Research Article

Comparative Analysis of Deep Learning Algorithms in Fire Detection

Remzi Gocmen, Musa Cibuk and Erdal Akin

***Abstract*—As technology rapidly advances, deep learning applications, a subset of machine learning, are becoming increasingly relevant in various aspects of our lives. Essential daily applications like license plate recognition and optical character recognition are now commonplace. Alongside current technological progress, the development of future-integrated technologies such as suspicious situation detection from security cameras and autonomous vehicles is also accelerating. The success and accuracy of these technologies have reached impressive levels. This study focuses on the early and accurate detection of forest fires before they cause severe damage. Using primarily forest fire images from datasets obtained from Kaggle, various deep learning algorithms were trained via transfer learning using MATLAB. This approach allowed for comparing different deep learning algorithms based on their efficiency and accuracy in detecting forest fires. High success rates, generally exceeding 90%, were achieved.**

Index Terms—**Fire Detection, Image Processing, Deep Learning, Convolutional Neural Networks, Deep Learning Algorithms**

I. INTRODUCTION

FIRE HAS contributed to human development in many areas since its discovery. However, when a fire goes out of control, it can cause serious damage, making preventing loss of life and property critically important [1]. Early detection and intervention can significantly reduce the damage caused by fires. Despite the efforts of fire departments, timely intervention is sometimes hindered by factors such as traffic, delayed notifications, and the fire's location being unsuitable for immediate access. This tragic reality has driven researchers to develop and improve firefighting systems.

When it comes to forest fires, time constraints are the most significant obstacle, rather than urban factors like traffic or fire location. Forest fires spread rapidly and are often the last type of fire to be noticed. In our region, the frequent late detection and delayed intervention of forest fires are shrinking the green areas daily. This reduction in green spaces, combined with increased greenhouse gas emissions, accelerates global warming, climate change, environmental pollution, drought, and other undesirable negative effects, making our country and the world increasingly uninhabitable.

Today, firefighting units actively use computerized fire detection systems. Nevertheless, effective intervention remains challenging. Considering the physical structure of forest lands and the rapid spread of forest fires, early and accurate detection is essential for timely intervention and preventing major damage. Confirming fire notifications with computer vision and artificial intelligence systems can also prevent false alarms, providing significant benefits.

Point-type thermal and smoke sensors are commonly used, but they are often located close to the fire zone, making them susceptible to malfunction or damage. Advances in computer image processing have introduced video and image-based fire detection methods, which offer fast response times and wide detection areas compared to traditional methods [2]. High sensitivity, accurate and early detection, and prompt alarms are necessary to reduce fire losses. However, traditional fire detection technologies like smoke and heat detectors are unsuitable for large areas, complex buildings, or many disadvantaged areas. Due to these limitations, missed detections, false alarms, and detection delays often occur, complicating early fire warnings [3].

Recently, visual fire detection has become a popular research topic due to its many advantages, such as early detection, high accuracy, flexible system installation, and effective detection in large areas [4]. Deep learning (DL) algorithms analyze image data from cameras to determine the presence of fire or fire risk. Therefore, the detection algorithm is the core of this technology and directly affects the performance of visual fire detectors.

The main objective of this study is to compare Deep Learning (DL) Algorithms, which are essentially Artificial Neural Networks (ANNs), using different datasets, primarily forest fire images, to evaluate their performance. This comparison will enable the assessment of fire detection accuracy, detection speed, and sensitivity of the DL Algorithms used.

In this study, the structure, function, and working principles of Convolutional Neural Networks (CNN) systems were first

Remzi GÖÇMEN, is with Department of Valide Sultan Vocational and Anatolian High School in the Ministry of National Education, Bitlis, Turkey,(e-mail: r.gocmen2012@gmail.com).

https://orcid.org/0009-0004-3653-2831

Musa ÇIBUK[*], is with Department of Computer Engineering, Bitlis Eren University, Bitlis, Turkey,(e-mail: mcibuk@beu.edu.tr)

https://orcid.org/0000-0001-9028-2221

Erdal AKIN, is with 1 - Department of Computer Engineering, Bitlis Eren University, Bitlis, Turkey, 2 - Department of Computer Science and Media Technology, Malmö University, Malmö, Sweden, 3 -Internet of Things and People Centre, Malmo University, Malmö, Sweden (e-mail: e.akin@beu.edu.tr

https://orcid.org/0000-0002-2223-3927

Manuscript received Aug 19, 2024; accepted Sep 12, 2024.
DOI: 10.17694/bajece.1533966

BALKAN JOURNAL OF ELECTRICAL & COMPUTER ENGINEERING, Vol. 12, No. 3, September 2024

examined. A detailed literature review was then conducted on using these systems in fire recognition systems, investigating similar studies. Additionally, the use of datasets in fire recognition systems and their application in recognition processes were explored, with deep learning algorithms being examined in detail. The MATLAB [5] program, which offers many conveniences in coding and image processing, was used for these processes and algorithms. This program was also utilized to compare results, as well as to train and test the datasets. The five-fold cross-validation method was applied for verification in this study. Specifically, 20% of the dataset was used for testing, while the remaining 80% was used for training.

The rest of the paper is organized as follows: In Section 3, DL models used in this study are presented. In Section 4, we provided methods used for testing and training parameters of the compared algorithms. In Section 4, the results of the algorithms used in training are given with the details of the table graph and accuracy-time bubble graph. In Section 5, the conclusion section, the CNN architectural network with the fastest fire detection time and the best accuracy rate sensitivity is determined, and comments about the algorithm are given.

II. LITERATURE REVIEW

This section discusses DL-based CCN algorithms and datasets that will be used for this study

II.I. DL-based CNN algorithms

The emergence of DL algorithms is the result of the interest and concern for AI. . CNNs, one of the deep learning algorithms, are used in many areas such as classification [6] and object recognition [7]. In this study, 20 DL-based CNNs with different features and capabilities, developed to solve different problems, were preferred. These Algorithms are AlexNet [8-10], VGG-16 [11], VGG-19 [12, 13], GoogLeNet [14-18], Places-365 [19], ResNET-18 [20], ResNET-50 [21], ResNET-101 [22], ShufleNET [23, 24], MobileNET [25], NASNET-Mobile [26], EfficientNET-B0 [27], Inception-v3 [28, 29], DarkNET-19 [30, 31], DarkNET-53 [32-34], Xception [35], Inception-ResNet [36], DenseNET-201 [37, 38], SqueezeNET [39, 40], and NASNet-Large [41, 42].

II.II. Dataset

In this study, we used two different datasets obtained from Kaggle [43, 44]. The images are in JPG and PNG formats with varying sizes. One dataset contains only fire images, 110 of which are in JPG format. The other dataset includes images categorized as either fire or non-fire, with 755 fire images and 244 non-fire images in PNG format. The combined dataset is summarized in Table 1.

The data is organized into two folders: the "fire" folder contains 865 fire images, some of which include heavy smoke, while the "no fire" folder contains 244 images of natural scenes (e.g., forest, grass, river, people, foggy forest, lake and animal). For comparative analysis of deep learning algorithms in fire detection, both fire and non-fire images are needed. To measure the accuracy, speed, and sensitivity of fire detection algorithms, we utilized the Fire Images and Non-Fire Images datasets from the Fire Dataset, which are publicly available.

The dataset comprises a total of 1,109 images: 865 fire images and 244 non-fire images. In this study, 80% of the images (both fire and non-fire) were used for training, while the remaining 20% were used as test data. As the dataset size increases, the training process for the model also lengthens, which is a disadvantage of having a larger and more diverse dataset.

TABLE I.
FIRE DATASET INFORMATION

	Percent	**Dataset**	**Fire**	**Non-Fire**	**Total**
Training Data	80%	Kaggle	865	244	1109
Test Data	20%				

III. METHODOLOGY

A hybrid dataset was created using the images described in Section II.2. The images in this dataset were trained with the 5-fold cross-validation technique for each algorithm using MATLAB. In transfer learning, the fully connected and classification layers of a previously trained network were adapted to match the number of classes in the new dataset. Additionally, since the input dimensions of each network may vary, the images in the dataset were resized (preprocessed) to be compatible with the respective network. These operations were performed for each CNN used in the study. Fig. 1 illustrates the training of artificial neural networks via transfer learning.

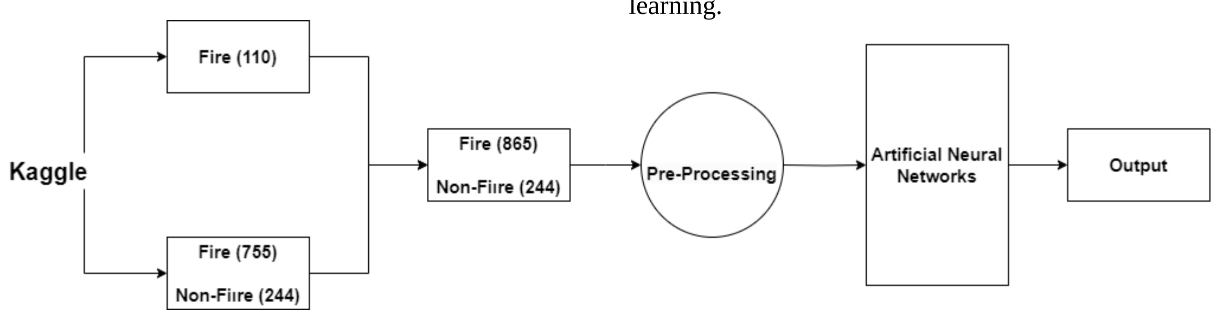

Fig. 1. Training ANNs with Transfer Learning

 ISSN: 2147-284X http://dergipark.gov.tr/bajece

It has been observed that the 5-fold cross-validation technique used in the classification phase provides better performance in ensemble learning classifiers [45]. The cross-validation technique is a model validation method that tests how a statistical analysis will yield a result on an independent data set [46]. Its main use is to estimate how accurately a prediction system will work in practice. The cross-validation technique involves dividing the original data set into training and test sets. In this study, the data set was divided into five parts. One of these five parts is used for testing, and the other for training. If we train once with the training set after dividing and measuring its accuracy with the test set, it may not give accurate results. Therefore, cross-validation is used as a more robust method. We obtain results by training on the other parts, excluding one of the divided training sets in each iteration. We find our verified training success by averaging these results. The main purpose of this process is to obtain a more general performance result by eliminating random errors or successes that may occur during the random division of the data set into training and test sets.

IV. EVALUATION

In this section, we present a comparative analysis of the twenty models mentioned above on the fire dataset. First, we introduce the environmental setup, training parameters, and comparison metrics. Then, we discuss the experimental results.

IV.I. Setup

The computer used in this work has an AMD Ryzen Threadripper PRO 3975WX 32 processor, 128 GB RAM, and an NVIDIA RTX 3090 GPU. The algorithms are programmed, trained, and tested using MatLab programming language.

IV.II. Training Parameters

All training and testing processes in this thesis were carried out using the parameters shown in Table II.

TABLE II.
TRAINING PARAMETERS

Training Options	**Value**
Solver	sgdm
MaxEpoch	100
MiniBatch	64
InitialLearnRate	0.001
VerboseFrequency	20
ExecutionEnviroment	gpu
InitialLearnRate	0.001
LearnRateDropFactor	30
LearnRateSchedule	Piecewise

IV.III. Evaluation Metric

In this study, some basic evaluation metrics were used to compare the success of the CNNs in fire detection. These can be explained using confusion matrix expressions. A confusion matrix summarizes the number of correctly or incorrectly predicted examples made by a classification model [47]. In our study, we compared the algorithms using the accuracy metric derived from the confusion matrix.

The calculation relies on a confusion matrix that consists of four key components. True Positives (TP) indicate instances where the predicted label correctly aligns with the ground truth label. True Negatives (TN) occur when an object is present in the image but is not labeled in either the ground truth or the model's prediction. Conversely, False Positives (FP) refer to cases where the predicted label does not exist in the ground truth, while False Negatives (FN) represent instances where the ground truth label is missed. Accordingly, accuracy can be calculated as followed Eq.(1)

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

IV.IV. Results and Discussion

In this study, the aim is to compare architectures used for the fast and accurate detection and classification of fire images. Transfer learning and CNN architectures from deep learning tools were utilized to classify fire images. The 5-fold cross-validation method was employed to train the fire images in the dataset used in the thesis. This method aimed to increase performance accuracy. Transfer learning was applied using various CNN architectures during training. The correct prediction rates of the CNN architectures used are given in Table III. Additionally, the minimum, maximum and average accuracy rates are determined and shown. Upon examining these results, it is evident that CNN models provide accuracy rates above 95%, with the best-performing model being Places365, which achieved 98.92% accuracy. Therefore, training with 0 success (numbers 1 and 4) are not included in the average success calculation.

TABLE III.
PERFORMANCE RESULTS OBTAINED AFTER THE TRAINING

Arch. Name	Train Nr.	Accuracy Rate	Min. Accuracy Rate	Max. Accuracy Rate	Average Accuracy Rate
AlexNet	1	0,977	0,968	1,000	0,980
	2	1,000			
	3	0,977			
	4	0,968			
	5	0,977			
VGG-16	1	0,982	0,982	1,000	0,988
	2	1,000			
	3	0,986			
	4	0,982			
	5	0,991			
VGG-19	1	0	0,779	0,986	0,917
	2	0,779			
	3	0,986			
	4	0			
	5	0,986			
GoogleNet	1	0,986	0,986	0,991	0,988
	2	0,991			
	3	0,986			
	4	0,986			
	5	0,991			
Places365	**1**	**0,977**	**0,977**	**1,000**	**0,989**
	2	**0,991**			
	3	**0,991**			
	4	**0,986**			
	5	**1,000**			
ResNet-18	1	0,973	0,973	0,991	0,984
	2	0,991			
	3	0,982			
	4	0,986			
	5	0,986			

ResNet-50	1	0,964	0,964	0,995	0,984
	2	0,986			
	3	0,995			
	4	0,986			
	5	0,986			
ResNet-101	1	0,968	0,968	0,991	0,984
	2	0,991			
	3	0,991			
	4	0,982			
	5	0,986			
Inception ResNetV2	1	0,968	0,941	0,982	0,959
	2	0,946			
	3	0,955			
	4	0,982			
	5	0,941			
Inception V3	1	0,959	0,959	0,986	0,974
	2	0,986			
	3	0,973			
	4	0,977			
	5	0,973			
Xception	1	0,977	0,973	0,986	0,980
	2	0,982			
	3	0,982			
	4	0,986			
	5	0,973			
MobileNet V2	1	0,955	0,955	0,995	0,978
	2	0,977			
	3	0,982			
	4	0,995			
	5	0,982			
DenseNet 201	1	0,964	0,964	0,991	0,981
	2	0,986			
	3	0,977			
	4	0,991			
	5	0,986			
ShuffleNet	1	0,968	0,968	0,995	0,984
	2	0,986			
	3	0,986			
	4	0,982			
	5	0,995			
DarkNet-19	1	0,982	0,977	1,000	0,986
	2	0,991			
	3	0,982			
	4	0,977			
	5	1,000			
DarkNet-53	1	0,977	0,973	0,991	0,982
	2	0,991			
	3	0,982			
	4	0,973			
	5	0,986			
Squeeze Net	1	0,968	0,968	0,991	0,980
	2	0,991			
	3	0,986			
	4	0,973			
	5	0,982			
Efficient Netb0	1	0,964	0,964	0,991	0,981
	2	0,991			
	3	0,977			
	4	0,982			
	5	0,991			
Nasnet Mobile	1	0,955	0,955	0,977	0,964
	2	0,968			
	3	0,964			
	4	0,977			
	5	0,955			
Nasnet Large	1	0,973	0,964	0,977	0,971
	2	0,968			
	3	0,964			
	4	0,973			
	5	0,977			

TABLE IV.
TRANSFER LEARNING TIME OF CNNS

Architecture Name	Train Nr.	Train Time (sec)	Min. Train Time(sec)	Max. Train Time(sec)	Average Train Time(sec)
AlexNet	1	1491,388	1197,836	1491,388	1363,152
	2	1304,289			
	3	1410,524			
	4	1197,836			
	5	1411,724			
VGG-16	1	1635,139	1375,095	1635,139	1552,484
	2	1375,095			
	3	1599,697			
	4	1565,348			
	5	1587,144			
VGG-19	1	102,502	1505,025	1662,549	1592,657
	2	1610,398			
	3	1662,549			
	4	56,179			
	5	1505,025			
Google Net	**1**	**1240,027**	**1190,833**	**1301,264**	**1247,654**
	2	**1190,833**			
	3	**1290,660**			
	4	**1301,264**			
	5	**1215,483**			
Places365	1	1437,911	1313,062	2013,673	1714,209
	2	2013,673			
	3	2001,327			
	4	1805,073			
	5	1313,062			
ResNet-18	1	1300,322	1289,123	1485,557	1354,328
	2	1311,404			
	3	1289,123			
	4	1485,557			
	5	1385,236			
ResNet-50	1	1645,830	1623,943	1667,269	1651,649
	2	1661,276			
	3	1659,924			
	4	1667,269			
	5	1623,943			
ResNet-101	1	2879,792	2780,590	2965,620	2868,681
	2	2780,590			
	3	2851,332			
	4	2866,070			
	5	2965,620			
Inception ResNetV2	1	6976,200	5199,917	6976,200	5825,375
	2	5344,980			
	3	5199,917			
	4	5611,612			
	5	5994,166			
Inception V3	1	2619,109	2498,057	2619,109	2539,804
	2	2505,140			
	3	2546,038			
	4	2498,057			
	5	2530,675			
Xception	1	2808,732	2808,732	2899,488	2853,818
	2	2899,379			
	3	2899,488			
	4	2823,931			
	5	2837,557			
MobileNet V2	1	1500,697	1471,133	1910,811	1651,502
	2	1471,133			
	3	1910,811			
	4	1582,319			
	5	1792,549			
DenseNet 201	1	6037,628	6037,628	6168,289	6121,425
	2	6093,988			
	3	6168,289			
	4	6157,341			
	5	6149,883			
ShuffleNet	1	1517,110	1336,175	1711,300	1519,916
	2	1472,689			
	3	1562,304			
	4	1336,175			
	5	1711,300			
DarkNet-19	1	1277,612	1277,612	1526,030	1426,151
	2	1444,860			
	3	1526,030			
	4	1379,137			
	5	1503,114			

DarkNet-53	1	2492,582	2433,511	2554,308	2490,212
	2	2486,306			
	3	2554,308			
	4	2484,351			
	5	2433,511			
Squeeze Net	1	1363,345	1278,258	1564,867	1391,218
	2	1278,258			
	3	1445,699			
	4	1303,923			
	5	1564,867			
Efficient Netb0	1	3425,724	3352,167	3566,620	3440,623
	2	3566,620			
	3	3355,723			
	4	3502,878			
	5	3352,167			
Nasnet Mobile	1	4083,264	3862,247	4117,353	4039,916
	2	4088,674			
	3	4117,353			
	4	4048,043			
	5	3862,247			
Nasnet Large	1	64203,660	50707,532	64203,660	55155,781
	2	53066,790			
	3	56133,841			
	4	50707,532			
	5	51667,084			

The transfer learning times of the CNNs on which the dataset was trained in this study are shown in Table IV. The table provides the minimum, maximum, and average learning times of 5 different CNNs in seconds. Although the most successful architecture according to the learning time results was VGG-19 (987.33 seconds), it was not taken into consideration due to the computational problems in VGG-19 (training numbered 1 and 4). When considering the learning times, it is evident that the CNN architecture with the most successful average results is GoogLeNet.

The accuracy-time bubble graph in Figure II allows us to comment on the performance rates and learning times of the CNN architectures used in the study by presenting both data together. This figure is essentially a visual combination of Table 3 and Table 4. When examining the accuracy-time bubble graph, it is evident that the fire detection times and correct prediction rates of CNN architecture models such as Place365, GoogLeNet, VGG-16, and DarkNet are quite high. Although 100% accuracy rates were obtained, this can be attributed to the ease of the selected dataset pieces. Therefore, the 5-fold cross-validation results are more significant.

Another notable point is that the VGG-19 architecture showed 0 success in 2 trainings. This can be explained by a computational error or a procedural error in the learning of VGG-19 for those distributions in the dataset. On the other hand, the fact that almost all CNN models (except VGG-19) have an average performance above 95% can be attributed to the small number of classes (2 classes) and the dataset adequately representing the purpose. As a result, in this study conducted using transfer learning for fire detection, it is seen that the GoogLeNet architecture stands out in terms of duration, and the Place365 architecture, based on GoogLeNet, stands out in terms of performance. With its complex and advanced structure, NasNet architectures did not perform as expected (specifically for this dataset) in terms of duration and performance.

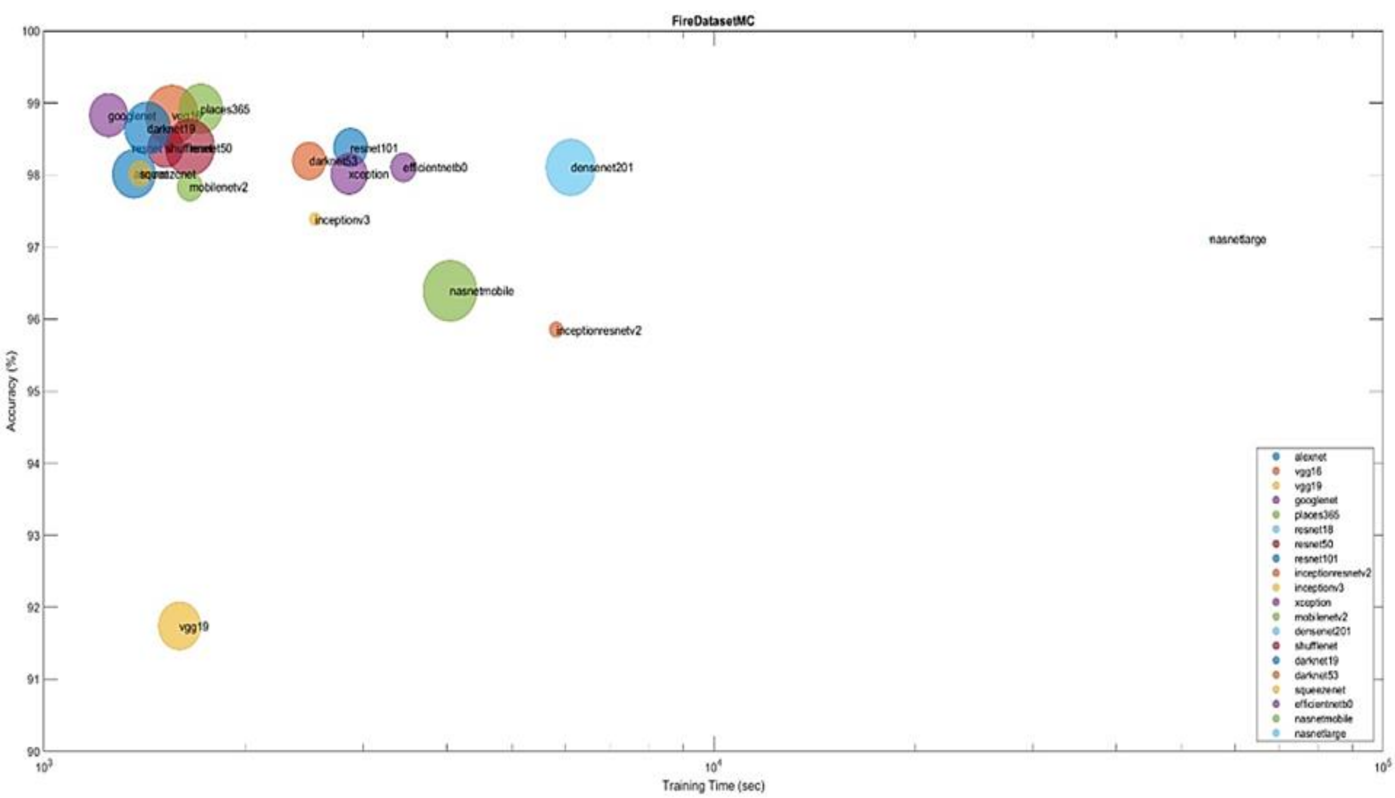

Fig. 2. Accuracy-Time Bubble Chart of Used Algorithms

V. CONCLUSION

In this study, although the performances (accuracy, training time) of CNNs, which we used for the development of fire detection technology from images and which are essentially ANNs, achieved 100% accuracy rates, this can be attributed to the ease of the selected dataset pieces. Therefore, the 5-fold cross-validation results are more significant. Another notable point is that the VGG-19 architecture (numbered 1 and 4) showed 0 performance in 2 trainings. This can be explained by a computational problem or an error in learning VGG-19 for those distributions in the dataset. On the other hand, the fact that almost all CNN models (except VGG-19) showed an average performance above 95% can be attributed to the small number of classes (2 classes) and the dataset adequately representing the purpose. Detailed accuracy-time bubble charts in Table 3, Table 4, and Figure 2 illustrate this.

According to the results obtained from 20 different CNN architectures, almost all algorithms demonstrate over 95% accuracy (generally around 98%) in detecting fire. If we compare the test results in the study, the algorithm with the highest success rate was Place365, with 98.92%. GoogLeNet was the best network, with an average training time of 1247.65 seconds.

The high performance of these algorithms highlights that the use of deep learning-based CNN architectures is an important alternative solution for preventing or minimizing loss of life and property by detecting fire early. This study has demonstrated that GoogLeNet-based CNN architectures (GoogLeNet and Place365) provide more effective results in fire detection. For future studies, it is recommended to further enhance the performance of these GoogLeNet-based deep learning architectures by customizing them specifically for fire detection purposes.

REFERENCES

[1]. K. Avazov, M. Mukhiddinov, F. Makhmudov, and Y. I. Cho, “Fire detection method in smart city environments using a deep-learning-based approach,” *Electron.*, vol. 11, no. 1, pp. 1–17, 2022, doi: 10.3390/electronics11010073.

[2]. 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)*, Dec. 2016, pp. 150–153. doi: 10.1109/ICIICII.2016.0045.

[3]. P. Li and W. Zhao, “Image fire detection algorithms based on convolutional neural networks,” *Case Stud. Therm. Eng.*, vol. 19, p. 100625, Jun. 2020, doi: 10.1016/j.csite.2020.100625.

[4]. K. Muhammad, J. Ahmad, I. Mehmood, S. Rho, and S. W. Baik, “Convolutional Neural Networks Based Fire Detection in Surveillance Videos,” *IEEE Access*, vol. 6, pp. 18174–18183, 2018, doi: 10.1109/ACCESS.2018.2812835.

[5]. G. Lindfield and J. Penny, “Numerical methods: Using MATLAB,” *Numer. Methods Using MATLAB*, pp. 1–608, 2018, doi: 10.1016/C2016-0-00395-9.

[6]. M. Cıbuk, U. Budak, Y. Guo, M. Cevdet Ince, and A. Sengur, “Efficient deep features selections and classification for flower species recognition,” *Meas. J. Int. Meas. Confed.*, vol. 137, pp. 7–13, 2019, doi: 10.1016/j.measurement.2019.01.041

[7]. R. Daş, B. Polat, and G. Tuna, “Derin Öğrenme ile Resim ve Videolarda Nesnelerin Tanınması ve Takibi,” *Fırat Üniversitesi Mühendislik Bilim. Derg.*, vol. 31, no. 2, pp. 571–581, 2019, doi: 10.35234/fumbd.608778.

[8]. C. Alippi, S. Disabato, and M. Roveri, “Moving Convolutional Neural Networks to Embedded Systems: The AlexNet and VGG-16 Case,” in *2018 17th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, Apr. 2018, vol. 30, no. 2010, pp. 212–223. doi: 10.1109/IPSN.2018.00049.

[9]. H. Ismail Fawaz *et al.*, “InceptionTime: Finding AlexNet for time series classification,” *Data Min. Knowl. Discov.*, vol. 34, no. 6, pp. 1936–1962, 2020, doi: 10.1007/s10618-020-00710-y.

[10]. A. LeNail, “NN-SVG: Publication-Ready Neural Network Architecture Schematics,” *J. Open Source Softw.*, vol. 4, no. 33, p. 747, Jan. 2019, doi: 10.21105/joss.00747.

[11]. S. Tammina, “Transfer learning using VGG-16 with Deep Convolutional Neural Network for Classifying Images,” *Int. J. Sci. Res. Publ.*, vol. 9, no. 10, p. p9420, 2019, doi: 10.29322/ijsrp.9.10.2019.p9420.

[12]. K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, pp. 1–14, 2015.

[13]. M. Mateen, J. Wen, Nasrullah, S. Song, and Z. Huang, “Fundus Image Classification Using VGG-19 Architecture with PCA and SVD,” *Symmetry (Basel).*, vol. 11, no. 1, p. 1, Dec. 2018, doi: 10.3390/sym11010001.

[14]. G. Zeng, Y. He, Z. Yu, X. Yang, R. Yang, and L. Zhang, “Preparation of novel high copper ions removal membranes by embedding organosilane-functionalized multi-walled carbon nanotube,” *J. Chem. Technol. Biotechnol.*, vol. 91, no. 8, pp. 2322–2330, 2016, doi: 10.1002/jctb.4820.

[15]. M. Längkvist, L. Karlsson, and A. Loutfi, “Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning,” *Pattern Recognit. Lett.*, vol. 42, no. 1, pp. 11–24, 2014, [Online]. Available: http://arxiv.org/abs/1512.00567

[16]. H.-J. Yoo, “Deep Convolution Neural Networks in Computer Vision: a Review,” *IEIE Trans. Smart Process. Comput.*, vol. 4, no. 1, pp. 35–43, 2015, doi: 10.5573/ieiespc.2015.4.1.035.

[17]. C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the Inception Architecture for Computer Vision,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-Decem, pp. 2818–2826, 2016, doi: 10.1109/CVPR.2016.308.

[18]. C. Szegedy *et al.*, “Going deeper with convolutions,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 07-12-June, pp. 1–9, 2015, doi: 10.1109/CVPR.2015.7298594.

[19]. “GoogLeNet evrişimli sinir ağı-MATLAB googlenet.” [Online]. Available: https://www.mathworks.com/help/deeplearning/ref/googlenet.html#mw_d60d4ed6-a2c9-44f1-93b8-977191c6cfea

[20]. M. Guo and Y. Du, “Classification of Thyroid Ultrasound Standard Plane Images using ResNet-18 Networks,” in *2019 IEEE 13th International Conference on Anti-counterfeiting, Security, and Identification (ASID)*, Oct. 2019, pp. 324–328. doi: 10.1109/ICASID.2019.8925267.

[21]. I. Z. Mukti and D. Biswas, “Transfer Learning Based Plant Diseases Detection Using ResNet50,” in *2019 4th International Conference on Electrical Information and Communication Technology (EICT)*, Dec. 2019, pp. 1–6. doi: 10.1109/EICT48899.2019.9068805.

[22]. P. Ghosal, L. Nandanwar, S. Kanchan, A. Bhadra, J. Chakraborty, and D. Nandi, “Brain Tumor Classification Using ResNet-101 Based Squeeze and Excitation Deep Neural Network,” in *2019 Second International Conference on Advanced Computational and Communication Paradigms (ICACCP)*, Feb. 2019, pp. 1–6. doi: 10.1109/ICACCP.2019.8882973.

[23]. J. Bobo, C. Hudley, and C. Michel, “The Black studies reader,” *Black Stud. Read.*, pp. 1–488, 2004, doi: 10.4324/9780203491348.

[24]. R. Zhang *et al.*, “Automatic Segmentation of Acute Ischemic Stroke From DWI Using 3-D Fully Convolutional DenseNets,” *IEEE Trans. Med. Imaging*, vol. 37, no. 9, pp. 2149–2160, 2018, doi: 10.1109/TMI.2018.2821244.

[25]. B. Khasoggi, E. Ermatita, and S. Samsuryadi, “Efficient mobilenet architecture as image recognition on mobile and embedded devices,” *Indones. J. Electr. Eng. Comput. Sci.*, vol. 16, no. 1, p. 389, Oct. 2019, doi: 10.11591/ijeecs.v16.i1.pp389-394.

[26]. F. Saxen, P. Werner, S. Handrich, E. Othman, L. Dinges, and A. Al-Hamadi, “Face Attribute Detection with MobileNetV2 and NasNet-Mobile,” in *2019 11th International Symposium on Image and Signal*

Processing and Analysis (ISPA), Sep. 2019, pp. 176–180. doi: 10.1109/ISPA.2019.8868585.

[27]. R. H. Hridoy, F. Akter, M. Mahfuzullah, and F. Ferdowsy, “A Computer Vision Based Food Recognition Approach for Controlling Inflammation to Enhance Quality of Life of Psoriasis Patients,” in *2021 International Conference on Information Technology (ICIT)*, Jul. 2021, pp. 543–548. doi: 10.1109/ICIT52682.2021.9491783.

[28]. M. Nikhitha, S. Roopa Sri, and B. Uma Maheswari, “Fruit Recognition and Grade of Disease Detection using Inception V3 Model,” in *2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA)*, Jun. 2019, vol. 9, pp. 1040–1043. doi: 10.1109/ICECA.2019.8822095.

[29]. Xiaoling Xia, Cui Xu, and Bing Nan, “Inception-v3 for flower classification,” in *2017 2nd International Conference on Image, Vision and Computing (ICIVC)*, Jun. 2017, pp. 783–787. doi: 10.1109/ICIVC.2017.7984661.

[30]. Redmon, Joseph, and Ali Farhadi. "YOLO9000: better, faster, stronger." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 7263-7271. 2017.

[31]. J. Redmon and A. Farhadi, “YOLOv3: An Incremental Improvement,” Apr. 2018, doi: 10.11772/j.issn.1001-9081.2018102190.

[32]. Huang, Xin, Xinxin Wang, Wenyu Lv, Xiaying Bai, Xiang Long, Kaipeng Deng, Qingqing Dang et al. "PP-YOLOv2: A practical object detector." arXiv preprint arXiv:2104.10419 (2021).

[33]. J. Deng, W. Dong, R. Socher, L.-J. Li, Kai Li, and Li Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2009, pp. 248–255. doi: 10.7287/peerj.preprints.27880v1.

[34]. R. S. T. De Menezes, J. V. A. Luiz, A. M. Henrique-Alves, R. M. Santa Cruz, and H. Maia, “Mice Tracking Using The YOLO Algorithm,” pp. 162–173, 2020, doi: 10.5753/semish.2020.11326.

[35]. F. Chollet, “Xception: Deep Learning with Depthwise Separable Convolutions,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017, pp. 1800–1807. doi: 10.1109/CVPR.2017.195.

[36]. Y. Bhatia, A. Bajpayee, D. Raghuvanshi, and H. Mittal, “Image Captioning using Google’s Inception-resnet-v2 and Recurrent Neural Network,” in *2019 Twelfth International Conference on Contemporary Computing (IC3)*, Aug. 2019, pp. 1–6. doi: 10.1109/IC3.2019.8844921.

[37]. G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 2261–2269, 2017, doi: 10.1109/CVPR.2017.243.

[38]. K. Zhang, Y. Guo, X. Wang, J. Yuan, and Q. Ding, “Multiple Feature Reweight DenseNet for Image Classification,” *IEEE Access*, vol. 7, pp. 9872–9880, 2019, doi: 10.1109/ACCESS.2018.2890127.

[39]. F. Iandola, A. Shaw, R. Krishna, and K. Keutzer, “SqueezeBERT: What can computer vision teach NLP about efficient neural networks?,” in *Proceedings of SustaiNLP: Workshop on Simple and Efficient Natural Language Processing*, 2020, pp. 124–135. doi: 10.18653/v1/2020.sustainlp-1.17.

[40]. F. Özyurt, E. Sert, E. Avci, and E. Dogantekin, “Brain tumor detection based on Convolutional Neural Network with neutrosophic expert maximum fuzzy sure entropy,” *Measurement*, vol. 147, p. 106830, Dec. 2019, doi: 10.1016/j.measurement.2019.07.058.

[41]. D. Ghimire, D. Kil, and S. Kim, “A Survey on Efficient Convolutional Neural Networks and Hardware Acceleration,” *Electronics*, vol. 11, no. 6, p. 945, Mar. 2022, doi: 10.3390/electronics11060945.

[42]. S. I. Hossain *et al.*, “Exploring convolutional neural networks with transfer learning for diagnosing Lyme disease from skin lesion images,” *Comput. Methods Programs Biomed.*, vol. 215, p. 106624, Mar. 2022, doi: 10.1016/j.cmpb.2022.106624.

[43]. A. Saied, “FIRE Dataset,” *Kaggle.com*, 2018. https://www.kaggle.com/datasets/phylake1337/fire-dataset (accessed Jul. 30, 2024).

[44]. A. Kumar, “Fire Detection Dataset,” *Kaggle.com*, 2024. https://www.kaggle.com/datasets/atulyakumar98/test-dataset (accessed Jul. 30, 2024).

[45]. M. Burukanli, M. Çibuk, and Ü. Budak, “Saldırı Tespiti için Makine Öğrenme Yöntemlerinin Karşılaştırmalı Analizi Comparative Analysis of Machine Learning Methods for Intrusion Detection,” *BEU J. Sci.*, vol. 10, no. 2, pp. 613–624, 2021.

[46]. R. Kohavi, “A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection,” *Int. Jt. Conf. Artif. Intell.*, no. March 2001, 1995.

[47]. F. Uyanık and M. C. and Kasapbaşı, “Telekomünikasyon Sektörü için Veri Madenciliği ve Makine Öğrenmesi Teknikleri ile Ayrılan Müşteri Analizi,” *Düzce Üniversitesi Bilim ve Teknoloji Dergisi*, 2021.

BIOGRAPHIES

Remzi GÖÇMEN, He received his undergraduate degree from Marmara University Istanbul Department of Electrical and Electronics Education in 1998, and his master's degree from Bitlis Eren University Department of Electrical and Electronics Engineering in 2023. He currently works within the Ministry of National Education. His main areas of interest are: Deep Learning, Artificial Neural Networks

Musa ÇIBUK, He got his bachelors' degree in the Electrical-Electronic Engineering Department at Fırat University, Elazığ/Turkey in 1997, his master degree in the Electrical-Electronic Engineering Department at Fırat University, Elazığ/Turkey in 2002 and PhD degree in the Electrical-Electronic Engineering Department at Fırat University, Elazığ/Turkey in 2009. He is still an academic member of the Computer Engineering Department at Bitlis Eren University. His major areas of interests are: Protocol Design, Sensor Networks, Artificial Neural Networks and Image Processing

Erdal AKIN was born in Ceyhan, Adana, Turkey. He received the degree from the Department of Mathematics, Yildiz Technical University, Istanbul, in 2008, and the master’s and Ph.D. degrees from the Department of Computer Science, The University of Texas at San Antonio (UTSA), San Antonio, TX, USA, in 2014 and 2018, respectively. His research interests include deep reinforcement learning, software-defined networks, security, the IoT, computer vision, and blockchain. In 2009, he received the Republic of Turkey Ministry of National Education Scholarship to study in the USA.