A COMPARISON STUDY ON DEEP LEARNING MODELS FOR BUILDING ROOFTOP CLASSIFICATION

Angel Spasov [1] *, Dessislava Petrova-Antonova [1, 2], Emil Hristov [1]

[1] GATE Institute, Sofia University, Sofia, Bulgaria – (angel.spasov, dessislava.petrova, emil.hristov)@gate-ai.eu
[2] Faculty of Mathematics and Informatics, Sofia University, Sofia, Bulgaria – d.petrova@fmi.uni-sofia.bg

KEY WORDS: Rooftop Classification, Convolutional Neural Networks, 3D City Models.

ABSTRACT:

The availability of semantic information about a cityscape is essential for understanding and analysing urban processes. Automatic gathering of such information is important due to the enormous amount of data. A great number of building features could be gained solely by visual inspections. Therefore, it is meaningful to utilize recent advancements in automatic image recognition technologies to extract these properties automatically.
This paper proposes an optimized solution for the classification of rooftops from aerial imagery based on a deep learning model using Convolutional Neural Networks (CNNs). It describes the architecture of the network, the training procedure and important hypermeters. A model analysis using advanced interpretability and explainability tools is conducted. The model's superiority is demonstrated by comparing its performance against several state-of-the-art image classification architectures, including CNN-based ones such as Xception and Efficientnet, pure Visual Transformers (ViTs) based architectures such as BEiT, and hybrid architectures.

1. INTRODUCTION

Automatic image recognition plays a crucial role in understanding and analysing urban processes, particularly in the context of the built environment. By harnessing recent advancements in deep learning and Convolutional Neural Networks (CNNs), it becomes possible to automatically extract valuable information from urban imagery, facilitating effective decision-making and urban analysis. In this regard, building rooftop classification holds significant importance as it provides essential semantic information about the cityscape.

Accurate modelling of buildings and their rooftops is essential for various applications, including infrastructure and service planning, solar potential estimation, green roof analysis, and social space assessment. 3D city models and City Digital Twins (CDTs) in general replicate the physical environment of a city and enable comprehensive analysis of urban processes. Proper modelling of buildings at different levels of detail (LOD) is crucial for generating detailed 3D city models and functional CDTs. Rooftop modelling, classified at LOD2 or LOD3, enhances the visual perception of 3D city models and facilitates various urban analyses (Biljecki et al., 2015; Julin et al., 2018; Suszanowicz, 2019; Shao et al., 2021; Pomeroy, 2012).

To achieve accurate rooftop modelling, access to high-quality data is vital, including aerial imagery and semantic information. The latter can be extracted through automatic image recognition methods. Recent research has focused on developing optimized deep-learning models for automatic rooftop classification, leveraging the capabilities of CNNs. These models can efficiently extract building features solely through visual inspection, thereby improving the modelling process within CDTs (Spasov, 2021; Castagno, 2018; Cai et al., 2021).

This paper proposes an enhanced approach for categorizing rooftops from aerial images, utilizing a CNN deep learning model. It outlines the network's structure, the training process, and significant hyperparameters. The fine-tuned model weights trained on images of Sofia (Bulgaria) are shared on GitHub (2023). Advanced tools are employed for model analysis to ensure model interpretability and explainability. Moreover, the performance of the proposed model is compared against several state-of-the-art image classification architectures, including CNN-based models like Xception and Efficientnet, pure Visual Transformers (ViTs) such as BEiT, and hybrid architectures.

The rest of the paper is organised as follows. Section 2 presents the methodology applied in the study. Section 3 shows the results obtained from the proposed fine-tuned model compared to other state-of-the-art image classification models. Finally, Section 4 concludes the paper and outlines future work.

2. METHODOLOGY

This section describes the methodology followed for the rooftop classification, including data preparation and labelling, model selection and optimisation and its performance evaluation.

2.1 Data Preparation

The classification models utilised in this study have deep learning architectures and are trained using a supervised learning approach. Depending on the problem to be solved, i.e., the object to be identified and classified, supervised models could require a substantial amount of data to achieve high (classification) performance. For example, the high optical variability of objects belonging to the same class (intra-class variability) and the high visual similarity of objects containing different classes make the classification task more difficult to solve. Other factors are image quality (such as resolution, noise and illumination) and the proportion of objects of interest to the area of the entire image. In addition, the unambiguity of the objects to be classified, as well as the presents of a single object of interest on an image, are prerequisites for a single-class prediction. Considering these

* Corresponding author

factors, the data definition, collection and all preprocessing steps applied in this study are carefully selected and performed.

A dataset consisting of 3,517 rooftop images encompassing the district "Lozenets" of Sofia was employed for the study (Hristov, 2023). It is derived from a solitary orthophoto, made available in TIFF format and represented in the RGB colour space. The orthophoto was acquired in 2020 using aerial photography techniques, employing an ultra-wide range digital camera. The acquisition process involved a longitudinal overlap of 60% and a transverse overlap of 30% to ensure comprehensive coverage and accurate representation of the district. Notably, the orthophoto's Ground Sampling Distance (GSD) was 10 cm, which is considered highly detailed and distinctive for an urban environment like the city of Sofia.

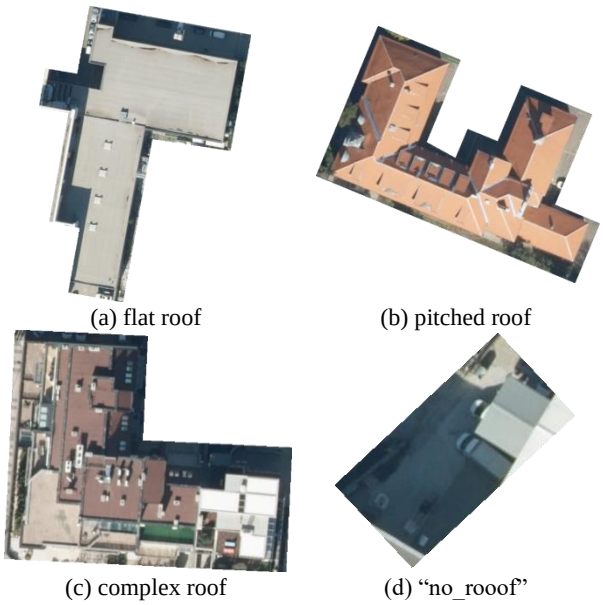
The preparation of the dataset involved a meticulous process executed in multiple steps to support the classification models. Initially, a QGIS plugin named Mapflow is used to localise the buildings from the orthoimage. This automated procedure helped to identify the approximate outlines of the buildings based on the available data. Subsequently, a manual adjustment is performed to refine the inferred buildings' outlines. Third, the orthoimage is tiled based on the resulting outlines from the previous step aiming to extract each building in a separate image. Specifically, the applied tiling rule produces images containing detected building boundaries with an additional outer buffer of 2 meters (see Figure 1). This buffer ensures that the extracted images encompass the whole outlines of each building, separating neighbouring structures in another tile.

Figure 1. Generated 2 m. buffer (orange outlines) by the roof outlines (purple).

By following this procedure, the resulting dataset was optimally prepared for a one-class prediction. The combination of automated techniques and manual refinement allowed for the creation of a comprehensive dataset efficiently. This dataset serves as the foundation for training and evaluating the models in the current study.

2.2 Data Labelling

The study area is distinctive for its complex architecture, characterised by various roof shapes. Therefore, a careful selection labelling strategy is essential to balance intra-class variability and inter-class similarity, which is also advantageous for the purpose of the model. Based on this consideration, the single rooftop images are classified into three main classes, namely “pitched”, “flat”, and “complex”. An additional helper class, “no_roof”, is introduced to cover cases where the image doesn’t represent a roof. Figure 2 shows examples from the four classes.



(a) flat roof (b) pitched roof

(c) complex roof (d) “no_rooof”

Figure 2. Sample roof classes.

The “flat” roof class encompasses completely flat roofs with a minimal slope (see Figure 2a). Key identifying features typically include a simple rectangular shape, perpendicular angles, uniformity in terms of pixels, colours, and the absence of distinct planes. It is important to note that buildings with flat roofs spanning multiple levels are also included in this category, which may lead to potential overlap with the complex roof category.

The „pitched” roof class includes all sloped roof types, including hip and gable roofs, and their various configurations (see Figure 2b). Roofs are considered part of this class regardless of the number of planes they comprise, as long as they possess a sloping structure. The criteria and key features utilised for the classification of a roof as pitched include the presence of hips and ridges, which form clear demarcation lines between the planes. An identifiable diagonal hip line and darker or shaded planes on the opposite side of the ridge serve as indicators for a pitched roof.

The “complex” roof class covers roofs that incorporate a combination of pitched and flat geometry. Additional criteria for inclusion in this category include roofs with multiple levels and terraces, roofs with intricate shapes featuring numerous slopes, and roofs with oval or spherical forms. A roof is classified as complex when multiple buildings with distinct roof types and varying shapes share walls, giving the appearance of a unified roof area or building.

The “no_ruf” class incorporates images that do not illustrate buildings, including construction sites, unclear or blurry images, extremely small sections of roofs, or shapes that are inherently unidentifiable to the human eye.

The above considerations aim to minimise ambiguity among the images and overlap between the classes and act as annotation rules. Furthermore, following them consistently while annotating is important, since the inconsistency would provide contradictive examples for the models to learn from. Consequently, the classification performance is affected due to the relative amount of such cases to the size of the dataset and the respective classes.

Given the extensive range of roof architectures, however, a certain level of subjectivity and partial overlap are inevitable.

2.3 Model Selection and Optimisation

For the current classification task, a widely used ResNet architecture is selected. This type of architecture has been successfully utilised as a feature extractor and serves as a backbone for various image recognition tasks such as classification, detection and segmentation. The ResNet architectures contain residual functions, so-called "identity blocks", which effectively tackle the problem of vanishing gradients. This problem refers to the fact that the gradients of deep neural networks become increasingly small as they propagate backwards. As a result, the network is unable to update its parameters effectively. The ResNet architectures provide an effective solution for this phenomenon by incorporating an additional connection called a skip connection or residual connection in the network. It allows the network to "choose" whether to use a learned transformation or to simply propagate its input to the next layer if this is the optimal solution.

Several experiments are conducted using different model sizes, namely ResNet18, ResNet50 and ResNet101. All three CNNs consist of similar building blocks, composed of convolutional layers, pooling layers, normalisation layers and Rectified Linear Unit (ReLU) activations. The main difference between the networks is in the number of building blocks, presented in brackets in Table 1, and consequently, the number of learnable parameters (He, 2016). ResNet18 is the smallest network with ca. 11.7 million parameters and ResNet101 with ca. 44.5 million parameters. The output of these networks for an image is a 512 or 2048-dimensional feature vector, which is a dense representation of the image. Based on these feature vectors a fully connected layer assigns a "score-value" to each of the classes.

Layer name	Conv1	Conv2_x	Conv3_x	Conv4_x	Conv5_x
ResNet 18	7x7, 64, stride 2	3x3 max pool, stride 2	$\begin{bmatrix}3x3,64\\3x3,64\end{bmatrix}\times 2$	$\begin{bmatrix}3x3,128\\3x3,128\end{bmatrix}\times 2$	$\begin{bmatrix}3x3,256\\3x3,256\end{bmatrix}\times 2$	$\begin{bmatrix}3x3,512\\3x3,512\end{bmatrix}\times 2$
ResNet 34			$\begin{bmatrix}3x3,64\\3x3,64\end{bmatrix}\times 3$	$\begin{bmatrix}3x3,128\\3x3,128\end{bmatrix}\times 4$	$\begin{bmatrix}3x3,256\\3x3,256\end{bmatrix}\times 6$	$\begin{bmatrix}3x3,512\\3x3,512\end{bmatrix}\times 3$
ResNet 50			$\begin{bmatrix}1x1,64\\3x3,64\\1x1,256\end{bmatrix}\times 3$	$\begin{bmatrix}1x1,128\\3x3,128\\1x1,512\end{bmatrix}\times 4$	$\begin{bmatrix}1x1,256\\3x3,256\\1x1,1024\end{bmatrix}\times 6$	$\begin{bmatrix}1x1,512\\3x3,512\\1x1,2048\end{bmatrix}\times 3$
ResNet 101			$\begin{bmatrix}1x1,64\\3x3,64\\1x1,256\end{bmatrix}\times 3$	$\begin{bmatrix}1x1,128\\3x3,128\\1x1,512\end{bmatrix}\times 4$	$\begin{bmatrix}1x1,256\\3x3,256\\1x1,1024\end{bmatrix}\times 23$	$\begin{bmatrix}1x1,512\\3x3,512\\1x1,2048\end{bmatrix}\times 3$
ResNet 152			$\begin{bmatrix}1x1,64\\3x3,64\\1x1,256\end{bmatrix}\times 3$	$\begin{bmatrix}1x1,128\\3x3,128\\1x1,512\end{bmatrix}\times 8$	$\begin{bmatrix}1x1,256\\3x3,256\\1x1,1024\end{bmatrix}\times 36$	$\begin{bmatrix}1x1,512\\3x3,512\\1x1,2048\end{bmatrix}\times 3$
Output size	112x112	56x56	28x28	14X14	7x7

Table 1. Architectures for ImageNet (He, 2016).

ResNet101 showed slightly better performance. Even though the training time was around 30% slower for the same number of epochs compared to ResNet50 and around 50% slower than ResNet18. Nevertheless, the larger network is selected due to the still relatively short training time (under 3 minutes for 15 epochs with early stopping rules on a single RTX 3080 NVIDIA GPU with 16GB RAM) and the prioritised performance.

The network is initialised with parameters (weights) pre-trained on the ImageNet database. ImageNet is a large dataset consisting of images with diverse objects and backgrounds. Its characteristics make it particularly suitable for Transfer Learning where knowledge obtained from one domain is applied to a new domain. The parameters could be used without finetuning in the convolutional blocks or at initialisation. In the latter case, models pre-trained on this dataset show faster convergence and often better performance than those without using pre-trained weights. In CNNs, earlier layers of the networks extract more general low-level features, whereas deeper layers extract more domain-specific high-level features. Therefore, finetuning the deeper layers solely is often advantageous when there is a specific domain and initial weights trained on a large, diverse dataset are available. This is the case in the current study. Since ImageNet does not contain areal imagery, finetuning of the deeper convolutional layers should lead to better performance. Experiments are conducted with and without finetuning the ResNet backbone. However, finetuning the last convolutional block increased the performance significantly; thuss, the next iterations in the hyperparameter tuning processes are conducted with finetuning of this block. Figure 3 illustrates the ResNet101 architecture with the finetuned blocks.

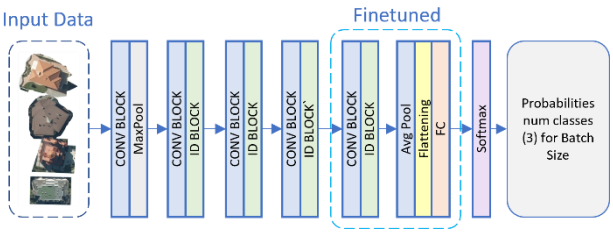


Figure 3. ResNet101 architecture with the finetuned blocks.

An extensive hyperparameter tuning is performed, experimenting with elements such as loss function, regularisation techniques, data augmentation and optimisation strategies. Selected values in the experiments were based on common practices and the results of the previous iterations in a semi-manual manner utilising grid search optimization strategies as well. In the following, the main elements of the optimized training design are presented.

2.3.1 Optimisation Loss: The network is optimized using a Negative Log Loss in combination with the Log of Softmax, which minimising is equivalent to maximising the entropy of the classification. The following implementation of the loss is used:

$$\sum_{n=1}^{N} l_n, l_n = -w_{y_n} log \frac{\exp(x_{n,y_n})}{\sum_{c=1}^{C} \exp(x_{n,c})} \quad (1)$$

where
x is the input
y is the target
w is the weight
C is the number of classes
N spans the minibatch dimension.

2.3.2 Optimiser: The Stochastic Gradient Descent (SGD) and Adam optimizer are tested with different initial learning rates and learning rate scheduling tactics, including annealing techniques such as stepwise, cosine and warm restart cosine annealing. With neglectable impact on performance, the final model used "reduced on plateau" scheduling with 3e-4 initial learning rate.

2.3.3 Data Augmentation: The network is trained with the stated details for 100 epochs with and without data augmentation. The vertical and horizontal flip is applied and in addition, random rotation, augmentation of brightness, contrast, saturation and hue of the images is performed. With augmentation, the training loss converged similarly as in the case where no augmentation was applied. However, the variance was significantly reduced with the application of the augmentation techniques.

2.4 Performance Evaluation

The open-source machine learning platform Mlflow is used for tracking model performance during the research and development phase (Gundersen, 2022). The system allows for logging experiments and better comparison of different versions of models and datasets. The overall performance on the dataset as well as for each class separately, was assessed using precision, recall and F1-score using the following definitions.

- Class-based Precision: How many of the predictions for a class were correctly predicted
$$Precition = \frac{TP}{FP+FP}, \quad (2)$$
- Class-based Recall: How many of the examples belonging to a certain class were correctly predicted
$$Recall = \frac{TP}{TP+FN}, \quad (3)$$
- Class-based F1-score: Harmonic mean of Precision and Recall
$$F1 = 2TP2TP + FP + FN, \quad (4)$$
- Dataset-based accuracy: How many of all predictions were correct
$$Accuracy = \frac{TP}{FP+FN}, \quad (5)$$
- Average precision over all classes in the dataset
$$\sum_{n_i} Accuracy_n, \quad (6)$$
- Recall and F1-score over all classes in the dataset
$$\sum_{n_i} F1_n, \quad (7)$$
- Weighted F1-score
$$\sum_i^N F1 \times \frac{k_i}{l}, \quad (8)$$

where *TP* is a true positive prediction
FP is a false positive prediction
FN is a false negative prediction
n denotes a class
Ki is the number of samples in a class
l is the number of all samples.

Several analytical and interpretability techniques are applied to explore the model's performance more in-depth. For each class and model, True Positives, False Positives and False Negatives (correct and misclassified images) are visually inspected. This helps in understanding what "confuses" the model and finding misclassified images and is especially useful for decisions regarding data cleaning and data enrichment. To understand what is an image exactly contributes most to a certain class prediction a type of Gradient-weighted Class Activation Map (CAM) – GradCAM++ has been implemented (Chattopadhay, 2018). This method generates a saliency map showing which special pixels have the largest contribution to a class prediction. It is based on the gradients of the last convolutional layer's kernels and the resulting feature maps. This method is especially useful to analyse whether a model makes its predictions based on the right features, such as characterising a certain class. In addition, a TracInCP (Pruthi, 2020) is applied to find the most influential train images for a given prediction. This algorithm calculates the influential score for a given train example on a specified test image. This is achieved by estimating the change in the loss on the test image when the given train example is removed and the model retrained. In this case, one can find the train images with the most positive score – the proponents; as well as with the most negative score – opponents.

The final model is compared to other state-of-the-art models on a randomly selected training-validation split. The models were selected so that different types of deep learning models are covered. Their weights were pretrained on ImageNet1k or ImageNet22k. CNN models tested are Xception (Chollet, 2016) and EfficientNet (Tan, 2019). Xception is reported to show slightly better performance than ResNet on some benchmark datasets. The architecture introduces modified separable depth-wise convolution (a depth-wise convolution followed by a pointwise convolution) first introduced by the Inception model. EfficientNet on the other hand is developed to strive for optimal trade-off between model size, computational efficiency and model performance. It uses the concept of compound scaling, which systematically scales the network's depth, width and resolution simultaneously. The optimized ResNet model is also compared against pure Visual Transformer – ViT (Dosovitskiy, 2020) and BEiT (Bao, 2021) and hybrid architectures incorporating Visual Transformer (Steiner, 2021) and ResNet backbone.

3. RESULTS

The final optimized ResNet model shows consistent overall results over the 5-fold cross-validation with average accuracy of 85%, average F1-score of 85%, average precision of 84%, and average recall of 84%. The observation suggests that the model performance on the dataset is independent of the exact images in the train and validation split. The results of each validation split are depicted in Table 2.

Fold	Accuracy	Weighted F1-score	Average F1-score	Precision	Recall
Fold 1	87%	87%	85%	88%	87%
Fold 2	88%	85%	88%	87%	85%
Fold 3	84%	84%	81%	82%	82%
Fold 4	83%	85%	82%	82%	82%
Fold 5	83%	85%	82%	84%	82%

Table 2. Results from the 5-fold cross-validation.

To compare the performance to the other selected deep learning models, a random train validation split is performed. The optimized ResNet showed the highest performance across all observed metrics. It achieved an accuracy of 84.8%, a weighted F1-score of 85.2%, an average F1-score of 82.7% an average precision of 84.1% and a recall of 82.0% on the validation dataset. BEiT showed slightly lower performance with an Accuracy of 83.0% and F1-score of 80.1%. The other visual transformers ranked thereafter. The other CNN-based deep learning models, Xception and EfficientNet performed worse with 79.2% and 77.0% Accuracy and F1-score of 78.5 and 76.3 respectively. Table 3 summarizes the results.

Model	Accuracy	Average F1-score	Average Precision	Average Recall
ResNet101, Fine Tuned	84.8%	82.7%	84.1%	82.0%
BEiTv2	83.0%	80.1%	81.4%	79.5%
Hibrid ViT with ResNet50	82.3%	80.0%	81.1%	79.2%
ViT	82.0%	79.5%	81.4%	79.1%
Xception65	79.2%	78.5%	79.0%	77.5%
EfficientNet_b7	77.0%	76.3%	76.9%	76.1%

Table 3. Comparison of models' performance.

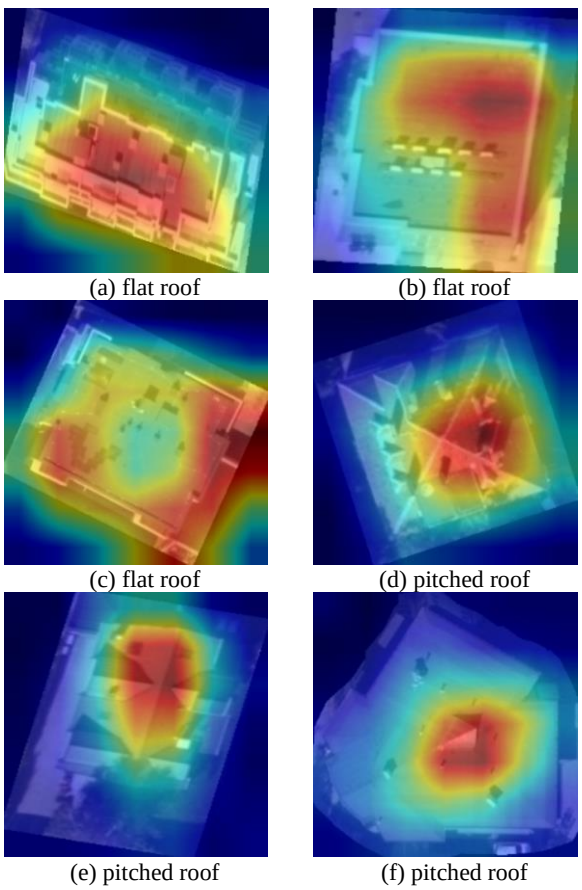
The performance metrics declined slightly once the additional class “bugs” was added to the classification task and dataset. The accuracy dropped to 82.6% and the weighted F1-score to 83.7% from the previous 84.8% and 85.2%, respectively. Out of the 64 validation samples of the class “bugs”, 19 were confused with flat roofs and 9 with pitched roofs. This is understandable, considering that bugs were mostly rectangle-like shapes such as shadows, football fields or started constructions.

The performance of the optimised ResNet shows differences between the individual classes. The higher accuracy is achieved for the class “pitched”, namely 91%. For the “flat” class 81% of the samples were predicted as annotated and for the class “complex” this was 77% of the samples. Table 4 shows the confusion matrix on the validation data.

	Number of samples	Complex	Flat	Pitched
Complex	169	77%	11%	12%
Flat	211	10%	81%	9%
Pitched	410	4%	5%	91%

Table 4. Confusion matrix on the validation data.

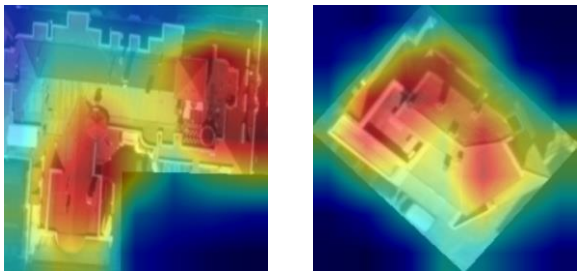
Further exploration of the CAMs generated by GradCAM++ reveals the regions of each image that most contributed to the inference made. Looking at the CAMs of the correct prediction for the “flat” and “pitched” classes, it seems that these regions are distinctive for the respective class. Figure 4 shows examples of the CAMs for these categories. For the “flat” class, the roof features that contributed most are the flat parts of the roofs, often less covered with additional roof elements or the outlines of the roofs. For the “pitched” class, mostly the ridges or the hips of the roofs were highlighted by the GradCAM++.



(a) flat roof (b) flat roof (c) flat roof (d) pitched roof (e) pitched roof (f) pitched roof

Figure 4. Class Activation Maps for flat and pitched roofs.

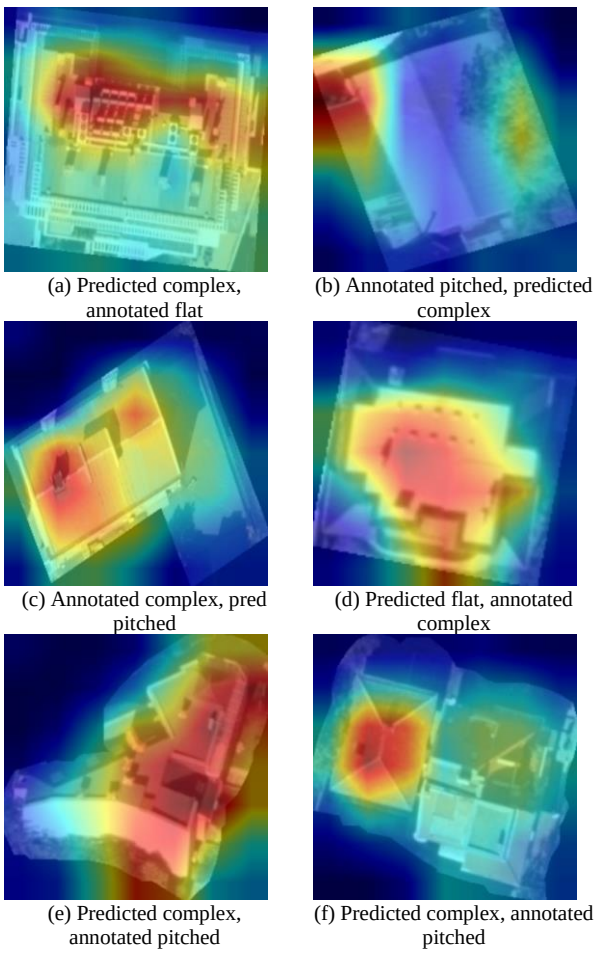
For the “complex” class, the regions contributing the most to the correct result often cross a more significant part of a roof. This is in line with the fact that the complexity of the geometry and architecture of the roofs comes from the combination of different roof styles across the roof (see Figure 5).



(a) (b)

Figure 5. Class Activation Maps for complex roofs.

Analysis of GradCAM++ heatmaps of the misclassified samples gives insights into the reason for the misclassification. In most cases, the reasons lie in the ambiguity of the images. For example, roofs mainly composed of a flat part with a very small gable part could be annotated as complex but predicted by the model as flat and vice versa. Examples of misclassified roofs are shown in Figure 6.

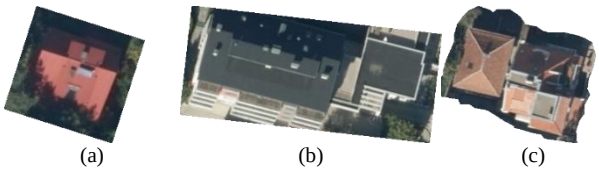


(a) Predicted complex, annotated flat (b) Annotated pitched, predicted complex (c) Annotated complex, pred pitched (d) Predicted flat, annotated complex (e) Predicted complex, annotated pitched (f) Predicted complex, annotated pitched

Figure 6. Class Activation Maps for misclassified roofs.

In addition, observing the individual misclassified samples, it is found that they are the most ambiguous roof types or, in some cases, those that are misannotated. Examples of such annotations are shown in Figure 7.

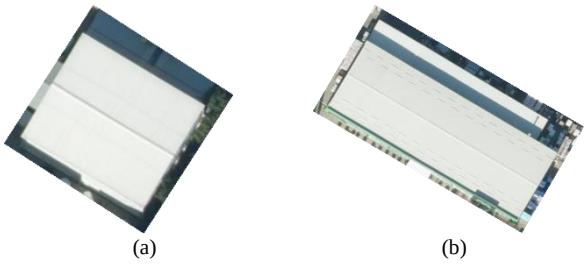
https://doi.org/10.5194/isprs-archives-XLVIII-1-W2-2023-47-2023



(a) (b) (c)

Figure 7. Misclassified roofs.

A more profound observation could reveal specific roofs that are misclassified. For example, white-pitched roofs are incorrectly recognised as flat roofs (see Figure 8). This might be because there are only two white-pitched roofs in the dataset.



(a) (b)

Figure 8. Incorrectly recognised white roofs.

4. CONCLUSION AND FUTURE WORK

This paper demonstrates the model development process for rooftop classification. The developed deep learning model demonstrated a solid ability to differentiate between the three categories, namely "flat", "pitched", and "complex. It achieved performance metrics outperforming other state-of-the-art classification model architectures with 84.8% accuracy and 85.2% class-weighted F1-score. The interpretability analysis demonstrates that it predicts precisely based on correct features. It reveals that most misclassifications are due to ambiguity in the samples concerning the defined classes. Adding a class for detecting objects wrongly identified as buildings slightly decreased performance. The model is developed and tested on the highly diverse architectural landscape of the city of Sofia. In the case of simpler urban architecture, significantly higher classification results can be expected. Testing the model with and without additional training on different georgical locations would potentially prove this hypothesis. Future work includes training and evaluating the model on different image types, such as satellite imagery. Retraining the model on combined images of different resolutions will raise its applicability.

ACKNOWLEDGEMENTS

This research is part of the GATE Project supported by the Horizon 2020 WIDESPREAD-2018-2020 TEAMING Phase 2 Programme under Grant Agreement No. 857155 and by Operational Programme Science and Education for Smart Growth under Grant Agreement No. BG05M2OP001-1.003-0002-C0. The authors gratefully acknowledge the support of the Scientific Fund of Sofia University, under agreement No. 80-10-209/22.05.2023.

REFERENCES

Bao, H., Dong, L., Piao, S., & Wei, F. 2021: BEiT: BERT Pre-Training of Image Transformers. ArXiv. /abs/2106.08254

Biljecki, F., Stoter, J., Ledoux, H., Zlatanova, S., Çöltekin, A. 2015: Applications of 3D City Models: State of the Art Review. *ISPRS International Journal of Geo-Information*, no. 4, pp. 2842-2889.

Cai, Y., He, H., Yang, K., Fatholahi, S. N., Ma, L., Xu, L., & Li, J., 2021: A comparative study of deep learning approaches to rooftop detection in aerial images. [Une étude comparative des approches d'apprentissage en profondeur pour la détection des toits dans les images aériennes] *Canadian Journal of Remote Sensing*, 47(3), 413-431. https://doi.org/10.1080/07038992.2021.1915756.

Castagno, J., Atkins, E., 2018: Roof Shape Classification from LiDAR and Satellite Image Data Fusion Using Supervised Learning. *Sensors* 2018, 18, 3960. https://doi.org/10.3390/s18113960.

Chattopadhay, A., Sarkar, A., Howlader, P., Balasubramanian, V. N. 2018: Gradcam++: Generalized gradient-based visual explanations for deep convolutional networks. In 2018 IEEE winter conference on applications of computer vision (WACV), pp. 839-847.

Chollet, F. 2016: Xception: Deep Learning with Depthwise Separable Convolutions. *ArXiv*. /abs/1610.02357

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. 2020: An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. ArXiv. /abs/2010.11929.

Fine-tuned model weights on GitHub, https://github.com/enerhy/rooftop-classification (18 June 2023).

Julin, A., Jaalama, K., Virtanen, J.-P., Pouke, M., Ylipulli, J., Vaaja, M., Hyyppä, J., 2018: Characterizing 3D City Modeling Projects: Towards a Harmonized Interoperable System. *ISPRS International Journal of Geo-Information*, vol. 7, no. 55.

He, K., Zhang, X., Ren S., Sun J., 2016: Deep Residual Learning for Image Recognition. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016 pp. 770-778. https://doi.org/10.1109/CVPR.2016.90

Hristov, E., Petrova-Antonova, D., Petrov, A., Borukova, M., Shirinyan, E. 2023: Remote Sensing Data Preparation for Recognition and Classification of Building Roofs. *Data* 2023, 8, 80. https://doi.org/10.3390/data8050080.

Gundersen, O. E., Shamsaliei, S., Isdahl, R.J. 2022: Do machine learning platforms provide out-of-the-box reproducibility? *Future Generation. Computer Systems* 2022, 126, pp. 34-47.

Pomeroy, J., 2012: Room at the Top—The Roof as an Alternative Habitable / Social Space in the Singapore Context. *Journal of Urban Design*, vo. 17, no. 3, pp. 413-424, http://dx.doi.org/10.1080/13574809.2012.666176, 2012.

Pruthi, G., Liu, F., Kale, S., Sundararajan, M. 2020: Estimating training data influence by tracing gradient descent. 34th Conference on Advances in Neural Information Processing Systems, pp. 1-11.

Steiner, A., Kolesnikov, A., Zhai, X., Wightman, R., Uszkoreit, J., & Beyer, L. 2021: How to train your ViT? Data, Augmentation, and Regularization in Vision Transformers. ArXiv. /abs/2106.10270.

Shao, H., Song, P., Mu, B., Tian, G., Chen, Q., He, R., Kim, G., 2021: Assessing city-scale green roof development potential using Unmanned Aerial Vehicle (UAV) imagery. *Urban Forestry & Urban Greening*. vol. 57. https://doi.org/10.1016/j.ufug.2020.126954.

Spasov, A., Petrova-Antonova, D., 2021. Transferability assessment of open-source deep learning model for building detection on satellite data. International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences, pp. 107-110, https://doi.org/10.5194/isprs-archives-XLVI-4-W4-2021-107-2021.

Suszanowicz, D., Ratuszny, P., Wróbel, R., 2019. The potential of roofs in city centers to be used for photovoltaic micro-installations. IOP Conference Series Materials Science and Engineering, vol. 564(1). http://dx.doi.org/10.1088/1757-899X/564/1/012128.

Tan, M., & Le, Q. V. 2019: EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. *ArXiv*. /abs/1905.11946.