FIgLib & SmokeyNet: Dataset and Deep Learning Model for Real-Time Wildland Fire Smoke Detection

Anshuman Dewangan
adewangan@ucsd.edu

Yash Pande
ypande@ucsd.edu

Hans-Werner Braun
hwb@ucsd.edu

Frank Vernon
flvernon@ucsd.edu

Ismael Perez
i3perez@sdsc.edu

Ilkay Altintas
ialtintas@ucsd.edu

Gary Cottrell
gary@ucsd.edu

Mai H. Nguyen
mhnguyen@ucsd.edu

Abstract

The size and frequency of wildland fires in the western United States have dramatically increased in recent years. On high fire-risk days, a small fire ignition can rapidly grow and get out of control. Early detection of fire ignitions from initial smoke can assist response to such fires before they become difficult to manage. Past deep learning approaches for wildfire smoke detection have suffered from small or unreliable datasets that make it difficult to extrapolate performance to real-world scenarios. In this work, we present the Fire Ignition Library (FIgLib), a publicly-available dataset of nearly 25,000 labeled wildfire smoke images as seen from fixed-view cameras deployed in Southern California. We also introduce SmokeyNet, a novel deep learning architecture using spatio-temporal information from camera imagery for real-time wildfire smoke detection. When trained on the FIgLib dataset, SmokeyNet outperforms comparable baselines. We hope that the availability of the FIgLib dataset and the SmokeyNet architecture will inspire further research into deep learning methods for wildfire smoke detection, leading to automated notification systems to reduce the time to wildfire response.

1 Introduction

The size and frequency of wildland fires in the western United States have increased in recent years. In 2018 alone, 8,527 fires burned an area of 1.9 million acres (7,700 km^2; nearly 2% of the state's area) in California, with an estimated economic cost of $148.5 billion [1].

On high fire-risk days, a small fire ignition can rapidly grow and get out of control. Consequently, the detection of wildfires in the first few minutes after ignition is essential to minimizing their destruction. However, it can take much longer for a fire to be reported using existing methods, especially in areas with less human activity. Deep learning-based wildfire smoke detection systems can accurately and consistently detect wildfires and provide valuable intel to reduce the time to alert authorities.

The goal of a wildfire smoke detection system can be structured as a binary image classification problem to determine the presence of smoke within a sequence of images. Priorities include quick time-to-detection, high recall to avoid missing potential fires, high precision to avoid frequent alarms that undermine trust in the system [2], and efficient performance to operate in real-time on edge devices. However, the task proves challenging in real-world scenarios given the transparent and amorphous nature of smoke; faint, small, or dissipating smoke plumes; and false positives from clouds, fog, and haze.

Tackling Climate Change with Machine Learning: workshop at NeurIPS 2021.

Even before the rise in popularity of deep learning methods, the visual (e.g. color), spatial, and temporal (i.e. motion) features of smoke were recognized as essential for the machine detection of wildfires [3, 4, 5, 6]. More recently, deep learning approaches use a combination of convolutional neural networks (CNNs) [7, 8, 9, 10, 11, 12, 13, 14], background subtraction [7, 12, 15], and object detection methods [14, 16, 17, 18, 19] to incorporate visual and spatial features. Long short-term memory (LSTM) networks [12, 18] or optical flow [9, 15, 20] methods have been applied to incorporate temporal context from video sequences. However, the difficulty of acquiring a large, labeled wildfire smoke dataset has limited researchers to using small or unbalanced datasets [6, 18], manually searching for images online [8, 11, 13, 18], or synthetically generating datasets [13, 16, 21].

To address the need for a consistent evaluation benchmark for real-world performance, we present the Fire Ignition Library (FIgLib), a publicly-available dataset of nearly 25,000 labeled wildfire smoke images as seen from fixed-view cameras in Southern California. We also introduce SmokeyNet, a novel deep learning architecture using spatio-temporal information from camera imagery for real-time wildfire smoke detection. When trained on the FIgLib dataset, SmokeyNet outperforms comparable baselines in terms of accuracy and rivals human classification performance. We hope that the availability of the FIgLib dataset and the SmokeyNet architecture will inspire further research into deep learning methods for wildfire smoke detection, leading to automated notification systems to reduce the time to wildfire response.

2 FIgLib Dataset

The High Performance Wireless Research and Education Network (HPWREN) FIgLib dataset reflects sequences of wildland fire images as seen from fixed-view cameras on remote mountain tops in Southern California. As of September 2021, the dataset consists of 315 fires from 101 cameras across 30 stations occurring between June 2016 and July 2021. Each sequence typically contains images from 40 minutes prior to and 40 minutes following the start of the fire, serving as binary smoke/no-smoke labels for each image. The images are 2048x1536 or 3072x2048 pixels in size, depending on the camera model used, and are spaced approximately 60 seconds apart for a total of 24,800 high-resolution images. The ignition detection and view prior to the ignition are enabled by a cluster deployment of cameras, where four 90+ degree views stay consistent for years as 360 degrees around a mountain top. The FIgLib dataset can be accessed at the following link: `http://hpwren.ucsd.edu/HPWREN-FIgLib/`

3 Methods

3.1 Data Preparation

To avoid out-of-distribution sequences for our machine learning task, we removed fires with black & white images (N=10), night fires (N=19), and fires with questionable presence of smoke (N=16) from the dataset (3,700 images removed in total). In addition to binary smoke/no-smoke labels for each image, the smoke in 144 fires has been manually annotated with bounding boxes and contour masks. We used images from these 144 annotated fires for training (45.6%, 11,300 images); the remaining 126 fires (9,800 images) are split between the validation and test sets such that the number of images in each split is roughly equivalent. Additional data pre-processing steps are covered in **Appendix 6.1**.

3.2 SmokeyNet

Tiling Our goal is the binary classification of images to determine the presence of smoke as early in the sequence as possible. Training the model with standard CNN techniques by leveraging solely image labels provides insufficient training signal for the model to identify small plumes of smoke within the large images. Object detection models using bounding box and contour mask annotations can better localize the target object using anchors and a regression head [22]; however, these models require precise annotations, which poses a challenge in our scenario given the amorphous and transparent nature of smoke.

Consequently, we build upon previous work by tiling the image into 224x224 tiles, overlapping by 20 pixels for a total of 45 tiles [2]. We also generate corresponding binary *tile labels*: positive if the number of pixels of smoke in the tile, determined by the filled polygon of the contour mask, is

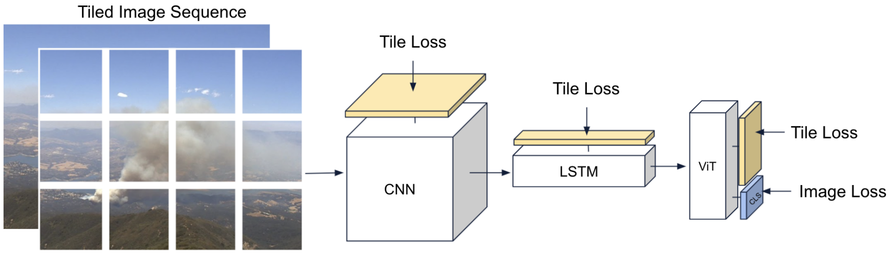

Figure 1: The SmokeyNet architecture takes two frames of the tiled image sequence as input and combines a CNN, LSTM, and ViT. The yellow blocks denote "tile heads" used for intermediate supervision while the blue block denotes the "image head" used for the final image prediction.

greater than an empirically-determined smoke detection threshold of 250 (0.5% of the total pixels in the tile). Tile labels provide the entirety of our localized feedback signal; we do not otherwise use the bounding box or contour mask annotations during training.

Architecture The SmokeyNet architecture, depicted in **Figure 1**, is a novel spatio-temporal gridded image classification approach combining three different types of neural networks: a CNN [23], an LSTM [24], and a vision transformer (ViT) [25]. The input to our model is the tiled raw image and its previous frame in the wildfire image sequence to incorporate the motion of the smoke. A CNN, pre-trained on the ImageNet dataset [26], initially extracts representations of the raw image pixels from each tile of the two frames independently. A ResNet34, a lighter-weight version of the popular ResNet50 model, is our preferred choice of CNN backbone. Then, an LSTM combines the temporal information of each tile from the current frame with its counterpart from the previous frame. Finally, all temporally-combined tiles are fed into a ViT, which incorporates spatial information *across* tiles to improve the image prediction.

The outputs of the ViT are spatio-temporal embeddings for each tile, as well as a CLS token embedding that summarizes representations for the whole image [25]. The CLS token embedding is passed to an "image head," consisting of three linear layers of output sizes 256, 64, and 1, respectively, and a sigmoid layer to generate a single prediction for the whole image. Given the modular nature of each of the components, we can experiment with different approaches to capture spatio-temporal information while still training the model end-to-end.

Loss The initial component of our loss applies standard binary cross-entropy (BCE) loss between the outputs of the image head and the ground-truth binary image labels. We can increase the weight of positive examples when calculating this BCE image loss to trade off precision for higher recall. We use the empirically-tested positive weight of 5 to achieve more balanced precision and recall and improve overall accuracy and F1 score. To leverage the localized information provided by the tile labels, we also apply intermediate supervision to each of the model components [27]. Since the model's components, the CNN, LSTM, and ViT, also produce embeddings on a per-tile basis, we pass each component's embeddings through individual "tile heads," consisting of three linear layers of output sizes 256, 64, and 1, respectively, and a sigmoid layer to generate predictions for each *tile*. We then apply BCE loss between the outputs of the tile heads and the binary tile labels. To address the class imbalance in which negative tiles occur more frequently than positive tiles, we weight positive examples by 40, the ratio of negative tiles to positive tiles.

If I is the total number of tiles, the overall training loss can be summarized as:

$$loss = BCE^{image} + \sum_i^I \{BCE_i^{CNN} + BCE_i^{LSTM} + BCE_i^{ViT}\}$$

Since we have tile labels for only the training data, we define our validation loss as the average number of *image* prediction errors and use this validation loss for early stopping.

3.3 Baselines & Experiments

We experiment with alternate CNN backbones to the ResNet34, including a MobileNetV3Large [28], MobileNetV3Large with a Feature Pyramid Network [29] to better incorporate spatial scales,

	Params	Time					TTD
Model	(M)	(ms/it)	A	F1	P	R	(mins)
Variants of SmokeyNet:							
ResNet34 + LSTM + ViT	56.9	51.6	83.49	82.59	89.84	76.45	3.12
ResNet34+LSTM+ViT (3 frames)	56.9	80.3	**83.62**	**82.83**	**90.85**	76.11	2.94
MobileNet + LSTM + ViT	36.6	28.3	81.79	80.71	88.34	74.31	3.92
MobileNetFPN + LSTM + ViT	40.4	32.5	80.58	80.68	82.36	79.12	2.43
EfficientNetB0 + LSTM + ViT	52.3	67.9	82.55	81.68	88.45	75.89	3.56
TinyDeiT + LSTM + ViT	22.9	45.6	79.74	79.01	84.25	74.44	3.61
ResNet34 (1 frame)	22.3	29.7	79.40	78.90	81.62	76.58	2.81
ResNet34 + LSTM	38.9	53.3	79.35	79.21	82.00	76.74	2.64
ResNet34 + ViT (1 frame)	40.3	30.8	82.53	81.30	88.58	75.19	2.95
ResNet50 (1 frame)	26.1	50.4	68.51	74.30	63.35	**89.89**	**1.01**
FasterRCNN (1 frame)	41.3	55.6	71.56	66.92	81.34	56.88	5.01
MaskRCNN (1 frame)	43.9	56.9	73.24	69.94	81.08	61.51	4.18
ResNet34 + ResNet18-3D	38.0	57.5	83.10	82.26	88.91	76.65	2.87

Table 1: Accuracy (A), F1, precision (P), recall (R) and average time-to-detection (TTD) evaluation metrics on the test set, with 2 frames of input (unless otherwise stated) averaged over 5 runs. Best results are **bolded**; second-best results are underlined. Number of parameters (in millions) and inference time (ms/image) should be minimized for deployment to edge devices.

EfficientNet-B0 [30], and Data Efficient Image Transformer (DeiT-Tiny) [31]. Using the ResNet34 as the backbone, we also experiment with 3 input frames (i.e., two additional frames of temporal context instead of one) and conduct an ablation study by removing different parts of the model to evaluate each component's benefits. Finally, we compare the model's performance to four alternate architectures: ResNet50, the standard for image classification models [32]; Faster-RCNN, a standard object detection model [22]; Mask-RCNN, an image segmentation model leveraging both contour masks as well as bounding boxes for training signal [33]; and an alternate architecture using a CNN and a ResNet18-3D CNN that captures spatio-temporal relationships as a replacement to our LSTM + ViT [34]. Additional training and model implementation details are covered in **Appendix 6.2**.

4 Results & Discussion

Table 1 reports test evaluation performance for each of the architectures. SmokeyNet delivers on the objectives of high precision, high recall, fast performance, and low average time to detection, calculated as the number of minutes until the model correctly predicts the first positive frame of a wildfire sequence, averaged over all fires. Additional frames of input marginally improve performance while drastically increasing inference time. Large backbones such as the ResNet34 or EfficientNet-B0 trade off model size and inference time for better accuracy compared to smaller backbones such as the MobileNetV3Large or MobileNetFPN.

From the ablation study, we observe that the stand-alone CNN or CNN+LSTM models perform poorly at the task. Adding the ViT to the CNN significantly improves performance with little impact to inference speed. The SmokeyNet architecture clearly outperforms standard image classification, object detection, and image segmentation models. The CNN+ResNet18-3D architecture performs slightly worse than SmokeyNet, but provides another viable alternative if prioritizing model size.

A video of SmokeyNet's performance on images from the test set can be viewed at this link: `https://youtu.be/cvXQJao3m1k`. The model performs well in a variety of real-world scenarios, correctly identifying apparent smoke plumes while avoiding clouds and haze. However, the model still makes systematic misclassifications of low-altitude clouds as false positives.

For future work, we will continue improving the performance of SmokeyNet in these difficult scenarios, particularly incorporating ignition location information and exploring self-supervised methods using additional unlabeled data. We also aim to reduce the model size for better compatibility with edge devices using modified hyperparameters, pruning [35], and quantization.

5 Acknowledgements

The authors would like to thank Brian Norton for sharing his invaluable expertise on wildfire management; Stephen Jarrell, Duolan Quyang, and Atman Patel for their collaboration and insights; Scott Ehling and his team of labellers at Argonne National Laboratory for contributing to dataset annotations; and the UC San Diego CHASE-CI team for computer support.

This work was supported in part by NSF award numbers 1730158, 2100237, 2120019 for Cognitive Hardware and Software Ecosystem Community Infrastructure (CHASE-CI), and 1331615, 2040676 and 1935984 for WIFIRE, WIFIRE Commons, and SAGE.

References

[1] Daoping Wang, Dabo Guan, Shupeng Zhu, Michael Mac Kinnon, Guannan Geng, Qiang Zhang, Heran Zheng, Tianyang Lei, Shuai Shao, Peng Gong, et al. Economic footprint of california wildfires in 2018. *Nature Sustainability*, 4(3):252–260, 2021.

[2] Kinshuk Govil, Morgan L Welch, J Timothy Ball, and Carlton R Pennypacker. Preliminary results from a wildfire detection system using deep learning on remote camera images. *Remote Sensing*, 12(1):166, 2020.

[3] Chao-Ching Ho. Machine vision-based real-time early flame and smoke detection. *Measurement Science and Technology*, 20(4):045502, 2009.

[4] B Ugur Toreyin and A Enis Cetin. Wildfire detection using lms based active learning. In *2009 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASP-09)*, pages 1461–1464. IEEE, 2009.

[5] Angelo Genovese, Ruggero Donida Labati, Vincenzo Piuri, and Fabio Scotti. Wildfire smoke detection using computational intelligence techniques. In *2011 IEEE International Conference on Computational Intelligence for Measurement Systems and Applications (CIMSA) Proceedings*, pages 1–6. IEEE, 2011.

[6] Byoung Chul Ko, Joon-Young Kwak, and Jae-Yeal Nam. Wildfire smoke detection using temporospatial features and random forest classifiers. *Optical Engineering*, 51(1):017208, 2012.

[7] Yanmin Luo, Liang Zhao, Peizhong Liu, and Detian Huang. Fire smoke detection algorithm based on motion characteristic and convolutional neural networks. *Multimedia Tools and Applications*, 77(12):15075–15092, 2018.

[8] Mengxia Yin, Congyan Lang, Zun Li, Songhe Feng, and Tao Wang. Recurrent convolutional network for video-based smoke detection. *Multimedia Tools and Applications*, 78(1):237–256, 2019.

[9] Arun Singh Pundir and Balasubramanian Raman. Dual deep learning model for image based smoke detection. *Fire technology*, 55(6):2419–2442, 2019.

[10] Rui Ba, Chen Chen, Jing Yuan, Weiguo Song, and Siuming Lo. Smokenet: Satellite smoke scene detection using convolutional neural network with spatial and channel-wise attention. *Remote Sensing*, 11(14):1702, 2019.

[11] Tingting Li, Enting Zhao, Junguo Zhang, and Chunhe Hu. Detection of wildfire smoke images based on a densely dilated convolutional network. *Electronics*, 8(10):1131, 2019.

[12] Yichao Cao, Feng Yang, Qingfei Tang, and Xiaobo Lu. An attention enhanced bidirectional lstm for early forest fire smoke recognition. *IEEE Access*, 7:154732–154742, 2019.

[13] Minsoo Park, Dai Quoc Tran, Daekyo Jung, Seunghee Park, et al. Wildfire-detection method using densenet and cyclegan data augmentation-based remote camera imagery. *Remote Sensing*, 12(22):3715, 2020.

[14] Salman Khan, Khan Muhammad, Tanveer Hussain, Javier Del Ser, Fabio Cuzzolin, Siddhartha Bhattacharyya, Zahid Akhtar, and Victor Hugo C de Albuquerque. Deepsmoke: Deep learning model for smoke detection and segmentation in outdoor environments. *Expert Systems with Applications*, 182:115125, 2021.

[15] Jie Yuan, Lidong Wang, Peng Wu, Chao Gao, and Lingqing Sun. Detection of wildfires along transmission lines using deep time and space features. *Pattern Recognition and Image Analysis*, 28(4):805–812, 2018.

[16] Qi-xing Zhang, Gao-hua Lin, Yong-ming Zhang, Gao Xu, and Jin-jun Wang. Wildland forest fire smoke detection based on faster r-cnn using synthetic smoke images. *Procedia Engineering*, 211:441–446, 2018.

[17] Xiuqing Li, Zhenxue Chen, QM Jonathan Wu, and Chengyun Liu. 3d parallel fully convolutional networks for real-time video wildfire smoke detection. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(1):89–103, 2018.

[18] Mira Jeong, MinJi Park, Jaeyeal Nam, and Byoung Chul Ko. Light-weight student lstm for real-time wildfire smoke detection. *Sensors*, 20(19):5508, 2020.

[19] Parul Jindal, Himanshu Gupta, Nikhil Pachauri, Varun Sharma, and Om Prakash Verma. Real-time wildfire detection via image-based deep learning algorithm. In *Soft Computing: Theories and Applications*, pages 539–550. Springer, 2021.

[20] Taanya Gupta, Hengyue Liu, and Bir Bhanu. Early wildfire smoke detection in videos. In *2020 25th International Conference on Pattern Recognition (ICPR-20)*, pages 8523–8530. IEEE, 2021.

[21] Feiniu Yuan, Lin Zhang, Xue Xia, Boyang Wan, Qinghua Huang, and Xuelong Li. Deep smoke segmentation. *Neurocomputing*, 357:248–260, 2019.

[22] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems (NeurIPS-15)*, 28:91–99, 2015.

[23] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems (NeurIPS-12*, pages 1097–1105, 2012.

[24] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[25] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR-21)*, 2021.

[26] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR-09)*, pages 248–255. IEEE, 2009.

[27] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machineskinshuk govil, morgan l welch, j timothy ball, and carlton r pennypacker. preliminary results from a wildfire detection system using deep learning on remote camera images. remote sensing, 12(1):166, 2020. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR-16)*, pages 4724–4732, 2016.

[28] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV-19)*, pages 1314–1324, 2019.

[29] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR-17)*, pages 2117–2125, 2017.

[30] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning (ICML-19)*, pages 6105–6114. PMLR, 2019.

[31] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning (ICML-21)*, pages 10347–10357. PMLR, 2021.

[32] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR-16)*, pages 770–778, 2016.

[33] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV-17)*, pages 2961–2969, 2017.

[34] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR-18)*, pages 6450–6459, 2018.

[35] Hongyi Pan, Diaa Badawi, and Ahmet Enis Cetin. Fourier domain pruning of mobilenet-v2 with application to video based wildfire detection. In *2020 25th International Conference on Pattern Recognition (ICPR-20)*, pages 1015–1022. IEEE, 2021.

[36] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV-17)*, pages 2980–2988, 2017.

[37] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European Conference on Computer Vision (ECCV-16)*, pages 21–37. Springer, 2016.

[38] Olivier Barnich and Marc Van Droogenbroeck. Vibe: A universal background subtraction algorithm for video sequences. *IEEE Transactions on Image processing*, 20(6):1709–1724, 2010.

[39] Gunnar Farnebäck. Two-frame motion estimation based on polynomial expansion. In *Scandinavian Conference on Image Analysis*, pages 363–370. Springer, 2003.

6 Appendix

6.1 Data Pre-Processing

Model	# Fires	# Images
Train	144	11.3K
Validation	64	4.9K
Test	62	4.9K
Omitted	45	3.7K
Total	315	24.8K

Table 2: Number of fires and images in the training, validation, and test splits of the FIgLib dataset. "Omitted" fires include fires with black & white images, night fires, and fires with with questionable presence of smoke to avoid out-of-distribution sequences.

Additional transformations during data loading improve the performance of our model. We resize the images to 1392x1856 pixels to improve training and inference speed. We also crop the top 352 pixels of the image to reduce false positives from clouds for additional performance gains. Data augmentations include horizontal flip, random vertical crop, color jitter, brightness & contrast jitter, and blur jitter. Finally, images are normalized to 0.5 mean and 0.5 standard deviation as expected by the deep learning package we used (PyTorch).

One challenge of the dataset is that 1,213 (approximately 20%) of the positive images are missing contour mask annotations. 280 annotations are missing because the smoke is difficult to see, generally occurring at the beginning of the fire sequence or at the end, when the smoke has dissipated. 486 annotations are missing contour masks but have bounding box annotations, generally because the smoke is too small to reasonably outline a fine contour mask. The remaining 447 missing annotations are randomly spread throughout the fires.

For images with bounding box annotations where contour masks are not available, we determined the tile labels by filling the bounding boxes as polygons instead of the contour masks. We attempted other methods to incorporate feedback from images with missing annotations, including using feedback from only image labels (as opposed to both image and tile labels) and copying contour masks from the closest available image in the sequence. However, neither of these methods improved model performance. For future work, we aim to resolve these missing labels for more robust training data.

6.2 Training Details

Hyperparameter tuning was performed over learning rate, architecture, backbone, data augmentation, dropout, weight decay, smoke detection threshold, resized dimensions, cropped height, and BCE loss positive weights. Final models were trained using an SGD optimizer with learning rate 0.001 and weight decay 0.001. The batch size used was the larger of 2 or 4 depending on which would fit on GPU memory and gradient batches were accumulated such that the effective batch size was 32. Models were trained for 25 epochs using a single NVIDIA 2080Ti GPU; the model with the lowest validation loss was used for evaluation on the test set.

For baseline models that do not use a ViT as the last architectural component (e.g. ResNet34+LSTM, ResNet50, ResNet34+ResNet18-3D, etc.), we determine the overall image prediction by passing the model's *tile* predictions into a single fully connected layer with sigmoid activation, outputting a single prediction for the image. We also experimented with the simple decision rule that if the prediction for any tile is positive for smoke, the full image is also classified as positive; however, this resulted in worse performance. Image predictions for object detection models (e.g. FasterRCNN, MaskRCNN) were determined as positive if the model predicted any bounding box with a confidence score above the empirically-tested threshold of 0.5.

6.3 Human Performance Baseline

Due to the lack of suitable benchmarks for performance, we measured human performance of smoke classification on the FIgLib dataset. Participants were three lab members experienced in classifying

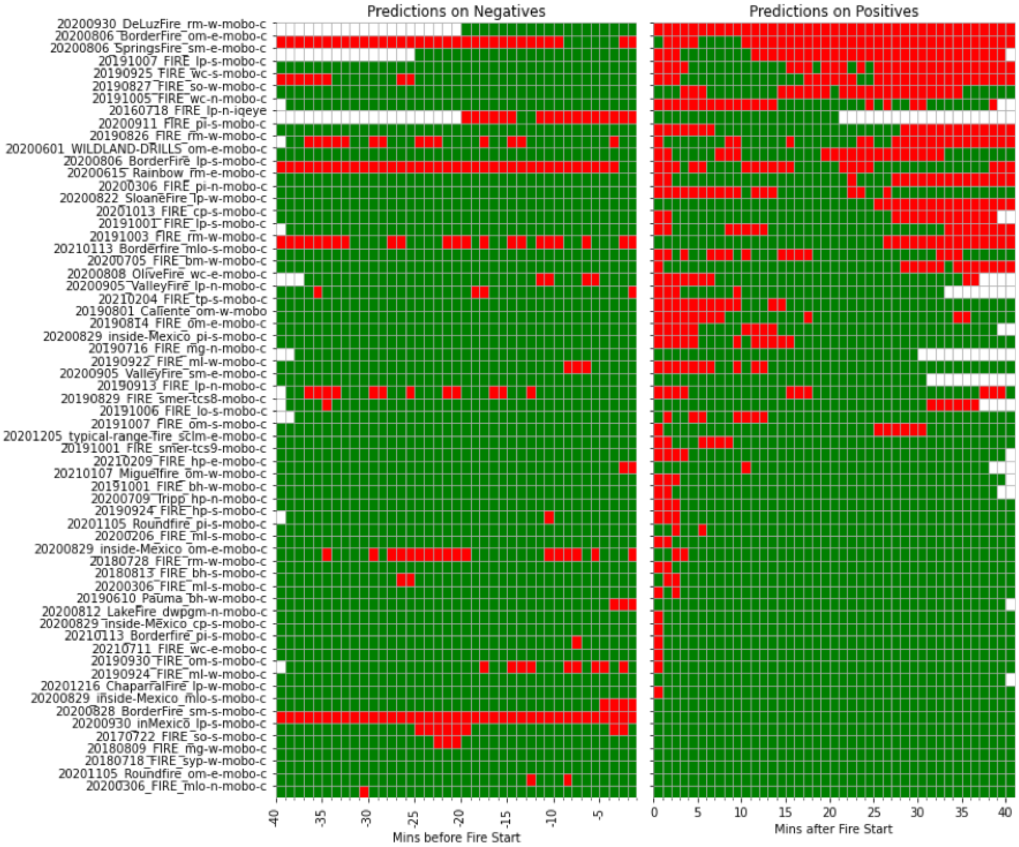

Figure 2: SmokeyNet's performance per fire on both negative and positive images. Green denotes a correct prediction; red denotes an incorrect prediction; white denotes images missing from the sequence. Common false negatives include faint smoke occurring at the start of the fire or dissipating smoke at the end of the fire sequence. Common false positives include low-altitude clouds and haze.

images for the presence of wildfire smoke. For the experimental setup, 44 fires (50 minus 6 that were omitted due to erroneous labels following further data cleaning) were randomly selected from the test set and one image among each of the fires was randomly selected for prediction. Participants were presented with the images for prediction, each preceded by the previous frame of the image sequence to replicate the temporal context our machine learning model also receives. The participants then recorded if they believed wildfire smoke was present in the image. The average accuracy of the three participants was 85.6% (σ = 2.83%), slightly higher than that of SmokeyNet. Resolving all systematic false positives from low-altitude clouds would enable SmokeyNet to achieve 85.8% accuracy and surpass human performance.