Computer Vision Aided Fire Localization for Wildfire Monitoring

Laura Spector
Department of Genetics
lspec@stanford.edu

Abstract

In this work I investigate the localization of fires in wildfire images by classification of image superpixels. Building on the performance of previous deep convolutional neural networks developed for fire localization, I re-purpose these models and fine tune them for classification of a distinct subtype of fire: wildfires. This work achieves maximal accuracy of 93% and F1 score of 0.84 for classification of fire-containing superpixels, leading to an average intersection-over-union score of 0.71 (SD = 0.2) for localized fires. These metrics outperform the baseline model on the wildfire data set, suggesting that wildfires indeed have unique features–such as variation in color, shape, texture (flame and/or smoke), and especially background–that require additional consideration in order to achieve accurate localization.

1. Introduction

Every year since 2000, an average of 72,400 wildfires burned an average of 7.0 million acres [6], resulting in substantial environmental, economic, and social losses. Computer vision offers a promising strategy for early wildfire localization. The increased use of general video surveillance and deployment of unmanned aerial vehicles for fire detection and monitoring provide ready methods to capture images and videos. However, due to the scarcity of wildfire image and video databases, which are more costly to obtain than those of fires in controlled or urban environments, a viable option for training small wildfire datasets is needed. Here, I present a transfer learning approach that improves our ability to specifically localize wildfires. It takes as input isolated superpixels that are pre-segmented from images of wildfires. I fine tune a reduced Inception v1 convolutional neural network to classify wildfire superpixels as fire or non-fire. The positively predicted (fire) superpixels for each image are then reconstructed to localize the predicted fire within the original image space. Finally, I generate saliency maps to visualize the spatial support of a particular class in a given image.

2. Related Work

2.1. Fire localization

Several groups have attempted pixel-level fire identification for the purpose of non-temporal fire localization (i.e., not relying on measurements of dynamic flame movement) using deep neural networks, either end-to-end or for classification of image patches or superpixels pre-processed from fire images. Chenebert et al. [3] isolated candidate fire regions using a basic color spectroscopy approach, which were then classified using a neural network with 88.1% accuracy (architecture not provided). Zhao et al. [13] achieved 98% accuracy by using a Bayesian saliency detection method plus logisitic regression classifier to first segment out core fire areas in images from unmanned aerial vehicles then classify them as fire/nonfire using a CNN based on AlexNet. Akhloufi et al. [2] achieved average F1 score of 0.91 and accuracy of 93.17% using a U-Net-based architecture and Soerensen-Dice similarity coefficient as a loss function to segment forest fires out from background. Zhang et al. [12] used a cascading CNN to both classify whole video frames as fire/nonfire and identify fire-containing patches to localize the fires, with 90% accuracy. Dunnings and Breckon [4] achieved 89% accuracy in classifying superpixels that were pre-segmented from urban fire images for in-frame localization, using a reduced complexity CNN based on Inception v1.

2.2. Superpixel segmentation

Superpixel segmentation is controlled over-segmentation of an image into a pre-defined number of fairly homogeneous regions. It is an important complement to computer vision. Superpixels are advantageous in that they group pixels into perceptually meaningful regions that replace the grids used in sliding window approaches, reducing image patch redundancy and thus the complexity of subsequent image processing tasks [5]. Meanwhile, superpixels together maintain all the characteristics of the image since the contours are defined by algorithms such as K-means that group pixels using a distance metric based on color and texture (see Fig. 3, step 2 for an example). In par-

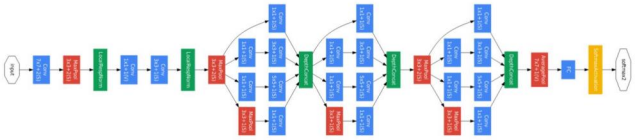

Figure 1. InceptionV1-OnFire

ticular, superpixels are useful for remote sensing images–a primary method by which wildfires are sensed and tracked–for which the target area may be very small relative to the frame. SLIC superpixel segmentation (described in 4.2) [1] combined with CNN-based classification achieved similar classification accuracy but dramatically reduced compute time compared to the sliding window approach on the ISPRS remote sensing semantic labeling benchmark [5].

3. Dataset and Features

The collection of wildfire images was downloaded with permission from the Corsican Fire Database (http://cfdb.univ-corse.fr) [11], and comprises 595 images each paired with a binary, ground truth, same-sized image (Fig. 2B).

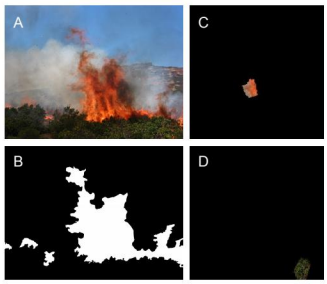

Figure 2. Examples of original images and input. A) original image, B) ground truth image, C) fire superpixel (input), D) non-fire superpixel (input).

3.1. Data preprocessing

The workflow is illustrated in Fig. 3. The images are shuffled and split 7:2:1 into train/dev/test sets, resized to 224x224 pixels (but not normalized, to be consistent with the pre-training input) and segmented into 100 superpixels, resulting in 41,594/11,899/6,000 isolated superpixels in the train/val/test sets, respectively. Each superpixel is isolated in a new 224x224x3 image in which only pixels contained within the superpixel region retain their RGB values; all other pixels are set at (0,0,0). Instead of using bounding boxes like [4], each isolated superpixel image is labeled "fire" if $\geq$ 25% of the superpixel overlaps with the originating image's ground truth pair. Since splitting is done prior to superpixel segmentation, it ensures that both classes are equally represented in all splits. Indeed, about 20% of each set are fire superpixels and the rest are non-fire. The non-fire superpixels contain a mix of typical forest elements including trees, grass, and sky.

3.2. Data augmentation

Training set images were augmented on the fly where indicated by adding random blur using a Gaussian filter with sigma $= 5$ for the Gaussian kernel and allowing randomly for up to three 90-degree rotations. These seemed the most logical choice for the expected test set distribution.

4. Methods

4.1. Model

A pre-trained, reduced Inception v1 [10] model was adopted from Dunnings and Breckon [4] (Fig. 1, described in 2.1). It contains three rather than the usual nine inception modules, allowing it to (1) retain the same level of accuracy on the dataset used for pre-training (urban fires) as the original architecture while reducing the number of parameters, and (2) reduce the potential for overfitting by reducing architecture complexity. Dropout with `keep_prob` $= 0.4$ is applied after the average pooling filter and before the two-node fully connected layer. The output activation is a softmax function, which computes the normalized probabilities across classes, as this is a single-label classification task.

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \tag{1}$$

4.2. SLIC superpixel segmentation

Simple linear iterative clustering (SLIC) [1] was used to segment images into regions that are similar in color and texture. It adapts K-means clustering to reduced spatial dimensions for computational efficiency. The distance measure D combines color proximity (d_c, in CIELAB color space) and spatial proximity (d_s) and normalizes each by its respective maximum within a cluster (N_s and N_d).

$$\begin{aligned} d_c &= \sqrt{(l_j - l_i)^2 + (a_j - a_i)^2 + (b_j - b_i)^2} \\ d_s &= \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2} \\ D' &= \sqrt{\left(\frac{d_c}{N_c}\right)^2 + \left(\frac{d_s}{N_s}\right)^2} \end{aligned} \tag{2}$$

4.3. Loss function

The categorical cross entropy loss was used

$$L = -\sum_i y_i log\hat{y}_i + (1 - y_i) log(1 - \hat{y}_i) \tag{3}$$

4.4. Metrics

The primary metrics were accuracy and F1 score. F1 score is a harmonic average of precision and recall that better reflects a model's performance when the classes are not

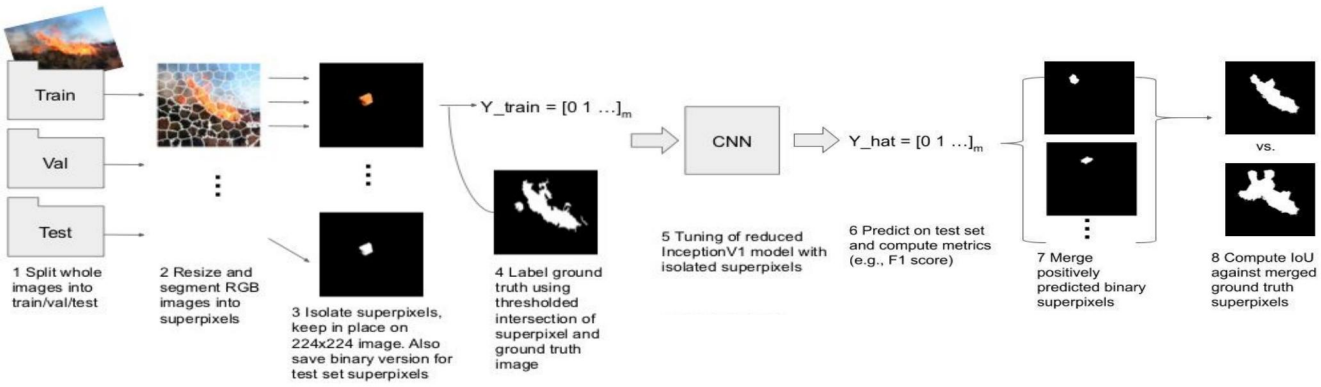

Figure 3. Experimental workflow

balanced.

$$\text{precision} = \frac{\text{tp}}{\text{tp} + \text{fp}} \qquad \text{recall} = \frac{\text{tp}}{\text{tp} + \text{fn}}$$

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{4}$$

Intersection-over-Union (IoU), or Jaccard index, is typically a measure of similarity between predicted and ground truth bounding boxes in object detection. Instead of bounding boxes, I use merged superpixels to measure how well a fire can be localized. It is a more stringent method, but wildfires in particular can rarely be well defined by a rectangle.

$$\text{IoU} = \frac{A \cap B}{A \cup B} \tag{5}$$

4.5. Class saliency

This method ranks the pixels of an image I_0 based on their influence on the class score $S_c(I)$ [9], which can be approximated by

$$S_c(I) \approx w^T I + b$$

where w is the derivative of S_c with respect to the image I at the point (image) I_0:

$$w = \frac{\partial S_c}{\partial I}\bigg|_{I_0} \tag{6}$$

The map M is generated at each pixel (i, j) by taking the maximum magnitude of w across all color channels

$$M_{ij} = \max_c |w_{h(i,j,c)}| \tag{7}$$

5. Experiments/Results/Discussion

5.1. Model fine tuning

Evaluating the test set on the original model resulted in a poor F1 and IoU score (Table 1), with misclassification of superpixels containing thin and wispy fires, suggesting that it could be improved by fine tuning of later layers, which contain lower level features. The first two inception modules were frozen, allowing updates to only the third inception module during backpropagation. The pre-trained model was fine tuned by loading the pre-trained weights to all except the softmax layer and training for at least 100 epochs. I tested batch sizes of 64 and 256. While smaller batch sizes tend to be noisier and have a slight regularization effect, larger batch sizes allow for a more stable model and in this case do not really slow down learning even though there are fewer updates per epoch.

The Adam optimization algorithm [7] was tested because it might allow the model to converge more quickly. Adam optimization implements adaptive learning rates for different parameters by storing both the exponentially decaying average of past squared gradients (like RMSProp) and past gradients (like momentum). I reasoned that, in general, learning rate adaptation or decay–generally, a reduced learning rate–might be useful as the model already had some familiarity with fire as input. I tuned the learning rate to 0.001, which was originally used for training the model and worked well; trials with $\alpha = 0.01$ did not converge. The model converged after about 70 epochs, however it clearly overfit the data, as training accuracy was 99% (not shown) while validation accuracy was only 93% (Table 1) and validation loss increased after 30 epochs (Fig. 4). Early stopping allowed me to extract the weights before the model started to overfit.

The momentum optimization algorithm [8] was also tested, in part because it was successfully used to train the original model. I implemented learning rate decay of $\alpha = \alpha \cdot x^{(\text{total steps}/100)}$, where $x = 0.97$ or $x = 0.993$ and $\alpha = 0.01$. For $x = 0.97$, learning appeared to converge, but at a higher loss and lower training accuracy (94%) than the Adam-optimized model after 100 epochs (Fig. 4); decreasing the decay rate with $x = 0.993$ allowed the model to reach a lower training loss and training accuracy of 97%

but it started to overfit the validation set after 100 epochs and validation accuracy did not improve (not shown).

For all successful fine-tuned models, the F1 score of 0.83-0.84 for predicting fire on the test set was a marked improvement over that of 0.70 obtained from the original, un-tuned model (Table 1, also see Fig. 5A). I consider F1 score to be a better reflection of performance because the proportion of fire to nonfire superpixels was 1:4, not 1:1.

Model	Accuracy	F1	IoU
Original pre-trained	0.90	0.70	0.49
Adam	0.93	0.84	0.71
Adam w/data aug.	0.93	0.83	0.69
Momentum w/α decay	0.93	0.84	0.70

Table 1. Metrics evaluated on the models for the wildfire test set

I reasoned that one technique to reduce overfitting is data augmentation. I fine tuned the original model with an Adam optimizer, augmenting images in the training set as described in section 3.2. Since the original model was pre-trained on fire-type images, I anticipated the model might at first perform better on the validation than the training set, as the former was not augmented. Based on the loss curves, this model was no longer overfitting (Fig. 4; 100 epochs shown; loss did not decrease if $\alpha > 0.001$). After 300 epochs the model had not converged and training accuracy was 90%, but validation accuracy was not improving beyond 93%. It appears that the model is slower to train but augmentation has mitigated overfitting. Manual learning rate decay might be a good option to improve training.

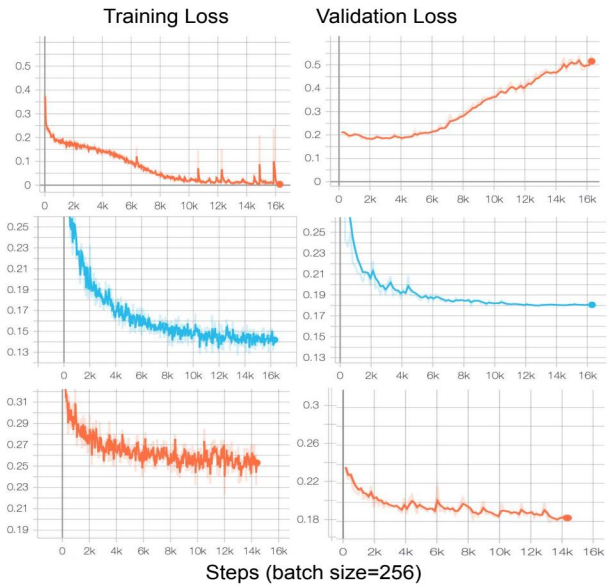

Figure 4. Loss curves over 100 epochs. Top to bottom: adam, momentum w/learning rate decay, adam w/data augmentation. Note the difference in y-axis scales.

5.2. Fire reconstruction and localization

After evaluating the fine tuned model on the test set, I used the predictions to reconstruct and localize the predicted fire within each image by merging positively predicted superpixels (binary version). I then computed the IoU score relative to the merged ground truth positive binary superpixels using bitwise operators (Fig. 3 and 6). The average IoU score computed after evaluating the wildfire test set on the fine tuned model was 0.71 (distribution in Fig. 5B), a marked improvement over the score of 0.49 obtained using the original, un-tuned model (Table 1).

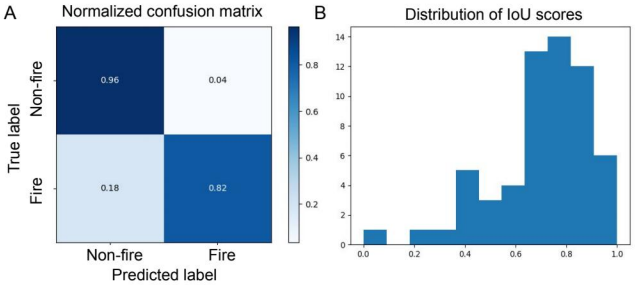

Figure 5. A) Representative normalized confusion matrix for isolated superpixel classification. B) Histogram showing distribution of IoU scores for reconstructed wildfires.

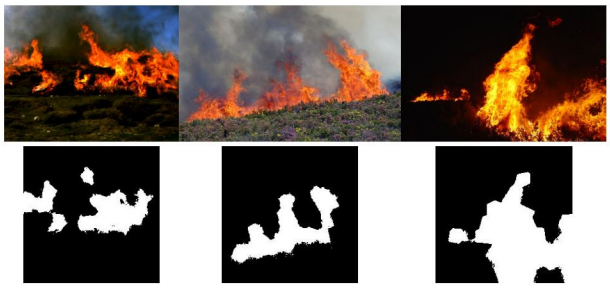

Figure 6. Examples of original images and fires reconstructed from classified superpixels (following image resizing).

5.3. Error analysis

I manually examined correctly and incorrectly classified superpixels and images with small and large IoU scores (Fig. 7). Incorrectly classified superpixels tended to contain colors similar to fire, like orange fire-glow or orange/yellow grass. This observation suggests that the model classifies fires in large part based on color and less on shapes, which makes sense given the lack of context provided in a superpixel. Images with low IoU scores (< 0.5) tended to have thin, spread out flames while those with high IoU scores (≥ 0.5) were of spatially concentrated fires. One benefit of the pre-segmentation method appears to be the ability to reconstruct distinct fires within the same frame. U-Net has also proven its ability to identify distinct fires on the same data set used in this project [2].

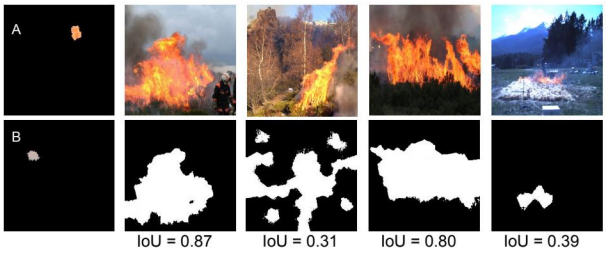

Figure 7. Error analysis. A) correctly classified, B) incorrectly classified. Resized original images and reconstructed classified superpixels.

5.4. Saliency maps

I generated saliency maps [9] to gain some intuition as to which pixels need to be changed the least to have the greatest effect on class prediction. In all cases, the network has identified the superpixel as the only relevant information-containing region of the input, characterized by the absence of signal everywhere else. Interestingly, saliency maps show larger magnitudes on the edge and interior of superpixels where there are not fire pixels (Fig. 8). One hypothesis is that fire is fairly easy to recognize, whether yellow, orange, or red, so the magnitude of fire pixels is less. Greens and light browns (the latter of which are sometimes misclassified) are less easy to predict but are close in RGB value. Consequently, pixels in those colors are more active in contributing to class prediction.

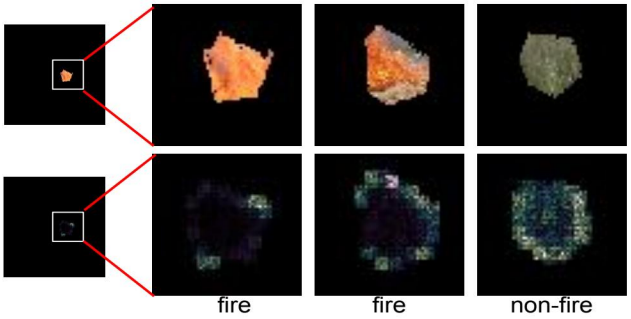

Figure 8. Magnified input images (top) and class saliency maps showing magnitude of the saliency (bottom).

5.5. Local wildfire images

I downloaded a number of public domain images from recent wildfires in California, Colorado, and Oregon to test the localization method developed in this project. From the examples in Fig. 9, we can see that the model correctly predicts fire regions while excluding smoke, but does classify areas of orange fire-glow as having fire as might be expected due to the lack of context, discussed in 5.3.

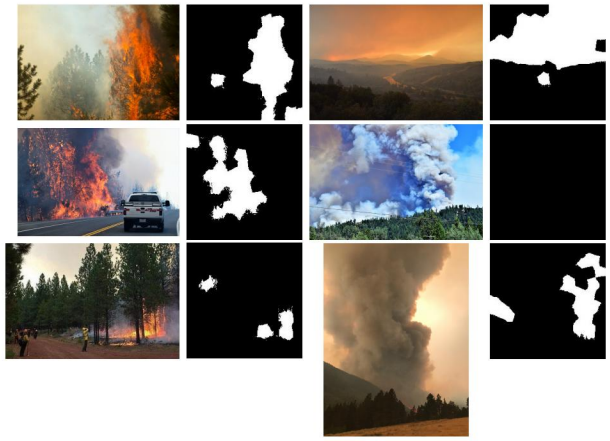

Figure 9. Examples from recent California, Colorado, and Oregon fires including the Woosley Fire, Camp Fire, and Carr Fire. Original images (left); resized and reconstructed superpixels (right).

6. Conclusion/Future Work

The fine tuned model shows a marked improvement over the original model that was pre-trained on urban-type fire superpixels, suggesting that wildfire superpixels have unique features requiring additional training. The lack of context in close-up images where superpixels may contain entirely fire appears to influence misclassification errors. In the future, training the model on remote sensing images would allow the superpixel approach to prove its efficiency and provide a meaningful benchmark for the types of images used to assist Forest Service personnel in tracking fires.

To improve the specific model used in this project, additional regularization methods could be pursued to reduce overfitting without requiring early stopping, hopefully allowing validation and test accuracy to exceed the current ceiling of 93%, such as reducing `keep_prob` when applying dropout. Changing the threshold for ground truth labeling might affect model performance. A more extensive random but informed search for hyperparameters could be attempted; however, since the model was pre-trained, the viable search space was already reduced. Finally, it would be feasible to unfreeze additional inception modules for training given the size of the training set used here.

7. Code and Dependencies

Code: https://github.com/laurapspector/wildfire-localization-cnn. Dependencies: tensorflow, tflearn, fire-detection-cnn, opencv-contrib-python, saliency, tensorpack.

8. Contributions

L.S. conceived of and conducted the experiments. Many thanks to the CS230 teaching staff for their support.

References

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Ssstrunk. SLIC Superpixels Compared to State-of-the-Art Superpixel Methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2274–2282, Nov. 2012. 2

[2] M. A. Akhloufi, R. B. Tokime, and H. Elassady. Wildland fires detection and segmentation using deep learning. In *Pattern Recognition and Tracking XXIX*, volume 10649, page 106490B. International Society for Optics and Photonics, Apr. 2018. 1, 4

[3] A. Chenebert, T. P. Breckon, and A. Gaszczak. A non-temporal texture driven approach to real-time fire detection. In *2011 18th IEEE International Conference on Image Processing*, pages 1741–1744, Brussels, Belgium, Sept. 2011. IEEE. 1

[4] A. J. Dunnings and T. P. Breckon. Experimentally Defined Convolutional Neural Network Architecture Variants for Non-Temporal Real-Time Fire Detection. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 1558–1562, Athens, Oct. 2018. IEEE. 1, 2

[5] C. Gonzalo-Martin, A. Garcia-Pedrero, M. Lillo-Saavedra, and E. Menasalvas. Deep learning for superpixel-based classification of remote sensing images. In *GEOBIA 2016 : Solutions and Synergies*, University of Twente Faculty of Geo-Information and Earth Observation (ITC), Sept. 2016. University of Twente Faculty of Geo-Information and Earth Observation (ITC). 1, 2

[6] K. Hoover and L. A. Hanson. Wildfire Statistics. page 2, May 2019. 1

[7] D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. *arXiv:1412.6980 [cs]*, Dec. 2014. arXiv: 1412.6980. 3

[8] N. Qian. On the momentum term in gradient descent learning algorithms. *Neural Networks*, 12(1):145–151, Jan. 1999. 3

[9] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. *arXiv:1312.6034 [cs]*, Dec. 2013. arXiv: 1312.6034. 3, 5

[10] C. Szegedy, Wei Liu, Yangqing Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, June 2015. 2

[11] T. Toulouse, L. Rossi, A. Campana, T. Celik, and M. A. Akhloufi. Computer vision for wildfire research: An evolving image dataset for processing and analysis. *Fire Safety Journal*, 92:188–194, Sept. 2017. 2

[12] Q. Zhang, J. Xu, L. Xu, and H. Guo. Deep Convolutional Neural Networks for Forest Fire Detection. Atlantis Press, Jan. 2016. 1

[13] Y. Zhao, J. Ma, X. Li, and J. Zhang. Saliency Detection and Deep Learning-Based Wildfire Identification in UAV Imagery. *Sensors (Basel, Switzerland)*, 18(3), Feb. 2018. 1