Fire Detection using Principal Components Analysis and Inception-V3 Transfer Learning CNNs

**Abstract**

Image fire detection has emerged as a vital technology for early fire detection, significantly reducing fire-related losses by promptly alerting users. This method relies on algorithmic analysis of images. However, common detection algorithms, involving both manual and automated extraction of image features, suffer from drawbacks such as lower accuracy, delayed detection, and extensive computational requirements. To address these issues, this paper introduces innovative algorithms for fire detection in images using advanced object detection Convolutional Neural Network architecture, including ResNet50, SSD, YOLO v8, and Inception V3. This study presents a model based on deep learning specifically crafted for recognizing fire and smoke within images. The research entails customizing the widely utilized Inception-V3 Convolutional Neural Network, a common choice for tasks like image classification and object recognition. The modified Inception-V3 is employed on a dataset containing fire images, including those with smoke, for the purpose of detection. An additional optimization function is introduced to improve computational efficiency. The results, inclusive of comparisons with alternative deep learning models and early fire detection approaches, reveal that the Inception-V3-based model excelled, demonstrating superior performance with fewer false positives compared to earlier investigations.

**Index Terminology**:Image Processing, Deep Learning CNNs, Convolutional Neural Network, Inception V3, Fire Detection.

1. **INTRODUCTION**

The swift economic progress has brought about significant challenges in fire control due to the escalating scale and intricacy of constructions. Consequently, there is a pressing need for early fire detection and alarms characterized by heightened accuracy and sensitivity to mitigate fire dectection losses. Conventional fire detection methods, such as smokes and heat detectors, prove inadequate for vast spaces, intricate structures, or environments with numerous disturbances. The shortcomings of these established detection technologies often lead to issues such as missed detections, false alarms, and delays, complicating the achievement of timely fire warnings.

In recent times, there has been a surge in research interest surrounding image fire detection. This method boasts numerous benefits, including early detection of fires, superior accuracy, adaptable system installation, and the capability to effectively detect fires in large spaces and complex building structures [1]. The technology operates by utilizing algorithms to analyze image data captured by a camera, discerning the detection of a fire or the possibility of a fire hazard in the images. As a result, the effectiveness of the effectiveness of the image fire detector is directly dependent on the performance of the detection algorithm making it the central component of this technology.

The image fire detection algorithms comprise three primary stages: in the sequence of image preprocessing, feature extraction, and fire detection, feature extraction takes center stage as the pivotal element in these algorithms. Conventional algorithms depend on manually selected characteristics of fires and classification using machine learning. The drawback of many algorithms lies in the necessity for professional knowledge in manually choosing fire features. Despite researchers conducting numerous studies on image features related to smoke and flames, the findings are limited to simple image features like color, edges, and basic textures. This limitation arises due to the complexity of fire types and scenes, presenting a challenge for comprehensive feature discovery. Yet, in practical applications, the complexity of fire types and scenes, coupled with numerous interference events, poses a challenge for algorithms that extract low and middle complex image features. These challenges make it difficult to differentiate between actual fires and fire-like phenomena, resulting in decreased accuracy and weakened generalization ability. Convolutional Neural Networks (CNNs) serve as a foundation for image recognition algorithms, enabling the automatic learning and efficient extraction of complex image features. These algorithms have garnered significant attention and demonstrated outstanding performance in different areas like visual search, self-driving vehicles, and medical diagnostics. Consequently, scholars have incorporated CNNs into the domain of image fire detection, pioneering self-learning algorithms for the extraction of features from fire images [1-9]. Altered the cutting-edge models including ResNet50, SSD, YOLO v8, and others, and formulated algorithms specifically designed for the detection of smoke and flames [7,9].

Recent progress in training multilayer neural networks has significantly impacted a diverse range of machine learning challenges, encompassing tasks such as classification and regression. Systems employing multiple layers, commonly termed "deep" architecture, possess the capability to derive more abstract and features that remain consistent or unchanged from data. This characteristic is believed to endow them with the potential to achieve higher classification accuracy compared to traditional classifiers [10]. The utilization of deep learning models for classification, specifically incorporating spectral and spatial information, has been explored in [11][12]. Additionally, Convolutional Neural Network (CNN) applications in diverse image processing applications, particularly for pixel-level labeling issues, have been well-documented. The CNN model facilitates the learning of robust feature representations, enabling the seamless execution of end-to-end labeling tasks. This study involves the development of a cost-effective modified CNNs for identifying fires in images. Drawing inspiration from the architecture of AlexNet [13], the Inception V3 model is specifically customized for the task of detecting fires. The investigation aims to overcome certain limitations present in conventional deep learning models designed for fire and smoke detection. Notably, the Inception V3 architecture is concerning loss rate, and incorporates various elements. The adapted model applied to detect fire in images. The resulting model demonstrates efficient real-time detection of fire. The significance of this model lies in its potential to contribute to society by enabling the early detection of fires and smoke, thereby aiding in the prevention of potential disasters.

1. **PROPOSED METHOD**
   1. ***Convolutional Neural Network Architectures***

CNNs operate on the principles of weight sharing, spatial feature extraction, and minimized computational costs [14]. First introduced by LeCun in 1989 for visual imagery analysis [15], recent developments in CNN architectures have demonstrated impressive performance in object detection, especially in the realm of face mask detection. These CNN-based models adopt the structure of artificial neural networks (ANNs) and serve as classifiers, extracting hierarchical features from image data. The network utilizes activation functions and training algorithms to gradually acquire spatial hierarchies of image features [16], with input images serving as automatic training labels.

In CNN, the architecture extends beyond simple input and output layers, featuring additional types of layers such as convolutional layers, pooling layers, and fully connected layers [17]. The essential convolutional layer convolves the input image with trainable filters, extracting features. Each filter consists of neurons identifying features for layer inputs, and convolutional learning occurs by sliding the filter over the input image, resulting in feature maps. The quantity of filters dictates the number of feature maps generated. To decrease dimensionality while retaining crucial information, subsampling methods such as average pooling and max pooling are utilized [18]. The pooling layer is inserted between two convolutional layers to control overfitting by minimizing redundant representations from the preceding layers. Pooling operations, such as max pooling and average pooling, reduce the number of neurons in the previous convolutional layer. Max pooling is suitable for sparse pooled features, while average pooling enables the network to operate on various frequencies at each layer, enhancing invariance and reducing redundancies [19]. The fully connected layer plays a crucial role in mapping the features extracted from pooling and convolutional layers to the final output, such as object detection [14]. Subsampling, involving various types like average, sum, and maximum, continues until the network is fully connected. In adherence to the principles of in the context of object detection algorithms, the process of image fire detection algorithms based on convolutional neural networks is depicted in the workflow in Figure 1. The detection CNN encompasses functions such as region proposals, feature extraction, and classification. Initially, the CNN processes an image as input and produces region proposals through convolution and pooling operations. Subsequently, the object detection CNN based on regions utilizes convolutional layers, pooling layers, and fully connected layers to ascertain whether fire is present or absent in the proposed regions.

The convolutional layer stands as a pivotal element within CNNs. Unlike other neural networks that employ connection weights and weighted sums, the convolutional layer employs image transforms filters, referred to as convolution kernels, are used to produce feature maps from original images. This layer consists of a set of convolution kernels sliding across images, computing new pixels through a sum of pixel values weighted by specific coefficients they traverse to create a feature map. The feature map functions as a representation of distinct aspects present in the original image. Equation (1) delineates the formula for calculating the convolutional layer.

**Table 1** Complete designation for convolutional neural network

Name Full name of convolutional neural network

AlexNet AlexNet  
VGG Very Deep Convolutional Networks for Large-Scale Image Recognition  
Inception Inception  
ResNet Residual Network  
Inception Resnet V2 Inception Residual Version 2  
Darknet-53 Darknet-53  
SSD Single Show MultiBox Detector  
YOLO v8 You Only Look Once Version 8

Ảnh có chứa lửa, ngọn lửa, nhiệt, cháy rừng

Mô tả được tạo tự độngẢnh có chứa cây cối, ngoài trời, nhiệt, thiên nhiên

Mô tả được tạo tự độngẢnh có chứa cây cối, ngoài trời, nhiệt, thiên nhiên

Mô tả được tạo tự động

Ảnh có chứa cây cối, lửa, ngoài trời, nhiệt

Mô tả được tạo tự độngẢnh có chứa cây cối, ngoài trời, nhiệt, thiên nhiên

Mô tả được tạo tự độngẢnh có chứa cây cối, ngoài trời, nhiệt, thiên nhiên

Mô tả được tạo tự động

**No Fire**

**Fire**

**Fully-connected layer**

**Pooling Layer**

**Convolutional Layer**

**Pooling Layer**

**Convolutional Layer**

**Fig. 1**: Fire detection architecture based on detection CNNs.

In this formula, x denotes an input image with dimensions W x H, w represents a convolution kernel with dimensions J x I, b signifies bias, and y indicates the output feature maps. The specific values of w and b are determined during the training process.

* 1. **Principal Components Analysis (PCA) for Data Augmentation**

Data augmentation is a pivotal element in the classification of pulmonary images, exerting a substantial influence on the final classification results. The efficacy of data augmentation has a direct bearing on the overall performance of fire image classification. To notably decrease computational costs, Principal Component Analysis (PCA) has been employed. While ensuring calculation accuracy, PCA primarily focuses on selecting essential components to reduce the dimensionality of features. In practical terms, image feature vectors often entail high-dimensional data, posing challenges in storage and computation during training. Consequently, the reduction of dimensional data is a critical step in various problems, essentially amounting to efficient data handling. In simpler terms, dimensionality reduction involves identifying a function that takes an initial data point as input.

When contending with a vast and high-dimensional space denoted as D, addressing the challenges inherent in such dimensions becomes imperative. Merely preserving the K most crucial elements to reduce the dimensions from D to K<D may not result in optimal performance. The challenge lies in the uncertainty about which elements genuinely bear the most significance among the features. In the least favorable scenario, every element might be considered equally important, leading to a substantial loss of valuable information. However, a more efficient strategy entails representing the original data vectors in a new basis system where the importance of components varies. This approach allows for the exclusion of the least important elements, thereby enhancing the overall efficiency of dimensionality reduction. PCA facilitates this process by identifying a new basis system, concentrating the data's information in a few coordinates, and simplifying calculations through the determination of an orthonormal system to serve as the new basis.

Suppose the new orthnormal basis system is U and want to keep it K coordinates in this new base system. Without loss of generality, assume that it is K first ingredient, as depicted in figure 2 below.

Ảnh có chứa ảnh chụp màn hình, biểu đồ, Hình chữ nhật, hàng

Mô tả được tạo tự động

**Fig 2:** PCA Architecture conception

In the figure 2, with new basic is an orthonormal system with is sub-matrix formed by first column of , the purpose of PCA is to find orthogonal matrices to majority of information remained and to be removed and replace with an independent matrix in each data point. Approximately by one matrix have all identical rows with . is row vector have all elements equal to 1.

To implement PCA, following steps:

1. Calculate the expected vector of the entire data

2. Subtract each data point from the expected vector of the entire data

3. Calculate the covariance matrix

4. Calculate the eigenvalues and eigenvectors with norm equal to 1 of this matrix, arrange them in descending order of eigenvalues.

5. Choose eigenvectors corresponding to the largest eigenvalues to build a matrix with columns forming an orthogonal system. These vectors, also called principal components, form a subspace close to the distribution of the original normalized data.

6. Project orginal normalization data down to finded space.

7. New data is the coordinators of all data points in new space.

The original data can be calculated by approximating via new data:

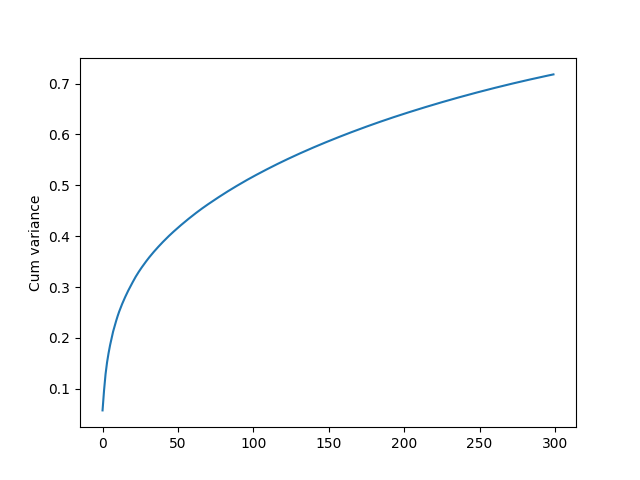
All the following PCA procedure can be done by Fig. 3

Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ, hàng

Mô tả được tạo tự động

**Fig 3:** PCA Architecture Implementation

For this study, the n\_components=300 was used for PCA Architecture to fit all the data which reach a peak cum variance value to process all the images as shown as Fig 4.



**Fig 4**: Choose the best n\_components for PCA.

* 1. **Pre-processing**

During this phase, we have the capability to mitigate unwanted distortions and accentuate specific features essential to the given application. These features can vary based on the application's requirements. Image pre-processing encompasses tasks like standardizing images to a uniform resolution and augmenting images to enhance the model's ability to generalize its learning. Resizing images holds particular importance in computer vision, and for the training of the proposed model, each image has been resized to 224x224 pixels. Data augmentation [20] involves artificially creating variations in images to enhance the model's generalization. One technique, horizontal flip augmentation, involves reversing entire rows and columns of images horizontally. One hot encoding which used to represent categorical variables as binary vector. In this con text, this acts as encode class labels (0,1). A Convolutional Neural Network (CNN) has the capability to process input pictures within its intricate structure of the neural network. It assigns learnable weights and biases to different elements, facilitating the extraction of features from unprocessed images. The network then makes decisions based on the organized feature set [21][22]. In this architecture, a Max Pooling 2D layer is incorporated with a pool size of 3x3. A Conv2D layer employs ReLU as the activation function and has strides set to 4x4, operating on an input size of 224x224x3. The softmax activation function is utilized for the dense layer.

If we denote an input image as X and a filter as f, the expression would be:

The \* indicates the convolutional operator and x is the input vector. The linear transformation of weight vectors and input vectors as

Here, X is the input, W is the weight, and b (bias) is a constant.

* 1. **Inception V3 model for fire detection**

Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ, hàng

Mô tả được tạo tự động

**Fig.5:** High-Level Diagram of Inception V3 Model [23]

As depicted in Figure 5, the structure of the Inception V3 Model is organized into distinct inception blocks. Each inception block may include various combinations of layers, such as Convolution layer, AvgPool layer, MaxPool layer, Concat layer, Dropout layer, Fully Connected layer, and Softmax output. The diagram illustrates that during the learning process, the Inception V3 Model can generate multiple softmax outputs. This model incorporates Factorizing Convolutions, which accelerates the training process by reducing the number of connections. Moreover, it acts as a preventive measure against overfitting by minimizing the parameters that need to be learned [23].

Google designed the Inception models primarily for image classification within CNN frameworks, setting them apart from conventional CNNs due to differences in architecture and the arrangement of inception blocks. In the instance of Inception V3, it processes the same input tensor, utilizes multiple filters, and concatenates their outcomes efficiently, as depicted in Figure 2. The integration of a dropout mechanism introduces a probabilistic approach, randomly excluding certain layers from the entire architecture during training, effectively lowering computational costs. In this study, a dropout ratio of 0.5 is considered.

In the realm of machine learning and statistics, the learning rate plays a crucial role in tuning the speed of the model. For the proposed models, a learning rate of 0.001 is employed after fine-tuning. Simplistically, optimizers, along with the loss function, shape the model by adjusting weights to produce the most accurate form. This investigation utilizes the Adam optimizer and RMSProp as moment. Optimization involves calculating the exponentially weighted average of past gradients (vdW) and the exponentially weighted average of the squares of past gradients (sdW). Bias correction is computed using Eq. (5) and Eq. (6), tending towards zero. Parameter tuning optimizes the model by updating parameters in each iteration, minimizing the loss function temporarily. Ultimately, parameters are updated using information from the calculated averages. The Adam optimizer can be computed using Eq. (3) and Eq. (4).

Where is the exponentially represents the exponentially weighted average of past gradients, while stands for the exponentially weighted average of past squares of gradients. is hyperparameter to be tuned. is hyperparameter to be tuned. is cost gradient with respect to current layer. Eq (5) (6) depict the modified optimization, derived from the calculations in Eq (3) (4). The ultimate adjustment to the weight matrix of the network is executed using Equations (5) and (6) as presented below:

(7)

W is the weight matrix (parameter to be updated) and is the learning rate, is very small value to avoid dividing by zero.

* 1. **RMSProp**

RMSProp is employed as the loss function in this study. The computation of RMSProp involves the exponentially weighted average of squares, aimed at achieving fast convergence, as expressed below:

Eq. (8) shows RMSProp where s is the exponentially weighted average of past squares of gradients. ∂J/∂W refer the gradient in terms of current layer weight vector. W denotes the weight vector. β is hyperparameter to be tuned, and α denotes the learning rate. ϵ is taken as very small to avoid dividing by zero.

Equation (10) presents the Loss function equation, where y represents the scalar value in the model output, computed according to Equation (11). Here, x corresponds to the corresponding target value, and the output size indicates the number of scalar values in the model output.

1. **EXPERIMENTAL RESULT**

This section presents the experimental outcomes of the proposed Inception V3 model for the fire detection task, including a comparative analysis. The dataset details are also outlined in this section. The implemented deep learning model, alongside baseline models, was executed using Python 3.11 with standard python libraries, such as Keras, Tensorflow were employed for this study.

* 1. **Collection of Dataset**

To effectively train and validate the neural network, a substantial number of images is necessary. The dataset utilized in this study was sourced from Kaggle and is categorized into two sets: Training and Testing. The training dataset is employed to teach the neural network to discern images containing fire from those without fire. Each set further comprises two classes: Fire and Non\_fire. The training dataset encompasses 1167 images in the Fire calss and 275 images in the Non\_fire class, resulting in a total 1,442 images. Meanwhile, the testing part consisting of 998 images in total. Additionally, the proposed model was tested on real-time datasets created for this experiment.



***Fig.6:*** Samples images of Fire dataset

* 1. **Result and Discussion**

After training, both the proposed model and existing deep learning-based models underwent testing on real datasets. The model obtains high accuracy and low validation loss score. To assess the performance of the transfer learning model, various performance metrics were employed, including Accuracy, Precision, Sensitivity, Specificity, Intersection over Union (IoU). The Classification error is formulated in terms of where represents the one-hot encoded vector, and represents the predicted probability.

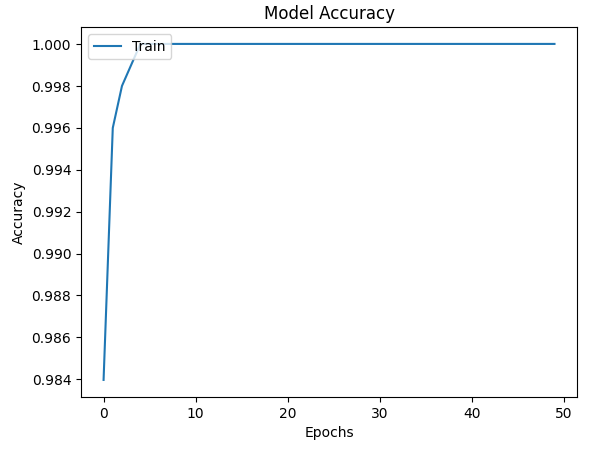
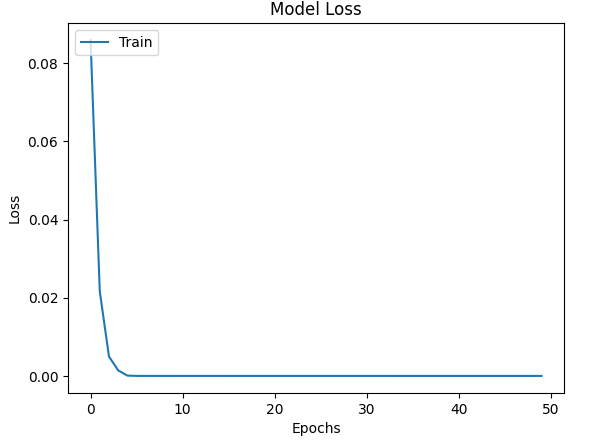
Accuracy = (1)

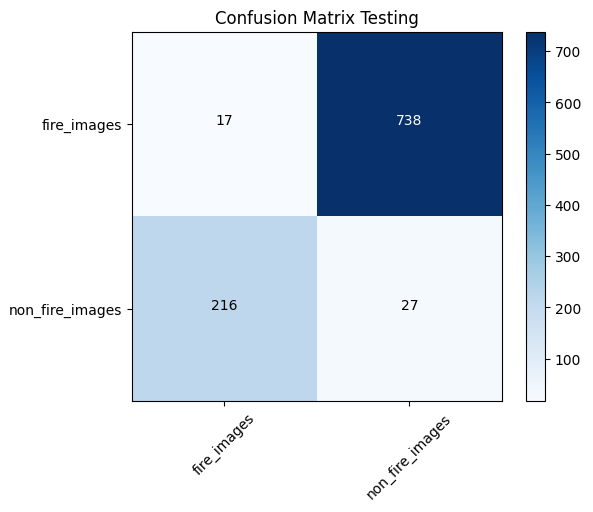
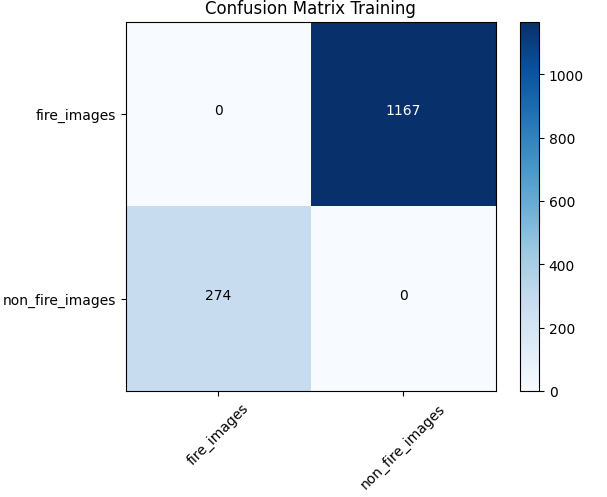
(2)

(3)

Specificity = (4)

IoU = (5)

***Fig.7,8:*** Comparision of performance of model during training and Confusion Matrix

Additional performance metrics are expressed in terms of True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN). These metrics are organized into a grid-like structure known as the confusion matrix. In this study, two confusion matrices are created to assess the model's performance during both training and testing phases. The two confusion matrices are visually presented in Fig. 8. Furthermore, Fig. 7 illustrates a comparison of the area under precision, loss, and accuracy curves during the training and testing of the model. The transfer learning model Inception V3 neural network demonstrates high accuracy in detecting fires in both indoor and outdoor environments, the evaluation of the proposed system incorporates standard metrics such as precision, recall, and F1 score. Additionally, the comparision includes an assessment of dataset size, considering whether previous methods focus on fire, smoke, or both. The system is also benchmarked against two state-of-art fire detection systems, specifically the fire detection system based on deep learning.

1. **Conclusion**

From this study, the leading proposed deep learning model is constructed through the transfer learning of Inception V3. Image augmentation techniques are applied to enhance the model’s performance by increasing the diversity of the training data. The model achieved 100% accuracy in both training and validation during testing on the train dataset. Moreover, by leveraging large volumes of data, the model can be extended to classify different types of fires in both indoor and outdoor environments. The ensemble approach not only contributes to high accuracy but also significantly improves detection speed. Additionally, transfer learning on pretrained models, combined with rigorous testing on an unbiased dataset, results in a reliable and cost-effective solution. The data augmentation be achieved by using Pricipal Components Analysis, to remain the most essential components to booster training process and reduce the computation cost which is huge to handle in multiple dimensional data. In the future, combined with several methods to surveillance and analysis the devastating caused by fire in the early stages.

**References**

**[1]** K. Muhammad, J. Ahmad, I. Mehmood, et al., Convolutional neural networks based fire detection in surveillance videos, IEEE Access 6 (2018) 18174–18183.  
**[2]** C. Tao, J. Zhang, 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), 2016, pp. 150–153.   
**[3]** A. Filonenko, L. Kurnianggoro, K. Jo, Comparative study of modern convolutional neural networks for smoke detection on image data, in: 2017 10th International Conference on Human System Interactions (HSI), 2017, pp. 64–68.   
**[4]** Z. Yin, B. Wan, F. Yuan, et al., A deep normalization and convolutional neural network for image smoke detection, IEEE ACCESS 5 (2017) 18429–18438.  
**[5]** A.J. Dunnings, T.P. Breckon, Experimentally defined convolutional neural network architecture variants for non-temporal real-time fire detection, in: 2018 25th IEEE International Conference on Image Processing (ICIP), 2018, pp. 1558–1562.   
**[6]** A. Namozov, Y. Cho, An efficient deep learning algorithm for fire and smoke detection with limited data, Adv. Electr. Comput. Eng. 18 (2018) 121–128.   
**[7]** W. Mao, W. Wang, Z. Dou, Y. Li, Fire recognition based on multi-channel convolutional neural network, Fire Technol. 54 (2018) 531–554.   
**[8]** K. Muhammad, J. Ahmad, S.W. Baik, Early fire detection using convolutional neural networks during surveillance for effective disaster management, Neurocomputing 288 (2018) 30–42.   
**[9]** Y. Hu, X. Lu, Real-time video fire smoke detection by utilizing spatial-temporal ConvNet features, Multimed. Tool. Appl. 77 (2018) 29283–29301.  
**[10]** N. Kruger, P. Janssen, S. Kalkan, M. Lappe, A. Leonardis, J. Piater, A. J. Rodriguez-Sanchez, and L. Wiskott, “Deep hierarchies in the primate visual cortex: What can we learn for computer vision?,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35, no. 8, pp. 1847–1871, Aug 2013.   
**[11]** Y. Chen, X. Zhao, and X. Jia, “Spectral-spatial classification of hyperspectral data based on deep belief network,” IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 8, no. 6, pp. 2381–2392, 2015.   
**[12]** J. Li, J. M. Bioucas-Dias, and A. Plaza, “Spectral-spatial classification of hyperspectral data using loopy belief propagation and active learning,” IEEE Transactions on Geoscience and Remote Sensing, vol. 51, no. 2, pp. 844– 856, 2013.  
**[13]** Krizhevsky A, Sutskever I, Hinton GE (2012). “ImageNet Classification with Deep Convolutional Neural Networks”. Advances in Neural Information Processing Systems. vol. 25.  
**[14]** M. Jiang, X. Fan, Retinamask: a face mask detector. ArXiv 2020.  
**[15]** S.V. Militante, N.V. Dionisio, Deep learning implementation of facemask and physical distancing detection with alarm systems, Proceeding - 2020 3rd Int. Conf. Vocat. Educ. Electr. Eng. Strength. Framew. Soc. 5.0 through Innov. Educ. Electr. Eng. Informatics Eng. ICVEE 2020, 2020  
**[16]** R. Yamashita, M. Nishio, RKG. Do, K. Togashi, Convolutional neural networks: an overview and application in radiology, Insights Imaging (2018)  
**[17]** AR. Pathak, M. Pandey, S. Rautaray, Application of deep learning for object detection, in: Procedia Comput. Sci., vol. 132, Elsevier B.V., 2018, pp. 1706–1717  
**[18]** M. Mahdianpari, B. Salehi, M. Rezaee, F. Mohammadimanesh, Y. Zhang, Very deep convolutional neural networks for complex land cover mapping using multispectral remote sensing imagery, Remote Sens. (2018)  
**[19]** J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, et al., Recent advances in convolutional neural networks, Pattern Recognit. (2018)

**[20]** Sun D, Roth S, Black MJ (2010). “Secrets of optical flow estimation and their principles”. in: Proceedings of the 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, pp.2432–2439  
**[21]** Li P, Zhao W (2020). “Image fire detection algorithms based on convolutional neural networks”. 2020 Elsevir Fire Safety Journal - Case Studies in Thermal Engineering 19, 100625  
**[22]** Krizhevsky A, Sutskever I, Hinton GE (2012). “ImageNet Classification with Deep Convolutional Neural Networks”. Advances in Neural Information Processing Systems. vol. 25.  
**[23]** Patel, Khush. (2020) “Architecture comparison of AlexNet, VGGNet, ResNet, Inception, DenseNet”