*applied sciences*

MDPI

*Article*

# Deep Learning for Building Attribute Classification from Street-View Images for Seismic Exposure Modeling

**Rajesh Kumar, Claudio Rota, Flavio Piccoli and Gianluigi Ciocca ***

Department of Informatics, Systems and Communication, University of Milano-Bicocca, 20126 Milan, Italy; rajesh.kumar@unimib.it (R.K.); claudio.rota@unimib.it (C.R.); flavio.piccoli@unimib.it (F.P.)
* Correspondence: gianluigi.ciocca@unimib.it

**Abstract**

Exposure models are essential for seismic risk assessment to determine environmental vulnerabilities during earthquakes. However, developing these models at scale is challenging because it relies on manual inspection of buildings, which increases costs and introduces significant delays. Developing fast, consistent, and easy-to-deploy automated methods to support this process has become a priority. In this study, we investigate the use of deep learning to accelerate the classification of architectural and structural attributes from street-view imagery. Using the Alvalade dataset, which contains 4007 buildings annotated with 10 multi-class attributes, we evaluated the performance of multiple architecture types. Our analysis shows that deep learning models can successfully extract key structural features, achieving an average macro accuracy of 57%, and a Precision, Recall, and F1-score of 61%, 57%, and 56%, respectively. We also show that prediction quality is further improved by leveraging multi-view imagery of the target buildings. These results demonstrate that deep learning can be an effective solution to reduce the manual effort required for the development of reliable large-scale exposure models, offering a practical solution toward more efficient seismic risk assessment.

**Keywords:** building classification; transfer learning; seismic risk analysis; exposure modeling; deep learning

## 1. Introduction

Building collapse is one of the most significant contributors to human casualties and economic losses during seismic events [1]. Seismic risk assessment is commonly structured around three components: hazard, exposure, and vulnerability. Hazard characterizes the probability and spatial distribution of ground shaking. Exposure describes the spatial distribution and characteristics of buildings, occupants, and replacement costs. Vulnerability is captured through fragility or vulnerability functions that relate ground shaking intensity to expected damage or loss for each building class. Our work contributes to the exposure modeling stage of this workflow. In particular, the building attributes that can be estimated automatically are key inputs for vulnerability assessment and subsequent damage estimation [2].

In recent years, seismic risk assessment has become a critical method for evaluating the seismic performance of buildings. The process of obtaining seismic vulnerability curves is a measure of the likelihood that a building will be damaged at different seismic intensities [3]. However, creating reliable building exposure models at scale remains a challenge, particularly in dense urban areas where building diversity and informal



Academic Editor: Zhengjun Liu

Received: 22 December 2025
Revised: 10 January 2026
Accepted: 14 January 2026
Published: 14 January 2026

structures complicate vulnerability assessment. These challenges are further amplified by the reliance on manual surveys and expert inspections, which are slow, costly and resource-intensive [4].

Accurate classification of buildings based on attributes such as number of floors, roof type, construction material, and structural system is essential for seismic risk assessment. These characteristics are directly integrated into vulnerability models, enabling more realistic estimates of expected damage during seismic events [5]. For example, distinguishing between reinforced-concrete and masonry structures allows for assigning more appropriate vulnerability functions to different building classes. Traditional approaches to building classification, including manual inspection and heuristic-based categorization, are time-consuming, often inconsistent, and difficult to scale beyond localized studies. Recent advancements in deep learning (DL) offer an opportunity to overcome these limitations. By leveraging street-level imagery, DL models can automatically infer relevant structural and architectural attributes from widely available visual data, reducing the need for manual data collection and enabling faster, more consistent generation of exposure models. Prior works explored the use of deep learning for building attribute classification in the context of seismic risk assessment [6–9], achieving impressive results. However, these studies typically consider a narrow range of deep learning architectures, evaluate only a limited subset of building attributes, and rely on proprietary datasets with heterogeneous attribute definitions, which limits reproducibility and prevents direct comparison across studies.

In this work, we investigate the use of deep learning for the classification of structural and architectural building attributes relevant to seismic risk assessment. We focus on the Alvalade dataset [10], which provides street-level images paired with building characteristics such as construction material, age of construction, and number of stories, collected through detailed field surveys. The labels follow the standardized taxonomy of the Global Earthquake Model (GEM), ensuring compatibility with international exposure-modeling practices. To the best of our knowledge, this is the only publicly available dataset that integrates visual data with ground-truth structural attributes specifically curated for seismic exposure assessment. Our approach involves selecting and comparing multiple deep learning architectures for building-attribute classification. We evaluate convolutional neural networks (CNNs) and vision transformers (ViTs) as backbone feature extractors, each paired with fully connected layers designed for multi-attribute prediction. A multi-attribute learning framework is adopted to jointly infer several structural and architectural characteristics from a single input image. Model performance is assessed independently for each attribute to better understand the reliability and limitations of the predictions across different categories. The results indicate that certain attributes, such as construction age and material, can be estimated with comparatively high accuracy, while others, such as the number of basements or the building position within the block, remain challenging due to their subtle or non-visible features.

The main contributions of this work are the following:

- We provide a systematic and comprehensive comparison of feature extraction models, ranging from widely used CNN backbones (e.g., ResNet [11]) to recent vision foundation models such as DINO-v2 [12];
- We focus on a publicly available dataset (Alvalade) with a standardized taxonomy of architectural and structural attributes tailored to seismic exposure modeling, establishing a reproducible baseline for future research;
- We investigate multi-view aggregation strategies using multiple street-view images of the same building and analyze their impact at the attribute level;
- We conduct an analysis of computational cost to provide guidance for selecting models under different computational constraints.

Overall, this work helps bridge the gap between structural engineering and computer science by demonstrating the potential of deep learning to perform large-scale building-attribute classification, supporting the development of exposure models that can significantly improve seismic risk assessment.

The remainder of this paper is organized as follows. Section 2 reviews related work on deep learning and computer-vision methods for building characterization and seismic exposure modeling. Section 3 describes the dataset, the preprocessing steps, and the architectural design of the proposed multi-attribute classification framework. Section 4 presents the experimental setup, including training strategy, evaluation metrics, and implementation details. Section 5 reports and discusses the quantitative and qualitative results, including model comparison, attribute-level analysis, efficiency trade-offs, and multi-view aggregation. Finally, Section 6 concludes the paper and outlines directions for future research.

## 2. Related Works

In recent years, convolutional neural networks (CNNs) have emerged as a dominant approach in the field of image classification, significantly outperforming traditional computer vision techniques in various applications such as object detection, scene recognition, and medical imaging [13]. These deep learning models, known for their ability to automatically extract spatial hierarchies of features, have demonstrated remarkable performance in handling large-scale image data. In structural engineering, for example, CNNs have been effectively employed to assess the seismic response of buildings and infrastructures, offering an efficient alternative to conventional simulation-based methods. Several studies have reviewed the applications of DL techniques in seismic damage evaluation, emphasizing their potential in predictive modeling and risk assessment [14–16].

Building on these developments, recent research has increasingly focused on leveraging publicly available imagery, such as Google Street View, combined with deep learning techniques to address challenges in the built environment. Gonzalez et al. [4] presented one of the earliest systematic efforts to automate the detection of building typologies relevant to seismic exposure modeling using deep learning. Using approximately 10,000 manually annotated Google Street View images from Medellín, Colombia, the authors fine-tuned several state-of-the-art CNN architectures to classify buildings according to their lateral load-resisting systems, materials, and ductility. Pelizari et al. [6] proposed an automated method for large-scale building characterization using CNNs and Google Street View imagery. The approach identified key seismic vulnerability parameters such as structural material, height, and seismic building type with over 80% accuracy. Tested in Santiago de Chile, it demonstrated that deep learning on street-level images can efficiently map structural exposure for seismic risk assessment. Rueda et al. [17] classify one-story unreinforced masonry buildings using different CNN models and the best VGG19 model achieves 80% average accuracy. Chen et al. [18] proposed a deep learning framework analyzes Google Street View images to estimate multiple attributes of buildings simultaneously—including foundation height, foundation type, building type, and number of stories. Iturburu et al. [7] introduced a building pose detection framework using deep learning to automatically identify the structural frame of reinforced concrete buildings from façade images. Gouveia et al. [19] developed an end-to-end deep-learning pipeline to classify façade characteristics and generate large-scale exposure models. Gomez et al. [8] proposed a fully automated typology characterization system for exposure models, demonstrating portfolio-scale gains in processing time. Tocchi et al. [9] proposed a regional-scale exposure modeling methodology that incorporates local variability in building typologies across Italy. By integrating the Cartis rapid survey with census data, they developed recalibrated regional exposure models for masonry buildings in Abruzzo, Campania, and Emilia-Romagna.

The results showed that local typological distributions can significantly alter predicted seismic losses, highlighting the importance of region-specific exposure models.

Deep learning, combined with open-source datasets, has also enabled applications beyond seismic assessment. Bhatta et al. [20] developed a semi-automated framework integrating deep learning and open geospatial data for large-scale building detection and characterization to support flood exposure analysis. Yepes-Estrada et al. [21] created a comprehensive Global Building Exposure Model that quantifies global building stocks using a harmonized taxonomy. Satellite and aerial images have similarly been used for building detection and classification. Bittner et al. [22] used Digital Surface Models with CNNs for pixel-wise building segmentation. Zheng et al. [23] applied Faster R-CNN [24] to UAV imagery for building detection. Zhang et al. [25] used Mask R-CNN [26] combined with Sobel edge detection to refine building boundaries. Ma et al. [27] used an improved Inception-v3 [28] for post-earthquake damage analysis. Huang et al. [29] introduced a dataset for building location, functionality, and roof type classification.

## 3. Materials and Methods

### 3.1. Dataset

In this study, we focus on the Alvalade dataset [10], a publicly available dataset based on buildings located in the parish of Alvalade in Lisbon, Portugal. The parish includes a diverse range of building structures typical of mid-20th-century urban development. The dataset is composed of 5276 images covering 2457 buildings. Each building in the dataset may have multiple images captured from different angles or perspectives. Specifically, buildings with 2 or 3 images represent multiple views of the same structure to enhance spatial context. Figure 1 shows some examples of building images.

**Figure 1.** Example of images contained in the Alvalade dataset. They are associated with different attributes, such as construction material, number of stories, and number of basements.

The dataset contains a taxonomy of building attributes, following a standardized taxonomy defined by the Global Earthquake Model (GEM). These attributes were obtained from field surveys and include the age of construction, construction material, number of stories, number of basements, height of floor, occupation, position within the block, vertical irregularities, horizontal irregularities, and roof cover. These attributes provide essential insights into the structural configuration and potential vulnerabilities of each building. The dataset was prepared to ensure an effective use in machine learning models for structural classification and seismic risk analysis. Table 1 provides a comprehensive overview of the various attributes utilized in the dataset, outlining the key characteristics and classifications associated with the data.

A missing values check was conducted across the dataset, revealing that multiple attribute columns contained incomplete entries. To address this, we systematically replaced all missing values with the placeholder `UNKNOWN` in the relevant attributes. This ensured data integrity and prevented disruptions during model training. In addition, the dataset contained some image entries that did not correspond to any actual image file in the folder. These entries were removed to ensure consistency between image files and their

associated attributes in the final dataset. Also, the dataset exhibits some missing images. Thus, the corresponding attributes have been removed.

**Table 1.** Description of attributes used in the Alvalade dataset. Encoded categories are shown in brackets, while `UNKNOWN` denotes missing or unavailable values.

| ID | Attribute | Values |
|---|---|---|
| A01 | Construction material | Adobe (ADO), Unreinforced Masonry (MUR), Reinforced Concrete w/infilled frames (RC/LFINF), Reinforced Concrete w/shear walls (RC/LWAL), Wooden (W), Steel (S) |
| A02 | Age of construction | Pre 1960, 1960–1985, 1985–2000, 2000–2010, 2010–2022 |
| A03 | Number of storeys | 1, 2, 3, . . . , 17, 18 |
| A04 | Number of basements | 0, 1, `UNKNOWN` |
| A05 | Height of the ground floor (mt) | 0.12, 0.2, 0.22, 0.3, 0.32, 0.34, 0.42, 0.5, 0.52, 0.53, 0.6, 0.64, 0.65, 0.78, 1.0, 1.2, 1.3, 1.5, 1.52, 1.72, 1.95, 2.8, `UNKNOWN` |
| A06 | Occupancy | Residential (RES), Commercial (COM), Educational (EDU), Office (OCO), Mixed-Use (MIX), Public, Healthcare (HEA), Industrial (IND), `UNKNOWN` |
| A07 | Position within the block | Isolated (BPD), Single-sided (BP1), Double-sided (BP2), `UNKNOWN` |
| A08 | Vertical irregularities | Other (IRVO), Pounding (POP), Soft Story (SOS), Setback (SET), `UNKNOWN` |
| A09 | Horizontal irregularities | Other (IRHO), Torsion (TOR), `UNKNOWN` |
| A10 | Roof covering | Concrete (TMTO), Clay (RMT1), Composite (RMN), `UNKNOWN` |

After preprocessing, we obtained 4007 building images corresponding to a total of 1600 unique buildings. The distribution of images per building is thus as follows:

- Buildings with 1 image: 361;
- Buildings with 2 images: 71;
- Buildings with 3 images: 1168;
- Total buildings: 1600;
- Total images: 4007.

Figure 2 shows the distribution of attributes. The class distribution is unbalanced, with certain categories, such as reinforced concrete and residential occupancy, appearing far more frequently than others. This imbalance may affect model performance, particularly for underrepresented classes, and needs to be considered during model assessment.

### *3.2. Deep Learning Models*

In this study, we used transfer learning to leverage pretrained models for multi-attribute image classification problems. Transfer learning is a technique that allows knowledge acquired by a model on one task to be reused for a different but related task [30]. In this context, it involves using a model pretrained on a large dataset as a starting point, enabling faster convergence and better generalization, especially when the target dataset is small or less diverse. We used deep learning models pretrained on ImageNet [31], a large-scale image classification dataset containing over a million labeled images across 1000 categories.

We studied various architecture types, including several variants of ResNet models, such as ResNet-18, ResNet-50, and ResNet-152 [11], and other standard classification models, such as DenseNet-121 [32], EfficientNet-B4 [33], Inception-v3 [28], MobileNet-v3 [34], and SqueezeNet [35]. In addition, we included the DINO-v2 Vision Transformer (ViT) model [12]. In the following, we provide a brief overview of the main characteristics of the models we considered in this study.

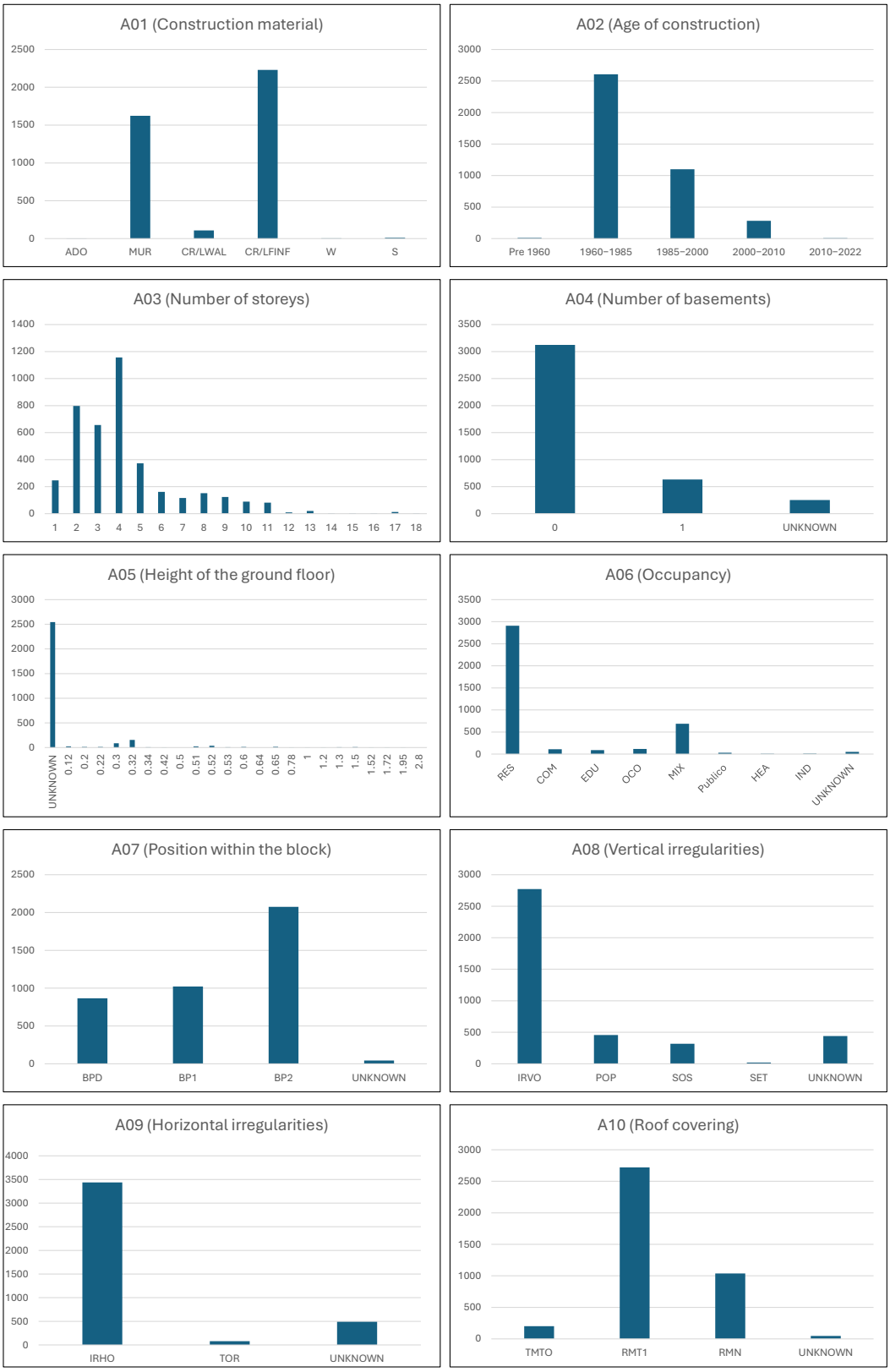


**Figure 2.** Distributions of the attributes in the dataset.

### 3.2.1. ResNet-Based Models

ResNet (Residual Network) introduces shortcut connections that allow gradients to flow directly across layers, enabling the training of very deep architectures. We use three variants: ResNet-18 (with a 512-dimensional output), ResNet-50, and ResNet-152 (both producing 2048-dimensional feature embeddings). All models are composed of

residual blocks with convolutional layers, batch normalization, and ReLU activations, with increasing depth yielding higher representational capacity.

#### 3.2.2. DenseNet-121

DenseNet-121 connects each layer to all subsequent layers within a dense block, promoting feature reuse and efficient gradient propagation. This design leads to compact models with strong performance and fewer parameters. The final feature vector produced by DenseNet-121 has 1024 dimensions.

#### 3.2.3. EfficientNet-B4

EfficientNet-B4 is a mid-sized model in the EfficientNet family, offering a balanced trade-off between accuracy and computational efficiency. It is based on a compound scaling method that uniformly scales network depth, width, and resolution using a fixed set of scaling coefficients. EfficientNet-B4 achieves high accuracy with significantly higher efficiency compared to traditional architectures of similar performance. It produces 1792-dimensional feature embeddings for each input image.

#### 3.2.4. Inception-v3

Inception-v3 is a deep architecture that captures multi-scale information using inception modules, which apply multiple filter sizes in parallel. It reduces computational cost through factorized convolutions and uses auxiliary classifiers and batch normalization to stabilize training. The model outputs a 2048-dimensional feature vector after global average pooling.

#### 3.2.5. MobileNet-v3

MobileNet-v3 is a lightweight architecture optimized for mobile and embedded applications. It builds on the inverted-residual and linear-bottleneck design of MobileNet-v2, adding squeeze-and-excitation blocks, the efficient H-swish activation, and Neural Architecture Search-guided refinements to improve the accuracy/latency trade-off. The network outputs a 1280-dimensional feature embedding after global average pooling, offering strong representational power while maintaining a low computational complexity.

#### 3.2.6. SqueezeNet

SqueezeNet is a lightweight CNN designed to drastically reduce the number of parameters without compromising accuracy. It achieves this by using "fire modules", which consist of a squeeze layer with $1 \times 1$ convolutions followed by expand layers with $1 \times 1$ and $3 \times 3$ filters. SqueezeNet generates a 512-dimensional output embedding.

#### 3.2.7. DINO-v2 Vision Transformer (ViT)

DINO-v2 is a self-supervised Vision Transformer (ViT) trained using a knowledge distillation framework without labeled data. It learns rich visual representations by aligning student and teacher networks over large-scale image datasets. Unlike convolutional models, DINO-v2 processes input images as sequences of patches and relies on multi-head self-attention [36] to capture global context. The versions used in this study have a 384-dimensional (S model) and 768-dimensional (B model) embeddings.

### *3.3. Architecture Design*

Given the multi-attribute nature of the task, where each input image must be classified according to 10 distinct structural and architectural attributes, we adopted a modular architecture composed of two main components. A single deep learning model serves as a shared feature extraction backbone responsible for generating high-level image embeddings. This

backbone is followed by 10 fully connected layers, each acting as an independent classification head dedicated to a specific attribute. Each head receives the same shared embedding as input and outputs class probabilities corresponding to the number of categories defined for that attribute. An overview of the architecture design is presented in Figure 3. This design enables efficient parameter sharing while allowing attribute-specific learning.

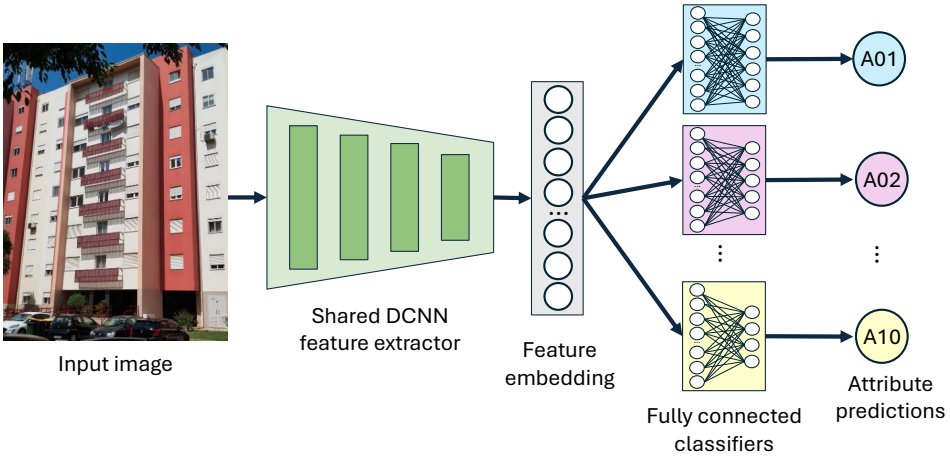


**Figure 3.** Overview of the adopted multi-attribute classification architecture. A shared backbone network extracts visual features from input images, which are then processed by different classification heads, one for each building attribute.

## 4. Experimental Setup

### *4.1. Training Details*

All the models were trained using the Adam optimizer [37]. The initial learning rate was set to $1 \times 10^{-4}$, with a batch size of 32. Training was conducted for 300 epochs to ensure sufficient convergence. The learning rate was reduced by a factor of 10 when the validation loss did not decrease for 5 consecutive evaluations, until reaching a minimum of $1 \times 10^{-6}$. We used cross-entropy loss to optimize classification performance across all attributes. Due to the imbalance problem of the data, we employed the class-balanced loss based on the effective number of samples, as proposed by Cui et al. [38], to weight the contribution of each class and mitigate bias toward majority classes. Given the multi-attribute nature of the task, the loss was computed independently for each attribute and then aggregated.

Despite the inherently continuous nature of attribute A05 (height of the ground floor), we considered it as a categorical variable to align it with our classification framework. We excluded `UNKNOWN` labels from the loss computation to ensure that the model is not penalized for uncertain or unavailable data. Due to the limited amount of data available, we used an augmentation pipeline to improve model generalization. The images were resized to a resolution of $224 \times 224$ pixels (except for Inception-v3, which requires a resolution of $299 \times 299$ pixels). Additional augmentations include horizontal flipping and slight image translation.

We used 5-fold cross-validation to robustly evaluate the model performance and ensure that the results are not dependent on a particular train-test split. Since there are multiple images of the same buildings, as described on Section 3.1, we carefully split the data to prevent the same building from being included in different sets, thereby ensuring a fair evaluation of the models.

In our experiments, the backbone networks of DINO-v2 S and B models were kept frozen during training to retain the representations learned from pretraining on large-scale datasets, and only the fully connected layers used for attribute classification were trained. For all the other models, the entire network (i.e., backbone and fully connected layers) was

trained end-to-end, allowing the model to better adapt its internal features to the specific characteristics of seismic exposure data. In this case, the backbone weights were initialized with ones of the models pretrained on ImageNet [31].

### 4.2. Evaluation Metrics

We evaluated model performance using standard classification metrics: accuracy, precision, recall, and F1 score. In this study, accuracy was computed using both macro and micro averaging strategies, while precision, recall, and F1 score were evaluated using the macro-averaged version to ensure equal contribution from each class regardless of class imbalance. The base definitions of these metrics are as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

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

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

$$\text{F1 score} = \frac{2 \cdot P \cdot R}{P + R} \tag{4}$$

where $TP$ denotes true positives (correctly predicted positive samples), $FP$ false positives (incorrectly predicted positive samples), $FN$ false negatives (missed positive samples), and $TN$ true negatives (correctly predicted negative samples).

Micro accuracy aggregates all predictions across all classes before computing the metric, thereby giving equal weight to each individual prediction. It is defined as:

$$\text{Micro accuracy} = \frac{\sum_{i=1}^{C}(TP_i + TN_i)}{\sum_{i=1}^{C}(TP_i + TN_i + FP_i + FN_i)} \tag{5}$$

where $C$ is the number of classes.

In contrast, macro accuracy treats each class equally by first computing the accuracy for each class independently and then averaging the results:

$$\text{Macro accuracy} = \frac{1}{C}\sum_{i=1}^{C}\frac{TP_i + TN_i}{TP_i + TN_i + FP_i + FN_i} \tag{6}$$

Similarly, macro-averaged versions of precision, recall, and F1-score are defined as:

$$\text{Macro precision} = \frac{1}{C}\sum_{i=1}^{C}\frac{TP_i}{TP_i + FP_i} \tag{7}$$

$$\text{Macro recall} = \frac{1}{C}\sum_{i=1}^{C}\frac{TP_i}{TP_i + FN_i} \tag{8}$$

$$\text{Macro F1 score} = \frac{1}{C}\sum_{i=1}^{C}\frac{2 \cdot P_i \cdot R_i}{P_i + R_i} \tag{9}$$

Micro accuracy reflects the overall classification correctness, while macro-averaged metrics help to mitigate the influence of dominant classes by giving each class equal importance.

We evaluated model efficiency using the number of floating-point operations (GFLOPs) required for a single forward pass, which provides a hardware-agnostic measure of computational complexity. We also report the total number of model parameters as an indicator of model size.

### *4.3. Hardware and Implementation Details*

We conducted all the experiments on a machine running Ubuntu 22.04 LTS, equipped with an Intel Core i7-7700 CPU, 32 GB of RAM, and an NVIDIA GeForce GTX 1080 GPU. The experiments were implemented in Python version 3.8.20 using the PyTorch (https://pytorch.org/, accessed on 20 December 2025) framework. Data augmentation was performed with Albumentations (https://albumentations.ai/docs/, accessed on 20 December 2025), providing flexible and efficient image transformations, while TorchMetrics (https://lightning.ai/docs/torchmetrics/stable/, accessed on 20 December 2025) was used to compute evaluation metrics.

## 5. Results and Discussion

We present the quantitative performance of the deep learning models on the multi-attribute building classification task. The performance of each model is summarized in Table 2 (macro accuracy), Table 3 (micro accuracy), and Table 4 (macro precision, recall and F1 score). The final column in each table shows the average metric across all attributes. Due to the strong class unbalance of the dataset, macro metrics are particularly informative because they better reflect the performance on minority classes. We thus consider macro metrics as the main reference when discussing the obtain results. Nevertheless, we also report micro accuracy as a complementary measure, as it reflects the overall proportion of correctly classified samples but is dominated by the majority classes. We reported the performance of a random classifier in Tables 2 and 3 as a lower-bound reference. Note that all the results correspond to the mean performance across the five test folds of the cross-validation procedure.

**Table 2.** Macro accuracy results (%) of various models across all attributes, averaged over the five test folds of the cross-validation procedure. For each attribute, the best result is shown in bold, the second-best is underlined. Attribute names correspond to those in Table 1.

| Model | A01 | A02 | A03 | A04 | A05 | A06 | A07 | A08 | A09 | A10 | Avg. |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Random | 16.67 | 20.00 | 5.56 | 50.00 | 4.34 | 12.50 | 33.33 | 25.00 | 50.00 | 33.33 | 25.07 |
| ResNet-18 | 46.25 | **64.01** | **47.36** | 75.45 | 26.09 | 49.08 | 67.05 | 59.76 | 55.41 | 65.82 | 55.63 |
| ResNet-50 | **50.88** | 61.91 | 44.04 | 75.49 | 20.12 | 44.06 | 66.29 | 59.43 | 60.77 | 68.78 | 55.18 |
| ResNet-152 | 49.66 | 57.09 | 46.99 | 76.58 | 25.08 | 45.81 | 68.32 | 58.57 | 61.84 | 70.62 | 56.06 |
| DenseNet-121 | 45.99 | 61.47 | 46.24 | 75.17 | 25.33 | 44.37 | 67.20 | 59.80 | 62.58 | 68.97 | 55.71 |
| EfficientNet-B4 | 40.39 | 59.40 | 42.76 | 75.30 | 22.49 | 44.56 | 66.79 | 60.46 | 64.02 | 67.06 | 54.32 |
| Inception-v3 | 45.95 | 59.26 | 46.79 | **77.36** | 22.98 | 42.63 | 68.16 | **62.00** | 58.24 | **70.74** | 55.41 |
| MobileNet-v3 | 46.28 | 54.68 | 46.70 | 74.55 | **26.21** | 45.54 | 66.62 | 55.97 | 62.85 | 68.25 | 54.76 |
| SqueezeNet | 40.44 | 50.93 | 41.70 | 74.16 | 23.37 | 36.79 | 63.14 | 52.60 | 54.89 | 62.98 | 50.10 |
| DINO-v2 S | 49.18 | 58.53 | 40.76 | 72.23 | 17.69 | 48.36 | 66.46 | 54.39 | 64.33 | 68.50 | 54.04 |
| DINO-v2 B | 48.75 | 58.35 | 45.79 | 74.58 | 17.09 | **57.97** | **69.31** | 56.41 | **68.27** | 70.71 | **56.72** |

**Table 3.** Micro accuracy results (%) of various models across all attributes, averaged over the five test folds of the cross-validation procedure. For each attribute, the best result is shown in bold, the second-best is underlined. Attribute names correspond to those in Table 1.

| Model | A01 | A02 | A03 | A04 | A05 | A06 | A07 | A08 | A09 | A10 | Avg. |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Random | 48.14 | 50.42 | 16.99 | 72.09 | 30.42 | 57.29 | 38.96 | 63.00 | 95.66 | 54.51 | 52.75 |
| ResNet-18 | 86.28 | **83.86** | 73.28 | 87.86 | 57.06 | 82.40 | 69.51 | 86.12 | 97.84 | 86.78 | 81.10 |
| ResNet-50 | 85.24 | 82.62 | 71.61 | 87.59 | 53.03 | 80.39 | 68.12 | 85.84 | 97.55 | 85.65 | 79.77 |
| ResNet-152 | 85.67 | **83.86** | 73.02 | 88.10 | 58.21 | 82.14 | 70.02 | 85.81 | 97.86 | 87.18 | 81.19 |
| DenseNet-121 | 81.26 | 78.66 | 67.60 | 85.28 | 45.79 | 80.87 | 68.41 | 79.37 | 96.06 | 82.71 | 76.60 |
| EfficientNet-B4 | 84.03 | 82.95 | 71.78 | 87.17 | 52.01 | 80.83 | 69.27 | 84.11 | 97.60 | 85.21 | 79.50 |
| Inception-v3 | **86.85** | 83.77 | **75.41** | **88.49** | 54.91 | 81.13 | 69.72 | **86.89** | **98.01** | **87.96** | **81.31** |
| MobileNet-v3 | 86.45 | 83.40 | 73.35 | 87.20 | **58.29** | 80.44 | 68.37 | 84.62 | 97.70 | 86.83 | 80.66 |
| SqueezeNet | 80.84 | 77.61 | 65.21 | 86.07 | 50.13 | 77.48 | 65.50 | 81.42 | 97.15 | 82.30 | 76.37 |
| DINO-v2 S | 81.26 | 78.66 | 67.60 | 85.28 | 45.79 | 80.87 | 68.41 | 79.37 | 96.06 | 82.71 | 76.60 |
| DINO-v2 B | 82.07 | 80.89 | 71.65 | 85.94 | 48.41 | **83.31** | **70.35** | 82.50 | 96.86 | 84.63 | 78.66 |

According to the obtained results, DINO-v2 B achieves the highest macro accuracy (56.72%) among all models, followed by ResNet-152 with 56.06%. Considering micro accuracy results, Inception-v3 is the top-performing model obtaining 81.31% accuracy, immediately followed by ResNet-152 with 81.19%.

The ResNet family shows an anomalous trend: performance was expected to improve as network depth increased. Instead, ResNet-18 and ResNet-152 achieve similar macro accuracy results (i.e., 55.63% and 56.06%), both higher than ResNet-50 (55.18%), which achieves worse performance than ResNet-18 when considering micro accuracy. Nevertheless, the performance gap among the ResNet models remains relatively small. This may be related to the limited data used for fine-tuning, which may not be sufficient to fully exploit the capacity of very deep architectures.

DINO-v2 B achieves the best macro accuracy (56.72%) outperforming the smaller DINO-v2 S model by 2.68%. Although not explicitly reported in the article, it is worth noticing that a considerable degradation (about -3%) in performance was observed when fine-tuning DINO-v2 models. This may be due to the ViT backbone, which lacks the strong spatial inductive biases of CNNs and typically requires large datasets and careful optimization strategies to be fine-tuned effectively. Without sufficient regularization or adaptation, ViT-based models may struggle to retain their pretrained knowledge or to align it with the building attribute classification task.

**Table 4.** Evaluation of precision, recall and F1 score (macro-averaged) for each model and attribute, averaged over the five test folds of the cross-validation procedure. All values are in %. For each attribute, the best result is shown in bold, the second-best is underlined. Attribute names correspond to those in Table 1.

| Metric | Model | A01 | A02 | A03 | A04 | A05 | A06 | A07 | A08 | A09 | A10 | Avg. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Precision | ResNet-18 | 50.77 | **67.37** | 47.09 | 79.11 | 26.45 | **63.79** | 67.64 | 62.50 | 73.46 | 69.01 | **60.72** |
| | ResNet-50 | **57.66** | 62.27 | 43.64 | 78.45 | 19.86 | 48.93 | 66.35 | 61.58 | 67.77 | 69.76 | 57.63 |
| | ResNet-152 | 53.84 | 60.04 | 46.10 | 78.91 | 25.54 | 55.07 | **68.56** | 61.76 | **76.92** | 73.04 | 59.98 |
| | DenseNet-121 | 49.66 | 63.69 | 46.19 | 79.27 | 25.25 | 54.40 | 67.88 | 62.55 | 76.83 | 71.76 | 59.75 |
| | EfficientNet-B4 | 42.37 | 60.61 | 42.48 | 77.22 | 22.88 | 49.56 | 67.04 | 58.90 | 72.11 | 67.05 | 56.02 |
| | Inception-v3 | 50.72 | 61.58 | **47.45** | **79.93** | 22.81 | 56.15 | 67.89 | **63.16** | 71.99 | **73.49** | 59.52 |
| | MobileNet-v3 | 52.37 | 59.77 | 46.80 | 77.31 | **27.60** | 53.07 | 66.72 | 55.81 | 74.25 | 69.28 | 58.30 |
| | SqueezeNet | 40.43 | 55.00 | 42.40 | 76.31 | 21.69 | 42.48 | 63.28 | 53.80 | 57.33 | 63.17 | 51.59 |
| | DINO-v2 S | 49.84 | 54.70 | 39.91 | 73.61 | 16.52 | 47.24 | 65.65 | 49.34 | 60.19 | 65.22 | 52.22 |
| | DINO-v2 B | 45.99 | 56.01 | 44.90 | 74.68 | 14.98 | 56.84 | 68.48 | 52.31 | 65.95 | 67.18 | 54.73 |
| Recall | ResNet-18 | 46.25 | **64.01** | **47.36** | 75.45 | 26.09 | 49.08 | 67.05 | 59.76 | 55.41 | 65.82 | 55.63 |
| | ResNet-50 | **50.88** | 61.91 | 44.04 | 75.48 | 20.12 | 44.06 | 66.28 | 59.43 | 60.78 | 68.78 | 55.18 |
| | ResNet-152 | 49.66 | 57.09 | 46.99 | 76.58 | 25.08 | 45.81 | 68.32 | 58.57 | 61.84 | 70.62 | 56.06 |
| | DenseNet-121 | 45.99 | 61.47 | 46.24 | 75.17 | 25.33 | 44.37 | 67.20 | 59.79 | 62.58 | 68.97 | 55.71 |
| | EfficientNet-B4 | 40.39 | 59.39 | 42.76 | 75.30 | 22.49 | 44.56 | 66.79 | 60.46 | 64.02 | 67.06 | 54.32 |
| | Inception-v3 | 45.95 | 59.26 | 46.79 | **77.36** | 22.98 | 42.63 | 68.16 | **62.00** | 58.24 | **70.74** | 55.41 |
| | MobileNet-v3 | 46.27 | 54.68 | 46.70 | 74.55 | **26.21** | 45.54 | 66.62 | 55.97 | 62.84 | 68.24 | 54.76 |
| | SqueezeNet | 40.44 | 50.93 | 41.70 | 74.15 | 23.37 | 36.79 | 63.14 | 52.60 | 54.89 | 62.98 | 50.10 |
| | DINO-v2 S | 49.18 | 58.53 | 40.76 | 72.23 | 17.69 | 48.36 | 66.45 | 54.39 | 64.33 | 68.50 | 54.04 |
| | DINO-v2 B | 48.75 | 58.35 | 45.80 | 74.58 | 17.09 | **57.97** | **69.31** | 56.41 | **68.27** | 70.71 | **56.72** |
| F1 score | ResNet-18 | 47.10 | **65.06** | **46.43** | 76.83 | 24.66 | 52.38 | 67.20 | 60.68 | 56.85 | 66.71 | 56.39 |
| | ResNet-50 | **52.75** | 61.71 | 43.06 | 76.69 | 18.59 | 45.01 | 65.98 | 60.07 | 62.90 | 68.53 | 55.53 |
| | ResNet-152 | 49.55 | 58.14 | 45.79 | 77.63 | 23.67 | 48.21 | 68.16 | 59.84 | 65.12 | 71.19 | **56.73** |
| | DenseNet-121 | 46.53 | 62.08 | 45.30 | 76.76 | 24.09 | 45.92 | 67.15 | 60.81 | **65.93** | 69.68 | 56.43 |
| | EfficientNet-B4 | 40.78 | 59.82 | 41.97 | 76.11 | 20.99 | 45.92 | 66.70 | 59.33 | 64.93 | 66.45 | 54.30 |
| | Inception-v3 | 47.13 | 60.17 | 46.31 | **78.40** | 22.04 | 46.31 | 67.86 | **62.31** | 59.80 | **71.48** | 56.18 |
| | MobileNet-v3 | 47.95 | 55.30 | 45.54 | 75.71 | **25.10** | 46.42 | 66.31 | 55.47 | 65.23 | 68.46 | 55.15 |
| | SqueezeNet | 40.19 | 51.71 | 40.85 | 74.27 | 20.16 | 36.71 | 62.61 | 52.67 | 54.95 | 62.60 | 49.67 |
| | DINO-v2 S | 47.93 | 56.12 | 39.65 | 72.80 | 16.15 | 45.85 | 65.82 | 51.11 | 61.29 | 66.15 | 52.29 |
| | DINO-v2 B | 46.69 | 56.95 | 44.30 | 74.58 | 15.42 | **54.98** | **68.68** | 53.96 | 65.67 | 68.10 | 54.93 |

In terms of precision, ResNet-18 is the overall top-performing model (60.72%), meaning that it makes fewer false positive predictions across the attributes. This makes it particularly suitable in applications where high confidence in predicted attributes is required. ResNet-152 is the second top-performing model, with a precision equal to 59.98%. Considering recall, DINO-v2 B outperforms all the other models on average, reaching a recall value of 56.72%. This suggests that it is more effective at capturing positive instances, making it a better choice in scenarios where missing relevant attributes is more problematic than predicting them incorrectly. Again, the second top-performing model is ResNet-152, which obtains a recall value of 56.06%, 0.66% lower compared to the one of DINO-v2 B. Overall, ResNet-152 represents a best balance between precision and recall, achieving the highest F1 score of 56.73%.

### *5.1. Attribute Level Insights*

The performance of the models varies significantly across the different building attributes, reflecting both the nature of the visual cues in street-level imagery and the challenges to correctly recognize specific categories. In addition, we can notice that, for some attributes like A01 (construction material), A06 (occupancy) and A09 (horizontal irregularities), the results for macro and micro accuracy are very different. This is due to strong class imbalance, as shown in Figure 2, where a few dominant classes are correctly classified, improving micro accuracy, while the rarer classes are often misclassified, leading to lower macro accuracy.

Attribute A01 (construction material) is best predicted by ResNet-50 and ResNet-152, achieving 50.88% and 49.66% micro accuracy, respectively. For A02 (age of construction) and A03 (number of storeys), ResNet-18 tops in macro accuracy with 64.01% and 47.36%, respectively. Inception-v3 is the top-performing model for A04 (number of basements), A08 (vertical irregularities) and A10 (roof covering) with 77.36%, 62% and 70.74% macro accuracy, respectively. Attribute A05 (height of the ground floor) is most accurately predicted by MobileNet-v3 (26.21%), while A06 (occupancy), A07 (position within the block) and A09 (horizontal irregularities) are better captured by DINO-v2 B with 57.97%, 69.31% and 68.27% macro accuracy, respectively.

Overall, these results highlight that there is no universally optimal architecture for all attributes. Instead, the most effective approach would be to adopt a specialized model per attribute, selecting the architecture that best captures the patterns relevant to each specific prediction task.

### *5.2. Performance vs. Efficiency Trade-Off*

Figure 4 illustrates the trade-off between performance and efficiency by plotting macro accuracy against GFLOPs, with marker size representing the number of model parameters.

DenseNet-121 achieves the best trade-off between accuracy and complexity among all the considered models: it achieves 55.71% macro accuracy while requiring only about 7 million parameters and 2.9 GFLOPs. ResNet-18 has a competitive accuracy of 55.63% at only 1.82 GFLOPs and about 11 million parameters, making it suitable for resource-constrained environments. Instead, SqueezeNet achieves much lower performance compared to the other lightweight architectures.

The two top-performing models, i.e., DINO-v2 B and ResNet-152 are also the ones with the highest model parameters and GFLOPs. Indeed, DINO-v2 achieves 56.72% average macro accuracy with 85 million parameters and 21.96 GFLOPs, while ResNet-151 achieves a little lower accuracy (56.06%) with much less parameters (58 million) and almost half GFLOPs (11.6).

Overall, this analysis highlights that shallower CNN architectures like DenseNet-121 and ResNet-18 exhibit the best balance between accuracy and efficiency. In contrast, very deep architectures like DINO-v2 B and ResNet-152 allows to achieve better performance at the cost of increased memory and time consumption. Despite their size, large models such as ResNet-152 and DINO-v2 B remain feasible for large-scale applications. Using the training setup described in Section 4.1, a single-fold training of DINO-v2 B with a pre-trained backbone completes in approximately 6 h and requires 1.4 GB of GPU memory. At inference time, it processes a 224 × 224 image in about 22 ms, using roughly 1.2 GB of GPU memory. By comparison, ResNet-152 completes training in about 5 h with backbone fine-tuning but requires more memory, i.e., 7.8 GB. During inference, it achieves similar latency (about 20 ms) using 1.8 GB of GPU memory.

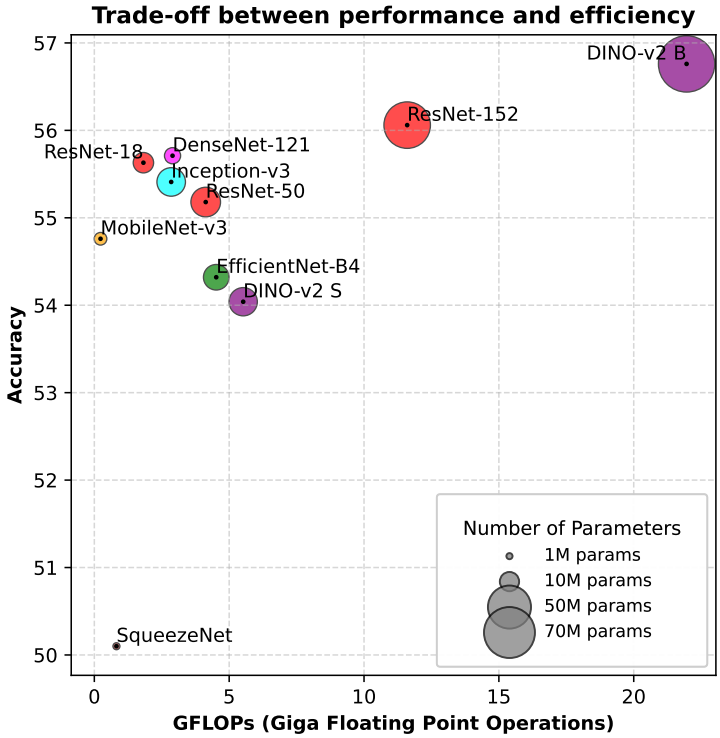


**Figure 4.** Comparison of different deep learning models to highlight the trade-offs between accuracy, efficiency, and model complexity. The horizontal axis shows computational cost measured in GFLOPs, and the vertical axis reports the macro accuracy (%) of the models. Marker size is proportional to the number of model parameters, while colors distinguish different model families. GFLOPs are computed using an input size of 224 × 224.

### *5.3. Qualitative Results*

We present some visual examples of classification results obtained using DINO-v2 B, which is the top-performing model considering macro accuracy. Figure 5 shows six classification results: three representing the best-case scenario, where all attributes are correctly predicted, and three from the worst-case scenario, where multiple attributes are misclassified.

In the best-case scenario (top row), all the attributes are correctly predicted. Interestingly, this includes attributes, such as A10 (roof covering), which are not directly visible in the image. This suggests that the model is inferring certain attributes based on contextual cues rather than direct visual evidence, leveraged from correlations learned during training. However, this may also lead to errors if the learned correlations do not hold in new or different contexts.

In the first example of the worst-case scenario (bottom row), the model misclassified 5 attributes, including A01 (construction material) and A03 (number of stories). In the other two cases, 6 and 5 attributes are misclassified, respectively. Although some attributes are visually clear from the images, several factors may have contributed to the high number of incorrect predictions. The presence of frontal objects, such as cars, may obscure critical architectural features, reducing the model ability to extract discriminative patterns. The brick facade and lighting conditions may have led the model to confuse this structure with visually similar examples from other classes in the training set. In addition, certain attributes, such as A07 (position within the block), are inherently difficult to infer from a single street-level image due to their spatial or occluded nature, thus incorporating multi-view imagery can improve robustness and interpretability when dealing with real-world urban data.

| Attribute | Prediction | Reference |
|---|---|---|
| A01 | CR/LFINF | CR/LFINF |
| A02 | 1960–1985 | 1960–1985 |
| A03 | 5 | 5 |
| A04 | 0 | 0 |
| A05 | 0.6 | UNKNOWN |
| A06 | MIX | MIX |
| A07 | BP2 | BP2 |
| A08 | SOS | SOS |
| A09 | IRHO | IRHO |
| A10 | RMN | RMN |

| Attribute | Prediction | Reference |
|---|---|---|
| A01 | CR/LFINF | CR/LFINF |
| A02 | 1985–2000 | 1985–2000 |
| A03 | 2 | 2 |
| A04 | 1 | 1 |
| A05 | 0.32 | 0.32 |
| A06 | RES | RES |
| A07 | BP2 | BP2 |
| A08 | IRVO | IRVO |
| A09 | IRHO | IRHO |
| A10 | RMT1 | RMT1 |

| Attribute | Prediction | Reference |
|---|---|---|
| A01 | MUR | MUR |
| A02 | 1960–1985 | 1960–1985 |
| A03 | 4 | 4 |
| A04 | 0 | 0 |
| A05 | 0.65 | 0.65 |
| A06 | RES | RES |
| A07 | BP2 | BP2 |
| A08 | IRVO | IRVO |
| A09 | IRHO | IRHO |
| A10 | RMT1 | RMT1 |

| Attribute | Prediction | Reference |
|---|---|---|
| A01 | CR/LFINF | MUR |
| A02 | 1960–1985 | 1960–1985 |
| A03 | 4 | 3 |
| A04 | 0 | 1 |
| A05 | 0.32 | UNKNOWN |
| A06 | COM | MIX |
| A07 | BP2 | BP1 |
| A08 | POP | UNKNOWN |
| A09 | IRHO | UNKNOWN |
| A10 | RMT1 | RMT1 |

| Attribute | Prediction | Reference |
|---|---|---|
| A01 | CR/LFINF | MUR |
| A02 | 1960–1985 | 1960–1985 |
| A03 | 5 | 4 |
| A04 | 0 | UNKNOWN |
| A05 | 0.32 | UNKNOWN |
| A06 | MIX | RES |
| A07 | BP2 | BP1 |
| A08 | IRVO | POP |
| A09 | IRHO | UNKNOWN |
| A10 | RMN | RMT1 |

| Attribute | Prediction | Reference |
|---|---|---|
| A01 | CR/LFINF | MUR |
| A02 | 1985–2000 | 1960–1985 |
| A03 | 2 | 3 |
| A04 | 0 | 0 |
| A05 | 1.95 | UNKNOWN |
| A06 | RES | RES |
| A07 | BPD | BP1 |
| A08 | IRVO | UNKNOWN |
| A09 | IRHO | UNKNOWN |
| A10 | RMN | RMT1 |

**Figure 5.** Examples of classification results from the top-performing model (DINO-v2 B). Attribute names correspond to those in Table 1. The top row shows a best-case scenario where all building attributes are correctly predicted. The bottom row shows a worst-case example with multiple misclassifications. Predicted labels are displayed alongside the corresponding reference values for comparison. Green and red colors highlight correct and wrong predictions, respectively.

### *5.4. Aggregation of Images Capturing Multiple Viewpoints*

The results previously reported are computed considering a single image for each building as input and the respective model predictions as output. As discussed in

Section 3.1, each building is represented by one or more street-level images taken from different viewpoints. Some buildings have a single image, while others are captured from two or three distinct perspectives.

In this experiment, we aggregate the model predictions from all available views of the same building to study its impact on the classification results. Specifically, for each attribute, we collect the probability scores produced by the model across all views and apply either "max" or "mean" aggregation. The aggregated probability vector is then used to select the most probable class.

Table 5 reports the results obtained by only considering the 1168 buildings with three photos taken. When no aggregation is applied (first row of each group in the table), the performance for each building is computed by randomly selecting one of its three images. The obtained results show a general improvement in performance when predictions are aggregated across multiple views of the same building.

**Table 5.** Performance evaluation for each model using max or mean aggregation. All values are in %. Only buildings represented by three images are considered here. For each group, the first row represent the results of no aggregation (a single image is randomly selected for each building). The second and third rows show the performance improvement/deterioration compared to no aggregation by aggregating the scores of the three images representing the same buildings. Color scale is from red (deterioration) to green (improvement).

| Model | Aggregation | A01 | A02 | A03 | A04 | A05 | A06 | A07 | A08 | A09 | A10 | Avg. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ResNet-18 | No | 51.11 | 69.39 | 48.25 | 75.94 | 29.63 | 49.24 | 67.48 | 59.91 | 53.48 | 66.54 | 57.10 |
| | Max | +2.55 | −0.82 | +3.52 | +0.86 | +0.50 | −2.78 | +1.46 | +1.58 | +0.81 | +1.62 | +0.93 |
| | Mean | +2.54 | −1.14 | +5.15 | +1.57 | +3.91 | −0.82 | +2.18 | +1.32 | +0.81 | +0.74 | +1.62 |
| ResNet-50 | No | 56.73 | 66.70 | 45.43 | 75.77 | 20.88 | 45.52 | 66.73 | 59.71 | 59.23 | 69.73 | 56.64 |
| | Max | +2.41 | +1.51 | +1.70 | +0.57 | +4.09 | +4.04 | +1.65 | +0.57 | +2.25 | +0.86 | +1.97 |
| | Mean | +1.90 | +1.70 | +2.29 | +0.44 | +4.18 | +3.45 | +0.65 | +0.49 | +2.25 | +2.39 | +1.98 |
| ResNet-152 | No | 56.30 | 63.02 | 48.07 | 77.01 | 28.20 | 45.68 | 68.65 | 58.79 | 60.45 | 71.52 | 57.77 |
| | Max | −0.87 | +0.73 | +3.62 | +1.81 | −2.16 | −0.88 | +2.11 | +2.62 | −4.16 | +1.85 | +0.47 |
| | Mean | +1.56 | +0.81 | +3.49 | +1.70 | −1.07 | +0.44 | +1.14 | +0.09 | −4.21 | +1.62 | +0.56 |
| DenseNet-121 | No | 52.99 | 66.68 | 48.09 | 75.39 | 27.09 | 43.40 | 67.83 | 60.51 | 60.43 | 69.92 | 57.23 |
| | Max | +2.37 | +1.55 | +4.64 | +0.20 | +1.89 | +1.06 | +1.76 | +0.84 | −4.24 | −1.95 | +0.82 |
| | Mean | +2.33 | +0.76 | +5.02 | +0.06 | +2.29 | −0.71 | +1.30 | +0.85 | −3.24 | −0.55 | +0.81 |
| EfficientNet-B4 | No | 44.74 | 64.41 | 43.45 | 75.46 | 24.08 | 43.74 | 66.64 | 60.98 | 62.85 | 68.68 | 55.50 |
| | Max | +4.61 | +1.23 | +7.88 | +0.15 | +5.71 | +2.78 | +3.49 | +2.30 | −5.32 | +3.52 | +2.64 |
| | Mean | +4.34 | +0.94 | +8.81 | +0.94 | +4.47 | +1.89 | +2.30 | +1.20 | −1.99 | +2.30 | +2.52 |
| Inception-v3 | No | 53.03 | 66.54 | 48.06 | 77.17 | 25.79 | 40.87 | 70.08 | 62.39 | 56.23 | 69.65 | 56.98 |
| | Max | +0.46 | +1.01 | +4.12 | +0.27 | +2.20 | −2.78 | +1.43 | +0.02 | +0.01 | +1.05 | +0.78 |
| | Mean | +0.42 | +0.92 | +4.81 | +0.78 | +2.01 | +0.58 | +0.42 | +0.01 | +0.06 | +3.73 | +1.37 |
| MobileNet-v3 | No | 47.78 | 59.54 | 48.01 | 74.51 | 28.58 | 45.09 | 67.30 | 56.30 | 61.13 | 69.23 | 55.75 |
| | Max | +4.93 | +3.08 | +4.67 | +1.12 | −0.81 | +1.89 | +1.63 | +1.38 | −5.93 | −1.15 | +1.08 |
| | Mean | +9.56 | +3.27 | +5.91 | +1.07 | +0.56 | +3.29 | +1.03 | +5.12 | +0.40 | +1.42 | +3.16 |
| SqueezeNet | No | 45.29 | 51.87 | 42.91 | 74.29 | 26.26 | 35.82 | 63.81 | 53.01 | 53.46 | 63.77 | 51.05 |
| | Max | +5.71 | +10.47 | +4.95 | +0.88 | +6.44 | +0.61 | +2.71 | +5.49 | −2.55 | +0.68 | +3.54 |
| | Mean | +2.18 | +8.77 | +5.55 | +1.39 | +6.85 | +1.63 | +4.26 | +3.95 | −1.65 | +0.69 | +3.36 |
| DINO-v2 S | No | 52.63 | 63.24 | 40.86 | 72.20 | 20.63 | 49.08 | 66.40 | 54.93 | 65.37 | 69.70 | 55.50 |
| | Max | +2.82 | +3.76 | +2.10 | +2.18 | +2.63 | +6.64 | +1.34 | +3.30 | −3.11 | +1.34 | +2.30 |
| | Mean | +5.76 | +4.13 | +3.81 | +2.23 | +6.20 | +9.12 | +1.84 | +3.32 | −3.11 | +2.15 | +3.55 |
| DINO-v2 B | No | 51.68 | 62.94 | 46.15 | 74.71 | 19.56 | 57.60 | 69.43 | 56.58 | 65.82 | 72.26 | 57.67 |
| | Max | +3.42 | +5.55 | +2.15 | +0.78 | +7.17 | +2.51 | +1.09 | +5.33 | −1.04 | +0.37 | +2.74 |
| | Mean | +2.85 | +6.15 | +4.08 | −0.39 | +6.34 | +3.72 | +1.68 | +5.79 | −1.09 | +1.18 | +3.03 |

Considering average accuracy, both mean and max aggregation led to better results compared to using single-view predictions. This performance gain ranges from +0.47% in ResNet-152 max aggregation to +3.55% in DINO-v2 S mean aggregation. Observing the performance changes for every single attribute, we can notice that almost all the attributes benefit from aggregation. Some attributes, such as A03 (number of storeys) and A05 (height of the ground floor), show a considerable performance improvement. This may be due to the fact that these attributes are inherently difficult to infer from a single viewpoint, as perspective distortions or partial occlusions can hide key visual cues. Instead, attribute A09 (horizontal irregularities) shows a decrease in performance when predictions are aggregated. This behavior may be related to the view-dependent nature of horizontal irregularities, which may only be visible from specific angles, leading to errors when multi-view aggregation is used. Overall, DINO-v2 B confirms to be the best model, with an average accuracy of 60.70%, 3.03% higher than the one obtained without prediction aggregation.

These results suggest that multi-view aggregation can considerably improves the final model predictions on almost all the attributes leading to more accurate overall classification.

Figure 6 illustrates an example of a building captured from three different viewpoints. In this case, excluding attribute A05 (height of the ground floor), which is `UNKNOWN`, DINO-v2 B correctly predicted at most 8 out of 9 attributes when using any single image. However, when predictions from all three views are aggregated, all the attributes are correctly classified.


| Attribute | Prediction 1 | Prediction 2 | Prediction 3 | Aggregated | Reference |
|---|---|---|---|---|---|
| A01 | CR/LFINF | CR/LFINF | CR/LFINF | CR/LFINF | CR/LFINF |
| A02 | 1960–1985 | 1960–1985 | 1985–2000 | 1960–1985 | 1960–1985 |
| A03 | 9 | 7 | 9 | 9 | 9 |
| A04 | 0 | 1 | 0 | 0 | 0 |
| A05 | 0.52 | 0.12 | 0.32 | 0.52 | `UNKNOWN` |
| A06 | RES | MIX | MIX | MIX | MIX |
| A07 | BP2 | BP2 | BP2 | BP2 | BP2 |
| A08 | IRVO | IRVO | IRVO | IRVO | IRVO |
| A09 | IRHO | IRHO | IRHO | IRHO | IRHO |
| A10 | RMT1 | RTM1 | RMT1 | RMT1 | RMT1 |

**Figure 6.** Example of attribute classification results using mean aggregation of DINO-v2 B predictions for a building captured from three different street-level viewpoints. Attribute names correspond to those in Table 1. Predicted labels are displayed alongside the corresponding reference values for comparison. Green and red colors highlight correct and wrong predictions, respectively.

## 6. Conclusions

In this study, we demonstrated the effectiveness of deep learning for the automated classification of building attributes relevant to seismic risk assessment. Using the Alvalade dataset, we evaluated a diverse set of architectures, including CNNs and Vision Transformers, to assess their suitability for the development of large-scale exposure models. The obtained results highlight the potential of deep learning to accelerate the development of exposure models by reducing reliance on, for example, manual building

surveys and enabling faster, more consistent extraction of key structural attributes from street-level imagery.

Among the compared models, DINO-v2 B achieved the highest macro accuracy and also delivered the best recall performance. ResNet-18 and ResNet-152 outperformed other architectures in terms of precision and F1-score, respectively. DenseNet-121 and ResNet-18 offered the best trade-off between performance and efficiency, achieving competitive accuracy at a significantly lower computational cost, an important consideration for real-world, large-scale deployment.

Our findings also show that deep learning models can sometimes infer attributes that are not directly visible in single-perspective imagery by exploiting contextual visual cues and correlations captured during training.

The results also show that aggregating information from multiple viewpoints of the same building improves model accuracy. This finding suggests that incorporating complementary data sources, such as aerial imagery, could further improve attribute prediction, especially for structural features that are difficult to capture from a street-level perspective. We identify this multi-view integration as a key direction for future research, with the potential to significantly increase the completeness and reliability of seismic exposure models.

We recognize a limitation of this study is its reliance on a single dataset. Although it is currently the only publicly available resource tailored to this task, it is limited in size and geographically specific. These factors may constrain the generalization of our findings. Future work to address this limitation may include the collection of new datasets covering multiple regions, architectural styles, and urban contexts. Extending the Alvalade dataset with these data would improve model robustness and the applicability of automated building attribute classification across diverse environments.

**Author Contributions:** Conceptualization, R.K., C.R., F.P. and G.C.; methodology, R.K., C.R., F.P. and G.C.; formal analysis, R.K., C.R., F.P. and G.C.; data collection, R.K. and C.R.; data curation, R.K. and C.R.; software, R.K. and C.R.; validation, R.K. and C.R.; visualization, R.K. and C.R.; writing—original draft, R.K.; writing—review and editing, R.K., C.R., F.P. and G.C.; supervision, G.C.; funding acquisition, G.C. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research is part of the PRIN–PNRR project SMILE: Statistical Machine Learning for Exposure Development, funded by the European Union—Next Generation EU, Mission 4 Component 1 (CUP F53D23010780001).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** All the data used in this work are publicly available.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Dong, L.; Shan, J. A comprehensive review of earthquake-induced building damage detection with remote sensing techniques. *ISPRS J. Photogramm. Remote Sens.* **2013**, *84*, 85–99. [CrossRef]
2. Crowley, H.; Despotaki, V.; Rodrigues, D.; Silva, V.; Toma-Danila, D.; Riga, E.; Karatzetzou, A.; Fotopoulou, S.; Zugic, Z.; Sousa, L.; et al. Exposure model for European seismic risk assessment. *Earthq. Spectra* **2020**, *36*, 252–273. [CrossRef]
3. Zhang, L.W.; Lu, Z.H.; Chen, C. Seismic fragility analysis of bridge piers using methods of moment. *Soil Dyn. Earthq. Eng.* **2020**, *134*, 106150. [CrossRef]
4. Gonzalez, D.; Rueda-Plata, D.; Acevedo, A.B.; Duque, J.C.; Ramos-Pollán, R.; Betancourt, A.; García, S. Automatic detection of building typology using deep learning methods on street level images. *Build. Environ.* **2020**, *177*, 106805. [CrossRef]
5. Pavić, G.; Hadzima-Nyarko, M.; Bulajić, B.; Jurković, Ž. Development of seismic vulnerability and exposure models—A case study of Croatia. *Sustainability* **2020**, *12*, 973. [CrossRef]

6. Pelizari, P.A.; Geiß, C.; Aguirre, P.; Santa María, H.; Peña, Y.M.; Taubenböck, H. Automated building characterization for seismic risk assessment using street-level imagery and deep learning. *ISPRS J. Photogramm. Remote Sens.* **2021**, *180*, 370–386. [CrossRef]
7. Iturburu, L.; Liu, X.; Zhang, X.; Wogen, B.E.; Villamizar, J.N.; Dyke, S.J.; Ramirez, J.; Choi, J.B.; Valencia, G.; Alcocer, S.M. Building pose detection for the characterization of reinforced concrete buildings. *Struct. Des. Tall Spec. Build.* **2024**, *33*, e2120. [CrossRef]
8. Gomez, D.; Charris, D.; Percybrooks, W.; Arteta, C.A. Automating building typology identification for seismic risk assessment using deep learning. *Earthq. Spectra* **2025**, *41*, 1833–1862.
9. Tocchi, G.; Polese, M.; Di Ludovico, M.; Prota, A. Regional based exposure models to account for local building typologies. *Bull. Earthq. Eng.* **2022**, *20*, 193–228. [CrossRef]
10. Silva, V.; Sousa, R.; Ribeiro Gouveia, F.; Lopes, J.; Guerreiro, M.J. A building imagery database for the calibration of machine learning algorithms. *Earthq. Spectra* **2024**, *40*, 1577–1590. [CrossRef]
11. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
12. Oquab, M.; Darcet, T.; Moutakanni, T.; Vo, H.; Szafraniec, M.; Khalidov, V.; Fernandez, P.; Haziza, D.; Massa, F.; El-Nouby, A.; et al. Dinov2: Learning robust visual features without supervision. *arXiv* **2023**, arXiv:2304.07193.
13. Li, Z.; Liu, F.; Yang, W.; Peng, S.; Zhou, J. A survey of convolutional neural networks: Analysis, applications, and prospects. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *33*, 6999–7019.
14. Harirchian, E.; Hosseini, S.E.A.; Jadhav, K.; Kumari, V.; Rasulzade, S.; Işık, E.; Wasif, M.; Lahmer, T. A review on application of soft computing techniques for the rapid visual safety evaluation and damage classification of existing buildings. *J. Build. Eng.* **2021**, *43*, 102536. [CrossRef]
15. Xie, Y.; Ebad Sichani, M.; Padgett, J.E.; DesRoches, R. The promise of implementing machine learning in earthquake engineering: A state-of-the-art review. *Earthq. Spectra* **2020**, *36*, 1769–1801. [CrossRef]
16. Sun, H.; Burton, H.V.; Huang, H. Machine learning applications for building structural design and performance assessment: State-of-the-art review. *J. Build. Eng.* **2021**, *33*, 101816. [CrossRef]
17. Rueda-Plata, D.; González, D.; Acevedo, A.; Duque, J.; Ramos-Pollán, R. Use of deep learning models in street-level images to classify one-story unreinforced masonry buildings based on roof diaphragms. *Build. Environ.* **2021**, *189*, 107517. [CrossRef]
18. 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.* **2022**, *36*, 04022031. [CrossRef]
19. Gouveia, F.; Silva, V.; Lopes, J.; Moreira, R.S.; Torres, J.M.; Simas Guerreiro, M. Automated identification of building features with deep learning for risk analysis. *Discov. Appl. Sci.* **2024**, *6*, 466. [CrossRef]
20. Bhatta, S.; Dang, J. Seismic damage prediction of RC buildings using machine learning. *Earthq. Eng. Struct. Dyn.* **2023**, *52*, 3504–3527. [CrossRef]
21. Yepes-Estrada, C.; Calderon, A.; Costa, C.; Crowley, H.; Dabbeek, J.; Hoyos, M.C.; Martins, L.; Paul, N.; Rao, A.; Silva, V. Global building exposure model for earthquake risk assessment. *Earthq. Spectra* **2023**, *39*, 2212–2235. [CrossRef]
22. Bittner, K.; Cui, S.; Reinartz, P. Building extraction from remote sensing data using fully convolutional networks. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2017**, *42*, 481–486. [CrossRef]
23. Zheng, L.; Ai, P.; Wu, Y. Building recognition of UAV remote sensing images by deep learning. In Proceedings of the IGARSS 2020—2020 IEEE International Geoscience and Remote Sensing Symposium, Waikoloa, HI, USA, 26 September–2 October 2020; pp. 1185–1188.
24. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. In Proceedings of the Advances in Neural Information Processing Systems 28, Montreal, QC, Canada, 7–12 December 2015.
25. Zhang, L.; Wu, J.; Fan, Y.; Gao, H.; Shao, Y. An efficient building extraction method from high spatial resolution remote sensing images based on improved mask R-CNN. *Sensors* **2020**, *20*, 1465. [CrossRef]
26. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
27. Ma, H.; Liu, Y.; Ren, Y.; Wang, D.; Yu, L.; Yu, J. Improved CNN classification method for groups of buildings damaged by earthquake, based on high resolution remote sensing images. *Remote Sens.* **2020**, *12*, 260.
28. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2818–2826.
29. Huang, X.; Ren, L.; Liu, C.; Wang, Y.; Yu, H.; Schmitt, M.; Hänsch, R.; Sun, X.; Huang, H.; Mayer, H. Urban building classification (ubc)—A dataset for individual building detection and classification from satellite imagery. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 1413–1421.
30. Zhuang, F.; Qi, Z.; Duan, K.; Xi, D.; Zhu, Y.; Zhu, H.; Xiong, H.; He, Q. A comprehensive survey on transfer learning. *Proc. IEEE* **2020**, *109*, 43–76. [CrossRef]

31. Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; Li, F.-F. Imagenet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255.
32. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
33. Tan, M.; Le, Q. EfficientNet: Rethinking model scaling for convolutional neural networks. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 10–15 June 2019; pp. 6105–6114.
34. Howard, A.; Sandler, M.; Chu, G.; Chen, L.C.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V.; et al. Searching for MobileNetV3. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 1314–1324.
35. Iandola, F.N.; Han, S.; Moskewicz, M.W.; Ashraf, K.; Dally, W.J.; Keutzer, K. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size. *arXiv* **2016**, arXiv:1602.07360.
36. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems 30 (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017.
37. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
38. Cui, Y.; Jia, M.; Lin, T.Y.; Song, Y.; Belongie, S. Class-balanced loss based on effective number of samples. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 9268–9277.