Environment-Agnostic Wildfire Detection with Satellite Imagery

Jennifer Lin and Yanbing Zhu
Stanford University
jennlinc, yanbingz@stanford.edu

Abstract

Wildfires are a major environmental issue, causing economical and ecological damage while posing a safety risk factor. $224x224$ pixel satellite images from Landsat 8 with and without active fires are used as inputs to various ConvNet architectures. Existing deep nets (Inception, ResNet50, VGG) with pretrained ImageNet weights resulted in a high variance, while a simple, custom architecture with 6 convolutional layers and 3 fully connected layers was able to achieve 87 percent training accuracy and 86 percent test accuracy.

1 Introduction

Various methods exist to detect and predict the cause, burn area, spread rate etc. of a wildfire. These include using traditional GIS systems overlapped with related features including as altitude, vegetation type, highways or other natural boundaries that have known correlations [1] and machine learning with structured weather and geographical data [2, 3, 4]. While these methods provide insight for mapping fires descriptors to geological features, they rely on expensive terrestrial data gathering.

High resolution satellite data from Landsat 8 is used as input for ConvNet architectures to detect fires and improve upon currently used equation based models [5, 6]. The current active fire detection model for Landsat uses a two-channel fixed-threshold plus contextual approach that takes the difference between the data recorded in two different wavelength ranges ("bands" or "channels") with additional parameters and mathematical formulations. A fire is labeled if the difference is above a numerical threshold. The goal is to use deep learning to extend this model beyond an equation, particular due to the intrinsic complexity of satellite images (difference in cloud coverage, terrain, land reflectance, time of day the image is taken). Numerical calculation requires calibrating images with regarding to these parameters

The input to our algorithm is a $224x224$ image that is either RBG, what we term IR (which consist of bands $7, 5,$ and 1 of the satellite), or both RBG and IR images for a total of 6 wavelength channels. The image corresponds to a geometric footprint of roughly 7 km by 7 km. We then use different convolutional network designs to output a prediction of whether the image has a fire or not. With a custom Convolutional Network architecture, our model can achieve 86 % accuracy onon he held-out test set.

2 Related work

Current forest fire monitoring and detection include the National Fire Danger Rating System (NFDRS), which presents fire ratings in maps using the Wildland Fire Assessment System (WFAS) that takes in weather information from 1,800 fire weather stations across the U.S along with on-site fuel and topography data. Another index is the forest Fire Weather Index (FWI), developed by Canada in the 1970's and based on meteorological information i.e. as temperature, relative humidity, wind speed, and precipitation [7]. Statistical models also exist that correlate fire risk with historical weather and fire data including the Fire Family Plus program from the US Forest Service.

Artificial intelligence and machine learning have been applied to fire prediction, but largely used structured data or is limited to specific geographic regions. Cortez et al. used a data mining approach with meteorological data such temperature, rain, relative humidity and wind speed with Support Vector Machines (SVM) and random forests approaches for the Montesinho natural park in Portugal [8]. Stojanova et al. combined weather modeling information in conjunction with vegetation modeling and GIS data, but focused on Slovenia [12]. Sakr et al. used SVMs and focused on Lebanon [2]. The first instance of using AI to classify fires was in 1996 when Vega-Garcia et al. adopted Neural Networks (NN) to predict human caused wildfire occurrence using a 314 fire and no-fire dataset for 1986-1990 [9]. They used logistic regression analysis and accurately predicted 85 percent of the no-fire observations and 78 percent of the fire observations.

3 Dataset and Features

Satellite images from the Operational Land Imager (OLI) on the Landsat 8 satellite launched 2013 were chosen for its high 30 m spatial resolution. The footprint, or individual scene, size is 185 km by 180 km.

Data retrieval is processed in 11 bands separated by wavelength ranging from blue at 0.43 microns to thermal infrared at 12.51 microns. Bands 4, 3, and 2 corresponding to RGB and band 7, 5, and 1, shown in Table 1, were selected since RGB bands are combinable in a preprocessed image that is more similar to traditional images/easier to understand for human, and band 7 is useful for the high emissivity from fires, whereas 5 also provides supplementary near IR data and band 1 features the opposite coastal blue. In addition, we use cloud coverage field in the QA band to filter out images with more than 30 percent cloud coverage to ensure remotely sensed image actually contains a fire visible to the model.

The coordinate of fires for labeling purposes is taken directly from Landsat active fire dataset, which was identified using their mathematical detection algorithms. This was selected over alternative sources since it comes directly from Landsat images and ensures an image at the time of the fire is available. The images were preprocessed by mapping the fire lat/lon coordinates to the images and cropped into 224x224 pixel blocks around the fire. For each set of active fire coordinates, a set of random coordinates was chosen from the same image for the negative labels after confirming the 224x224 pixel block didn't overlap with any active fire image blocks. The raw image with labeled fires and an example of a cropped IR image is shown in Figure 1 and Figure 2. The images had 50% fires and 50 % no fires, and were split into a train/dev/test set of 39642/4955/4957. The images are collected from all wildfire fire instances that occurred in the continental United States from 2016-2017.

To augment the dataset, the images were rotated and flipped, and additional noise was added, but this was only done with the basic CNN since we found it didn't yield large improvements. We suspect that this is because the cropping method already introduce randomness to the data set. Hence, additional noise will not help generalization

Band	Wavelength (μm)
1	0.435 - 0.451
2	0.452 - 0.512
3	0.533 - 0.590
4	0.636 - 0.673
5	0.851 - 0.879
7	2.107 - 2.294

Table 1: Landsat 8 OLI Band Wavelengths

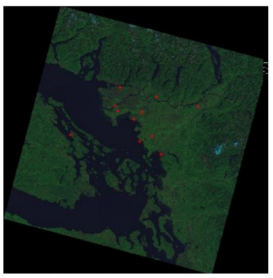

Figure 1: Landsat8 IR image with fires labeled as red points

Figure 2: Sample of a cropped image from Figure 1

4 Methods

A basic CNN model (Arch 1 in Table 2 with 4 convolutional layers was used as a preliminary run to ensure the data preprocessing and input stream was functional. 3 different image inputs were tested: RGB (bands 4, 3, 2), what we term "IR" which consists of the IR band and two shorter wavelengths (bands 7, 5, 1), and a 6 channel version with both RGB and IR. The IR image input yielded better performance than RGB, but using 6 channels further improved the model as shown in Figure 4. Hence, the 6 channel version was used for future models and the IR input was used for models with pretrained weights. While we recognize the pretrained weights are designated for RGB images, we ordered the IR image inputs from longest wavelength to shortest wavelength as well. Since several we trained several deeper layers using the pretrained weights only as a starting point, we figured this was not a significant issue.

All models are followed by a single output unit without sigmoid activation and optimized with regard to binary entropy loss

$$Loss(y, \hat{y}) = -ylog(\hat{y}) - (1 - y)(1 - log\hat{y})$$

Our implementation for the project can be found in the following github repository: `https://github.com/jenniferlin0902/wildfire_prediction/tree/final_submission`

VGG

We used a pretrained model based off an open source project [13], which was a Tensorflow implementation with weight from the original VGG net authors. The first three blocks of convolutional layers are held fixed with the initial weights, while the rest of the 2 blocks are trainable variables initialized with pre-trained weights. A single fully connected layer with 256 units is added to the end of the convolutional blocks with Xaviar initialization. Note that the fully connected units does not follow the set up in the original VGG16 net due to the limitation in computation power.

InceptionV3

A pre-trained model InceptionV3 (GoogLeNet) using ImageNet weights was adapted using Keras [11]. A single fully connected 1024 neuron layer (ReLu) added in the end. This was trained with top two inception blocks unfrozen for 20 epochs with batch size 128 using Adam optimization. RMSProp offered similar results.

ResNet50

Similar to the Inception model, a pretrained ImageNet weight version from Keras was adapted with the same 2 additional fully connected layers. The top convolutional block (1 out of 5 possible) was trained for 15 epochs with batch size 128 and Adam optimization.

Custom ConvNet

VGG, InceptionV3, and ResNet50 were not pursued further using a thorough hyperparameter search and optimization since the architecture is likely too complicated for fire identification task given inital results. Fully training the models with all weights unfrozen were considered, but not pursued to due to time constraints and the large computational cost.

We instead iterated through more complex architectures building off the basic CNN model. Table 2 shows the different architectures tested. Note that each convolutional block is separated by a BatchNorm, Relu activation, and a 3x3 Max pooling layer.

5 Experiments/Results/Discussion

Model Selection

Table 3 shows the training and test accuracy for the models tested. Note that all pre-trained model overfit the training data. This largely result from the fact these models are trained for multi-classes classification tasks and the expected input features are much richer than the satellite inputs. Although

Table 2: Custom Convnet Architecture

Model	block 1	block 2	block 3	block 4				
Arch1	6 * 3x3	12 * 3x3	24 * 3x3	48 * 3x3				
Arch2	64 * 3x3	32 * 3x3	24 * 3x3	12 * 3x3	Fc 256	Fc 128		
Arch3	64 * 3x3	32 * 3x3	24 * 3x3	12 * 3x3	12 * 3x3	Fc 256	Fc 128	
Arch3	64 * 3x3	32 * 3x3	24 * 3x3	12 * 3x3	12 * 3x3	Fc 256	Fc 128	
Arch4	64 * 3x3, 64 * 3x3	32 *3x3	24*3x3	12 * 3x3	12 * 3x3	Fc 256	Fc 128	
Arch5	64 * 3x3, 64 * 3x3	32 *3x3	24*3x3	12 * 3x3	12 * 3x3	Fc 128	Fc 128	
Arch6	64 * 3x3, 64 * 3x3	128 *3x3	24*3x3	256 * 3x3	512 * 3x3	Fc 2048	Fc 1024	FC 128

previous work using deep learning on satellite images show pre-trained models such as VGG16 work well as a feature extractor for high resolution satellite images [7], the same result may no apply to our case. The images we are focusing on include all 6 channels. Even for the 3 channel IR images, the features learned for RGB inputs might not necessarily apply to IR features. Since simpler custom CNN yielded better test accuracies, we focused on refining the custom network.

Figure 3 shows the training and evaluation curve for the ConvNet Arch2 - Arch6. All models can reach training accuracy greater than 85 % within 8000 steps. However, deeper networks, i.e Arch4-Arch6 converge faster than simpler ones. Both Arch 4 and Arch 5 yield a 85 86% evaluation accuracy. The variation level between the 3 models are within noise. However, since Arch5 has less trainable parameters than Arch4, we select Arch 5 as our model to perform further analysis. Using a random grid search on learning rate and batch size hyperparameters, we found a batch size of 100 and a learning rate of 0.005 was optimal.

To visualize what the neural network is learning, we extract out activations from the first and third convolutional layers. We can first verify that our models is tuned to pick up landscape features such as edges between different vegetation types and blocks of vegetation or water body. However, higher level feature such as fire is harder to distinguish in the layers.

Table 3: Training and testing accuracy for different models

Model	Training	Testing
VGG	088	0.64
ResNet	098	0.69
Inception	099	0.67
Arch1	0.85	0.85
Arch5	0.87	0.86

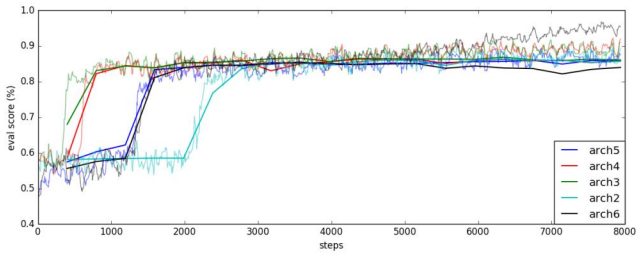

Figure 3: Training curve and evaluation curve for custom ConvNets.

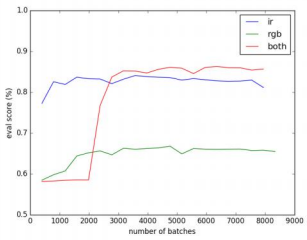

Figure 4: Training curve for ir, rgb and both images on simple conv net

Error analysis

A sample of 900 test images from the Arch5 custom architecture yielded a [420 34 98 348] confusion matrix. We found that false positive images largely had high cloud cover, buildings, or mountain

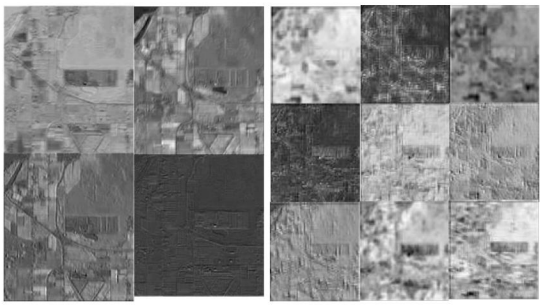

Figure 5: Activation for first (left) and third (right) convolution layer in Arch5

regions with dark crevasses and might have been misidentified as fire regions. False negatives contained poor resolution and more night-time images.

As a note, the Landsat active fire algorithm has a error of 4.8 percent in the United States with classification hand verified by fire experts, so at best our algorithm can perform is 95 percent accuracy. In the future, this can remedied by pulling satellite images from verified fire data by lat/lon and taking the satellite footprints either from Landsat or another source with the closest timestamp.

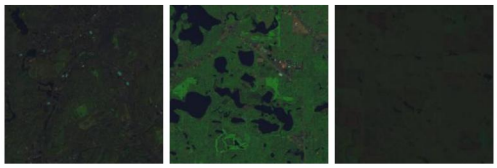

Figure 6: Sample of false negative test image results

Figure 7: Samples of false positive test image results

6 Conclusion & Future Work

The benefit of satellite images for fire detection is it's easily extended. Meteorological information (wind speed, rainfall etc.) can also be added in conjunction for prediction. This was at the present omitted since existing fire prediction research is heavily based on these weather conditions, and we wanted to focus on ConvNet performances rather than additional data processing.

We demonstrated a fairly simple ConvNet architecture, not as deep as those used for object detection/ImageNet, can be used to predict whether a fire is present in a 224x224 pixel satellite image with a resolution of 30 m per pixel with up to 86 percent accuracy. Given more time and computational resources, we would have like to try training deeper architectures with 6 channel inputs from scratch or possibly establish a series of pre-trained weights useful for all the bands satellites are capable of detecting beyond standard RGB.

As another future application, YOLO/object detection can be used to find bounding boxes, particularly for applications related to fire geometries including burn area prediction. This might be difficult since fire regions have drastically different shapes that is heavily time dependent. This particular method wasn't feasible for our current dataset since Landsat is on a 16 day cycle, but if further developed, it can be used in conjunction with MODIS satellite images which has a 1-2 day temporal resolution.

Another alternative extension is combining this method with U-net image segmentation architectures - i.e. high fire risk can be prioritized near densely populated areas with buildings or highway and other natural boundaries and can be considered as bounding zones for fire spread.

7 Contributions

Jennifer worked on the script for data download, setting up tensorflow project including dividing the data, preforming architecture search on custom conv net structure VGG. Yanbing worked on cropping, labeling, and combining the RGB with IR bands for NN input, the image normalization for the different bands and model selections for ResNet and Inception. Both worked on downloading the data, running models and error analysis.

References

[1] Jaiswal, Rajeev K, et al. "Forest fire risk zone mapping from satellite imagery and GIS." *International journal of applied earth observation and geoinformation* 4.1 (2002)

[2] Sakr, George E, et al. "Artificial intelligence for forest fire prediction." *IEEE/ASME International Conference on Advanced Intelligent Mechatronics* (2010)

[3] Romero, Adriana, Carlo Gatta, and Gustau Camps Valls. "Unsupervised Deep Feature Extraction for Remote Sensing Image Classification." *IEEE Transactions on Geoscience and Remote Sensing* 54.3 (2016)

[4] Albert, A, et al. "Using Convolutional Networks and Satellite Imagery to Identify Patterns in Urban Environments at a Large Scale." *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2017)

[5] Schroeder, Wilfrid, et al. "Active fire detection using Landsat-8/OLI data." *Remote sensing of environment* (2016)

[6] Roy, David P, et al. "Landsat-8: Science and product vision for terrestrial global change research." *Remote sensing of environment* (2014)

[7] Stocks, BJ et. al. Canadian Forest Fire Danger Rating System: An Overview The Forestry Chronicle (1989)

[8] Cortez, P et al. "A Data Mining Approach to Predict Forest Fires using Meteorological Data." *Proceedings of the 13th Portuguese Conference on Artificial Intelligence* (2007)

[9] Vega-Garcia, C., B. S. Lee, P. M. Woodard, and S. J. Titus. "Applying neural network technology to human-caused wildfire occurrence prediction." *AI Applications* (1996)

[10] Abadi, Martin et. al. "TensorFlow: A system for large-scale machine learning" *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)* (2016)

[11] Chollet, Francois. Keras, *Github respository* (2013)

[12] D. Stojanova, P. Panov, A. Kobler, S. Dzeroski, and K. Taskova. "Learning to Predict Forest Fires with Different Data Mining Techniques". *International multiconference Information Society (IS 2006)*(2006)

[13] "machrisaa/tensorflow-vgg", *Github repository* (2017)