The Rubber Ducks

Volume 1 March 2024

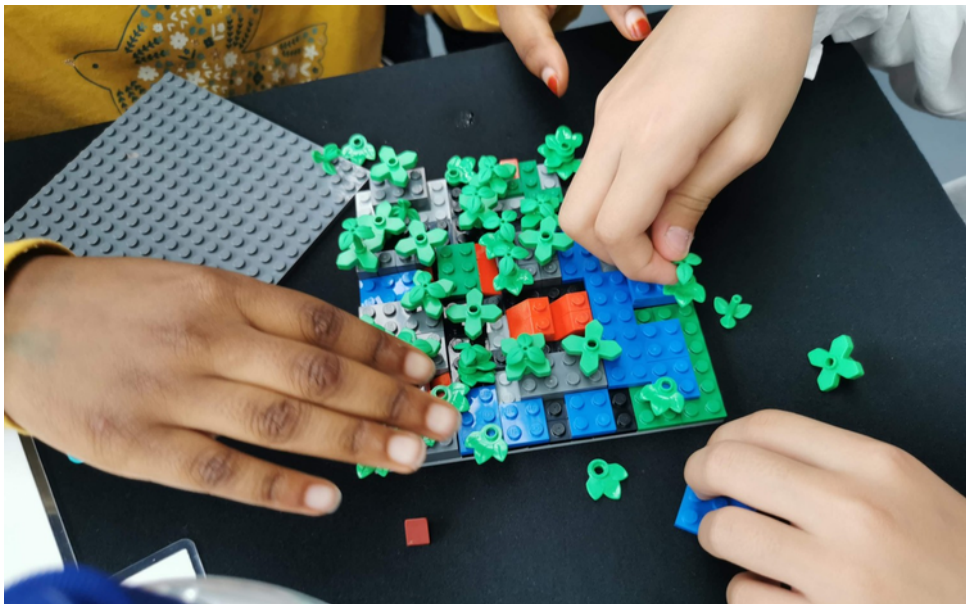
LEGO CITY PLANNER

THE LAB - TI KDG

Written and Developed by
Hannes Coenen, Jef Grupping & Jorre Verelst

Introduction

For several years now, the alarm bells about climate change on our planet have been ringing louder and louder with each passing year. Currently, both animals and vegetation alike have had the misfortune of finding out what the disastrous effects of this rapidly changing climate are and what it could mean for the further development of their species. But us humans are no exception for mother nature either. The architecture of our Belgian cities has not adapted to withstand the increasing heat during the summer, leading to a serious health risk for the people that live in densely populated areas. In these areas, we have observed a phenomenon called the urban heat island effect, where temperatures in the city are on average higher than those in the rural areas. Through the LEGO City Planner, we aspire to raise awareness around the importance of our cities' architecture in the current climatological circumstances, and what risks it could impose on our health.



Urban Heat Island Effect

To dive a little deeper into the issue at hand, we'd like to clarify what the urban heat island effect entails. As mentioned previously, the urban heat island effect is an effect where the average temperature in the city rises above the temperatures in the rural areas around the city. This is due to the lack of vegetation within the city, causing the abundance of pavement and stone to absorb and radiate more heat than natural surfaces. Trees and other kinds of vegetation provide shade and cool areas through evapotranspiration. The shade reduces the amount of heat absorbed by buildings and cools the air, whereas evaporation of the water on the leaves cools the atmosphere. Another cause for this effect is when the warm air near the ground gets trapped by taller buildings, thus increasing the temperature in that area. The effect is most pronounced during calm weather conditions with little wind. In order to mitigate this effect, we need to redesign our cities in a way that allows for more vegetation and enhanced wind flow.

Volume 1 March 2024

Purpose & Workings

The purpose of the LEGO City Planner is to spread awareness surrounding this dangerous effect as we feel it is becoming increasingly more important to address, especially with the rising temperature of the earth due to climate change. People can build their city on a small LEGO grid and place them in a box. After which a picture is taken and the city is converted to a heatmap to indicate the possible flaws in its design. This way, the user can improve on the design by allowing more plants to fit into the city in the high temperature areas.

Hardware

For our project we use a couple hardware components. Since we build our city out of LEGO we're going to need a lot of LEGO bricks and a floorboard to build those on. Next we have a big wooden box in which we can place our LEGO cities. This is done so there are no situational factors that can influence the result of our pictures. Situational factors include but are not limited to lighting, angle of the picture taken, shaking. There is a camera attached to the top lid of the box and there are lights attached to all sides. The last piece of hardware is an SFTP server to store all the images and heatmaps so the visitors can view them at home.

Software

Our project would not work without the implementation of a couple important libraries. In the next paragraphs we will walk through the most important ones in chronological order.

OpenCV

For the computer vision part of our project we opted to use OpenCV. The reason being that it is an open source library with many contributors that is actively being maintained and improved upon. It has clear documentation and is praised by many for its wide variety of functionalities. In our project we have used it to find the bounding rectangle around the LEGO city. Additionally we use it to rotate and cut that rectangle out of the full picture.

NumPy

NumPy is a widely used library for mathematical calculations. We have used this library for complex matrix operations. When we have the processed image from the previous step we can shrink it down to a matrix that has the same width and height as there are pins on top of the LEGO baseplate. Based on the HSV values of the picture we can determine whether a field in the matrix is considered grass, water or impervious.

SciPy

SciPy is a library focused on scientific mathematical calculations. Our use case for this library is the convolutional layer of the data pipeline. The SciPy library provides a function that performs 2 dimensional convolution given a matrix and a kernel. During convolution the kernel acts as a sliding window, by looking through it we get the surrounding of each pixel. A kernel can have different shapes depending on the use case, we needed a kernel with a circular shape. Our kernel consists of 1's for pixels that are entirely within the radius of the circle, 0's for pixels completely outside of the circle and floating points for the edge cases. We use 2D-convolution to distill the composition of the surrounding area from the matrix.

AI-Model

Our application uses an AI-model to predict the temperature. Our model uses a combination of features distilled from the image and parameters set by the user. The resulting matrices from the convolution mentioned above provides the model with the information about the surface material. The rest of the parameters can be set within certain ranges by the user, this ensures that the model will be able to give accurate predictions, extreme values would result in unpredictable results.

The Rubber Ducks

Volume 1 March 2024

Random Forest Regression

The model we used is a Random Forest Regression model, as seen below. The model was trained using meteorological data from the KMI. The data only spans a couple summer months so the model is best used for similar data, for this purpose we restricted the values the user can choose to use. The model was trained using the scikit-learn library and ended up with a RMSE < 1°C.

Seaborn

After we acquire a matrix with the predicted temperature values, we can transform this matrix into a heatmap image. For this job we have used the Seaborn library. This library is specifically designed to make visual representations of raw data. It is one of the easiest libraries to use and it has very clear documentation. Using one line of code we were able to transform the data into a visually pleasing heatmap.

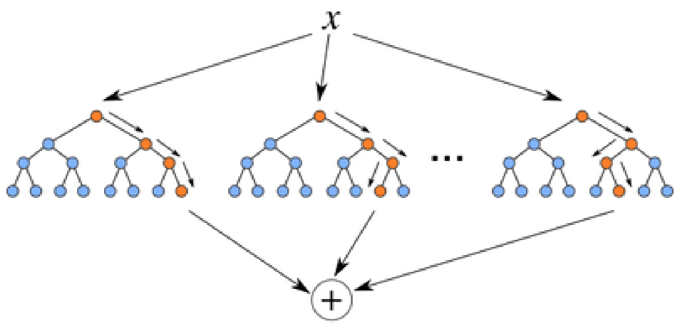


Image: Schematic Representation of a Random Forest Model

TTK Bootstrap

Last but not least, we get to the front-end of the whole application. Tkinter bootstrap was our personal favorite out of all the existing GUI frameworks. Ttkinter bootstrap is a framework built on top of the already built-in library called Tkinter. We have utilized this powerful library to enhance the styling of the GUI framework and make it look more modern. Python is not built for designing pretty user interfaces and is a more functional language, so this library helped us out with the styling aspect of this project quite a lot. Though we were still quite limited with what we could achieve in terms of styling, we have managed to create an appealing user interface that fits the LEGO theme the project was going for.

Conclusion

During this project, we got to know and learn about a variety of useful technologies we've never used before. Despite the obvious upsides to this, we have also known a plethora of headaches and encumbrances along the way. The primary example of this was setting up a connection to the remote server we used to upload our images. As we are not as familiar with the technology related to networking, we had to explore a lot of different solutions before eventually asking our coach, Geert de Paepe, for his expertise to aid us in this process. A great deal of time was lost due to this, but we managed to recover pretty well. Another example was the transformation we performed on all of the images, to make sure images were centered and cut properly. Figuring out how to accomplish this, took a lot of reading documentation and multiple days of experimenting with both the technology and the lights in the box itself. Speaking of the lights, we eventually bought a lamp to try and improve our images, which led to another impediment as the light was often too strong for our AI-model to recognize the proper grid.
All in all, everything ended up coming together just fine. Not only did we gain a lot of knowledge by the end of this project, but we also helped spread awareness about a dangerous, real-world problem.

Special thanks to Andrei Covaci, Geert De Paepe, Dieter De Leeuw, VUB, KMI and KdG