Wildfire Prediction Using Deep Learning Models for Remote Sensing Data

Raghav Sharma and Rishabh Aggarwal

Problem Statement

- Develop a deep learning model to predict wildfire spread using remote sensing data
- The objective is to predict *where* the fire will spread the *next* day given fire location the previous day and other features influencing wildfire

Data

Data

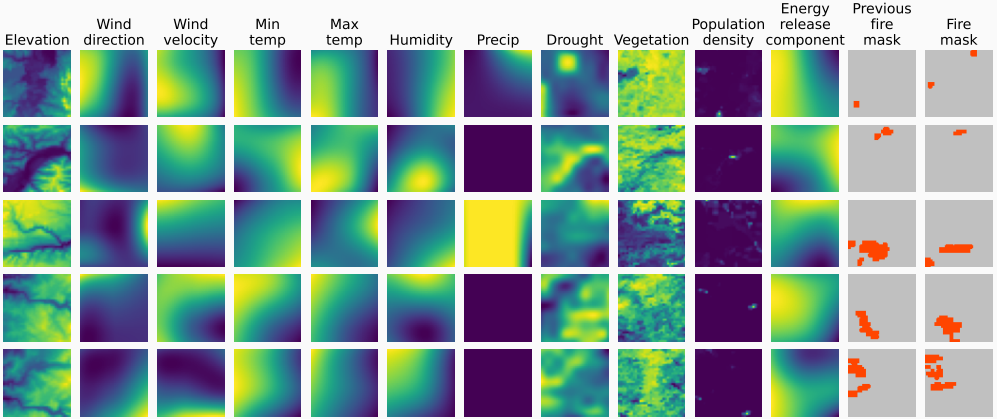
- Use publicly available 'Next Day Wildfire Spread' dataset[1] that uses remote sensing images from Google Earth Engine for the US from 2012-2020
- The data is extracted as images of 64km $\times$ 64km regions at 1 km resolution
- The data includes historical wildfire data and information on features that influence wildfire at same time and same location
- The historical wildfire images are processed as fire masks showing the locations of 'fire' versus 'no fire'. This includes fire masks for both days t and $t+1$
- 11 input features: elevation, wind direction and wind speed, minimum and maximum temperatures, humidity, precipitation, drought index, normalized difference vegetation index (NDVI), population density and energy release component (ERC)
- Treat previous day's fire-mask as an input feature

Preprocessing

- Each feature is clipped at 0.1% and 99.9% percentiles
- Normalized by subtracting mean and scaling by standard deviation
- Randomly crop 32km$\times$32km regions

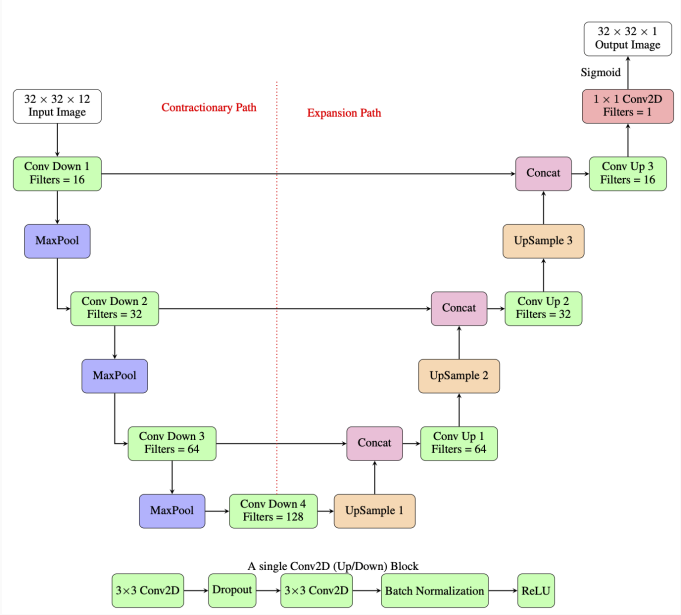
[1] https://www.kaggle.com/fantineh/next-day-wildfire-spread

Examples from the dataset



Model Architecture

- A U-Net like convolutional neural network for image segmentation problem



Hyperparameters Tuning

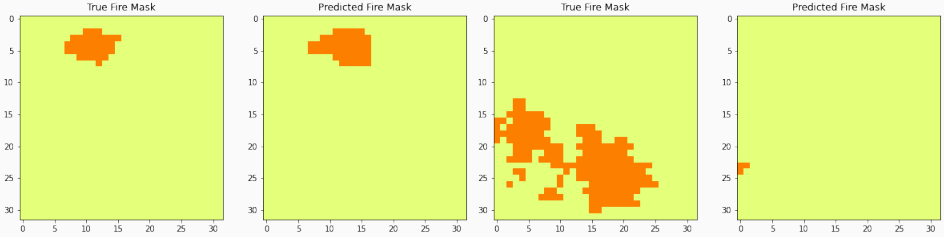
	(1) Loss	(2) Dice Coefficient	(3) IoU
	Panel A: Learning Rate (dropout rate = 0.1, batch size = 32)		
$\alpha = 0.0001$	0.0851	0.2155	0.1220
$\alpha = 0.001$	0.0844	0.2104	0.1187
$\alpha = 0.01$	0.0837	0.2018	0.1134
	Panel B: Batch Size (learning rate = 0.001, dropout rate = 0.1)		
Batch Size = 16	0.0853	0.2164	0.1225
Batch Size = 32	0.0844	0.2104	0.1187
Batch Size = 64	0.0858	0.2177	0.1234
	Panel C: Dropout Rate (learning rate = 0.001, batch size = 64)		
Dropout Rate = 0.1	0.0858	0.2177	0.1234
Dropout Rate = 0.2	0.1002	0.2276	0.1302

Results

Prediction Results

	(1) Loss	(2) Dice Coefficient	(3) IoU
Validation Set	0.1443	0.2577	0.1652
Test Set	0.1849	0.2154	0.1216

Prediction Examples



(a) Good Predictions (b) Bad Predictions

Feature Analysis

- Remove the features that contain similar information i.e. highly correlated with others
- E.g. Minimum temperature highly correlated with max temp

	(1) Loss	(2) Dice Coefficient	(3) IoU
All features	0.1443	0.2577	0.1652
7 features	0.1582	0.2397	0.1372
9 features	0.1593	0.2526	0.1460

- Models with 9 features perform similarly relative to the model with all features in terms of dice coefficient
- However, IoU is highest when all features are included

Conclusion

- Develop a U-Net like convolutional neural network to predict where the wildfire will spread the next day using remote sensing images from GEE
- The model has a dice coefficient of 21.5% and IoU of 12.2%
- Including 12 features does not lead to overfitting as evident from the feature analysis

Thank You!