BRAUDE
College of Engineering, karmiel

Capstone Project Phase A

The Impact of Urban Environment on Microclimate: A SegFormer-Based Study

24-2-R-19

Students:

Erik Pinhasov Nave Cohen

Lecturer:

Zakharia Frenkel

Table of contents

Abstract. Urban microclimates, driven by the built environment, grow with the expansion of cities in concert with the challenges of an increasingly extreme climate. Yet, most climate models lack the resolution to account for unique characteristics of urban areas, such as the Urban Heat Island effect. This research proposes a machine learning-based approach to predict urban microclimate conditions using high-resolution satellite imagery with particular emphasis on temperature variation and the impact of urban structure. By leveraging SegFormer, a modern transformer-based image segmentation model, we can accurately classify buildings, roads, vegetation, and more. These segmented objects will then be further analyzed in terms of, for example, Land Surface Temperature estimation and Surface Energy Balance models to estimate their contributions to localized climate effects, which would allow the exact prediction of UHI hotspots and other microclimate phenomena. Our approach prioritizes simplicity while achieving high performance by optimizing model parameters such as learning rates, epochs, and loss functions. The research aims to provide valuable insights for urban planners in designing more climate-resilient cities, offering a practical tool to address the growing challenges of urbanization and climate change.

Keywords: Urban Microclimate · Satellite Image Segmentation · SegFormer · Urban Heat Island (UHI) · Remote Sensing for Urban Planning

1 Introduction

By 2050, over 68% of the world's population will likely reside in cities, and rapid urbanization combined with climate change will pose increasing challenges for city planning [1]. Traditional climate data resolution and coverage are usually too low to effectively use in urban planning, as they fail to account for the unique characteristics of urban environments [2]. Additionally, most weather forecasting today does not sufficiently consider the complexities of urban microclimates, which are heavily influenced by factors such as building density, surface materials, and limited green spaces [1]. This paper, therefore, aims to discuss how these limitations could be addressed through advancements in satellite data analysis, making urban development more resilient and sustainable, with particular emphasis on the impact that urban structures and materials have on local climates [2].

Apart from the challenges regarding urban climate, perhaps the most famous one is the so-called Urban Heat Island effect (UHI). The UHI is a phenomenon in which urban areas are much warmer compared to their rural surroundings due to the concentration of heat-retaining materials and human activities. The UHI effect raises local temperatures, enhances energy demand, and degrades air quality, posing serious health risks to the general public. Understanding and reducing the UHI effect is crucial to increase urban resilience, particularly in fast-growing cities facing intensified climate change [5][10].

Several approaches have been taken to predict cities' climate with the help of meteorological networks, climate models, and remote-sensing technologies [5]. However, all these

approaches have their shortcomings. The most usable data are provided by meteorological networks, which, however, often have limited coverage due to sparsely distributed sensors. The climate models operate at resolutions too coarse to capture the local-scale variability needed for urban planning applications mostly [5]. While promising, remote sensing-based approaches often require more spatial and temporal resolution by dynamic urban environmental monitoring [6].

This paper talks about using high-resolution satellite imagery with machine learning techniques to better understand the impact of urban structures on climatic parameters. The high spatial and temporal resolution of satellite data, against the costliness of extensive networks of ground sensors, provides much better decision-making for urban planners and policymakers together with already available urban planning and development tools [6].

Such challenges involve stakeholders such as urban planners and municipal officials interested in long-term planning, environmental scientists who study the current condition of the climate in urban areas, residents who are exposed to hazards brought about by the climate, decision-makers who are interested in adaptation, and mitigation strategies, and real estate developers who would like to build climate-resilient buildings [7].

Approaches reviewed in this work contribute to urban planning strategies that deal better with climate-related problems. These can improve the practice of urban planning by adding new insights into ecologically appropriate, resilient, and sustainable urban development. By embedding these, cities can become more adaptable to the challenges the future brings forth due to climate change, thus making more informed and forward-looking decisions for planning.

2 Literature Review

This section will review the background and studies on urban microclimates, traditional methods and models, recent advancements, and the application of remote sensing and data sources in this field.

2.1 Urban Microclimate Background

Urban microclimate prediction has seen significant advancement over the past few decades due to the increasing need to understand the impacts of urbanization on local climates. With the expansion of cities, features of urban microclimates have attracted the growing interest of researchers. Early studies relied on field observations, but studies of the urban climate have increased with the development of satellite technology, computational models, and advanced data collection methods (Fig 1). These advancements have enabled more accurate predictions of urban climate behavior and better-informed urban planning strategies to improve sustainability and resilience [1]. By understanding and managing factors like temperature

variations, wind patterns, and humidity locally, urban planners can design cities that mitigate the negative effects of heat, reduce energy costs, and improve overall living conditions.

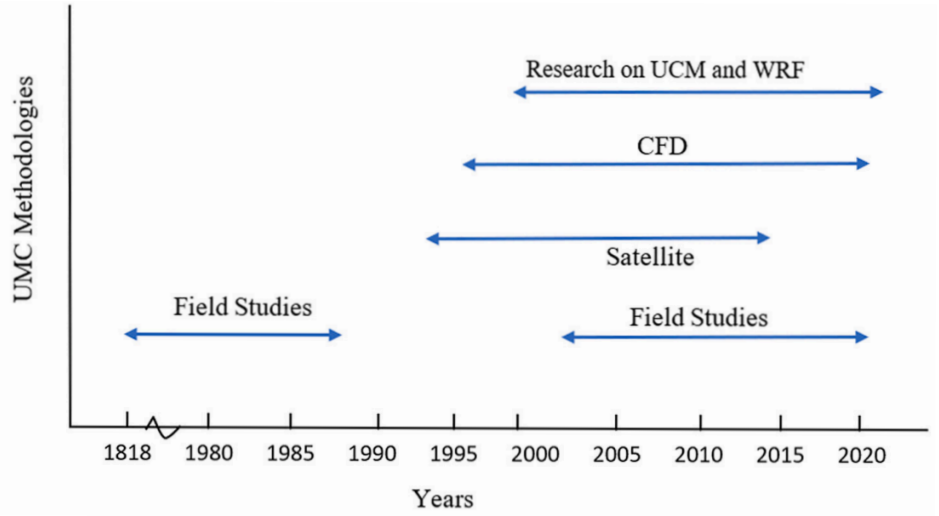

Fig. 1. A timeline of major focussed methodologies of urban microclimate research [1].

The Urban Heat Island (UHI) effect occurs when the temperatures in urban areas are higher than those of rural setups, mainly owing to several heat-absorbing materials like asphalt and concrete coupled with negligible vegetation. All these factors heighten energy consumption, degrade air quality, and bring hazards to public health due to the potential effects on the ecosystem [1][10]. Developing mitigation strategies in urban planning is essential through greening and using reflective material to lower UHI and enhance climate resilience [10][20]. In 2020, researchers employed an urban climate model that combined global climate scenarios and urban growth data to study the urban heat island (UHI) effect. They applied the Town Energy Balance (TEB) scheme, which tracks the energy exchanges between the buildings, streets, and atmosphere to assess how urban areas raise the temperatures in rural regions. The image in Fig. 2 shows the UHI of Brussels, comparing simulations with and without the TEB scheme [9]. This highlights the importance of using detailed urban models like TEB for accurate urban climate simulations.

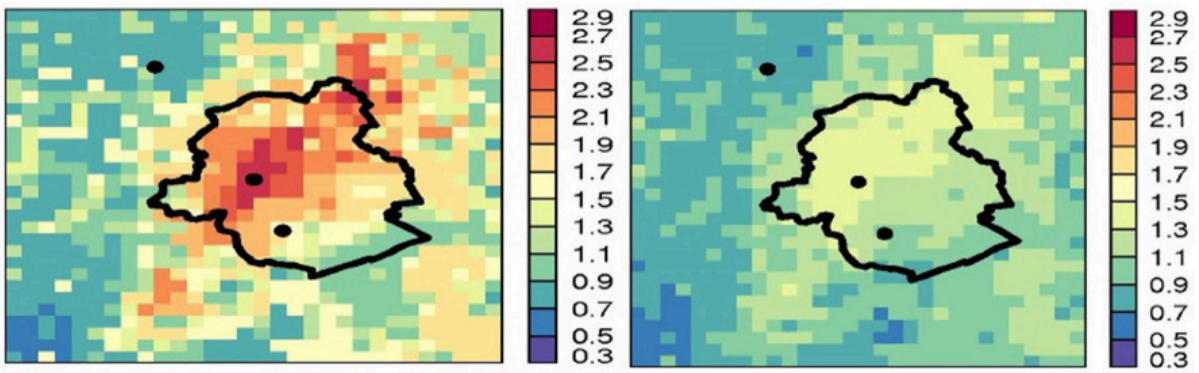

Fig. 2. The spatial distribution of 30-year average UHI [°C]. The left column uses the town energy balance (TEB) scheme, and the right column without TEB [9].

Green infrastructure (GI) helps mitigate urban heat by using vegetative elements such as green roofs, parks, and trees to cool the environment. It features lower surface temperatures, including evapotranspiration and shading [20]. Segmentation of green infrastructure in images taken of urban areas can be done to analyze further how these particular features factor into temperature regulation. This enables more specific assessments of how GI impacts urban microclimate and helps urban planners understand precisely how different features of the built environment contribute to local temperature variation.

In 2012, researchers found there are three factors influencing urban climate:

Vegetation: Plants cool urban areas by up to 1–4.7°C through evapotranspiration and shading, with the effects spreading out for distances as far as 1000 meters [20].

Water: Water bodies, such as ponds and fountains, cool the air by evaporation and heat absorption, with temperature drops of 1–3°C within 30–35 meters [20].

Built Form: Dense urban layouts trap heat through multiple reflections and reduce natural ventilation. A smaller height-to-width ratio provides much better airflow, but high-rise buildings often reduce wind speed and limit the cooling effect [20].

There are several key metrics and methods used to show how the urban structure and environment affect local temperature:

Land Surface Temperature (LST), derived from satellite thermal imagery, is an essential metric for understanding city surface heat patterns. LST is generally used to find hotspots and to analyze the characteristics of heat absorption and retention by various urban surfaces, such as buildings, roads, and green spaces [2][19].

Sky View Factor (SVF) measures the amount of sky visible from a given point. SVF directly influences the amount of solar radiation that urban surfaces receive and how heat is emitted back into the atmosphere. A lower SVF typically corresponds to more obstructed urban canyons, which trap heat [19].

Building density is the ratio between the area covered by buildings and the total land area. High building density is usually associated with poor ventilation and reduced cooling potential in the urban area. It is expected to reduce airflow and trap heat, enhancing the Urban Heat Island effect [19].

Floor Area Ratio (FAR) is the building's total floor area ratio against the occupied land area. A high value of FAR usually designates a highly built area with taller buildings that trap heat and decrease cooling by natural airflow [19].

2.2 Climate Modeling Methods

The Weather Research and Forecasting (WRF) model is a comprehensive atmospheric simulation system for research and operational use in numerical weather prediction (NWP).

WRF is a traditional model with a scalable architecture to support parallel computing. The model suits large-scale simulations such as weather forecasting, climate modeling, and air quality research. Physics modules and dynamic cores in this model help it simulate different regional and global weather phenomena [5].

Urban Canopy Models (UCMs) simulate the energy and mass exchanges between urban surfaces and the atmosphere, capturing the effects of buildings, roads, and vegetation. It is usually incorporated into larger weather forecasting models, such as The WRF model, to allow better urban climate simulations. These models efficiently capture the behavior of the UHI effect and, therefore, enable the evaluation of benefits through greening or other reflective materials over urban areas for lowering local temperatures [5].

Computational Fluid Dynamics (CFD) models simulate the movement of air and heat in urban environments, providing insights into airflow, pollutant dispersion, and temperature (Fig 3). ENVI-met is a widely used CFD tool designed for urban microclimate simulations, particularly the UHI effect. ENVI-met models the interactions between urban surfaces, vegetation, and atmosphere, enabling urban planners to assess the impact of green infrastructure and other mitigation strategies on microclimates [9][10].

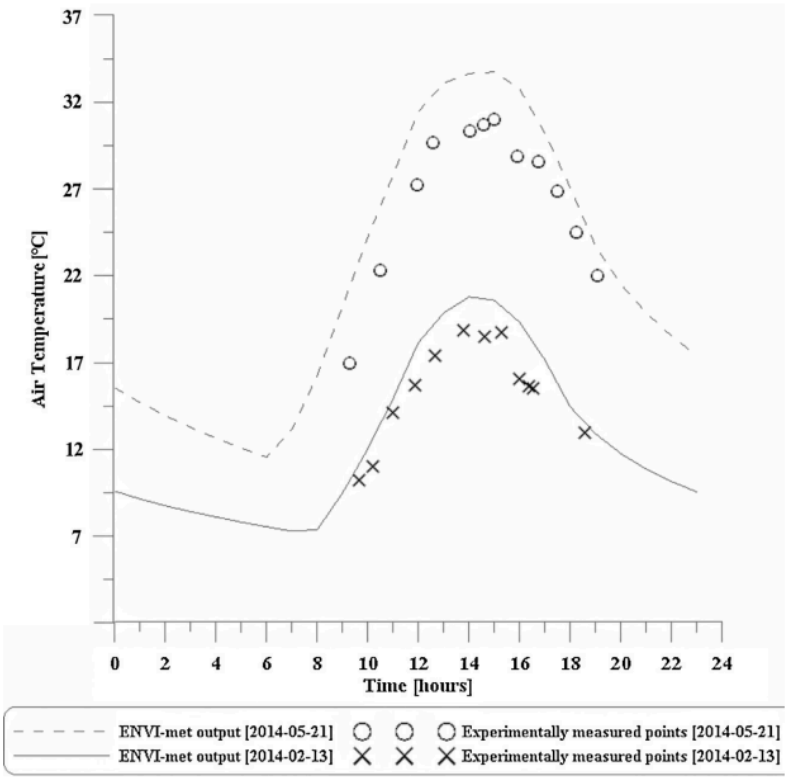

Fig. 3. Air temperature furnished by ENVI-met and experimentally measured values [10].

2.3 Remote Sensing

Optical and infrared Remote Sensing (RS) are two primary techniques for studying urban environments by capturing different aspects of surface properties. Optical RS records

reflected sunlight in visible and near-infrared wavelengths, providing detailed spatial information on urban features, such as buildings, roads, and vegetation [2]. Infrared RS measures the emitted thermal radiation of surfaces, recording data related to the surface temperatures, which can help understand how heat retention in the environment affects local temperature variations [2]. Researchers found that data fusion between optical and thermal imagery developed a more complete understanding of urban heat patterns. This advanced integration of such data types further enriches the observation of heat retention on the urban surface and offers a better understanding of the Urban Heat Island effect. This approach helps identify hotspots more precisely and improves understanding of how different urban materials contribute to temperature changes [6].

The satellite images capture heat islands and land-use changes (Fig 4), which are relevant for urban climate monitoring. Estimating Land Surface Temperature (LST) by thermal infrared remote sensing from satellites like Landsat enables planners to locate anomalies and adopt necessary mitigation strategies accordingly [2]. Sentinel-2 provides high-resolution multispectral imagery but has some drawbacks when its application involves micro-level information in urban areas [6]. Further studies are required regarding data fusion techniques, where information from multiple sources is integrated [6]. Additionally, UAV (Unmanned Aerial Vehicle) images are becoming increasingly relevant for urban microclimate studies. UAVs provide high-resolution imagery with flexibility in capturing low-altitude, site-specific data, which complements satellite imagery by offering detailed insights into small-scale urban environments.

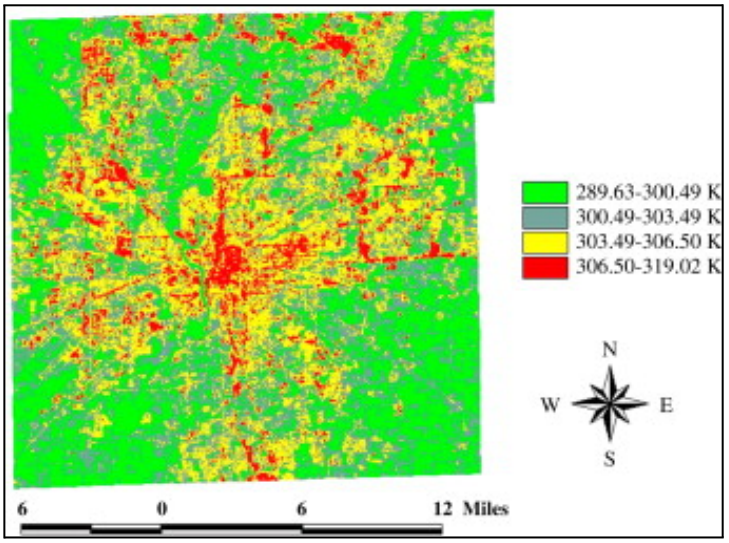

Fig. 4. Emissivity-corrected LST map derived from a Landsat thermal infrared image [2].

2.4 Machine Learning for Urban Analysis

Machine learning (ML) models have emerged as powerful tools for analyzing and predicting urban microclimates by identifying complex relationships between climate variables and urban form, significantly improving air temperature predictions in urban settings. Convolutional Neural Networks (CNNs) and Random Forest models are commonly used to

predict Land Surface Temperature (LST) and classify urban features (Fig 5), which provides better accuracy for research in urban climate studies [8].

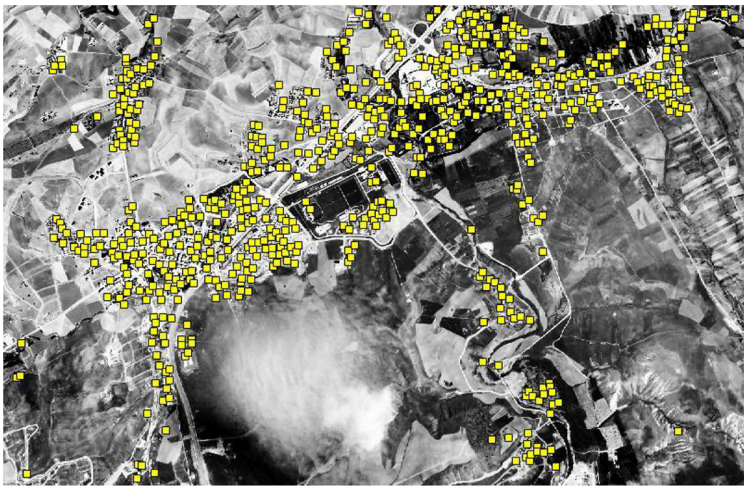

Fig. 5. Detected buildings from a satellite image of a neighborhood in Turkey [8].

Image segmentation is a core technique in computer vision that involves partitioning an image into distinct regions or objects (Fig 6) to simplify the representation and make analysis more efficient. Deep learning has improved segmentation accuracy by enabling the models to learn complex image patterns and features. One type of segmentation is semantic segmentation, which assigns a class label to every pixel in an image and provides fine and detailed scene understanding, enabling the in-depth and accurate visualization of scenes [14].

Semantic segmentation in the urban microclimate analysis refers to identifying and classifying features like buildings, roads, and vegetation from satellite imagery. The high-level classification shall allow for further understanding of how these features impact the local microclimate and, therefore, more accurate predictions for planning and climate studies.

Fig. 6. Three sample images and segmentation maps from the PASCAL context dataset.

Data augmentation is a technique that generalizes ML models by incorporating transformations on the training data, reducing overfitting and increasing performance on previously unseen data. This technique is essential in the field of remote sensing, where collecting big and diverse labeled datasets might be challenging. Some standard methods applied to simulate real-world changing environmental conditions are scaling, random cropping, spectral shift, noise injection, and more (Fig 7) for lighting changes, cloud cover, and sensor angles, increasing the models' robustness [13].

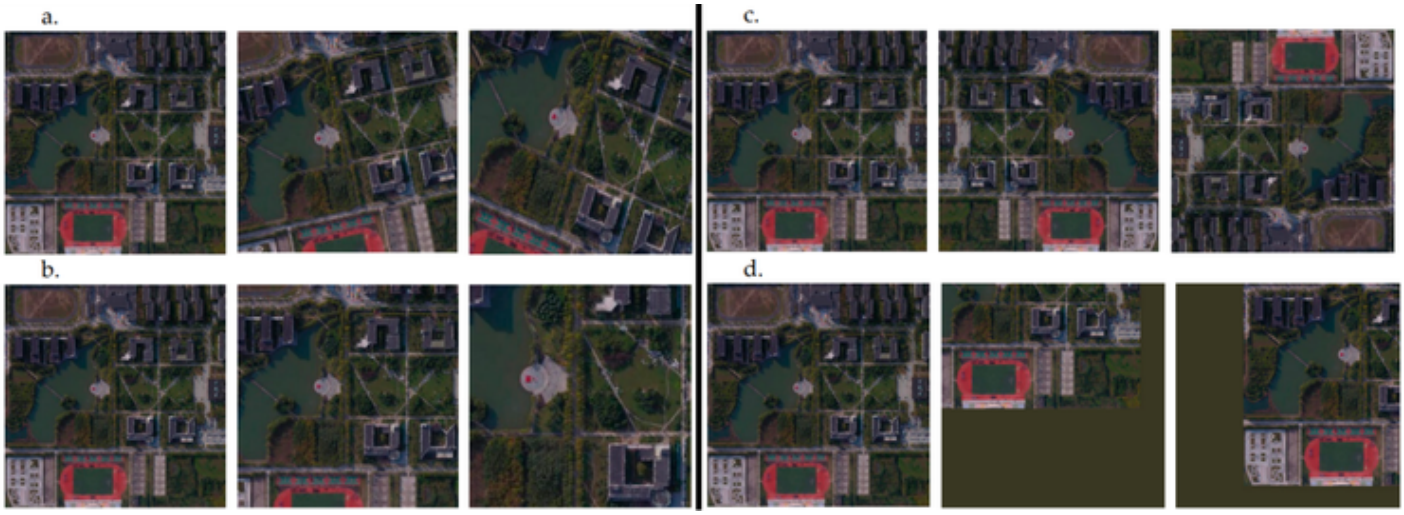

Fig. 7. (a-d) stands for rotation, zoom, flip, and shift. The first column of images is the original image, and the second and third columns are the images after transformation [13].

Land Use and Land Cover (LULC) refers to data describing the Earth's surface features derived from urban, agriculture, and land cover as vegetation and water. In 2024, researchers applied a machine learning model together with LULC data to forecast urban microclimate weather conditions, they used the Geo-LSTM-Kriging model, a spatio-temporal fusion approach, to enhance the predictions of temperature and humidity. The image in Fig 8 shows the temperature prediction errors of four data sources where their method consistently performed better [12].

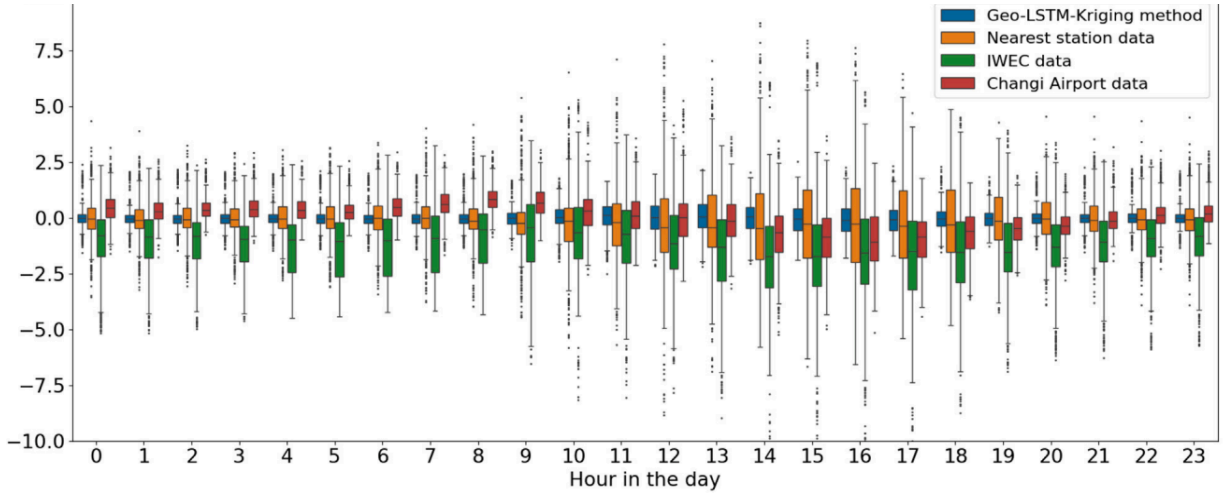

Fig. 8. Comparison of Temperature prediction error (predicted minus observed values) [12].

2.5 Data Sources for Urban Analysis

LoveDA dataset consists of RGB images with a resolution of 0.3 m from urban and rural areas in China and has a total area of 536.15 km². It includes 5,987 images and pixel-level semantic segmentation annotations of 20,658 labeled objects across seven classes. It encourages diversity by covering highly populated cities and less developed rural areas [16].

WHU Building dataset consists of aerial imagery of Christchurch, New Zealand, and satellite images of several other cities worldwide. It focuses on building segmentation and includes pixel-level annotations and an aerial subset containing over 220,000 buildings at a 0.075 m spatial resolution (later downsampled to 0.3 m).

DeepGlobe dataset is a large-scale satellite imagery dataset for land cover classification, road extraction, and building detection. It consists of RGB images in very high resolution taken from different parts of the world at 50 cm per pixel resolution. The dataset covers a wide range of urban and rural areas with annotations on seven different land cover classes [17].

Inria Aerial Image Labeling dataset comprises high-resolution aerial imagery taken from the United States and Austria urban areas. This aerial view has a spatial resolution of 0.3 meters and covers an area of 810 km². The train/test sets are divided by cities and not randomly by pixels, enabling the exact assessment of generalization [18].

Dataset	Total Images	Images Size (px)	Resolution (per pixel)	Classes
LoveDA	5,987	1024 × 1024	30 cm	Background, Road, Building, Forest, Water, Agriculture, Barren
DeepGlobe	1,146	2448 × 2448	50 cm	Urban land, Agriculture land, Rangeland, Forest land, Water, Barren land, Unknown
WHU Building	8,189	512 × 512	7.5 cm, 30 cm	Building, Background
Inria Aerial Image Labeling	360	5000 × 5000	30 cm	Building, Background

Table. 1. Overview of the key specifications of the datasets.

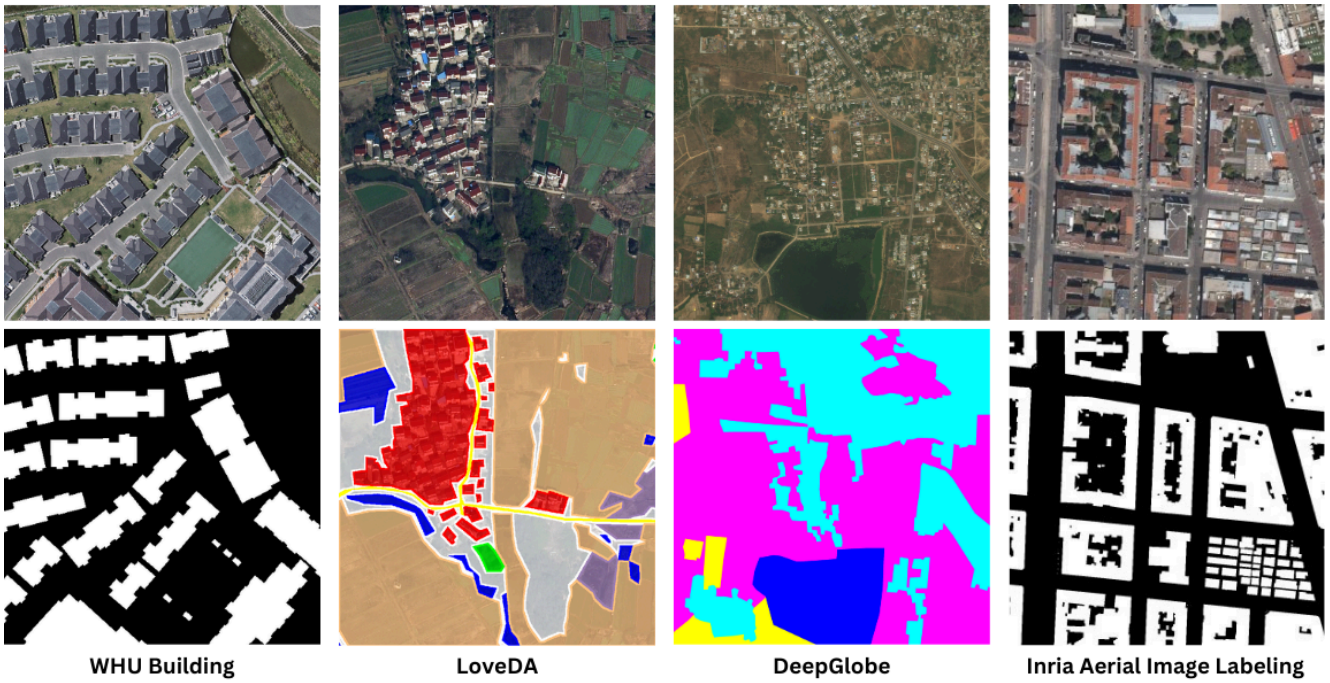

Fig. 11. Examples of image and its segmentation map of each dataset.

Google Earth Engine (GEE) is a cloud-based platform that helps researchers access and analyze vast amounts of geospatial data, including multi-spectral and infrared images from such satellites as Landsat and Sentinel-2 (Fig 12). GEE supports ML integration and advanced analytics, enabling users to monitor large-scale environments by extracting information like land surface temperature, vegetation indices, and change-over-time land-use changes. These features help analyze the interaction between urban structures and environmental factors, allowing for a study of how different surfaces like buildings, roadways, and green spaces contribute to thermal properties and heat distribution. [11].

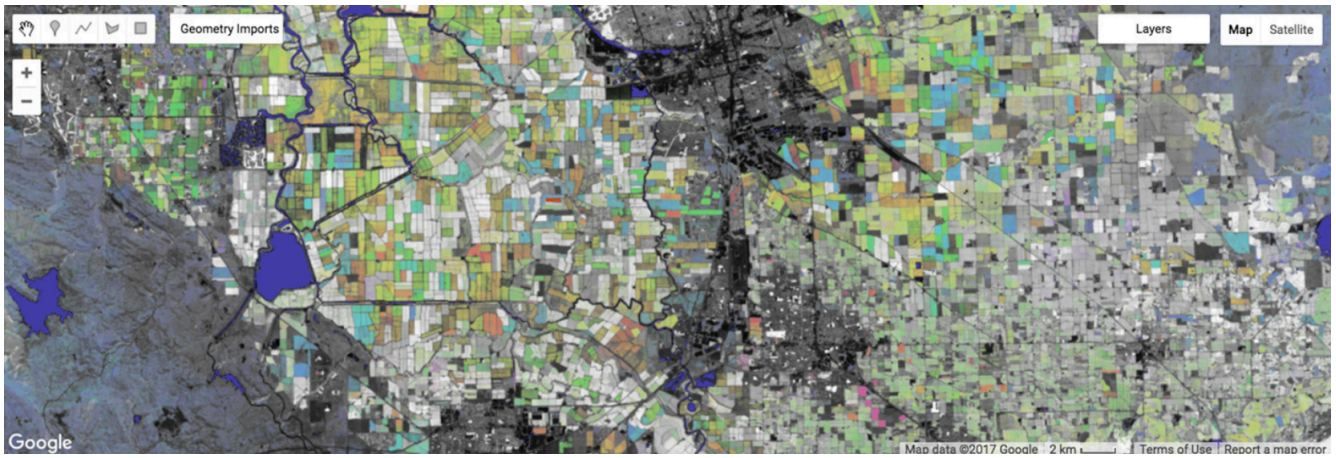

Fig. 12. Satellite image example from GEE [11].

3 SegFormer background

SegFormer follows a hierarchical multi-stage architecture similar to U-Net but replaces the convolutional layers with transformers in the encoder. The nature of self-attention within transformers means they are especially better positioned to model long-range dependencies. Hence, SegFormer can capture both the minute local information and the global context simultaneously. The encoder divides the input images into patches and embeds them into multiple transformer layers. These layers apply self-attention to capture relations between pixels in the entire image dynamically, performing very well on tasks requiring finer details and a broader context, such as semantic segmentation [15].

The SegFormer backbone consists of a hierarchical transformer encoder and an MLP-based decoder (Fig 13). The transformer encoder captures multiscale features by progressively shrinking the spatial resolution while enlarging the receptive field, enabling the model to capture fine-grained details and broader contextual information from the images. Unlike many other transformer models, SegFormer does not require position embeddings and is adaptable to varying input sizes [15].

It effectively fuses multi-level features of the encoder using the MLP-based decoder to generate the final segmentation map. This further leads to a streamlined architecture whereby SegFormer can reach high accuracy without compromising memory efficiency and computational speed. As a result, no complex computation layers or intensive post-processing steps are required for the model [15]. It, therefore, tends to be very efficient for segmentation tasks. In general, SegFormer represents a new architecture that achieves a great balance between high performance and resource efficiency. SegFormer can hence be adapted to a wide range of applications, from real-time to large-scale segmentation tasks.

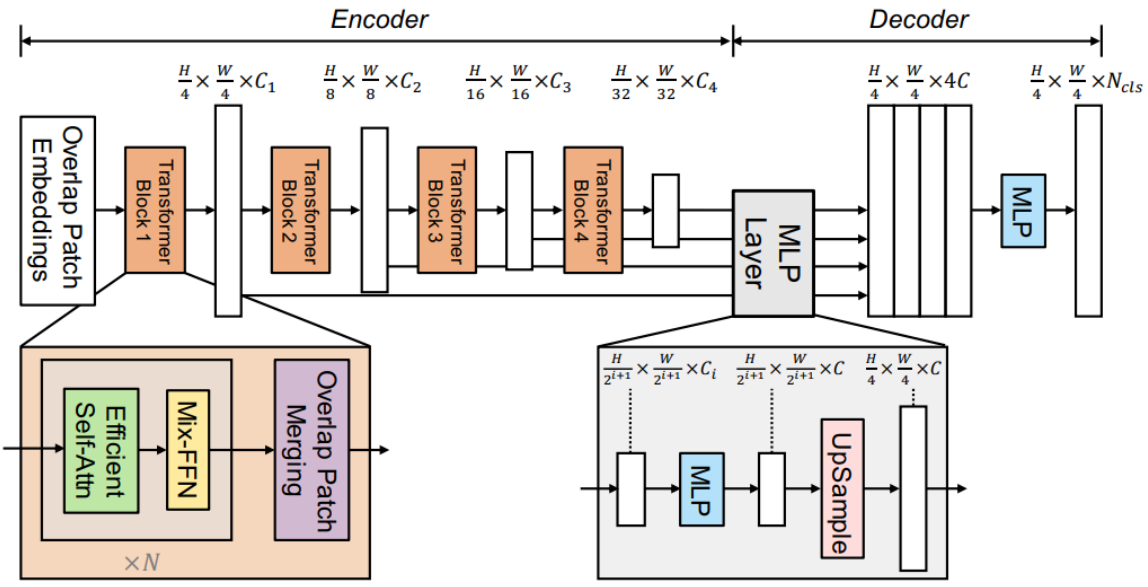

Fig. 13. The proposed SegFormer framework consists of a hierarchical transformer encoder and an MLP-based decoder [15].

3.1 Pre-trained SegFormer Models

SegFormer has an entire family of pre-trained models, ranging from small to large, depending on the task and hardware constraints. These range from SegFormer-B0 through SegFormer-B5, wherein the number of layers, embedding dimensionality, and even the size of feature maps across stages change. The "B" in these model names stands for "Base" and the larger the number, the bigger the model. Each of these variants is trained with large-scale data, hence it can be a very effective starting point for fine-tuning different segmentation tasks (Fig 14) [15].

The most lightweight and efficient model, the SegFormer-B0, is supposed to be applied in resource-constrained environments and low-latency real-time applications.

The SegFormer-B1 balances speed with accuracy very well, extracting more features at very minimal additional computation.

SegFormer-B2 scales up the capacity with more transformer layers and much larger feature maps, in a trade-off between efficiency and accuracy.

SegFormer-B3, which improves accuracy for more complicated scenes, finding its perfect fit in urban planning or medical imaging where high boundary precision is required.

SegFormer-B4 enables high-resolution segmentation of minute scenes due to natural environments or aerial imagery.

The largest model, SegFormer-B5, targets state-of-the-art accuracy for resource-consuming tasks such as medical imaging and autonomous driving.

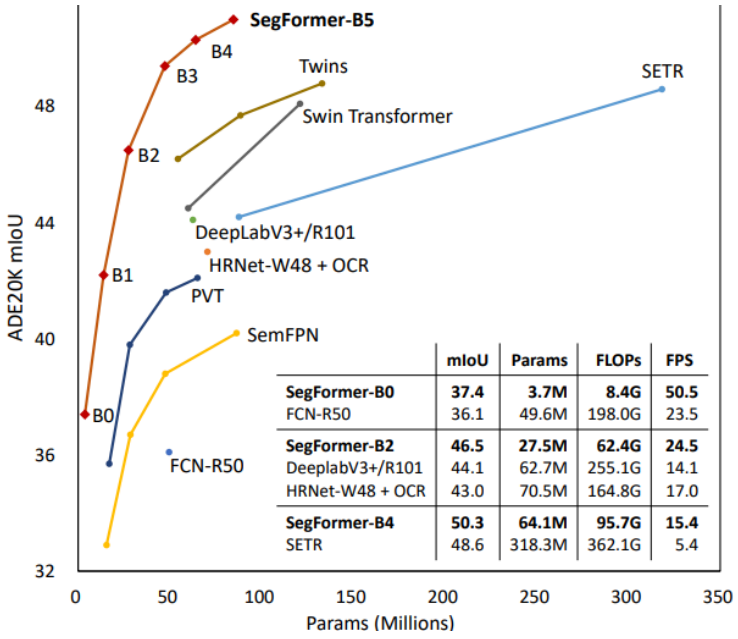

Fig. 14. Performance vs. model efficiency on ADE20K of different models [15].

3.2 Pre-training Datasets

1. **ImageNet-1K**
   Many variants were pre-trained using the ImageNet-1K dataset as a large-scale image classification dataset, and for the model to learn general visual features, it will be transferred to segmentation tasks with fine-tuning. Although it is not a segmentation dataset, diversity helps build a solid foundation for later use.

2. **ADE20K**
   SegFormer models also use ADE20K for pre-training, a dataset with 150 object categories ideal for semantic segmentation. Its variety of scenes—urban, natural, and indoor—enables the model to generalize well to real-world segmentation tasks.

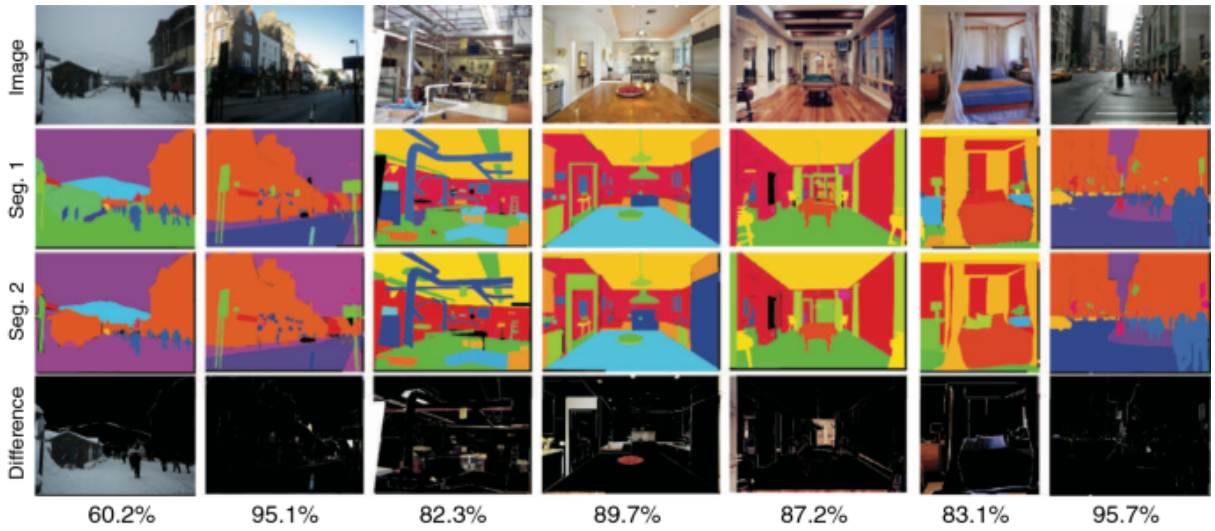

Fig. 15. Semantic segmentation of B1, B2 SegFormer variants on ADE20K dataset.

SegFormer models have been applied to various real-world applications, proving their versatility across different domains. Variants of SegFormer are fine-tuned for medical image segmentation of organs and tissues, locating the exact boundaries and classifying regions with high accuracy. In urban planning studies and climate-related analyses, SegFormer has been employed for land cover segmentation, finding changes in urban development areas over time, and identifying buildings, roads, water bodies, and vegetation with high accuracy.

Besides, SegFormer is so light that it fits perfectly for resource-constrained environments such as mobile or embedded systems since they typically require real-time performance. Even larger variants, like SegFormer-B5, can be deployed on powerful GPUs and thus yield high-resolution outcomes, making the model suitable for lightweight and high-complexity tasks.

SegFormer marks a huge leap in segmentation models by marrying flexibility in transformers with lightweight, efficient architectures. The various pre-trained models let the user choose appropriate models depending on the user task's complexity and hardware limitations. With

its scalable architecture added to pre-trained weights, SegFormer raises the bar for semantic segmentation tasks across several domains [15].

4 Expected Achievements

This project aims to develop a comprehensive toolset that includes a trained segmentation model, a dataset documenting urban feature changes, and actionable findings to support urban planners in decision-making. The first step will be training a SegFormer model on remote sensing imagery to accurately classify buildings, roads, water bodies, forests, barren, and agriculture from a satellite image.

The trained model would be applied to the satellite images of an urban area for various years. It would do segmentations for each year, generating segmentation maps and calculating the percentage share of each class in that area. Additionally, temperature data corresponding to each particular date will be gathered to directly compare changes in urban features versus temperature variations.

After identifying the segmented maps and calculating the area percentages of urban features for each year, the project will conduct a correlation analysis to determine how these changes are associated with temperature changes over time, such as a rise in building density or a decline in green space.

This information would lead to a dataset that can be useful to urban planners in devising heat-reducing measures and understanding how infrastructure development in buildings and roads affects how heat is distributed.

Success Criteria:

- Accuracy: The segmentation model should achieve over 85% accuracy in classifying the features in the urban setting.
- Model evaluation on the test dataset will achieve an IoU / CE / Dice of 85%.
- The model will result as well as or be better than the baseline models on benchmark datasets like “DeepGlobe Land Cover Classification Challenge” and others, with metrics reported clearly.
- Correlations findings: Clear and actionable relationships between urban changes and temperature variations.
- Researcher usability: The dataset and results need to be usable, explicit, and convenient for analysis to provide direct insight into urban development and heat mitigations for urban planners and researchers.

5 Proposed Research Plan

The proposed project will develop a model for the semantic segmentation of urban and rural images using the SegFormer framework. Our goal is to accurately identify and classify six major land cover classes: buildings, roads, water bodies, forests, barren, and agriculture. We aim to use the model to analyze images of selected urban areas from different years, combining them with historical temperature data. This will create a dataset for analyzing urban climate changes over time.

We will, therefore, test various pre-trained SegFormer models and fine-tune a set of hyperparameters in order to achieve better results.

5.1 Hyperparameters

Training of a deep learning model involves various hyper-parameters that are crucial for the performance of the model. In this research project, hyper-parameters will be considered and compared to get the best segmentation. We will try to suggest the best hyperparameters that can provide improved model performance. The models' accuracy will be compared using a full assessment. Moreover, architecture learning behavior will be studied in order to understand if any change is needed, for example, about the learning rate needed to avoid a vanishing gradient or about the batch size in cases where the model accuracy is too low.

5.1.1 Learning Rate

The learning rate controls how fast the model learns from the training data. It is a very important hyperparameter in the process of segmentation. The high learning rate can result in faster learning with this model but may cause overshooting of the best solution, leading to poor performance. In contrast to that, the low learning rate enables slower learning and can allow the model to converge to a more optimal solution. This work will compare the influence of different initial learning rates. The learning rate will range from 1e-4 to 5e-5 using the AdamW optimizer, which 5e-5 is typically preferred for fine-tuning [3].

5.1.2 Optimization with Optuna

We will use Optuna, which is one of the state-of-the-art optimization frameworks for hyperparameter tuning. Unlike traditional grid search methods, Optuna performs hyperparameter optimization in a Bayesian optimization style with a variant of Tree-structured Parzen Estimator (TPE). This allows the search to be conducted in an adaptive manner, concentrating the effort around the most promising areas and dynamically adjusting based on past evaluations. Besides, the pruning feature of Optuna will be applied to stop those poorly promising trials as early as possible and save the computing resources during model training. In this way, we will try to find an optimal set of hyperparameters that contributes to improved segmentation performance.

5.1.3 Epochs

In training related to machine learning, an epoch is a single complete pass of the whole training dataset through the model. At each epoch, the model views all the training samples and modifies its internal parameters or weights via backpropagation and optimization. While higher training epochs normally improve performance, doing so beyond an optimal limit can lead to overfitting. We will try to avoid that by having an early stopping mechanism for our research, which stops the training if the performance on the validation set has stopped improving. In this manner, efficient training will be assured, and there will be no assumptions about any arbitrary numbers of epochs because this whole process will self-complete once optimal performance is reached.

5.1.4 Batch size

The batch size is, in essence, the number of training examples fed to the model within a single iteration before it updates its internal parameters, which are called weights. Instead of passing all the data at once, which would be an expensive task, the data gets divided into small subsets called batches. On every iteration, the model processes one batch and updates the weights based on the collective error of that batch. Our research will test its models using 16, 32, and 64 batch sizes. Smaller batch size frequency is higher and generally requires less memory but will be noisy during the training. A larger batch size might be smoother and get more stable updates; however, it will require more memory and maybe a little more time per iteration. Batch size impacts model performances and efficiency a lot during training.

5.1.5 Loss functions

Loss functions play an important role in model performances. These functions can quantify the difference between predicted and true segmentation maps. The chosen loss function is going to have a huge impact on how well the model will learn and perform. Common choices include Cross-Entropy Loss, Dice Loss, and IoU Loss. Some of these most common ones will be compared in this work [4].

Dice Loss measures overlap between the predicted segmentation and ground truth. This loss is designed to handle class imbalance by focusing on the overlap between predicted and true segments. It can be defined by an equation for Dice Loss as follows.

$$L_{dice} = 1 - \frac{2\sum_{i=1}^{N} p_i g_i + \epsilon}{\sum_{i=1}^{N} p_i^{\,2} + \sum_{i=1}^{N} g_i^{\,2} + \epsilon}$$

$$N - number,\ p_i - predicted\ probability,\ g_i - ground\ truth\ binary\ label,$$
$$,\ \epsilon - small\ constant\ to\ avoid\ division\ by\ zero$$

Categorical Cross-Entropy Loss is widely used in multi-class segmentation tasks. It measures the dissimilarity between the predicted probability distribution and the actual distribution of classes. This loss is computed by comparing the predicted class probabilities against the ground truth labels, with the goal of driving the probability of the correct class high and the rest low. The Categorical Cross-Entropy Loss between the prediction samples p and the ground-truth annotation g is defined as follows.

$$L_{CE} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y_{i,c} log(\hat{y}_{i,c})$$

$$y_{i,c} - ground\ truth\ label\ for\ each\ class\ c,\ \hat{y}_{i,c} - predicted\ probability\ for\ each\ class\ c$$

$$N - total\ number\ of\ pixels,\ C - total\ number\ of\ classes$$

The idea here is that the Combined Dice + Cross-Entropy Loss tries to combine the strengths of Dice Loss and Categorical Cross-Entropy Loss into a single loss. The combined loss function then balances out both metrics' benefits, offering a wider optimization objective that considers both overlap and pixel-wise classification accuracy. This could be written as the following equation.

$$L_{combined} = \alpha * L_{Dice} + (1 - \alpha) * L_{CE}$$

$$\alpha - weight\ factor\ that\ balances\ the\ contribution\ of\ each\ loss\ component\ [0, 1]$$

5.1.6 Evaluation metrics

Mean Intersection Over Union (IoU) is the measure for semantic segmentation tasks, referring to the average overlap of the predicted segments versus the ground truth segments over all classes. The computation it does is to calculate IoU over each class individually and then average the values. mIoU gives a view of the overall performance; thus, in multiclass segmentation tasks, it sees how well the model performs across all classes.

$$Mean\ IoU = \frac{1}{C}\sum_{i=1}^{C}\frac{|Predicted_i \cap Ground\ Truth_i|}{|Predicted_i \cup Ground\ Truth_i|}$$

$$C - the\ total\ number\ of\ classes$$

The second widely used evaluation metric for segmentation is the Dice Similarity Coefficient (DSC). We mentioned the dice coefficient loss in the previous section. Using the loss Dice Loss function as described above, the dice coefficient DSC can be described as follows.

$$DSC = 1 - L_{Dice}$$

5.2 Methodology

The main datasets used in this work are WHU Building and Inria Aerial Image Labeling datasets for binary segmentation and LoveDA and DeepGlobe datasets for multi-class segmentation.

5.2.1 Preprocessing

In the preprocessing step, since each image in the LoveDA dataset has a size of 1024×1024 pixels, we divide those images into patches of smaller 512×512 pixels size images. We can calculate the number of patches along each dimension by using the following formulas:

$$Number\ of\ patches\ along\ the\ width\ = \frac{W_{image}}{W_{patch}} = \frac{1024}{512} = 2$$

$$Number\ of\ patches\ along\ the\ width\ = \frac{H_{image}}{H_{patch}} = \frac{1024}{512} = 2$$

Since original image sizes are exactly divisible by the patch sizes, no padding is required.

$$Total\ number\ of\ patches\ per\ image\ = \frac{W_{image}}{W_{patch}} * \frac{H_{image}}{H_{patch}} = 2\ *\ 2 = 4$$

Since the total labeled images in the LoveDA dataset is 4,191 images, we get

$$Total\ number\ of\ images/patches\ = 4191\ *\ 4 = 16,764$$

Since the dataset contains 1,833 images for an urban environment and 2,358 for a rural one, for the LoveDA dataset, we kept the same urban-to-rural ratio during train, validation, and test splits. The whole dataset was split into 80% for training, 10% for validation, and 10% for testing without changing the share of urban and rural classes [16].

With this in mind, first of all, the dataset needed to be divided into two subsets: urban and rural. In the next step, each subset was then further divided into train, validation, and test sets in the 80/10/10 ratio individually. Then, the resulting train, validation, and test sets from both subsets are combined to form the final splits. This ensures that the final splits retain the same urban-to-rural ratio as the original in both training and evaluation. This can ensure equal representation for both urban and rural areas in the testing and training of the model.

For the Inria Aerial Image Labeling dataset, which contains high-resolution images of 5000 × 5000 pixels [18]. We will also divide the images into 512 × 512 patches. The number of patches per image can be calculated similarly:

$$Number\ of\ patches\ along\ the\ width\ = \frac{W_{image}}{W_{patch}} = \frac{5000}{512} \approx 9.8$$

$$Number\ of\ patches\ along\ the\ width\ = \frac{H_{image}}{H_{patch}} = \frac{5000}{512} \approx 9.8$$

Since the image dimensions are not perfectly divisible by 512, some padding will be applied to ensure the patches are of uniform size. The total number of patches per image will be approximately:

$$Total\ number\ of\ patches\ per\ image = \frac{W_{image}}{W_{patch}} * \frac{H_{image}}{H_{patch}} = 10 * 10 = 100$$

Since the total labeled images in the Inria aerial image labeling dataset is 360 images, we get

$$Total\ number\ of\ images/patches = 360 * 100 = 36,000$$

For the WHU Building dataset, where the images are 512 × 512 pixels, no further division into patches will be necessary, as the image size already matches the desired patch size.

WHU Building and Inria Aerial datasets: images will be concatenated, and then the splits for train, validation, and testing will be applied. Both deal with building segmentation and are thus appropriate for training together. Since the WHU dataset is made up of images of size 512 × 512 pixels, and the Inria dataset will be cut into patches of 512 × 512 pixels, we can easily merge them. This unified dataset will then be divided into training, validation, and testing sets in the ratio 80/10/10. In that way, both datasets will have a balanced and representative distribution so as to let the model benefit from the various strengths of each dataset. The combined dataset shall provide diverse urban and suburban building structures that will enhance the model's generalizing ability across geographical regions and image sources.

5.2.2 Training Procedure

In this work, the Training Procedure section is supposed to design a unified data loader for all four datasets: LoveDA, DeepGlobe, WHU Building, and Inria Aerial Image Labeling. The data loader dynamically applies dataset-specific logic to use the right segmentation task for training on WHU and Inria binary and LoveDA multi-class tasks.

Instead, it would be trained on a combination of these datasets together, hence enabling the model to understand both binary and multi-class segmentation tasks. Training on complex multi-class tasks from the LoveDA DeepGlobe datasets and simpler binary ones from the WHU and Inria datasets it would give robust feature representations that generalize across different challenges in segmentation.

In this, unified training features learning across datasets to improve the performance of the model for binary and multi-class segmentations. The idea of training all datasets together will make the model capture the diversity of both the urban and rural settings while enhancing its capability to find buildings across sources and image conditions.

5.2.3 Data Augmentation

For enhancement in model robustness and generalization, a number of data augmentation methods will be performed dynamically while training. These augmentations aim to deal with certain specific challenges that WHU Building, Inria Aerial Building, DeepGlobe, and LoveDA datasets introduce, such as light conditions, geographic diversity, and noise or occlusion.

Geometric Transformation
Random cropping and rotations introduce spatial variability, helping the model recognize features in different orientations across datasets.

Color and Brightness Changes
The changes in brightness, contrast, hue, and saturation are used to simulate different lighting conditions and seasonal changes.

Noise and occlusion simulation
Gaussian noise and occlusions, including cloud and shadow, make the model more robust.

Elastic Distortions
Elastic and optical distortions prepare the model for generalizing natural land cover in LoveDA and for differently oriented satellite perspectives of the scene.

Cutout (Random Erasing)
This forces the model to understand the surroundings much better because it randomly erases parts of images, thus being more robust in occlusions for all datasets.

These augmentation techniques will improve robustness in generalizing across varying conditions and environments for overall better performance.

5.2.4 Dataset creation

In our research, we will generate a dataset of diverse urban areas across multiple time periods to analyze the relationship between changes in land usage and temperature variations. For example, we will prepare satellite images, segmentation maps, and temperature graphs for the same urban area from the years 2000, 2005, 2010, 2015, 2020, and 2024, providing a detailed view of how urban development impacts local climate over time (Table. 2.).

Our main task will be mapping changes in the urban landscape and focusing our attention on key elements like buildings, roads, green spaces, and water bodies. For each time period, there should be segmentation maps that indicate and quantify the particular percentage of those elements that fall into each image. Quantification would allow observing how land use has shifted over time and detailed views on changes in urban areas.

We are going to use our fine-tuned model for the classification and segmentation of these land use elements. In addition, we will provide the segmentation maps, including temperature data, and match them with the respective years and regions to enable us to examine if there is a possible association between urbanization and infrastructure changes to changes in

temperature (Fig. 16). This study aims to identify the percentage of buildings, roads, green spaces, and other categories of land use to answer whether specific types of changes such as high rises or low increases in density of buildings, reduction in green areas, have marked temperature changes.

Our dataset will be based on open-source satellite imagery repositories and temperature records from reliable weather data sources. This dataset will serve to train predictive models that could forecast future changes in temperature based on projected urban developments. The research could have broader implications for urban planners and researchers by providing insight into how growth and land use decisions may influence local climate patterns.

year	location_id	building percentage	road percentage	forest percentage	water percentage	temperature
2000	location_1	40	15	35	10	30.5
2005	location_1	45	18	28	9	32.1
2010	location_1	50	20	22	8	33.8
2015	location_1	55	22	18	5	35.2
2020	location_1	60	24	14	2	36.9

Table. 2 Example of the result dataset that illustrates Land Use Changes and Temperature for Multiple Locations (2000–2020).

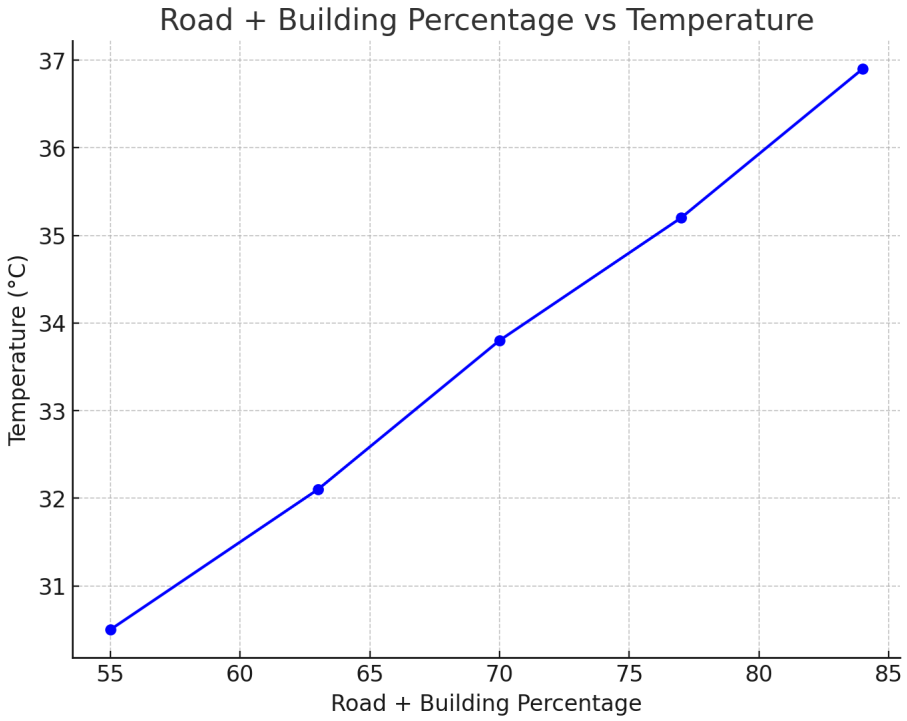

Fig. 16. Example of analysis from the dataset that illustrates the relationship between Roads and Building Percentage vs Temperature (2000–2020).

6 Evaluation/Verification Plan

Step	Verification/Use Case	Expected Outcome
1	The dataset successfully loaded and preprocessed	Datasets are preprocessed and patched as explained in the formulas. Number of patches equivalent to formulas outcome, all images have equivalent masks
2	Model initialization and forward pass test: Verify the model initializes properly and performs a forward pass with mock input.	The model is initialized without errors. The model forward pass produces output with the expected shape (batch_size, num_classes, height, and width) based on mock image input.
3	Loss function test: Check that the model computes the loss correctly, given the input and ground truth mask.	The model's computed loss (e.g., CrossEntropyLoss) is greater than zero and valid. No errors or crashes during computation.
4	Prediction output test: Ensure that the model generates valid segmentation predictions (classes) from the forward pass.	The predictions are of the correct shape and contain valid class indices (e.g., between 0 and num_classes - 1).
5	Edge case handling test: Test how the model handles edge cases such as large inputs or invalid inputs (e.g., incorrect dimensions).	The model handles edge cases appropriately: for large images, it produces correct output; for invalid input (wrong dimensions), it raises appropriate errors.
6	Training step test: Simulate a single training step, including a forward pass, loss computation, backpropagation, and optimizer step.	The model completes a training step successfully, with loss decreasing and the optimizer updating the model's parameters without errors.
7	Early stopping functionality test: Test that the early stopping mechanism halts training when the validation loss stops improving	Early stopping is triggered correctly after X epochs without validation loss improvement.

7 General Specification

Functional Requirements:

- Segmentation Model: Satellite imagery is categorized by a SegFormer model into buildings, roads, water bodies, forests, barren, and agriculture.
- Data Processing: Satellite imagery of successive years pre-processed; segmentation and creation of maps corresponding to each time period.
- Temporal Analysis: Calculating the percentage of each feature for each year and its correlation with temperature data.
- Hyperparameter Tuning: Perform hyperparameter tuning to get the model's optimal performance by dynamically adjusting the learning rates and batch sizes.
- Data Augmentation: Dynamic data augmentation with the view to increasing the robustness of model performance under varying conditions.
- Dataset Creation: Segmented maps and temperature data over the years to analyze the changes in land use and variation in temperature.

Non-Functional Requirements:

- Scalability: The system should handle large datasets and segmentation maps for high-resolution imagery with ease. This will also include the usage of different types of data with different classes in order to train our model.
- Accuracy: The system should classify features of an urban environment with accuracy above 85%.
- Efficiency: Employ methods for efficient model training and performance optimization to ensure that it works fast.
- Usability: The system should provide clear and understandable dataset formatting to the researchers for the use of data findings and segmentation results.
- Robustness: Implementation of data augmentation techniques along with mechanisms of early stopping so as not to be overfitted and to ensure generalization in a wide set of urban environments.

Expected Challenges:

- The lack of an appropriate dataset poses a significant challenge for achieving accurate and consistent segmentation results.
- Dataset Variability: Inconsistent quality and resolution across different years of the same area could affect the precision of segmentation.
- Model Generalization: Ensuring good generalization performance for both urban and rural areas at different periods of time may be difficult.
- Computational Resources: Large-scale satellite datasets for training may require significant GPU or cloud resources, which are not always available.

- Dataset Integration: Merging multiple datasets with different formats and resolutions can complicate data management and training. In addition, using different datasets with varying class definitions could result in segmentations that are inconsistent.
- Temperature Data Alignment: Aligning satellite images with accurate temperature data may be challenging and affect correlation results.
- Segmentation Performance: Achieving high-detail segmentation for all urban features without missing smaller objects can be difficult.
- Correlation Interpretation: Drawing actionable insights from the correlation between urban changes and temperature may be complex.

Research Tools:

- Google Scholar: The search engine we used to find relevant research papers for our project.
- ScienceDirect, arXiv, and ResearchGate are article storage sites that we mostly visited.
- Google Docs: Collaborative writing, sharing project documentation throughout the project.
- GitHub: GitHub will be used for version control, collaborative coding, and sharing the project's codebase.
- CoPilot: Summarize the research papers we found to ensure they are relevant to our project.
- Grammarly: Ensure the clarity and grammar check of our research papers.
- ChatGPT: Requests for complicated technical information, for example:
    - Explain the different pre-trained variants of segFormer by huggingface.
    - Show me an example of the segFormer model written in Python.
    - Show me different tools/libraries for dataset augmentation.
    - Which tools are able to help me find the best hyperparameters for my model.

Solution Tools:

- Programming Language: We will use Python as the primary programming language due to its rich ecosystem of libraries for deep learning.
- Google Earth Engine (GEE) and Google Earth Pro: For accessing and processing satellite imagery and environmental data.
- Aerial and satellite imagery datasets: WHU Building, Inria Aerial Image, LoveDA, and DeepGlobe datasets for the training part.
- Cloud Computing Resources: Training our model with large datasets on cloud-based platforms that provide GPUs, such as Google Colab, which will also be used as a substitute for an IDE, offering an accessible environment.
- Python Libraries: PyTorch Lightning, Optuna, NumPy, Pandas, Matplotlib, Seaborn, Rasterio, Shapely, scikit-learn, OpenCV, TensorBoard, torchmetrics, albumentations, huggingface_hub.

- Hugging Face: We will use Hugging Face Transformers to leverage pre-trained SegFormer models for efficient transfer learning and fine-tuning on urban segmentation tasks. We will also use the huggingface hub for our datasets and models.

Research papers, presentations, and further work will be published on the GitHub repository: https://github.com/erik-pinhasov/ML-Microclimate-Analysis.git

References:

[1] H. Bherwani, A. Singh, and R. Kumar, “Assessment methods of urban microclimate and its parameters: A critical review to take the research from lab to land,” *Urban Climate*, vol. 34, p. 100690, Dec. 2020, doi: 10.1016/j.uclim.2020.100690.

[2] Q. Weng, “Thermal infrared remote sensing for urban climate and environmental studies: Methods, applications, and trends,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 64, no. 4, pp. 335–344, Jul. 2009, doi: 10.1016/j.isprsjprs.2009.03.007.

[3] I. Loshchilov and F. Hutter, “Decoupled Weight Decay Regularization,” arXiv.org. Accessed: Sep. 21, 2024. [Online]. Available: https://arxiv.org/abs/1711.05101

[4] M. Yeung, E. Sala, C.-B. Schönlieb, and L. Rundo, “Unified Focal loss: Generalising Dice and cross entropy-based losses to handle class imbalanced medical image segmentation,” *Computerized Medical Imaging and Graphics*, vol. 95, p. 102026, Jan. 2022, doi: 10.1016/j.compmedimag.2021.102026.

[5] F. Chen *et al.*, “The integrated WRF/urban modelling system: development, evaluation, and applications to urban environmental problems,” *International Journal of Climatology*, vol. 31, no. 2, pp. 273–288, Jan. 2011, doi: 10.1002/joc.2158.

[6] Z. Zhu *et al.*, “Understanding an urbanizing planet: Strategic directions for remote sensing,” *Remote Sensing of Environment*, vol. 228, pp. 164–182, Jul. 2019, doi: 10.1016/j.rse.2019.04.020.

[7] H. Bulkeley and M. M. Betsill, “Revisiting the urban politics of climate change,” *Environmental Politics*, vol. 22, no. 1, pp. 136–154, Feb. 2013, doi: 10.1080/09644016.2013.755797.

[8] M. Koc and A. Acar, “Investigation of urban climates and built environment relations by using machine learning,” *Urban Climate*, vol. 37, p. 100820, May 2021, doi: 10.1016/j.uclim.2021.100820.

[9] R. Hamdi *et al.*, “The State-of-the-Art of Urban Climate Change Modeling and Observations,” *Earth Systems and Environment*, vol. 4, no. 4, pp. 631–646, Nov. 2020, doi: 10.1007/s41748-020-00193-3.

[10] F. Salata, I. Golasi, A. de L. Vollaro, and R. de L. Vollaro, “How high albedo and traditional buildings’ materials and vegetation affect the quality of urban microclimate. A case study,” *Energy and Buildings*, vol. 99, pp. 32–49, Jul. 2015, doi: 10.1016/j.enbuild.2015.04.010.

[11] N. Gorelick, M. Hancher, M. Dixon, S. Ilyushchenko, D. Thau, and R. Moore, “Google Earth Engine: Planetary-scale geospatial analysis for everyone,” *Remote Sensing of Environment*, vol. 202, pp. 18–27, Dec. 2017, doi: 10.1016/j.rse.2017.06.031.

[12] J. Han, A. Chong, J. Lim, S. Ramasamy, N. H. Wong, and F. Biljecki, “Microclimate spatio-temporal prediction using deep learning and land use data,” *Building and Environment*, vol. 253, p. 111358, Apr. 2024, doi: 10.1016/j.buildenv.2024.111358.

[13] X. Hao, L. Liu, R. Yang, L. Yin, L. Zhang, and X. Li, “A Review of Data Augmentation Methods of Remote Sensing Image Target Recognition,” *Remote Sensing*, vol. 15, no. 3, p. 827, Feb. 2023, doi: 10.3390/rs15030827.

[14] S. Minaee, Y. Y. Boykov, F. Porikli, A. J. Plaza, N. Kehtarnavaz, and D. Terzopoulos, “Image Segmentation Using Deep Learning: A Survey,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2021, doi: 10.1109/tpami.2021.3059968.

[15] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, “SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers,” arXiv.org. Accessed: Sep. 16, 2024. [Online]. Available: https://arxiv.org/abs/2105.15203

[16] J. Wang, Z. Zheng, A. Ma, X. Lu, and Y. Zhong, “LoveDA: A Remote Sensing Land-Cover Dataset for Domain Adaptive Semantic Segmentation,” arXiv.org. Accessed: Sep. 19, 2024. [Online]. Available: https://arxiv.org/abs/2110.08733

[17] I. Demir *et al.*, “DeepGlobe 2018: A Challenge to Parse the Earth through Satellite Images,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, IEEE, Jun. 2018. Accessed: Sep. 19, 2024. [Online]. Available: http://dx.doi.org/10.1109/cvprw.2018.00031

[18] E. Maggiori, Y. Tarabalka, G. Charpiat, and P. Alliez, “Can semantic labeling methods generalize to any city? the inria aerial image labeling benchmark,” in *2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, IEEE, Jul. 2017.

Accessed: Sep. 20, 2024. [Online]. Available: http://dx.doi.org/10.1109/igarss.2017.8127684

[19] C. Yin, M. Yuan, Y. Lu, Y. Huang, and Y. Liu, “Effects of urban form on the urban heat island effect based on spatial regression model,” *Science of The Total Environment*, vol. 634, pp. 696–704, Sep. 2018, doi: 10.1016/j.scitotenv.2018.03.350.

[20] L. Kleerekoper, M. van Esch, and T. B. Salcedo, “How to make a city climate-proof, addressing the urban heat island effect,” *Resources, Conservation and Recycling*, vol. 64, pp. 30–38, Jul. 2012, doi: 10.1016/j.resconrec.2011.06.004.