INL/CON-21-61720-Revision-0

DETECTING FIRE WITH MACHINE LEARNING-ENABLED VISUAL MONITORING FOR NUCLEAR POWER PLANT ENVIRONMENTS

June 2021

Changing the World's Energy Future

Mike Griffel, Ahmad Y Al Rashdan, Roger Boza

Idaho National Laboratory

INL is a U.S. Department of Energy National Laboratory operated by Battelle Energy Alliance, LLC

INL/CON-21-61720-Revision-0

DETECTING FIRE WITH MACHINE LEARNING-ENABLED VISUAL MONITORING FOR NUCLEAR POWER PLANT ENVIRONMENTS

Mike Griffel, Ahmad Y Al Rashdan, Roger Boza

June 2021

Idaho National Laboratory
Idaho Falls, Idaho 83415

http://www.inl.gov

Prepared for the
U.S. Department of Energy
Under DOE Idaho Operations Office
Contract DE-AC07-05ID14517

DETECTING FIRE WITH MACHINE LEARNING-ENABLED VISUAL MONITORING FOR NUCLEAR POWER PLANT ENVIRONMENTS

L. Michael Griffel*[1], Ahmad Al Rashdan[1], and Roger Boza[1]
Idaho National Laboratory
P.O. Box 1625, MS 3570, Idaho Falls, ID, 83415 USA
Mike.Griffel@inl.gov; Ahmad.AlRashdan@inl.gov; Roger.Boza@inl.gov

[Digital Object Identifier (DOI) placeholder]

ABSTRACT

Nuclear power plants are experiencing significant cost challenges to remain competitive with other energy-generation utilities. Unlike other industries, the cost of operation and maintenance activities is mostly attributed to workforce costs. To mitigate this, nuclear power plant stakeholders are increasingly interested in the development and deployment of machine learning methods to potentially automate or augment manually intensive tasks to reduce O&M costs, especially for monitoring activities. One monitoring function that is visually demanding and that can frequently occur to meet the requirements of a fire protection program is visually monitoring an area for fire occurrence. Currently, fire watch activities consist of a worker physically stationed at a given location with the sole responsibility of observing a given area to ensure a fire is detected and mitigated promptly.

This effort focused on the development and evaluation of a suitable deep convolutional neural network to classify individual video frames at a sub-second frequency for the occurrence of “fire” and “no fire” in varying industrial environments similar to nuclear power plants. It is believed that a trained network architecture could be integrated with existing facility video surveillance camera networks to generate alerts when “flame” inferences occur in individual frames captured at sub-second temporal resolutions. Extensive effort was dedicated to identifying and curating suitable imagery training data representing varying environments and scene settings with and without flame features to maximize generalization in nuclear power plant environments. The data collection effort resulted in the aggregation of a large, labeled image library exceeding 12,000 images to support model training for diverse industrial environments.

A deep network architecture incorporating parallel multi-scale capabilities was developed and trained to support accurate image-based detection of flame incidents of varying size and spectral feature properties within heterogeneous scenes with low false positive incidents. Analysis results show that the trained model architecture is capable of achieving high inference accuracy despite heterogeneous scene environments and components. Testing accuracy exceeded 95 percent with very low false positive and false negative inferences.

Key Words: fire watch, nuclear power plant, machine learning, computer vision, convolutional neural networks

1 INTRODUCTION

Nuclear power plants (NPPs) are experiencing a significant cost challenge to remain competitive with other energy-generation industries. Unlike other industries, the cost of operations and maintenance (O&M) activities at an NPP is mostly attributed to workforce cost. The NPP industry has therefore resorted to automation to reduce O&M costs, especially for monitoring manually intensive tasks. One monitoring function that is visually demanding and can be required frequently in an NPP is fire watch.

A fire watch ensures fire is detected and mitigated in time. The U.S. Nuclear Regulatory Commission (NRC) Regulatory Guide 1.189 defines the fire watch as "Individuals responsible for providing additional (e.g., during hot work) or compensatory (e.g., for system impairments) coverage of plant activities or areas to detect fires or to identify activities and conditions that present a potential fire hazard. The individuals should be trained in identifying conditions or activities that present potential fire hazards, as well as in the use of fire extinguishers and the proper fire notification procedures" [1]. A person conducting a fire watch could spend hours watching a certain equipment, room, or environment.

Throughout the nuclear power industry level of fire watch performed varies depending on the condition of the plant and plant-safety policies and requirements. If the plant is in a condition that results in frequent fire-watch needs, then the cost burden may be elevated. Because of advancements in sensor technology and increased video and image data capture over recent years, there is a greater potential for automating detection of fire or smoke in real time using remote sensing and robust trained algorithms.

A significant amount of literature is available on detecting fire in video imagery. From a high-level perspective, many approaches incorporate multistage methods using color detection and differentiation, image-change detection, image-edge softening (due to the presence of smoke), feature extraction, and either a rules-based segmentation approach or a classification model, such as a support vector machine (SVM) [2], artificial neural networks (ANNs) [3], etc. Çetin et al. [4] provide a comprehensive review of recent efforts in video-based fire detection (VFD). Many VFD algorithms address specific problems, each with its own characteristics, as opposed to a more generalized problem resulting in a broad solution. Specific engineering-based criteria and parameters, including whether single or multiple sensors are utilized, whether they are active or passive instruments, and their spatial, spectral, and temporal resolutions, influence algorithm choice and performance. Although it is possible to model fire behavior in video imagery using varying methods, many current systems can yield false positives or alarms, resulting from changing light conditions, shadows, and other image features. VFD methods reviewed in this section can be subsumed into the following sections: fire detection using either image-processing or advanced machine-learning methods. This review is intended to capture high-level insights from some recent research in both areas.

1.1 Image Processing Methods

Many VFD techniques focus on fire color and shape characteristics. Although research in this area shows much promise, inconsistencies in the behavior of fire or smoke, background scenes, combustion sources, and sensors impede development of generalized solutions. Color detection (based on visible wavelengths relative to the human eye) was an early technique employed for fire detection and often makes use of sensor red, green, and blue (RGB) channels. These channels are often included in modern sensors. Visual sensors are also less costly and more widely available relative to spectrally higher-resolution sensors. Rules-based methods can exploit a known relationship among RGB values where pixel values of visible channels comprised of flames often exhibit a consistent relationship of $R > G > B$. Smoke-based pixels often exhibit RGB values that are close to each other [4].

Çelik & Demirel, [5] tested for a set of expected spectral relationships between channels derived from the RGB image, including luminance, the distance of the luminance value from red, the distance of the luminance value from blue, and saturation for each pixel. If these conditions are met, the pixel is considered a fire pixel. A similar approach is used by Thou-Ho et al. [6] using a different rules-based detection in the RGB feature space.

Marbach et al. [7] utilized spectral relationships of luminance and chrominance derived from RGB channels coupled with temporal variation occurring in image sequences to identify candidate regions within scenes and classify as fire or non-fire based on the persistence of potential fire features.

Motion and flicker analysis have also been of interest relative to VFD as a means of reducing error that can occur from dependence upon color signatures alone. Fire flicker has been found to occur near 10 Hz and is not greatly affected by the combustion material [8]. Good results have been shown when attempting to detect flame in video-image data at a frame rate of 24 frames-per-second (fps) by combining a Gaussian mixture model to extract moving foreground objects from a still background and categorizing moving objects into flame or non-flame regions using a color-filtering algorithm [9].

Although image processing techniques show much promise, multiple confounding factors impact classification accuracies and overall generalization capabilities. Industrial scenes such as those found in NPPs have the potential to include moving objects of varying size, shape, color, and movement speed, which introduce potential sources of error. Also, uncontrolled fires exhibit chaotic tendencies and nonlinear instabilities, making it difficult to classify based on flicker patterns alone. Motion-based approaches also represent a significant challenge in that models need to be developed for specific sensor frame rates, which can vary widely in existing systems.

1.2 Machine-Learning Methods

Recently, machine-learning methodologies based on ANNs have gained traction due to advancements and improvements in computing power and the increasing availability of training data and deep learning libraries such as TensorFlow [10] for common programming languages such as Python [11]. The advantage of machine learning over the previously referenced methods is that these methods do not have to be explicitly programmed with the differentiation rules, such as those needed for color feature extraction or motion detection. However, due to their supervised learning process, machine-learning methods require large amounts of diverse training data to achieve adequate generalization. In order to successfully develop a robust fire-detection system using machine learning, it is important to account for varying parameters impacting a classification model's development and biases. A few examples of machine-learning methods are listed here for demonstration.

Muhammad et al. [12] proposed a system using convolutional neural networks (CNNs) to predict fire occurrences in video-imagery data. Their experimental results indicate high classification accuracies, greater than 90%, are achievable. However, the authors acknowledged lower accuracies on image datasets dominated by red fire-colored objects and fire-like sunlight scenarios that yielded higher levels of false positives. Dunnings and Breckon [13] demonstrated high classification accuracies by coupling a super-pixel localization framework using the K-means clustering algorithm [14] to a simplified CNN architecture. However, parametrization of the K-means algorithm used to delineate super-pixel boundaries and scene variability could greatly impact classification accuracy and model generalization.

Researchers achieved high fire-detection accuracies in video-imagery data using deep convolutional long-recurrent networks (DCLRNs) combined with optical flow methods, indicating that temporal or sequential dynamics within the data can improve model accuracy [15]. Their analysis also indicated DCLRN outperformed a CNN when flame features were not as dominant in the images due to increasing distances between the camera and the target. However, model performance was negatively impacted by very slow-moving fires, especially at longer distances from the sensor, and dynamic image disturbances such as moving lights.

Traditional machine-learning classifiers are often developed using balanced datasets where the number of instances for each class are similar. However, when developing classifiers intended to detect rare occurrences such as fire, data among classes are often imbalanced. Sharma et. al [16] successfully implemented deep, pretrained CNN architectures enhanced with additional dense layers trained on imbalanced data favoring "no-fire" scenes.

1.3 Project Approach

Although image processing methods have demonstrated some success for the detection of flame features in imagery, it was decided to utilize machine learning for detection of fire in imagery data for this analysis. Algorithms such as CNNs represent the state-of-the-art in computer vision and offer advantages over traditional image processing methods including inference speed and the fact that they are "self-taught" in that they are capable of learning complex patterns not obvious to humans during training.

The overall project approach was divided into two tasks: (1) aggregate and curate suitable training and testing image data and (2) develop an image-based classification CNN fire watch model to determine whether or not flame features were present within an image.

2 MATERIALS AND METHODS

2.1 Data Collection

Unlike other approaches, in which models are developed on known and explicitly defined rules, machine-learning algorithms require large, heterogeneous, and labeled datasets for training and validation to maximize model accuracy and improve generalization. If suitable data are not available, they must be generated. In this case, it was not feasible to set up and film fires in varying industrial scenes approximating NPPs at a scale necessary for the development of a computer vision model. This, in turn, necessitated searching for publicly available datasets containing fire imagery and industrially relevant scenes. Sourcing, organizing, and formatting suitable datasets are complex and time-consuming endeavors. If data used are not suitable, resultant models will not have the ability to generalize a problem, which severely limits their real-world applications. During the literature review, multiple online repositories were discovered containing imagery data. Extensive efforts were made to aggregate suitable data consisting of varying environments (indoor, outdoor, low-light, daylight, artificial illuminance, etc.) and scene composition (people, equipment, vehicles, various activities, etc.) with and without flame features of varying size and shape in an effort to account for what could be encountered in industrial settings like that of an NPP.

The Yahoo Flickr Creative Commons 100 Million (YFCC100m) dataset was utilized as the primary source for image data [17]. YFCC100m was compiled from publicly available Yahoo's Flickr content and houses 100 million total image and video records (99.2 million and 0.8 million respectively) in native formats (i.e., .mp4 and .jpg) suitable for viewing. Although the dataset is too large to download in its entirety, an index table containing a complete list of the unique file IDs and associated user content tags supplied by the content provider at the time of upload was downloaded and parsed to identify suitable data.

Additionally, images were downloaded from The Fire Smoke (FiSmo) dataset, a compilation of images and videos taken from scenes showing emergency situations where fire, smoke, or explosions occur [18]. This dataset was compiled from online repositories including Yahoo's Flickr and Google's YouTube. The intended purpose is to support ongoing research to apply computational techniques to support video and image processing for emergency situations. Data supporting [16] were also publicly available and downloaded to support this modelling effort.

2.2 Model Development

CNNs are a type of ANN with fully connected layers designed to mimic the natural image-processing pathways found in humans. The CNN concept has seen significant utilization and development over the past several years as improvements in computer processing power have facilitated widespread use. At a base level, CNNs were designed to recognize important features in imagery data through a network of convolution steps producing modified inputs for a densely connected neural network trained to deliver a classification as an output. Like other neural networks, CNNs are considered a black-box learner in that no rules are explicitly determinative of the classification output. Instead, the model learns the mathematical relationships between the input features and output class during training.

An input image array comprised of red, green and blue color channels is passed into a series of convolutional moving-window filters (kernels). Features are extracted from the various kernel layers by Equation 1:

$$F_i = X * W_i + b_i \quad (1)$$

where X represents the input features, W_i, the shared weights, and b_i, a bias factor for the i^{th} filter kernel.

A nonlinear activation function encapsulates F_i to calculate the signal value passed to the next neuron. The rectified linear unit activation function (ReLU) [19] is commonly used as it typically leads to faster convergence, does not plateau or saturate in the positive direction, and it is sparsely activated as all negative inputs are converted to zero within the network. ReLU is shown in Equation 2:

$$f(x) = \begin{Bmatrix} 0, F_i < 0 \\ F_i, F_i \geq 0 \end{Bmatrix} \quad (2)$$

where $f(x)$ is always positive and is not bounded in the positive direction.

Additionally, batch normalization functions [20] are added to the network to mitigate a potential gradient diffusion problem that can occur within deep CNN structures. Batch normalization utilizes the mean and variance of batched input data to normalize the inputs for the subsequent layer. This is shown in Equation 3:

$$\hat{X}_i = \frac{X_i - \mu}{\sqrt{\sigma^2 + \in}} \quad (3)$$

where input X_i is normalized by subtracting μ, the batch mean and dividing by the σ^2, the batch variance plus $\in$, a small value to ensure that the denominator remains nonzero.

Softmax regression is used to determine the resultant class of the input features at the end of the network by calculating a normalized probability score for each possible class using an exponential function. The class with highest score is then assigned as the output. This is shown in equation 4:

$$\hat{P}_k = \frac{\exp\left(s_k(x)\right)}{\sum_{j=1}^{K} \exp\left(s_j(x)\right)} \quad (4)$$

where K is the number of classes and $s_k(x)$ is a vector containing the scores of each class for the input data for instance x. The output is normalized by dividing by the sum of all exponentials. Additionally, a categorical cross-entropy loss function is applied to support the model's learning progress resulting from back propagation.

For this analysis, the CNN architecture was assembled using TensorFlow 2.1 based on a modular, multi-scale structure introduced by Yuan et al. [21]. This was done to accommodate varying flame feature sizes occurring at varying distances from image sensors. Figure 1 shows the individual module architectures. The basic module accepts an input and passes it to three parallel convolutional filter steps each with a different kernel size of 1 x 1, 3 x 3, and 5 x 5 windows for ***n*** number of kernels. Those outputs are normalized individually and merged via concatenation. A ReLU activation function is applied to the concatenation output which is then passed out of the module. A second version of the module includes a pooling step that was not included in [21] that reduces data dimensionality and further extracts potential useful features. It is important to note that [21] used a summation method to merge the normalized convolution outputs to reduce data dimensionality. For this analysis, concatenation was used instead to preserve the input features while using pooling to reduce data dimensionality.

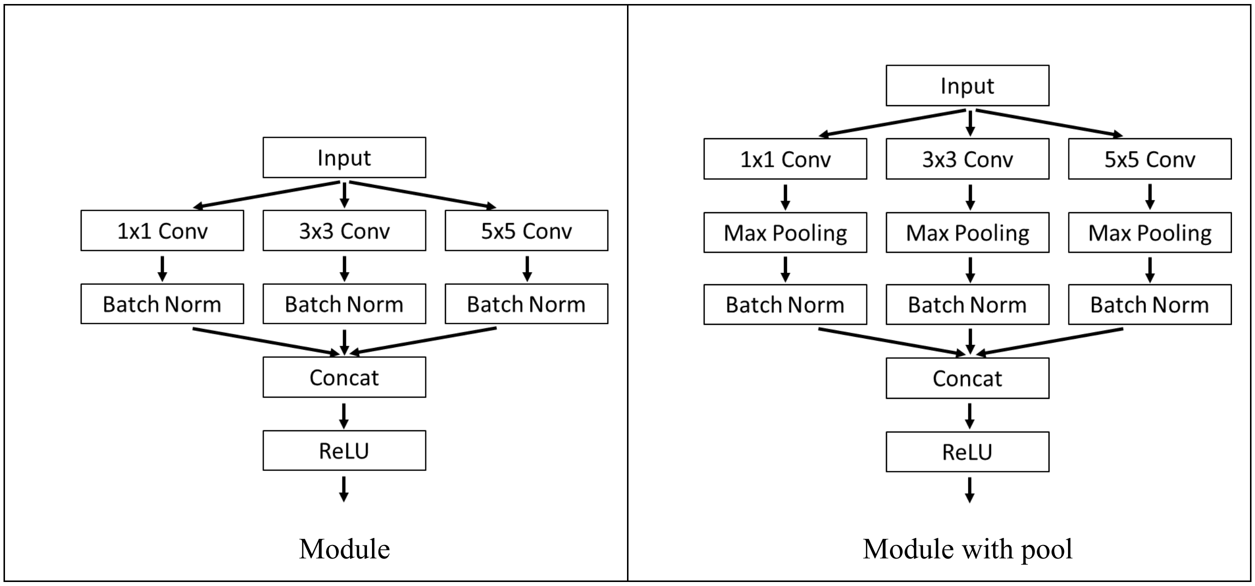

Figure 1. The module architectures used for CNN fire watch model. The basic module (left) includes parallel convolution steps of varying window sizes, batch normalization, and concatenation to merge the outputs, and ReLU activation function output. The module with pooling (right) includes a max pooling step in between the convolution and normalization steps.

The modules with varying numbers of convolutional kernel filters were assembled to form the deep CNN fire watch classification model shown in Figure 2. Although [21] included a convolution step between the image input and the first module, initial testing showed the classifier was more accurate by passing the input image directly to the first module. Starting with an image input size of 400 x 400 pixels with red, green, and blue channels, the data are passed sequentially to two modules each generating 16 convolutional layers in parallel based on the different window sizes. Next, the number of convolutional filters is increased by 32 with a pooling step followed by three modules each generating 32 convolutional filters in parallel. The number of parallel convolutional outputs are increased to 64 with a pooling module followed by two additional non-pooling modules yielding 64 parallel convolutional filters each. Next, convolutional filter outputs are increased to 128 with a pooling module followed by a non-pooling module with another 128 parallel convolutional filter outputs. At this point, global average pooling is applied to the resultant features which generates the average output of each feature map extracted during convolution and pooling steps which further reduces data dimensionality while extracting useful features for the final classification output that incorporates the binary softmax function to predict "flame" or "no flame" labels.

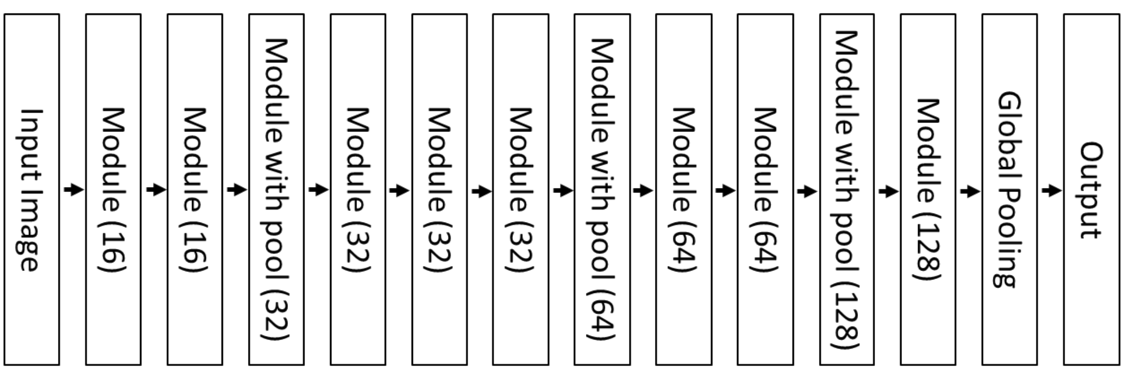

Figure 2. The fire watch deep CNN model architecture incorporating the modules shown in Figure 2.

The fire watch CNN was compiled and executed on Idaho National Laboratory's High Performance Computing Sawtooth environment. Each node consists of 4 NVIDIA V100 GPUs along with a total of 384 GB of RAM.

3 RESULTS

3.1 Data Collection

A total of 12,372 images were aggregated and curated split evenly between scenes with and without fire. The data were then split randomly into training and testing data using an 80/20 split ratio (9,898 and 2,474 images respectively) while maintaining a balanced ratio of fire and no-fire scenes. Additionally, 10 percent of the training data (990 images) were randomly selected for validation after each training epoch. Figure 3 shows a sample of downloaded images.

Figure 3. A sample of images collected to support the development of the fire-watch model. Images on the right depict scenes with flame features. Those on the left show normal states.

3.2 Model Development

The fire watch deep CNN model trained for 905 epochs where training terminated when the model's loss function output stopped decreasing over 150 consecutive epochs. To mitigate potential overfitting, only the model was saved that yielded the lowest loss function output at epoch 755. Model training and validation loss and accuracy plots by epoch are shown in Figure 4. They show that model validation accuracy exceeded 90 percent by the end of training and that the loss function decreased to slightly less than 0.2 at the lowest point (epoch 755).

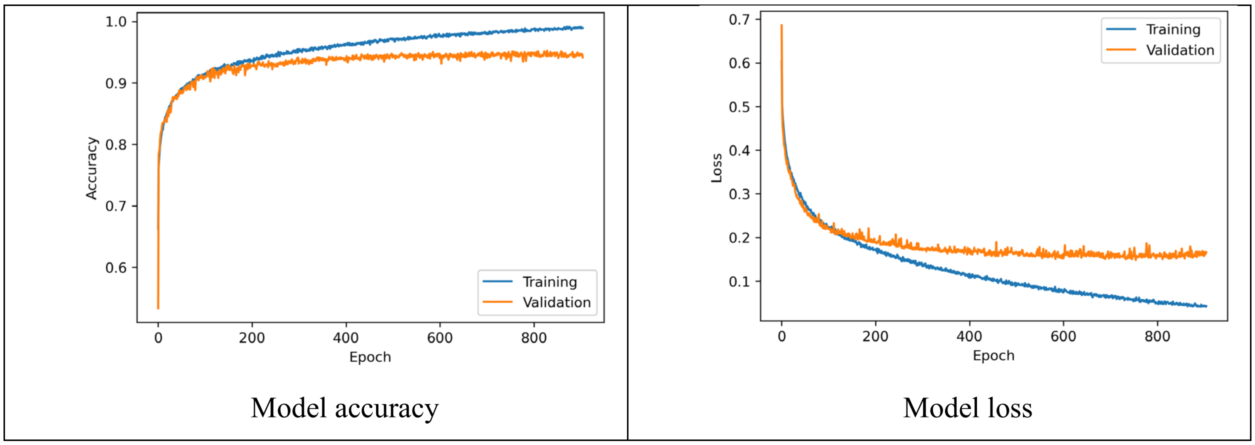

Figure 4. Model accuracy (left) and loss (right) metrics for training and validation datasets during training.

After training and validation, the model was tasked to predict image labels ("flame", "no flame") for the randomly selected test image dataset consisting of 2,474 images. The trained fire watch deep CNN model achieved an accuracy of 0.954 on the test data. The mean inference time was 40.2 milliseconds per image with a standard deviation of 1.5 milliseconds per image. Figure 5 shows a confusion matrix for the test dataset. It shows the trained model misclassified 45 images without flame as "flame" and a slightly higher number, 69, as "no flame" that did contain flame features.

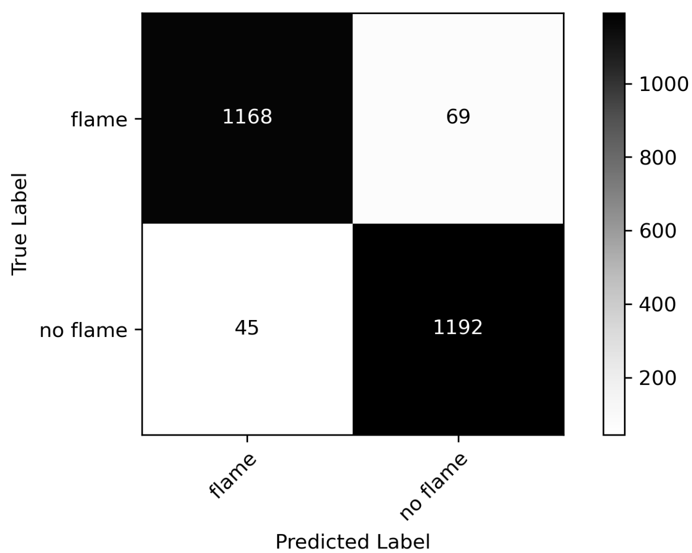

Figure 4. Model accuracy (left) and loss (right) metrics for training and validation datasets during training.

4 CONCLUSIONS

For this analysis, over 12,000 images containing scenes of varying environments (indoor, outdoor, low-light, daylight, artificial illuminance, etc.) and scene composition (people, equipment, vehicles, various activities, etc.) with and without flame features of varying size and shape where aggregated and curated. The purpose of this effort was to try to account for what could be encountered in industrial settings like that of an NPP. The majority of the images were downloaded from the YFCC100m repository along with additional images from datasets used to support previous analysis efforts to develop image-based fire detection algorithms.

A deep CNN architecture was developed incorporating modules consisting of parallel convolutional channels of varying spatial scales to improve model robustness for complex environments where potential fire events could vary relative to size, distance from the image sensor, and background complexity. The results indicate that the fire watch deep CNN architecture was able to learn necessary features to achieve a high classification accuracy (0.954) on the test dataset consisting of heterogenous flame features and scene environments and compositions. Additionally, the model was able to make image inferences at a mean rate of 40.2 milliseconds per image indicating that it could be used to monitor video-based images at a sub-second interval.

Because fire watch is a safety requirement for the nuclear regulator license, performance needs to be maximized to present the case for regulators to accept such solutions as a replacement for the conventional fire watch. Although this effort yielded promising results, additional testing is recommended to evaluate performance and identify potential performance gaps leading to false negative and false positive outcomes.

5 ACKNOWLEDGMENTS

The authors would like to acknowledge the U.S. Department of Energy Light Water Reactor Sustainability (LWRS) program for funding this effort. INL is a multi-program laboratory operated by Battelle Energy Alliance LLC, for the United States Department of Energy under Contract DE-AC07-05ID14517. This work of authorship was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately-owned rights. The United States Government retains, and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a nonexclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or any agency thereof. The INL issued document number for this paper is: INL/JOU-21-XXXXX.

6 REFERENCES

1. US Nuclear Regulatory Commission. “The U.S. Nuclear Regulatory Commission (NRC) Regulatory, 2009. Guide 1.189-Fire Protection for Operating Nuclear Power Plants.” *The U.S. Nuclear Regulatory Commission (NRC) Regulatory, 2009. Guide 1.189-Fire Protection for Operating Nuclear Power Plants*. (2009).

2. Cortes, C. and Vapnik, V., Support-vector networks. *Machine learning*, **20(3)**, pp.273-297 (1995).

3. A. K. Jain, Jianchang Mao and K. M. Mohiuddin, "Artificial neural networks: a tutorial,", *Computer*, **29(3)**, pp. 31-44, (1996).

4. Çetin, A.E., Dimitropoulos, K., Gouverneur, B., Grammalidis, N., Günay, O., Habiboğlu, Y.H., Töreyin, B.U. and Verstockt, S., “Video fire detection–review,” *Digital Signal Processing*, **23(6)**, pp. 1827-1843, (2013).
5. Çelik T, Demirel H, “Fire detection in video sequences using a generic color model,” *Fire Safety Journal*, **44(2)**, pp. 147-158 (2009).
6. Thou-Ho C, Ping-Hsueh W, Yung-Chuen C, “An early fire-detection method based on image processing,” *International Conference on Image Processing, 2004. ICIP '04*., pp. 1707-1710 (2004).
7. Marbach G, Loepfe M, Brupbacher T, “An image processing technique for fire detection in video images,” *Fire Safety Journal*, **41(4)**, pp. 285-289 (2006).
8. Hamins A, Yang JC, Kashiwagi T, “An experimental investigation of the pulsation frequency of flames,” *Symposium (International) on Combustion*, **24(1)**, pp. 1695-1702 (1992).
9. Chen J, He Y, Wang J, “Multi-feature fusion based fast video flame detection,” *Building and Environment*, **45(5)**, pp. 1113-1122 (2010).
10. Abadi M, Barham P, Chen J, Chen Z, Davis A, Dean J, Devin M, Ghemawat S, Irving G, Isard M. “Tensorflow: A system for large-scale machine learning,” *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pp. 265-283 (2016).
11. Oliphant TE, “Python for Scientific Computing,” *Computing in Science & Engineering*, **9(3)**, pp. 10-20 (2007).
12. Muhammad K, Ahmad J, Mehmood I, Rho S, Baik SW, “Convolutional Neural Networks Based Fire Detection in Surveillance Videos,” *IEEE Access*, **6**, pp. 18174-18183 (2018).
13. Dunnings AJ, Breckon TP. “Experimentally Defined Convolutional Neural Network Architecture Variants for Non-Temporal Real-Time Fire Detection,” *2018 25th IEEE international conference on image processing (ICIP)*, pp. 1558-1562 (2018).
14. Achanta R, Shaji A, Smith K, Lucchi A, Fua P, Süsstrunk S, “SLIC Superpixels Compared to State-of-the-Art Superpixel Methods,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **34(11)**, pp. 2274-2282 (2012).
15. Hu C, Tang P, Jin W, He Z, Li W. “Real-Time Fire Detection Based on Deep Convolutional Long-Recurrent Networks and Optical Flow Method,” *2018 37th Chinese Control Conference (CCC)*, pp. 9061-9066 (2018).
16. Sharma, J., Granmo, O.C., Goodwin, M. and Fidje, J.T., “Deep convolutional neural networks for fire detection in images,” *International conference on engineering applications of neural networks*, pp. 183-193 (2017).
17. Thomee B, Elizalde B, Shamma DA, Ni K, Friedland G, Poland D, Borth D, Li LJ, “YFCC100M: The New Data in Multimedia Research,” *Communications of the ACM*, **59(2)**, pp. 64-73 (2016).
18. Cazzolato M, Avalhais L, Chino D, Ramos J, Souza J, Rodrigues Jr J, Taina A, “FiSmo: A Compilation of Datasets from Emergency Situations for Fire and Smoke Analysis,” *Brazilian Symposium on Databases-SBBD*, pp. 213-223 (2017).
19. Glorot, X., A. Bordes, and Y. Bengio. “Deep Sparse Rectifier Neural Networks,” *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 315-323 (2011).
20. Ioffe, Sergey, and Christian Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *International conference on machine learning*, pp. 448-456 (2015).

21. Yuan, F., Zhang, L., Wan, B., Xia, X. and Shi, J., "Convolutional neural networks based on multi-scale additive merging layers for visual smoke recognition," *Machine Vision and Applications*, **30(2)**, pp. 345-358 (2019).