Predicting Climate using Aerial Imagery

ABSTRACT

A well-known issue throughout the world is Climate Change. To combat this problem, aerial imagery can be classified into different Koeppen-Geiger climate types, which can be compared to themselves in future years, showing any type of evidence that the climate is changing. We approach the problem by using Convolutional Neural Networks and the Haversine formula, which classify images to their respective climate type. The model predicted the climates with reasonable accuracy, fluctuating around 50% validation accuracy.

Keywords - Koeppen-Geiger Climates, Image Classification, Convolutional Neural Networks, Keras

Introduction

Climate Change, often associated with global warming, occurs from the emission of greenhouse gasses. Effects of climate change include: rising temperatures, sea levels, and precipitation, resulting in ice glaciers melting. A potential way to detect climate change would include classifying images by their climate type and comparing it to itself in future years.

One way to explore climate change and the climate imagery is to pull the nearest aerial imagery data from a satellite, given the latitude and longitude. We can then classify this data using the Koeppen-Geiger climate classification system. The climate classification system works by classifying imagery anywhere from Tropical regions to Polar regions, by assigning it different Koeppen-Geiger types, which is useful for simplicity and detection by climate observers.

The Koeppen-Geiger climate classification system is based on temperature, precipitation, and other factors. It is divided into different categories, having a first letter, middle letter, and a last letter. A represents tropical, B is dry, C being temperate, D is continental, and E is polar climates. The other letters build upon these ideas in more detail. For climate A, monsoons are also a part of the category, using annual precipitation to calculate if this category applies to an input.

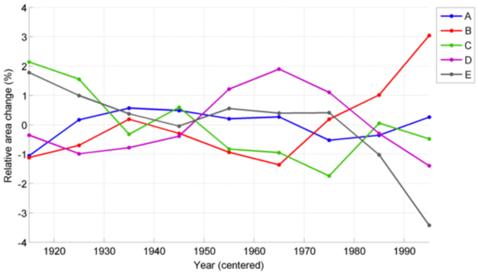


Figure 1. Area change of major Koeppen types [2].

Considering Figure 1, we can see that the lowest covering climate type in 1901 is dry climate. E is the second highest climate following temperate climate. Over the course of the period, climate E is steadily decreasing as well as climate C, dropping significantly in 1975. The dry climate starts increasing; decreases to its lowest point in 1965, then steeply increases to being the highest covering climate type in 2010. This is significantly concerning, since it shows that polar climates are decreasing at a steady rate, proving that ice is melting. The dry climate is also seeming to increase, showing evidence of overall global warming and climate change [2].

In this work, we propose a method to classify areas of land by Koeppen type using convolutional neural networks on aerial imagery. The neural network used in this program takes in an image of a random surface area, outputting the climate it has identified.

Related Work

Although satellite imagery is fascinating, another approach was using landscape imagery. Johnson et al. used three different models to classify landscape imagery. The dataset used were images by Flickr, labeling them with tags and managed to pull 320,000 images. The first method used was having a logistic regression model classify the images, giving them around 10% accuracy for the testing data. As shown in Figure 6, the prediction will lie somewhere between 0 and 1, meaning that it will classify a model with a certain probability.

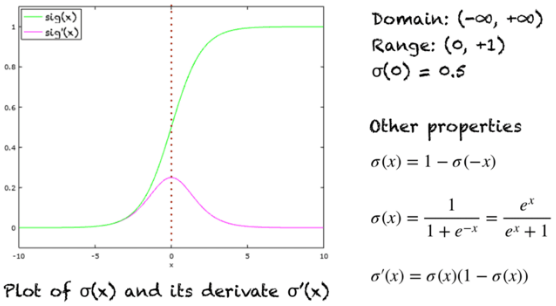


Figure 2. Logistic Sigmoid Function [7].

Although this system won't be using logistic regression for the model, it is still worth mentioning, as the model does involve a confidence probability value. With the second model, the Support Vector Machine model, the students were predicting with 15% accuracy, and finally with the CNN, they were able to predict with 31% accuracy.

The CNN model predicted twice as accurate as the other models. Even with some errors, the model predicted the images closely to the actual value. This is what we also experienced, as shown later in the confusion matrix. Therefore, the CNN model will mostly be used throughout this project to have the most accurate predictions [3].

Since latitude and longitude are important in determining the climate, we believe that it would be best to fetch the nearest images of given coordinates. Additionally, there is a relationship between latitude and temperature; temperature increases as it approaches the equator, and decreases as it approaches the poles.

Waters et al. were able to classify urban tree species with a CNN model. They were able to use image augmentation to improve their results, leading to around 50% accuracy. The model includes training on a dataset of six trees.

This study was important, since it is strongly related to this project, but also provides insights to CO_2 levels and other classification techniques that are mentioned throughout this article [10].

Dataset

In order to identify certain climates, I decided to train the dataset on a set of classified images. I first pulled data from Sentinel-2 satellite imagery and I stored this dataset under the name of Sentinel-2 Satellite Imagery Sample Set. In order to find which climate, type a certain location is given in, we have to look at the Koeppen-Geiger climate classification system.

In order to retrieve data for a specific climate using latitude and longitude I have referred to Hans Chen [2]. I am using the Koeppen climates from 1901 to 2010, which compromises a longitude, latitude, and the climate type. That being said, to find the closest climate type for that climate, given the coordinates, we would have to calculate the distance in order to find which climate is the most reasonable, given the latitude and longitude.

Although the data is compromised from values between 1901 and 2010, I think this is highly appropriate for my research. As the data is older, we can use this to compare newer data to this old data to understand the parts of the imagery/area that are changing. Additionally, since the data is fairly recent, the climate shouldn't have changed all too much, which makes our measurements fairly accurate, despite the time difference.

The dataset showcases a bit more than 1,000 images with more than 800 of them belonging to a training class, and 200 images being used for validation and testing respectively.

Methods

In order to create this program, we would need the closest aerial imagery corresponding to the respective latitude and longitude. We can then use this data to find its climate given previous data. We would also need to measure the distance of a sphere between two points to match which area is nearest to that given coordinate.

$$\eta = \sin\frac{\Delta\phi^2}{2} + \cos\phi\cos\phi_2\sin\frac{\Delta\lambda^2}{2}$$

(ϕ representing latitudes, λ representing longitudes)

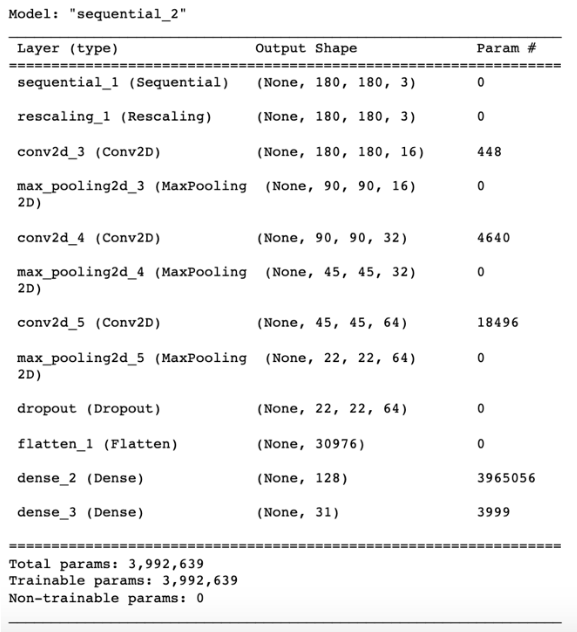
Equation 1: To measure the distance on a sphere I used the haversine formula (shown above).

$$\delta = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Equation 2: In a two-dimensional world, we are most familiar with the formula above.

The distance formula can be rewritten from Pythagoras's Theorem. While in the two-dimensional world it may work, since we live on a sphere, it would not be sufficient for our needs. Using the haversine formula we can find an accurate distance for the latitude and longitude on the globe. Imagine the following example: when you look at a short distance on the earth, you would say it is mostly flat. This is great for the distance formula. Now say we zoom out to around 20km. The shape of it is more spherical, requiring spherical geometry. If we would use the distance formula we would receive large errors [4].

The type of system we will be using to train our image classifier will be a Convolutional Neural Network. This means that it can add weights to images and learn how to differentiate between them. Interestingly, CNN's can be trained by feeding it more complex data and depending on filters to perform better over time. Therefore, CNN's can transform complex images to less sophisticated images, while keeping all the necessary information [8]. See Figure 3 for a complete summary of the CNN model being used. The model features 12 layers, with the input shape being (180, 180, 3) representing the image height, width, and RGB values respectively. The model features a cross entropy loss function, which quantizes mistakes made by the network, so that the network weights can then be adjusted for better performance [6]. Then, the model classifies each image to a total of 31 climate types.



```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 sequential_1 (Sequential)   (None, 180, 180, 3)       0

 rescaling_1 (Rescaling)     (None, 180, 180, 3)       0

 conv2d_3 (Conv2D)           (None, 180, 180, 16)      448

 max_pooling2d_3 (MaxPooling  (None, 90, 90, 16)       0
 2D)

 conv2d_4 (Conv2D)           (None, 90, 90, 32)        4640

 max_pooling2d_4 (MaxPooling  (None, 45, 45, 32)       0
 2D)

 conv2d_5 (Conv2D)           (None, 45, 45, 64)        18496

 max_pooling2d_5 (MaxPooling  (None, 22, 22, 64)       0
 2D)

 dropout (Dropout)           (None, 22, 22, 64)        0

 flatten_1 (Flatten)         (None, 30976)             0

 dense_2 (Dense)             (None, 128)               3965056

 dense_3 (Dense)             (None, 31)                3999

=================================================================
Total params: 3,992,639
Trainable params: 3,992,639
Non-trainable params: 0
_________________________________________________________________
```

Figure 3. CNN Model Summary.

Figure 4 illustrates 9 example images classified by the system.

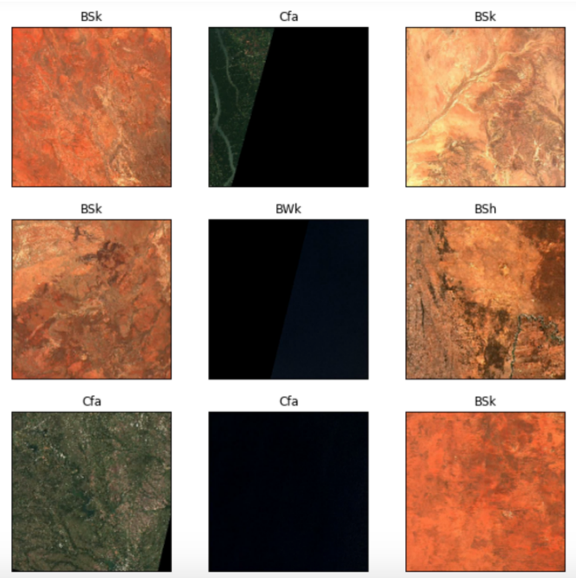


Figure 4. Koeppen Climate Types with Respective Images.

This system is trained on images from 1901-2010 [2] and was implemented in Keras from Tensorflow. The dataset is used due to its public ability and due to its reliability in finding patterns. For example, we can see long term changes in climates for some areas, since it is active for a long range of years.

With some readjustments to the neural network including rescaling and dropout, we found some interesting results.

Results

Our analysis of the results includes a decent accuracy, but more importantly to classify climates from a range of climates. For the first-time results, it did not perform accurately. As shown in Figure 5, the validation accuracy barely reached 45%, while the training accuracy was barely soaring above 50%. Compared to other models, this model seems to be on par with them. For example, the urban tree classifier had a range between 40% and 60%, while the landscape imagery only seemed to have 31%. This being stated, this model is mostly consistent with other models in the same category of classification. The below table summarizes these results:

Model	Sentinel-2 Satellite Imagery Sample Set	Performance
Keras Sequential	>1,000 images	~50% validation accuracy

Table 1. Dataset Summary.

Sadly, it was difficult to download imagery, since the server would shut down after a few cycles. However, we were able to pull enough images to show contrasts between the accuracies and number of images pulled, summarized in the table below. [9]

	1,000	1,250	1,500	1,750
Training	57%	85%	87%	88%
Validation	48%	50%	53%	55%

Table 2. Performance Summary.

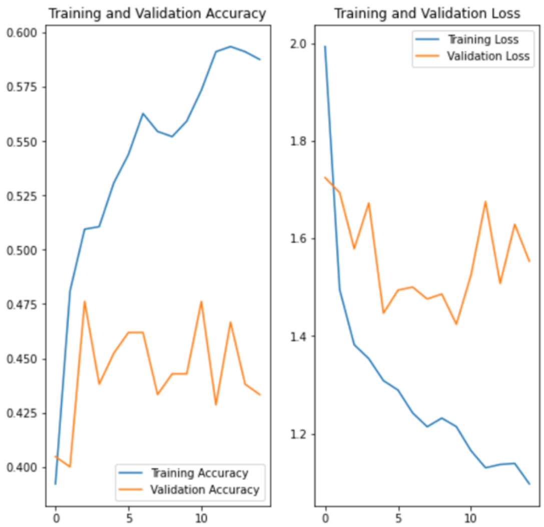


Figure 5. Graphs of Training and Validation Accuracy and Loss.

This table shows that we are experiencing an increase in the percentages; however, since we were only able to pull a small number of images at a time, we only see small increases. We observe that adding more data improves model performance, thus recommending an expanded dataset for future work to improve accuracy.

When testing on imagery, with an 80-10-10 percent split between the train, test, and validation data respectively, we have an average of 45% accuracy, which is perfectly reasonable as supported by the graphs. In order to try to improve accuracy, we used two techniques: data augmentation and dropout techniques.

By using data augmentation, we can flip the image horizontally and vertically and rotate it by a factor of 0.1. Then we can increase our zoom percentage by the same amount to obtain different results. This is shown by the imagery in Figure 6.

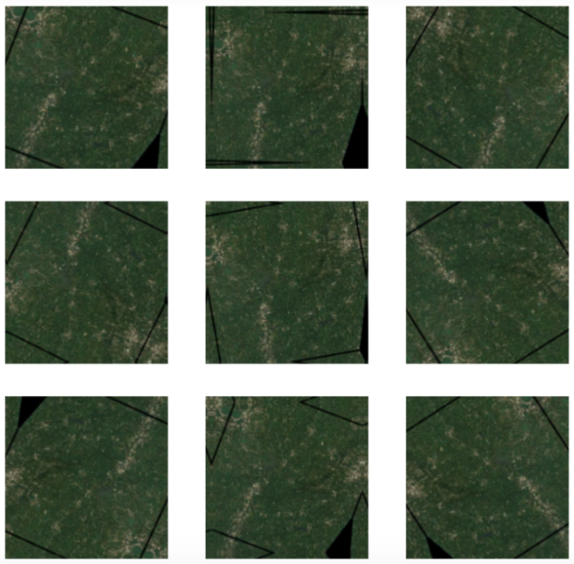


Figure 6. Data Augmented Imagery.

Our second technique introduced the idea of dropout. This is designed to prevent overfitting, by setting random input units to be ignored during the training phase of the model [1].

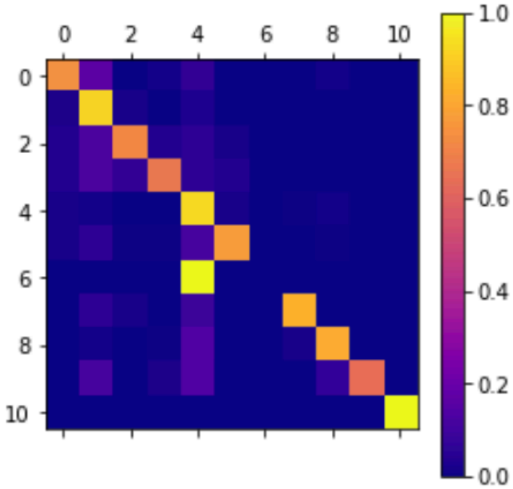


Figure 7. Confusion Matrix with Climate Types.

In Figure 7 we can see that the predictions are somewhat accurate, which is expected, since it has around 50% accuracy. It also only has 10 classes, due to some data not being accessible when pulling from the satellite imagery.

Most classes are performing relatively well, with the only discrepancy being that of class 6, which is the Csb climate type. This climate type had only a few images, hence resulting in the overfitting of a certain class; however, as shown by the other values, if we expand our dataset, we should find rather constant values. [10]

Conclusion

Although we have a climate classifier using the methods mentioned previously, future work in this area includes filtering ocean imagery. Although the ocean might have a climate, it isn't helpful to pass a black image of it with different types of climates.

In the future, I am planning on including potential mappings categorized by different colors on the areas being affected the most by climate change. I also would like to add CO_2 levels in specified areas to measure how these areas are affected by global warming due to CO_2. This could predict potential wildfires that could occur and the response time when predicting for them would increase drastically, helping save the environment. It could also be useful for other purposes, such as: habitats for birds' identification or where certain trees could be grown.

Code

Github: https://github.com/emirdur/PredictingClimateUsingAerialImagery

Acknowledgements

I would like to thank Ross Greer at UC San Diego for teaching me the process of researching in detail.

References

[1] Brownlee, J. (2016, June 20). *Dropout Regularization in Deep Learning Models With Keras*. Machine Learning Mastery. Retrieved February 7, 2022, from https://machinelearningmastery.com/dropout-regularization-deep-learning-models-keras/

[2] Chen, D. and H. W. Chen (2013): Using the Köppen classification to quantify climate variation and change: An example for 1901–2010. *Environmental Development*, 6, 69-79, doi:10.1016/j.envdev.2013.03.007.

[3] Johnson, D., Ngo, T., & Fernández, A. (2019, June 11). *Climate Classification Using Landscape Images*. Retrieved February 7, 2022, from http://cs229.stanford.edu/proj2019spr/report/23.pdf

[4] Kettle, S. (2017, October 5). *Distance on a sphere: The Haversine Formula*. ESRI Community. Retrieved February 7, 2022, from https://community.esri.com/t5/coordinate-reference-systems-blog/distance-on-a-sphere-the-haversine-formula/ba-p/902128

[5] Narkhede, S. (2018, May 9). *Understanding Confusion Matrix*. Towards Data Science. Retrieved August 29, 2022, from https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62

[6] Parmar, R. (2018, September 2). *Common Loss Functions in Machine Learning*. Towards Data Science. Retrieved February 7, 2022, from https://towardsdatascience.com/common-loss-functions-in-machine-learning-46af0ffc4d23

[7] Saeed, M. (2021, August 25). *A Gentle Introduction To Sigmoid Function*. Machine Learning Mastery. Retrieved February 7, 2022, from https://machinelearningmastery.com/a-gentle-introduction-to-sigmoid-function/

[8] Saha, S. (2018, December 15). *A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way*. Towards Data Science. Retrieved February 7, 2022, from https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

[9] Tensorflow. (2022, August 12). *Image classification*. Retrieved August 29, 2022, from https://www.tensorflow.org/tutorials/images/classification

[10] Waters, E., Oghaz, M. M., & Saheer, L. B. (2021, July 7). *Urban Tree Species Classification Using Aerial Imagery*. Retrieved March 4, 2022, from https://arxiv.org/pdf/2107.03182.pdf