Fire Detection and Verification using Convolutional Neural Networks, Masked Autoencoder and Transfer Learning

Z. Abed Almoussawi[1*], R. Khalid[2], Z. Salam Obaid[3], Z. I. Al Mashhadani[4], K. Al-Majdi[5], R. E. Alsaddon[6], H. Mohammed Abed[7]

1- College of Islamic Science, Ahl Al Bayt University, Kerbala, Iraq.
Email: zainabalmoussawiii@gmail.com (Corresponding author)
2- Medical technical college; Al-farahidi University, Baghdad, Iraq.
Email: read.k.ibrahim@uoalfarhidi.iq
3- Building and Construction Engineering Technology Department, AL-Mustaqbal University College, Hillah 51001, Iraq.
Email: zahrra.salam@mustaqbal-college.edu.iq
4- Al-Nisour University College, Baghdad, Iraq.
Email: zuha.h.bs@nuc.edu.iq
5- Department of biomedical engineering, Ashur University College, Baghdad, Iraq.
Email: dr.kadhum@au.edu.iq
6- Department of Prosthetic Dental Technology, Hilla University college, Babylon, Iraq.
Email: e.refad@yahoo.com
7- Mazaya University College, Iraq.
Email: eng.hassan@mpu.edu.iq

Received: 11 October 2022 Revised: 25 October 2022 Accepted: 24 November 2022

ABSTRACT:
Wildfire detection is a time-critical application since it can be challenging to identify the source of ignition in a short amount of time, which frequently causes the intensity of fire incidents to increase. The development of precise early-warning applications has sparked significant interest in expert systems research due to this issue, and recent advances in deep learning for challenging visual interpretation tasks have created new study avenues. In recent years, the power of deep learning-based models sparked the researcher's interests from a variety of fields. Specially, Convolutional Neural Networks (CNN) have become the most suited approach for computer vision tasks. As a result, in this paper we propose a CNN-based pipeline for classifying and verifying fire-related images. Our approach consists of two models, first of which classifies the input data and then the second model verifies the decision made by the first one by learning more robust representations obtained from a large masked auto encoder-based model. The verification step boosts the performance of the classifier with respect to false positives and false negatives. Based on extensive experiments, our approach proves to improve previous state-of-the-art algorithms by 3 to 4% in terms of accuracy.

KEYWORDS: Fire Detection, Convolutional Neural Networks, Masked Auto Encoder, Vision Transformers, Transfer Learning.

1. INTRODUCTION

The size and complexity of structures have grown in tandem with the economy's fast growth, creating significant fire control issues [1]. To minimize fire losses, early fire detection and alarm with high sensitivity and accuracy are therefore crucial. Traditional fire detection methods, such as smoke and heat detectors, are ineffective in big areas, intricate structures, or areas with a lot of disturbances [2]. Missed detections, false alarms, detection delays, and other issues frequently arise as a result of the limitations of the aforementioned detection technologies, making it even more challenging to accomplish early fire warnings [3].

Image fire detection has recently gained popularity as a study subject. The method has various benefits, including the capacity to efficiently detect fires in wide areas and complicated building structures, high accuracy, flexible system installation, and early fire detection [4]. It uses algorithms to analyze visual data from a camera to find the existence of a fire or a fire danger. As a result, the foundation of this technology is

Paper type: Research paper
DOI: 10.30486/mjee.2023.1977460.1055
How to cite this paper: Z. Abed Almoussawi, R. Khalid, Z. Salam Obaid, Z. I. Al Mashhadani, K. Al-Majdi, R. E. Alsaddon and H. Mohammed Abed, "Fire Detection and Verification using Convolutional Neural Networks, Masked Autoencoder and Transfer Learning", Majlesi Journal of Electrical Engineering, Vol. 16, No. 4, pp. 159-166, 2022.

the detection method, which directly affects how well the image fire detector performs [5].

The three primary steps of an image fire detection method are fire detection, feature extraction, and picture preprocessing. The main component of algorithms is feature extraction, among others [6]. The human selection of fire feature and machine learning classification are the foundations of the traditional approach. The drawback of algorithms is that manual feature selection needs to be done by experts [7]. Even though the researchers conduct several investigations on the picture characteristics of smoke and flame, they only manage to identify basic image characteristics like color, edges, and simple textures. The algorithms that extract low and middle complex image information, however, find it challenging to distinguish between fire and fire-like due to complex fire types and scenes as well as numerous interference events in practical application, resulting in lower accuracy and weak generalization performance [8].

Convolutional Neural Network (CNN)-based image recognition algorithms have the ability to automatically learn and successfully extract complicated visual information [9]. The performance of these algorithms in areas like visual search, autonomous driving, medical diagnosis, etc. has raised significant concerns. As a result, some researchers use CNNs to identify fire in images, creating a self-learning system for gathering fire picture attributes [10]. Built smoke and flame detection algorithms modify cutting-edge models like AlexNet, VGG, Inception, and ResNet.

Although fire detection systems based on CNNs have improved over traditional algorithms in terms of detection accuracy in complicated scenarios, several issues still persist [11]. First, the region proposal stage was overlooked in current machine learning algorithms that primarily treated picture fire detection as a classification problem. The algorithms assign a single class to the entire picture. However, just a tiny portion of the image was engulfed in smoke and flame in the early stages of the fire. Use of the complete picture feature without area recommendations would reduce detection accuracy and postpone fire detection and alarm activation if the feature of smoke and flame was not immediately apparent [12].

Therefore, to increase the algorithm's capacity for early fire detection, proposal areas have to be established prior to picture classification. Second, several researchers created algorithms for creating proposal areas by choosing characteristics by hand and categorizing proposal regions using CNNs [13]. This sort of approach uses individual computation to generate each proposal region instead of using CNNs for the overall detection process, which results in a lot of computation and a sluggish detection rate.

Vision Transformers (ViTs) achieve state-of-the-art performance on image recognition and is renowned as the pioneer of ViTs by replacing all CNN structures with multiple transformer layers [14]. Patch and positional embedding, Transformer encoder, and multi-layer perceptron (MLP) head are the three segments that make up the ViT. One of the most useful developments in deep learning research is the attention mechanism, which assesses the significance of a characteristic that affects the outcome. An attention mechanism can be used to train a model to focus on particular features. The input and output sizes are the same for self-attention, but the process allows for interaction between the inputs to choose which ones they should focus on more [15].

When a model is trained using self-supervised learning (SSL), labels are generated automatically from the data rather than manually. SSL offers a significant benefit over data exploitation, which is especially helpful for those large datasets. Additionally, it helps the model discover crucial information hidden in the data and strengthens and expands its applicability [16]. SSL in computer vision is now generally divided into contrastive learning and pretext tasks. The first is to provide a specific goal for a model to learn before honing it for later tasks, including predicting rotation degree, coloring, or jigsaw puzzle solving. The latter, however, produces comparable features for data of the same class and excludes more negative samples [17].

Our main contributions in this paper are as follows:

1. We propose a robust pipeline for detecting fire images which can be very useful in hazardous situations management.
2. We design a transformer-based model which verifies the decision made by the classifier.
3. We use both CNN-based and transformer-based models in the proposed pipeline and this ensures that the model learns both local and universal features making it more robust to augmented inputs.
4. We introduce a new dataset collected from real-world scenarios.

2. METHODOLOGY

This section includes our proposed approach for fire image detection. Fig. 1 demonstrates the overall procedure of the proposed algorithm.

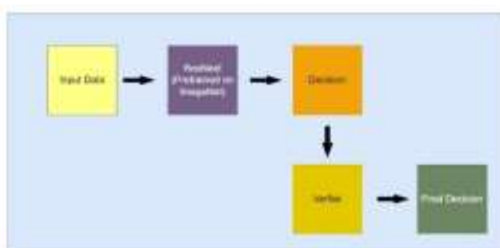

Fig. 1. An overview of the proposed methodology.

Based on Fig. 1, it can be seen that our proposed methodology includes two main stages. The first one is a CNN-based classifier and the second one is a transformer-based module, following the architecture of Masked Autoencoders for verification. This module, is in fact, beneficial for reducing false positive and false negative rates.

2.1. ResNext

By revisiting the 3 X 3 and 5 X 5 branches of Inception using the following illustration: Assume there are 256 channels in the input. The 256-dimensional input would be separated into lower-dimensional data using two sets of 1 X 1 convolutions (for example, 160 channels for the 3 X 3 branch and 128 channels for the 5 X 5 branch), transformed using 3 X 3 and 5 X 5 convolutions, and then combined using concatenation (leaving us with 288 = 160 + 128 channels) [18]. The results would be excellent. Unfortunately, as already said, inception layers must be thoroughly modified, making it challenging to modify them for new jobs and datasets.

The split-transform-merge approach used by Inception is therefore used by the developers of ResNeXt while also adhering to VGG's recurring blocks and branches with identical topologies tenet [19]. Using 1 X 1 convolutions, we may divide the input into several lower-dimensional tensors (for example, 4 channels). The next thing to think about is how many branches there should be: We want for all of our branches to have the same topologies so that we are not constrained by kernel sizes and may have as many branches as we wish. Inception has four branches for three different kernel sizes plus a pooling layer. 32 will do for the time being [20].

After that, we must change our data. We may use the tried-and-true 3 X 3 convolutions for all the branches since, once more, the transformations must be homogenous. Last but not least, the 32 branches' outputs need to be combined. Why not sum the 32 256-dimensional tensors to give the model access to the data in a tensor the same size as the input by once again using 1 X 1 convolutions to raise the breadth of the output of the transformations from 4 to 256? After that, we must change our data. We may use the tried-and-true 3 X 3 convolutions for all the branches since, once more, the transformations must be homogenous. Last but not least, the 32 branches' outputs need to be combined. We sum the 32 256-dimensional tensors to give the model access to the data in a tensor the same size as the input by once again using 1 X 1 convolutions to raise the breadth of the output of the transformations from 4 to 256 [21].

2.2. Masked Autoencoder

Autoencoders are a subset of the family of neural networks used in deep learning, which is mostly used to computer vision and Natural Language Processing (NLP). It can carry out the activities that are necessary for unsupervised learning to efficiently learn codes for unlabeled data. Regenerating the encoder's input serves to validate encoding in an autoencoder [22]. A neural network combination called an encoder learns how to represent a collection of data. The primary goal of unsupervised learning is dimension reduction, and we may train the network to reject irrelevant data representations by employing an autoencoder. Fig. 2 shows the overall architecture of an autoencoder [23].

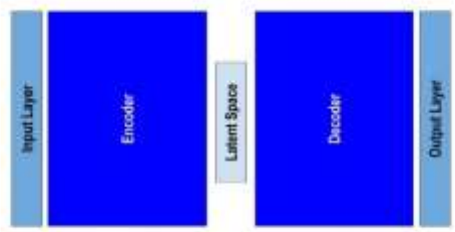

Fig. 2. A general architecture of an Autoencoder.

We can see, the autoencoder has successfully used the field of computer vision in several applications. Particularly when the picture space is continuous, however these autoencoders do not do well in the NLP field. Because the data are often discrete, autoencoding in NLP has historically been ineffective [24]. However, by encoding the words as vectors in word embedding, we frequently pretend that text is not discrete. However, word interpolation is frequently not as simple as it is in computer vision, which causes issues when we attempt to model the data in practice. Reconstruction loss is an issue brought on by the autoencoder's discreteness [25]. With the invention of the transformer, we were able to use NLP data to solve this issue. Consider the operation of BERT models, where the model first learns to anticipate the missing values in the data by corrupting a fraction of the tokens. Prediction may be the precise word rather than the mask tokens by identifying more than 30,000 words in the English lexicon. The mask autoencoder and this BERT model's method appear to be comparable [26].

We may use the masked autoencoder to carry out this similar technique for the computer vision data. Data encoding and decoding using an autoencoder has the benefit of self-supervision, as we are only reconstructing the input rather than requiring labelled data [27]. The autoencoder also has the advantage of being able to learn the generic representation of the modality. As opposed to the supervised learning component, when a model is permitted to forget anything from the label of that data sample, self-supervision requires a model to retain just the essential information. It is much simpler to use the NLP data than the picture data [28].

The majority of computer vision tasks use CNNs, and because these CNNs depend on the data regularity of picture grids, it is difficult to conceal objects in an image without imposing human-specified limitations in the convolutional output. Language and visual data have extremely different information densities. The pixels do not contain a lot of information when a word includes its meaning. Language and visual data are decoded differently from one another. We can create an autoencoder to predict comparatively few objects with strong semantic information by using a low-level MLP network in a decoder. Fig. 3 illustrates a general overview of a masked autoencoder.

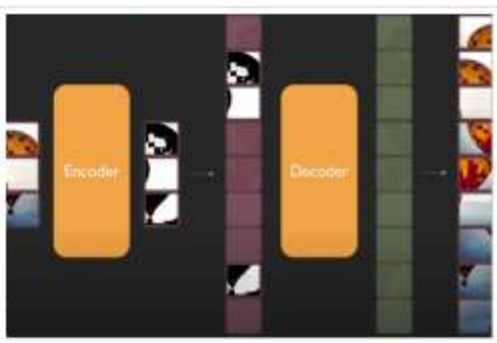

Fig. 3. The architecture of MAE.

2.3. Verifier

The proposed verifier module acts as a corrector of the classifier module. For this purpose, we first train a large vision learner, which in our proposed pipeline is based on MAE to learn general representation of the two classes in the dataset. Then the generated feature maps are gathered in one group in order to find their medoids. Next, we train a binarized CNN-based classifier on these embeddings to identify if the input of the pipeline is homogenous with the medoids of the classifier's decision. This way we can diagnose if the decision of the classifier is correct or not.

2.4. Transfer Learning

In the traditional supervised learning scenario of machine learning, we assume that we are given annotated data for the same task and domain when we want to train a model for some goal and domain A [29]. This is evident in Fig. 4, where the task and domain of our model A's training and test data are identical.

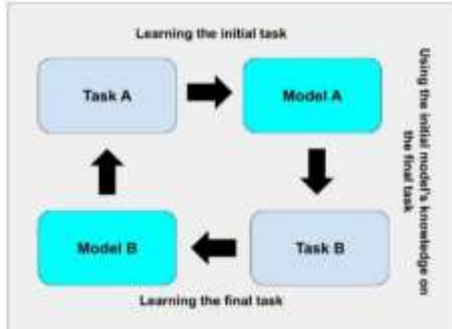

Fig. 4. The procedure of transfer learning in machine learning-based algorithms.

When we do not have enough labeled data for the task or we care about to train a solid model, the conventional supervised learning paradigm fails. We could use a model that has been trained in a related domain, such as recognize pedestrians on nighttime photographs [30]. However, in fact, we frequently see a decline or collapse in performance because the model has carried over the bias from its training data and is unable to generalize to the new domain. We cannot even re-train an existing model to do a new job, like spotting bikers, because the labels for the different tasks are different.

3. RESULTS AND DISCUSSION

3.1. Dataset

In this paper, we introduce a newly collected of 5000 fire images along with 5000 normal ones. The class of fire images contain a wide set of real-world fire images including fire in residential areas, forest fire, industrial areas, and etc. The normal images include hard examples of sunset scenes which resemble fire. Fig. 5 shows samples of the images from the collected dataset.

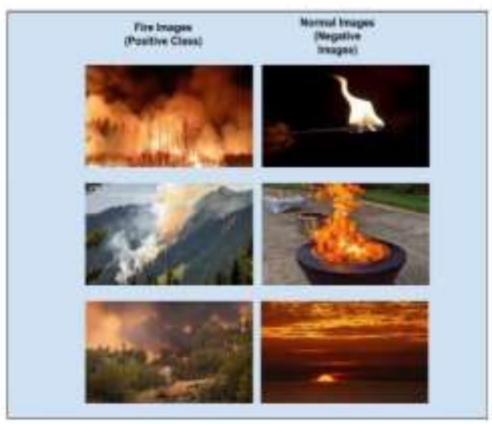

Fig. 5. Samples from the dataset.

3.2. Evaluation metrics

This subsection introduces all the metrics which are used for evaluating the proposed algorithm.

- Confusion Matrix (CM): a matrix containing number of True Positive (TP) , True Negative (TN), False Positive (FP), and False Negative (FN) samples.
- TP: positive samples correctly classified as positive
- TN: negative samples correctly classified as negative
- FN: positive samples incorrectly classified as negative
- FP: negative samples incorrectly classified as positive
- Accuracy: (TP + TN) / (TP+FP+TN+FN)
- Sensitivity or recall: TP / TP+FN
- Specificity: TN / TN + FP
- Precision: TP/TP+FP
- F1-Score: 2 * precision * recall / (precision + recall)

3.3. Classification Results

In this subsection, the achieved results are detailed. Figs. 6-10 show the CM achieved by the proposed classifier in a 5-fold cross validation setting. Also, Table 1 contains the achieved accuracy, precision, recall and F1-Score for the classifier.

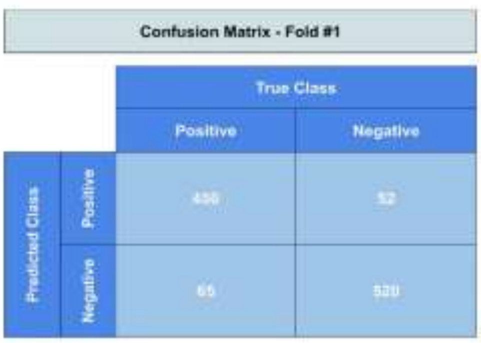

Fig. 6. CM for fold 1 before verification.

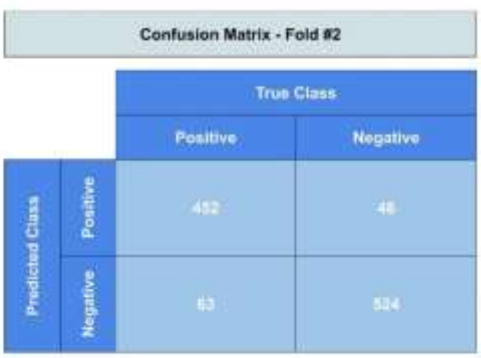

Fig. 7. CM for fold 2 before verification.

Fig. 8. CM for fold 3 before verification.

Fig. 9. CM for fold 4 before verification.

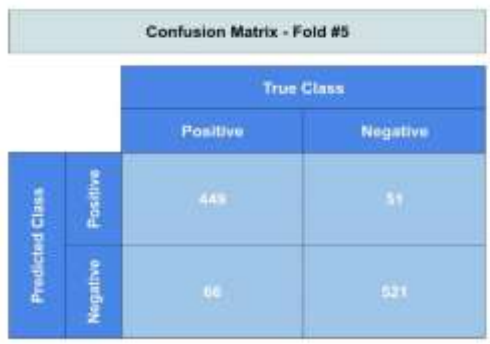

Fig. 10. CM for fold 5 before verification.

Table 1. Evaluation results based on the introduced metrics.

Fold	Accuracy	Precision	Recall	F1-Score
1	92.73	92.65	92.81	93.21
2	93.11	93.15	93.14	93.17
3	93.21	93.24	93.28	93.84
4	92.10	92.16	92.54	92.29
5	93.01	93.03	93.61	93.22

3.4. Verification Improvement

This subsection details the achieved CMs after verification module corrects the output of the classifier. Figs. 11-15 show the CMs after verification module is applied on the classifier's decision. Further, Table 2 shows the same metrics used in Table 1 after verification.

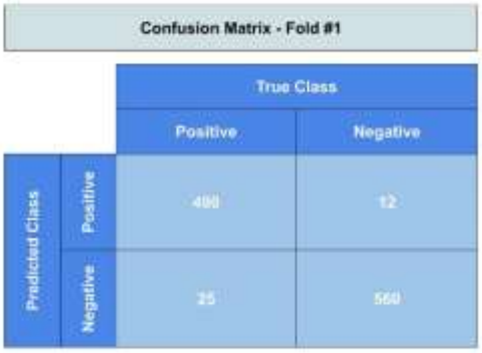

Fig. 11. CM for fold 1 after verification.

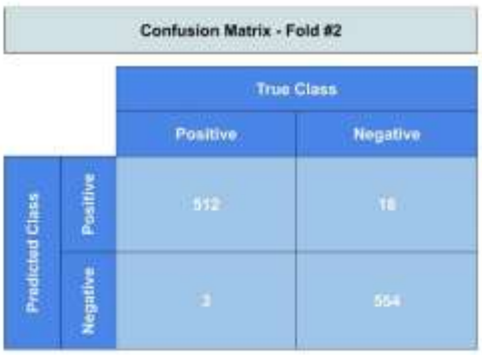

Fig. 12. CM for fold 2 after verification.

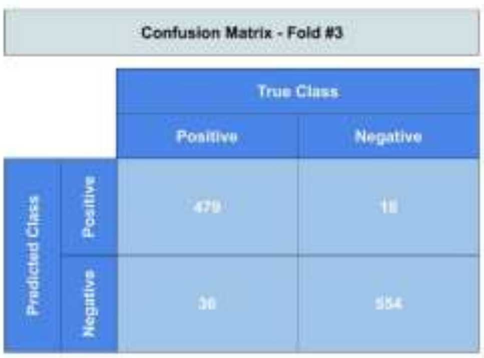

Fig. 13. CM for fold 3 after verification.

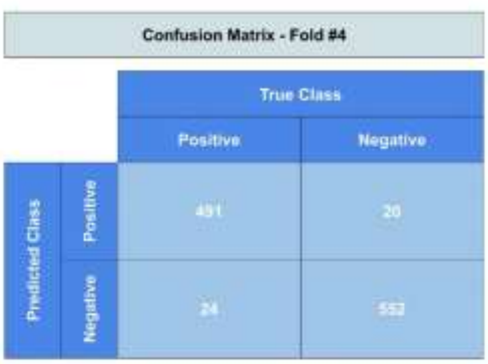

Fig. 14. CM for fold 4 after verification.

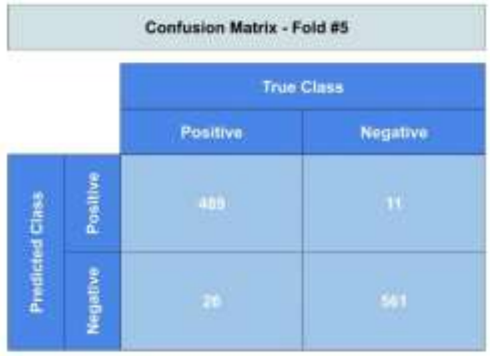

Fig. 15. CM for fold 5 after verification.

Table 2. Evaluation results after verification based on the introduced metrics.

Fold	Accuracy	Precision	Recall	F1-Score
1	95.24	95.65	95.62	95.64
2	95.13	95.19	95.51	95.27
3	96.19	95.71	95.28	95.91
4	95.18	95.25	95.46	95.31
5	95.60	95.61	95.22	95.34

As is seen from Table 1, it is proved that the verifier module successfully corrects the decision made by the classifier. To be more specific, the number of FN and FP samples are reduced after the verifier module is applied on the classifier. This is of great importance since the FN and FP samples in hazardous situations like forest fire can lead to mal-functioning in managerial procedures. According to Fig. 16, we can witness 3-4% improvement for the classifier.

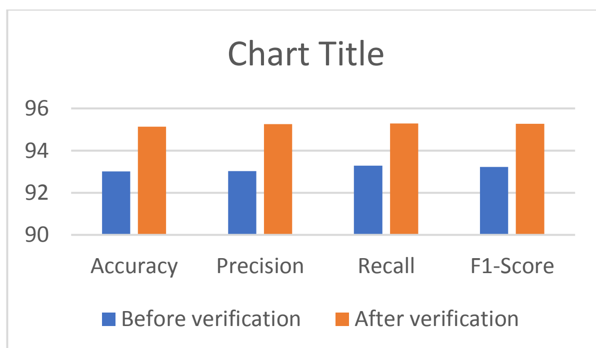

Fig. 16. Before and after verification accuracy, precision, recall, and F1-Score.

4. CONCLUSION

Since it can be difficult to locate the source of ignition in a timely manner, which commonly results in an increase in fire event intensity, wildfire detection is a time-critical application. This problem has generated a great deal of interest in the study of expert systems, and recent developments in deep learning for difficult visual interpretation tasks have opened up new research directions. The effectiveness of deep learning-based models has recently piqued researchers' curiosity from a wide range of domains. CNNs in particular have emerged as the best method for computer vision problems. Therefore, we provide a CNN-based pipeline in this study for identifying and validating fire-related photos.

REFERENCES

[1] Kim, Byoungjun, and Joonwhoan Lee. **"A video-based fire detection using deep learning models."** *Applied Sciences* 9, no. 14, p. 2862, 2019.

[2] Avazov, Kuldoshbay, Mukhriddin Mukhiddinov, Fazliddin Makhmudov, and Young Im Cho. **"Fire Detection Method in Smart City Environments Using a Deep-Learning-Based Approach."** *Electronics* 11, no. 1, p. 73, 2021.

[3] Xu, Renjie, Haifeng Lin, Kangjie Lu, Lin Cao, and Yunfei Liu. **"A forest fire detection system based on ensemble learning."** *Forests* 12, no. 2, p. 217, 2021.

[4] Kang, Li-Wei, I-Shan Wang, Ke-Lin Chou, Shih-Yu Chen, and Chuan-Yu Chang. **"Image-based real-time fire detection using deep learning with data augmentation for vision-based surveillance applications."** In *2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, no. 2, pp. 1-4. IEEE, 2019.

[5] Barmpoutis, Panagiotis, Kosmas Dimitropoulos, Kyriaki Kaza, and Nikos Grammalidis. **"Fire detection from images using faster R-CNN and multidimensional texture analysis."** In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, no.14, pp. 8301-8305. IEEE, 2019.

[6] Xue, Zhenyang, Haifeng Lin, and Fang Wang. **"A Small Target Forest Fire Detection Model Based on YOLOv5 Improvement."** *Forests* 13, no. 8, p. 1332, 2022.

[7] Molajou, Amir, Vahid Nourani, Abbas Afshar, Mina Khosravi, and Adam Brysiewicz. **"Optimal design and feature selection by genetic algorithm for emotional artificial neural network (EANN) in rainfall-runoff modeling."** *Water Resources Management* 35, no. 8, 2369-2384, 2021.

[8] Sharghi, Elnaz, Vahid Nourani, Hessam Najafi, and Amir Molajou. **"Emotional ANN (EANN) and wavelet-ANN (WANN) approaches for Markovian and seasonal based modeling of rainfall-runoff process."** *Water resources management* 32, no. 10, pp. 3441-3456, 2018.

[9] Nourani, Vahid, Zahra Razzaghzadeh, Aida Hosseini Baghanam, and Amir Molajou. **"ANN-based statistical downscaling of climatic parameters using decision tree predictor screening method."** *Theoretical and Applied Climatology* 137, no. 3, pp. 1729-1746, 2019.

[10] Monteiro, Vitor, Julio S. Martins, João Carlos Aparício Fernandes, and Joao L. Afonso. **"Review of a disruptive vision of future power grids: a new path based on hybrid AC/DC grids and solid-state transformers."** *Sustainability* 13, no. 16, pp. 9423, 2021.

[11] Monteiro, Vitor, Julio S. Martins, João Carlos Aparício Fernandes, and Joao L. Afonso. **"Review of a disruptive vision of future power grids: a new path based on hybrid AC/DC grids and solid-state transformers."** *Sustainability* 13, no. 16, p. 9423, 2021.

[12] Xiao, Tete, Mannat Singh, Eric Mintun, Trevor Darrell, Piotr Dollár, and Ross Girshick. **"Early convolutions help transformers see better."** *Advances in Neural Information Processing Systems* 34, pp. 30392-30400, 2021.

[13] Zhang, Jinnian, Houwen Peng, Kan Wu, Mengchen Liu, Bin Xiao, Jianlong Fu, and Lu Yuan. **"MiniViT: Compressing Vision Transformers with Weight Multiplexing."** In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, no. 5, pp. 12145-12154. 2022.

[14] Bazi, Yakoub, Laila Bashmal, Mohamad M. Al Rahhal, Reham Al Dayil, and Naif Al Ajlan. **"Vision transformers for remote sensing image classification."** *Remote Sensing* 13, no. 3, p. 516, 2021.

[15] Zhuang, Fuzhen, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. **"A comprehensive survey on transfer learning."** *Proceedings of the IEEE* 109, no. 1, pp. 43-76, 2020.

[16] Raffel, Colin, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. **"Exploring the limits of transfer learning with a unified text-to-text transformer."** *J. Mach. Learn. Res.* 21, no. 140, pp. 1-67, 2020.

[17] He, Kaiming, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. **"Masked autoencoders are scalable vision learners."** In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, no. 8, pp. 16000-16009. 2022.

[18] Giri, Ritwik, Srikanth V. Tenneti, Karim Helwani, Fangzhou Cheng, Umut Isik, and Arvindh Krishnaswamy. **"Unsupervised anomalous sound detection using self-supervised classification and group masked autoencoder for density estimation."** *Challenge on Detection and Classification of Acoustic Scenes and Events (DCASE 2020 Challenge), Tech. Rep*, no.7, p. 23, 202.

[19] Weihan, Wang. **"MAGAN: A masked autoencoder generative adversarial network for processing missing IoT sequence data."** *Pattern Recognition Letters*, no. 138, pp. 211-216, 2020.

[20] Khajenezhad, Ahmad, Hatef Madani, and Hamid Beigy. **"Masked autoencoder for distribution estimation on small structured data sets."** *IEEE Transactions on Neural Networks and Learning Systems* 32, no. 11, pp. 4997-5007, 2020.

[21] Shao, Zhenzhou, Ying Qu, Guangli Ren, Guohui Wang, Yong Guan, Zhiping Shi, and Jindong Tan. **"Batch Normalization Masked Sparse Autoencoder for Robotic Grasping Detection."** In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, no. 6, pp. 9614-9619. IEEE, 2020.

[22] Matsui, Ayaka, Shota Asahi, Satoshi Tamura, Satoru

Hayamizu, Ryosuke Isashi, Akira Furukawa, and Takayoshi Naitou. **"Anomaly detection in mechanical vibration using combination of signal processing and autoencoder."** *Journal of Signal Processing* 24, no. 4, pp. 203-206, 2020.

[23] Brand, Fabian, Kristian Fischer, and André Kaup. **"Rate-Distortion Optimized Learning-Based Image Compression using an Adaptive Hierachical Autoencoder with Conditional Hyperprior."** In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, no.22, pp. 1885-1889. 2021.

[24] Dou, Zeyang, Kun Gao, Xiaodian Zhang, Hong Wang, and Lu Han. **"Band selection of hyperspectral images using attention-based autoencoders."** *IEEE Geoscience and Remote Sensing Letters* 18, no. 1, pp. 147-151, 2020.

[25] de Rooij, Mark, and Wouter Weeda. **"Cross-validation: A method every psychologist should know."** *Advances in Methods and Practices in Psychological Science* 3, no. 2, pp. 248-263, 2020.

[26] Wong, Tzu-Tsung, and Po-Yang Yeh. **"Reliable accuracy estimates from k-fold cross validation."** *IEEE Transactions on Knowledge and Data Engineering* 32, no. 8, pp. 1586-1594, 2019.

[27] Fong, Edwin, and Chris C. Holmes**. "On the marginal likelihood and cross-validation."** *Biometrika* 107, no. 2, pp. 489-496, 2020.

[28] Lopez-del Rio, Angela, Alfons Nonell-Canals, David Vidal, and Alexandre Perera-Lluna. **"Evaluation of cross-validation strategies in sequence-based binding prediction using deep learning."** *Journal of chemical information and modeling* 59, no. 4, pp. 1645-1657, 2019.

[29] Tian, Chunwei, Lunke Fei, Wenxian Zheng, Yong Xu, Wangmeng Zuo, and Chia-Wen Lin. **"Deep learning on image denoising: An overview."** *Neural Networks*, no. 131, pp. 251-275, 2020.

[30] Bartlett, Peter L., Andrea Montanari, and Alexander Rakhlin. **"Deep learning: a statistical viewpoint."** *Acta numerica*, no. 30, pp. 87-201, 2021.