ART AND MACHINE LEARNING

CMU 2023 SPRING

**PROJECT 4**

**Forecast**

****

Joong Ho Choi, Aditti Ramsisaria, Zun Wang, Kyle Yang

**Concept**

Climate change is one of the most pressing issues facing the world today, and its impacts are already being felt in the form of extreme weather events, sea level rise, and ecosystem disruptions. To help people understand the potential impacts of climate change on natural disasters, this project uses real-time data from sensors that measure air quality, temperature, pressure, altitude, humidity, and eCO2 levels. By analyzing this data with machine learning algorithms, the project can predict how climate change will affect natural disasters such as floods, wildfires, and hurricanes.

To help people visualize these impacts, the project uses neural style transfer to generate images of natural disasters that reflect the predicted impacts of climate change. Neural style transfer is a technique that can combine the style of one image with the content of another image to create a new image. In this project, the style image represents the predicted climate conditions that will impact the natural disaster scene, while the content image represents the original scene before the impact of climate change.

The resulting images provide a visual representation of the potential impacts of climate change on natural disasters, making it easier for people to understand the urgency of the problem. By showing what the world could look like in the future, we aim to inspire people to take action to address climate change.

**Techniques**

1. **Sensor Fusion:**

Our project utilizes sensor readings from two sensors comprising of 7 channels: the BME20 which measures temperature (in C), pressure (hPa), humidity (%) and altitude, and the ENS160 which measures equivalent CO2 concentrations (in ppm), TVOC (total volatile organic compounds in ppb) and AQI (in the range of 1-5). The datasheets for the same are linked in the references section. We sample from these sensors at a rate of 5Hz through a raspberry pi 4 and store the readings in csv files, which are then preprocessed before classification. For preprocessing, we ignore the zero values of the ENS160 (due to sensor warm up time) and normalize the data. We then find the average, min, max, standard deviation, and root-mean-squared values of the samples over a period of 1 second per channel (except altitude) to generate a feature vector of size 30 x 1.

1. **Classification:**

Classifiers were trained to predict the climate labels given sensor data collected from around campus. The classifiers we considered were Random Forest, SVM, and KNN for their ability to handle nonlinear relationships between features and labels. They were trained on sensor data collected in settings that simulate the various climate crises, and of the classifiers we trained, KNN yielded the best testing accuracy, therefore, we used this trained KNN classifier to predict climate labels for various campus locations.

1. **Neural Style Transfer:**

Neural style is an optimization method commonly used in the art domain. It takes in two image inputs, one for style and one for content. In layman’s terms, it learns the artistic style of the ‘style’ image and embellishes the content image with the ‘style’ learned. In order to achieve this effect, NST uses a pre-trained model trained on ImageNet- VGG in TensorFlow. To be specific, it uses the model for feature extraction and separation of content and style representations from an image. The architecture of the model performs the training using two loss terms: Content Loss and Style Loss. Content loss is calculated by measuring the difference between the higher-level intermediate layer feature maps. On the other hand, style loss can be measured by the degree of correlation between the responses from different filters at a level.

Instead of generating the style image based on the natural disaster classification label, we pre-specified images for each class and used them as style images for inference.

**Process**

1. **Identifying hardware:**

We looked at multiple microcontrollers that we could potentially use for this project, including an Arduino Uno, Arduino Mega and the RPi 4. We decided to use the Raspberry Pi 4 Model B due to its ease of use and our familiarity with python, as well as the library support. Our initial plan was to deploy the machine learning model for classification on the pi itself. Being a faster processor and having more memory, the pi is better suited for running machine learning algorithms. It gives us enough GPIO pins and is I2C compatible which makes it suitable for reading from sensors. The Raspberry Pi has built-in Wi-Fi and Ethernet connectivity, which makes it easier to connect to the internet and retrieve real-time data from sensors. This connectivity is also useful for remote monitoring and control of the system.

We then researched which factors can affect weather patterns and found sensors that could help measure them. We used the BME280 to monitor properties like environmental temperature, pressure, humidity and altitude, as well as an air quality sensor (ENS160) which gave us more insight into the changes in climatic conditions. We wanted our sensors to be I2C compatible and not require too much warm up time, as well as cheap and durable. Using these sensors meant that we had to generate our own datasets as the calibration of the sensors used in publicly available datasets would be different from ours.

We also initially planned on using an ArduCam, keeping in line with our plan to create an embedded system that could compute everything on the edge. However, we eventually ended up using a regular phone camera to take photos of the environments around campus.

1. **Dataset generation:**

Based on our research into climatic conditions required for different natural disasters to occur, we compiled the following information. Since it would not be possible to collect data in these environments, we tried to simulate these environments as well as we could within our environment.

We collected data on a warm rainy day outdoors at different parts of campus for our flooding/hurricane case. For example, outside Tepper, next to Wean Hall and near the tennis courts for our flooding scenario. To emulate drought conditions, we collected data in a dry room with an incense stick burning. To collect air pollution data, we collected data close to Forbes Ave on a busy afternoon. For our blizzard case, we collected data inside of a fridge. We generated close to 10k samples per class.

As the data was not gaussian, we normalized the data instead of standardizing it, and after trial and error found that the best results were obtained when we used a feature vector consisting of the min, max, avg, rms and std dev of the readings over a second. We think this is because the raw sensor readings are not very reliable due to the random jumps that occur because of noise.

| **Label** | **Temperature** | **Pressure** | **Humidity** | **CO2** | **TVOC** | **AQI** |
| --- | --- | --- | --- | --- | --- | --- |
| Flooding | warm | low | high | decent | decent | decent |
| Drought | high | high | extremely low | high | high | bad |
| Air pollution | high | low | high | high | high | bad |
| Hurricanes / cyclones / typhoons | high | low | high | high | high | bad |
| Blizzard | extremely low | low | high | decent | decent | decent |

To simplify our solution and have a better dataset size, we combined our collected samples into three categories: ambient/humid, dry, and cold.

1. **Prediction:**
   1. **Clustering**

KNN Clustering was chosen as our classifier to predict climate labels of various campus locations due to its high prediction accuracy. We decided to give clustering a try after not yielding ideal results from classification methods because it might be able to detect underlying patterns in the data that classification failed to notice and use it to more accurately predict the climate label. Fortunately, our thinking was correct and KNN was able to yield a high prediction accuracy of ~85%.

* 1. **Classification**

We first tried using different classifiers like Random Forest and SVM to predict climate labels due to their capability of handling nonlinearity in data. Unfortunately, both Random Forest and SVM failed to yield ideal results. To improve their performance, we tried tuning their hyperparameters like number of trees and kernel function but no significant changes were seen. We hypothesize this might be due to the training data containing more of one climate label than the others. After adjusting our training data, we saw a notable improvement in performance, however, since the prediction accuracy was still lower than that of the KNN clustering method we tried afterwards, we decided not to move forward with these.

1. **Image generation:**

We decided to use neural style transfer as our image generation algorithm, because NST is the most straightforward way of applying our classifications to our campus images and giving them an artistic transformation based on their respective climate readings. To do so, we needed two sets of images: style images and content images. We used our campus images as our content to perform style transfer on. For our style images, we found artworks that we found representative of what we would expect from each classification. Then, for each image, we use the classifier to categorize the climates of the content images. Using these categories, we determined which style image to use for neural style transfer.

1. **Alternative approaches:**

We also considered many alternative approaches to this project. One of these approaches was to generate textual information based on the sensor readings which could then be fed into a text-to-image generator (like runway ML) to modify the image. However, we could not find much literature to help us get started on the text generation for this approach. Another approach consisted of training a model from scratch that took in sensor readings and an image as input, and generated the resulting image as an output. Since finding or collecting a dataset of such image pairs would be near impossible, we considered using existing image modification tools to generate the output pair corresponding to each image. However, due to time constraints and the scope of the project, we were unable to explore this approach as it would require a lot of research into ML methods for generation as well as manual dataset collection. Both of these approaches would be ideal for future exploration in this space.

**Reflection**

One limitation of collecting our own data over a short time period is that we could only collect a small dataset of similar readings, because climate does not vary very drastically in one location over a short period of time. Because of this limitation in dataset variation, we had issues with finding example locations for each class. Although we generated datasets of differing environments by simulating them, and were able to generate a model that could differentiate between them, we were limited at the time of inference by the climatic conditions of Pittsburgh. We tried to account for this limitation by going to places that we thought would have differing AQI and environmental variables, but results remained pretty consistent throughout campus. Ideally, we would like to collect data from outside campus and over a longer period of time so we could observe larger differences in all these variables, or run inference tests in differing environments.

Overall, we see this project as a step in the right direction to building a fully automated system that could be hosted on a Pi and could potentially use TinyML to deploy the classification model on the Pi itself, or use cloud computing in conjunction to deploy the classification and NST model to run in real-time, albeit with a small lag.

**RESULT**

****

Our result has the images of campus on the left and the results of using them for neural style transfer on the right. The top left and bottom right were classified as dry, the top right was classified as cold, the bottom left was classified as ambient. We framed them in a movie strip to represent the transition from their current state to their artistic rendition taking into account their climate.

**CODE**

<https://github.com/JC-78/artML>

**REFERENCES**

BME280 Sensor: <https://www.adafruit.com/product/2652>

ENS160 Sensor: <https://www.adafruit.com/product/5606>