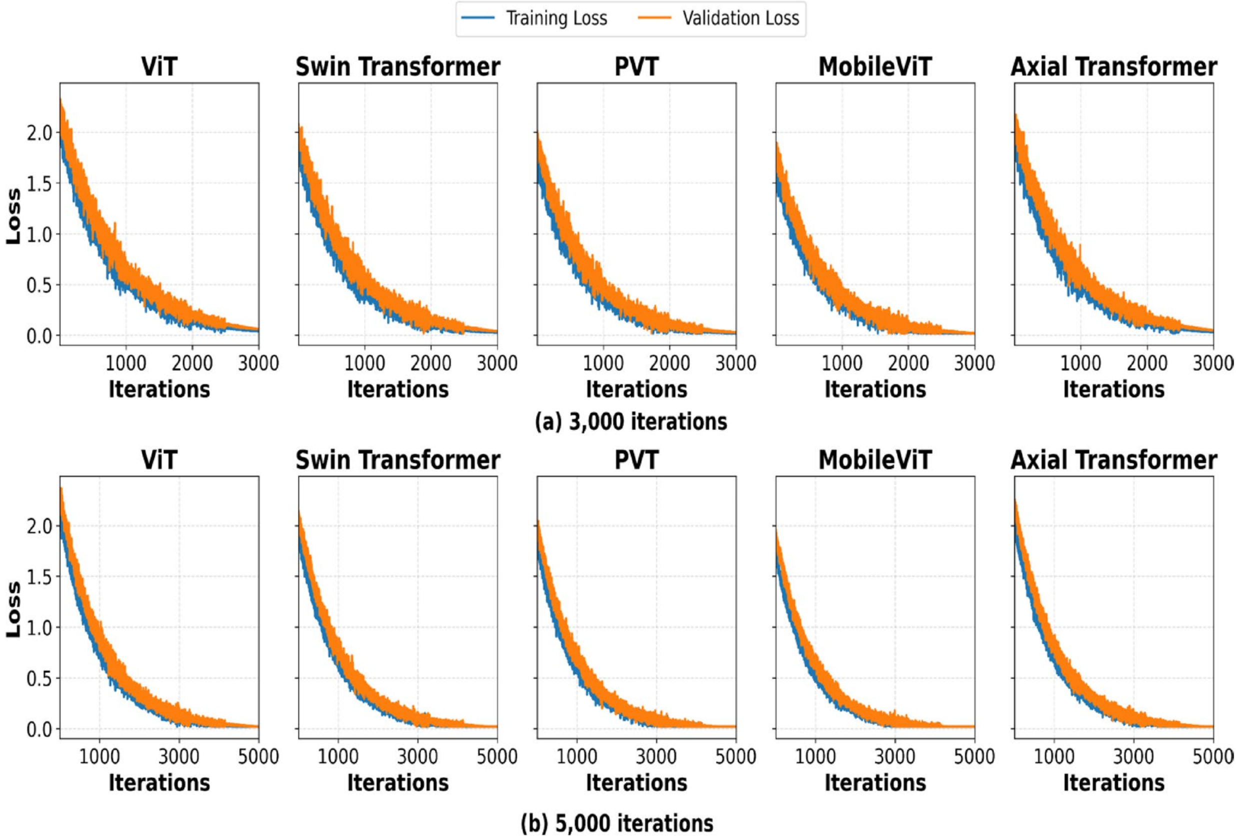


Fig. 13. Training and validation loss graphs.

exhibited a steady reduction in both training and validation losses as training progressed, with occasional fluctuations between epochs rather than a perfectly smooth decline. This behavior indicates that the models effectively learned from the training data, improving validation performance and converging toward an optimal loss value. The consistent drop in training loss across iterations rules out underfitting concerns. Similarly, the parallel decrease and close alignment of training and validation losses suggest that overfitting was not an issue, so no models were excluded on those grounds.

Effectiveness of hyperparameter on model performance

The impact of various hyperparameters on model performance was assessed by clustering results according to each hyperparameter configuration. The datasets considered were Raw Source (RS), Augmented Source (AS), Raw Target (RT), and Augmented Target (AT).

Analysis of the mean variation values in Table 4 shows that the number of training iterations is the most influential hyperparameter in most scenarios. For example, for the Swin Transformer under the “RS + AT” setting, increasing iterations improves the average F1 score by 1.91 points, clearly exceeding the effects of learning rate (0.18), weight decay (0.30), batch size (0.52), and momentum (0.22). Similar behaviour appears for PVT and MobileViT in the "AS + AT" configuration (1.92 and 2.13 points, respectively) and for the axial transformer in "AS + RT" (2.62 points). ViT shows a more nuanced pattern. Although iterations dominate in "AS + RT" (1.96) and "AS + AT" (1.83), weight decay and learning rate become more important in "RS + AT" and "RS + RT," where their variations exceed those of iterations.

Analysis of best performing model in each architecture and dataset

Table 5 reports, for each architecture (ViT, Swin Transformer, PVT, MobileViT, and Axial Transformer) and each source/target configuration, the best model selected from the 32 hyperparameter settings. A clear pattern emerges when comparing configurations that use only raw data with those that include augmented images. Across all architectures, incorporating augmented source and target data consistently increases the average F1-score. For example, the ViT architecture attains an average F1-score of 74.18 in the raw-only setting (608 raw source images and 928 raw target images), but this improves to 81.89 when both source and target sets are augmented (928 raw + 3648 augmented source images and 928 raw + 5568 augmented target images). Swin Transformer, PVT, MobileViT, and Axial Transformer exhibit similar gains when moving from raw-only to augmented conditions.

Beyond the overall performance boost, the per-class results indicate that augmentation particularly benefits more challenging cladding types such as Glass and Stone, which show larger improvements than Brick and Concrete. These findings support the view that domain-adaptation techniques, when combined with augmented training data, help mitigate shifts in data distributions and enhance model robustness and generalization.

Table 6 summarizes the training times for two iteration counts (3000 and 5000) and the detection speeds of the five architectures. As expected, increasing the number of iterations leads to longer training times, while detection speeds remain similar across iteration counts. MobileViT achieves the highest detection speed (approximately 33 FPS), indicating strong potential for real-time or low-latency applications. ViT, Swin Transformer, PVT, and Axial Transformer operate at around 20–25 FPS, which is still adequate for many façade-analysis scenarios.

Model test in unseen dataset

Among the 640 trained models, selection of the final model for evaluation on the unseen London dataset followed a two-stage procedure. First, for each architecture and each of the four dataset configurations, all 32 hyperparameter combinations were trained and evaluated on the validation set. The configuration with the highest validation macro F1-score was chosen as the best model for that architecture–dataset pair.

Second, these 20 best models were compared in terms of both validation macro F1-score and detection speed. Priority was given to higher validation F1-score, while detection speed was used to ensure that the final model remained suitable for practical deployment (i.e., at least 20 FPS).

Based on this procedure, the axial transformer architecture with the configuration corresponding to Case 20—batch size 1, learning rate 0.00025, weight decay 0.0005, momentum 0.9, and 5000 training iterations, trained on augmented source and target datasets—was selected as the final model. As shown in Fig. 14, this

		Hyperparameter				
Architecture	Dataset	Batch size	Learning rate	Weight decay	Momentum	Iteration
ViT	RS + RT	0.17	0.23	0.09	0.11	0.15
	AS + RT	0.39	0.35	0.87	0.05	1.96
	RS + AT	0.15	0.57	1.07	0.05	0.37
	AS + AT	0.12	0.30	0.06	0.36	1.83
Swin transformer	RS + RT	0.43	0.74	0.02	0.22	0.86
	AS + RT	0.20	0.52	0.13	0.82	0.92
	RS + AT	0.52	0.18	0.30	0.22	1.91
	AS + AT	0.12	0.62	0.26	0.28	1.64
PVT	RS + RT	0.57	0.17	0.18	0.26	0.98
	AS + RT	0.14	0.13	0.25	0.32	1.46
	RS + AT	0.47	0.02	0.24	0.13	0.53
	AS + AT	0.23	0.41	0.27	0.35	1.92
MobileViT	RS + RT	0.01	0.06	0.13	0.19	0.30
	AS + RT	0.13	0.48	0.12	0.28	1.13
	RS + AT	0.05	0.57	0.67	0.22	1.05
	AS + AT	0.23	0.07	0.03	0.21	2.13
Axial transformer	RS + RT	0.01	0.13	0.16	0.07	0.19
	AS + RT	0.22	0.06	0.16	0.14	2.62
	RS + AT	0.16	0.14	0.62	0.52	1.94
	AS + AT	0.01	0.01	0.17	0.23	2.11

Table 4. Mean variations of F1-score by each dataset and architecture.

Architecture	Source data		Target data		Class						
	Raw	Aug	Raw	Aug	Brick	Concrete	Glass	Stone	Mixed	Others	Average
ViT	608	–	928	–	79.34	71.52	58.23	57.34	91.89	86.77	74.18
	608	3648	928	–	83.59	78.45	63.51	67.65	96.95	86.80	79.49
	928	–	928	5568	82.85	77.02	65.23	65.34	95.89	81.77	78.02
	928	3648	928	5568	87.73	73.29	73.75	77.91	91.89	86.77	81.89
	Average				83.38	75.07	65.18	67.06	94.16	85.53	78.40
Swin transformer	608	–	928	–	80.53	75.05	59.78	59.42	92.83	87.35	75.83
	608	3648	928	–	85.87	72.37	65.32	64.55	98.93	79.41	77.74
	928	–	928	5568	83.83	78.81	66.54	62.14	96.62	85.29	78.87
	928	3648	928	5568	88.21	83.58	69.31	70.24	92.03	85.71	81.51
	Average				84.61	77.45	65.24	64.09	95.10	84.44	78.49
PVT	608	–	928	–	79.48	74.45	58.48	58.31	91.66	76.71	73.18
	608	3648	928	–	83.22	78.53	63.76	62.20	95.53	77.67	76.82
	928	–	928	5568	82.59	77.56	65.85	61.53	94.42	73.92	75.98
	928	3648	928	5568	88.31	83.72	69.42	68.51	91.57	85.22	81.13
	Average				83.40	78.57	64.38	62.64	93.30	78.38	76.78
MobileViT	608	–	928	–	80.53	74.33	58.77	58.42	92.22	77.15	73.57
	608	3648	928	–	84.27	79.45	64.16	63.31	96.93	78.18	77.72
	928	–	928	5568	83.33	78.12	65.53	62.15	95.76	74.83	76.62
	928	3648	928	5568	88.15	83.98	73.15	69.43	91.33	85.52	81.93
	Average				84.07	78.97	65.40	63.33	94.06	78.92	77.46
Axial transformer	608	–	928	–	79.63	74.14	58.53	58.41	91.87	76.65	73.21
	608	3648	928	–	82.93	78.76	63.98	62.34	96.13	77.50	76.94
	928	–	928	5568	83.12	78.97	65.15	62.95	96.20	74.43	76.80
	928	3648	928	5568	88.51	84.24	72.64	69.14	92.38	86.29	82.20
	Average				83.55	79.03	65.08	63.21	94.15	78.72	77.29

Table 5. Model performance by each dataset and architecture.

Iteration	Training time (min)						
	Mean	Std	Min	25%	50%	75%	Max
3000	75.64	2.97	73.71	73.87	75.05	77.21	79.82
5000	105.84	0.83	74.32	75.13	75.92	76.44	76.93
Architecture	Detection speed (FPS)						
	Mean	Std	Min	25%	50%	75%	Max
ViT	25	0	20	25	25	25	25
Swin transformer	20	0	16.67	20	20	20	20
PVT	20	0	16.67	20	20	20	20
MobileViT	33.33	0	33.33	33.33	33.33	33.33	33.33
Axial transformer	20	0	16.67	20	20	20	20

Table 6. Training time by iterations, and detection speed by architectures.

model achieved an average F1-score of 82.04% on the unseen London test set, comparable to its validation performance. With a detection speed of approximately 20 FPS (Table 6), this configuration offers a favorable balance between accuracy and efficiency.

Figure 15 presents a subset of images from the London dataset, illustrating both successful and erroneous detections. A more extensive collection of detection outcomes is available at the following link: https://figshare.com/articles/dataset/Dataset_of_building_characteristics_from_building_fa_ade_images/25931941.

Feature-space analysis of source–target alignment

To provide mechanistic evidence of how the proposed framework aligns source and target representations, the penultimate-layer features of validation images from Scotland (source) and London (target) were visualized using t-SNE (Fig. 16). Each point represents a single validation image, with color indicating one of the six cladding classes and the legend identifying the domain. Under raw training without DA, source and target samples of the same class form clearly separated clusters, indicating a pronounced domain shift. When DA is

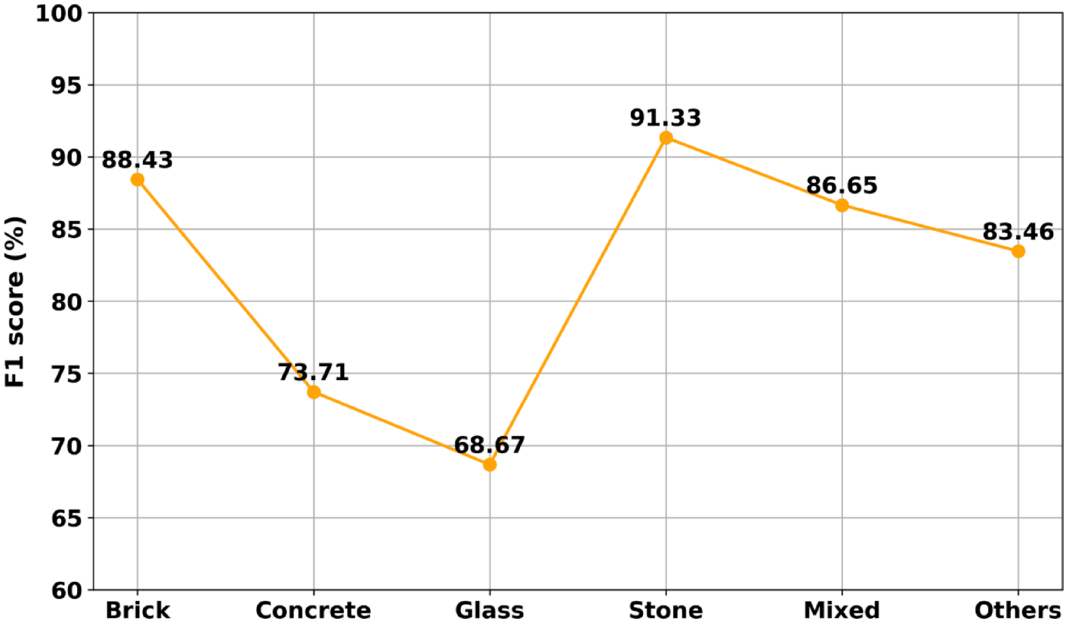


Fig. 14. Best model performance on test data.

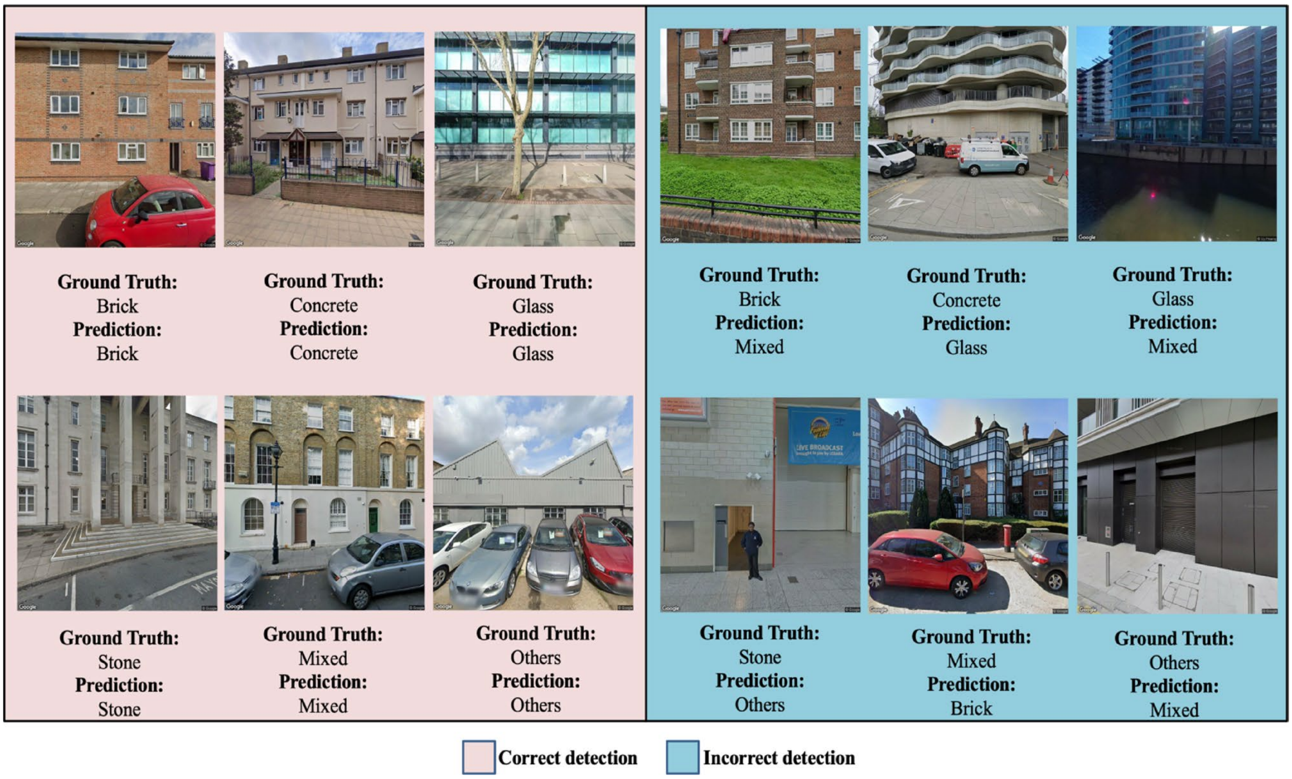

Ground Truth: Brick Prediction: Brick
Ground Truth: Concrete Prediction: Concrete
Ground Truth: Glass Prediction: Glass
Ground Truth: Brick Prediction: Mixed
Ground Truth: Concrete Prediction: Glass
Ground Truth: Glass Prediction: Mixed
Ground Truth: Stone Prediction: Stone
Ground Truth: Mixed Prediction: Mixed
Ground Truth: Others Prediction: Others
Ground Truth: Stone Prediction: Others
Ground Truth: Mixed Prediction: Brick
Ground Truth: Others Prediction: Mixed
Correct detection
Incorrect detection

Fig. 15. Examples of correct and incorrect results in test data.

applied to models trained only on raw data, the distance between these clusters is reduced, although the feature space remains relatively diffuse.

Comparison with different baseline model

To examine the effectiveness of both supervised learning and domain adaptation under domain shift, experiments were carried out using an axial transformer-based architecture across various training and testing scenarios involving data from London and Scotland. Table 7 presents a comparison of the results for each dataset under both methods.

Under supervised learning, models trained and tested within the same domain performed reasonably well (e.g., London→London: 72.75%, Scotland→Scotland: 77.10%), indicating that the model effectively learned domain-specific features. However, in cross-domain scenarios without adaptation (e.g., London→Scotland or Scotland→London), the average scores consistently dropped to the mid- to high-60% range. Even when combining training data from both London and Scotland, the improvements remained modest, suggesting that simple data diversification is insufficient to fully overcome the domain shift.

In contrast, applying domain adaptation techniques substantially improved cross-domain generalization. For instance, configurations like AS (Scotland) + AT (London) and AS (London) + AT (Scotland) resulted in the average gains surpassing those achieved in the best supervised scenarios, frequently exceeding 80%. These domain-adapted models also demonstrated more balanced performance across challenging classes, such as Glass and Stone, which are prone to variability in appearance.

Conclusions

This study introduced DA framework designed to address the challenges of exterior cladding material classification in SVI. The proposed method leverages a DANN alongside advanced transformer-based backbones (ViT, Swin Transformer, PVT, MobileViT, and Axial Transformer) and several image augmentation techniques. The following key conclusions can be drawn:

1. Experiments confirmed that DA significantly enhances classification performance in cross-domain scenarios (e.g., London→Scotland or Scotland→London). In many cases, domain-adapted models achieved average scores exceeding 80%, surpassing those trained using conventional supervised learning alone.

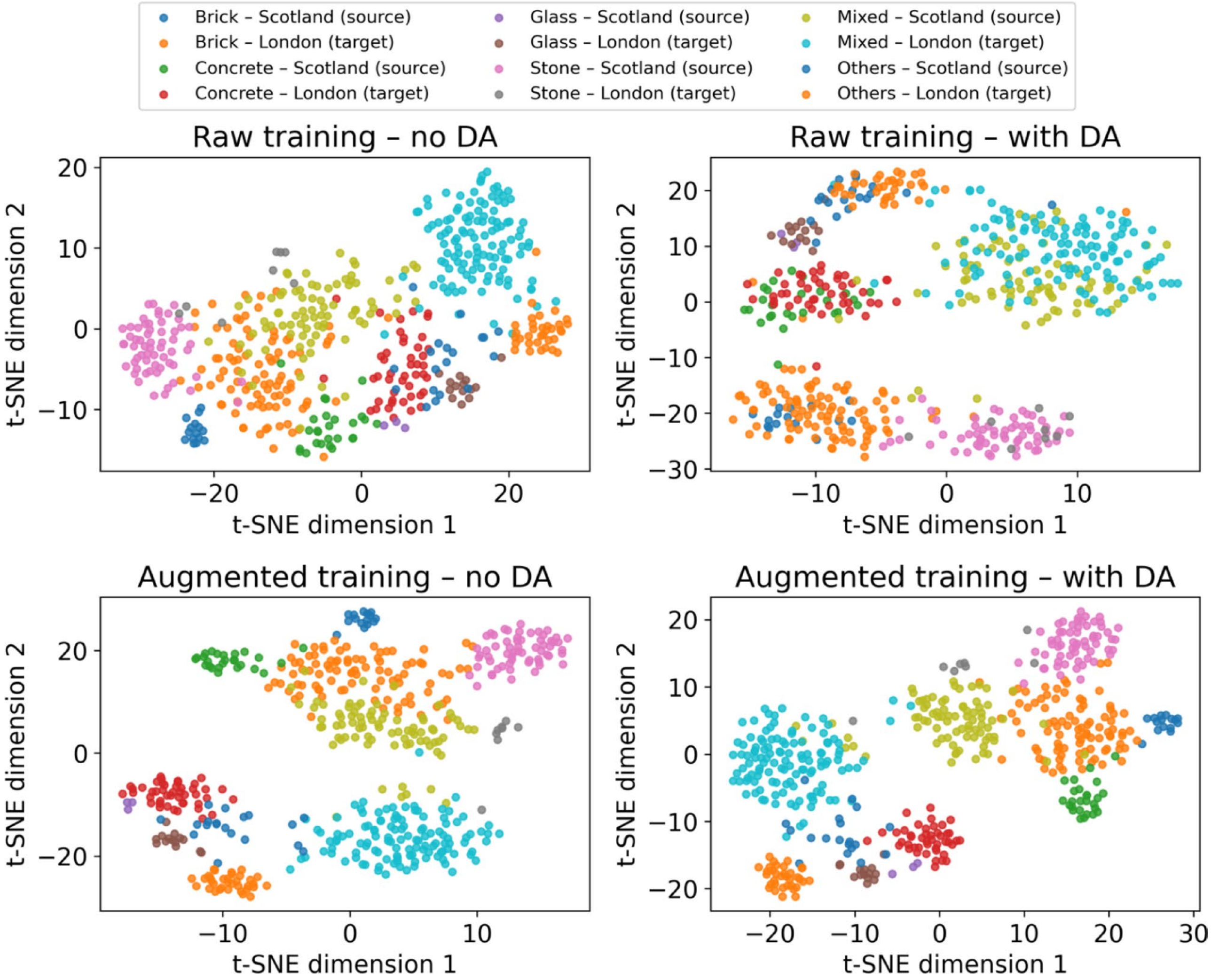


Fig. 16. Visualization of T-SNE using the best model.

Training data	Test data	Training method	Class						
			Brick	Concrete	Glass	Stone	Mixed	Others	Average
London	London	Supervised learning	78.02	71.18	66.45	72.87	66.53	81.44	72.75
Scotland	London		53.24	68.92	63.2	47.22	64.02	83.53	63.36
Scotland	Scotland		82.12	91.23	67.92	68.56	69.44	83.32	77.10
London	Scotland		83.45	72.67	64.78	49.23	61.1	80.55	68.63
London + Scotland	London		83.45	75.98	69.34	75.12	84.67	84.23	78.80
London + Scotland	Scotland		83.77	85.33	68.56	73.98	83.45	85.1	80.03
London + Scotland	London		82.02	76.34	67.81	75.88	85.12	84.77	78.66
London + Scotland	Scotland		82.35	85.78	71.43	74.33	83.83	85.56	80.55
AS (Scotland) + AT (London) AS (Scotland) + AT (London)	London	Domain adaptation	88.43	73.71	68.67	91.33	86.65	83.46	82.04
	Scotland		86.32	72.55	67.93	88.87	85.93	84.12	80.95
AS (London) + AT (Scotland) AS (London) + AT (Scotland)	London		85.23	71.37	70.46	87.12	88.33	82.73	80.87
	Scotland		89.45	83.12	70.82	92.45	88.99	84.67	84.92

Table 7. Comparison results of supervised learning and domain adaptation in each dataset.

2. Evaluations of five transformer architectures underlined the versatility and potential of vision transformers for building façade analysis. Among these, the axial transformer often displayed a competitive balance of accuracy and computational efficiency.
3. Data augmentation substantially improved classification accuracy, particularly when addressing material classes with limited examples in raw training sets (e.g., "Glass" or "Stone"). By producing synthetic samples under diverse brightness, scale, and perspective parameters, the model more effectively generalized to real-world conditions.
4. Tuning of hyperparameters showed that the number of training iterations often exerted the largest influence on model performance—sometimes improving average score by more than two percentage points. However, depending on the architecture, other hyperparameters (like weight decay in ViT) could also introduce notable gains.
5. Despite the computational overhead typically associated with transformers, the tested models maintained practical detection speeds, with MobileViT in particular achieving real-time or near-real-time inference.

However, this research is limited by geographic scope, labeling costs, and the complexity of transformer-based architectures. Future studies could broaden datasets across more regions, explore advanced or multi-step domain adaptation strategies, and investigate lightweight solutions for on-device deployment. Integrating synthetic data may further enhance performance and generalization across a wider variety of building facades.

Data availability

The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

Code availability

The code developed for this study is available at the Figshare repository[58].

Received: 17 September 2025; Accepted: 10 December 2025
Published online: 18 December 2025

References

1. Hill, C., Kymäläinen, M. & Rautkari, L. Review of the use of solid wood as an external cladding material in the built environment. *J. Mater. Sci.* https://doi.org/10.1007/s10853-022-07211-x (2022).
2. Takano,A. et al. The effect of material selection on life cycle energy balance: A case study on a hypothetical building model in Finland. *Build. Environ.* (2015). https://doi.org/10.1016/j.buildenv.2015.03.001
3. Rahiminejad, M. & Khovalyg, D. Review on ventilation rates in the ventilated air-spaces behind common wall assemblies with external cladding. *Build. Environ.* https://doi.org/10.1016/j.buildenv.2020.107538 (2021).
4. Yan, B., Meng, X., Ouyang, J. & Long, E. Impact of occupant behavior on thermal performance of the Typical-Composite walls of a Building. *J. Energy Eng.* https://doi.org/10.1061/(asce)ey.1943-7897.0000788 (2021).
5. Lee, J. S. Life cycle costing for exterior materials on Building Façade. *J. Constr. Eng. Manag.* https://doi.org/10.1061/(asce)co.1943-7862.0002068 (2021).
6. Wang, S. & Han, J. Automated detection of exterior cladding material in urban area from street view images using deep learning. *J. Build. Eng.* **96**, 110466. https://doi.org/10.1016/j.jobe.2024.110466 (2024).
7. Chen, F. C., Subedi, A., Jahanshahi, M. R., Johnson, D. R. & Delp, E. J. Deep Learning–Based Building attribute Estimation from Google street view images for flood risk assessment using feature fusion and task relation encoding. *J. Comput. Civ. Eng.* https://doi.org/10.1061/(asce)cp.1943-5487.0001025 (2022).
8. Cha, Y. J., Choi, W., Suh, G., Mahmoudkhani, S. & Büyüköztürk, O. Autonomous structural visual inspection using Region-Based deep learning for detecting multiple damage types. *Comput. Civ. Infrastruct. Eng.* https://doi.org/10.1111/mice.12334 (2018).
9. Wang, S., Park, S., Park, S. & Kim, J. Building façade datasets for analyzing Building characteristics using deep learning. *Data Br.* **57**, 110885. https://doi.org/10.1016/j.dib.2024.110885 (2024).

10. Wang, S. Evaluating Cross-Building transferability of Attention-Based automated fault detection and diagnosis for air handling units: auditorium and hospital case study. *Build. Environ.* 113889. https://doi.org/10.1016/j.buildenv.2025.113889 (2025).
11. Oliveira Santos, B., Valença, J., Costeira, J. P. & Julio, E. Domain adversarial training for classification of cracking in images of concrete surfaces. *AI Civ. Eng.* https://doi.org/10.1007/s43503-022-00008-6 (2022).
12. Ilic, L., Sawada, M. & Zarzelli, A. Deep mapping gentrification in a large Canadian City using deep learning and Google street view. *PLoS One*. https://doi.org/10.1371/journal.pone.0212814 (2019).
13. Hu, C. B., Zhang, F., Gong, F. Y., Ratti, C. & Li, X. Classification and mapping of urban Canyon geometry using Google street view images and deep multitask learning. *Build. Environ.* https://doi.org/10.1016/j.buildenv.2019.106424 (2020).
14. Campbell, A. & Both, A. (eds), Q. (Chayn) Sun, Detecting and mapping traffic signs from Google Street View images using deep learning and GIS, Comput. Environ. Urban Syst. (2019). https://doi.org/10.1016/j.compenvurbsys.2019.101350
15. Yan, Y. & Ryu, Y. Exploring Google street view with deep learning for crop type mapping. *ISPRS J. Photogramm Remote Sens.* https://doi.org/10.1016/j.isprsjprs.2020.11.022 (2021).
16. Zou, S. & Wang, L. Detecting individual abandoned houses from Google street view: A hierarchical deep learning approach. *ISPRS J. Photogramm Remote Sens.* https://doi.org/10.1016/j.isprsjprs.2021.03.020 (2021).
17. Kim, J., Kamari, M., Lee, S. & Ham, Y. Large-Scale Visual Data–Driven Probabilistic Risk Assessment of Utility Poles Regarding the Vulnerability of Power Distribution Infrastructure Systems, *J. Constr. Eng. Manag* https://doi.org/10.1061/(asce)co.1943-7862.0002153. (2021).
18. Kalfarisi, R., Hmosze, M. & Wu, Z. Y. Detecting and geolocating City-Scale Soft-Story buildings by deep machine learning for urban seismic resilience. *Nat. Hazards Rev.* https://doi.org/10.1061/(asce)nh.1527-6996.0000541 (2022).
19. Tan, B., Zhang, Y., Pan, S. J. & Yang, Q. Distant domain transfer learning. *Proc. 31st AAAI Conf. Artif. Intell.* 2604–2610. https://doi.org/10.1109/CVPR.2017.754 (2017).
20. Rosenstein, M. T., Marx, Z., Kaelbling, L. P. & Dietterich, T. G. *To Transfer or not To Transfer, NIPS Work* (Inductive Transf, 2005).
21. Pan, S. J. & Yang, Q. A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.* https://doi.org/10.1109/TKDE.2009.191 (2010).
22. Wang, S. A hybrid SMOTE and Trans-CWGAN for data imbalance in real operational AHU AFDD : A case study of an auditorium Building. *Energy Build.* **348**, 116447. https://doi.org/10.1016/j.enbuild.2025.116447 (2025).
23. Hong, S., Im, W., Ryu, J. & Yang, H. S. SSPP-DAN: Deep domain adaptation network for face recognition with single sample per person, in: Proc. - Int. Conf. Image Process. ICIP, (2018). https://doi.org/10.1109/ICIP.2017.8296396
24. Hu, D., Gan, V. J. L. & Zhai, R. Automated BIM-to-scan point cloud semantic segmentation using a domain adaptation network with hybrid attention and whitening (DawNet). *Autom. Constr.* **164**, 105473. https://doi.org/10.1016/j.autcon.2024.105473 (2024).
25. Hong, Y., Chern, W. C., Nguyen, T. V., Cai, H. & Kim, H. Semi-supervised domain adaptation for segmentation models on different monitoring settings. *Autom. Constr.* **149**, 104773. https://doi.org/10.1016/j.autcon.2023.104773 (2023).
26. Duan, B., Qian, K., Liu, A. & Luo, S. Visual–tactile learning of robotic cable-in-duct installation skills. *Autom. Constr.* **170**, 105905. https://doi.org/10.1016/j.autcon.2024.105905 (2025).
27. Tran, D. Q. et al. Leveraging semisupervised learning for domain adaptation: enhancing safety at construction sites through Long-Tailed object detection. *J. Constr. Eng. Manag.* **151**, 4024190. https://doi.org/10.1061/JCEMD4.COENG-15259 (2025).
28. Wang, S. Effectiveness of traditional augmentation methods for rebar counting using UAV imagery with faster R-CNN and YOLOv10-based transformer architectures. *Sci. Rep.* **15**, 33702. https://doi.org/10.1038/s41598-025-18964-1 (2025).
29. Wang, S., Korolija, I. & Rovas, D. Impact of traditional augmentation methods on window state detection. *CLIMA 2022 Conf.* **1-8** https://doi.org/10.34641/clima.2022.375 (2022).
30. Wang, S. Development of approach to an automated acquisition of static street view images using transformer architecture for analysis of Building characteristics. *Sci. Rep.* **15**, 29062. https://doi.org/10.1038/s41598-025-14786-3 (2025).
31. Kandel, I., Castelli, M. & Manzoni, L. Brightness as an augmentation technique for image classification. *Emerg. Sci. J.* https://doi.org/10.28991/ESJ-2022-06-04-015 (2022).
32. Shorten, C. & Khoshgoftaar, T. M. A survey on Image Data Augmentation for Deep Learning, *J. Big Data* https://doi.org/10.1186/s40537-019-0197-0. (2019).
33. Ottoni, A. L. C., de Amorim, R. M., Novo, M. S. & Costa, D. B. Tuning of data augmentation hyperparameters in deep learning to Building construction image classification with small datasets. *Int. J. Mach. Learn. Cybern*. https://doi.org/10.1007/s13042-022-01555-1 (2023).
34. Wang, S. Evaluation of impact of image augmentation techniques on two tasks: window detection and window States detection. *Results Eng.* **24**, 103571. https://doi.org/10.1016/j.rineng.2024.103571 (2024).
35. Wang, S., Kim, J., Park, S. & Kim, J. Fault Diagnosis of Air Handling Units in an Auditorium Using Real Operational Labeled Data Across Different Operation Modes, *Comput. Civ. Eng.* **39** https://doi.org/10.1061/JCCEE5/CPENG-6677. (2025).
36. Wang, S., Moon, S., Eum, I., Hwang, D. & Kim, J. A text dataset of fire door defects for pre-delivery inspections of apartments during the construction stage. *Data Br.* **60**, 111536. https://doi.org/10.1016/j.dib.2025.111536 (2025).
37. Wang, A. et al. Yolov10: Real-time end-to-end object detection. *ArXiv Prepr* (2024). ArXiv2405.14458.
38. Zhao, K. et al. ST-YOLOA: a Swin-transformer-based YOLO model with an attention mechanism for SAR ship detection under complex background. *Front. Neurorobot.* **17**, 1170163 (2023).
39. Zou, W., Xie, K. & Lin, J. Light-weight deep learning method for active jamming recognition based on improved MobileViT, IET radar. *Sonar Navig.* **17**, 1299–1311 (2023).
40. Wang, S. Automated fault diagnosis detection of air handling units using real operational labelled data and Transformer-based methods at 24-hour operation hospital. *Build. Environ.* 113257. https://doi.org/10.1016/j.buildenv.2025.113257 (2025).
41. Khan, A. et al. A survey of the vision Transformers and their CNN-transformer based variants. *Artif. Intell. Rev.* https://doi.org/10.1007/s10462-023-10595-0 (2023).
42. Liu, Z. et al. Swin transformer: Hierarchical vision transformer using shifted windows, in: Proc. IEEE/CVF Int. Conf. Comput. Vis., : pp. 10012–10022. (2021).
43. Wang, S. Automated non-PPE detection on construction sites using YOLOv10 and transformer architectures for surveillance and body worn cameras with benchmark datasets. *Sci. Rep.* **15**, 27043. https://doi.org/10.1038/s41598-025-12468-8 (2025).
44. Zhang, X. & Zhang, Y. Conv-PVT: a fusion architecture of Convolution and pyramid vision transformer. *Int. J. Mach. Learn. Cybern*. **14**, 2127–2136 (2023).
45. Wang, S., Park, S., Kim, J. & Kim, J. Safety helmet monitoring on construction sites using YOLOv10 and advanced transformer architectures with surveillance and Body-Worn cameras. *J. Constr. Eng. Manag.* https://doi.org/10.1061/JCEMD4/COENG-16760 (2025).
46. Mehta, S. & Rastegari, M. Mobilevit: light-weight, general-purpose, and mobile-friendly vision transformer. *ArXiv Prepr* (2021). ArXiv2110.02178.
47. Ho, J., Kalchbrenner, N., Weissenborn, D. & Salimans, T. Axial attention in multidimensional Transformers. *ArXiv Prepr* (2019). ArXiv1912.12180.
48. Liu, Y., Yu, W., Zhang, Z., Wang, Q. & Che, L. Axial Attention Transformer for Fast High-quality Image Style Transfer, 2024 IEEE Int. Symp. Circuits Syst. 1–5. (2024). https://api.semanticscholar.org/CorpusID:270926962
49. Li, J. et al. Domain adaptation based object detection for autonomous driving in foggy and rainy weather. *IEEE Trans. Intell. Veh.* (2024).
50. Liu, X. et al. Deep unsupervised domain adaptation: A review of recent advances and perspectives. *APSIPA Trans. Signal. Inf. Process.* **11** (2022).

51. Wang, S., Eum, I., Park, S. & Kim, J. A labelled dataset for rebar counting inspection on construction sites using unmanned aerial vehicles. *Data Br.* **110720** https://doi.org/10.1016/j.dib.2024.110720 (2024).
52. Wang, S., Eum, I., Park, S. & Kim, J. A semi-labelled dataset for fault detection in air handling units from a large-scale office. *Data Br.* **57**, 110956. https://doi.org/10.1016/j.dib.2024.110956 (2024).
53. Wang, N. et al. Automatic damage detection of historic masonry buildings based on mobile deep learning. *Autom. Constr.* https://doi.org/10.1016/j.autcon.2019.03.003 (2019).
54. Wang, S. Real operational labeled data of air handling units from office, auditorium, and hospital buildings. *Sci. Data*. https://doi.org/10.1038/s41597-025-05825-9 (2025).
55. Park, S., Kim, J., Wang, S. & Kim, J. Effectiveness of image augmentation techniques on Non-Protective personal equipment detection using YOLOv8. *Appl. Sci.* **15** https://doi.org/10.3390/app15052631 (2025).
56. Hwang, D., Kim, J. J., Moon, S. & Wang, S. Image augmentation approaches for Building dimension Estimation in street view images using object detection and instance segmentation based on deep learning. *Appl. Sci.* **15** https://doi.org/10.3390/app15052525 (2025).
57. Han, J., Kim, J., Kim, S. & Wang, S. Effectiveness of image augmentation techniques on detection of Building characteristics from street view images using deep learning. *J. Constr. Eng. Manag.* **150**, 1–18. https://doi.org/10.1061/JCEMD4.COENG-15075 (2024).
58. Wang, S. Building facade images for classifying Building stories and identifying Building typologies. *Figshare Data Repos*. https://doi.org/10.6084/m9.figshare.24979947.v1 (2024).

Author contributions

Conceptualization, Methodology, Software, Validation, Formal Analysis, Investigation, Data Curation, Resources, Writing—Original Draft Preparation, Writing—Review and Editing: S. Wang.

Funding

This research received no external funding.

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to S.W.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.