Categorising Buildings in Lausanne Based on Façade Material

Youssef Belghmi, Hamza Morchid
Machine Learning Project, Department of Data Science, EPFL, Switzerland

***Abstract*—This project is part of a collaboration with HERUS (Laboratory for Human-Environment Relations in Urban Systems) aimed at harnessing the capabilities of machine learning to solve a critical urban challenge. Our main goal is to classify buildings in Lausanne based on the composition of their facade materials, using advanced machine learning techniques. Our project aims to provide the HERUS laboratory with crucial data to inform their future initiatives in the expansion and transformation of Lausanne's built environment.**

I. Introduction

This report documents our collaborative research project with HERUS, which aims to solve a crucial challenge related to the architecture and composition of buildings in the city of Lausanne. The HERUS laboratory has clearly established its objectives in anticipating the planned expansion of the built environment in Lausanne, necessary to meet the needs of residents by 2050. To realize this vision, it is essential to obtain in-depth knowledge of the materials currently used in the existing building stock, with a view to their reuse.

It is in this context that our project takes off, by exploiting advanced machine learning techniques to accomplish this complex and essential task. In this report, we will explore our research methodologies, tools and results in detail.

II. Data Collection

Data collection involves collecting essential information to train and test a model. In our project, we collect images of these facades to train our model to recognize them. The quality and diversity of training data is essential because it determines the accuracy and performance of classification.

A. Dataset of Lausanne Buildings

As part of our project, we were responsible for collecting, on our own, images of the facades of buildings in Lausanne. However, the HERUS laboratory also offered two valuable resources for this task :

1) Swiss Real Estate Database [1] : This resource offers detailed information on buildings in Switzerland, taken from the official directory of building addresses.
2) Google Street View via the Google Maps API : Using this API allows us to download images of building facades at specific geographic locations in Lausanne.

By strategically leveraging these two resources, we managed to generate a dataset consisting of images captured from Google Street View, specifically of building facades in Lausanne. We accomplished this task using the geographic coordinates available in the official Swiss buildings database.

B. Data Cleaning and Labeling

After gathering our image dataset, the next step is to perform a preliminary analysis to better understand the 1,437 images we acquired. First, we identified a significant number of empty images, resulting from missing data in Google Street View. To ensure the quality of our dataset, we eliminated these empty images, retaining only 918 relevant images containing building facades.

We also notice a significant data imbalance in our dataset, to address this we have expanded the classification categories beyond the laboratory's initial request of three classes (brick, stone and mixed material facades). Of the 918 usable images, fewer than 10 fall into each of the categories ”stone”, ”brick” and ”mix”. To better adapt to the diversity of facades, we introduced a new class, “concrete”, to represent buildings with concrete facades.

This is not the only limitation of our dataset. Indeed, our dataset exhibits significant limitations, primarily related to image quality and clarity. These limitations include blurriness caused by camera movement, obstructions by objects like trees and fences, inaccuracies in building coordinates, and images captured from unfavorable angles. These imperfections are likely to impact our model's performance. To address the challenge of identifying challenging facades, we've introduced a new class called ”other.” This class will encompass images that don't depict buildings, have significant obstructions in front of facades, or display different facade materials, such as wood or glass, distinct from the other four classes.

The final step involves manually categorizing buildings into classes that correspond to their respective facade construction materials. This manual labeling process is indispensable in establishing an accurate and trustworthy dataset for our model. Each image will be assigned to one of the predefined classes, including ”brick,” ”stone,” ”mixture,” ”concrete,” or ”other.” To provide a visual representation, here are exemplary images from Google Street View for each of the five classes :

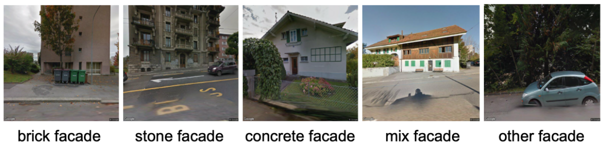


Figure 1. Examples of facades for each class

C. Creating a New Dataset

This in-depth dataset analysis revealed significant diversity among the images, with the majority falling into the ”concrete” and ”other” classes, highlighting the need for a more balanced representation. Thus, the varied image quality and limited data for certain classes pose real challenges for our model’s accuracy and efficiency.

To address these issues, we’ve decided to create a new dataset. This dataset will consist of the highest-quality images from our original Google Street View API collection, supplemented by manually captured screenshots to overcome perspective and obstruction problems. We’re also enriching it with images from external datasets, including those from a Czech study [2], the University of Bonn dataset [3], and from a GitHub repository [4].

This methodical approach aims to achieve a more even distribution of data among classes and ensure superior image quality for robust model training. However, it’s important to acknowledge that building facade image collection is challenging, and despite our efforts, we’ve obtained limited data, necessitating careful utilization of these valuable resources.

III. Data Preprocessing

Data preprocessing is an important step in data preparation, involving the cleaning, transformation, and structuring of raw data to establish its quality, consistency, and relevance. This essential step creates a robust basis for conducting subsequent data analysis and modeling.

A. Data Partitioning

To prepare our dataset for data augmentation, a technique used to expand the dataset by creating new variants of existing data, we must first distribute the images within each class into the training and testing folders following predefined proportions. Specifically, we allocate 70% of the dataset for training, to enable effective learning of class characteristics, and reserve 30% for testing, to evaluate the model’s generalization performance while preventing overfitting. This strategic balanced data distribution step is crucial to accurately gauge the model’s performance in real and unfamiliar scenarios.

B. Data Augmentation

We now enter the data augmentation phase, where we apply image transformations for improved model generalization during training, reducing the risk of overfitting. These transformations include resizing and normalization based on mean and standard deviation values from our data. Given our limited dataset size, these diverse transformations, such as flips, translations, cropping, zooming, grayscale conversion, shearing, blur, and brightness adjustments, are crucial to enhance model robustness and generalization.

Due to the limited amount of images available in our dataset, applying various transformations to training images is essential to improve the robustness of the model and its ability to generalize. For example, from a building facade image in the training set, we can randomly obtain a transformed image as shown below :

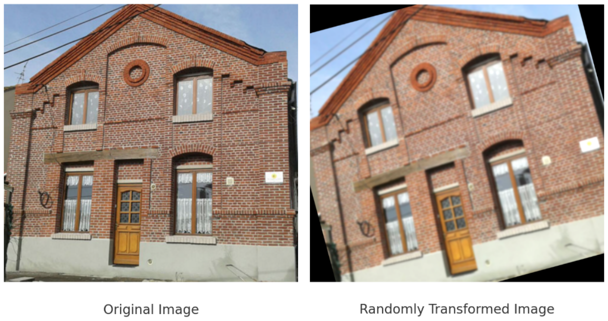


Figure 2. Examples of random transformations made on an image

In this way, data augmentation allows us to simulate different scenarios and enrich the diversity of images provided to the model, despite the limited amount of training images.

IV. Building the Model

This section discusses the development of image classification models for building facade materials. We’ll compare three approaches : pre-trained models on ImageNet like VGG-16 and ResNet-50, as well as our custom model named SimpleCNN, created from scratch.

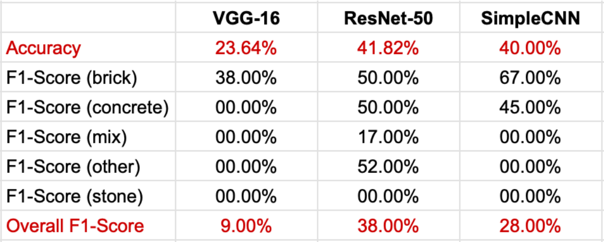
A. Model Development Process

Developing a machine learning model comprises three fundamental phases. The first is model preparation, where the model is customized to suit the specific classification task. Next, the training phase, which iteratively refines the model through repeated exposure to the training data, with gradual adjustments to improve its performance. Finally, model evaluation to assess its ability to generalize to new, unseen data, effectively gauging its proficiency in solving the classification task.

B. Comparison of Models

The three models that we used in our study have specific characteristics that differentiate them. The VGG-16 pre-trained model is characterized by its simplicity and linear architecture. It includes a total of 13 convolutional layers, for extracting features from the input data, and 3 fully connected layers, used for classification decisions. On the other hand, the ResNet-50 pre-trained model, more advanced than VGG-16, stands out for its complexity with its 48 convolutional layers, as well as a Max Pooling layer and an Average Pooling layer, reducing the dimension of the features . Finally, our custom model, SimpleCNN, which consists of five convolutional layers followed by ReLU activation functions, allowing the model to learn complex patterns, and Max Pooling layers, with a dropout regularization layer to avoid overfitting by randomly deactivating a fraction of neurons during training. Ultimately, this results in a fully connected layer that performs the classification task.

In our performance benchmarking study, we will use evaluation metrics to determine the ideal model for our classification task. Accuracy, to measure the percentage of correct predictions across all data samples, assessing the overall classification ability of the model, and F1 score which evaluates performance by class, providing a more comprehensive assessment than accuracy alone. The table below summarizes the results obtained.



	VGG-16	ResNet-50	SimpleCNN
Accuracy	23.64%	41.82%	40.00%
F1-Score (brick)	38.00%	50.00%	67.00%
F1-Score (concrete)	00.00%	50.00%	45.00%
F1-Score (mix)	00.00%	17.00%	00.00%
F1-Score (other)	00.00%	52.00%	00.00%
F1-Score (stone)	00.00%	00.00%	00.00%
Overall F1-Score	9.00%	38.00%	28.00%

Figure 3. Model performance comparison table

ResNet-50 and SimpleCNN demonstrate significantly higher accuracy compared to VGG-16, both achieving an accuracy of approximately 40%, nearly doubling VGG-16's performance, which is close to that of a random classifier (20%). By considering F1 scores, which reflect ResNet-50's remarkable ability to balance precision and recall across diverse classes, with an overall F1 score of 38%, it becomes evident that ResNet-50 is the preferred model for our task.

Although ResNet-50's initial results are based on a limited but reasonable number of iterations (only 10 training epochs), we can plot train and test loss curves to understand model behavior, evaluate learning progress, and detect possible overfitting or underfitting.

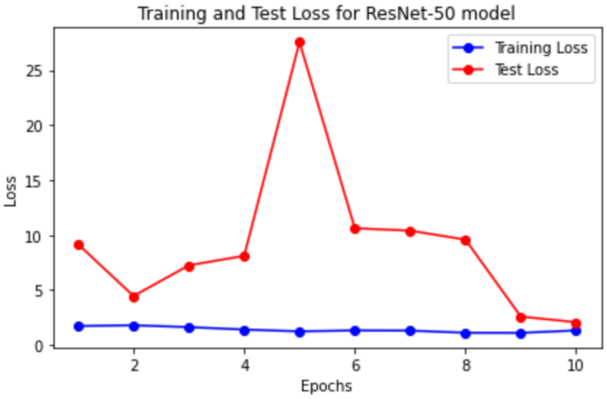


Figure 4. Evolution of training and test losses

The fluctuation in test loss implies potential overfitting of the model to the training data, as the ideal scenario involves a stable or constant decrease in test loss for proper generalization. Since, at the end, we notice a convergence of losses, implementing regularization techniques or fine-tuning hyperparameters could enhance the model's capacity to generalize effectively to new data.

C. Regularization and Hyperparameter Tuning

Regularization and hyperparameter tuning are essential for optimizing machine learning models. Regularization prevents overfitting by penalizing model complexity and promoting simplicity and generalization. We'll employ ridge regularization, which adds a penalty term to the loss function based on parameter squares. Hyperparameter tuning fine-tunes the learning parameters to improve the efficiency and performance of the model on unseen data. Together, these methods enhance model robustness and overall performance.

To determine the optimal hyperparameters, we will focus on adjusting the number of training epochs and weight decay. For the number of epochs, we will assess accuracy at each epoch across a range of 200 epochs, plotting the average cumulative accuracy against the epoch count. The epoch where the average cumulative accuracy stabilizes indicates the optimal training epoch. Regarding weight decay, we will calculate accuracy for various logarithmically scaled values between 0 and 0.1. The weight decay value producing the highest accuracy will be considered optimal for our model. The experiment results are detailed below.

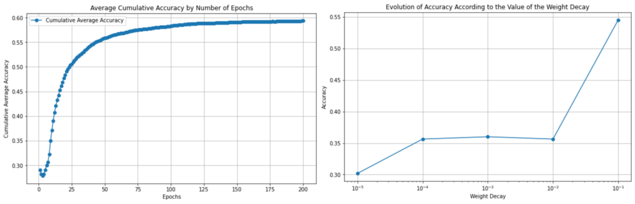


Figure 5. Evolution of training and test losses

Thus, since the accuracy stabilizes around 125 epochs, indicating that the model is no longer improving significantly, 125 seems to be the optimal number of training iterations. As for the weight decay, the hyperparameter determining the strength of the regularization applied to the model weights, the best accuracy is achieved for the value of 0.1. This is therefore a strong regularization leading to smaller weights.

To optimize the model, we also adjust the learning rate, which determines the step size at each iteration. Rather than setting a single value, we used a scheduler to improve performance and accelerate convergence, it gradually reduces the learning rate allowing the model to first take large steps to quickly explore solutions, then scale back step size gradually as it approaches an optimal solution. As for batch size, which determines the number of training examples used in each iteration, we know that using a large batch size with a small dataset poses a risk of overfitting. Thus, we chose a value of 32, rather than 64 or 128, due to our limited data set.

D. Potential Improvements

To improve the performance of our model, we undertook a different approach that aims to strengthen its ability to distinguish between different types of building materials. Our approach consists of creating a new dataset containing around ten specific images for each of five distinct material classes. These images have been meticulously selected to highlight the unique characteristics of each material type, including different colors, textures and shapes. Here is an example of an image used for each material class :

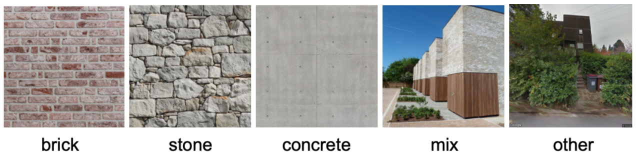


Figure 6. Examples of materials for each class

This approach aims to break down the learning of the model into two distinct phases, the “first training” on images of materials, followed by a “second training” on images of building facades. The first phase aims to familiarize the model with the specific characteristics of construction materials, while the second phase exploits the knowledge acquired in the first to improve the performance of the model on images of entire building facades.

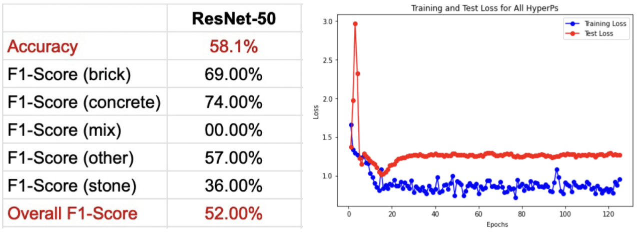
This two-step approach proved extremely effective, resulting in a significant increase in model accuracy in just 10 epochs, going from 41.82% for the ResNet-50 model without hyperparameterization to 52.73% for the same model but with the two training phases. This demonstrates the positive impact of this strategy on the performance of the model.

V. Summary

A. Final Results

We would like to highlight the results obtained by optimizing the best of the three models tested, ResNet-50. By comparing these results to those initial from the ResNet-50 model, as detailed in the comparison table, we can better appreciate the impact of our optimization. The optimized ResNet-50 model was trained for 125 epochs, with the use of a scheduler, an optimal weight decay of 0.1, and a batch size of 32, due to the small amount of data.

The regularization process and hyperparameter optimization therefore resulted in a significant improvement in model accuracy, reaching an accuracy above 58.18% (initially 41.82%). This notable increase in accuracy suggests that the model is now better suited to the specific data it was trained on, allowing it to generalize its predictions more successfully. Additionally, the improvement in F1 score for each class suggests a better balance between precision and recall, crucial aspects in evaluating models for multi-class classification tasks. Below is a table showing the updated results, along with visualizations to monitor the evolution of training and test losses across different epochs.



	ResNet-50
Accuracy	58.1%
F1-Score (brick)	69.00%
F1-Score (concrete)	74.00%
F1-Score (mix)	00.00%
F1-Score (other)	57.00%
F1-Score (stone)	36.00%
Overall F1-Score	52.00%

Figure 7. Results of the regularized and hyperparameterized model

Despite the difficulties encountered for the mix class, due to the complexity of distinguishing different materials on a facade, we are pleased with the progress of the model, which has seen its accuracy and F1 scores improve considerably. The absence of overfitting is confirmed by the loss curves which remain aligned in time over almost all of the 125 epochs, attesting to the capacity of the model to generalize.

B. Project Boundaries

One limitation of our project stems from time constraints, which hindered our ability to further enhance the two-step training model with effective regularization and hyperparameter tuning. However, the primary limitation revolves around our dataset. The lack of usable images of Lausanne buildings in Google Street View limited our use of city-specific building images, forcing us to search online sources for suitable building facade images. This collection process proved to be long and laborious, but despite these challenges, we explored creative solutions to obtain a relevant data set and achieved notable results despite the data limitations.

C. Ethical Risk

As part of our project, we can identify a major ethical risk linked to the potentially inappropriate use of data. This risk mainly concerns the potential violation of the privacy of building owners and residents, as the acquisition and analysis of data on building facades could involve access to sensitive information without the explicit consent of the parties involved. The severity of this risk is high, because privacy is a fundamental right, and the likelihood of this risk occurring is not negligible in the current context of increased awareness of personal data issues.

Unfortunately, we did not assess this risk thoroughly during our project, although this should have been done. However, we only used official addresses from the Lausanne building directory, as well as images from Google Street View which blur sensitive information such as license plates and faces of passers-by, although ultimately, we did not use this image data. Despite this, appropriate data management while respecting the rights of individuals remains a key concern and should have been examined more thoroughly in our project.

Project Credits and Contact Information

This research was conducted by Youssef Belghmi and Hamza Morchid, graduate students specializing in Data Science at EPFL, as part of their semester project for the Machine Learning course (CS-433).

We extend our sincere gratitude to HERUS, particularly to Francisco Xavier Felix Martin Del Campo, our scientific collaborator at HERUS, for his invaluable collaboration, trust, unwavering support, and wise counsel throughout the execution of this project.

Should you have any inquiries, require additional information, or encounter any issues, please do not hesitate to reach out to us via the following email addresses : youssef.belghmi@epfl.ch and hamza.morchid@epfl.ch. We trust that you will find our project to be informative and insightful !

References

[1] Federal Office of Topography, swisstopo, ”Official directory of building addresses,” 2022.

[2] R. Tyleček and R. Šára, “Spatial pattern templates for recognition of objects with regular structure,” 2013.

[3] F. Korč and W. Förstner, “eTRIMS Image Database for interpreting images of man-made scenes,” 2009.

[4] R. Gadde, R. Marlet, and N. Paragios, “Learning grammars for architecture-specific facade parsing,” 2016.