A Fire Hazard Identification and Evaluation Framework for Fire Extinguishing Applications

Irene Alisjahbana
Civil & Environmental Engineering
Stanford University, CA 94305
alisjahbana@stanford.edu

Vivian Wong
Civil & Environmental Engineering
Stanford University, CA 94305
vwwong3@stanford.edu

Abstract

In the future, fire fighting robots might completely replace human fire fighters. These robots would need to accurately distinguish if objects are on fire and be able to prioritize which hazards to handle first based on their combustibility. In this project, we propose a framework to be able to evaluate the fire hazard level of any given image the robot captures. To do this, two deep convolution neural networks are deployed to distinguish whether an image contains fire and the material catching on fire. Ultimately, the framework will classify each image into one of the three discrete fire hazard levels: no hazard, moderate hazard, and high hazard.

1. Introduction

Motivated by the Notre Dame fire tragedy that occurred a couple months ago, we realize that in the future, it is possible that human fire fighters would be replaced by firefighting robots to improve fire-extinguishing capacity and reduce human casualties. In fact, a firefighting robot named Colossus was deployed during the scene and helped extinguish the flames [9].

One of the limitations of current firefighting robots are that they are manually controlled with joysticks, and still require humans to manually detect objects to extinguish. In order for firefighting robots to be effectively deployed in the future, they need to be more autonomous. In that case, the robots would need to be able to distinguish if objects are on fire or not. In addition, given the limited number of robots and resources (such as water and time), the robots must be able to prioritize which fires to extinguish first. For example, the robots should prioritize extinguishing wood that is on fire compared to other materials as it is the most combustible. In the Notre Dame incident, the wooden sections of the church were engulfed in fire much sooner than the stone sections.

Given this motivation, we aim to create a new framework that is capable of evaluating the fire hazard level of an image received by a fire-fighting robot's camera, thereby prioritizing which fire to extinguish first. The framework 1) performs a binary classification to identify the existence of any fire, and 2) performs a multi-class classification of the materials contained in the fire. Based on those information, we can generate a rating on the seriousness of the fire hazard.

1.1. Statement of Work

The rest of the paper is organized as follows. Section 2 summarizes the related work on fire identification using CNNs. Section 3 describes the technical approach and methods we intend to use to solve the problem. Section 4 describes the dataset we used to train our models. In Section 5, we show our results and discuss our findings. Finally, section 6 concludes our report and suggest possible directions for future work.

2. Related Work

Various algorithms have been proposed to detect fire in both video and images. Horng and Peng [10] used neural networks to detect fire in images using flame color features. Namozov and Cho [15] proposed a novel CNN algorithm using adaptive piecewise linear units in order to accurately detect fire images using limited data. Work has also been done on using smoke as a way to detect fire in images and videos [14, 7, 4, 5]. In addition, there has been attempts in detecting fire from real-time footage [18, 2, 6, 11].

However, the goals of most existing approaches are only to detect whether an image contains smoke or fire. To the best of our knowledge, there is no additional work being done on fire hazard evaluation, or classifying sources of fires.

There have been some attempts to classify materials with machine learning techniques. In [12], they were able to achieve an accuracy of 54%. This accuracy is so low that we decided to not use it as our baseline.

As for the assessment of fire hazards, [16] ranked materials based on their combustibility. We believe that more combustible materials will cause fire to spread more easily, and are therefore given higher hazard ratings, indicating that a fire extinguishing agent needs to resolve them sooner.

In summary, to the best of our knowledge, existing work have not tackled this problem, and other related work either have low accuracies or very different applications.

3. Methods

3.1. Problem Formulation

The problem is to be able to evaluate how urgent a fire hazard needs to be resolved by giving a potential hazard a rating. This information should be generated as the robot processes images from its camera. The input and output of the problem is thus defined as follows.

Input: RGB 2D images

Output: A hazard rating of:

- High hazard
- Moderate hazard
- No hazard

3.2. Framework

Our framework consists of 2 major parts. It will first receive images as input where the model will classify whether the image contains fire or not. For images that contains fire, it will then detect the materials that are within that image. The framework would then output a hazard level (no fire, moderate hazard and high hazard) based on the material and the presence of fire in the image. The general framework can be seen in Figure 1

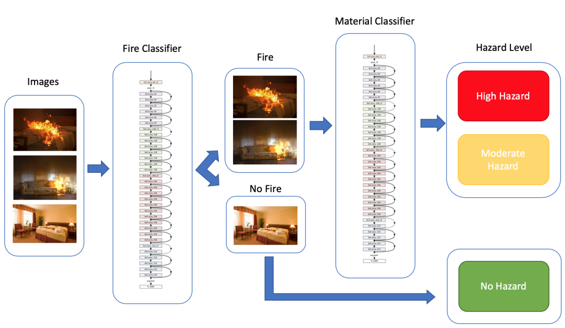

Figure 1. General Framework

Both the fire classifier and the material classifier are CNN's that are trained offline prior to the deployment into the fire. When a fire extinguishing device goes in, it should already have the models pre-loaded, and can therefore run inference on the spot. Since inference is a fast process, our primary goal is to optimize the models' accuracy, while not investing in too much time on training given the scope of this project.

3.3. Baseline

Due to the fact that, to the best of our knowledge, there is no existing work on image-based fire hazard evaluation, we set two baselines to evaluate our performance for the two major parts of our framework.

The first baseline is for the binary fire classifier, which is obtained from [10]. Their model is able to obtain an accuracy of 96.47%.

The second baseline is for the material classifier. The image classifier classifies input images into one of the 7 material categories. This model is a ResNet-18 model pretrained on ImageNet data, then trained on clean material data from the Flickr Material Databse (FMD) for 25 epochs. The baseline's test data is also from FMD. It is compared against our classifier which will be tested on images containing fire.

3.4. Starter Code and Project Contribution

To build this framework, we utilize starter codes from PyTorch's Transfer Learning Tutorial[3]. The starter code allowed us to import pretrained ResNet models from PyTorch, and taught us how to freeze layers. Our contribution achieves the follow:

- data collection (See Section 4.1, 4.2)
- manual labeling of some of the data (See Section 4.3)
- modifying parameters and data augmentation techniques in the starter code to adjust better to our data
- experiment on the effect of various CNN architectures and transfer learning approaches (freezing intermediate layers vs. not freezing)
- streamlining the end-to-end pipeline to follow the framework stated in Section 3.2
- extracting qualitative and quantitative results for reporting

3.5. Data Augmentation

In addition to the database that we used for our model, we also did data augmentation to increase the number of images in our training dataset. For the fire and materials (baseline) classifier, we used random cropping, and random horizontal flipping to generate more data. On the other hand, for our final model, we only used random horizontal flipping. We believe that random cropping would not be a good idea as the fire might only be concentrated on a specific area of the image.

3.6. Model Architectures

Given our offline training and online inference pipeline, we tested and evaluated various ImageNet pre-trained models provided by PyTorch[1], and eventually decided to select ResNet models as our final model architectures. Since the material classifier takes on a much more challenging task than the fire classifier does, a ResNet-18 model achieves excellent result on simple fire binary classification, but is not sufficient for the material classifier. Therefore, we employed ResNet-152 on the material classifier.

Although ResNet is not the architecture that gives the highest accuracy, ResNet-18 models are very fast, and ResNet-152, although slower, still maintains high accuracy. As stated in [8] ResNet models are convolutional layers stacked in building blocks, as shown in Figure 2.

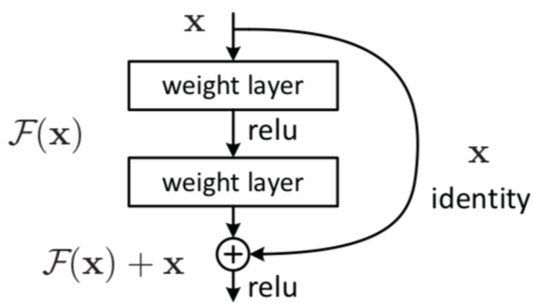

Figure 2. A Single ResNet Building Block [8]

As an example, a ResNet-18 model would therefore look like Figure 3.

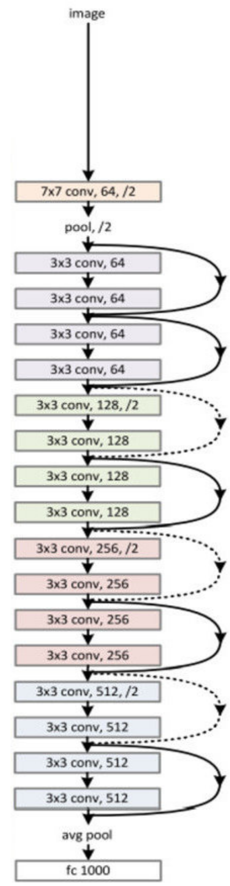

Figure 3. ResNet-18 Architecture [13]

3.7. Loss Function

The loss function used to solve the above-mentioned classification problems is the cross entropy loss, given by (1) below.

$$L_i = -\log(\frac{\exp(f_{y_i})}{\sum_j \exp(f_j)}) \tag{1}$$

Given a total number of classes C, L_i is the loss for data point i in the training batch, j is a class index in {0,...,C-1}, y_i is the ground truth class for data point i, and f_{class} is the score of a class index, where $class \in \{y_i, j\}$.

Backpropagating with this loss function allows us to optimize for classification accuracy.

3.8. Transfer Learning

Using model with pre-trained weights save time and computing resources in training. This is especially important given the limited size of our dataset.

We used ResNet models pre-trained on ImageNet classes. We experimented with training by both freezing intermediate layers (therefore only learning the weights of the final fully-connected layer) as well as not freezing any layers.

4. Dataset

The data required for our framework can be divided into two: fire/non fire images and images containing materials from 7 categories. Each dataset was split into 70/30 for training and validation.

4.1. Fire

The dataset used for the fire/non-fire images was obtained from a GitHub repo (https://github.com/cair/Fire-Detection-Image-Dataset). This database contains 110 images of fire and 600 non-fire images and has been curated to reciprocate real world situations. The fire images consists of different types of fire in a variety of situation. Some sample images from this database can be seen in Figure 4

4.2. Material

The dataset used to classify materials was obtained from the Flickr Material Database (FMD) [17]. Originally, this

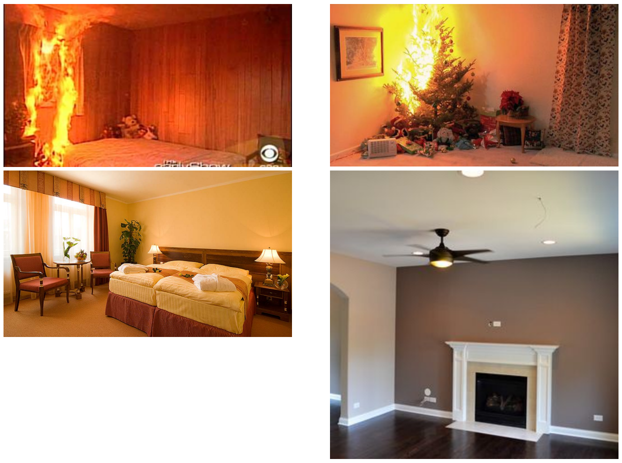

Figure 4. Sample images from the fire image database: fire images (top) and non-fire images (bottom)

database consists of 10 different categories that were obtained and selected manually from Flickr. For the purposes of our project, we used 7 of the categories provided: Fabric, Glass, Leather, Metal, Plastic, Stone and Wood. Each category consists of 100 images each that has a variety of different illumination condition, composition, and colors. Some sample images from this database can be seen in Figure 5.

Figure 5. Sample images from the material image database: (left to right, top to bottom) fabric, glass, leather, plastic, stone, metal and wood

4.3. Fire-burned Materials

Since no previous work has been done on identifying fire sources, there is no existing dataset that provides labels for the fire sources. Therefore, in order to generate enough training and validation data, we manually labeled the fire images in Section 4.1 by identifying the type of material that is being burned, marking each major fire source as one of the categories specified in Section 4.2.

Since there are no or very few available images of glass, leather and plastic being burned, we removed these categories from the classifier. Materials are therefore classified into one of the 4 categories, wood, fabric, stone, and metal, which are still challenging tasks for any classifier, since fire is often obstructing and tainting nearby objects.

5. Results and Discussion

As mentioned before, we implemented our framework using Pytorch, and used the transfer learning tutorial for Pytorch as a starter code for the project. To evaluate our models, the primary metrics we used are validation accuracy as well as confusion matrix. We also validate our results qualitatively by manually visualizing the images.

5.1. Fire Classifier

The first classifier of our framework was trained using the fire/non-fire dataset. Here, we used a ResNet-18 model pretrained on ImageNet. At first, we trained our model by using all layers, and though the training process did not take too long, we also trained the model by freezing all the layers except the last fully connected layer. The accuracy obtained for both methods were comparable, with the first method producing a higher accuracy but slower training time.

The optimizer we used was the Stochastic Gradient Descent (SGD) with a learning rate of 0.001, and a momentum of 0.9. These hyperparameters were chosen because these values have been shown to generally give good results. We also decayed the learning rate by a factor of 0.1 every 7 epochs. Again, these values were chosen as they are typical values found in literature.

Using these values, we were able to obtain a best validation accuracy of 97.89%, thereby slightly outperforming the baseline model. Though ImageNet doesn't have an explicit class for fire, we believe that we were able to obtain high accuracy for fire/non-fire classification because fires have very sharp and distinct orange pixels that enables the network to learn quickly even with very few fire images.

The percentage accuracy for fire images is 87.88% and non-fire images 99.36%. The confusion matrix can be seen below:

True Class	Classified as Fire	Classified as Non-Fire
Fire	87.88%	12.12%
Non-Fire	0.64%	99.36%

Table 1. Confusion matrix of the fire/non-fire binary classifier after 25 epochs on a NVIDIA GeForce GTX 1080 Ti GPU

We also considered using deeper networks such as Inception, ResNet-50 or ResNet-152, and other optimizers that have been shown to converge faster such as Adam. However, because of our good accuracy, we decided to stay with simpler and faster architectures.

Classifier	Best Val. Acc.	Training Time
No Freezing	**97.37%**	1m 23s
Freezed Layers	96.84%	**1m 13s**
Baseline [10]	96.47%	-

Table 2. Transfer Learning Results of the fire/non-fire binary classifier after 25 epochs on a NVIDIA GeForce GTX 1080 Ti GPU

Figure 6. Sample images from the output of our classifier using images from the fire dataset. Top two: correctly classified; Bottom two: incorrectly classified

In general, we believe that our model was able to accurately classify which images contain fire or not, even with harder to classify images, such as those of sunsets, or the Northern Lights (which contains very bright orange pixels and texture similar to fire).

5.2. Material Classifier

After classifying whether the image contains fire or not, we fed our second network (the material classifier) images from the output of the fire classifier, in particular, the images containing fire.

As mentioned before, though our original material database contains 7 different classes of material, we decided to use only 4 classes for our final model: fabric, metal, stone and wood. In addition to the fact that there are few few labeled images of glass, leather and plastic, we also assume that these four materials are the most relevant to fire-fighting in buildings such as the Notre Dame cathedral.

Here, we used a ResNet-152 model pre-trained on ImageNet. We improved this from using a ResNet-18 model that became our baseline model because we realized the network was not deep enough to learn all the features that are necessary to accurately classify the images and produce good results. This baseline model was trained on images from the FMD database for 30 epochs.

Similar to the fire classifier, we used the SGD optimizer with a learning rate of 0.001 and momentum of 0.9. We also decayed the learning rate by a factor of 0.1 every 7 epochs.

Once the model has been trained, we used that model to do second level of transfer learning by training the images to images of fire with classified materials in it. The same hyperparameters were used for this model also.

We were able to obtain a best validation accuracy of 72.22% for our final model, which is slightly higher than our baseline model. We believe the fact that the accuracy value is not that much higher than the baseline model is because images of materials on its own is very different from images of those same materials on fire.

Classifier	Best Val. Acc.	Training Time
No Freezing	**77.78%**	0m37s
Freezed Layers	66.67%	**0m22s**
Baseline	77.14%	5m23s

Table 3. Transfer Learning Results of the material classifier after 30 epochs on a NVIDIA GeForce GTX 1080 Ti GPU

The confusion matrix can be seen below. The lowest percentage accuracy is obtained for the fabric class with 60%. On the other hand, wood has an accuracy of 100% which means that all images that contain wood was accurately classified. The lower percentage of accuracy in the other classes is caused by the fact that fire obstructs a lot of view and is very different from images of the material itself without fire.

True Class	Fabric	Metal	Stone	Wood
Fabric	60.00%	20.00%	0	20.00%
Metal	0	66.67%	0%	33.33
Stone	0	0	75.00%	25.00%
Wood	0	0	0	100.00%

Table 4. Confusion Matrix for the material classifier after 30 epochs on a NVIDIA GeForce GTX 1080 Ti GPU

Class	Accuracy
Fabric	60%
Metal	66.67%
Stone	75.0%
Wood	100.0%

Table 5. Percentage Accuracy for Each Class

Looking even more closely at the results, we notice that materials are often false-positively classified as wood. This might be due to the fact that usually wood in training images are often brown-orange colored, which a lot of things in fire would resemble.

In general, we notice that when fire obstructs a lot of view or when there are a multiple materials in one image, the classifier normally fails.

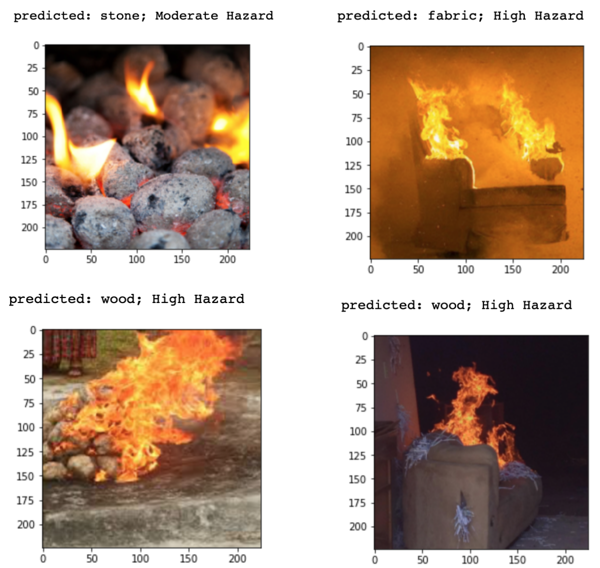

Figure 7. Sample images from the output of our classifier using images from the fire dataset. Top two: correctly classified; Bottom two: incorrectly classified

Figure 7 also shows the final output of the framework. In the end, we output a hazard rating based on the source of fire. The fact that our model sometimes misclassifies fabric and wood is fortunately not a major problem as both materials will be classified as high hazard. Distinguishing the materials accurately will be more important if the hazard levels are classified into more detailed classes.

6. Conclusion and Future Work

In conclusion, this project implements and tests a framework that input images and output a hazard level rating. Our work has tackled a problem that very few or none existing work has tackled. We were able to generate labels with acceptable accuracy with very few image data (only around 100 fire images). Compared to the material classifier by [12], our results are outstanding, improving their accuracy by at least 6% even for our worst material category. We also managed to improve the two baselines we set for the two classifiers.

We also learned that freezing intermediate layers reduces transfer learning effectiveness but improves computational speed significantly.

However, there are still plenty of room for improvements. We suffer from overfitting since we only have a small number of fire source labels (our training accuracy in the end was 92.11%, much higher than the best validation accuracy in Section 5). Future work could include hiring people to annotate labels for fire sources. In addition, better results might be obtained by first isolating the fire in each image before feeding it to the network to help reduce the noise from other materials. We believe that this is crucial for any successful and efficient autonomous fire extinguishing device that uses cameras to detect fire.

References

[1] Torchvision.models-pytorch master documentation.

[2] T.-H. Chen, C.-L. Kao, and S.-M. Chang. An intelligent real-time fire-detection method based on video processing. In *IEEE 37th Annual 2003 International Carnahan Conference onSecurity Technology, 2003. Proceedings.*, pages 104–111. IEEE, 2003.

[3] S. Chilamkurthy. Transfer learning tutorial.

[4] Y. Chunyu, F. Jun, W. Jinjun, and Z. Yongming. Video fire smoke detection using motion and color features. *Fire technology*, 46(3):651–663, 2010.

[5] Y. Chunyu, Z. Yongming, F. Jun, and W. Jinjun. Texture analysis of smoke for real-time fire detection. In *2009 Second International Workshop on Computer Science and Engineering*, volume 2, pages 511–515. IEEE, 2009.

[6] N. Dedeoglu, B. U. Toreyin, U. Gudukbay, and A. E. Cetin. Real-time fire and flame detection in video. In *Proceedings.(ICASSP'05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.*, volume 2, pages ii–669. IEEE, 2005.

[7] S. Frizzi, R. Kaabi, M. Bouchouicha, J.-M. Ginoux, E. Moreau, and F. Fnaiech. Convolutional neural network for video fire and smoke detection. In *IECON 2016-42nd Annual Conference of the IEEE Industrial Electronics Society*, pages 877–882. IEEE, 2016.

[8] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[9] P. Holley. Firefighters had a secret weapon when notre dame caught fire: A robot named colossus, 2019.

[10] W.-B. Horng and J.-W. Peng. Image-based fire detection using neural networks. In *9th Joint International Conference on Information Sciences (JCIS-06)*. Atlantis Press, 2006/10.

[11] W.-B. Horng, J.-W. Peng, and C.-Y. Chen. A new image-based real-time flame detection method using color analysis. In *Proceedings. 2005 IEEE Networking, Sensing and Control, 2005.*, pages 100–105. IEEE, 2005.

[12] D. Hu and L. Bo. Toward robust material recognition for everyday objects. BMVC, 2011.

[13] C. Kwan, B. Chou, J. Yang, and T. Tran. Compressive object tracking and classification using deep learning for infrared videos. 04 2019.

[14] Z. Li, A. Khananian, R. H. Fraser, and J. Cihlar. Automatic detection of fire smoke using artificial neural networks and

threshold approaches applied to avhrr imagery. *IEEE Transactions on geoscience and remote sensing*, 39(9):1859–1870, 2001.

[15] C. Y. Namozov, A. An efficient deep learning algorithm for fire and smoke detection with limited data. *Advances in Electrical and Computer Engineering*, 18(4), 2018.

[16] N. O. E. Services. Classification of building materials according to combustibility. 2011.

[17] R. R. . A. E. H. Sharan, L. Material perception: What can you see in a brief glance? *Journal of Vision*, 14(9), 2014.

[18] B. U. Töreyin, Y. Dedeoğlu, U. Güdükbay, and A. E. Cetin. Computer vision based method for real-time fire and flame detection. *Pattern recognition letters*, 27(1):49–58, 2006.