Image Segmentation For Wildfire Prediction

Nevin George
Stanford University
ngeorge4@stanford.edu

Anthony Maltsev
Stanford University
amaltsev@stanford.edu

Abstract

Predicting the spread of wildfires day over day is crucial for disaster preparedness and for planning emergency services responses. We approach the problem of wildfire spread prediction by implementing different computer vision models, such as convolutional auto encoders and vision transformers, and applying them to a large multivariate dataset curated for this task. We analyze the models to identify features that carry significant predictive power to identify areas of critical interest when dealing with wildfires. Despite the expressive power of these architectures, our models underperform relative to a baseline convolutional autoencoder previously proposed for this task, highlighting the difficulty of learning from real-world spatial data.

1. Introduction

The ever increasing frequency and intensity of wildfires, driven by global climate change, are an important environmental challenge in the modern era. Accurate and timely prediction of how a wildfire will spread day by day is critical for effective management by local authorities. Accurate predictions would enable authorities to efficiently allocate limited resources for fire fighting as well as potentially coordinating evacuations of people living in affected areas. Predicting the spread of a wildfire is a complicated prediction task influenced by a number of climatological and topographic features associated with the area around a fire, making this a task suitable for end-to-end deep learning computer vision models.

In this paper, we investigate the performance of several different models on the task of wildfire spread prediction, including convolutional auto encoders (CAEs) such as U-Nets which combine high level learned representations with fine grained local features, and visual transformers (ViTs) which capture global context from images through attending across patches. After benchmarking the different models, we also perform feature analyses on the trained models in order to identify which features have the most predictive power and if there are specific patterns that are highly influential in predicting wildfire spread. We will look at model-agnostic feature analysis methods such as gradient based saliency maps as well as model specific methods like Class Activation Mappings (CAM) for CNNs and attention rollout for ViTs. We provide human-comprehensible interpretations of high importance features to identify high risk landscape features.

1.1. Problem Statement

The problem we are trying to solve is image segmentation over a high-dimensional input image. Given a 64x64x12 input image, which represent information about the landscape around an active fire (including a mask of the current fire), the goal is to produce a 64x64x1 map where for each pixel (representing a portion of the landscape) we predict the probability that the fire will spread to that area in the next day. Some ground truth masks contain 'unknown' pixel labels which are discarded during loss computations and evaluation for that example.

2. Related Work

There is a significant amount of prior technical methods that are related to our work, including model architectures for segmentation and post-hoc feature analysis. There has also been a significant body of work related to wildfire prediction since Huot et al released their dataset in 2022 [11]. These can be largely grouped into two categories: technical methods and dataset expansions / alterations.

2.1. Related Technical Methods

There is a large variety of model architectures that have been designed for semantic segmentation in computer vision. One popular such architecture is a UNet [17], which can be thought of as an encoder-decoder model with residual connections to preserve finegrained detail. We discuss this model more later in this paper. A similar model is the Convolutional Auto Encoder (CAE), which uses convolutional downsampling followed by upsampling, but without residual connections [6]. This is the model used by Huot et al.

More recently, transformer based architectures for semantic segmentation have been quite popular, achieving the state of the art on many benchmarks, including the wildfire prediction dataset we are studying. One such popular model is the SegFormer, which combines hierarchical encodings with MiT-B0 transformer-based attention blocks with feature level information similar to a UNet. We discuss this model more in future sections of the paper. A significant recent advancement in this field has been the FuseFormer [14]. The FuseFormer is specifically designed for high dimensional data such as the spatial data present in the wildfire dataset. It uses custom fusion blocks efficiently fuse a large number of homogeneous modalities as well as a novel transformer decoder architecture to recover pixel level segmentation predictions. The authors report **43.5** precision, **48.8** recall, and **39.0** F1 score, the current state of the art for the wildfire spread prediction benchmark.

There is also a significant body of work on interpreting and analyzing trained models. We are particularly interested in this paper in broadly applicable gradient based methods, which highlight areas of the input that carry strong predictive signal for any model. We rely on two such methods, Smoothgrad [19] and Grad-CAM [18], both of which we discuss in more detail later.

2.2. Related Data Works

There has been a significant amount of work by people in constructing similar datasets for tasks related to wildfire prediction, such as fusing spatial information for wildfire prediction in Morocco [12], collecting more granular data [4], relaxing the task to binary classification of whether there will be a fire the next day [12], and constructing newer similar datasets from SENTINEL satellite imagery [23]. These new datasets provide a wide range of benchmarks for training computer vision models on highly complicated, real world data.

3. Methods

3.1. Models

In this section, we describe the technical details of the U-Net model and Vision Transformer model that we trained for this task.

U-Nets, introduced by Ronneberger et al in 2015 [17], is a convolutional neural network architecture specifically designed for image segmentation tasks. This model architecture consists of two parts: an encoder pass which downsamples the image into an abstract representation through sequential convolutional and pooling layers, followed by a decoder which upsamples the encoded image representation by concatenating with residual connections from the encoder pass and then passing that through inverse convolutional layers until the original image resolution is recovered. The high level idea of the model is to combine fine grained features with high level abstract representations to combine semantic/abstract understanding with detailed location information, both of which are important for image segmentation.

SegFormers, introduced by Xie et al in 2021 [22], is a vision transformer architecture that is specifically designed for image segmentation tasks and shares some conceptual similarities with U-Nets. The model architecture consists again of two parts: an encoder pass which consists of 4 stages of mix transformer encodings which downsample the original input into a higher level, abstract representation, followed by a decoder which projects all of the different encoder stage outputs to a common shape, concatenates them, and then passes them through a simple MLP segmentation head to produce the final predictions. The mix transformer stages of the encoder differ from the typical vision transformer layers in that the patches overlap, encouraging continuity/smoothness between adjacent patches. Because of this, these stages forego the positional encodings that are typically used by vision transformers, instead relying on the spatial continuity of overlapping adjacent patches, similar to a convolutional neural network. The authors also implement an efficient self attention mechanism that reduces runtime from $O(n^2)$ to $O(\frac{n^2}{R})$, where R is a chosen reduction factor (ranging from 64 to 1 throughout the different encoder stages). We forego this in our model because our images are low resolution, so this is not a significant improvement in our use case. This architecture is similar to the concept of U-Nets in that both start with an encoder pass to extract high level semantic information and concatenate many layers of the encoder during the decoder phase.

3.2. Feature Analysis

In this section, we will describe the different methods we will use for feature analysis. Primarily, we will focus on gradient based methods.

Gradient methods for feature analysis are architecture-agnostic methods (so long as the architecture is differentiable) for interpreting the predictions of neural network classifiers. These methods compute the gradient of the output with respect to the input via backpropagation which produces a saliency map indicating how sensitive the model's predictions are to each individual input pixel. Input features with larger gradients can be interpreted as having high influence or predictive power in the model. Typically, gradients are gated to only positive values to indicate inputs are supporting the final prediction. For segmentation tasks like ours, we compute the gradient with respect to all pixels where the model predicted the positive class (eg 'fire') and aggregate those into a single attribution saliency map. In our project, we implement SmoothGrad, introduced by Smilkov et al in 2017 [19]. SmoothGrad addresses the issue

of high variance in the gradient maps by computing saliency maps for many perturbed versions of the input and averaging them to a single saliency map. The inputs are perturbed by adding gaussian noise, with a standard deviation of 15% of the range of input values.

We also investigate Grad-CAM, introduced in 2016 by Selvaraju et al [18], which is a spatial feature attribution method for analyzing convolutional networks. This method works by computing the importance of each feature dimension/channel by taking the average gradient, and sum over the channels weighted by their average gradient. This is useful to identify locations in the image (a subset of the land) which is important to the class predictions without breaking it down by each feature. In the context of our task, this simply identifies areas of interest to the fire prediction without identifying why they are important (e.g., some feature in elevation, or the amount of precipitation).

4. Dataset and Features

In this paper, we work with the Next Day Wildfire Spread dataset introduced by Huot et al in 2021 [11]. This dataset contains multidimensional information about 64 kilometer by 64 kilometer patches of land area in which a fire occurred on a specific day. The data includes a mask of the fire's spread on a specific day as well as many explanatory/informational features such as elevation maps, wind speed and direction, temperatures, humidity, precipitation, drought index, vegetation, population density, and energy release component. This data is combined into 64x64x12 images, along with ground truth results about the next day's wildfire spread. This naturally induces an image segmentation task where a model predicts the next day's fire spread from the features provided.

The dataset contains 18445 64x64x12 images, each of which represents a 64 kilometer by 64 kilometer region in which there is an active fire. This is a low resolution representation, as each pixel represents a 1 square kilometer area in real life. See figure 1 for examples of images, broken down by feature. The fire masks are grey in areas with no fire, red in areas with a fire, and dark grey in unknown areas (these are discarded during loss computation).

This dataset collates domain relevant feature information from a number of different sources: historic wildfire data from the MOD14A1 dataset from the Google Earth Engine project [7], topographic information from the Shuttle Radar Topography Mission [13], weather and drought data from the GRIDMET project [1] [2], vegetation information from the NASA VIIRS project [3], and population density information from GPWv4 [5].

5. Experiments/Results/Discussion

5.1. Primary Metrics

Like Huot et al. in [11], we primarily optimize for AUC PR (area under the precision-recall curve). The AUC PR calculates the precision and recall for different classification thresholds. It plots these (precision, recall) points in a curve using interpolation techniques, and then calculates the area under the curve. AUC PR ranges between 0 and 1, with 1 indicating a perfect classifier.

AUC PR is particularly useful when the dataset contains class imbalance with positive examples appearing less often. This is because 1) AUC PR provides a more informative picture of model performance by considering different classification thresholds, and 2) AUC PR optimizes for precision and recall, which only consider positive predictions. The Next Day Wildfire Spread dataset contains significant class imbalance, with around 1% of the total number of pixels in the fire masks containing a fire. Hence, AUC PR is a suitable metric for our task.

Along with AUC PR, we also calculate the F1 Score, precision, and recall for a threshold of 0.5.

5.2. Baseline Results

The authors of the dataset [11], Huot et al., include results from a baseline convolutional auto encoder, random forest classifier, and logistic regression model. We copy their baseline results into table 1. Their results show that the convolutional auto encoder performs better than the more traditional machine learning methods (random forest classifiers and logistic regression) under the AUC PR and precision metrics.

Baseline Model	AUC PR	Precision	Recall
CAE	28.4	33.6	43.1
Random Forest	22.5	26.3	46.9
Logistic Regression	19.8	32.5	35.3

Table 1: Huot et al. baseline results

The authors also include a feature analysis via an ablation study in which they remove one of the features at a time, retrain the models, and observe how performance changes. The authors observe similar performance among all of these ablations except when the fire mask feature is removed, which causes a sharp drop in performance from the baselines. With this result, the authors also experiment with only including one feature at a time along with the fire mask. They observe that the 'elevation' and 'vegetation index' features perform best, achieving 27.0 and 28.2 AUC respectively. The authors provide no feature analysis beyond these dimensional ablation studies.

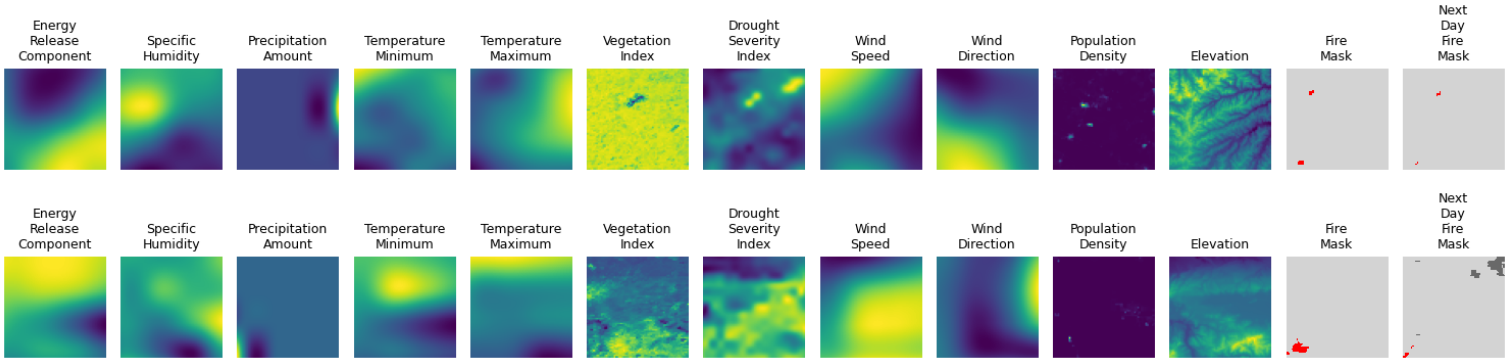


Figure 1: Dataset examples broken up into feature maps

5.3. Experiments

5.3.1 Model Architecture

U-Net: We follow the standard U-Net architecture fairly closely. Our U-Net consists of three encoder blocks, a bottleneck block, and three decoder blocks. With the input being 12x64x64, the encoder and bottleneck blocks each halve the height and width and double the number of channels (beginning with 32 channels). The decoder blocks follow a reverse process, doubling the height and width and halving the number of channels. Each block contains two convolutional layers (kernel=3, padding=1) along with batch norm, ReLU, and dropout layers (p=0.3). Each encoder block is followed by a 2x2 max pool layer, and before each decoder block, there is a 2x2 up-convolution layer. Our model also uses skip connections between corresponding encoder-decoder layers.

SegFormer: We implement the the MiT-b0 and MiT-b1 SegFormer model variants available on HuggingFace [9, 22]. The MiT-b0 and MiT-b1 models have 3.7M and 14.0M parameters, respectively. The added complexity in the MiT-b1 model is due to larger hidden layer sizes: [32, 64, 160, 256] versus [64, 128, 320, 512].

5.3.2 Hyperparameters

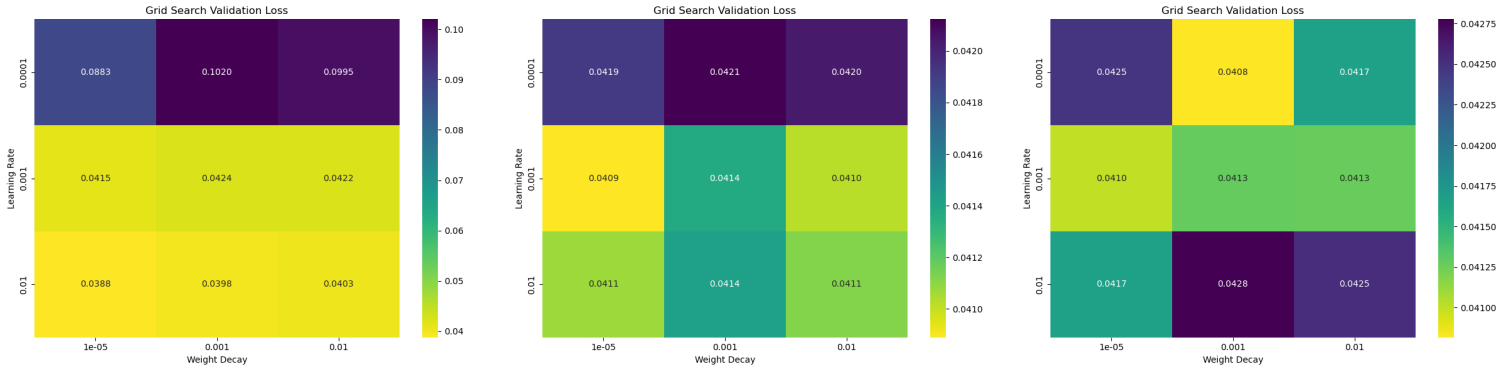
- We use an approximately 80/10/10 train, validation, test split of the 18445 data points.
- Batch sizes of 4-16 are standard for image segmentation tasks of our dataset size, with larger batch sizes preferred. We use a batch size of 16, as we encountered no issues with fitting the larger batches in memory.
- We use the AdamW optimizer. This is a variant of Adam that decouples weight decay from the gradient update, leading to better generalization. The AdamW optimizer is standard for transformers and many common image segmentation architectures (e.g., U-Net).
- To determine the optimal loss function and weight decay, we employ a grid search, the results of which are in Figure 2. The optimal found (learning rate, weight decay) pairs are (1e-02, 1e-05), (1e-03, 1e-05), and (1e-04, 1e-03) for the U-Net, SegFormer-B0, and SegFormer-B1 architectures, respectively.
- During training, we use a weighted cross-entropy loss function, a learning rate scheduler (reduce on plateau), and early stopping after five epochs of no improvement in the validation loss.
- For each of the models, we determine the confidence threshold for pixel classification that yields the best F1 score on the validation set. We do this using the function `precision_recall_curve` from `sklearn.metrics`, which calculates the precision and recall for various thresholds. The optimal thresholds found for the U-Net, SegFormer-B0, and SegFormer-B1 models are 0.97, 0.26, 0.10, respectively.

5.4. Results

Model	AUC PR	F1 Score	Precision	Recall
U-Net	19.39	26.56	23.47	30.58
SFB0	17.72	25.84	22.71	29.98
SFB1	17.91	27.24	24.21	31.12

Table 2: Model performance

Our best model results on the test set after the hyperparameter search are in Table 2. The U-Net architecture achieved the best AUC PR, performing slightly better than the SegFormer models which had roughly similar performance. The U-Net AUC PR is slightly worse than than the



(a) U-Net (b) SegFormer-B0 (c) SegFormer-B1

Figure 2: Learning rate and weight decay grid search results

Logistic Regression baseline model, and significantly worse than the best baseline model (the CAE). The F1 score, precision, and recall are relatively the same for all three models.

Figure 3 contains the training and validation loss curves for the model training. After the first epoch, the validation accuracy fails to improve significantly for all three models, leading to early stopping within ten epochs. For all three models, training loss continued to decrease slightly over time, while validation loss either roughly stabilized or got worse as in the case of the U-Net. This suggests that the models struggle to generalize and may overfit to the training set without early stopping. For the U-Net and SegFormer-B0 models, the training loss didn't decrease significantly from the starting loss, which suggests that they lack the necessary expressivity and feature extraction capabilities for the task. The SegFormer-B1 had a larger drop in training loss in the first epoch, and its results on the test are also slightly better than that of the SegFormer-B0 model, which suggests that in this case having a larger transformer model was useful.

The difficulty of the task is augmented by the extreme class imbalance. As previously mentioned, only around 1% of the total pixels contain a fire. The SegFormer models' low optimal threshold values suggest that they are highly influenced by the negative examples and have low confidence in whether a pixel contains a fire. To help address this, we experiment with different class weights for the loss function, the results of which are in Table 3. The U-Net model achieved the highest AUC PR with weights inversely proportional to the class sizes. The SegFormer-B0 and SegFormer-B1 performed the best with class weight distributions of 0.25/0.75 and 0.5/0.5, respectively.

The relatively low results of the models overall raise into question the usefulness of the twelve input features in predicting the next day fire masks. The authors of the dataset [11] also mention this, saying that the previous day fire mask was the only particularly useful feature out of the twelve for the segmentation task in their experiments.

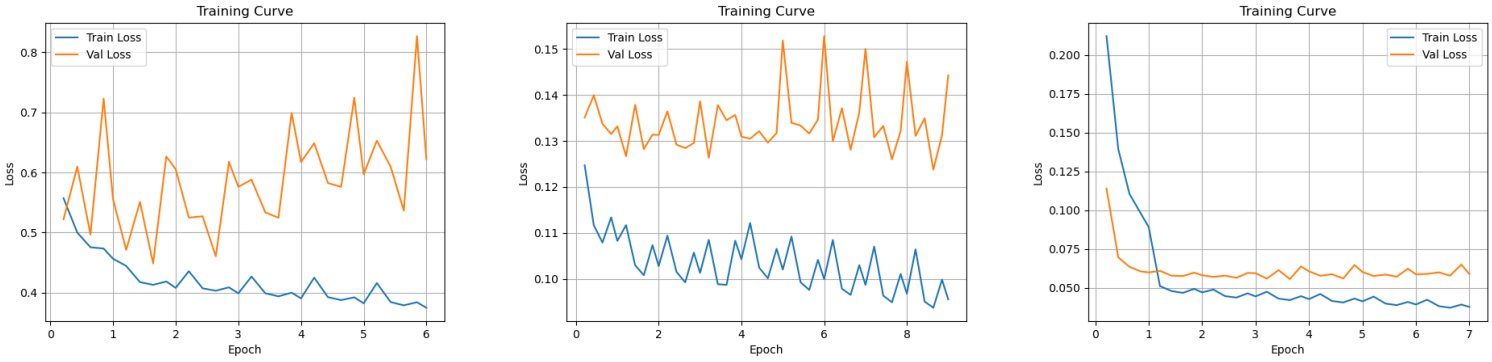
Model	Inv class size	0.1/0.9	0.25/0.75	0.5/0.5
U-Net	**17.59**	15.14	17.13	12.77
SFB0	10.26	13.33	**14.44**	13.17
SFB1	8.33	10.52	11.43	**12.34**

Table 3: AUC PR for the models with different loss function weights

5.5. Feature Analysis/Qualitative Results

In this section, we will discuss some gradient based feature analysis (SmoothGrad and Grad-CAM) of our best trained models.

We first analyze the UNet model, whose feature analysis can be found in figure 4. In this model, we observe that the model is very imprecise: the predicted fire map rarely coincides with the actual fire mask, instead aligning closely with the input features of vegetation index, drought severity, and elevation. In the example provided, we observe that the area of predicted fire aligns with the areas with low vegetation and low drought, as well as low elevation. This likely corresponds to a lake, or something of the sort. We also observe that both the Smoothgrad saliency maps and the Grad-CAM from the final convolutional layer of the UNet closely match the spatial location of those features and the predicted firemap. This suggests that the model is learning primarily from the finegrained local features of the input image, disregarding the input from the down-to-upsampling path of the UNet (which encodes more global, semantic information than finegrained pixel-level information). This is qualitatively true even when the model performs fairly well, predicting fire in roughly the same locations as the actual firemaps (as in figure 5). We also observe that most of the features have roughly the same saliency maps, sug-



(a) U-Net (b) SegFormer-B0 (c) SegFormer-B1

Figure 3: Training and validation loss curves

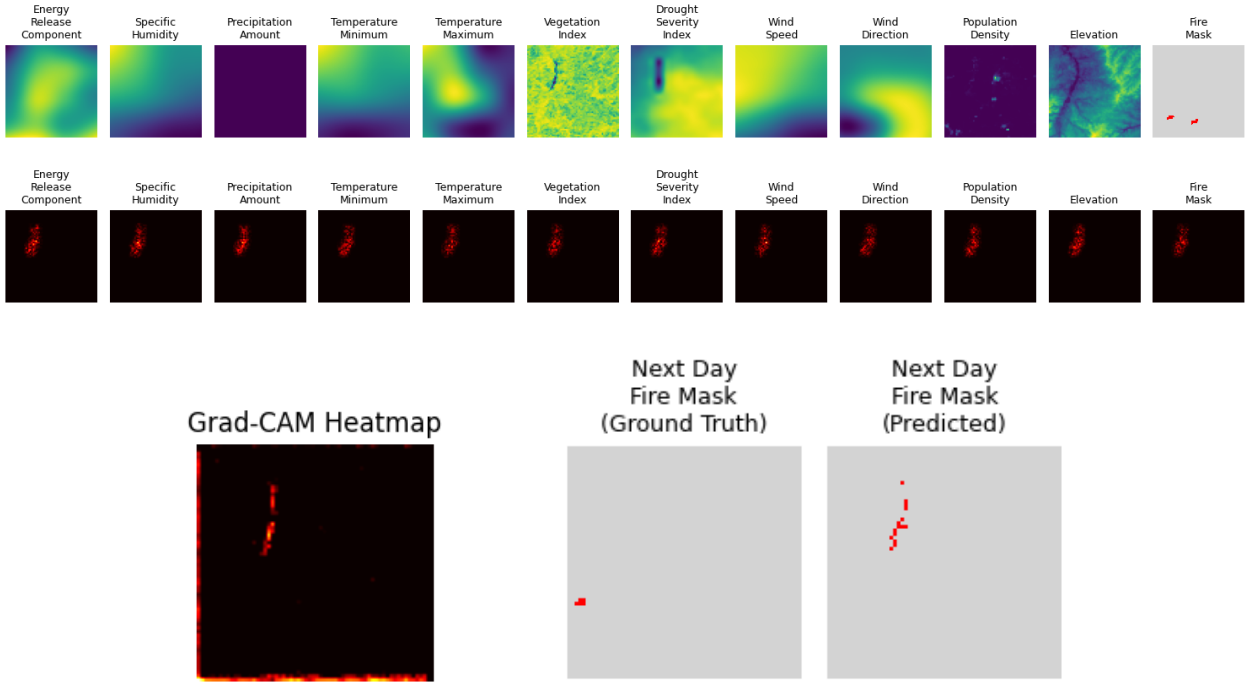


Figure 4: UNet Example 1: (a) Input features. (b) Smoothgrad saliency map. (c) Grad-CAM heatmap. (d) Next-day fire mask prediction vs. ground truth.

gesting that the model does not differentiate between the channels, instead favoring spatial information. We also, interestingly, see little to no influence from the previous day's fire map, which the authors of the dataset identify as the single most important feature which carried the most signal for their experiments. This might explain why our model provided quantitatively worse results than the baseline models provided by Huot et al [11].

We now analyze the SegFormerB0 model, whose feature analysis can be found in figure 6. Qualitatively, the results of this model are far less cleanly interpretable than the UNet architecture, however there are still some similarities between the models. In the provided example, the saliency maps suggest that model attended to a strip on the left side of the image corresponding to drastically lower vegetation and higher elevation. This suggests that the model attended more to interesting spatial features of the image rather than specific features. However, this model differs in the fact that the model prediction seems largely unrelated to the features that it attended to, instead predicting a large blob of fire in

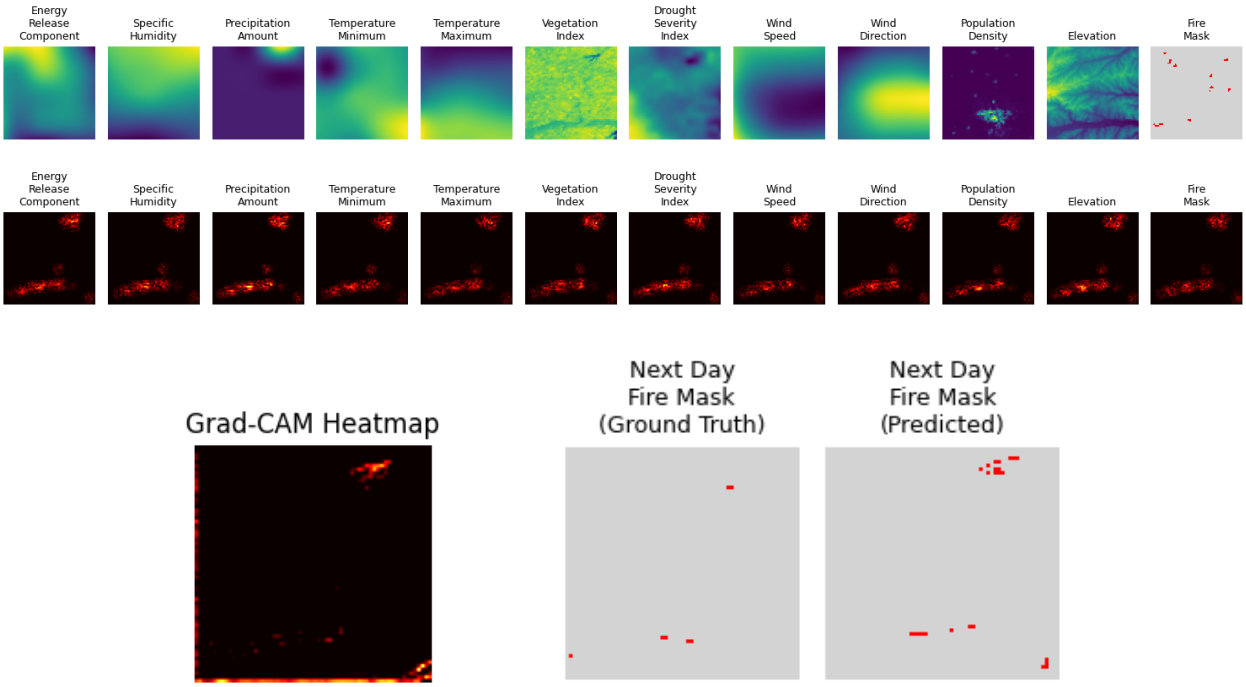


Figure 5: UNet Example 2: (a) Input features. (b) Smoothgrad saliency map. (c) Grad-CAM heatmap. (d) Next-day fire mask prediction vs. ground truth.

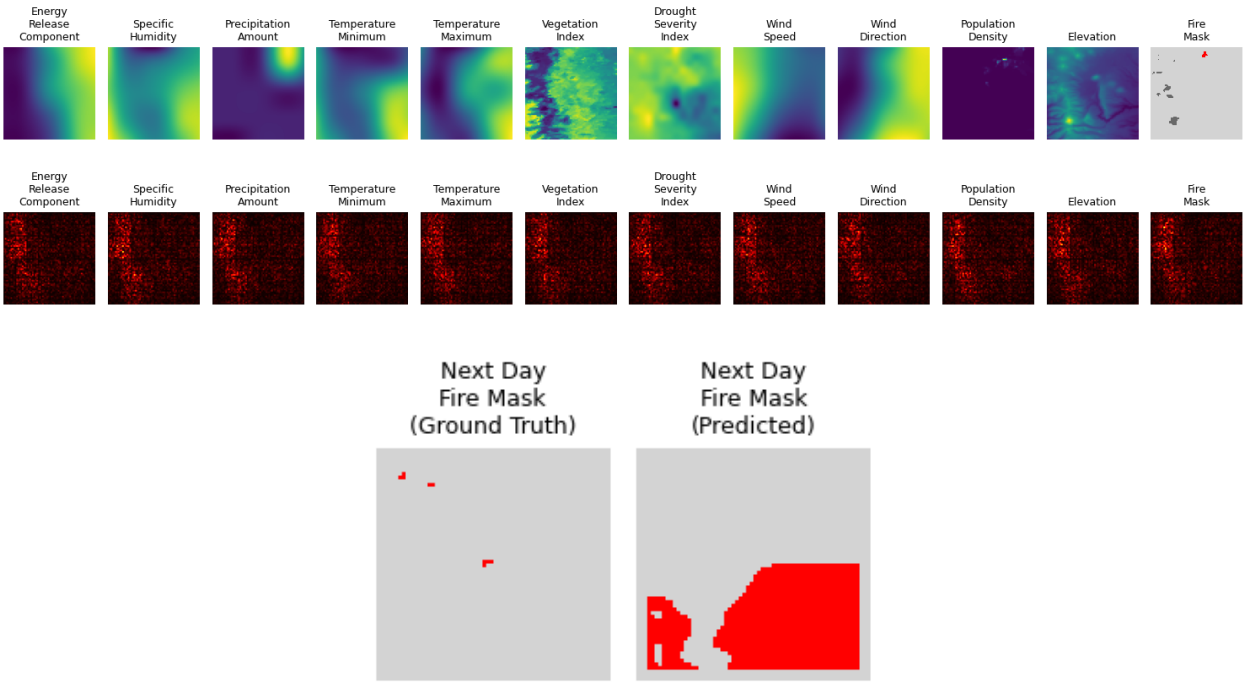


Figure 6: SegFormerB0 Example 1: (a) Input features. (b) Smoothgrad saliency map. (c) Next-day fire mask prediction vs. ground truth.

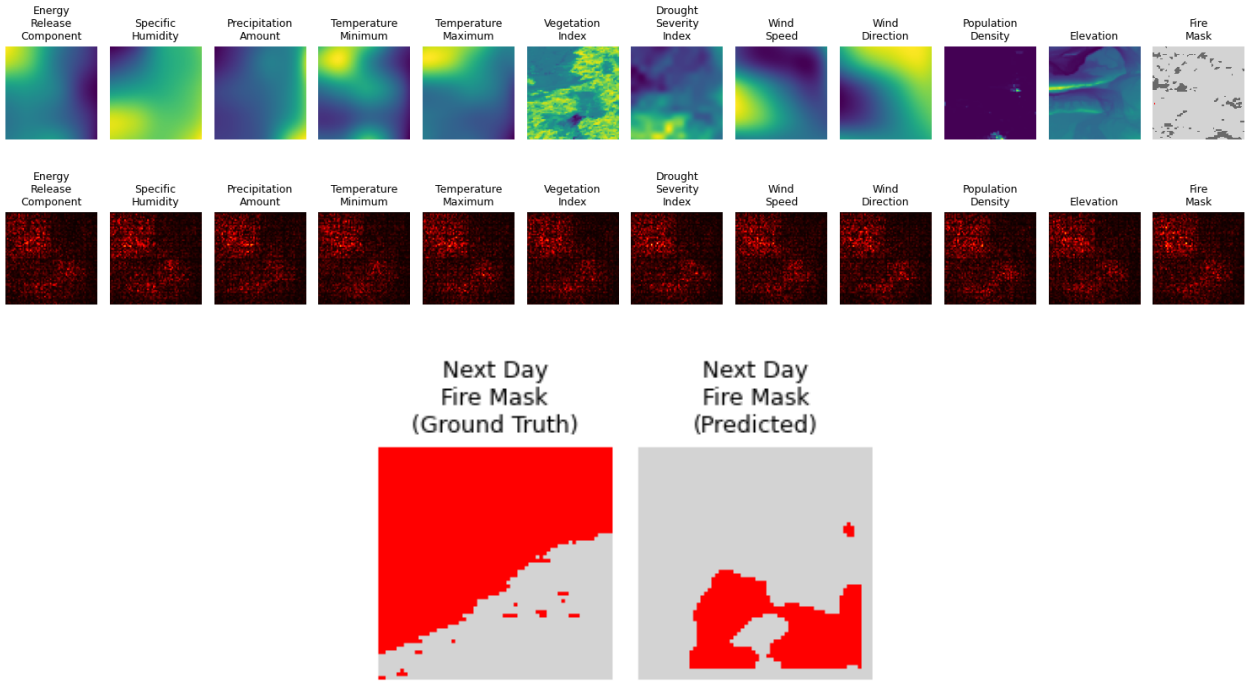


Figure 7: SegFormerB0 Example 2: (a) Input features. (b) Smoothgrad saliency map. (c) Next-day fire mask prediction vs. ground truth.

the lower right hand corner. Qualitatively, this model would heavily overpredict fires, producing huge blobs of fire, usually centered on the lower left hand side of the screen. In the second example in figure 7, we observe similarly that attention is focused on the upper left hand corner, where there is a significant drop in vegetation and lower elevation, spatially significant features. The SegFormer model, similar to UNet, also does not appear to attend to the fire mask of the current day.

We also observe that the SegFormer model attends to far more of the image than the very localized UNet. This reflects the fact that the SegFormer uses a Vision Transformer architecture which is able to attend to all patches in the image, instead of just the local field around the patch, as in the case of the CNN based UNet. This global reach means that the saliency maps for the SegFormer are significantly more noisy than for UNet.

We omit analysis for the larger SegFormer model because it is qualitatively similar to the smaller model.

6. Conclusion/Future Work

In this paper, we applied UNet and SegFormer models to the task of wildfire spread prediction. Our experiments revealed that both of these models struggled to outperform even simplistic baseline models provided by the dataset authors, suggesting that these architectures may have difficulty learning from the provided feature set.

Our feature analysis using two gradient based methods – Smoothgrad and Grad-CAM – showed that both models largely ignored a feature that had strong predictive power for the dataset authors, the fire mask. Instead the models focused on spatial features such as elevation, vegetation index, and drought severity, often producing spatially biased predictions. The best model provided by the dataset authors [11] specifically discarded feature level information from residual connections in their convolutional auto encoder, forcing the model to make predictions from the higher level semantic features learned during encoding. This suggests a future direction for work on this task: it would be interesting to experiment with methods to reduce dependence on finegrained spatial features and perform feature analysis on those new architectures. Another future direction of work would be in incorporating more temporal information in the form of previous days' fire masks which would show the historical spread of the fire. Since the current day's mask was a strong feature for the dataset authors, we think this would yield stronger results. Also, one could incorporate domain specific physical knowledge like physics based fire spread models into deep learning models, which might yield more interpretable results. Alternatively, scaling our existing approaches to have larger, deeper architectures might yield better results, as the dataset authors' models are larger than ours.

Overall, this work highlights the difficulty of training models for complicated real world tasks, such as wildfire spread prediction. The data may not carry enough signal to accurately predict wildfire spread, causing difficulty learning accurate computer vision models.

7. Contributions & Acknowledgments

Nevin: model architecture, most of the model training, hyperparameter searches

Anthony: feature analysis, some of the model training, poster

Both: debugging models, writing the report

Code for the project can be found here: `https://github.com/nevingeorge/wildfire_prediction`

We would like to acknowledge and thank the course staff of CS231N for their guidance and support throughout the project.

References

[1] J. T. Abatzoglou. GRIDMET: University of Idaho Gridded Surface Meteorological Data. Google Earth Engine, 2013. Accessed: 2025-05-14.

[2] J. T. Abatzoglou, D. J. McEvoy, and K. T. Redmond. GRIDMET: University of Idaho Gridded Surface Meteorological Data - Drought Indices. Google Earth Engine. Dataset based on methodology from Abatzoglou, J. T. (2013), International Journal of Climatology, 33(1), 121-127. Earth Engine dataset provider: University of Idaho, Desert Research Institute., 2013. Accessed: 2025-05-14.

[3] N. Aeronautics and S. A. N. L. DAAC. VIIRS/NPP Vegetation Indices 16-Day L3 Global 500m SIN Grid V001. NASA EOSDIS Land Processes DAAC, 2018. Accessed: 2025-05-14.

[4] S. H. Ali, S. Zhang, S. Mhatre, and T. Xiao. Advancing wildfire predictive models: A novel dataset for next-day wildfire spread. In *2024 Conference on AI, Science, Engineering, and Technology (AIxSET)*, pages 69–76, 2024.

[5] Center for International Earth Science Information Network - CIESIN - Columbia University. Gridded Population of the World, Version 4 (GPWv4): Population Density, Revision 11. NASA Socioeconomic Data and Applications Center (SEDAC), 2018. Accessed: 2025-05-14.

[6] J. Dong, X.-J. Mao, C. Shen, and Y.-B. Yang. Learning deep representations using convolutional auto-encoders with symmetric skip connections, 2017.

[7] L. Giglio, W. Schroeder, and C. O. Justice. MODIS/Terra Thermal Anomalies/Fire Daily L3 Global 1km SIN Grid V006. NASA EOSDIS Land Processes DAAC, 2016. Accessed: 2025-05-14.

[8] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, et al. Array programming with numpy. *Nature*, 585(7825):357–362, 2020.

[9] Hugging Face. Segformer model documentation, 2025. Accessed: 2025-05-30.

[10] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.

[11] F. Huot, R. L. Hu, N. Goyal, T. Sankar, M. Ihme, and Y.-F. Chen. Next day wildfire spread: A machine learning dataset to predict wildfire spreading from remote-sensing data, 2022.

[12] A. Jadouli and C. E. Amrani. Advanced wildfire prediction in morocco: Developing a deep learning dataset from multisource observations. *IEEE Access*, 12:191733–191747, 2024.

[13] N. JPL. Nasa shuttle radar topography mission (srtm) global 1 arc second v003. NASA EOSDIS Land Processes DAAC, 2013. Accessed: 2025-05-14.

[14] J. McMillen and Y. Yilmaz. Fuseform: Multimodal transformer for semantic segmentation. In *Proceedings of the Winter Conference on Applications of Computer Vision (WACV) Workshops*, pages 618–627, February 2025.

[15] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32, 2019.

[16] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[17] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation, 2015.

[18] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, 128(2):336–359, Oct. 2019.

[19] D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg. Smoothgrad: removing noise by adding noise, 2017.

[20] M. L. Waskom. Seaborn: statistical data visualization. *Journal of Open Source Software*, 6(60):3021, 2021.

[21] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, and J. Brew. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, 2020.

[22] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo. Segformer: Simple and efficient design for semantic segmentation with transformers, 2021.

[23] Y. Xu, A. Berg, and L. Haglund. Sen2fire: A challenging benchmark dataset for wildfire detection using sentinel data. In *IGARSS 2024 - 2024 IEEE International Geoscience and Remote Sensing Symposium*, pages 239–243, 2024.