Motivation

Urban heat islands (UHIs) represent a growing danger as they intensify the heat in densely populated areas, exacerbating the health risks and discomfort for residents. As urbanization expands, impervious surfaces like concrete and asphalt absorb and retain more heat, leading to higher temperatures compared to rural areas. This phenomenon not only strains energy resources due to increased air conditioning use but also elevates the risk of heat-related illnesses and mortality, particularly among vulnerable populations. Recognizing the urgent need to address these escalating issues, we have developed a tool aimed at alleviating the effects of UHIs. By incorporating factors such as population density and land use, our tool provides actionable insights to urban planners and policymakers, enabling them to implement strategies that mitigate heat retention and improve overall urban resilience.

Goal

- Predict temperature changes in a city using data on population density, water availability, and vegetation cover.
- Provide users with actionable steps to effectively reduce temperatures in high-impact areas.

Process

1. Data Collection

We used publicly available New York City data as images, which were further modified to be easily read through libraries like pillow and OpenCV. The training was split into overlapping chunks, and unique pieces of data were reflected and rotated into 8 possibilities, leading to a training set of over 20,000 data points.

2. Model Training

We used a Convolutional Neural Network (CNN) due to its effectiveness in handling spatial data, such as images.

The model was trained for 20 epochs with a batch size of 32, using 20% of the training data as a validation set.

The model consists of the following layers:

- Convolutional Layers: Three convolutional layers with increasing filter sizes (32, 64, 128) and a kernel size of (3, 3). We used the ReLU activation function.
- MaxPooling Layers: After each convolutional layer, a max-pooling layer is used to reduce the spatial dimensions of the data.
- Dense Layers: Fully connected layers with 256 neurons and 50% dropout rate to prevent overfitting.

- Output Layer: The final dense layer has a number of neurons equal to the number of output features (576 which is equal to the dimensions of the input 24x24)

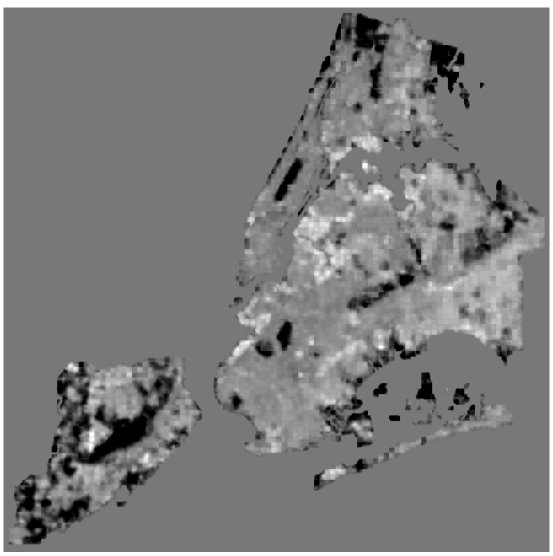
The dimensions of the input and output were specially selected so as to better predict temperatures.

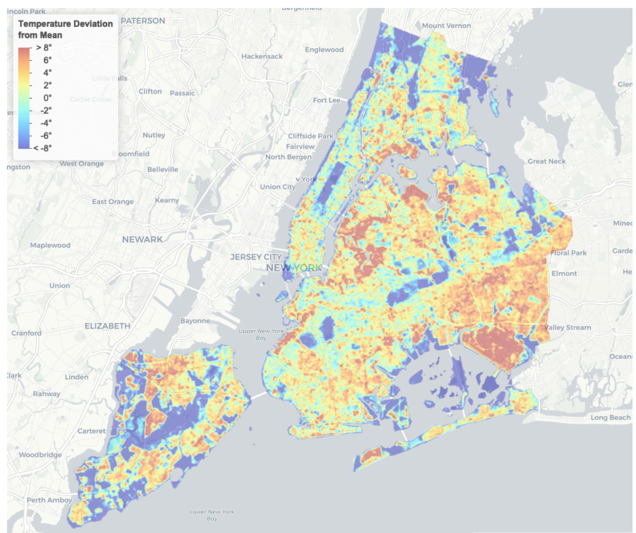
3. Frontend

We used Tkinter, a Python frontend development tool that helped us quickly prototype and connect to our Tensorflow based temperature model.

4. Initial Results

Our initial results for predicting the temperatures of New York City were very promising. The temperature data highly correlated with real, recorded temperature data.





Future Steps

There are several possible future steps, including training on even more data on cities like Shanghai, Boston, or London, improving the user interface, or pushing the use of the application to further positive change in cities.