



Bayesian Optimization of Convolutional Neural Networks for Categorizing the Height and Nonregularities of Low to Mid-Rise Buildings in Google Street View

Paolo Giancarlo Guzman, Diocel Harold Aquino,
Ammiel Mac Barros and Karlo Daniel Colegio

EasyChair preprints are intended for rapid dissemination of research results and are integrated with the rest of EasyChair.

November 9, 2022

Bayesian Optimization of Convolutional Neural Networks for Categorizing the Height and Nonregularities of Low to Mid-rise Buildings in Google Street View

Paolo Giancarlo Guzman,^{1[0000-0002-2377-1292]} Diocel Harold Aquino¹, Ammiel Mac Barros¹, and Karlo Daniel Colegio¹

¹ Institute of Civil Engineering, University of the Philippines Diliman, Quezon City, Philippines

Abstract. More efficient methods for mitigating disaster and risk are constantly sought in earthquake-prone countries. In the Philippines, numerous frameworks for assessing building vulnerability have been integrated into common practice and continue to be developed until now. The process of visual assessment, however, may still be further expedited to avoid additional costs and effort in the assessment process. The current study capitalizes on the growing field of machine learning and seeks to find out if the visual assessment of buildings can be done using a trained convolutional neural network. The research classified Google Street View images of buildings in the Greater Metro Manila Area according to their height; identified out-of-plane setbacks, soft stories, split levels, and short columns; and entered this data into an optimized ResNet50 network. The hyperparameters were obtained through Bayesian optimization, and its performance was compared to a base network with training hyperparameters obtained from a past research. A total of 2100 images were obtained. The results showed that (1) there were significant imbalances in the overall image data set; and (2) the optimized networks were able to best identify three out of the five classifications, excluding soft stories and short columns. The main source of error was associated with the lack of statistical analysis as only the averages of accuracies and F1 scores of the networks were compared; the study may thus be unrepresentative of the accuracy of the trained networks. Hence, it is recommended that future studies apply techniques to mend the class imbalances and perform statistical analysis via student's t-test or ANOVA (among others) for a more grounded conclusion. Nonetheless, there is promising potential for Bayesian optimization in automating building categorization, and it remains to be a systematic process for obtaining hyperparameters for classification tasks.

Keywords: Seismic Risk, Building Classification, Height, Nonregularities, Deep Learning, Convolutional Neural Network, Bayesian Optimization

1 Introduction

1.1 Background of the Study

Disaster risk management methods have always been a necessity and will remain to be so for as long as natural disasters exist. The Philippines is no exception to this fact especially with regards to seismic risks, due to geological hazards such as fault zones present in the country (Philippine Institute of Volcanology and Seismology [PHIVOLCS] et al., 2014). Historically, the Philippines has experienced numerous large-scale earthquakes along these fault zones (Fukushima, et al., 2019), mapping out miscellaneous vulnerability points on the country.

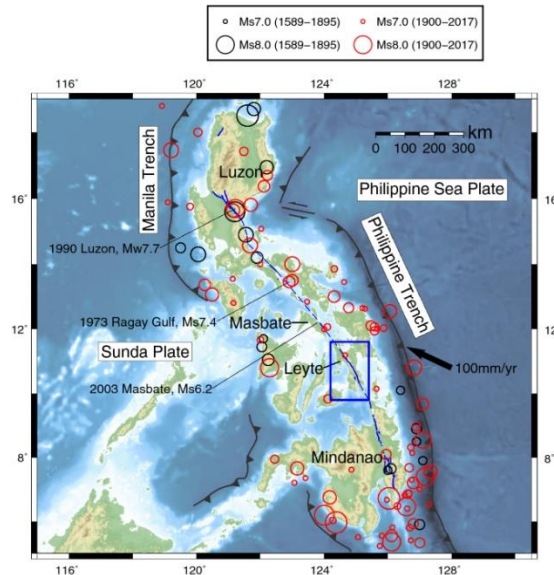


Fig. 1. History of large earthquakes in the Philippines with fault traces labelled in blue lines (Fukushima et al., 2019).

Available data on seismic risks such as these have enabled the derivation of key building type groupings in certain areas for ease of comprehensive analysis (Pacheco et al., 2014). These typologies have allowed for convenient seismic performance analysis through visual assessment.

However, it may not always be practical to physically go to buildings to assess their vulnerability, as evidenced by the COVID-19 pandemic, where mobility was significantly reduced. This opens an opportunity for the automation of visual assessment through Google Street View and Neural Networks (Gonzalez et al., 2020, Yu et

al., 2020, Escobedo, 2017, Bezak, 2016). However, with the complexity of deep learning methods such as these, it is important to optimize their hyperparameters in order to reduce the errors a network makes, and maximize its efficiency (Wu et al., 2019).

Ultimately, the success of a tool that rapidly classifies buildings according to said typologies could catalyze faster and more efficient preliminary seismic vulnerability assessment in the Philippines.

1.2 Research Problem and Objectives

The overall goal of this research was to find the optimal configuration of a convolutional neural network for detecting physical characteristics of buildings in the Philippines. This was achieved through the attainment of the following objectives:

1. Create a database of images of buildings in the Greater Metro Manila for the training and validation of the networks, with each image having labels that have been manually inputted;
2. Create ResNet50 networks for classifying the building images according to their height and vertical nonregularities; and
3. Optimize the hyperparameters of the networks and compare the performance of the optimized networks to the base networks.

1.3 Significance of the Study

The reduced mobility brought about by safety protocols enforced during the COVID-19 pandemic has urged many to seek more efficient alternatives in seismic risk management. In this case, a network that automatically classifies buildings could not only make risk assessment faster by classifying masses of images at once, but also rule out the necessity of having to travel to assess buildings.

As engineers, safety should always be the topmost priority. Therefore, one must always endeavor to innovate more efficient ways to manage risk assessment, more avenues for retrofitting and building resilience, and ultimately creating the means to help communities thrive.

1.4 Scope and Limitations

Images of the buildings to be used for training and validation were taken from Google Street View, and were taken from the areas of Metro Manila, Bulacan, Cavite, Laguna, and Rizal – all areas included in the Greater Metro Manila Area Risk Assessment Project (GMMA-RAP) and the subsequent scope of the antecedent study.

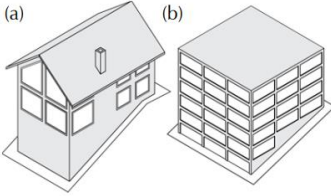
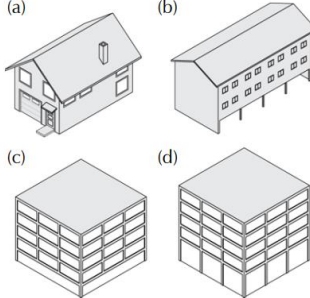
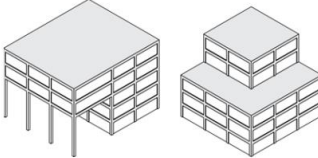
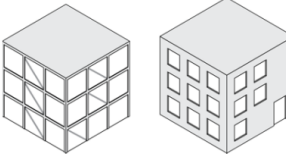
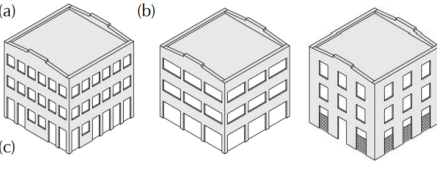
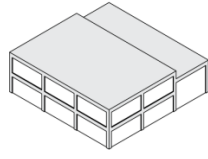
The buildings would be classified using two seismic assessment frameworks. For the height, the study limited its classification into either low-rise or mid-rise according to the classifications proposed by the University of the Philippines Diliman Institute of Civil Engineering (UPD-ICE) as shown in Table 1 (Pacheco et al., 2014). This was done as images of high-rise buildings on Google Street View lack the visible details in the upper stories needed for accurate analysis.

Table 1. UPD-ICE building classifications according to height (Pacheco et al., 2014).

Building type	Height
L	Low; 1-2 stories
M	Mid; 3-7 stories
H	High; 8-15 stories
V	Very high; 16-25 stories
E	Extremely high; 26-35 stories
S	Super high; more than 35 stories

Meanwhile, for the vertical nonregularities, the network would detect the presence of soft stories, out-of-plane setbacks, short columns, and split levels according to FEMA P-154 (2015), as shown in Table 2. The sloping site and in-plane setback nonregularities were not included in this research.

Table 2. Classification of Vertical Nonregularities (FEMA P-154, 2015).

Sloping Site	
Soft Story	
Out-of-Plane Setback	
In-plane Setback	
Short Column/Pier	
Split Levels	

As previously stated, images not only of high-rise buildings but also makeshift buildings were excluded from the database to control the complexity of the images to be analyzed and to limit the amount of noise in each image. It appears that these buildings were excluded due to the lack of visible details in upper stories caused by photo angle limitations and the scattered components and claddings of the captured makeshift buildings adding noise to the photos.

As neural networks are limited to analyzing the exterior visual characteristics of buildings, sight obstructions in Google Street View may have limited the analysis of the network. On the other hand, manual classifications are subject to an appreciable amount of subjectivity especially in classifying vertical nonregularities. As such, the criteria for identifying these nonregularities should be followed consistently while creating the image database, so as not to confuse the network and cause anomalies in the results.

Finally, the implementation of machine learning was conducted using the functions of MATLAB. This includes MATLAB's deep learning toolbox, Bayesian optimization functions, and confusion matrix generation functions, among others.

2 Review of Related Literature

2.1 Building classifications

In the endeavor of creating better risk analysis frameworks for the Philippines, a project known as the Greater Metro Manila Area Risk Assessment Project or GMMA-RAP was developed to appraise the geological and hydrometeorological hazards in the areas of Metro Manila, Bulacan, Cavite, Laguna, and Rizal (PHIVOLCS et al., 2014). Together with existing building types in the GMMA, this data has been used by the UPD-ICE to make seismic vulnerability curves to visualize and subsequently mitigate the effects of such disasters. These curves were obtained with respect to a building's height, structural type, construction material, and quality of construction among other factors (Pacheco et al., 2014). The specificity of these classifications that are unique to Philippine building typology provides a convenient method for assessing seismic vulnerability in arguably the most urbanized area of the Philippines.

However, it is important to note that the GMMA-RAP framework does not have standards to account for vertical nonregularities. For instance, other international building assessment frameworks and standards such as the widely used FEMA P-154 (2015) make use of basic scores which are derived values that summarize the expected seismic performance of a building. Furthermore, as mentioned in the scope of the study, Table 2 shows that FEMA P-154 has standards and criteria for identifying vertical nonregularities, as opposed to the GMMA-RAP framework which does not account for such standards. Hence, it appears that one of the limitations of GMMA-

RAP is its lack of such nonregularity types. For the current research, the term ‘non-regularity’ was used to identify external building geometries that could be susceptible to damage from seismic activity. This has been done to distinguish it from the term ‘irregularity’ which accounts for anomalies in the building, whether external or internal.

2.2 Convolutional Neural Networks

The rapid advancement of technology has paved the way for the automatization of miscellaneous tasks for a variety of purposes. For instance, deep learning neural networks have been a breakthrough in artificial intelligence as it allows the computer to recognize patterns with high accuracy through connected processors called neurons. Similar to how the neurons in our nervous system work, the neurons in neural networks have layers to process information: an input layer, processing (hidden) layers, and an output layer. The learning in “deep learning” is “about finding weights that make the neural network exhibit desired behavior” (Schmidhuber, 2015).

A common subfield in deep learning neural networks is the convolutional neural network or CNN, proven to be exceptional particularly in image processing, voice recognition, and the likes (Albawi, 2017). As shown in Figure 2, the network (1) takes raw pixels of an image as input, (2) connects one neuron to a local part of the image, and strides that window to the next part of the image until the whole image has been “scanned.”

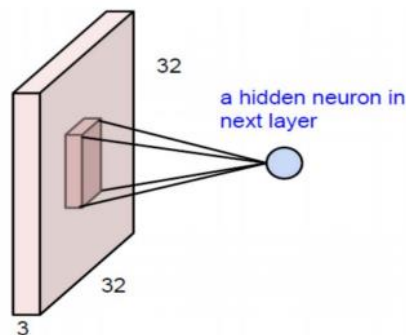


Fig. 2. Convolution of a CNN (Albawi, 2017).

More layers are then added to this input layer to extract various features of the image, such as the convolutional layer, which uses an activation function to obtain a feature map of the image (Zou, Chen, & Chen, 2020). Other layers that follow a convolutional layer are the pooling layer, which compresses data and parameters, thereby effectively reducing overfitting and improving fault tolerance, and the fully connected layer, which acts as the classifier of the CNN (Zou, Chen, & Chen, 2020).

One common network architecture known to have high classification accuracy is ResNet50. This network architecture has an image input of 224 x 224 pixels and is 50 layers deep, as shown in Table 3. Its adoption of shortcut connections between convolutional layer blocks improves the performance of the network without introducing new parameters or complicating the network. From He's findings, ResNet50 had better performance yet lower complexity than existing architectures in VGGnets and GoogleNet (He et al., 2016).

Table 3. ResNet50 layer architecture (He et al., 2016).

Layer name	Output size	Composition
Conv1	112x112	7x7, 64, stride 2
Conv2_x	56x56	3x3 max pool, stride 2
		$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
Conv3_x	28x28	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$
Conv4_x	14x14	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$
Conv5_x	7x7	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1x1	Average pool, 1000-d fc, softmax

2.3 Use of convolutional neural networks in building categorization

The application of convolutional neural networks in building categorization has become a fairly common topic, with multiple studies having been conducted in various countries and with varying results (Gonzalez et al., 2020, Harirchian et al., 2020, Yu et al., 2020, Escobedo, 2017). These studies made use of transfer learning of pre-trained networks with ImageNet weights, which helps the networks learn features from a general dataset to classify a more specific dataset (Yosinski et al., 2014).

Gonzalez et al. (2020) and Yu et al. (2020) compared the performances of different CNN architectures and found that ResNet50 showed exceptional performance in classification yielding the highest values for precision, recall, and accuracy compared to

other architectures. These studies also provided results with varying numbers of building images for network training and their corresponding number of classes, as shown in Table 4.

Table 4. Number of training images and classes used by different studies for building recognition.

Author	No. of images	No. of classes	Accuracy
Gonzalez et al. (2020)	10,000	9	95%
Yu et al. (2020)	1,000 to 17,000	2	82-87%
Escobedo (2017)	3,000	15	83%
Harirchian et al. (2020)	1,000	4	67%
Alegre (2020)	1,000	7	79%

2.4 Bayesian Optimization

A common yet effective algorithm used in optimizing neural network hyperparameters is Bayesian optimization or BayesOpt. Considered as a form of global optimization, BayesOpt is preferred over other procedures such as grid and random search for its ability to “learn” from past iterations using probabilistic models (Wu et al., 2019). The general outline of a Gaussian Process Bayesian Optimization is:

1. Create an objective function of performance measure versus the choice of parameters based on existing training data
2. Obtain an initial evaluation of the function from a random set of hyperparameters within an input search interval
3. Fit a Gaussian process to the objective function and train the network
4. Find a new set of hyperparameters in the search space using an acquisition function.
5. Iterate the process of finding new sets of hyperparameters until termination conditions are met.

BayesOpt has been found to increase the accuracies of deep learning mechanisms including convolutional neural networks by modifying network and training hyperparameters, reaching up to 99% accuracy in classification (Miranda, 2021, Satoto et al., 2020).

3 Methodology

3.1 Data Collection

The input data required for the training and validation of the networks are pictures of building facades which were directly taken from Google Street View screenshots. Figure 3 illustrates the locations of interest–Metro Manila, Bulacan, Cavite, Laguna, and Rizal– which are the basis of the GMMA-RAP classifications (Pacheco, et al. 2014).

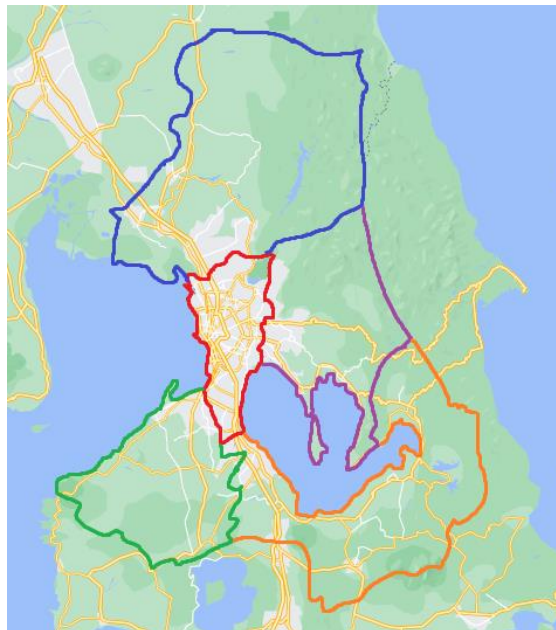
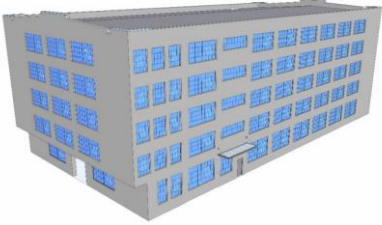
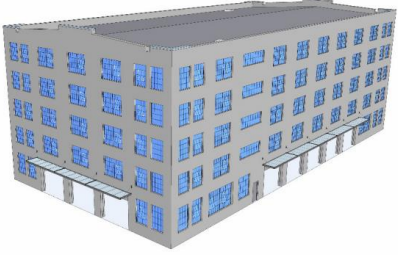
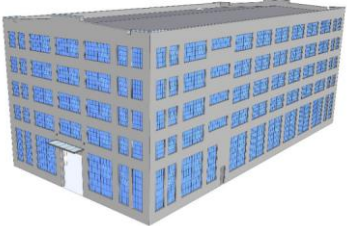
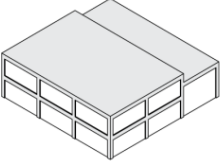
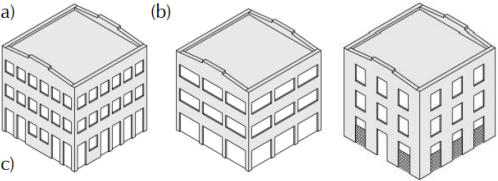


Fig. 3. Map of the Greater Metro Manila Area consisting of: Metro Manila (Red), Bulacan (Blue), Cavite (Green), Laguna (Orange), and Rizal (Violet).

Each image was manually classified as either low-rise or mid-rise (Pacheco et al., 2014). For the nonregularities, buildings were labelled according to the presence of nonregularities (FEMA P-154, 2015) with the following criteria:

Table 5. Vertical nonregularities and criteria according to FEMA P-154 (2015).

	Vertical Nonregularity	Criteria
Out-of-Plane Setback		<p>Has an upper story protruding from the building's lower stories.</p> <p>Buildings with balconies are also classified with this nonregularity.</p>
Soft Story		<p>Has large openings on the first story.</p> <p>This is usually depicted through garage doors or non-structural elements such as glass walls that cover a big part of the first story</p>
		<p>Has a first story that is evidently taller than the stories above it</p>
Split Levels		<p>Has floor or roof levels that aren't aligned with other parts of the building</p>
Short Columns		<p>Usually depicted through having (a) columns with reduced height, (b) deep spandrels, or (c) infill walls with small openings.</p>

3.2 Creating a base ResNet50 network

The topology for a ResNet50 network was imported from MatLab and fine-tuned. This was done by (1) replacing the last fully-connected layer with a new one with labels for only two classes for binary classification, and (2) replacing the last classification layer with one without labels. The network was then trained and validated using Alegre’s (2020) training hyperparameters, as shown in Table 6. This process was repeated twice for a total of three trials.

Table 6. Training hyperparameters for base ResNet50 network (Alegre, 2020)

Dataset split	60% Training 20% Validation 20% Testing
Solver	SGDM
Initial Learning Rate	0.001
Learn Rate Drop Factor	0.01
Epochs	50
Mini Batch Size	5
L2 Regularization	0.00001
Momentum	0.89

3.3 Optimizing training hyperparameters and comparison with base networks

Similar to 3.2, the topology for the ResNet50 networks was imported from MatLab, with the last fully-connected layer and the last classification layer replaced. This code for importing the network was embedded into the code for Bayesian optimization since the optimization is simultaneous with the training and validation of the network. The optimizable variables of the network and the ranges chosen are shown in Table 7:

Table 7. Initial search intervals for implementing Bayesian optimization.

Hyperparameter	Initial Search Interval
Initial Learning Rate	[0.0001, 0.01]
Epochs	[25, 60]
Mini Batch Size	[5, 50]
L2 Regularization	[1e-10, 0.01]
Momentum	[0.7, 1]

These values were estimated based on suggested intervals by past studies (Miranda, 2021, Satoto, 2020, Bihl, 2020) as well as from preliminary testing on the capabilities of the computer used for the research. The parameters of the Bayesian optimization were set to iterate at a maximum of 30 times, while the activation function was set to the default: expected improvement plus per second. The final hyperparameter values

obtained from the 30 iterations were then used to train optimized ResNet50 networks for each of the five classifications with three trials each.

The efficiencies of all networks for all five classifications and their respective trials were then evaluated by calculating the following metrics:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

$$precision = \frac{TP}{TP + FP} \quad (2)$$

$$recall = \frac{TP}{TP + FN} \quad (3)$$

$$F1\ score = \frac{2(precision)(recall)}{precision + recall} \quad (4)$$

where TP is for the total number of True Positives, TN for True Negatives, FP for False Positives, and FN for False Negatives. For nonregularity classifications, the “positive” would be detecting that specific nonregularity.

4 Results and Analysis

4.1 Image Dataset

A total of 2100 images were obtained with the distributions shown in Table 8. The distributions for all the classifications were imbalanced, with the Split Levels and Short Columns classifications being the most severely imbalanced as shown in the last column of Table 8. Image samples under each classification are shown in Figure 4.

Table 8. Image Dataset Distribution.

Classification	L	M	Ratio
Height	848	1,252	1 : 1.5
Classification	With	Without	Ratio
Out-of-Plane Setback	1,147	953	1 : 1.2
Soft Story	1,572	528	1 : 3.0
Split Levels	384	1,716	1 : 4.5
Short Columns	95	2,005	1 : 21.1

a) Low-Rise



b) Mid-Rise



c) Out-of-Plane Setback



d) Soft Story



e) Split Levels



f) Short Columns



Fig. 4. Examples of a) low-rise buildings, b) mid-rise buildings, c) buildings with out-of-plane setbacks, d) short stories, e) split levels, and f) short columns.

4.2 Hyperparameters obtained from optimization

The hyperparameters obtained from the implementation of Bayesian optimization on the dataset are shown in the first five rows of Table 9. Other parameters of the optimization process such as the total evaluation time and number of iterations are also shown in the bottom rows of the table.

Table 9. Obtained hyperparameters from Bayesian Optimization process.

	Height	Out-of-Plane Set-back	Soft Story	Split Levels	Short Columns
Initial Learning Rate	0.003161	0.003094	0.008956	0.009404	2.353e-05
Epochs	31	57	58	27	60
MiniBatchSize	23	48	20	28	33
L2 Regularization	1.129e-10	4.230e-10	0.0006755	0.006814	1.088e-07
Momentum	0.83181	0.99839	0.92656	0.73813	0.73101
No. of iterations	30	30	30	30	30
Elapsed Time	6 hr 32 min	7 hr 21 min	6 hr 7 min	8 hr 11 min	9 hr 24 min

4.3 Evaluation and Comparison

Table 10 shows the average evaluation metrics obtained for the height classification and each nonregularity classification after three trials.

Table 10. Average evaluation metrics for all classifications.

		Precision	Recall	Accuracy	F1-score
Height	Base	96.3%	91.9%	95.4%	94.3%
	Optimized	95.1%	94.7%	95.8%	94.9%
Out-of-Plane Setback	Base	66.1%	74.5%	64.5%	69.5%
	Optimized	64.6%	81.9%	65.5%	72.1%
Soft Story	Base	84.5%	94.8%	82.3%	89.3%
	Optimized	80.0%	98.7%	79.4%	88.3%
Split Levels	Base	31.3%	41.2%	72.8%	35.1%
	Optimized	36.1%	43.4%	75.2%	37.5%
Short Columns	Base	4.3%	19.1%	79.5%	6.8%
	Optimized	NaN%	0%	95.5%	NaN%

Height

The optimized network outperformed the base network in accuracy and F1 score by a margin of less than one percent. However, it can also be observed that the values of the optimized network are more precise, deviating less than one percent from the average, whereas the deviation of the base network is larger by nearly two percent from the average.

Table 11. Comparison of evaluation metrics for the height classification

	Precision				Recall			
	Trial 1	Trial 2	Trial 3	Average	Trial 1	Trial 2	Trial 3	Average
Base Network	94.7%	96.8%	97.4%	96.3%	94.1%	93.5%	88.2%	91.9%
Optimized Network	96.4%	94.2%	94.7%	95.1%	95.3%	94.7%	94.1%	94.7%
	Accuracy				F1-score			
	Trial 1	Trial 2	Trial 3	Average	Trial 1	Trial 2	Trial 3	Average
Base Network	95.4%	96.8%	94.1%	95.4%	94.4%	96.1%	92.6%	94.4%
Optimized Network	96.6%	95.4%	95.4%	95.8%	95.9%	94.4%	94.4%	94.9%

Out-of-Plane Setback

For the out-of-plane setback, it can be seen that both networks experienced significantly lower performances than the height, with 60-70% in accuracy and F1 score. However, the optimized network performed moderately better than the base network in all trials.

Table 12. Comparison of evaluation metrics for the out-of-plane setback classification

	Precision				Recall			
	Trial 1	Trial 2	Trial 3	Average	Trial 1	Trial 2	Trial 3	Average
Base Network	61.3%	64.9%	72.2%	66.1%	79.1%	82.2%	62.2%	74.5%
Optimized Network	65.8%	64.2%	63.9%	64.6%	73.5%	83.5%	88.7%	81.9%
	Accuracy				F1-score			
	Trial 1	Trial 2	Trial 3	Average	Trial 1	Trial 2	Trial 3	Average
Base Network	61.2%	66.0%	66.2%	64.5%	69.1%	72.5%	66.8%	69.5%
Optimized Network	64.5%	65.5%	66.4%	65.5%	69.4%	72.6%	74.3%	72.1%

Soft Story

For this classification, although the optimized network has better performance in recall, the base network performed better in all other metrics in terms of the mean of

three trials. It was also found that the optimized network has trouble classifying buildings without soft story, and instead classifies most of the images as with soft story. Although it can be said that it is acceptable to diagnose a building with a nonregularity if there is uncertainty, this goes to show how the optimized network is biased towards the larger class of data.

Table 13. Comparison of evaluation metrics for the soft story classification

	Precision				Recall			
	Trial 1	Trial 2	Trial 3	Average	Trial 1	Trial 2	Trial 3	Average
Base Network	82.9%	85.3%	85.3%	84.5%	96.7%	92.7%	94.9%	94.8%
Optimized Network	82.2%	79.2%	78.5%	80.0%	97.6%	99.1%	99.4%	98.7%
	Accuracy				F1-score			
	Trial 1	Trial 2	Trial 3	Average	Trial 1	Trial 2	Trial 3	Average
Base Network	81.8%	81.8%	83.2%	82.3%	89.3%	88.9%	89.8%	89.3%
Optimized Network	81.0%	79.0%	78.3%	79.4%	89.2%	88.0%	87.7%	88.3%

Split Levels

For the Split Levels, the performance of the optimized network was overall better. However, the precision, recall, and F1 scores for both networks are noticeably low. This is because of the imbalance in classes, better visualized by the confusion matrices of the first trial of training shown in Figure 5 where both networks evidently experienced difficulty in detecting split levels. As mentioned in Table 8, the ratio of buildings with split levels to those without is 1:4.5, and it is likely that the network requires more training images – especially of buildings with the nonregularity – for it to learn and consistently classify the images correctly.

Table 14. Comparison of evaluation metrics for the split levels classification

	Precision				Recall			
	Trial 1	Trial 2	Trial 3	Average	Trial 1	Trial 2	Trial 3	Average
Base Network	32.4%	29.0%	32.6%	31.33%	30.3%	52.6%	40.8%	41.23%
Optimized Network	45.3%	32.5%	30.4%	36.07%	31.6%	52.6%	46.1%	43.43%

	Accuracy				F1-score			
	Trial 1	Trial 2	Trial 3	Average	Trial 1	Trial 2	Trial 3	Average
Base Network	76.0%	68.2%	74.1%	72.77%	31.3%	37.7%	36.2%	35.08%
Optimized Network	80.8%	71.7%	71.3%	75.23%	37.2%	40.2%	36.6%	37.5%

		Confusion Matrix		
		With Split Levels	Without Split Levels	Average
Output Class	With Split Levels	23 5.5%	48 11.4%	32.4% 67.6%
	Without Split Levels	53 12.6%	297 70.5%	84.9% 15.1%
	Average	30.3% 69.7%	86.1% 13.9%	76.0% 24.0%
		With Split Levels	Without Split Levels	Target Class

		Confusion Matrix		
		With Split Levels	Without Split Levels	Average
Output Class	With Split Levels	24 5.7%	29 6.9%	45.3% 54.7%
	Without Split Levels	52 12.4%	316 75.1%	85.9% 14.1%
	Average	31.6% 68.4%	91.6% 8.4%	80.8% 19.2%
		With Split Levels	Without Split Levels	Target Class

Fig. 5. Confusion matrix for the first trial of the base network (left) and the optimized network (right) for split levels classifications**Short Columns**

Lastly, for the short columns, the results are quite skewed due to the networks having trouble detecting short columns at all; similar to what was observed in the split levels. To further investigate this, Figure 6 shows the confusion matrices for both the base and optimized network for the first trial of the training for this nonregularity:

Output Class	Confusion Matrix		
	With Short Columns	Without Short Columns	Average
With Short Columns	10 (2.4%)	183 (43.6%)	5.2% (94.8%)
Without Short Columns	9 (2.1%)	218 (51.9%)	96.0% (4.0%)
Average	52.6% (47.4%)	54.4% (45.6%)	54.3% (45.7%)

Output Class	Confusion Matrix		
	With Short Columns	Without Short Columns	Average
With Short Columns	0 (0.0%)	0 (0.0%)	NaN% (NaN%)
Without Short Columns	19 (4.5%)	401 (95.5%)	95.5% (4.5%)
Average	0.0% (100%)	100% (0.0%)	95.5% (4.5%)

Fig. 6. Confusion matrix for the first trial of the base network (left) and the optimized network (right) for short columns classifications

The base network was able to classify 10 of the 19 images with short columns correctly, but tended to classify too many images with the nonregularity when they should not have been. The optimized network, though, classified all of the test images as without short columns, causing higher accuracy, precision, and recall, but evidently, the optimized network was not able to detect short columns at all. The optimized network was consistent with this throughout all trials, whereas this same result appeared in only the third trial for the base network. The most plausible source of error for these skewed results is the severe imbalance of class data in the short column classifications. In fact, this may be a limitation of Bayesian optimization where there is a class imbalance, in that it might have chosen to classify all images as without short columns because it was more likely to yield a high accuracy, but on the contrary, this is not desired for a network that is meant to identify vulnerability points in buildings.

Table 15. Comparison of evaluation metrics for the short columns classification

	Precision				Recall			
	Trial 1	Trial 2	Trial 3	Average	Trial 1	Trial 2	Trial 3	Average
Base Network	5.2%	3.3%	NaN%	4.25%	52.6%	5.3%	0%	19.3%
Optimized Network	NaN%	NaN%	NaN%	NaN%	0%	0%	0%	0%
	Accuracy				F1-score			
	Trial 1	Trial 2	Trial 3	Average	Trial 1	Trial 2	Trial 3	Average
Base Network	54.3%	88.8%	95.5%	79.53%	9.46%	4.07%	NaN%	6.77%
Optimized Network	95.5%	95.5%	95.5%	95.5%	NaN%	NaN%	NaN%	NaN%

4.4 Possible Sources of Error

It was expected that the optimized network should have performed better than the base network for all five of the classifications, but the results obtained showed otherwise. This contradiction was especially evident in the soft story classification, the opposite was observed. The problem may lie in the insufficient analysis of the evaluation metrics, since only the means of the accuracies, precision, recall, and F1 scores were taken. Since the outputs of convolutional neural networks follow a multivariate normal distribution (Borovykh, 2019), it is possible that the trials obtained in this research were located near the tail ends of the 3D bell curves, and are not representative of the true characteristics of the data. This then suggests that the results obtained in this research are inconclusive, and requires further statistical analysis. However, due to time restrictions in the implementation, further analysis is beyond the scope of this research.

Another major source of error is the class imbalance. As evident in the soft story, short columns, and split levels results, the optimized network tends towards the larger class, which, in turn, affected the metrics of the results. This paired with the relatively low amount of training images – since the testing dataset was allotted 20% of the 2100 images – giving both the optimized and base networks insufficient ground to learn the features of the nonregularities; hence, the low metrics. Undoubtedly, adding more images to the dataset could have helped in avoiding this error (Lei et al., 2022). Furthermore, this study could have been more selective in gathering images for the dataset to make the classes more balanced. Widening the scope of the locations and possibly avoiding taking multiple buildings in close proximity to each other may have also helped in achieving this.

Other possible sources of error may have arisen from the training data themselves. Some images may be subject to too much noise like obstructions in the form of a car, trees, or tarpaulins that make the image an unfavorable sample, with some images still being subject to obstructions despite changing the angle of the street view. Noise causes an image to be more detrimental to classification accuracy rather than helping the network learn (Wang, et al., 2018). This is especially applicable to the nonregularity classification dataset, since there is a number of criteria one must understand to detect these nonregularities, and with problematic datasets, naturally, it would be more difficult for the network to classify images. Nonetheless, the exact behavior of the neural network remains difficult to be analyzed due to the black box problem being too complex to trace (Theobald, 2017).

5 Conclusion and Recommendations

The research was able to collate a database of 2100 images from the Greater Metro Manila Area in the Philippines to train ResNet50 networks. The networks were taught to classify the images according to height and to detect the presence of nonregularities, limited to: soft stories, out-of-plane setbacks, short columns, and split levels. After Bayesian optimization was successfully implemented, it was found that the optimized networks produced a higher mean in F-1 scores than the base network, therefore performing better in three out of the five classifications. Meanwhile, it appeared that the base network outperformed the optimized network in the two other classifications, namely the soft story and short columns. Anomalies were observed in the precision and recall metrics, thereby affecting the F-1 scores of the classifications. The anomalies or errors documented were attributed to the severe imbalance of images per class and an appreciable amount of noise in the images collected.

The data presented provides grounds for the potential of Bayesian optimization to enhance the performance of building-categorizing neural networks as it is a computationally efficient and systematic process for determining suitable hyperparameters while still maintaining or even improving the performance in classification. These results set the precedent for faster and more efficient building categorization, particularly in the Philippines. However, due to shortcomings in analysis and significant limitations in the dataset, there is further potential for future studies to implement neural networks in building categorization, particularly Bayesian optimization, with more substantial datasets and more thorough analysis.

To unlock that potential, future studies may focus on obtaining larger datasets with more balanced classes. Conversely, techniques may be performed to manage these imbalances, possibly through (1) data-level methods like random oversampling and undersampling, wherein the dataset is modified either by duplicating or removing images respectively; or (2) through algorithm-level methods like cost-sensitive learning wherein adjustments to the network are applied mid-training (Johnson & Khoshgoftaar, 2019). The noise in image datasets can also be managed appropriately to possibly improve the performance of the network. Ilesanmi and Ilesanmi (2021) enumerate different CNN methods for denoising images, such as MP-DCNN and CDNet, but these, too, could be developed further for specific classification tasks like building recognition. In performing analyses, statistical methods such as the student's t-test and the analysis of variance may help yield a more accurate picture of the results and provide a more grounded comparison between the base and optimized network. Similarly, other evaluation metrics such as the AUC-ROC curve may be explored as well to present a different perspective on the behavior of the evaluation metrics amidst imbalanced classifications. Other options of Bayesian optimization may also be experimented with, whether in reducing the number of hyperparameters optimized or in adding them, inclusive of the search range of these as well. Other CNN architectures

like InceptionResNetV2 and NasNetLarge may also be explored instead of ResNet50, as the choice of architecture may also affect the accuracy of the network.

References

1. Albawi, S., Al-Zawi, S., & Mohammed, T.A. (2017). *Understanding of a Convolutional Neural Network*. International Conference on Engineering and Technology 2017, Antalya, Turkey.
2. Alegre, E. DM. (2020). Recognition of Typical Building Typology and Characteristics Using Deep Neural Network on Street Level Images on Philippine Structures [Unpublished dissertation]. UP Institute of Civil Engineering.
3. Bezak, P. (2016). Building Recognition System Based on Deep Learning. *2016 Third International Conference on Artificial Intelligence and Pattern Recognition (AIPR)*. <https://doi.org/10.1109/icaipr.2016.7485230>
4. Bihl, T., Schoenbeck, J., Steeneck, D., & Jordan, J. (2020). Easy and Efficient Hyperparameter Optimization to Address Some Artificial Intelligence “ilities”. *Proceedings of the 53rd Hawaii International Conference on System Sciences*.
5. Borovykh, A. (2019). A Gaussian Process perspective on Convolutional Neural Networks. *University of Bologna Department of Mathematics*.
6. Branco, P., Torgo, L., & Ribeiro, R. P. (2015). A Survey of Predictive Modelling Under Imbalanced Distributions. <https://doi.org/10.48550/arXiv.1505.01658>
7. Escobedo, J. D., Enciso-Rodas, L., & Vargas-Muaoz, J. E. (2017). Towards accurate building recognition using convolutional neural networks. *Proceedings of the 2017 IEEE 24th International Congress on Electronics, Electrical Engineering and Computing, INTERCON 2017*. <https://doi.org/10.1109/INTERCON.2017.8079686>
8. Federal Emergency Management Agency (2015). *Rapid Visual Screening of Buildings for Potential Seismic Hazards: A Handbook*, Third Edition.
9. Fukushima, Y., Hashimoto, M., Miyazawa, M., Uchida, N., & Taira, T. (2019). Surface creep rate distribution along the Philippine fault, Leyte Island, and possible repeating of Mw ~6.5 earthquakes on an isolated locked patch. *Earth, Planets and Space*, 71(118), <https://doi.org/10.1186/s40623-019-1096-5>
10. Gonzalez, D., Rueda-Plata, D., Acevedo, A., Duque, J., Ramos-Pollán, R., Betancourt, A., & García, S. (2020). Automatic detection of building typology using deep learning methods on street level images. *Building and Environment*, 177. <https://doi.org/10.1016/j.buildenv.2020.106805>
11. Harirchian, E., Kumari, V., Jadhav, K., Das, R.R., Rasulzade, S., & Lahmer, T. (2020). A Machine Learning Framework for Assessing Seismic Hazard Safety of Reinforced Concrete Buildings. *Applied Sciences* 10(7153), 1-18. doi:10.3390/app10207153
12. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition.
13. Ilesanmi, A.E. & Ilesanmi, T.O. (2021). Methods for image denoising using convolutional neural network: a review. *Complex and Intelligent Systems* 7, 2179-2198. <https://doi.org/10.1007/s40747-021-00428-4>
14. Johnson, J.M., Khoshgoftaar, T.M. (2019). Survey on deep learning with class imbalance. *Journal of Big Data* 6(27). <https://doi.org/10.1186/s40537-019-0192-5>

15. Lei, S., Zhang, H., Wang, K., & Zhendong, S. (2019). How Training Data Affect the Accuracy and Robustness of Neural Networks for Image Classification. *ICLR 2019 Conference Blind Submission*.
16. Miranda, M., Kid, V. V., & Sulla-Torres, J. (2021). A Detailed Study on the Choice of Hyperparameters for Transfer Learning in Covid-19 Image Datasets using Bayesian Optimization. *International Journal of Advanced Computer Science and Applications*. <https://doi.org/10.14569/IJACSA.2021.0120441>.
17. Pacheco, B.M., Hernandez Jr., J.Y., Castro, P.P.M., Tingatinga, E.A.J., Suiza, R.M., Tan, L.R.E., Longalong, R.E.U., Veron, M.C.L., Aquino, D.H.M., Macuha, R.N., Mata, W.L., Villalba, I.B.O., Pascua, M.C.L., Ignacio Jr., U.P., Germar, F.J., Dino, J.R.M., Reyes, G.P.D., Tirao, J.L.M., & Zarco, M.A.H. (2014). *Development of Vulnerability Curves of Key Building Types in the Greater Metro Manila Area, Philippines: Final Report*. Institute of Civil Engineering, University of the Philippines Diliman.
18. Philippine Institute of Volcanology and Seismology, Philippine Atmospheric, Geophysical, and Astronomical Services Administration, National Mapping and Resource Information Authority, Mines and Geosciences Bureau, Metro Manila Development Authority, Department of Public Works and Housing, Laguna Lake Development Authority, & Geoscience Australia (2014). *Enhancing Risk Analysis Capacities for Flood, Tropical Cyclone Severe Wind and Earthquake for Greater Metro Manila Area: Summary Report*.
19. Satoto, B.W., Utoyo, M.I., Koendhori, E.B., & Rulaningtyas, R. (2020). Custom Convolutional Neural Network with Data Augmentation and Bayesian Optimization for Gram-Negative Bacteria Classification. *International Journal of Intelligent Engineering and Systems*, 13(5), 524-538. doi: 10.22266/ijies2020.1031.46
20. Schmidhuber, J. (2015). "Deep learning in neural networks: An overview." *Neural Networks* 61, 85-117.
21. Snoek, J., Rippel, O., Swersky, K., Kiros, R., Satish, N., Sundaram, N., Patwary, M.M.A., Prabhat, Adams, R.P. (2015). Scalable Bayesian Optimization Using Deep Neural Networks. *Proceedings of the 32nd International Conference on Machine Learning*. <http://proceedings.mlr.press/v37/snoek15.pdf>
22. Tan, E., Raissa, L., Hernandez, J. Y., Diola, N. B., Albert, M., Zarco, H., Victor, O., Antonio, M., Germar, F. J., Orozco, C. R., Torio, L. v, Cariño, R., Tan Tian, J., Angelo, J., & Longalong, R. E. (2013). *Development of a Rapid Condition Assessment Tool for Reinforced Concrete Moment-Resisting Frame Buildings in the Philippines: Structural Component*. Institute of Civil Engineering, University of the Philippines Diliman.
23. Theobald, Oliver (2017). *Machine Learning for Absolute Beginners: A Plain English Introduction* 2nd ed.
24. Wang, T., Huan, J., & Li, B. (2018). Data Dropout: Optimizing Training Data for Convolutional Neural Networks. <https://doi.org/10.48550/arXiv.1809.00193>
25. Wu, J., Chen, X., Zhang, H., Xiong, L., Lei, H., & Deng, S. (2019). Hyperparameter Optimization for Machine Learning Models Based on Bayesian Optimization. *Journal of Electronic Science and Technology*, 17(1), 26-40. <https://doi.org/10.11989/JEST.1674-862X.80904120>
26. Yosinski, J., Clune, J., Bengio, Y., & Lipson, H. (2014). How transferable are features in deep neural networks? *In Advances in Neural Information Processing Systems*, 27.
27. Yu, Q., Wang, C., McKenna, F., Yu, S., Taciroglu, E., Cetiner, B., & Law, K. (2020). Rapid visual screening of soft-story buildings from street view images using deep learning

- classification. *Earthquake Engineering and Engineering Vibration*, 19(4), 827-838.
<https://doi.org/10.1007/s11803-020-0598-2>
28. Zou, S., Chen, W., & Chen, H. (2020). Image Classification Model Based on Deep Learning in Internet of Things. *Wireless Communication and Mobile Computing*, 2020, <https://doi.org/10.1155/2020/6677907>