# Deep CNN-ELM Hybrid models for Fire Detection in Images

Jivitesh Sharma, Ole-Christopher Granmo, and Morten Goodwin

Center for Artificial Intelligence Research, University of Agder,
Jon Lilletuns vei 9, 4879 Grimstad, Norway
jivitesh.sharma@uia.no
ole.granmo@uia.no
morten.goodwin@uia.no

B

**Abstract.** In this paper, we propose a hybrid model consisting of a Deep Convolutional feature extractor followed by a fast and accurate classifier, the Extreme Learning Machine, for the purpose of fire detection in images. The reason behind using such a model is that Deep CNNs used for image classification take a very long time to train. Even with pre-trained models, the fully connected layers need to be trained with backpropagation, which can be very slow. In contrast, we propose to employ the Extreme Learning Machine (ELM) as the final classifier trained on pre-trained Deep CNN feature extractor. We apply this hybrid model on the problem of fire detection in images. We use state of the art Deep CNNs: VGG16 and Resnet50 and replace the softmax classifier with the ELM classifier. For both the VGG16 and Resnet50, the number of fully connected layers is also reduced. Especially in VGG16, which has 3 fully connected layers of 4096 neurons each followed by a softmax classifier, we replace two of these with an ELM classifier. The difference in convergence rate between fine-tuning the fully connected layers of pre-trained models and training an ELM classifier are enormous, around 20x to 51x speed-up. Also, we show that using an ELM classifier increases the accuracy of the system by 2.8% to 7.1% depending on the CNN feature extractor. We also compare our hybrid architecture with another hybrid architecture, i.e. the CNN-SVM model. Using SVM as the classifier does improve accuracy compared to state-of-the-art deep CNNs. But our Deep CNN-ELM model is able to outperform the Deep CNN-SVM models. [1]

**Keywords:** Deep Convolutional Neural Networks, Extreme Learning Machine, Image Classification, Fire Detection

## 1 Introduction

The problem of fire detection in images has received a lot of attention in the past by researchers from computer vision, image processing and deep learning.

[1] Preliminary version of some of the results of this paper appear in Deep Convolutional Neural Networks for Fire Detection in Images, Springer Proceedings Engineering Applications of Neural Networks 2017 (EANN'17), Athens, Greece, 25-27 August

This is a problem that needs to be solved without any compromise. Fire can cause massive and irrevocable damage to health, life and property. It has led to over a 1000 deaths a year in the US alone, with property damage in access of one billion dollars. Besides, the fire detectors currently in use require different kinds of expensive hardware equipment for different types of fire [27].
What makes this problem even more interesting is the changing background environment due to varying luminous intensity of the fire, fire of different shades, different sizes etc. Also, the false alarms due to the environment resembling fire pixels, like room with bright red/orange background and bright lights. Furthermore, the probability of occurrence of fire is quite low, so the system must be trained to handle imbalance classification.
Various techniques have been used to classify between images that contain fire and images that do not. The state-of-the-art vision-based techniques for fire and smoke detection have been comprehensively evaluated and compared in [39]. The colour analysis technique has been widely used in the literature to detect and analyse fire in images and videos [4, 24, 31, 37]. On top of colour analysis, many novel methods have been used to extract high level features from fire images like texture analysis [4], dynamic temporal analysis with pixel-level filtering and spatial analysis with envelope decomposition and object labelling [40], fire flicker and irregular fire shape detection with wavelet transform [37], etc.
These techniques give adequate performance but are currently outperformed by Machine Learning techniques. A comparative analysis between colour-based models for extraction of rules and a Machine Learning algorithm is done for the fire detection problem in [36]. The machine learning technique used in [36] is Logistic Regression which is one of the simplest techniques in Machine Learning and still outperforms the colour-based algorithms in almost all scenarios. These scenarios consist of images containing different fire pixel colours of different intensities, with and without smoke.
Instead of explicitly designing features by using image processing techniques, deep neural networks can be used to extract and learn relevant features from images. The Convolutional Neural Networks (CNNs) are the most suitable choice for the task of image processing and classification.
In this paper, we employ state-of-the-art Deep CNNs for fire detection and then propose to use hybrid CNN-ELM and CNN-SVM models to outperform Deep CNNs. Such hybrid models have been used in the past for image classification, but the novelty of our approach lies in using state-of-the-art Deep CNNs like VGG16 and Resnet50 as feature extractors and then remove some/all fully connected layers with an ELM classifier. This models outperform Deep CNNs in terms of accuracy, training time and size of the network. We also compare the CNN-ELM model with another hybrid model, CNN-SVM and show that the CNN-ELM model gives the best performance.
The rest of the paper is organized in the following manner: Section 2 briefly describes the related work with CNNs for fire detection and Hybrid models for image classification. Section 3 explains our work in detail and section 4 gives

details of our experiments and presents the results. Section 5 summarizes and concludes our work.

## 2 Related Work

In this paper, we integrate state-of-the-art CNN hybrid models and apply it to the problem of fire detection in images. To the best of our knowledge, hybrid models have never been applied to fire detection. So, we present a brief overview of previous research done in CNNs used for fire detection and hybrid models separately in the next two sub-sections.



### 2.1 CNNs for Fire detection

There have been many significant contributions from various researchers in developing a system that can accurately detect fire in the surrounding environment. But, the most notable research in this field involves Deep Convolutional Neural Networks (Deep CNN). Deep CNN models are currently among the most successful image classification models which makes them ideal for a task such as Fire detection in images. This has been demonstrated by previous research published in this area.
In [7], the authors use CNN for detection of fire and smoke in videos. A simple sequential CNN architecture, similar to LeNet-5 [18], is used for classification. The authors quote a testing accuracy of 97.9% with a satisfactory false positive rate.
Whereas in [43], a very innovative cascaded CNN technique is used to detect fire in an image, followed by fine-grained localisation of patches in the image that contain the fire pixels. The cascaded CNN consists of AlexNet CNN architecture [17] with pre-trained ImageNet weights [28] and another small network after the final pooling layer which extracts patch features and labels the patches which contain fire. Different patch classifiers are compared.
The AlexNet architecture is also used in [34] which is used to detect smoke in images. It is trained on a fairly large dataset containing smoke and non-smoke images for a considerably long time. The quoted accuracies for large and small datasets are 96.88% and 99.4% respectively with relatively low false positive rates.
Another paper that uses the AlexNet architecture is [23]. This paper builds its own fire image and video dataset by simulating fire in images and videos using Blender. It adds fire to frames by adding fire properties like shadow, fore-ground fire, mask etc. separately. The animated fire and video frames are composited using OpenCV [2]. The model is tested on real world images. The results show reasonable accuracy with high false positive rate.
As opposed to CNNs which extract features directly from raw images, in some methods image/video features are extracted using image processing techniques and then given as input to a neural network. Such an approach has been used in [6]. The fire regions from video frames are obtained by threshold values in the

HSV colour space. The general characteristics of fire are computed using these values from five continuous frames and their mean and standard deviation is given as input to a neural network which is trained using back propagation to identify forest fire regions. This method performs segmentation of images very accurately and the results show high accuracy and low false positive rates.
In [11], a neural network is used to extract fire features based on the HSI colour model which gives the fire area in the image as output. The next step is fire area segmentation where the fire areas are roughly segmented and spurious fire areas like fire shadows and fire-like objects are removed by image difference. After this the change in shape of fire is estimated by taking contour image difference and white pixel ratio to estimate the burning degree of fire, i.e. no-fire, small, medium and large. The experimental results show that the method is able to detect different fire scenarios with relatively good accuracy.

B

### 2.2 Hybrid models for Image classification

The classifier part in a Deep CNN is a simple fully connected perceptron with a softmax layer at the end to output probabilities for each class. This section of the CNN has a high scope for improvement. Since it consists of three to four fully connected layers containing thousands of neurons, it becomes harder and slower to train it. Even with pre-trained models that require fine tuning of these layers. This has led to the development of hybrid CNN models, which consist of a specialist classifier at the end.
Some of the researchers have employed the Support Vector Machine (SVM) as the final stage classifier [1, 21, 25, 33, 38]. In [25], the CNN-SVM hybrid model is applied to many different problems like object classification, scene classification, bird sub-categorization, flower recognition etc. A linear SVM is fed 'off the shelf convolutional features' from the last layer of the CNN. This paper uses the OverFeat network [30] which is a state-of-the-art object classification model. The paper shows, with exhaustive experimentation, that extraction of convolutional features by a deep CNN is the best way to obtain relevant characteristics that distinguishes an entity from another.
The CNN-SVM model is used in [21] and successfully applied to visual learning and recognition for multi-robot systems and problems like human-swarm interaction and gesture recognition. This hybrid model has also been applied to gender recognition in [38]. The CNN used here is the AlexNet [17] pre-trained with ImageNet weights. The features extracted from the entire AlexNet are fed to an SVM classifier. A similar kind of research is done in [33], where the softmax layer and the cross-entropy loss are replaced by a linear SVM and margin loss. This model is tested on some of the most well known benchmark datasets like CIFAR-10, MNIST and Facial Expression Recognition challenge. The results show that this model outperforms the conventional Deep CNNs.
In 2006, G.B. Huang introduced a new learning algorithm for a single hidden layer feedforward neural network called the Extreme Learning Machine [13, 14]. This technique was many times faster than backpropagation and SVM, and

outperformed them on various tasks. The ELM randomly initializes the input weights and analytically determines the output weights. It produces a minimum norm least squares solution which always achieves lowest training accuracy, if there are enough number of hidden neurons. There have been many variants of ELM depending upon a specific application, which have been summarised in [12]. This led to the advent of CNN-ELM hybrid models, which were able to outperform the CNN-SVM models on various applications. The major advantage of CNN-ELM models is the speed of convergence. In [29], the CNN-ELM model is used for Wireless Capsule Endoscopy (WCE) image classification. The softmax classifier of a CNN is replaced by an ELM classifier and trained on the feature extracted by the CNN feature extractor. This model is able to outperform CNN-based classifiers.



The CNN-ELM model has also been used for handwritten digit classification [19, 22]. In [19], a 'shallow' CNN is used for feature extraction and ELM for classification. The shallow CNN together with ELM speeds up the training process. Also, various weight initialization strategies have been tested for ELM with different receptive fields. Finally, two strategies, namely the Constrained ELM (C-ELM) [44] and Computed Input Weights ELM (CIW-ELM) [35] are combined in a two layer ELM structure with receptive fields. This model was tested on the MNIST dataset and achieved 0.83% testing error. In [22], a deep CNN is used for the same application and tested on the USPS dataset.

A shallow CNN with ELM is tested on some benchmark datasets like MNIST, NORB-small, CIFAR-10 and SVHN with various hyper parameter configurations in [20]. Another similar hybrid model that uses CNN features and Kernel ELM as classifier is used in [9] for age estimation using facial features. Another application where a CNN-ELM hybrid model has been applied is the traffic sign recognition [41].

A different strategy of combining CNN feature extraction and ELM learning is proposed in [15]. Here, an ELM with single hidden layer is inserted after every convolution and pooling layer and at the end as classifier. The ELM is trained by borrowing values from the next convolutional layer and each ELM is updated after every iteration using backpropagation. This interesting architecture is applied to the application of lane detection and achieves excellent performance.

A comparative analysis of the CNN-ELM and CNN-SVM hybrid models for object recognition from ImageNet has been illustrated in [42]. Both these models were tested for object recognition from different sources like Amazon, Webcam, Caltech and DSLR. The final results show that the CNN-ELM model outperforms the CNN-SVM model on all datasets and using Kernel ELM further increases accuracy.

Using ELM as a final stage classifier does not end at image classification with CNNs. They have also been used with DBNs for various applications [3, 26].

## 3 The Fire Detector

In this paper, we propose to employ hybrid deep CNN models to perform fire detection. The AlexNet has been used by researchers in the past for fire detection which has produced satisfactory results. We propose to use two Deep CNN architectures that have outperformed the AlexNet on the ImageNet dataset, namely VGG16 [32] and Resnet50 [10]. We use these models with pre-trained ImageNet weights. This helps greatly when there is lack of training data. So, we fine-tune the ELM classifier on our dataset, which is fed the features extracted by the Deep CNNs.

### 3.1 Deep ConvNet Models

The Convolutional Neural Network was first introduced in 1980 by Kunihiko Fukushima [8]. The CNN is designed to take advantage of two dimensional structures like 2D Images and capture local spatial patterns. This is achieved with local connections and tied weights. It consists of one or more convolution layers with pooling layers between them, followed by one or more fully connected layers, as in a standard multilayer perceptron. CNNs are easier to train compared to Deep Neural Networks because they have fewer parameters and local receptive fields.
In CNNs, kernels/filters are used to see where particular features are present in an image by convolution with the image. The size of the filters gives rise to locally connected structure which are each convolved with the image to produce feature maps. The feature maps are usually sub-sampled using mean or max pooling. The reduction in parameters is due to the fact that convolution layers share weights.
The reason behind parameter sharing is that we make an assumption, that the statistics of a patch of a natural image are the same as any other patch of the image. This suggests that features learned at one location can also be learned for other locations. So, we can apply this learned feature detector anywhere in the image. This makes CNNs ideal feature extractors for images.
The CNNs with many layers have been used for various applications especially image classification. In this paper, we use two state-of-the-art Deep CNNs that have achieved one of the lowest error rates in image classification tasks.
In this work, we use VGG16 and Resnet50, pre-trained on the ImageNet dataset, along with a few modifications. We also compare our modified and hybrid models with the original ones. The VGG16 architecture was proposed by the Visual Geometry Group at the University of Oxford [32], which was deep, simple, sequential network whereas the Resnet50, proposed by Microsoft research [10], was an extremely deep graphical network with residual connections (which avoids the vanishing gradients problem and residual functions are easier to train).
We also test slightly modified versions of both these networks by adding a fully-connected layer and fine-tuning on our dataset. We also tested with more fully connected layers but the increase in accuracy was overshadowed by the increase in training time.

### 3.2 The Hybrid Model

We propose to use a hybrid architecture for fire detection in images. In this paper, instead of using a simple CNN as feature extractor, we employ state-of-the-art Deep CNNs like the VGG16 and Resnet50.
Figure 3(a) and 3(b) show the architecture of the VGG16-ELM and Resnet50-ELM hybrid models respectively. Usually, only the softmax classifier is replaced by another classifier (ELM or SVM) in a CNN to create a hybrid model. But, we go one step further by replacing the entire fully connected multi-layer perceptron with a single hidden layer ELM. This decreases the complexity of the model even further.



**The Theory of Extreme Learning Machine:** The Extreme Learning Machine is a supervised learning algorithm [13]. The input to the ELM, in this case, are the features extracted by the CNNs. Let it be represented as $x_i, t_i$, where $x_i$ is the input feature instance and $t_i$ is the corresponding class of the image. The inputs are connected to the hidden layer by randomly assigned weights $w$. The product of the inputs and their corresponding weights act as inputs to the hidden layer activation function. The hidden layer activation function is a non-linear non-constant bounded continuous infinitely differentiable function that maps the input data to the feature space. There is a catalogue of activation functions from which we can choose according to the problem at hand. We ran experiments for all activation functions and the best performance was achieved with the multiquadratics function:

$$f(x) = \sqrt{\|x_i - \mu_i\|^2 + a^2} \tag{1}$$

The hidden layer and the output layer are connected via weights $\beta$, which are to be analytically determined. The mapping from the feature space to the output space is linear. Now, with the inputs, hidden neurons, their activation functions, the weights connecting the inputs to the hidden layer and the output weights produce the final output function:

$$\sum_{i=1}^{L} \beta_i g(w_i.x_j + b_i) = o_j \tag{2}$$

The output in Matrix form is:

$$H\beta = T \tag{3}$$

The error function used in Extreme Learning Machine is the Mean Squared error function, written as:

$$E = \sum_{j=1}^{N} (\sum_{i=1}^{L} \beta_i g(w_i.x_j + b_i) - t_j)^2 \tag{4}$$

To minimize the error, we need to get the least-squares solution of the above linear system.

$$\|H\beta^* - T\| = min_\beta \|H\beta - T\| \tag{5}$$

The minimum norm least-squares solution to the above linear system is given by:

$$\hat{\beta} = H^{\dagger}T \tag{6}$$

Properties of the above solution:

1. *Minimum Training Error:* The following equation provides the least-squares solution, which means the solution for $\|H\beta - T\|$, i.e. the error is minimum. $\|H\beta^* - T\| = min_{\beta}\|H\beta - T\|$
2. *Smallest Norm of Weights:* The minimum norm of least-squares solution is given by the Moore-Penrose pseudo inverse of $H$.
   $\hat{\beta} = H^{\dagger}T$
3. *Unique Solution:* The minimum norm least-squares solution of $H\beta = T$ is unique, which is:
   $\hat{\beta} = H^{\dagger}T$

Detailed mathematical proofs of these properties and the ELM algorithm can be found in [14]. Both the VGG16 and Resnet50 extract rich features from the images. These features are fed to the ELM classifier which finds the minimum norm least squares solution. With enough number of hidden neurons, the ELM outperforms the original VGG16 and Resnet50 networks. Both VGG16 and Resnet50 are pre-trained with ImageNet weights. So, only the ELM classifier is trained on the features extracted by the CNNs.
Apart from fast training and accurate classification, there is another advantage of this model. This hybrid model does not require large training data. In fact, our dataset consists of just 651 images, out of which the ELM is trained on 60% of images only. This shows its robustness towards lack of training data. A normal Deep CNN would require much higher amount of training data to fine-tune its fully-connected layers and the softmax classifier. Even the pre-trained VGG16 and Resnet50 models required at least 80% training data to fine-tune their fully-connected layers.
And, as we will show in the next section, a hybrid CNN-ELM trained with 60% training data outperforms pre-trained VGG16 and Resnet50, fine-tuned on 80% training data.

### 3.3 Paper Contributions

1. The previous hybrid models have used simple CNNs for feature extraction. We employ state-of-the-art Deep CNNs to make feature extraction more efficient and obtain relevant features since the dataset is difficult to classify.
2. Other hybrid models simply replace the softmax classifier with SVM or sometimes ELM. We completely remove the fully connected layers to increase speed of convergence since no fine-tuning is needed and also reduce the complexity of the architecture. Since VGG16 and Resnet50 extract rich features and the ELM is an accurate classifier, we do not need the fully-connected layers. This decreases the number of layers by 2 in VGG16 and by 1 in Resnet50, which is 8192 and 4096 neurons respectively.

3. The above point also justifies the use of complex features extractors like VGG16 and Resnet50. If we used a simple CNN then, we might not be able to remove the fully-connected layers since the features might not be rich enough. Due to this, the fully-connected layers would have to be fine-tuned on the dataset which would increase training time and network complexity.
4. Also, we see that the data required for training the ELM classifier is lower than the data required for fine-tuning the fully-connected layers of a pre-trained Deep CNN.
5. We apply our hybrid model on the problem of fire detection in images (on our own dataset). And, to the best of our knowledge, this is the first time a hybrid ELM model has been applied to this problem.



## 4 Experiments

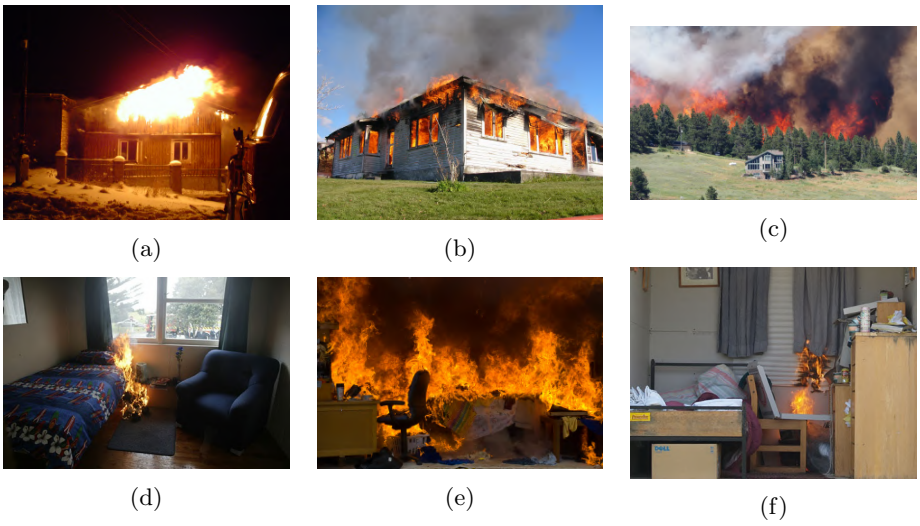
We conducted our experiments to compare training and testing accuracies and execution times of: the VGG16 and Resnet50 models including modifications, Hybrid VGG16 and Resnet50 models with ELM classifier. We also compare our hybrid VGG16-ELM and Resnet50-ELM models with VGG16-SVM and Resnet50-SVM as well. We used pre-trained Keras [5] models and fine-tune the fully-connected layers on our dataset. The training of the models was done on the following hardware specifications: Intel i5 2.5GHz, 8GB RAM and Nvidia Geforce GTX 820 2GB GPU. Each model was trained on the dataset for 10 training epochs. The ADAM optimizer [16] with default parameters $\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$ was used to fine-tune the fully-connected layers for VGG16 and Resnet50 and their modified versions. The details of the dataset are given in the next subsection.

### 4.1 The Real World Fire Dataset

Since there is no benchmark dataset for fire detection in images, we created our own dataset by handpicking images from the internet. [2]This dataset consists of 651 images which is quite small in size but it enables us to test the generalization capabilities and the effectiveness and efficiency of models to extract relevant features from images when training data is scarce. The dataset is divided into training and testing sets. The training set consists of 549 images: 59 fire images and 490 non-fire images. The imbalance is deliberate to replicate real world situations, as the probability of occurrence of fire hazards is quite small. The datasets used in previous papers have been balanced which does not imitate the real world environment. The testing set contains 102 images: 51 images each of fire and non-fire classes. As the training set is highly unbalanced and the testing set is exactly balanced, it makes a good test to see whether the models are able to generalize well or not. For a model with good accuracy, it must be able to

[2] The dataset is available here: `https://github.com/UIA-CAIR/Fire-Detection-Image-Dataset`

extract the distinguishing features from the small amount of fire images. To extract such features from small amount of data the model must be deep enough. A poor model would just label all images as non-fire, which is exemplified in the results.
Apart from being unbalanced, there are a few images that are very hard to classify. The dataset contains images from all scenarios like fire in a house, room, office, forest fire, with different illumination intensity and different shades of red, yellow and orange, small and big fires, fire at night, fire in the morning. Non-fire images contain a few images that are hard to distinguish from fire images like a bright red room with high illumination, sunset, red coloured houses and vehicles, bright lights with different shades of yellow and red etc.
The figures 4(a) to 4(f) show fire images in different environments: indoor, outdoor, daytime, nighttime, forest fire, big and small fire. And the figures 5(a) to 5(f) show the non-fire images that are difficult to classify. Considering these characteristics of our dataset, detecting fire can be a difficult task. We have made the dataset available online so that it can be used for future research in this area.



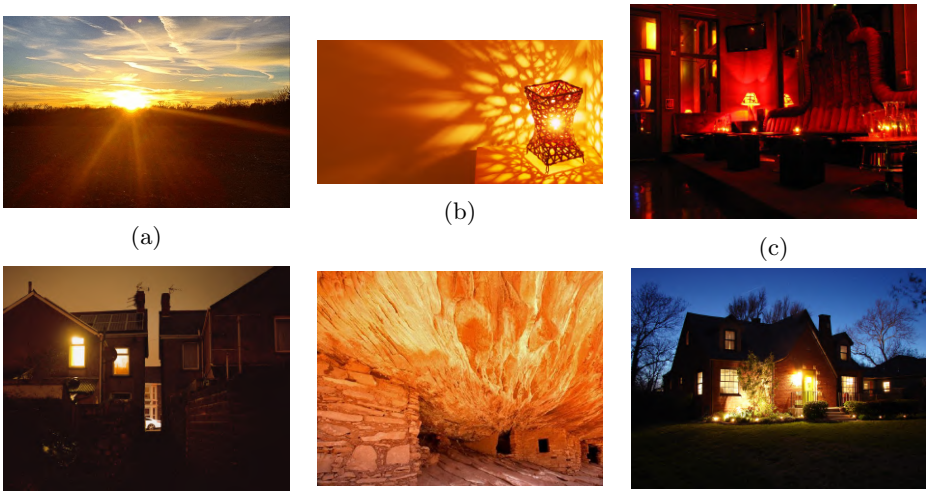
(a) (b) (c)

(d) (e) (f)

Fig. 1: Examples of Fire Images

### 4.2 Results

Our ELM hybrid models are tested on our dataset and compared with SVM hybrid models and the original VGG16 and Resnet50 Deep CNN models. Table 1 and Table 2 show the results of the experiments. The dataset was randomly split into training and testing sets. Two cases were considered depending on the amount of training data. The Deep CNN models (VGG16 and Resnet50) were



(a) (b) (c) (d) (e) (f)

Fig. 2: Examples of Non-Fire Images that are difficult to classify

Table 1: Accuracy and Execution time

| **Model** | $D_T$ | $Acc_{train}$ | $T_{train}$ | $T^C_{train}$ | $Acc_{test}$ | $T_{test}$ |
|---|---|---|---|---|---|---|
| VGG16 (pre-trained) | 80 | 100 | 7149 | 6089 | 90.19 | 121 |
| VGG16 (modified) | 80 | 100 | 7320 | 6260 | 91.176 | 122 |
| Resnet50 (pre-trained) | 80 | 100 | 15995 | 13916 | 91.176 | 105 |
| Resnet50 (modified) | 80 | 100 | 16098 | 13919 | 92.15 | 107 |
| VGG16+SVM | 60 | 99.6 | 2411 | 1352 | 87.4 | 89 |
| VGG16+SVM | 80 | 100 | 2843 | 1784 | 93.9 | 81 |
| VGG16+ELM | 60 | 100 | 1340 | 281 | 93.9 | 24 |
| VGG16+ELM | 80 | 100 | 1356 | 297 | 96.15 | 21 |
| Resnet50+SVM | 60 | 100 | 3524 | 1345 | 88.7 | 97 |
| Resnet50+SVM | 80 | 100 | 4039 | 1860 | 94.6 | 86 |
| Resnet50+ELM | 60 | 100 | 2430 | 251 | 98.9 | 32 |
| Resnet50+ELM | 80 | 100 | 2452 | 272 | 99.2 | 26 |

$D_T$ is the percentage of total data used for training the models.
$Acc_{train}$ and $Acc_{test}$ are the training and testing accuracies respectively.
$T_{train}$ and $T_{test}$ are the training and testing times for the models.
$T^C_{train}$ is the time required to train the classifier part of the models.

trained only on 80% training data, since 60% is too less for these models. All the hybrid models have been trained on both 60% and 80% of training data. One point to be noted here is that, the SVM hybrid models contain an additional fully-connected layer of 4096 neurons, while the ELM is directly connected to the last pooling layer.

Table 2: Number of Hidden Neurons in ELM

| **CNN Features** | **# hidden neurons** | **Testing accuracy** |
|---|---|---|
| VGG16 Feature Extractor | 4096 | 93.9 |
| VGG16 Feature Extractor | 8192 | 94.2 |
| VGG16 Feature Extractor | 16384 | 91.1 (Overfitting) |
| Resnet50 Feature Extractor | 4096 | 98.9 |
| Resnet50 Feature Extractor | 8192 | 99.2 |
| Resnet50 Feature Extractor | 16384 | 96.9 (Overfitting) |



The results in Table 1 show that the ELM hybrid models outperform the VGG16, Resnet50 and SVM hybrid models by achieving higher accuracy and learning much faster. In general, we can see that the hybrid models outperform the state-of-the-art Deep CNNs in terms of both accuracy and training time.
Apart from accuracy and training time, another important point drawn from the results is the amount of training data required. As we already know, Deep Neural Networks (DNN) require huge amount of training data. So, using pre-trained models can be highly beneficial, as we only need to fine-tune the fully-connected layers. But, with models like VGG16 and Resnet50 which have large fully-connected layers, even fine-tuning requires large amount of training data. We had to train the VGG16 and Resnet50 on at least 80% training data otherwise they were overfitting on the majority class, resulting in 50% accuracy.
But in case of hybrid models, especially ELM hybrid models, the amount of training data required is much less. Even after being trained on 60% training data, the ELM models were able to outperform the original VGG16 and Resnet50 models which were trained on 80% training data. This shows that reducing the fully-connected layers, or replacing them with a better classifier can reduce the amount of training data required. Also, the ELM is more robust towards lack of training data which adds to this advantage.
Among the hybrid models, the ELM hybrid models outperform the SVM hybrid models both in terms of testing accuracy and training time. Also, we can see that the hybrid models with Resnet50 as the feature extractor achieves better results than the hybrid models with VGG16 as the feature extractor. This is due to the depth and the residual connections in Resnet50 in contrast to the simple, shallower (compared to Resnet50) and sequential nature of VGG16.
Table 2 compares results between different number of hidden neurons used by ELM. The accuracy increases as the number of hidden neurons increase. The models are tested for $2^{12}$, $2^{13}$ and $2^{14}$ number of neurons. The testing accuracy starts to decrease for $2^{14}$ neurons, which means the model overfits. All the tests in Table 2 were conducted with 60% training data.

## 5 Conclusion

In this paper, we have proposed a hybrid model for fire detection. The hybrid model combines the feature extraction capabilities of Deep CNNs and the classification ability of ELM. The Deep CNNs used for creating the hybrid models are the VGG16 and Resnet50 instead of a simple Deep CNN. The fully connected layers are removed completely and replaced by a single hidden layer feedforward neural network trained using the ELM algorithm. This decreases complexity of the network and increases speed of convergence. We test our model on our own dataset which has been created to replicate a realistic view of the environment which includes different scenarios, imbalance due to lower likelihood of occurrence of fire. The dataset is small in size to check the robustness of models towards lack of training data, since deep networks require a considerable amount of training data. Our hybrid model is compared with the original VGG16 and Resnet50 models and also with SVM hybrid models. Our Deep CNN-ELM model is able to outperform all other models in terms of accuracy by 2.8% to 7.1% and training time by a speed up of 20x to 51x and requires less training data to achieve higher accuracy for the problem of fire detection.



## References

1. Hossein Azizpour, Ali Sharif Razavian, Josephine Sullivan, Atsuto Maki, and Stefan Carlsson. From generic to specific deep representations for visual recognition. *CoRR*, abs/1406.5774, 2014.
2. G. Bradski. Opencv. *Dr. Dobb's Journal of Software Tools*, 2000.
3. Le-le Cao, Wen-bing Huang, and Fu-chun Sun. *A Deep and Stable Extreme Learning Approach for Classification and Regression*, pages 141–150. Springer International Publishing, Cham, 2015.
4. Daniel Yoshinobu Takada Chino, Letricia P. S. Avalhais, José Fernando Rodrigues Jr., and Agma J. M. Traina. Bowfire: Detection of fire in still images by integrating pixel color and texture analysis. *CoRR*, abs/1506.03495, 2015.
5. Francois Chollet. Keras, 2015.
6. J. Zhao Z. Zhang C. Qu Y. Ke D. Zhang, S. Han and X. Chen. Image based forest fire detection using dynamic characteristics with artificial neural networks. In *2009 International Joint Conference on Artificial Intelligence*, pages 290–293, April 2009.
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, Oct 2016.
8. Kunihiko Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36(4):193–202, 1980.
9. F. Grpinar, H. Kaya, H. Dibeklioglu, and A. A. Salah. Kernel elm and cnn based facial age estimation. In *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 785–791, June 2016.

10. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.
11. Wen-Bing Horng and Jian-Wen Peng. Image-based fire detection using neural networks. In *JCIS*, 2006.
12. Gao Huang, Guang-Bin Huang, Shiji Song, and Keyou You. Trends in extreme learning machines: a review. *Neural Networks*, 61:32–48, 2015.
13. Guang-Bin Huang, Qin-Yu Zhu, and Chee-Kheong Siew. Extreme learning machine: a new learning scheme of feedforward neural networks. In *Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on*, volume 2, pages 985–990. IEEE, 2004.
14. Guang-Bin Huang, Qin-Yu Zhu, and Chee-Kheong Siew. Extreme learning machine: theory and applications. *Neurocomputing*, 70(1):489–501, 2006.
15. Jihun Kim, Jonghong Kim, Gil-Jin Jang, and Minho Lee. Fast learning method for convolutional neural networks using extreme learning machine and its application to lane detection. *Neural Networks*, 87:109 – 121, 2017.
16. Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.
17. Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.
18. Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, Nov 1998.
19. Mark D. McDonnell, Migel D. Tissera, André van Schaik, and Jonathan Tapson. Fast, simple and accurate handwritten digit classification using extreme learning machines with shaped input-weights. *CoRR*, abs/1412.8307, 2014.
20. Mark D. McDonnell and Tony Vladusich. Enhanced image classification with a fast-learning shallow convolutional neural network. *CoRR*, abs/1503.04596, 2015.
21. Jawad Nagi, Gianni A. Di Caro, Alessandro Giusti, Farrukh Nagi, and Luca Maria Gambardella. Convolutional neural support vector machines: Hybrid visual pattern classifiers for multi-robot systems. In *ICMLA (1)*, pages 27–32. IEEE, 2012.
22. Shan Pang and Xinyi Yang. Deep convolutional extreme learning machine and its application in handwritten digit classification. *Intell. Neuroscience*, 2016:6–, August 2016.
23. Bc. Tomas Polednik. Detection of fire in images and video using cnn. *Excel@FIT*, 2015.
24. K. Poobalan and S.C. Liew. Fire detection algorithm using image processing techniques. In *3rd International Conference on Artificial Intelligence and Computer Science (AICS2015)*, Ocotober 2015.
25. Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. CNN features off-the-shelf: an astounding baseline for recognition. *CoRR*, abs/1403.6382, 2014.
26. Bernardete Ribeiro and Noel Lopes. *Extreme Learning Classifier with Deep Concepts*, pages 182–189. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.
27. Richard Bright Richard Custer. Fire detection: The state of the art. *NBS Technical Note, US Department of Commerce*, 1974.
28. Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

29. J. s. Yu, J. Chen, Z. Q. Xiang, and Y. X. Zou. A hybrid convolutional neural networks with extreme learning machine for wce image classification. In *2015 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 1822–1827, Dec 2015.
30. Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *CoRR*, abs/1312.6229, 2013.
31. Jing Shao, Guanxiang Wang, and Wei Guo. An image-based fire detection method using color analysis. In *2012 International Conference on Computer Science and Information Processing (CSIP)*, pages 1008–1011, Aug 2012.
32. Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.
33. Yichuan Tang. Deep learning using support vector machines. *CoRR*, abs/1306.0239, 2013.
34. C. Tao, J. Zhang, and P. Wang. Smoke detection based on deep convolutional neural networks. In *2016 International Conference on Industrial Informatics - Computing Technology, Intelligent Technology, Industrial Information Integration (ICIICII)*, pages 150–153, Dec 2016.
35. Jonathan Tapson, Philip de Chazal, and André van Schaik. Explicit computation of input weights in extreme learning machines. *CoRR*, abs/1406.2889, 2014.
36. Tom Toulouse, Lucile Rossi, Turgay Celik, and Moulay Akhloufi. Automatic fire pixel detection using image processing: a comparative analysis of rule-based and machine learning-based methods. *Signal, Image and Video Processing*, 10(4):647–654, 2016.
37. B. Ugur Treyin, Yigithan Dedeoglu, Ugur Gdkbay, and A. Enis etin. Computer vision based method for real-time fire and flame detection. *Pattern Recognition Letters*, 27(1):49 – 58, 2006.
38. J. v. d. Wolfshaar, M. F. Karaaba, and M. A. Wiering. Deep convolutional neural networks and support vector machines for gender recognition. In *2015 IEEE Symposium Series on Computational Intelligence*, pages 188–195, Dec 2015.
39. Steven Verstockt, Peter Lambert, Rik Van de Walle, Bart Merci, and Bart Sette. State of the art in vision-based fire and smoke dectection. In Heinz Luck and Ingolf Willms, editors, *International Conference on Automatic Fire Detection, 14th, Proceedings*, volume 2, pages 285–292. University of Duisburg-Essen. Department of Communication Systems, 2009.
40. Jerome Vicente and Philippe Guillemant. An image processing technique for automatically detecting forest fire. *International Journal of Thermal Sciences*, 41(12):1113 – 1120, 2002.
41. Yujun Zeng, Xin Xu, Yuqiang Fang, and Kun Zhao. *Traffic Sign Recognition Using Deep Convolutional Networks and Extreme Learning Machine*, pages 272–280. Springer International Publishing, Cham, 2015.
42. Lei Zhang and David Zhang. SVM and ELM: who wins? object recognition with deep convolutional features from imagenet. *CoRR*, abs/1506.02509, 2015.
43. Qingjie Zhang, Jiaolong Xu, Liang Xu, and Haifeng Guo. Deep convolutional neural networks for forest fire detection. February 2016.
44. W. Zhu, J. Miao, and L. Qing. Constrained extreme learning machine: A novel highly discriminative random feedforward neural network. In *2014 International Joint Conference on Neural Networks (IJCNN)*, pages 800–807, July 2014.