INL/EXT-19-55703
Revision 0

Light Water Reactor Sustainability Program

Automating Fire Watch in Industrial Environments through Machine Learning-Enabled Visual Monitoring

September 2019

U.S. Department of Energy

Office of Nuclear Energy

INL/EXT-19-55703
Revision 0

Automating Fire Watch in Industrial Environments through Machine Learning-Enabled Visual Monitoring

Ahmad Al Rashdan, Michael Griffel, and Larry Powell

September 2019

Prepared for the
U.S. Department of Energy
Office of Nuclear Energy

ABSTRACT

Nuclear power plants (NPPs) are experiencing significant cost challenges 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 automating manually intensive tasks to reduce O&M costs, especially for monitoring activities. One monitoring function that is visually demanding and can occur frequently in order to meet the requirements of the fire protection program in an NPP is fire watch. A fire watch is a worker physically stationed at a given location (where normal fire protection measures are challenged) with the sole responsibility of observing a given area to ensure a fire is detected and mitigated promptly. This effort targets migrating fire watch from a manual model to an automated model. An automated visual monitoring approach using machine learning was pursued to closely resemble the actual manual visual monitoring performed by a fire watch.

Multiple machine-learning methods were evaluated for automated visual recognition of fire; long short-term memory (LSTM) neural networks, coupled to a color-based feature-extraction method to extract fire regions, resulted in the most accurate results. In addition to developing the method, a training data set for fire in industrial environments was developed. Because machine-learning methods are sensitive to the training data used, using only industrial fire videos were essential. Various sources of publicly available datasets containing scenes approximating industrial settings with and without fire were investigated and aggregated. They consist of YouTube-8M datasets that were downloaded and used to train and validate the neural network modelling approaches used in this analysis. Additional video datasets available from previously published literature sources and an online repository provided by Yahoo (YFCC100m and FiSmo) were also downloaded and aggregated to test the color feature extraction efforts and isolate fire regions in a frame.

The developed method was trained and validated using 1,000 industrial-only fire videos extracted from YouTube-8M datasets. After the developed method was trained and validated using the YouTube-8M data, it was tested using a set of 62 videos, split evenly between fire and no-fire scenarios, from a Yahoo videos repository. The accuracy achieved by the developed method was 0% missed positives (i.e. fire occurred but not detected) and 8% false positives (i.e. system indicates fire though there is no fire). Though smoke was not targeted in this effort, the method is expected to perform with similar accuracy for smoke detection, assuming a smoke training dataset is developed. The integration of this developed method with a suite of fire and smoke detection technologies (being researched or sold as products) is planned next to reduce false-positive events. Additionally, using a machine-learning method in a process to meet the licensing requirement necessitates unboxing the machine learning “black box,” i.e., explaining the rationale behind the decision-making and providing extensive validation to the regulator of the method’s ability to replace the human fire watch. This is also planned for future research.

ACKNOWLEDGEMENTS

The authors would like to thank the Light Water Reactor Sustainability (LWRS) program for funding and supporting this effort. Also, the authors would like to thank Talen Energy in specific and the Utilities Service Alliance in general for the insight provided on automating fire watch and current industry practices and needs.

CONTENTS

FIGURES

TABLES

ACRONYMS

ANN	artificial neural networks
CNN	convolutional neural networks
DCLRN	deep convolutional long-recurrent networks
FiSmo	fire smoke
fps	frames per second
HMM	hidden Markov model
ID	identification
IR	infrared
JSON	JavaScript object notation
LSTM	long short-term memory
LWIR	long-wave infrared
LWRA	light water reactor sustainability
NPP	nuclear power plant
NRC	Nuclear Regulatory Commission
O&M	operations and maintenance
PCA	principle components analysis
ReLU	rectified linear unit activation
RGB	red, green, and blue
RNN	recurrent neural network
SVM	support vector machine
UV	ultra-violet
VFD	video-based fire detection
YFCC100m	Yahoo flicker creative commons 100 million

AUTOMATING FIRE WATCH IN INDUSTRIAL ENVIRONMENTS THROUGH MACHINE LEARNING-ENABLED VISUAL MONITORING

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" (U.S. Nuclear Regulatory Commission 2009). A person conducting a fire watch (e.g., in Figure 1) could spend hours watching a certain equipment, room, or environment.

Figure 1. Fire watch for welding hot work performed in an industrial environment (U.S. Department of Labor 2019).

Fire watch is performed by operators or fire protection staff when work performed on fire protection systems prevents those systems from performing their functions, or when there is a fire hazard associated with a work activity—i.e., hot work, such as welding. Regulatory Guide 1.189 Section 2.4.C states,

> *Successful fire protection requires inspection, testing, and maintenance of the fire protection equipment. A test plan that lists the individuals and their responsibilities in connection with routine tests and inspections of the fire protection systems should be developed. The test plan should contain the types, frequency, and detailed procedures for testing. Frequency of testing should be based on the code of record for the applicable fire protection system. Procedures should also contain instructions on maintaining fire protection during those periods when the fire protection system is impaired or during periods of plant maintenance (e.g., fire watches).*

Also, Regulatory Guide 1.189 Section 2.2.1 states,

> *Work involving ignition sources such as welding and flame cutting should be carried out under closely controlled conditions. Persons performing such work should be trained and equipped to prevent and combat fires. In addition, a person qualified in performing hot-work fire watch duties should directly monitor the work and function as a fire watch.*

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.

Most industrial fire-detection systems are fixed in optimal locations (i.e., usually ceiling mounted) and typically detect smoke. These systems are coupled to a decision-making unit, often by a voting process through a dedicated safety system. For a fire watch, a temporary mobile setup is needed. Mobile smoke detectors can be placed adjacent to the work area. However, they might not be pointed in the direction of the fire flames or smoke and, therefore, introduce some uncertainties or delay in their detection capability. There are several technologies and concepts more sophisticated than smoke detectors—specifically, infrared (IR) or ultraviolet (UV) sensors that can detect fire (AZO Sensors 2017). However, such solutions can cause false positives (e.g., system indicates fire though there is no fire) or missed positives (e.g., fire not detected), depending on their sensitivity setting. Also, if the fire view is obstructed, IR and UV sensors would not detect the fire, because they require direct line of sight. Because fire watch is a safety requirement of the nuclear regulator license, these limitations must be overcome to present a case for regulators to accept automated solutions as an acceptable replacement for fire protection.

IR cameras—for situations in which visible light is limited or methodologies depending upon visible wavelengths are unable to differentiate features because of similar background features—are another type of fire-detection technology experiencing growing interest. Although IR cameras are well suited to detect heat or flame in low-visible-light conditions, other objects, such as a computer, heater, or light source, can result in misclassifications. Conversely, color sensors require constant illumination of targets, and changing light conditions or sources may confound classification algorithms. As a result, the ideal solution for fire watch would be to combine all these technologies into one decision, producing better decision making. In this effort, development of a method to detect smoke or fire from a distance in a visual video stream with high fire-detection performance and quantification is desired. An automated visual monitoring was pursued to closely resemble the actual manual visual monitoring performed by the fire watch. The resemblance is expected to simplify validation and deployment of the developed solution. The outcome of this effort would be one solution to combine with previously described technologies for superior overall performance.

The following sections will describe some recent image-processing and machine-learning methods that were developed to visually detect fire or smoke.

1.1 Literature Review

A significant amount of literature is available on detecting fire or smoke 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), artificial neural networks (ANNs), etc. Çetin et al. (2013) 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 using varying methods, many current systems yield false positives or alarms, resulting from changing light conditions, shadows, and other image features. Table 1 shows a breakdown of some key parameters.

Table 1. Parameters impacting model development.

Parameter	Description
Environment	Target area, e.g., indoors or outside
Detection Space	The size of the target area and distance of potential fire targets from the sensor
Objects	Objects within the target area (e.g., people, desk, light source, automobile, etc.)
Light Source	Light consists of artificial or natural sources, or both
Light Duration	Light source and duration vary by time of day, season, etc.
Sensor	Spatial, spectral, and temporal resolutions; stationary or moving

Earlier VFD methods focused on flame detection. Smoke detection in an image has recently gained interest because smoke typically spreads faster and represents an opportunity to detect combustion earlier (Çetin et al. 2013). 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.

1.1.1 Image Processing Methods

Many VFD techniques focus on fire or smoke 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 (Çetin et al. 2013). An example of each type of the surveyed methods is listed herein for demonstration.

A system was demonstrated that uses a Gaussian-smoothed color histogram to classify fire-colored (i.e., red-yellow) pixels coupled with temporal-variation detection to determine which pixels were likely derived from fire reflectance (Phillips et al. 2002). The paper indicates this approach works in varying conditions and is insensitive to camera motion. Borges and Izquierdo (2010) noted the green color band can be used to differentiate spatial color variation of flames, and if the standard deviation of the green color band exceeds 50 in a typical color video, the region is labeled as a candidate flame region.

Approaches such as these rely solely on color signatures and, therefore, can result in false positives when objects with similar color patterns as those of flames or smoke are present in scenes.

Methodologies to detect smoke in video images were successfully employed using spatial wavelet transformations of current and background images followed by edge-energy quantification and analysis of smoke boundaries using a hidden Markov model (HMM) (Töreyin et al. 2006).

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. 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 (Chen et al. 2010). However, 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. Also, industrial scenes have the potential to include moving objects of varying size, shape, color, and movement speed, which introduce potential sources of error.

1.1.2 Machine-learning Methods

Recently, machine-learning methodologies based on artificial neural networks (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 (Abadi et al. 2016) for common programming languages such as Python (Pedregosa et al. 2011). 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- or smoke-detection system using machine learning, it is important to account for varying parameters (Table 1) impacting a classification model's development and biases. A few examples of machine-learning methods are listed here for demonstration.

Muhammad et al. (2018) proposed a system using convolutional neural networks (CNNs) (discussed in Rawat and Wang 2017) 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. Dunnings and Breckon (2018) demonstrated high classification accuracies by coupling a super-pixel localization framework using the K-means clustering algorithm (Achanta et al. 2012) 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. Also, CNNs lack the ability to incorporate temporal learning (memory) which can improve online learners' ingesting of temporal data streams such as video-based images. This finding was verified in this effort.

Researchers achieved high fire- or smoke-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 (Hu et al. 2018). Their analysis also indicated DCLRN outperformed a CNN when flame or smoke 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.

1.1.3 Summary of Literature Findings

Although many literature resources exist showing varying approaches and outcomes for the development of a VFD, significant technical challenges exist to develop a robust system suitable for a NPP. Given the potential scene complexity and diversity of industrial power-plant environments, a color-feature extraction approach based on image processing methods will likely yield many false positives due to fire-colored objects passing into scenes. Development of a machine-learning model poses significant challenges in that machine-learning methods are sensitive to the training data used and available training data containing scenes with and without fire in an environment approximating an NPP are not readily available to support model development. Based on the literature review, it was decided to evaluate both an image-processing-based method and a machine-learning method and to develop a customized method to outperform both. The results of the first two methods evaluation will be used to benchmark the performance of the customized method.

A significant challenge to develop a fire-detection method is the aggregation of suitable datasets with which to train, validate, and test methods during development. The following sections provide details on the sources of data for method development, validation, and testing.

2. DATA SOURCES AND PREPARATION

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. This, in turn, necessitated searching for publicly available datasets containing fire imagery and industrial 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.

In this case, no known dataset exists explicitly to support the fire watch task. During the literature review, multiple online repositories were discovered containing video and imagery, some specific to the development of a VFD that were utilized to provide video datasets. Extensive efforts were made to aggregate suitable data for an industrial setting like that of an NPP. A well-curated dataset was explored first. Researchers also elected to compile datasets to demonstrate model robustness and generalization. The following sections describe the datasets used in detail.

2.1 YouTube-8M

YouTube-8M is a fully curated and validated frame-by-frame dataset of labeled video features that have been preprocessed into one-dimensional mathematical vectors for each frame containing 1,024 features (Abu-El-Haija et al. 2016). YouTube-8M contains unique 3,862 labels encoded as integers associated with features and 25 categories. Google has made this dataset publicly available to support the advancement of large-scale machine vision applications and model development for a wide variety of applications. YouTube-8M features are derived from over 237,000 labeled video segments from over 5 terabytes of video data. The data are derived from a diverse sampling of existing online content posted to YouTube. The original video-data content is not publicly available. Instead, Google has made the labeled, encoded feature arrays available for public use to reduce storage needs and maintain privacy. To generate this dataset, the following steps were applied:

- The original videos were decoded at 1-fps for 360 seconds.
- Feature extraction, using the publicly available CNN architecture within the Inception Network (Szegedy et al. 2015) and maintained by Google, was applied to reduce dataset size (Abu-El-Haija et al. 2016). Specifically, the video-image data were passed through the convolution and pooling steps to extract image features and reduce data dimensionality.

- Data compression, using principle components analysis (PCA, reviewed in Jolliffe and Cadima 2016) and whitening were applied to further reduce dimensionality, producing the video features as encoded vectors in lieu of original video frames. Whitening (sphering) is a common image-processing step to reduce redundancy within the output features so that the output vector contains features that are less correlated with each other.
- The feature values were normalized to 8-bit integer values ranging from 0 to 255, making the final encoding as efficient as possible for computational and memory requirements.
- The data were annotated such that specific features are temporally localized within the video segments via human verification. This information is compiled into JavaScript Object Notation (JSON) files containing "label" and "title" keys. The label keys are associated with numeric values mapped to specific features within the video, and title denotes a unique identifier associated with the video.

A total of 1,000 YouTube-8M datasets were downloaded using an executable script provided by Google. However, the data contained scenes with industrially relevant and non-industrially relevant categories and labels such as Games, Pets & Animals, Guitars, etc. During examination of the YouTube-8M video datasets and associated labels, it was also noted the individual video datasets often contained a mixture of labels and categories describing varying topical themes. For example, a video segment can have both Sports and Science categories with Fire and Juggling labels. Another segment can contain several categories (e.g., Autos & Vehicles, Science, Law & Government, etc.) and labels for Fire, Factory, Machine, Engine, Emergency, Truck, etc. This diversity within the data made this an appropriate dataset with which to train and validate machine-learning models because it would likely improve overall model generalization needed for a system monitoring the diverse scenes within an NPP complex. However, analysis was needed to filter the data to an appropriate subset to be used to train and validate the fire watch method to the point at which it was relevant to an industrial environment. To achieve this, specific environments descriptive of industrial settings were sought. Because it was not possible to review the native video files, YouTube-8M datasets were selected by identifying terms that could be associated with an industrially relevant environment within the associated JSON files. The labels selected were from the Business & Industrial category as no fire dataset. The fire label, part of the Science category, was selected for the fire dataset. Using these labels, 663 datasets were selected from the original 1,000 containing fire and no fire scenes of approximately equal proportions relative to industrial facility environments. Table 2 shows the final categories and labels of the subset data.

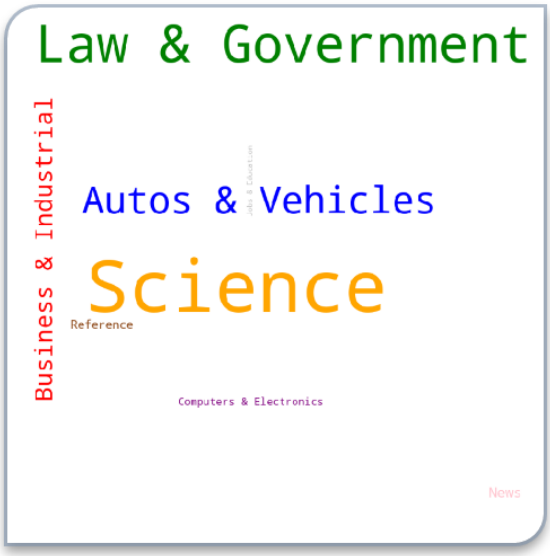
A typical approach for machine-learning model development is to randomly split a curated and labeled dataset into training and validation subsets. In this case, a 70/30 ratio was used where 70% of the subset was randomly selected for training, with the remaining 30% reserved for validation. Because the YouTube-8M datasets are derived from video segments containing multiple frames, researchers partitioned the training and validation data at the video level instead of separating individual frames. This was done to prevent overfitting and improve model generalization. To evaluate the diversity of the training and validation data, word clouds were produced to ascertain the label and categorical diversity of the fire and no-fire subsets.

Figure 2 shows the word-cloud plots for the training dataset derived from the YouTube-8M data for fire (top) and no fire (bottom) classes. The plots show the proportional occurrences of the category terms (left) and feature labels (right). The font size of each term is directly proportional to the number of occurrences. The word-cloud analysis shows that the feature labels and categories compiled for the analysis are diversely associated with other label and video categories. The fire dataset is dominated by Fire and, obviously, associated labels such as Fire Engine, which can also appear with the Fire label. The label terms commonly associated with industrial environments dominate the no-fire portion of the dataset and show significant diversity relative to objects commonly associated with industrial environment, such as Machine, Engine, Vehicle, Pump, Construction, etc. The occurrence of labels in both classes indicates

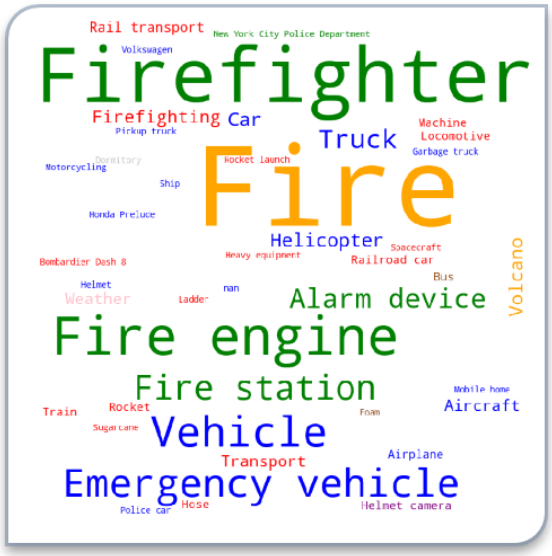
the data are well suited to develop a machine-learning model. For example, if a firefighter was only limited to scenes with a fire, the model could learn to recognize a firefighter as opposed to learning fire features. In that case, it would predict fire anytime it saw a firefighter regardless of whether fire was present. The inclusion of a firefighter in both fire and no-fire classes emphasizes model learning on fire features.

Table 2. YouTube-8M categories and labels used to develop a VFD.

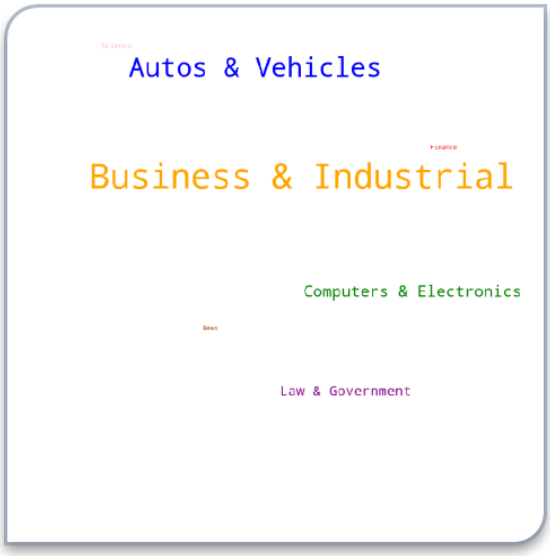
Label	Label Name	Category
62	Machine	Business & Industrial
208	Heavy equipment	Business & Industrial
270	Tool	Business & Industrial
347	Construction	Business & Industrial
359	Fire	Science
500	Road	Business & Industrial
498	Printing	Business & Industrial
522	Elevator	Business & Industrial
606	Concrete	Business & Industrial
616	Floor	Business & Industrial
750	Glass	Business & Industrial
745	Loader (equipment)	Business & Industrial
780	Crane (machine)	Business & Industrial
798	Metal	Business & Industrial
899	Ceiling	Business & Industrial
962	Rocket	Business & Industrial
1015	Pump	Business & Industrial
1025	Tile	Business & Industrial
1128	Turbine	Business & Industrial
1154	Manufacturing	Business & Industrial
1277	Brick	Business & Industrial
1311	Metalworking	Business & Industrial
1651	Vending machine	Business & Industrial
1673	Wind power	Business & Industrial
1739	Valve	Business & Industrial
2038	Steel	Business & Industrial
2805	Asphalt	Business & Industrial
3062	Heat pump	Business & Industrial



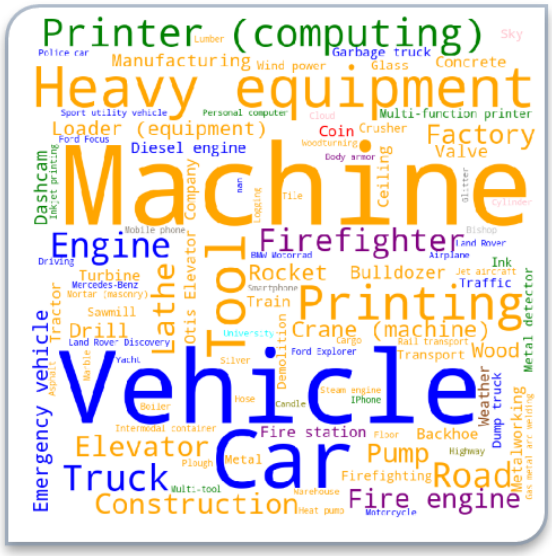
(a) Fire categories



(b) Fire labels



(c) No-fire categories



(d) No-fire labels

Figure 2. A word-cloud plot for the training data derived from the YouTube-8M dataset.

Figure 3 shows the resultant word cloud for the validation portion of the datasets, split between fire and no-fire and derived from the YouTube-8M data. Like Figure 2, the plots are split between fire (top) and no-fire (bottom) classes. The plots show the proportional occurrences (based on font size) of the category terms (left) and feature labels (right). The validation data exhibit similar patterns to the training data. This indicates the validation dataset was well suited for model validation, because the training and validation datasets should be similar for optimal machine-learning analysis.

(a) Fire categories

(b) Fire labels

(c) No-fire categories

(d) No-fire labels

Figure 3. A word-cloud plot for the validation data derived from the YouTube-8M dataset.

2.2 YFCC100m and FiSmo

Although the YouTube-8M dataset was shown to have significant diversity, it consisted of preprocessed labeled features encoded as mathematical vectors. The original image data are not available, requiring a full dependency upon the curation and labeling schemas. As a result, additional datasets were selected that consisted of actual video data that could be viewed by researchers and then processed to match the vector encoding of YouTube-8M data. These data were used to further evaluate model performance and generalization capabilities by testing the machine-learning models on data completely independent of the training and validation data. The Yahoo Flickr Creative Commons 100 Million (YFCC100m) dataset was therefore utilized as a source for additional fire videos (Thomee et al. 2015).

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.

The Fire Smoke (FiSmo) dataset is a compilation of images and videos taken from scenes showing emergency situations where fire, smoke, or explosions occur (Cazzolato et al. 2017). This dataset was compiled from online video-sharing repositories including Yahoo's Flickr and Google's YouTube[a]. The intended purpose is to support ongoing research to apply computational techniques to support video and image processing for emergency situations. Figure 4 shows a scene taken from a FiSmo video file showing active flames in a scene associated with an industrial setting containing metal structures, pipes, valves, and tanks.

Figure 4. A video frame from a FiSmo (Cazzolato et al. 2017) dataset showing a fire in an industrial setting.

The FiSmo video files were well annotated and organized. They contain a total of 81 video datasets. However, many of the videos showed similar or identical scene information, which reduces data heterogeneity needed to evaluate machine-learning performance. For example, many of the videos showed the same set of piping infrastructure from similar viewing angles in multiple videos.

To improve heterogeneity of the testing dataset, additional YFCC100m videos were identified and downloaded. Given the massive size of the YFCC100m dataset, a searching algorithm was developed to parse the table records and return video IDs with tags similar to those shown in Table 2.[b] Then, the IDs were randomly subset and used to reference and download the native video files. A total of 391 videos were downloaded and evaluated to select diverse, industrially relevant video datasets. After manual curation, 62 videos, split equally between fire and no-fire scenarios, but associated with industrial scenes, were selected from the downloaded FiSmo and YFCC100m datasets. The videos acquired from FiSmo and YFCC100m had a wide variety of variability relative to image resolution, frame rates, and duration. Analysis of the videos showed image resolution ranging from 292 × 240 pixels to 1,920 × 1,080 pixels.

[a] This is not the YouTube-8M data, but actual YouTube videos.

[b] In its compressed format, the table houses almost 15 Gigabyte (GB) of data, prohibiting the use of desktop software such as Excel to open and view the records. Python 3.6 was used instead.

Frame rates ranged from 7 to 30 fps, with video duration running from about 5 to 900 seconds. For this reason, additional standard video processing methods were developed to transform the video datasets to suitable formats for each respective VFD method. Because these video processing methods are unique for each method configuration, these details will be provided in the respective method subsections in Section 3.

3. METHODS

Section 1.1 revealed multiple peer-reviewed articles chronicling fire detection in video imagery using machine vision and machine learning. Some approaches relied on rules-based classification schemas based on pixel-color features. Others utilized ANNs and temporal algorithms to classify fire and no fire scenes. For this task, two methods were used and evaluated as baseline classifiers against an advanced long short-term memory (LSTM) classifier developed in this effort.

Figure 5 shows the high-level workflow for each method used. The color-feature classifier incorporated a simple pixel-extraction method based on fire-color features to output a binary classification of fire or no-fire at frame and video levels on the YFCC100m and FiSmo videos. The CNN classifier was trained on the downloaded YouTube-8M dataset. However, the CNN was not tested on the YFC100m and FiSmo videos because training and validation results indicated the model could not "learn" fire detection using the YouTube-8M data architecture and the selected datasets (as will be explained in Section 3.2). The LSTM classifier incorporated an LSTM neural network trained on the YouTube-8M dataset. Prior to classification of the YFCC100m and FiSmo video datasets, potential fire regions within the frames were identified by the pixel-color features and subset, and then processed to match the LSTM mode input configurations for final classification as fire or no fire.

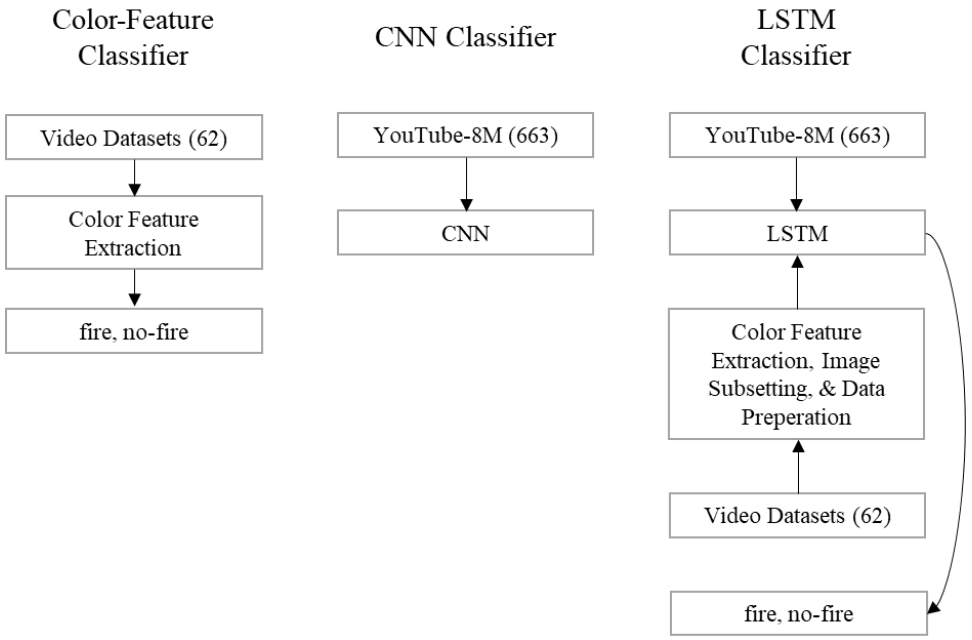


Figure 5. High-level method workflow for the three evaluated methods.

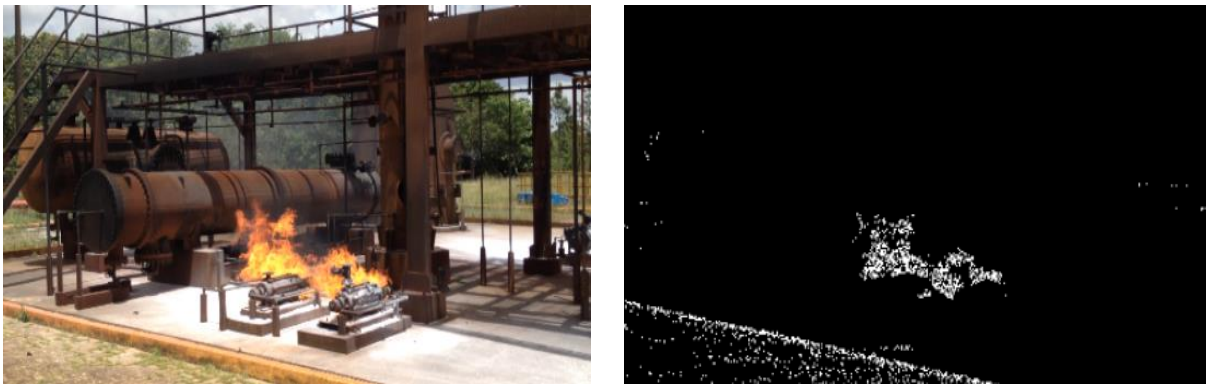
3.1 Color Feature Classifier

The method to extract and classify image features by pixel values is similar to several approaches outlined in Section 1.1.1 where pixels are categorized as either flame or no-flame based on color-channel thresholds. Pixels with color features matching known flame color are typically expressed as red-yellow, with specific attributes relative to image hue, saturation, and other properties. Thus, this method does not incorporate training and validation as it is a rules-based approach employing known pixel thresholds.

3.1.1 Method

For this analysis, it was decided to utilize several equations from multiple literature sources to extract image pixels matching with color features matching those of flame (Çelik and Demirel 2009, Thou-Ho et al. 2004). Çelik & Demirel, (2009) 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. (2004), using different rules-based detection in the RGB feature space. Both approaches were leveraged for flame detection, by which the percentage of pixels that are flagged as fire in a video was used to declare the frame as a fire frame.

The method developed parsed each test video file at a one-second interval. The equations identified in the literature where applied as conditions, and every pixel meeting all conditions was classified as a fire pixel. Remaining pixels were classifed as no fire. Figure 6 shows a video frame output from the algorithm. The image on the left is the native image, and the image on the right shows the binary pixel classificaiton output where all pixels classifed as fire are shown in white. In this image, the fire pixels make up 1.9% of the total number of image pixels. The video-frame image-classification threshold selected was set to 1% of the number of pixels in a frame as fire. A video-based classifcation rule was also applied in that any video with greater than 50% of frames classifed as fire was also classifed at a video level as fire.



(a) Native image (b) Pixels classified as fire

Figure 6. Example of pixels classification as fire pixels.

3.1.2 Results

The color-feature classifier was tested on the YFCC100m and FiSmo video dataset referenced in Section 2.2. The YouTube-8M data of Section 2.1 could not be utilized for color-feature extraction because the videos are not provided, as was discussed earlier. Figure 7 contains line plots for every video showing the percentage of pixels classified as fire for every frame at a 1-second interval. The results indicate that between 0 and 2 percent, there is significant intermingling of pixels classified as fire in both fire and no fire videos. This suggested a 1% threshold for fire detection would be suitable. The patterns also indicate sometimes dramatic scene-to-scene changes in the pixel classification occur.

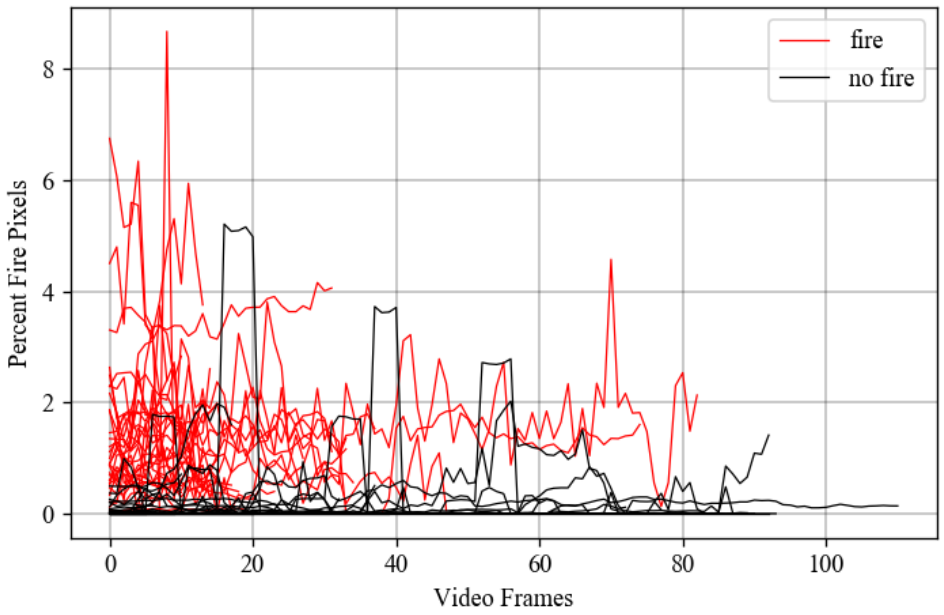


Figure 7. The percentage of pixels classified as fire for every 1-second frame in the video datasets.

Table 3 shows the results of a confusion matrix for the frame-based accuracy metrics. The analysis indicates the 1% threshold resulted in only identifying about 60% of actual fire frames. However, the no-fire classification was more accurate. It is likely that using a lower threshold will improve fire classification, but potentially result in increased false-positive error. Determining an optimal classification-threshold value (without incurring either high false-positive or false-negative outcomes) is difficult. Table 4 shows the results of the video-based classification confusion matrix. As in Table 3, the results indicate low detection probability based on the 1% threshold.

Table 3. Frame-based confusion matrix for the color feature classifier.

		Predicted	
		Fire	No-fire
True Value	Fire	0.60	0.40
	No-fire	0.03	0.97

Table 4. Video-based confusion matrix for the color feature classifier.

		Predicted	
		Fire	No-fire
True Value	Fire	0.61	0.39
	No-fire	0.00	1.00

3.2 CNN Classifier

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 and pooling 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.

Figure 8 shows an example of the inner workings of a CNN network. An input image array is passed into a convolutional moving-window filter (conv2d) coupled with a rectified linear unit activation function (ReLU) (conv2d+ReLu). The moving-window filter passes over the input array to produce a new array that contains high-level or dominant features such as edges. The ReLU function is then applied to increase the non-linearity of the output to further enhance the features. Next, the array is passed to a pooling step that comprises another moving window to reduce array dimensions so as to improve model efficiency and reduce noise. These steps are repeated to strengthen the feature extraction and further reduce dimensionality.

Next, the convoluted and pooled features are fed into a fully connected network of artificial neurons that are connected by weights (black lines in Figure 8) to the output node, which is responsible for determining the class output. Finally, the training data and their associated labels (classes) are inputted into the model, which used that information to modify the internal weight values to learn the mathematical relationship between the input features and the associated classes. During training, as the model learns, the weights are continually adjusted to mathematically map the known input features and output classes.

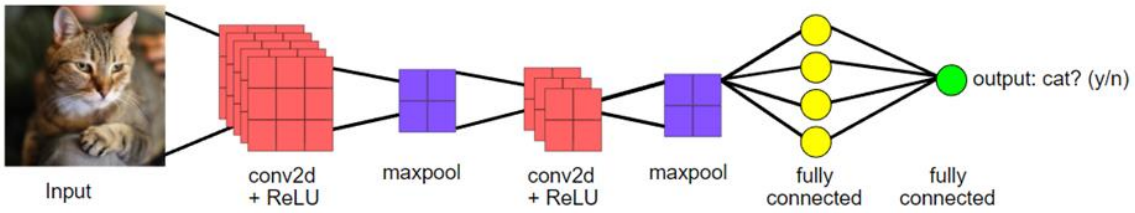


Figure 8. An example CNN architecture containing two convolution and pooling segments feeding into a densely connected layer for classification (from Google developers 2019).

3.2.1 Method

For this task, a CNN was trained and validated on the YouTube-8M dataset referenced in Section 2.1 to classify video-image data as fire or no-fire. The CNN classifier was set up using functions existing in the TensorFlow library. The network was configured to input the YouTube-8M features vectors and apply three convolution and pooling steps to extract the important features and reduce dimensionality and noise. After that, the modified features were inputted into a fully connected hidden layer connected to an output node responsible for the class determination. A 70-30 ratio of training and validation subsets respectively was used.

After each cycle (epoch) of training, the model was tasked to predict the fire or no-fire classification of the validation portion, which was next compared to the actual video fire or no-fire content. As the model learned over continual epochs, the validation accuracy was tracked to ensure model training was not stopped before the validation converged; otherwise, the model would likely perform poorly, given it did not learn all the patterns in the feature data. When validation accuracy failed to improve over two consecutive epochs, the training and validation regime was terminated. This was done to prevent the model from memorizing the training data, which negatively impacts generalization or the ability to predict from new data.

3.2.2 Results

Figure 9 shows the tracked model validation accuracy as the CNN model attempted to learn on the YouTube-8M dataset. Training and validation were terminated after 50 epochs as the model could not generate any appreciable overall improvement in prediction accuracy on the validation dataset. Overall training and validation accuracy only achieved about 50%, which is no better than a baseline guess. This pattern indicates the CNN architecture was not suited to learn using the YouTube-8M datasets downloaded for this effort.

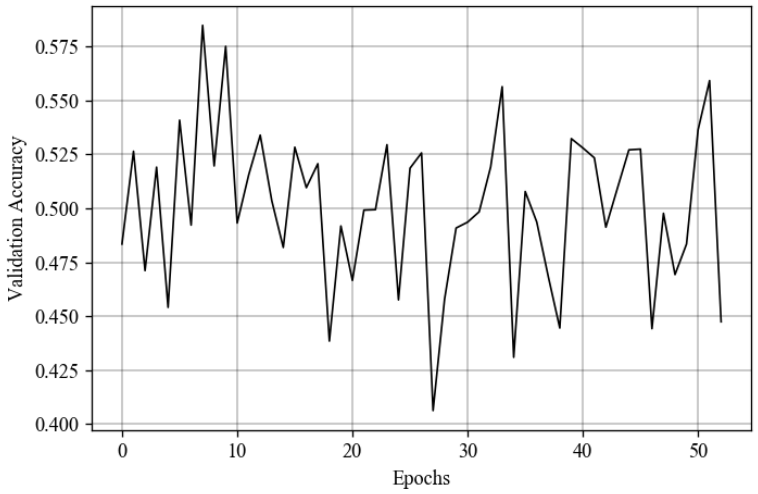


Figure 9. Accuracy of CNN model.

The poor performance of the CNN during training and validation was likely due to multiple issues. YouTube-8M datasets are made up of features (from the Google feature-extraction process described in Section 2.1) which are generated via a network of convolutions and pooling layers. It is possible that preprocessing step with the additional convolutions and pooling from the CNN classifier erased the features that the model needed to learn the patterns. Also, because videos represent temporal sequences of images, a CNN is not set up to learn the temporal patterns of scenes associated with or without fire.

3.3 LSTM Classifier

Given that a video comprises a sequence of images collected over time, a time element was introduced into the learning process. A third method was evaluated to detect fire using LSTM neural-network architecture. LSTMs are a type of recurrent neural network (RNN) that does not directly apply convolutions or feature extractions to incoming data and are capable of carrying forward temporal information or learning over time. First proposed in 1997 (Hochreiter and Schmidhuber 1997), LSTMs can learn long-term dependencies given the increased complexity of the network layers (compared to the more simplistic network structure of RNNs) and have been applied to YouTube-8M datasets in previous analyses with good results (Li et al. 2017). The short-term memory in the LSTM is exhibited through

persistent previous information that is used in the current neural network. It also mitigates the vanishing-gradient problem, which is the networks inability to learn because the updates to the weights within the nodes become too small (i.e., insignificant) to alter the output (a process known as fading). The LSTM does the mitigation by using a series of gates contained in memory blocks that regulate the flow of information.

3.3.1 Method

A typical LSTM unit is composed of a cell, an input gate, an output gate, and a forget gate. The cell maintains information learned over the input time intervals while the gates are used to regulate the sequential or time-based features flowing into and out of the cell. The input gate scales input to the cell and behaves as a write operation. The output gate scales output to cells and behaves as a read operation when the model accesses cell information. The forget gate scales old cell values and behaves as a reset operation. Each of the gates in the LSTM unit behaves as a switch to control read/write operations and thus gives the effect of long-term memory to the model.

For this analysis, an LSTM model was developed using the TensorFlow library. It was configured to input 10 YouTube-8M video-frame feature vectors, versus the single frame feature input used in the CNN model. This was done to give the model a temporal sequence of 10 seconds because YouTube-8M features are derived from videos decoded to 1 fps. The same training and validation approach as the methodology described in Section 3.2.2 was applied for LSTM model development.

In addition to training the LSTM based on the YouTube-8M data, an additional method was implemented to modify the 62 test videos from their native video-file format when applied to test the LSTM after training and validation. This was performed to extract fire-region pixels from component video frames using color feature extraction described in Section 3.1 and generate image subsets containing suspect flame pixels. This resulted in a hybrid color-feature extraction and LSTM method. The reason for this is that the compressed feature vectors available in the YouTube-8M dataset are likely focused on a given labelled object. For example, a feature vector labeled as a car is likely produced from a cropped image of a car with little background information. A fire vector likely is derived from a cropped image showing only flame colors and structure. Subsetting an image using color-feature extraction will generate subset images that better match YouTube-8M training and validation features while reducing background complexity that could harm model accuracy. The steps for color feature extraction and image subsetting were developed follows:

- Apply a rules-based binary classification, as described in Section 3.1, to each pixel matching a flame-color signature. The left of Figure 6 shows a video frame with flames and the resultant output (right) of the rules-based classification depicting pixels matching the flame signature as white.
- A convolution is applied to compute a binary output with extracted locations that have the highest number of pixels matching a flame color signature (Figure 10). An arbitrary minimum threshold of 300 pixels in the region was applied to designate an image section as fire or no fire. This threshold was based on exploratory analysis to find an appropriate balance of potential fire pixels and no fire pixels in a given region to maximize computational performance and model accuracy.
- Subset image regions with fire-matching pixels are generated containing pixels matching flame color signatures. Figure 11 shows two subset images with fire (left) and another two subset regions with color features matching the flame color signature (right).

The subsets were then preprocessed using 1024 feature extraction, as explained in Section 2.1, to generate model input vectors matching YouTube-8M configuration. Since the LSTM was configured for 10 sequential frames, only video segments with 10 sequential frames generating subsets were passed to the 1024 feature extraction and then to the LSTM classifier.

To support video-level classification, a sliding-window approach was used where each subsequent 10-second window had a 2-second overlap to cover the entire video. A 2-second overlap prevents the

model from having to process excessively redundant information while still having temporal resolution to capture high temporal frequency changes that can occur over short periods of time. This keeps any events from being lost when windows overlap. When most of the sliding windows are classified as fire, the videos were classified as fire.

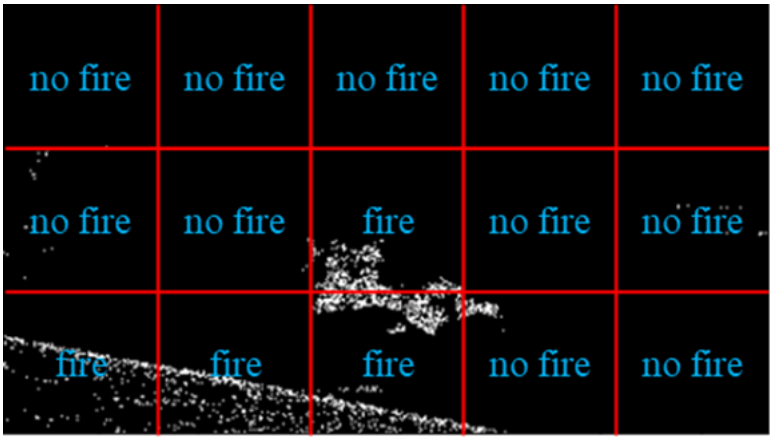


Figure 10. Convolution output of binary classification with pixels count meeting the 300 pixels threshold value.

(a) Subset 1 (b) Subset 2 (c) Subset 3 (d) Subset 4

Figure 11. Subset image outputs with regions dominated by pixels matching a flame color signature.

3.3.2 Results

The LSTM model architecture coupled with the color feature extraction image subsetting, and data-preprocessing methods applied to the testing dataset performed better than the baseline methods referenced above. Figure 12 shows the LSTM model accuracy metrics during training and validation with the YouTube-8M dataset. Training and validation on the YouTube-8M yielded a validation accuracy of 98% before accuracy improvements stopped at the 26th epoch.

When applied to the 62 test videos dataset, the LSTM classifier achieved the highest accuracies at both window- and video-levels. At the window-level, it achieved an accuracy of 95% success on fire classification, and a 85% accuracy on no-fire classification (Table 5). Only 5% of the frames with fire were misclassified as no-fire, versus 15% misclassified as fire when fire was not present in the scene. This indicates a slight skew towards false-positive error with even fewer occurrences of false-negative error. This is desirable in an industrial setting, because of the consequences of a false negative (i.e. fire occurred but was not detected).

Table 6 shows the confusion matrix of the video-based classification-accuracy metrics for the method. The classifier was able to achieve a higher validation accuracy. The model achieved a 100%

accuracy for fire classification, coupled with an accuracy of 92% accuracy for the no-fire classification. The classifier resulted in no false-negative error and only 8% of videos were classified as false positives.

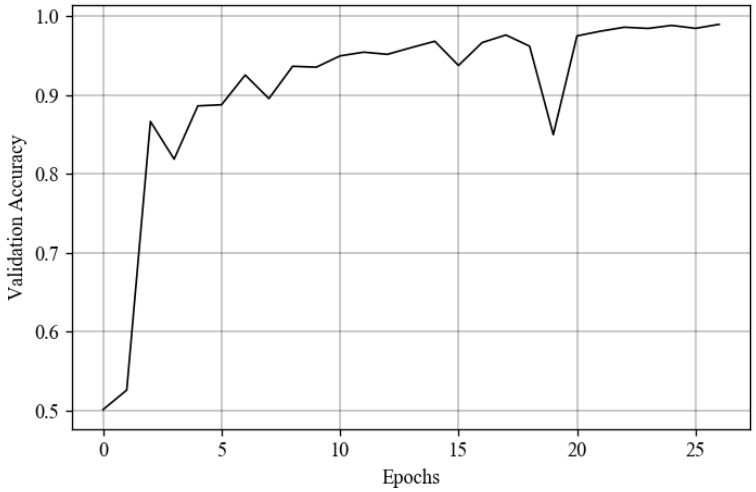


Figure 12. Accuracy growth of the LSTM neural network.

Table 5. Window-based confusion matrix for the LSTM classifier.

		Predicted	
		Fire	No Fire
True Value	Fire	0.95	0.05
	No Fire	0.15	0.85

Table 6. Video-based confusion matrix for the LSTM classifier.

		Predicted	
		Fire	No Fire
True Value	Fire	1.00	0.00
	No Fire	0.08	0.92

4. CONCLUSIONS

For this analysis, various sources of publicly available datasets containing scenes approximating industrial settings with and without fire were investigated and aggregated. They consist of YouTube-8M datasets that were downloaded and used to train and validate the neural-network modelling approaches used in this analysis. Additional video datasets available from previously published literature sources and an online repository provided by Yahoo (YFCC100m and FiSmo) were also downloaded and aggregated to test the color feature extraction efforts and isolate fire regions in a frame.

After data aggregation, two baseline classifiers were evaluated on industrial scenes approximating an NPP. The first was comprised of a color-feature extraction classifier using colors of pixels associated with fire and thresholding functions found in the literature. This was applied to the YFCC100m and FiSmo dataset because the method required the actual video to apply. The second baseline model consisted of a CNN classifier trained on the YouTube-8M. Its training and validation results indicated it was not suitable for this classification task. Neither method resulted in satisfactory results for the scope of this application.

A third, temporal method was developed and evaluated incorporating feature-color extraction and a data preparation methodology to subset image frames from the video datasets and format them to feed into a trained LSTM neural-network classifier. The LSTM classifier was trained with the same YouTube-8M dataset as the CNN model. The accuracy achieved by this method was 0% missed positives and 8% false positives. The overall success of the LSTM classifier is likely derived from the model's capability to account for temporal changes in video-image sequences. It is also likely the image subsetting, to remove background variance, resulted in model inputs better suited to the YouTube-8M features with which it was trained. The image subsetting also helped mitigate the effect of changing fire locations within a scene by providing a zoom-in effect, which helped standardize the test inputs to the training data. This makes the developed classifier potentially suitable for a complex environment given the potential everchanging variances related to fire size, distance from a camera sensor, image resolution, and background complexity.

5. FUTURE WORK

Through not targeted in this stage of the research, the method developed is expected to be able to detect smoke, assuming a smoke training dataset is developed. However, the YouTube-8M dataset lacks labeled smoke features. Hence, additional work would be needed to compile and curate a suitable training and validation dataset. Once developed, this can be combined with previous research using a rules-based approach (e.g. Chen et al. 2006) to detect and bound potential smoke.

The method developed can also be equipped on IR fire-detection cameras. IR cameras are used to detect fires and can be equipped with machine-learning methods to identify fire features within an IR image from noise such as a hot light bulb.

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. As a result, a solution that has relatively low false-positive detection probability while maintaining a high fire detection probability is desired. This could benefit from the incorporation of multimodal sensors and data fusion. Because varying sensor technologies have specific strengths and weaknesses relative to specific tasks, combining designs into multimodal sensors and, potentially, ensemble learners could improve detection accuracies while minimizing false-positive rates. Some examples of other sensors to fuse into the decision making are

- A nighttime fire and smoke detection system that incorporates an active laser light projected into the field view (Ho 2013). If aerosol particles such as those comprising smoke are accumulating in the air, the resulting light-scattering pattern will change over time, resulting in a differing return signal. The spectral, diffusing, and scattering characteristics of the return signals can then be classified using an SVM classifier.
- Fire-detection sensors using long-wave infrared (LWIR) light. LWIR is able to pass through smoke and detect flame features that would be otherwise obscured in RGB space. In LWIR space, flames appear as bright pixel regions that can be combined with known temporal flame-flicker processes to support classification (Töreyin et al. 2007).

Additionally, integrating a machine-learning method into a process to meet a licensing requirement necessitates unboxing the machine-learning black box, i.e., explaining the rationale behind the decision making and/or providing extensive validation and cases for the regulator of the ability to replace the human fire watch.

6. REFERENCES

Abadi, M., P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean et al., 2016. "TensorFlow: A System for Large-Scale Machine Learning," In *12th USENIX Symposium on Operating Systems Design and Implementation*, pp. 265-283.

Abu-El-Haija, S., N. Kothari, J. Lee, P. Natsev, G. Toderici, B. Varadarajan, and S. Vijayanarasimhan, 2016. *Youtube-8m: A large-scale video classification benchmark*, [online] Last Accessed: August 30th, 2019. Link: https://arxiv.org/abs/1609.08675.

Achanta, R., A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, 2012. "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE transactions on pattern analysis and machine intelligence* Vol. 34, No. 11, pp. 2274–2282.

AZO Sensors, 2017, "A Guide to Optical Flame Detection—How UV, IR and Imaging Detectors Work," [website]. Last accessed August 30th, 2019. Link: https://www.azosensors.com/article.aspx?ArticleID=815

Borges, P., and E. Izquierdo, 2010. "A probabilistic approach for vision-based fire detection in videos," *IEEE transactions on circuits and systems for video technology* Vol. 20, No. 5, pp. 721–731.

Cazzolato, M. , L. Avalhais, D. Chino, J., Ramos, J. de Souza, J. Rodrigues-Jr, and A. Traina, 2017. "FiSmo: A Compilation of Datasets from Emergency Situations for Fire and Smoke Analysis." In *Proceedings of the satellite events*.

Çelik, T., and H. Demirel, 2009. "Fire detection in video sequences using a generic color model," *Fire Safety Journal* Vol. 44, No. 2, pp. 147–158. doi:https://doi.org/10.1016/j.firesaf.2008.05.005

Çetin, A. , K. Dimitropoulos, B. Gouverneur, N. Grammalidis, O. Günay, Y. Habiboğlu, and S. Verstockt, 2013. "Video fire detection–review," *Digital Signal Processing* Vol. 23, No. 6, pp. 1827–1843.

Chen, J., Y. He, and J. Wang, 2010. "Multi-feature fusion based fast video flame detection," *Building and Environment* Vol. 45, No. 5, pp. 1113–1122.

Chen, T., Y. Yin, S. Huang, and Y. Ye, 2006. "The smoke detection for early fire-alarming system base on video processing," In *Proceedings of the 2006 International Conference on Intelligent Information Hiding and Multimedia*, pp. 427-430.

Dunnings, A., and T. Breckon, 2018. "Experimentally Defined Convolutional Neural Network Architecture Variants for Non-Temporal Real-Time Fire Detection," In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pp. 1558-1562.

Rawat, Waseem, and Z. Wang, 2017. "Deep convolutional neural networks for image classification: A comprehensive review." *Neural computation* Vol. 29, No. 9 pp. 2352-2449.

Google developers, 2019. "ML Practicum: Image Classification, Introducing Convolutional Networks," [website]. Last accessed August 30th, 2019. Link: https://developers.google.com/machine-learning/practica/image-classification/convolutional-neural-networks

Ho, C., 2013. "Nighttime fire or smoke detection system based on a support vector machine," *Mathematical Problems in Engineering*, Vol. 2013, Article ID 428545. Link: http://dx.doi.org/10.1155/2013/428545

Hochreiter, S., and J. Schmidhuber, 1997. "Long Short-term Memory," *Neural computation* Vol. 9, No. 8, pp. 1735–1780. doi:10.1162/neco.1997.9.8.1735

Hu, C., P. Tang, W. Jin, Z. He, and W. Li, 2018. "Real-Time Fire Detection Based on Deep Convolutional Long-Recurrent Networks and Optical Flow Method," In *2018 37th Chinese Control Conference (CCC)*, pp. 9061-9066.

Jolliffe, I., and J. Cadima, 2016. "Principal component analysis: a review and recent developments." *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences.* Vol. 374, No. 2065, ID. 20150202.

Li, F., C. Gan, X. Liu, Y. Bian, X. Long, Y. Li, and S. Wen, 2017. *Temporal modeling approaches for large-scale youtube-8m video understanding*, [online] Last Accessed: August 30th, 2019. Link: https://arxiv.org/abs/1707.04555

Muhammad, K., J. Ahmad, I. Mehmood, S. Rho, and S. W. Baik, 2018. "Convolutional Neural Networks Based Fire Detection in Surveillance Videos," *IEEE Access* Vol. 6, pp. 18174–18183.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, and V. Dubourg, 2011. "Scikit-learn: Machine learning in Python," *Journal of machine learning research* Vol. 12 (Oct), pp. 2825–2830.

Phillips III, W., M. Shah, and N. da Vitoria Lobo, 2002. "Flame recognition in video," *Pattern recognition letters* Vol. 23, Nos. 1–3, pp. 319–327.

Szegedy, C., W. Liu, Y. Jia, P. Sermanet, S. Reed, D., Anguelov, and A. Rabinovich, 2015. "Going deeper with convolutions," *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9.

Thomee, B., D. Shamma, G., Friedland, B. Elizalde, K. Ni, et al., 2015. *YFCC100M: The new data in multimedia research*, [online] Last Accessed: August 30th, 2019. Link: https://arxiv.org/abs/1503.01817

Thou-Ho, C., W. Ping-Hsueh, and C. Yung-Chuen, 2004. "An early fire-detection method based on image processing," In*2004 International Conference on Image Processing*, Vol. 3, pp. 1707-1710.

Töreyin, B. , R. Cinbis, Y. Dedeoglu, and A. Cetin, 2007. "Fire detection in infrared video using wavelet analysis," *Optical Engineering* Vol. 46, No. 6, 067204.

Töreyin, B. , Y. Dedeoglu, and A. Cetin, 2006. "Contour based smoke detection in video using wavelets," In *European Signal Processing Conference*, pp. 1-5.

U.S. Department of Labor,2019. "Occupational Safety and Health, General Safety and Health Hot Work," [Website]. Last accessed August 30, 2019, link: https://www.osha.gov/SLTC/etools/oilandgas/general_safety/hot_work_welding.html

U.S. Nuclear Regulatory Commission, 2009. Guide 1.189, "Fire Protection for Operating Nuclear Power Plants," Last accessed August 30, 2019, link: https://www.nrc.gov/docs/ML0925/ML092580550.pdf