

# Buildings Classification using Very High Resolution Satellite Imagery

Mohammad Dimassi<sup>a</sup>, Abed Ellatif Samhat<sup>a</sup>, Mohammad Zaraket<sup>b</sup>, Jamal Haidar<sup>b</sup>, Mustafa Shukor<sup>c</sup>, Ali J. Ghandour<sup>d,\*</sup>

<sup>a</sup>CRSI, Faculty of Engineering, Lebanese University — {md22.dimassi@gmail.com, samhat@ul.edu.lb}

<sup>b</sup>CCE, Faculty of Engineering, Islamic University of Lebanon — {mohzrkt99@gmail.com, jamal.haidar@iul.edu.lb}

<sup>c</sup>MINES ParisTech, Paris, France — mustafa.shukor@mines-paristech.fr

<sup>d</sup>National Center for Remote Sensing - CNRS, Beirut, Lebanon — aghandour@cnrs.edu.lb

---

## Abstract

Buildings classification using satellite images is becoming more important for several applications such as damage assessment, resource allocation, and population estimation. We focus, in this work, on buildings damage assessment (BDA) and buildings type classification (BTC) of residential and non-residential buildings. We propose to rely solely on RGB satellite images and follow a 2-stage deep learning-based approach, where first, buildings' footprints are extracted using a semantic segmentation model, followed by classification of the cropped images. Due to the lack of an appropriate dataset for the residential/non-residential building classification, we introduce a new dataset of high-resolution satellite images. We conduct extensive experiments to select the best hyper-parameters, model architecture, and training paradigm, and we propose a new transfer learning-based approach that outperforms classical methods. Finally, we validate the proposed approach on two applications showing excellent accuracy and F1-score metrics.

**Keywords:** Deep learning, CNN, Transfer Learning, Buildings Type Classification, Building Damage Classification.

---

## 1. Introduction

Buildings classification is vital for many applications, such as buildings damage assessment (BDA) and buildings type classification (BTC). Urban areas are constantly struck by man-made and/or natural disasters, such as wars, tornadoes, and earthquakes, resulting in large-scale buildings and urban infrastructure destruction. In the early reconstruction phase, damage assessment is conducted manually using crucial information about the area, amount, rate, and type of damage. In addition, buildings type classification (*e.g.*, residential/non-residential) pave the way for many real-world applications such as population estimation and resource allocation.

For these purposes, remote sensing techniques can play an important role, mainly due to their wide availability at relatively low cost, wide field of view, and fast response capacities. Using deep learning for building classification can speed up the process by reducing human intervention and saving considerable time and cost. Indeed, we witnessed in past years a rapid development in the field of deep learning and its applications in earth observation, remote sensing, and computer vision fields [1].

Contrary to other work that proposes to tackle such task by adopting one model for semantic segmentation with many classes [2, 3], here we follow a 2-stage approach where we disentangle the semantic segmentation from the classification. In a nutshell, a semantic segmentation model takes an input RGB image and predicts buildings' masks (*i.e.*, stage 1). In the second stage, buildings are cropped from the original image and

fed to a classification model to predict the class of each building. In addition, we propose to use only RGB satellite images, which is more efficient than using additional other modalities.

This paper focuses on the second classification stage and conducts extensive experiments to find the best hyper-parameter, training paradigms, and model architecture for the underlying task. Moreover, we propose a new transfer learning paradigm that extends the classical 2-stage approach (*i.e.*, pre-training then fine-tuning) by an additional stage that makes the model's layers more consistent and specific to the task, and we show that this outperforms the classical approach. We validate the proposed approach on two main tasks (*i.e.*, BDA and BTC), showing excellent performance in terms of accuracy and F1-score metrics. Finally, we propose a new dataset for BTC. To avoid redundancy, we optimized the hyperparameters and the training paradigm for BDA while we focused on the architecture for BTC. The list of contributions are the following:

- We propose to use only RGB satellite images for buildings classification. We validate the approach on two main tasks; Buildings type classification (BTC) and Buildings damage assessment (BDA).
- We propose a new dataset for BTC.
- We conduct extensive experiments to choose the best hyperparameters, training paradigm, and model architecture.
- We propose a new transfer learning paradigm that outperforms the classical 2-stage approach.

---

\*Corresponding Author.  
Email: aghandour@cnrs.edu.lb

The rest of this paper is organized as follows: Section 2 reviews some of the related works. Section 3 presents a brief background and details the methodology adopted in this work. In Sections 4 and 5, we validate the proposed approach using Building Damage Assessment and Building Damage Assessment applications, respectively, and discuss the proposed dataset. Finally, Section 6 concludes this manuscript.

## 2. Related Work

### 2.1. Buildings Damage Assessment

A vision-based approach for detecting cracks on concrete images is discussed in [4] using deep learning techniques. Based on Convolution Neural Network (CNN), the idea is to determine cracks in a specific zone such as a building, especially in the roof, and make decisions that help classify the building as either damaged or not. Authors in [5] assess the impact of the combined use of different resolution satellite images on improving classification accuracy of damaged buildings using CNN. In [6], the proposed method also detects Flooded/Damaged buildings from satellite imagery of an area affected by a hurricane. The authors prepared their dataset from online sources and applied several pre-processing steps before the training phase on the TensorFlow framework. The study in [7] proposes an algorithm for building damage detection from post-event aerial imagery, using a data expansion Single Shot multi-box Detector (SSD) algorithm for a small data set of Hurricane Sandy. The proposed algorithm relies on Feed Forward Neural Network (FFNN) and uses VGG-16 [8] as the primary network to extract feature information. In [9], the authors use CNN on a small set of candidates damaged buildings to reduce needed processing time. Deep learning is also used in [10] to improve the detection of rooftop hail damage. One CNN classifier is trained from scratch, while the second classifier relies on the set features from a pre-trained network. In addition, different input size images were tested to check the best one.

Moreover, researchers from various disciplines such as civil engineering, infrastructure, and mechanical engineering developed CNN models to detect different types of damage. The authors in [11] and [12] generate damage features' map using data extracted from sensors. In [11], authors provide structural damage localization with good accuracy using both noise-free and noisy datasets. In [12], as the variation of the temperature alters the structural model parameters, a damage detection technique is proposed to consider both uncertainties and varying temperatures. It is developed on the basis of Sparse Bayesian Learning (SBL). An application for an unsupervised learning approach was presented in [13] where the authors propose a transfer learning approach to use a source network trained on a labeled dataset to be able to train the target network on an unlabeled dataset.

In all previous work in the literature related to damaged buildings, the optimization of the hyper-parameters for the deep learning model was barely investigated.

### 2.2. Buildings Type Classification

In [14], authors use an ensemble of machine learning models to classify buildings as sprayable and not-sprayable based on buildings characteristics such as size, shape, and proximity to neighboring features. Similarly, in [15], authors use classical machine learning approaches to predict if the building is residential or not based on several input variables (stored in shape-files, CSV ...).

Work presented in [16] propose to classify buildings type using geospatial data (*e.g.*, point-of-interest (POI) data, building footprints, land use polygons, and roads) based on NLP and ratio-based techniques. In [17], an iterative clustering method to classify buildings based on spatiotemporal data (*e.g.*, population density and people interaction) is introduced. Random Forest Classifier is used in [18] to classify buildings' footprint from different data sources (*e.g.*, topographic raster maps, cadastral databases, or digital landscape models). Authors in [19] use object-Based Image Analysis (OBIA) and machine learning methods to extract and classify buildings from Airborne Laser Scanner (ALS). Gaussian finite mixture model is proposed in [20] to classify buildings based on several metrics extracted from high-resolution satellite images. Finally, [21] combines street view with satellite images to classify buildings using CNN models.

To the best of our knowledge, no method in the literature is solely based on RGB aerial images to classify buildings types.

## 3. Methodology

The primary motivation behind adopting the 2-stages approach is to fragment the main problem into two more straightforward tasks: semantic segmentation and image classification. This approach helps to leverage the recent advances in those two domains to solve the underlying task. In addition, besides being more efficient and cheaper, we argue that RGB images are enough to classify buildings, as this can be done relatively easily by humans.

In this section, we will focus on the second stage, which is buildings classification. In a nutshell, the model takes an RGB image of a building cropped using the predicted segmentation mask from the first stage and output the class of the building, either damaged or not and either residential or not.

### 3.1. Loss Function

Multiple loss functions exist and can be used during training. In this work, we focus on two of the widely used losses in order to compare their performance and choose the appropriate one for the task: (i) Cross-Entropy loss and (ii) Focal loss [22].

Cross-Entropy loss is defined in Equation 1, where  $t_i$  and  $s_i$  are, respectively, the ground-truth and the score for each class  $i$  in the universe of classes  $C$ .

$$CE = - \sum_i^C t_i \log(s_i) \quad (1)$$

In our case, we are dealing with a binary classification which means that  $C = 2$ , and this is referred to as the Binary Cross

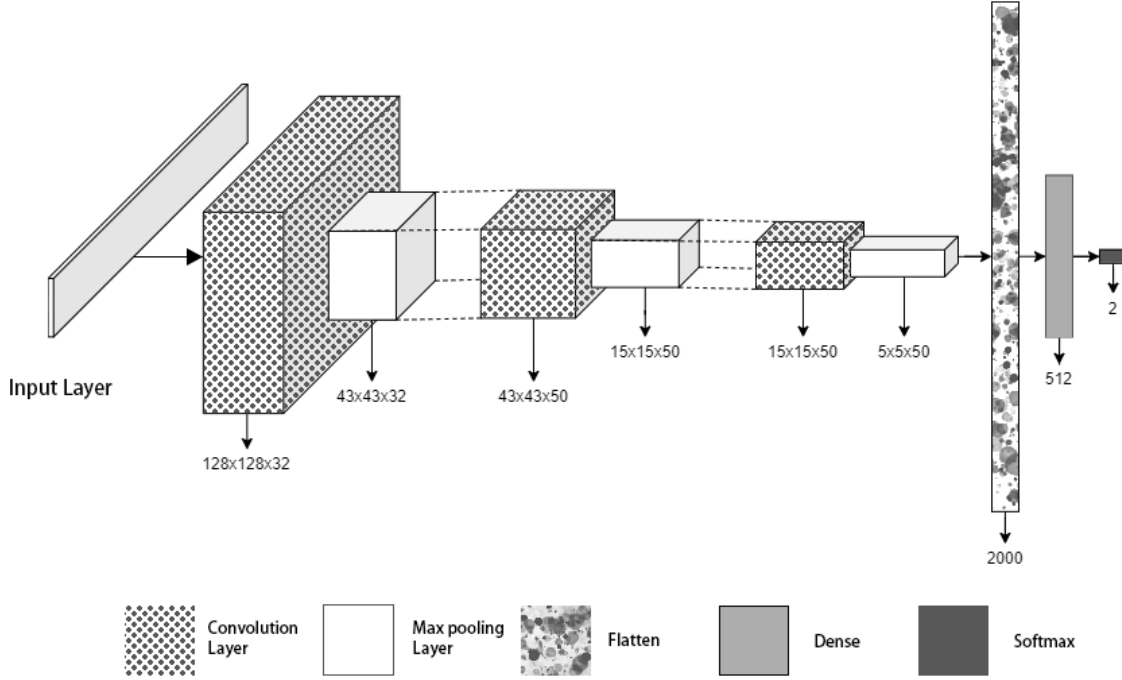


Figure 1: Architecture of the adopted CNN network model for BDA application.

Entropy defined in Equation 2.  $t_1$  and  $s_1$  are, respectively, the ground-truth and the score for class  $C_1$ .

$$BCE = - \sum_{i=1}^{C=2} t_i \log(s_i) = -t_1 \log(s_1) - (1 - t_1) \log(1 - s_1) \quad (2)$$

Focal loss [22] weighs the contribution of each sample to the loss based on the classification error. If a sample is classified correctly by the CNN, its contribution to the loss model should decrease. With this strategy, focal loss solves the problem of hard labels and plays a vital role in imbalanced classes dataset. Binary focal loss is described in Equation 3, where  $(1 - s_i)^\gamma$  is a modulating factor to reduce the influence of correctly classified samples in the loss and  $\gamma \geq 0$  is referred to as the focusing parameter. With  $\gamma = 0$ , focal loss is reduced to binary cross entropy.

$$FL = - \sum_{i=1}^{C=2} (1 - s_i)^\gamma t_i \log(s_i) \quad (3)$$

### 3.2. Optimizer

During CNN training, the role of the optimizer is to update the weight parameters to minimize the loss function. Multiple optimizers are suggested in the literature, such as Momentum, Regular Gradient Descent (RGD), Stochastic Gradient Descent (SGD), Adam, and RectifierAdam for classification tasks [23].

Although non-adaptive optimizers, such as SGD, help obtain better minima and generalization properties. Adam optimizer is widely used as it leads to faster convergence due to

its adaptive learning rate. However, Adam suffers from significant variance at the beginning of training. Rectifier-Adam (also known as RectAdam) [24] was proposed to improve the convergence of Adam.

In a nutshell, no optimizer works best for all the applications; thus, we propose a comparison of these optimizers in this work.

### 3.3. Transfer Learning

Transfer learning is considered an innovative approach in CNN training to achieve high accuracy with minimal time and effort, relying on existing pre-trained model. The idea is to use a model trained on a large dataset and transfer its knowledge to the application at hand. The classical transfer learning approach adopts 2-stage strategies: (i) a model is trained from scratch on a large dataset (*i.e.*, pre-training on ImageNet [25]), then (ii) the last layers of the model, are retrained for the downstream application. Here, we propose to extend this approach by adding a third and final stage: (iii) we freeze the last layers of the network and retrain initial layers (previously froze) in order to ensure the consistency between those layers and the last ones, and also to adjust all the weights according to the specific target application.

### 3.4. Evaluation and Metrics

We are relying on the classical metrics used in this context which are:

- True positive (TP): Both manual and automated methods label the object belonging to the buildings regions.



Figure 2: Sample damaged buildings scenes from the xView dataset.

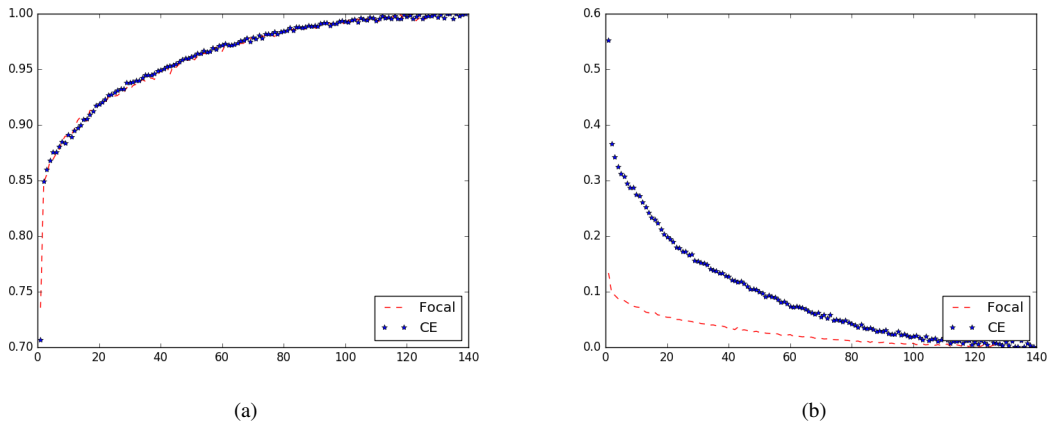


Figure 3: (a) and (b) show the accuracy and loss, respectively, in function of time for BDA: the model is trained using Cross Entropy and Focal Loss functions. From the loss curves, we can notice that the model using Focal Loss converges faster.

- True negative (TN): Both manual and automated methods label the object belonging to the background.
- False positive (FP): The automated method incorrectly labels the object as belonging to the building regions.
- False negative (FN): The automated method does not correctly label a pixel truly belonging to the building regions.

These global definitions can be used to generate the following performance metrics defined in Equations (4), (5) and (6).

$$Recall = \frac{TP}{TP + FN} \quad (4)$$

$$Precision = \frac{TP}{TP + FP} \quad (5)$$

$$F1 - Score = \frac{2 \times Pr \times Rc}{Pr + Rc} \quad (6)$$

Also one of the essential metrics is the accuracy given in equation (7)

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (7)$$

In addition, we report the speed of convergence for each scenario using the hyper-parameter "number of epochs" taken by the network to converge.

## 4. Buildings Damage Assessment

The following two sections detail how the proposed approach can be applied to BDA and BTC applications. For BDA, we focus on the choice of the training paradigm (*e.g.*, Transfer Learning). For BTC, we focus on the architecture.

Buildings Damage Assessment task consists of classifying buildings as damaged or not, in which a classification model takes an RGB image and outputs the image class. This section gives an overview of existing datasets and then explains the implementation details and experimental results.

### 4.1. Dataset

Dataset choice is an essential component of our work. CNN relies on labeled data to train its network and provides a high-accuracy classifier model. Four datasets are considered, analyzed, and compared to choose the appropriate one for our research.

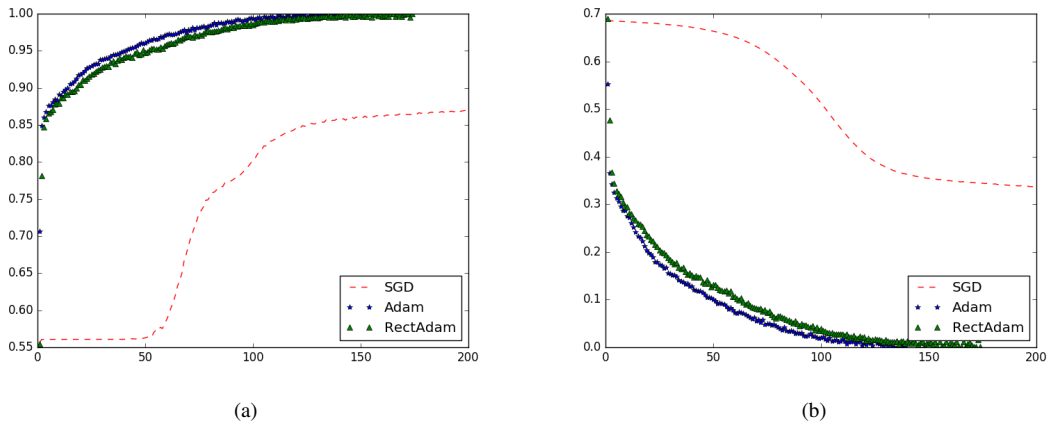


Figure 4: (a) and (b) show the accuracy and loss, respectively, in function of time for BDA: the model is trained with SGD, Adam and RectAdam optimizers. The model trained with SGD struggle to converge. Adam optimizer leads to faster convergence.

Dataset	Number of Objects	Resolution (in pixels)	Format	Generalizable
ABCD dataset [26]	8,500	128x128	.tif	No
Flooding dataset [6]	10,000	128x128	.jpeg	No
NOAA dataset [27]	500,000	9351x9351	.tif	No
xView dataset [28]	1,000,000	3000x3000	.tif	Yes

Table 1: Datasets survey and comparison.

We used several parameters for the datasets benchmark. Damage type was the critical metric for choosing the appropriate dataset. Damage type differs based on the disaster event. War, earthquakes, and wind (hurricanes and tornadoes) usually result in wholly or partly demolished and ruined buildings. Flooding produces water in areas where it is not normally expected, while volcano erupts lava.

For the scope of this work, we define a damaged building as a partly or wholly demolished building. This definition covers the most common damage type resulting from armed conflicts, earthquakes, tornadoes, and hurricanes.

We considered four different datasets as potential candidates for the BDA application: ABCD Dataset [26], Flooding Dataset [6], NOAA Dataset [27] and xView Dataset [28]. Table 1 shows a comparison between these four datasets. One can conclude that xView dataset is the best candidate for our research. xView dataset is well labeled and is big enough so that we will choose it for BDA task. Figure 2 shows sample damaged scenes from the xView dataset.

*xView Dataset.* xView is one of the largest publicly available datasets with annotated images from complex scenes around the world. xView dataset contains more than 1 million labeled objects covering over 1400 Km<sup>2</sup>. Images have an average of (3000x3000) pixels resolution using the “.tif” format and high-quality labeling, which provides a large amount of available imagery to understand the visual world in new ways and address a range of applications, including damaged building classification.

## 4.2. Implementation

For each scenario in the following subsections, we rely on a CNN network made from three layers of Convolutions + Max-Pooling followed by Flatten, Dense, and finally, Softmax layer. The used network is depicted in Figure 1. In our work, we used the Sigmoid activation function for each convolution layer and the ReLU activation function for the dense layer.

We carried an evaluation study over the test data. The convergence condition is to reach a high accuracy value (99%) and low loss threshold ( $\leq 0.001$ ) over the trained data.

## 4.3. Experimental Results

In this section, we present the results for the different scenarios discussed in Section 3 based on the xView dataset.

*Loss Function Results.* In order to choose the best loss function suitable for the damaged building application, we built two models using the architecture described in Figure 1, where the first one uses cross-entropy loss function and the second one uses focal loss function. After training the two models corresponding to cross-entropy loss and focal loss until converging, we present the results during the training phase and for the test dataset in Figure 3 and Table 2. Figures 3 (a) and (b) show the accuracy and the loss as a function of time (number of epochs) for Cross-Entropy and Focal Loss functions. Figure 3a shows clearly that both loss functions converge to 99% accuracy over the trained data, but focal loss converges faster during training as shown in Figure 3b. In addition, Table 2 confirms that focal

	Epochs	Accuracy (%)
Cross Entropy Loss	138	97.33
Focal Loss	<b>127</b>	<b>97.56</b>

Table 2: The model is trained on Cross Entropy and Focal Loss functions for BDA. We can notice that the model with Focal Loss converges faster.

	Epochs	Accuracy (%)
SGD	-	85.05
Adam	<b>138</b>	<b>97.33</b>
RectAdam	174	97.23

Table 3: The model is trained with different optimizers for BDA; we can notice that Adam leads to faster convergence while SGD struggle to converge.

loss converges faster than cross-entropy (127 epochs versus 138 epochs). Finally, we can notice that focal loss scores slightly higher accuracy than cross-entropy over the test dataset. To conclude, focal loss has better results against cross-entropy loss in terms of the following three metrics: accuracy, number of epoch to converge, and loss per epoch. Thus, we deduce that focal loss has a better performance for damaged building classification scope, and thus it will be adopted as part of the following scenario.

*Optimizer Results.* No single optimizer choice can be made for all different CNN applications. For instance, Adam optimizer is widely adopted and used for several applications, but still, in many use cases, SGD has reported better results. RectAdam [24] was introduced as an enhancement for Adam. However, in multiple scenarios, Adam has shown better and faster performance.

Accordingly, and for the scope of this work, we consider three optimizers: (i) SGD, (ii) Adam and (iii) RectAdam. We will compare their suitability for damaged building classification application and select the best option.

The results for SGD, Adam, and RectAdam optimizers during the training phase and for the test dataset using xView, are presented in Table 3 and Figure 4. Figures 4a and b show the accuracy metric and the loss metric, respectively, as a function of time (number of epochs) for SGD, Adam, and RectAdam optimizers.

Results in Figure 4 reveal that SGD struggled to converge and was not able to reach the desired accuracy and loss thresholds during training, whereas Adam and RectAdam optimizers both converge rapidly.

In terms of accuracy, Adam and RectAdam score better result (97.3% and 97.2%, respectively) than SGD (85.1%) as tabulated in Table 3. Regarding the number of epochs needed to converge, Adam is much faster than RectAdam (137 epochs vs. 174 epochs).

These results clearly show that Adam and RectAdam have better performance than SGD in terms of accuracy, the number of epochs to converge, and loss per epoch. We will adopt Adam optimizer for damaged building classification application in the following scenarios.

	Epochs	Accuracy (%)
Baseline	127	97.33
TL-VGG16	<b>23</b>	96.12
E-TL-VGG16 (ours)	69 (23 + 46)	<b>98.96</b>

Table 4: Comparison of different trained paradigms for BDA; ours (E-TL-VGG16) is the best in terms of accuracy and faster to train than the baseline.

*Transfer Learning Results.* Our goal is to apply multiple models based on Transfer Learning (TL) and find the best methodology for training our damaged building classifier using TL. For this sake, we consider the following three models:

- Model 1: network trained from scratch without using any pre-trained weights. We will refer to this model as **Baseline** model.
- Model 2: network based on transfer learning approach using VGG16 pre-trained weights on ImageNet dataset [8] where we freeze all layers and train only the last layers of the network. We will refer to this model as transfer learning - VGG16 (**TL-VGG16**) model.
- Model 3: enhanced transfer learning approach (discussed in Section 3.3) where an additional final step is added. We will refer to this model as enhanced - transfer learning - VGG16 (**E-TL-VGG16**) model.

In this scenario, the three models above are compared and analyzed. Table 4 clearly shows the difference in speed convergence, where TL-VGG16 iterates only for 23 epochs to converge, while the Baseline model needed 127 epochs.

From Table 4, one can notice that TL-VGG16 model converges to the desired thresholds faster than the Baseline model. Since using pre-trained weights would help the network to converge faster. However, this does not necessarily result in higher accuracy, as shown in Table 4. TL-VGG16 achieved 96% accuracy over the test data, which is lower than the 97.5% scored by the Baseline model.

The rationale behind E-TL-VGG16 model is to improve the accuracy of the transfer learning approach while maintaining good enough convergence speed. At epoch 23, when we unfreeze the VGG16 layers and de-freeze the last layers, accuracy metric sharply decreases, and loss sharply increases. After this glitch in performance, the network spends 69 epochs to re-converge to previously attained thresholds.

E-TL-VGG16 performance tabulated in Table 4 reveals an important improvement in the last percentages of the accuracy to reach 98.96%. Finally, we also compute the F1-score achieved by E-TL-VGG16, which turns to be equal to 99.4%.

## 5. Buildings Type Classification

This section elaborates on the approach used to classify buildings into residential or non-residential using only RGB aerial images. This task can pave the way for several applications, such as population estimation and resource allocation.

Due to the lack of an appropriate dataset for such a task, we introduce a new "Beirut Buildings Type Classification" (BBTC) dataset to help the community develop this field further. We start by introducing the dataset then explaining reported experimental results.

### 5.1. BBTC Dataset

Here, we propose a new dataset referred to as Beirut Buildings Type Classification (BBTC) after presenting an overview of existing datasets:

*Related Datasets.* To the best of our knowledge, there is no high-resolution satellite images dataset for classifying buildings into residential/non-residential. Among existing datasets, we found only DSTL dataset [29] that provides 1km x 1km satellite images in both 3-band and 16-band formats. However, DSTL images' resolution is not high (*i.e.*, 1.24 m) and the dataset is highly imbalanced (*i.e.*, 3.1% non-residential buildings).

*BBTC Description.* The dataset consists of 17,033 RGB images, where each image contains a single building and is annotated as residential (15,330 buildings) or non-residential buildings (1,703 buildings which accounts for 10% of annotated objects). The tiles are extracted from high resolution images (*i.e.*, ~ 50 cm) and covers Beirut city. Some examples can be seen in Figure 5.

*BBTC Creation.* The dataset was created and labeled using the following steps:

- buildings polygons extracted from OpenStreetMap as a shapefile.
- annotation done using QGIS software; shapefile placed on two base maps for visual confirmation (Bing Virtual Earth and Google Satellite Hybrid) and each building labeled as residential or non-residential. Entire annotated buildings are almost 20,000 objects. Non-residential objects include universities, schools, governmental facilities, mosques, churches, commercial centers, and industrial facilities.
- corresponding tile for each building requested from Mapbox at zoom level 17.
- finally, buildings cropped (slightly larger than the corresponding polygon to account for any inaccuracy) and saved. Duplicate buildings (*i.e.*, retrieved multiple times) are ignored, reaching a total of 17,033 objects.

BBTC dataset can be accessed via the following bucket url: [https://storage.googleapis.com/bbtc/bbtc\\_dataset.tar.gz](https://storage.googleapis.com/bbtc/bbtc_dataset.tar.gz)

### 5.2. Experimental Results

This section provides and discusses experimental results for BTC.

Model	Avg. time (per epoch)	Val. Acc	Epoch	Test Acc
VGG16	3m20s	0.953	95	0.935
ResNet 50	5m30s	0.947	44	0.933
Res2Net	9m30s	0.960	<b>40</b>	0.943
EfficientNet	<b>3m13s</b>	0.960	49	0.941
RexNet	3m15s	<b>0.962</b>	99	<b>0.948</b>

Table 5: Ablation study for BTC on the proposed dataset; We compare different architectures. RexNet outperforms all the other models in terms of accuracy.

Optimizer	Avg. time (per epoch)	Val. Acc	Epoch	Test Acc
SGD	3m15s	<b>0.962</b>	99	<b>0.948</b>
Adam	3m15s	0.957	90	0.945
RectAdam	3m15s	0.958	<b>68</b>	0.947

Table 6: Ablation study for BTC on the proposed dataset; RexNet is trained with several optimizers. SGD leads to the best accuracy while RectAdam leads to faster convergence.

*Architecture.* We compared 5 architectures including state of the art ones; VGG16, ResNet50 [30], Res2Net [31], EfficientNet-b0 [32] and RexNet [33]. All models trained with CE and SGD optimizers. Table 5 shows that RexNet gives the best results in terms of accuracy at the test set (94.8%), while average time per epoch (3minutes 15 seconds) is almost equal to EfficientNet. This result does not come as a surprise since RexNet is a newly proposed architecture where the authors claim to provide high accuracy while achieving low time complexity comparable to EfficientNet.

*Optimizer.* Finally, we adopt the winning model (*i.e.*, RexNet) and train it using different optimizers. From Table 6, one can notice that SGD optimizer was able to converge, in contrast to previous performance witnessed in Table 3. SGD achieved the best accuracy on the test set (94.8%) compared to Adam and RectAdam, although it is the slowest with convergence time equal to 99 epochs.

## 6. Conclusion

In this paper, we followed a 2-stage approach for buildings classification using only RGB satellite images. We focused here on the second stage, which consists of classification. We conducted extensive experiments to find the best hyperparameters, network architecture, and training paradigm, and we proposed a new transfer learning approach. Due to the lack of appropriate residential/non-residential buildings classification datasets, we proposed BBTC, a new dataset of high-resolution satellite images. We validated the proposed approach on two applications: Buildings Damage Assessment and Buildings Type Classification, which showed outstanding performance in several metrics on xView and BBTC datasets.



Figure 5: Sample buildings' images from the proposed BBTC dataset.

## Conflict of Interest

Declarations of interest: none

## References

- [1] L. Ma, Y. Liu, X. Zhang, Y. Ye, G. Yin, B. A. Johnson, Deep learning in remote sensing applications: A meta-analysis and review, *ISPRS Journal of Photogrammetry and Remote Sensing* 152 (2019) 166–177. doi:<https://doi.org/10.1016/j.isprsjprs.2019.04.015>.
- [2] V. Igloukov, S. Mushinskiy, V. Osin, Satellite imagery feature detection using deep convolutional neural network: A kaggle competition, arXiv preprint arXiv:1706.06169 (2017).
- [3] E. Guérin, K. Oechslein, C. Wolf, B. Martinez, Satellite image semantic segmentation, arXiv preprint arXiv:2110.05812 (2021).
- [4] Y.-J. Cha, W. Choi, O. Buyukozturk, Deep learning-based crack damage detection using convolutional neural networks, *Computer-Aided Civil and Infrastructure Engineering* 32 (2017) 361–378. doi:[10.1111/mice.12263](https://doi.org/10.1111/mice.12263).
- [5] D. Duarte, F. Nex, N. Kerle, G. Vosselman, Satellite Image Classification Of Building Damages Using Airborne And Satellite Image Samples In A Deep Learning Approach, 2nd Edition, Vol. IV, International Society for Photogrammetry and Remote Sensing (ISPRS), 2018, pp. 89–96. doi:[10.5194/isprs-annals-IV-2-89-2018](https://doi.org/10.5194/isprs-annals-IV-2-89-2018).
- [6] Q. D. Cao, Y. Choe, Deep learning based damage detection on post-hurricane satellite imagery (07 2018).
- [7] Y. Li, W. Hu, H. Dong, X. Zhang, Building damage detection from post-event aerial imagery using single shot multibox detector, *Applied Sciences* 9 (6) (2019). doi:[10.3390/app9061128](https://doi.org/10.3390/app9061128). URL <https://www.mdpi.com/2076-3417/9/6/1128>
- [8] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556 (2014).
- [9] F. Nex, D. Duarte, A. Steenbeek, N. Kerle, Towards real-time building damage mapping with low-cost uav solutions, *Remote Sensing* 11 (3) (2019). doi:[10.3390/rs11030287](https://doi.org/10.3390/rs11030287). URL <https://www.mdpi.com/2072-4292/11/3/287>
- [10] M. M. Hezaveh, C. Kanan, C. Salvaggio, Roof damage assessment using deep learning, in: 2017 IEEE Applied Imagery Pattern Recognition Workshop (AIPR), 2017, pp. 6403–6408. doi:[10.1109/AIPR.2017.8457946](https://doi.org/10.1109/AIPR.2017.8457946).
- [11] L. Yizhou, Z. Nie, H. Ma, Structural damage detection with automatic feature-extraction through deep learning, *Computer-Aided Civil and Infrastructure Engineering* 32 (11 2017). doi:[10.1111/mice.12313](https://doi.org/10.1111/mice.12313).
- [12] R. Hou, X. Wang, Q. Xia, Y. Xia, Sparse bayesian learning for structural damage detection under varying temperature conditions, *Mechanical Systems and Signal Processing* 145 (2020) 106965. doi:<https://doi.org/10.1016/j.ymsp.2020.106965>.
- [13] Y. Li, W. Hu, H. Li, H. Dong, B. Zhang, Q. Tian, Aligning discriminative and representative features: An unsupervised domain adaptation method for building damage assessment, *IEEE Transactions on Image Processing* 29 (2020) 6110–6122. doi:[10.1109/TIP.2020.2988175](https://doi.org/10.1109/TIP.2020.2988175).
- [14] H. J. Sturrock, K. Woolheater, A. F. Bennett, R. Andrade-Pacheco, A. Midekisa, Predicting residential structures from open source remotely enumerated data using machine learning, *PloS one* 13 (9) (2018) e0204399.
- [15] C. T. Lloyd, H. J. W. Sturrock, D. R. Leasure, W. C. Jochem, A. N. Lázár, A. J. Tatem, Using gis and machine learning to classify residential status of urban buildings in low and middle income settings, *Remote Sensing* 12 (23) (2020). doi:[10.3390/rs12233847](https://doi.org/10.3390/rs12233847). URL <https://www.mdpi.com/2072-4292/12/23/3847>
- [16] W. Chen, Y. Zhou, Q. Wu, G. Chen, X. Huang, B. Yu, Urban building type mapping using geospatial data: A case study of beijing, china, *Remote Sensing* 12 (17) (2020). doi:[10.3390/rs12172805](https://doi.org/10.3390/rs12172805). URL <https://www.mdpi.com/2072-4292/12/17/2805>
- [17] L. Zhuo, Q. Shi, C. Zhang, Q. Li, H. Tao, Identifying building functions from the spatiotemporal population density and the interactions of people among buildings, *ISPRS International Journal of Geo-Information* 8 (6) (2019). doi:[10.3390/ijgi8060247](https://doi.org/10.3390/ijgi8060247). URL <https://www.mdpi.com/2220-9964/8/6/247>
- [18] R. Hecht, G. Meinel, M. Buchroithner, Automatic identification of building types based on topographic databases – a comparison of different data sources, *International Journal of Cartography* 1 (1) (2015) 18–31. arXiv:<https://doi.org/10.1080/23729333.2015.1055644>, doi:[10.1080/23729333.2015.1055644](https://doi.org/10.1080/23729333.2015.1055644). URL <https://doi.org/10.1080/23729333.2015.1055644>
- [19] M. Belgiu, I. Tomljenovic, T. J. Lampoltshammer, T. Blaschke, B. Höfle, Ontology-based classification of building types detected from airborne laser scanning data, *Remote Sensing* 6 (2) (2014) 1347–1366. doi:[10.3390/rs6021347](https://doi.org/10.3390/rs6021347). URL <https://www.mdpi.com/2072-4292/6/2/1347>
- [20] W. C. Jochem, D. R. Leasure, O. Pannell, H. R. Chamberlain, P. Jones, A. J. Tatem, Classifying settlement types from multi-scale spatial patterns of building footprints, *Environment and Planning B: Urban Analytics and City Science* 48 (5) (2021) 1161–1179. arXiv:<https://doi.org/10.1177/2399808320921208>, doi:[10.1177/2399808320921208](https://doi.org/10.1177/2399808320921208). URL <https://doi.org/10.1177/2399808320921208>
- [21] J. Kang, M. Körner, Y. Wang, H. Taubenböck, X. X. Zhu, Building instance classification using street view images, *ISPRS journal of photogrammetry and remote sensing* 145 (2018) 44–59.
- [22] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 2980–2988.
- [23] S. Ruder, An overview of gradient descent optimization algorithms, arXiv preprint arXiv:1609.04747 (2016).
- [24] L. Liu, H. Jiang, P. He, W. Chen, X. Liu, J. Gao, J. Han, On the variance of the adaptive learning rate and beyond, in: International Conference on Learning Representations, 2019.
- [25] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 248–255. doi:[10.1109/CVPR.2009.5206848](https://doi.org/10.1109/CVPR.2009.5206848).
- [26] A. Fujita, K. Sakurada, T. Imaizumi, R. Ito, S. Hikosaka, R. Nakamura, Damage detection from aerial images via convolutional neural networks, in: 2017 Fifteenth IAPR International Conference on Machine Vision Applications (MVA), 2017, pp. 5–8. doi:[10.23919/MVA.2017.7986759](https://doi.org/10.23919/MVA.2017.7986759).
- [27] Hurricane harvey: Emergency response imagery of the surrounding regions, <https://storms.ngs.noaa.gov/storms/harvey/index.html>.
- [28] D. Lam, R. Kuzma, K. McGee, S. Dooley, M. Laielli, M. Klaric, Y. Bulatov, B. McCord, xviv: Objects in context in overhead imagery, arXiv preprint arXiv:1802.07856 (2018).
- [29] Dstl satellite imagery feature detection: Can you train an eye in the sky?, <https://www.kaggle.com/c/dstl-satellite-imagery-feature-detection/data>.
- [30] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.
- [31] S.-H. Gao, M.-M. Cheng, K. Zhao, X.-Y. Zhang, M.-H. Yang, P. Torr, Res2net: A new multi-scale backbone architecture, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43 (2) (2021) 652–662. doi:



[10.1109/TPAMI.2019.2938758](https://doi.org/10.1109/TPAMI.2019.2938758).

- [32] M. Tan, Q. Le, Efficientnet: Rethinking model scaling for convolutional neural networks, in: International Conference on Machine Learning, PMLR, 2019, pp. 6105–6114.
- [33] D. Han, S. Yun, B. Heo, Y. Yoo, Rethinking channel dimensions for efficient model design, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 732–741.