


Article

Automated Fire Extinguishing System Using a Deep Learning Based Framework

Senthil Kumar Jagatheesaperumal ¹, Khan Muhammad ^{2,*}, Abdul Khader Jilani Saudagar ³
and Joel J. P. C. Rodrigues ^{4,5}

¹ Department of Electronics & Communication Engineering, Mepco Schlenk Engineering College, Sivakasi 626005, Tamil Nadu, India

² Visual Analytics for Knowledge Laboratory (VIS2KNOW Lab), Department of Applied Artificial Intelligence, School of Convergence, College of Computing and Informatics, Sungkyunkwan University, Seoul 03063, Republic of Korea

³ Information Systems Department, College of Computer and Information Sciences, Imam Mohammad Ibn Saud Islamic University (IMSIU), Riyadh 11432, Saudi Arabia

⁴ College of Computer Science and Technology, China University of Petroleum (East China), Qingdao 266555, China

⁵ Instituto de Telecomunicações, 6201-001 Covilhã, Portugal

* Correspondence: khan.muhammad@ieee.org or khanmuhammad@g.skku.edu

Abstract: Fire accidents occur in every part of the world and cause a large number of casualties because of the risks involved in manually extinguishing the fire. In most cases, humans cannot detect and extinguish fire manually. Fire extinguishing robots with sophisticated functionalities are being rapidly developed nowadays, and most of these systems use fire sensors and detectors. However, they lack mechanisms for the early detection of fire, in case of casualties. To detect and prevent such fire accidents in its early stages, a deep learning-based automatic fire extinguishing mechanism was introduced in this work. Fire detection and human presence in fire locations were carried out using convolution neural networks (CNNs), configured to operate on the chosen fire dataset. For fire detection, a custom learning network was formed by tweaking the layer parameters of CNN for detecting fires with better accuracy. For human detection, Alex-net architecture was employed to detect the presence of humans in the fire accident zone. We experimented and analyzed the proposed model using various optimizers, activation functions, and learning rates, based on the accuracy and loss metrics generated for the chosen fire dataset. The best combination of neural network parameters was evaluated from the model configured with an Adam optimizer and softmax activation, driven with a learning rate of 0.001, providing better accuracy for the learning model. Finally, the experiments were tested using a mobile robotic system by configuring them in automatic and wireless control modes. In automatic mode, the robot was made to patrol around and monitor for fire casualties and fire accidents. It automatically extinguished the fire using the learned features triggered through the developed model.

Keywords: convolution neural networks; fire detection; surveillance; deep learning; computer vision

MSC: 68T01



Citation: Jagatheesaperumal, S.K.; Muhammad, K.; Saudagar, A.K.J.; Rodrigues, J.J.P.C. Automated Fire Extinguishing System Using a Deep Learning Based Framework.

Mathematics **2023**, *11*, 608. <https://doi.org/10.3390/math11030608>

Academic Editors: Vladimir V. Arlazarov and Konstantin Bulatov

Received: 26 December 2022

Revised: 18 January 2023

Accepted: 21 January 2023

Published: 26 January 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Fire accidents are drastically increasing in all parts of the world. This impacts people in large numbers and also affects the environment. Therefore, there are inevitable risks involved in industries and other organizations due to fire. With reference to the statistics reported in [1], fire risk dropped to the 10th rank in 2019 compared with the previous year, where it was ranked 3rd. In addition to these statistics, fire risks are comparatively high in private industries. Although there is a drop in ranking compared with other risks, a risk of fire is inevitable. The oxygen content in the air, the heat emitted from surfaces, and

sources of fuel are the three main ingredients of most fires. As heat can come from any hot surfaces, electrical equipment, static electricity, as well as open flames, appropriate countermeasures are required to resist them. Fuel can be any flammable liquid, gas, wood, paper, dust, and certain metals. The oxygen requirements are readily available in the air [2]. A fire only needs 16% of oxygen in the atmosphere to ignite, whereas humans need about 21% oxygen in the air in order to breathe. Once a fire has started, it can be stopped if any one of these three elements is eliminated [3]. This brings up probable solutions for fire prevention and shows the importance of keeping these three ingredients under control [4]. Hazardous fire accidents are happening in textile industries, chemical industries, forests, and even in subways. Subway fire accidents also occur all around the world and the number of accidents keep increasing over the year [5]. All fire accidents are severe life hazards and cause long-term effects in some cases. Extinguishing a fire after it spreads over a large area is a very tedious task and consumes too much time to extinguish the fire completely. Early fire detection is the only feasible solution to address such issues. Among the numerous fire detection techniques, most fire detection systems are implemented based on sensors to detect the fire [6]. Sensor-based fire detection systems require close proximity to detect the fire and cannot be used over a large area. Deep learning (DL) is excessively deployed in most of the domains [7], including natural language processing and image processing [8,9]. DL techniques are characterized by their capability to learn from data, creating a mathematical model with the provided data with identity recognition [10] as one of its crucial applications. DL models exhibit high accuracy of detection in all applications. Therefore, we used DL techniques to detect fire by building a convolutional neural network (CNN) model [6].

A DL approach, along with spatial and temporal features, were considered for fire and smoke detection through surveillance cameras in [11]. Here, region-based CNN was used for extracting local and global features to reduce false positives in fire and smoke detection. In another similar work [12], DL models were used in video-based fire detection using region-based CNN and LSTM models. Their method interpreted the dynamic behavior of fire and smoke through the temporal changes in their features. The experiments were well-performed over long- and short-range videos, thereby significantly reducing false alarms. Later on, Govil et al. [13] used remote cameras to monitor and detect wildfires using a DL approach. In addition, they used cloud-based task management in handling a stream of images from ground-based and terrestrial cameras. The complexity of the DL is described in detail in [14,15].

Salameh et al. [16] surveyed the usage of IoT-based systems for fire and gas leakage detection applications, particularly focusing on wireless sensor networks. In [17], the authors experimented with fire smoke detection in aircraft cargo for fire and non-fire aerosol identification. Further, they distinguished between black and white smoke based on asymmetry ratios. Oil storage tank monitoring with an IoT-based framework, presented in [18], focused on the detection of smoke and fire, as well as monitored temperature. Further, it also assisted with real-time surveillance to alert industrial authorities to prevent accidents.

1.1. Research Gap and Motivation

Although many studies have been conducted on the use of DL techniques for disaster management [19], studies on fire detection using robotic systems with onboard sensors are rather scarce. Few studies were found to address fire disasters through a genetic algorithm approach [20]. In addition, the growing interest in DL techniques and their applications for fire detection needs to be explored by addressing their potential and challenges. To better estimate the prediction accuracy for fire accidents, we need to explore the inherent capabilities of DL frameworks and their properties and impact on the accidents [21].

In this paper, the investigations were carried out using a fire dataset collected from the Kaggle benchmark dataset [22], named outdoor-fire and non-fire images for computer vision tasks (Version 1) to analyze and predict the occurrence of fires in a specific environment. The collected dataset was composed of 3638 images. Moreover, deep neural network implementation was performed by introducing various activation functions, optimizers, and learning rates that were applied to the collected dataset [23].

1.2. Objectives and Goal of the Study

Based on an in-depth analysis of the literature, it was revealed that many studies have used DL techniques for fire-extinguishing applications. However, none of these studies explored a fire accident scenario with automated extinguishing nor employed robotic systems for maneuvering across the environment for the detection and extinguishing of the fire. This work provides the following set of outcomes, with primary focus on the DL technique for fire detection tasks and initiating extinguishing operations:

- Fire detection is carried out using a CNN model configured to operate on the chosen fire dataset.
- A DL-based framework is developed and trained to automatically classify the fire scenarios.
- Human detection is carried out using Alex-net with a pre-trained set of data from the fire datasets with a human involved during fire accidents.
- A mobile robotic system with onboard sensors is employed for detecting the fire.

The remaining sections of this article are as follows: Section 2 provides a description of the proposed CNN for fire and human detection. Section 3 presents and compares the results obtained using different optimizers, activation layers, and learning rates. Finally, our conclusions are provided in Section 4, along with future research directions.

2. Proposed Framework

This section is dedicated to presenting the proposed framework for the automated fire extinguishing system and further discussions on the overall functional modules of the system.

2.1. Learning-Based Automated Fire Extinguishing System

The overall functional block diagram of the proposed system is shown in Figure 1. Initially, the system is in patrol mode around the suspicious fire-prone area. The data acquisition unit collects the data (image/video sequences) to be assessed for fire and humans in the environment. The processing unit is configured to access the neural network via a cloud service. The corresponding neural network model predicts the output scenario and sends back the predicted output to the processing unit, which in turn operates corresponding actuation and alert systems [24]. The main focus of this work was to analyze a fire detection model with various metrics. Alex-net architecture has shown great results for human detection [25]. Therefore, human detection was performed using the predefined Alex-net model, utilizing a transfer learning framework, and further analysis was performed on the proposed neural network model for fire detection.

2.2. Fire Detection Model

Deep CNN methods are extensively applied in the fire detection and classification process. A large number of CNN models have been built with their own advantages and limitations over the last decade, during which most research was focused on fundamental DL models.

Our proposed framework for fire detection using the CNN model was driven by a pre-processed sequence of images from the fire dataset and tested using a portion of the dataset to evaluate its performance. Further, during the testing phase, the fire images acquired from the camera mounted on the robotic system were fed to the CNN model to decide the actuation subsystem for extinguishing the fire [26].

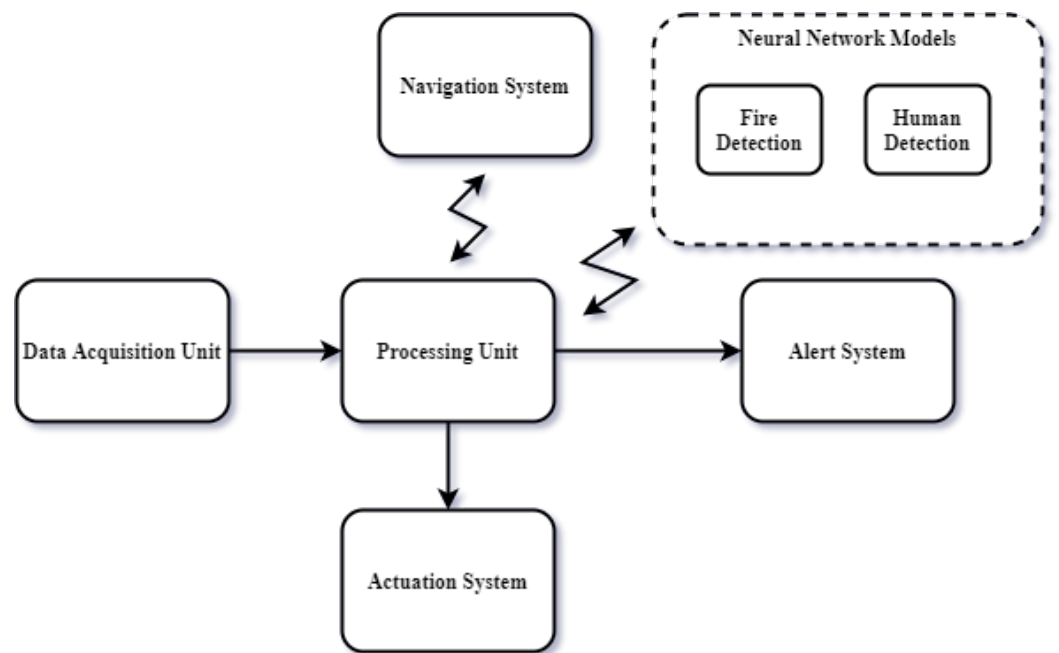


Figure 1. Overall functional block diagram of learning-based automated fire extinguishing system.

2.2.1. Description of the Dataset

The dataset of outdoor-fire and non-fire images for computer vision tasks (Version1) was created by a team of experts during the NASA Space Apps Challenge in 2018 [22], with the goal of using them in a benchmark dataset to develop a model that could recognize the fires in these scenarios. The chosen dataset with 999 images (755 fire and 244 non-fire) was augmented with a few more images, so that the final dataset comprised 3638 images (2528 fire, 737 non-fire, and 373 smoke) collected under different environmental conditions, such as heavy smoke. Further, the non-fire images consisted of natural images with trees, rivers, people, animals, roads, and waterfalls. Few studies have used this dataset among the research community [27,28] for fire detection applications. Some of the sample images in the training set are shown in Figure 2.

In the pre-processing stage, in order to classify the fire and non-fire images in the dataset, a histogram of the images was taken, which was meant to eliminate noise and other residual features. Further, based on the intensity of the background content in the images, redundant background features were eliminated. Subsequently, image enhancement was performed on the background-separated images. As a final stage input to the CNN model, object representation tasks were performed to identify the fire pattern in the images.



Figure 2. Sample fire and non-fire images in the training dataset [22].

2.2.2. Proposed CNN Architecture

The CNN architecture used here is similar to the architecture used in [29], with important tweaks in the layers and filter size to suit the input pre-processed fire dataset, and is shown in Figure 3. This developed model included three convolution layers, two

pooling layers, and two dense layers. The input color image (RGB), shaped to $150 \times 150 \times 3$ in size, was applied to the first convolution layer, which had 32 filters with a kernel size of 3×3 . The size of the kernel depends on the stride and slide, considering the size of the input image as $W \times W \times D$. Further, in a spatial dimension of F kernels, P padding, and S strides, the output volume could be computed as shown in Equation (1).

$$W_o = \frac{W - F + 2P}{S} + 1 \tag{1}$$

This would produce an output of size $W_o \times W_o \times D_o$. The resulting (148×148) feature maps were passed through the MaxPooling layer with strides $(2,2)$. For the MaxPooling layer, strides S and kernel size F could be specified based on the depth of the number of activation maps D . Equation (2) specifies the output dimension of the image, after passing through the MaxPool layer.

$$P_o = \left(\frac{W_o - F}{S} + 1 \right) \times D \tag{2}$$

The resultant 74×74 feature maps were passed through the second convolutional layer, which had 64 filters with the same 3×3 kernel size. Further, this was applied to a MaxPooling layer of the same specifications used in the first MaxPooling layer. The resultant 36×36 feature maps were passed on to the last convolutional layer. Finally, for the last convolutional layer, fully connected layers were included for performing the classification tasks. In the fully connected layer, all the inputs had separable weights for each of the output units. The final output layer consisted of two neurons engaged in classifying the fire and non-fire images.

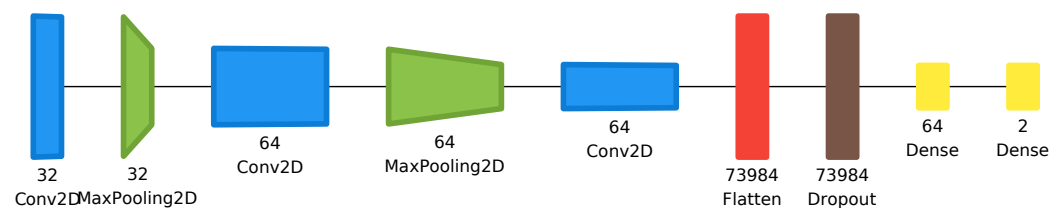


Figure 3. CNN architecture for fire detection [30].

3. Analysis of Results and Discussion

This section discusses the experimental results of our proposed CNN model with various metrics. The metrics used to analyze the model included different optimizers, activation functions, and learning rates, which are elaborated on in the subsequent sections. All the metrics were evaluated by experimenting with the model for 100 epochs. The rest of this section elaborates on the various metrics used in the proposed model.

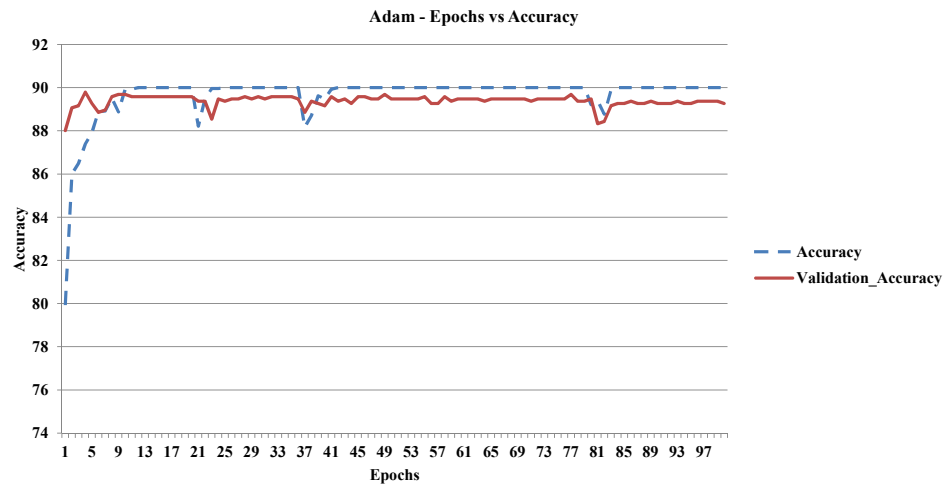
3.1. Optimizer Analysis

Optimizers are algorithms used to change the attributes of neural network such as weights and learning rates to reduce losses. Various optimizers are available for neural networks that could be used, based on the data and demands of the users [31]. The optimizers used for the analysis of the proposed network were Adam, Adamax, Nadam, RMSprop, and Sgd. Adam is an optimizer for CNN, developed to overcome the disadvantages of the stochastic gradient descent [32], which was first published in 2014 and presented at the machine learning conference ICLR in 2015.

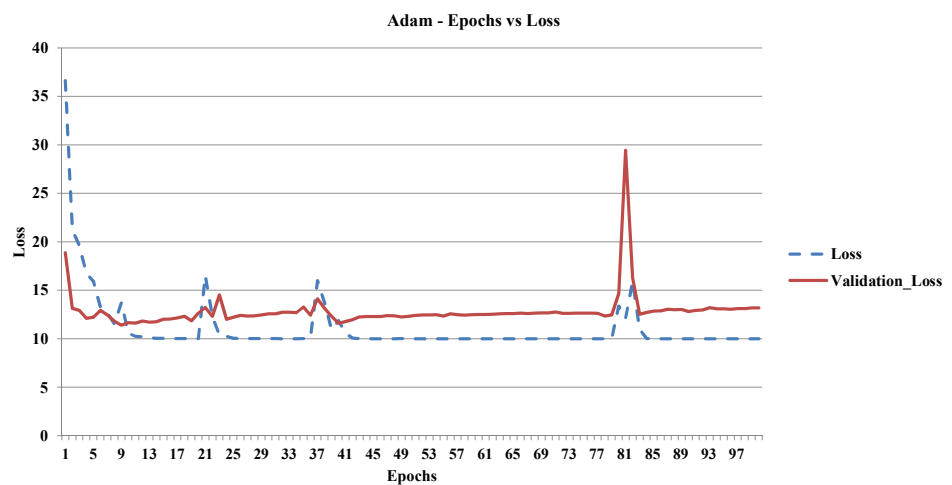
The performance evaluation of the developed model was analyzed with an appropriate choice of optimizer. Based on the choice of each optimizer, the corresponding observations in performance were discussed for each of the models, as follows:

Adam optimizer: With the choice of the Adam optimizer for analysis, Figure 4a shows a plot with the number of epochs vs accuracy (%) observed for 100 epochs. It is observed that an accuracy of approximately 90% was achieved at the 13th epoch in the training phase.

However, up to 85 epochs, there were a few distortions observed, which considerably affects the accuracy. After the 85th epoch, the training accuracy was saturated and only less deviation from training to validation accuracy was observed. Figure 4b shows the number of epochs vs. loss (%) graph. Similar to the accuracy plot, minimum loss was achieved in a fewer number of epochs. However, there were more distortions up to the 85th epoch. Further, the loss was also saturated and deviations from training loss to validation loss were minimized.



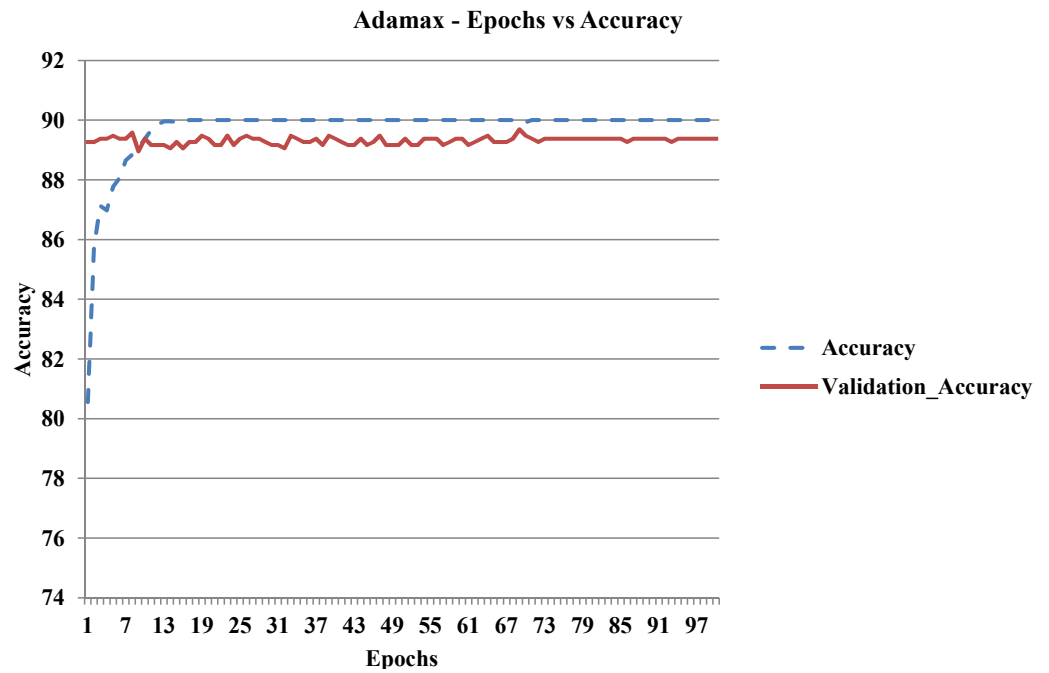
(a) Number of Epochs vs. Accuracy (%)



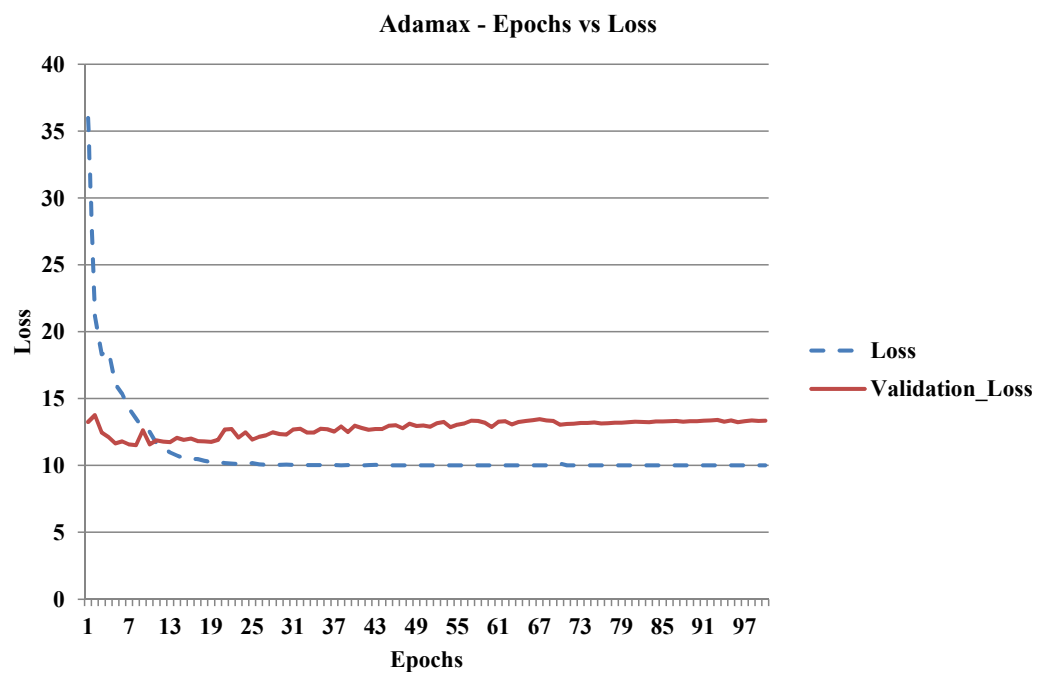
(b) Number of Epochs vs. Loss (%)

Figure 4. Accuracy and loss graph on training and validation data of Adam optimizer.

Adamax: Adamax is one of the variants of Adam based on the infinity norm feature [33]. The crucial advantage of Adamax is that it is less sensitive to hyperparameters such as learning rate. Figure 5a shows the observed accuracy with a plot showing the number of epochs vs. accuracy (%) for 100 epochs of the Adamax optimizer. Figure 5b shows the observed loss with a plot of the number of epochs vs. loss (%) for 100 epochs of the Adamax optimizer. While using the Adamax optimizer, the accuracy and loss were more or less saturated in fewer epochs with negligible deviations. The average percentage of accuracy and loss was also similar to Adam. However, the Adam optimizer was chosen for further analysis because the change in learning rate may not have produced notable changes in the results, as it was less sensitive to its hyperparameters [34].



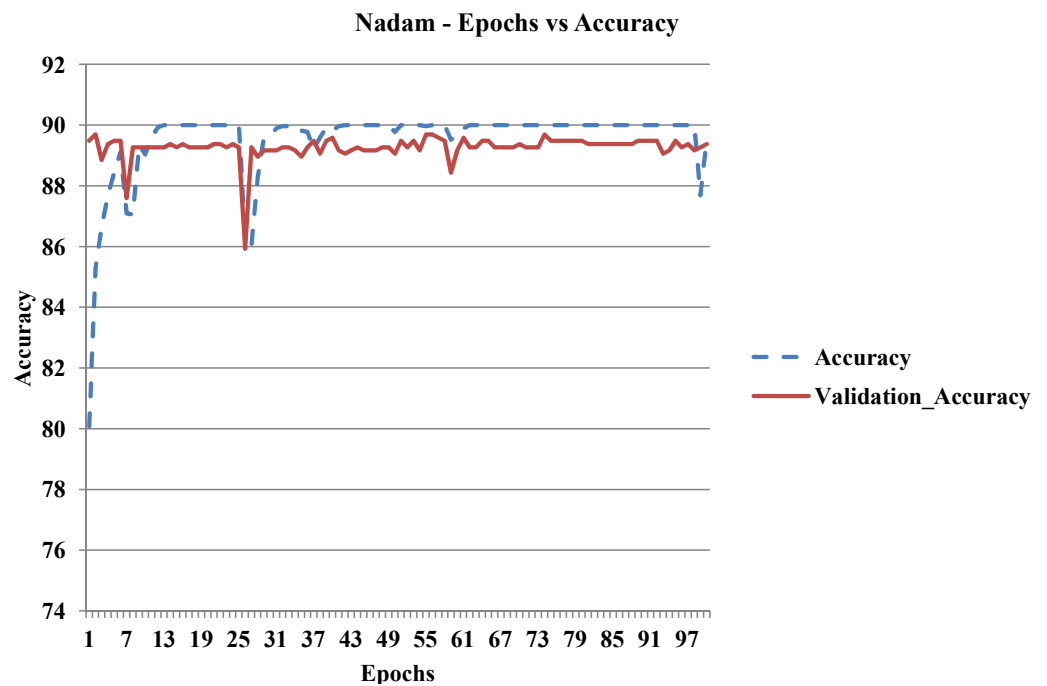
(a) Number of Epochs vs. Accuracy (%)



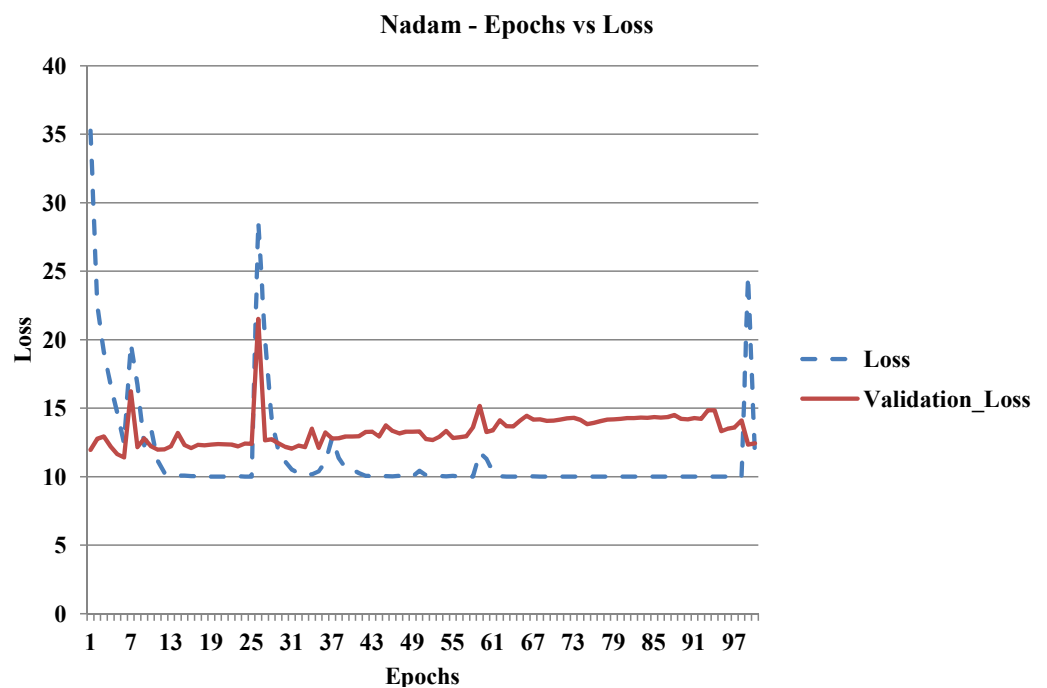
(b) Number of Epochs vs. Loss (%)

Figure 5. Accuracy and loss graph on training and validation data of Adamax optimizer.

Nadam: This is an acronym for Nesterov and Adam and was developed by modifying Adam’s momentum concept by incorporating the Nesterov momentum, which is the significant advantage of NAG (Nesterov’s accelerated gradient) [35]. Nadam works better for complex models; however, its prediction time is larger than the time taken for prediction using Adam [36]. Figure 6a,b shows the epochs vs. accuracy and loss graph for 100 epochs of the Nadam optimizer. In our case, the accuracy and loss were not saturated until 100 epochs. The deviations between the training and validation accuracy were permissible; however, loss variation was high.



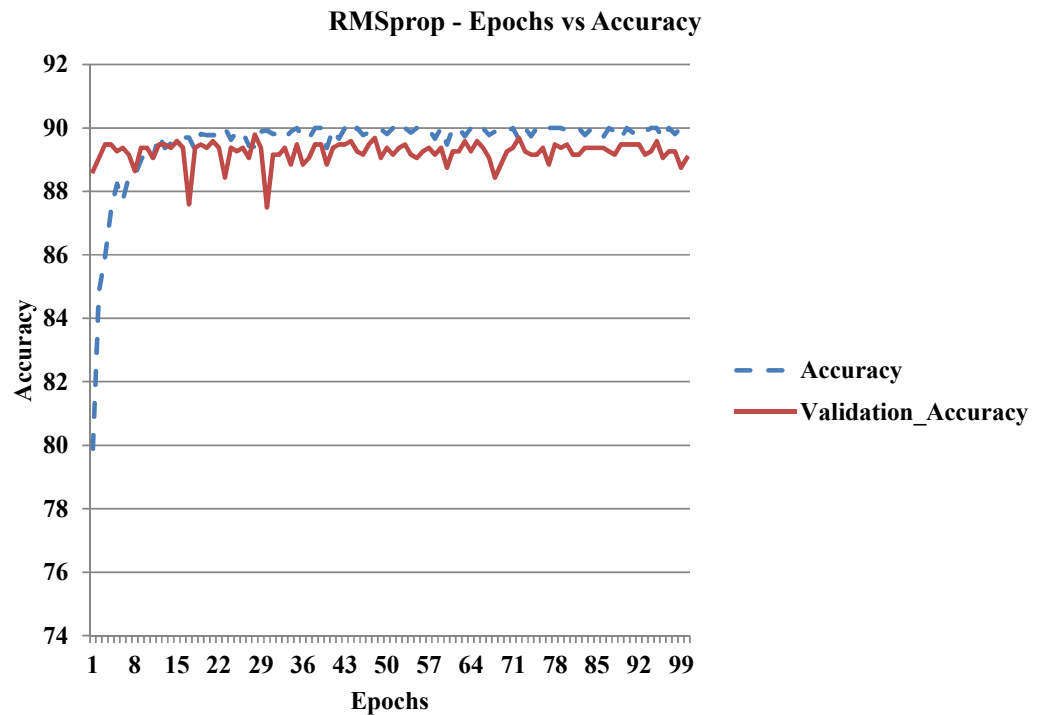
(a) Number of Epochs vs. Accuracy (%)



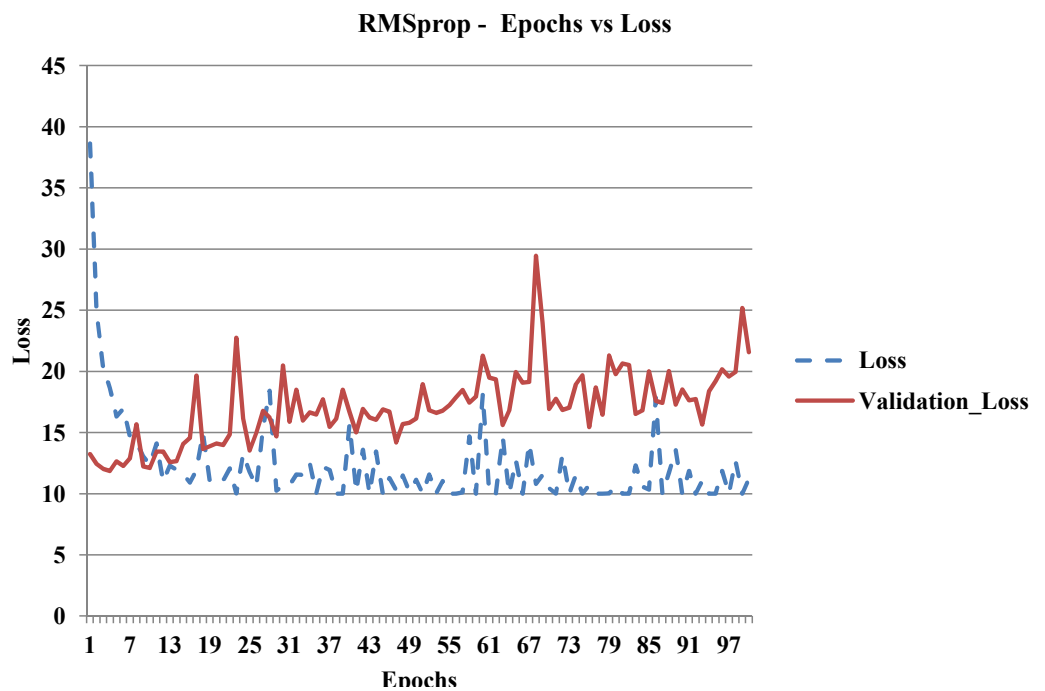
(b) Number of Epochs vs. Loss (%)

Figure 6. Accuracy and loss graph on training and validation data of Nadam optimizer.

RMSprop: RMSprop is an unpublished, adaptive learning rate algorithm; however, [37] shows its popularity until 2017 and it remains popular now. It uses the magnitude of recent gradients to normalize the gradients. Figure 7a shows a plot of the number of epochs vs. accuracy (%) based on the observation of 100 epochs of the RMSprop optimizer. Figure 7b shows a plot of the number of epochs vs. accuracy (%) observed for 100 epochs of the RMSprop optimizer. Unfortunately, both accuracy and loss were not saturated up to 100 epochs.



(a) Number of Epochs vs. Accuracy (%)

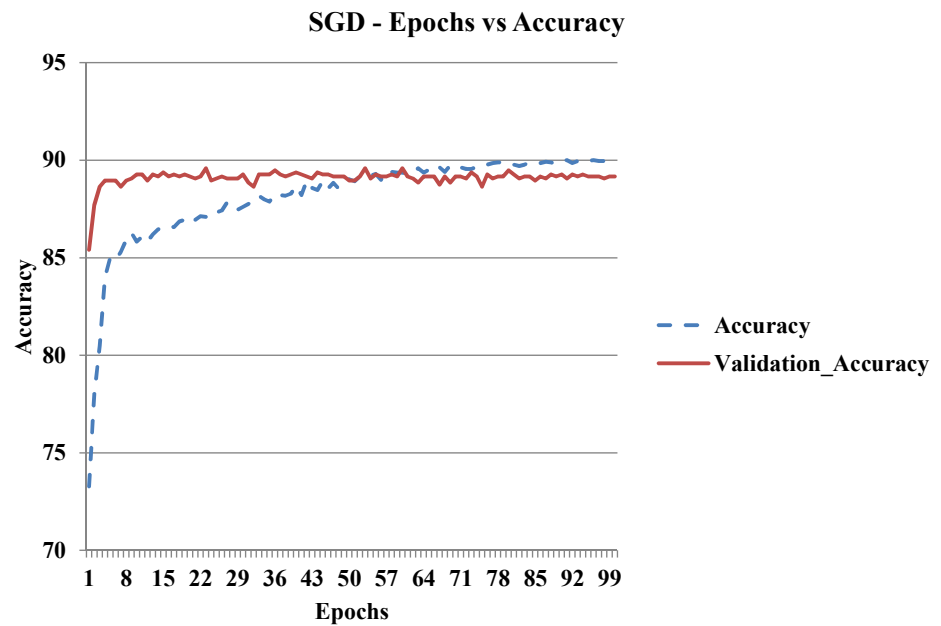


(b) Number of Epochs vs. Loss (%)

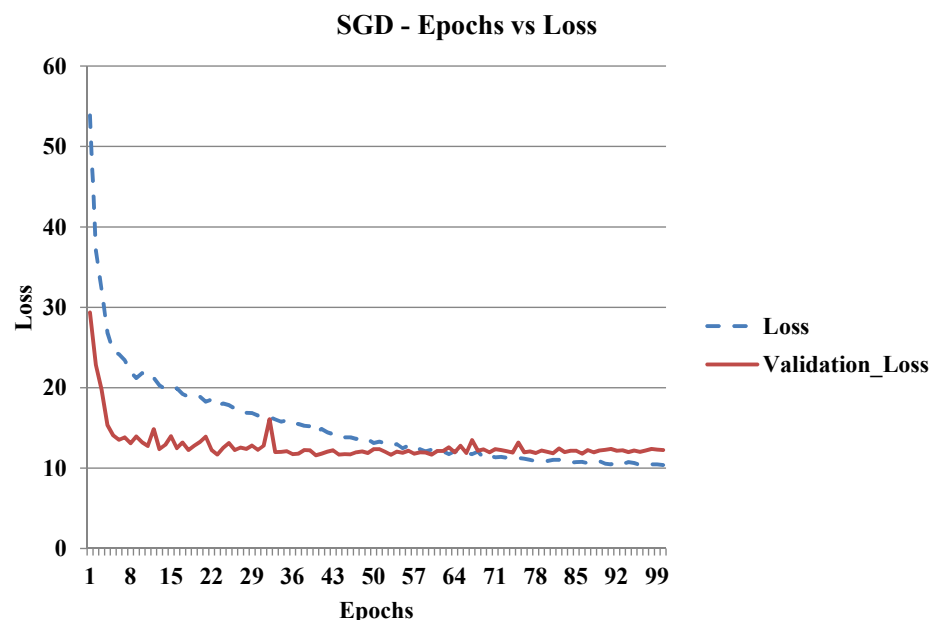
Figure 7. Accuracy and loss graph on training and validation data of RMSprop optimizer.

SGD: Stochastic gradient descent (SGD) is a simple, classical method that calculates the gradient of each weight on the network [38]. This method is very slow compared with other optimizers; nevertheless, it is good for shallow networks. Figure 8a shows a plot with the number of epochs vs. accuracy (%) observed for the SGD optimizer. Figure 8b shows the performance observations for the losses incurred in the SGD optimizer. The maximum range of accuracy was achieved gradually (epoch 50) compared with the other optimizers used for the analysis. Minimum loss was also obtained compared with another optimizers. SGD produced better accuracy and loss characteristics. However, due to its

slow convergence rate, SGD is not commonly used nowadays as numerous advanced optimization algorithms have evolved.



(a) Number of Epochs vs. Accuracy (%)



(b) Number of Epochs vs. Loss (%)

Figure 8. Accuracy and loss graph on training and validation data of SGD optimizer.

We further compared the average accuracy and loss values of each optimizer by taking the average of values generated for all epochs. Table 1 shows the performance of the developed model with various optimizers, tabulated with the average accuracy, validation accuracy, and validation losses. It was observed that the losses in Adamax were significantly lower than in the other variants.

Table 1. Comparison with various optimizers.

Optimizer	Accuracy (%)	Validation Accuracy (%)	Loss (%)	Validation Loss (%)
Adam	89.6463	89.3862	11.0794	12.8375
Adamax	89.7150	89.3163	10.8713	12.8152
Nadam	89.5313	89.2661	11.4997	13.3637
RMSprop	89.5390	89.2379	12.2243	17.0270
SGD	88.1809	89.0939	15.2573	12.7738

Although Adam and Adamax produced nearly the same accuracy (as shown in Table 1), the Adam Optimizer was used for further analysis due to having the best accuracy among the different optimizers, considering their validation accuracy and time.

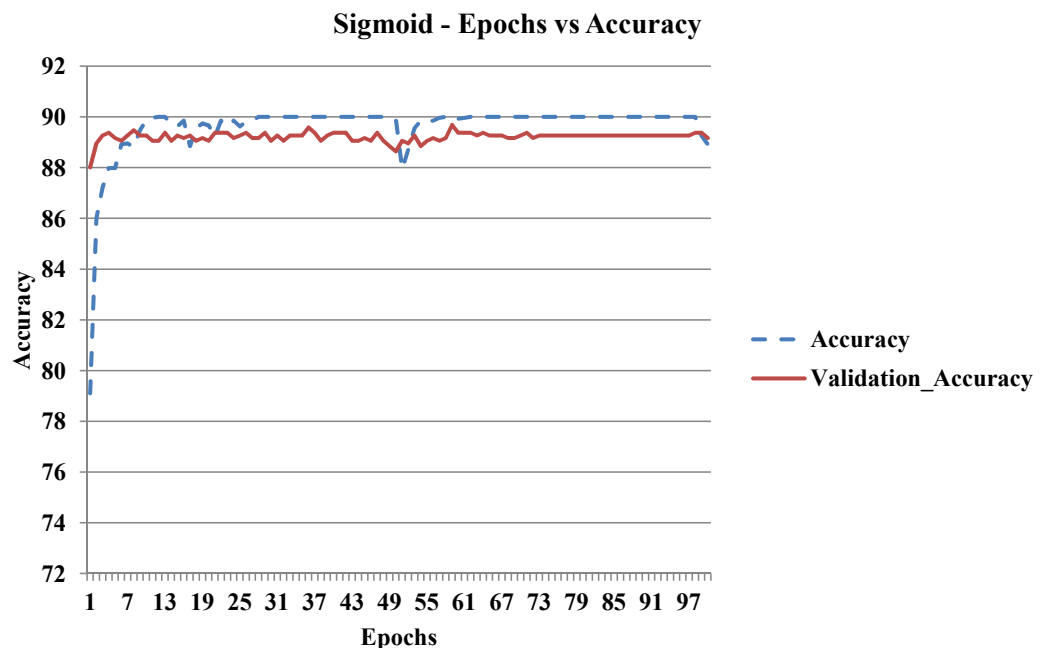
3.2. Activation Layer Analysis

The activation function is a node that is placed at the end or in between the neural networks, which decides whether the neuron will fire or not. Different activation functions can be used for different purposes [39]. Different activation functions produce different results based on their operations. Sigmoid, Softmax, and Softplus activation functions were used here to analyze our neural network. The Sigmoid function is mathematically expressed, as shown in Equation (3):

$$S(x) = \frac{1}{1 + e^{-x}} \tag{3}$$

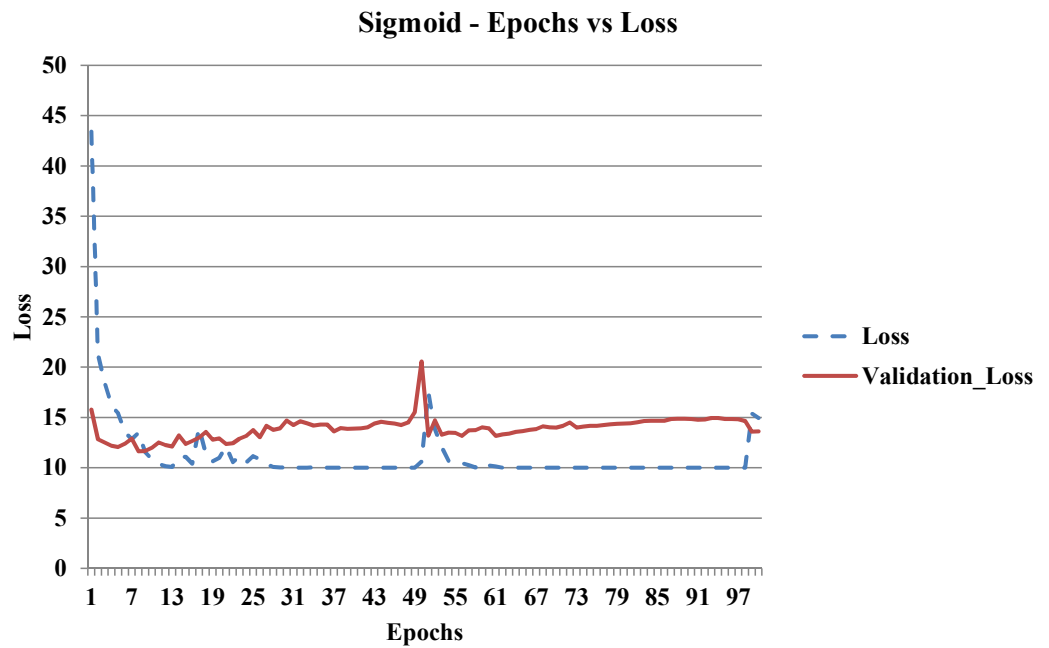
where $S(x)$ is the sigmoid function and e is Euler’s number.

The sigmoid function exists between 0 and 1. Therefore, it is especially used for models where the output prediction falls under ranges of probabilities. Figure 9a,b shows the number of epochs vs. accuracy (%) and loss graph for 100 epochs of the Adam optimizer using the Sigmoid activation function. From the graph, high accuracy and minimum loss were achieved in a fewer number of epochs, and accuracy and loss were saturated.



(a) Number of Epochs vs. Accuracy (%)

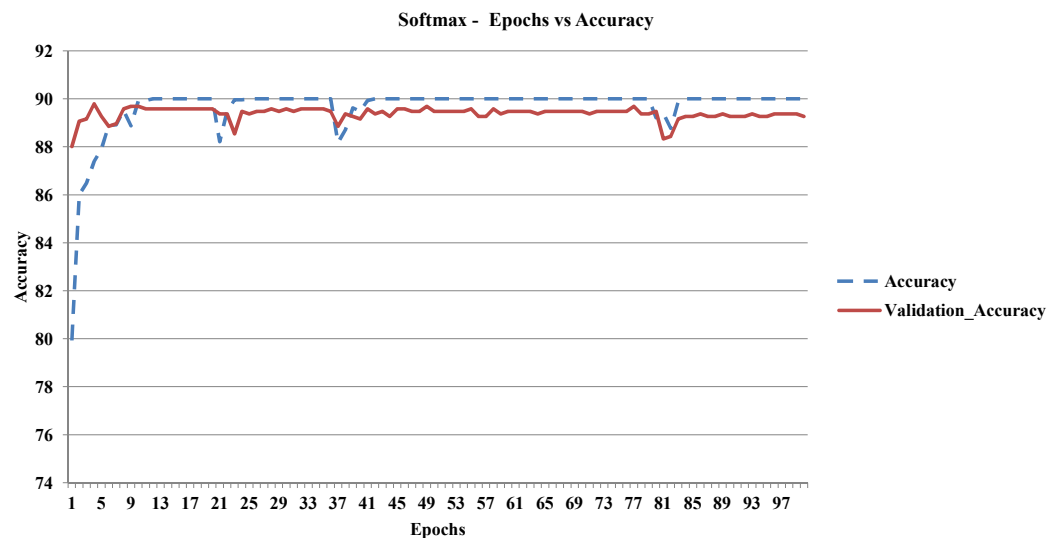
Figure 9. Cont.



(b) Number of Epochs vs. Loss (%)

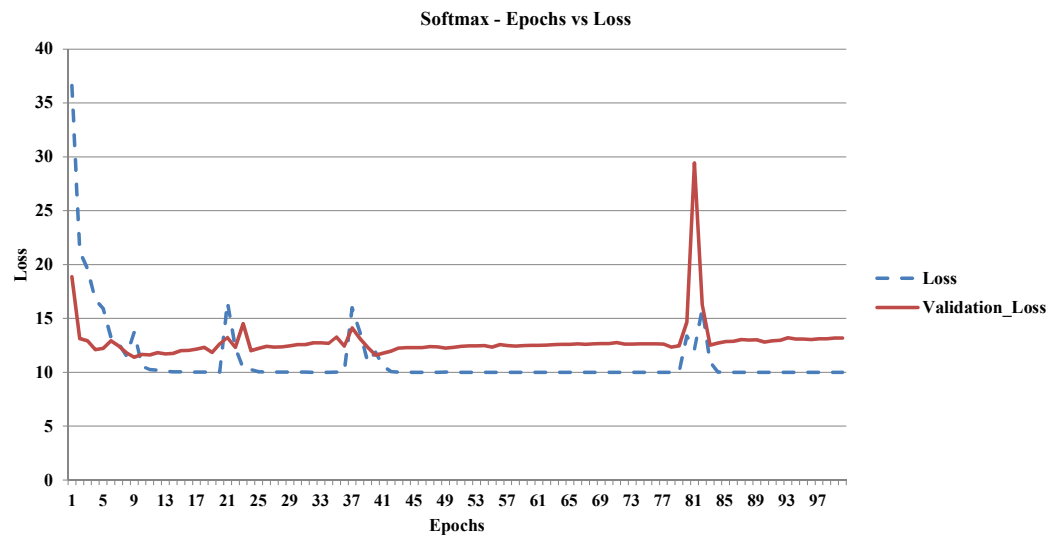
Figure 9. Accuracy and loss graph on training and validation data of Adam optimizer with Sigmoid activation function.

Softmax is used as the activation function for multi-class classification problems where class membership is required for more than two class labels. Although it can be used as an activation function in the hidden layers of networks, it is less commonly used. Here, the Softmax layer was used at the end of our neural network model. Figure 10a,b shows the number of epochs vs. accuracy (%) and loss graph for 100 epochs of the Adam optimizer using the Softmax activation function.



(a) Number of Epochs vs. Accuracy (%)

Figure 10. Cont.



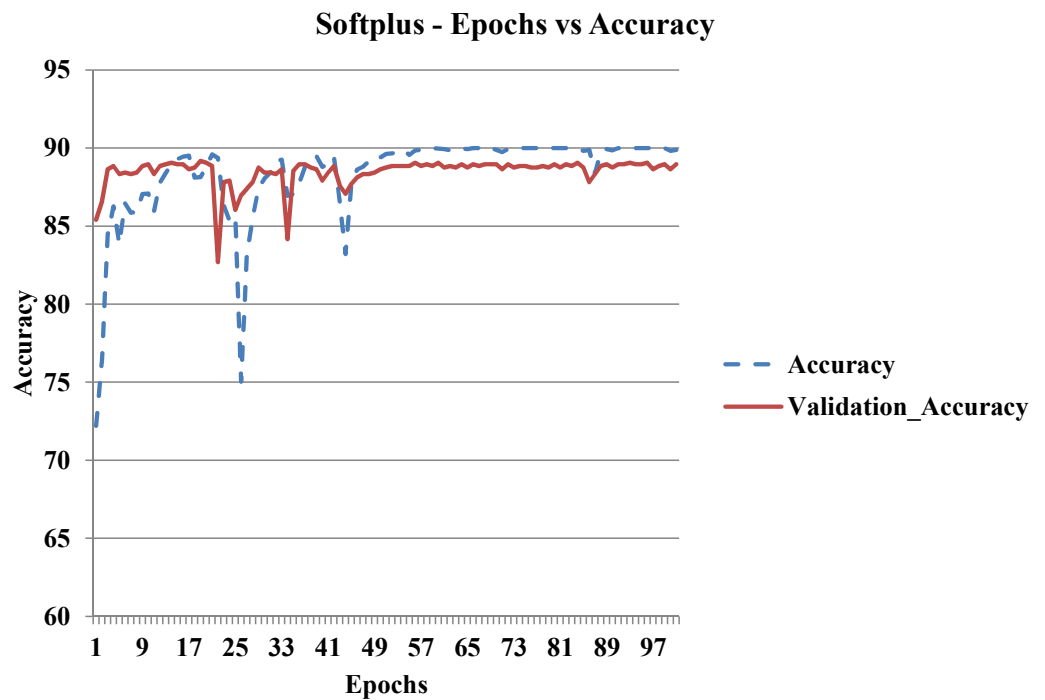
(b) Number of Epochs vs. Loss (%)

Figure 10. Accuracy and loss graph on training and validation data of Adam optimizer with Softmax activation function.

The Softplus function is mathematically expressed as shown in Equation (4),

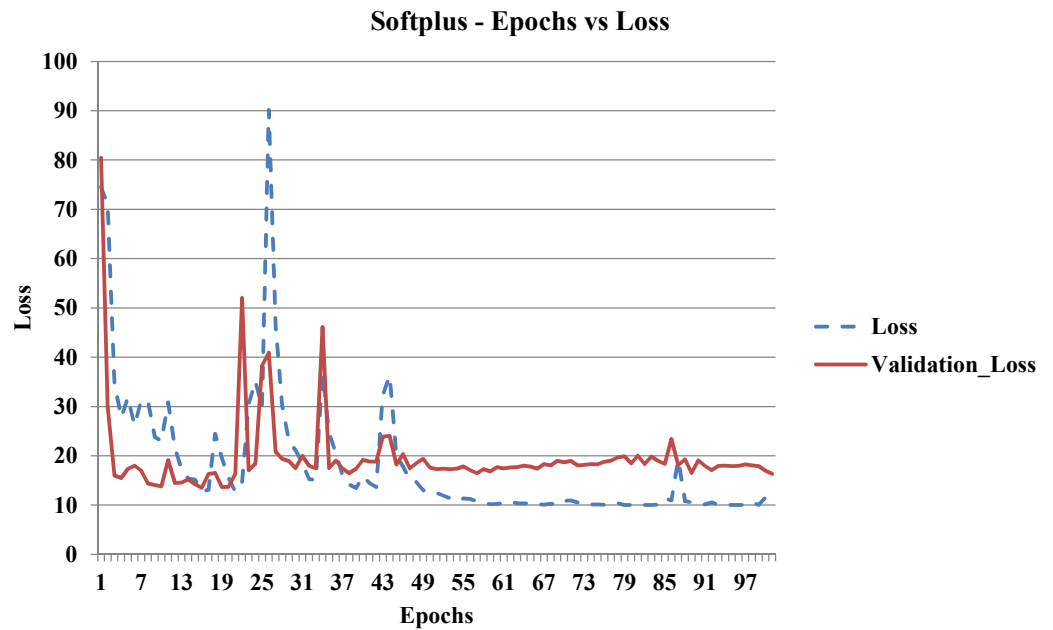
$$f(x) = \ln(1 + e^x) \tag{4}$$

It is an alternative to traditional activation functions and is less commonly used. Figure 11 shows the number of epochs vs. accuracy (%) and loss graph for 100 epochs of the Adam optimizer using the Softplus activation function.



(a) Number of Epochs vs. Accuracy (%)

Figure 11. Cont.



(b) Number of Epochs vs. Loss (%)

Figure 11. Accuracy and loss graph on training and validation data of Adam optimizer with Softplus activation function.

The average accuracy and loss were calculated and tabulated below for the previously discussed activation functions. Table 2 shows the accuracy, validation accuracy, loss, and validation loss for the various activation layers used.

Table 2. Comparison chart for various activation layers of Adam optimizer.

Activation Function	Accuracy (%)	Validation Accuracy (%)	Loss (%)	Validation Loss (%)
Sigmoid	89.6257	89.2181	11.2117	13.8817
Softmax	89.6462	89.3862	11.0794	12.8375
Softplus	88.3582	88.4686	18.0761	19.6317

As shown in Table 2, Softmax produced better accuracy than the other activation layers. It is also the most commonly used activation function at the end of neural networks. The Adam optimizer and Softmax activation function were used for further analysis.

3.3. Learning Rate Analysis

The learning rate or step size is the number of weights that are updated during the training phase. Usually, it lies in a range of 0 to 1. This hyperparameter controls how quickly the model learns a problem. A smaller learning rate requires more epochs, whereas a larger learning rate requires fewer epochs. A larger learning rate may cause the model to converge too quickly, whereas a very small learning rate may also have negative effects [40]. Therefore, learning rate is the most important hyperparameter and should be carefully selected. Table 3 shows the comparative analysis on the performance of the model considering different learning rates of Adam optimizer and Softmax activation.

Table 3. Comparison with different learning rates of Adam optimizer and Softmax activation layer.

Learning Rate	Accuracy (%)	Validation Accuracy (%)	Loss (%)	Validation Loss (%)
0.01	88.3164	88.7505	14.6927	14.9022
0.001	89.6462	89.3862	11.0794	12.8375
0.0001	89.5501	89.3503	11.4771	11.9575

The learning rates of 0.01, 0.001 (default), and 0.0001 were used for analysis. Table 4 shows the accuracy, validation accuracy, loss, and validation loss for the various learning rates of the model, along with the precision, recall, and F1-score statistics observed with the choice of the Adam optimizer and Softmax activation layer for the network. Figure 12 shows the confusion matrix representing the comparison among the predicted fire, non-fire, and smoke images in the dataset.

Table 4. Comparison with different learning rates of Adam optimizer and Softmax activation layer.

Learning Rate	Accuracy (%)	Validation Accuracy (%)	Loss (%)	Validation Loss (%)	Precision	Recall	F1-Score
0.01	88.3164	88.7505	14.6927	14.9022	0.89	0.88	0.87
0.001	89.6462	89.3862	11.0794	12.8375	0.91	0.90	0.85
0.0001	89.5501	89.3503	11.4771	11.9575	0.90	0.89	0.86

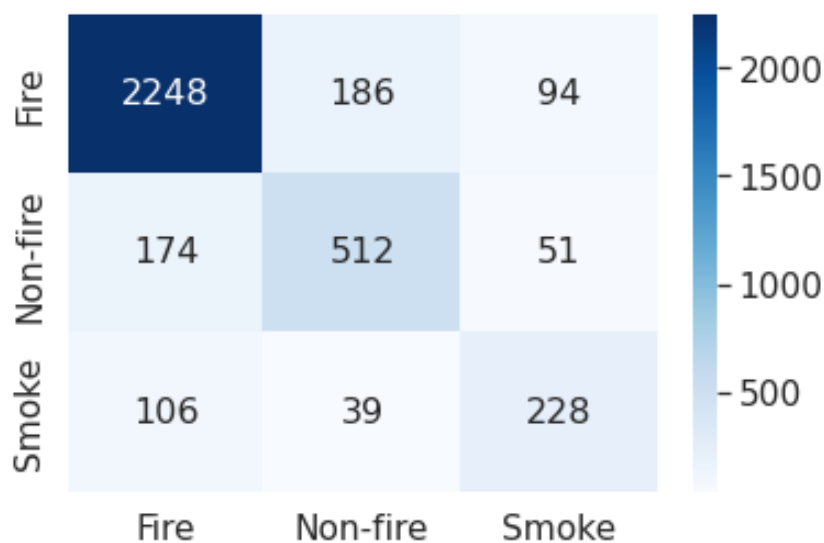


Figure 12. Confusion matrix representation for the CNN model with Adam optimizer and Softmax Activation layer.

As shown in Table 4, the learning rate of 0.001 produced better results than the others. Usually, a lower learning rate produces better results, but needs a large number of epochs. The best learning rate is based on the number of epochs used for training. In our case (for 100 epochs), the default learning rate of 0.001 for Adam produced the highest accuracy. Generally, it is obvious that a large dataset could have significant improvements in classification accuracy with reduced error. However, with the dataset chosen, with a small set of data on fire, non-fire, and smoke images, it was evident from the results that high error rates were observed. However, if the model involved larger standard datasets on fire and non-fire images, it could drastically enhance the classification accuracy and thereby reduce the higher error rates.

4. Conclusions and Future Work

Deep learning frameworks have been under the spotlight in recent years because of their flexibility and suitability for classification tasks. DL frameworks learn useful representations of features directly from the input image, text, audio, or video data using neural networks. In this article, fire detection was carried out using the proposed CNN architecture, which consisted of six layers. We experimented and analyzed the neural network using various optimizers, activation functions, and learning rates. From our results, the best optimizer, activation function, and learning rate for this neural network were observed, based on accuracy and loss metrics generated for both training and validation. Further, the robotic system was configured for navigation in autonomous and manual mode toward the fire zone and for actuation of the extinguishing unit based on the learned features triggered through the developed model. It was evident from the experimental results that the Adam optimizer with Softmax activation, driven with a learning rate of 0.001, provided enhanced accuracy compared with the other parameters tested. In addition, the chosen dataset investigated with the aforementioned configuration provided a better accuracy of 89.64% compared with the other techniques. Even though the system imposes limitations and concerns regarding its applicability in real-world scenarios due to its high error rates, consideration of larger datasets will enhance the accuracy of the system.

In future studies, we plan to devise a robust DL architecture for combined fire and human detection, resulting in a single model that can detect both humans and fire for saving the lives of humans while also extinguishing fires.

Author Contributions: Conceptualization, methodology, original draft preparation, and experiments: S.K.J.; data processing, experiments and analysis: K.M.; supervision, funding acquisition: K.M.; analysis, reviewing, and editing: A.K.J.S.; analysis, reviewing, and editing: J.J.P.C.R. All authors have read and agreed to the published version of the manuscript.

Funding: The authors extend their appreciation to the Deanship of Scientific Research at Imam Mohammad Ibn Saud Islamic University (IMSIU) for funding and supporting this work through the Research Partnership Program No. RP-21-07-02.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: This work was supported by the Deanship of Scientific Research at Imam Mohammad Ibn Saud Islamic University (IMSIU) through the Research Partnership Program No. RP-21-07-02.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Tomar, J.S.; Kranjčić, N.; Đurin, B.; Kanga, S.; Singh, S.K. Forest fire hazards vulnerability and risk assessment in Sirmaur district forest of Himachal Pradesh (India): A geospatial approach. *ISPRS Int. J. Geo-Inf.* **2021**, *10*, 447. [\[CrossRef\]](#)
2. de Dios, V.R.; Camprubí, À.C.; Pérez-Zanón, N.; Peña, J.C.; Del Castillo, E.M.; Rodrigues, M.; Yao, Y.; Yebra, M.; Vega-García, C.; Boer, M.M. Convergence in critical fuel moisture and fire weather thresholds associated with fire activity in the pyroregions of Mediterranean Europe. *Sci. Total Environ.* **2022**, *806*, 151462. [\[CrossRef\]](#) [\[PubMed\]](#)
3. Nielsen, M.B.; Bojsen-Hansen, M.; Stamatelos, K.; Bridson, R. Physics-Based Combustion Simulation. *ACM Trans. Graph.* **2022**, *41*, 1–21. [\[CrossRef\]](#)
4. Li, Z.; Zhu, H.; Zhao, J.; Zhang, Y.; Hu, L. Experimental research on the effectiveness of different types of foam of extinguishing methanol/diesel pool fires. *Combust. Sci. Technol.* **2022**, 1–19. [\[CrossRef\]](#)
5. Lin, X.; Song, S.; Zhai, H.; Yuan, P.; Chen, M. Using catastrophe theory to analyze subway fire accidents. *Int. J. Syst. Assur. Eng. Manag.* **2020**, *11*, 223–235. [\[CrossRef\]](#)
6. Majid, S.; Alenezi, F.; Masood, S.; Ahmad, M.; Gündüz, E.S.; Polat, K. Attention based CNN model for fire detection and localization in real-world images. *Expert Syst. Appl.* **2022**, *189*, 116114. [\[CrossRef\]](#)
7. Allaire, F.; Mallet, V.; Filippi, J.B. Emulation of wildland fire spread simulation using deep learning. *Neural Netw.* **2021**, *141*, 184–198. [\[CrossRef\]](#)

8. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [[CrossRef](#)]
9. Bulatov, K.; Chukalina, M.; Buzmakov, A.; Nikolaev, D.; Arlazarov, V.V. Monitored reconstruction: Computed tomography as an anytime algorithm. *IEEE Access* **2020**, *8*, 110759–110774. [[CrossRef](#)]
10. Arlazarov, V.; Arlazarov, V.; Bulatov, K.; Chernov, T.; Nikolaev, D.; Polevoy, D.; Sheshkus, A.; Skoryukina, N.; Slavin, O.; Usilin, S. Mobile ID Document Recognition—Coarse-to-Fine Approach. *Pattern Recognit. Image Anal.* **2022**, *32*, 89–108. [[CrossRef](#)]
11. Lee, Y.; Shim, J. False positive decremented research for fire and smoke detection in surveillance camera using spatial and temporal features based on deep learning. *Electronics* **2019**, *8*, 1167. [[CrossRef](#)]
12. Kim, B.; Lee, J. A video-based fire detection using deep learning models. *Appl. Sci.* **2019**, *9*, 2862. [[CrossRef](#)]
13. Govil, K.; Welch, M.L.; Ball, J.T.; Pennypacker, C.R. Preliminary results from a wildfire detection system using deep learning on remote camera images. *Remote Sens.* **2020**, *12*, 166. [[CrossRef](#)]
14. Ali, L.; Alnajjar, F.; Jassmi, H.A.; Gocho, M.; Khan, W.; Serhani, M.A. Performance evaluation of deep CNN-based crack detection and localization techniques for concrete structures. *Sensors* **2021**, *21*, 1688. [[CrossRef](#)] [[PubMed](#)]
15. Navaneeth, B.; Suchetha, M. PSO optimized 1-D CNN-SVM architecture for real-time detection and classification applications. *Comput. Biol. Med.* **2019**, *108*, 85–92. [[CrossRef](#)]
16. Salameh, H.B.; Dhainat, M.; Benkhalifa, E. A survey on wireless sensor network-based IoT designs for gas leakage detection and fire-fighting applications. *Jordanian J. Comput. Inf. Technol.* **2019**, *5*, 60–73. [[CrossRef](#)]
17. Zheng, R.; Lu, S.; Shi, Z.; Li, C.; Jia, H.; Wang, S. Research on the aerosol identification method for the fire smoke detection in aircraft cargo compartment. *Fire Saf. J.* **2022**, *130*, 103574. [[CrossRef](#)]
18. Prasad, B.; Manjunatha, R. Internet of Things Based Monitoring System for Oil Tanks. In Proceedings of the 2021 IEEE International Conference on Mobile Networks and Wireless Communications (ICMNBC), Tumkur, India, 3–4 December 2021; pp. 1–7.
19. Liu, X.; Zhang, C. Stability and Optimal Control of Tree-Insect Model under Forest Fire Disturbance. *Mathematics* **2022**, *10*, 2563. [[CrossRef](#)]
20. Pereira, J.; Mendes, J.; Júnior, J.S.; Viegas, C.; Paulo, J.R. A Review of Genetic Algorithm Approaches for Wildfire Spread Prediction Calibration. *Mathematics* **2022**, *10*, 300. [[CrossRef](#)]
21. Zhou, Y.C.; Hu, Z.Z.; Yan, K.X.; Lin, J.R. Deep learning-based instance segmentation for indoor fire load recognition. *IEEE Access* **2021**, *9*, 148771–148782. [[CrossRef](#)]
22. Saied, A.; Ahmed Alef, H.S.A.S. Outdoor-Fire Images and Non-Fire Images for Computer Vision Tasks (Version 1) 2018. Available online: <https://www.kaggle.com/datasets/phyllake1337/fire-dataset> (accessed on 11 October 2022).
23. Scheuerman, M.K.; Hanna, A.; Denton, E. Do datasets have politics? Disciplinary values in computer vision dataset development. *Proc. ACM Hum.-Comput. Interact.* **2021**, *5*, 1–37. [[CrossRef](#)]
24. Ghali, R.; Akhloufi, M.A.; Jmal, M.; Mseddi, W.S.; Attia, R. Forest fires segmentation using deep convolutional neural networks. In Proceedings of the 2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Melbourne, Australia, 17–20 October 2021; pp. 2109–2114.
25. Liu, W.; Bao, Q.; Sun, Y.; Mei, T. Recent advances of monocular 2d and 3d human pose estimation: a deep learning perspective. *ACM Comput. Surv.* **2022**, *55*, 1–41. [[CrossRef](#)]
26. Chernyshova, Y.S.; Sheshkus, A.V.; Arlazarov, V.V. Two-step CNN framework for text line recognition in camera-captured images. *IEEE Access* **2020**, *8*, 32587–32600. [[CrossRef](#)]
27. Muhammad, K.; Ahmad, J.; Lv, Z.; Bellavista, P.; Yang, P.; Baik, S.W. Efficient deep CNN-based fire detection and localization in video surveillance applications. *IEEE Trans. Syst. Man Cybern. Syst.* **2018**, *49*, 1419–1434. [[CrossRef](#)]
28. Khalil, A.; Rahman, S.U.; Alam, F.; Ahmad, I.; Khalil, I. Fire Detection Using Multi Color Space and Background Modeling. *Fire Technol.* **2021**, *57*, 1221–1239. [[CrossRef](#)]
29. Sun, Y.; Zhang, W.; Gu, H.; Liu, C.; Hong, S.; Xu, W.; Yang, J.; Gui, G. Convolutional neural network based models for improving super-resolution imaging. *IEEE Access* **2019**, *7*, 43042–43051. [[CrossRef](#)]
30. Muhammad, K.; Khan, S.; Elhoseny, M.; Ahmed, S.H.; Baik, S.W. Efficient fire detection for uncertain surveillance environment. *IEEE Trans. Ind. Inform.* **2019**, *15*, 3113–3122. [[CrossRef](#)]
31. Smolin, A.; Yamaev, A.; Ingacheva, A.; Shevtsova, T.; Polevoy, D.; Chukalina, M.; Nikolaev, D.; Arlazarov, V. Reprojection-Based Numerical Measure of Robustness for CT Reconstruction Neural Network Algorithms. *Mathematics* **2022**, *10*, 4210. [[CrossRef](#)]
32. Fang, Z.; Wang, J.; Du, J.; Hou, X.; Ren, Y.; Han, Z. Stochastic optimization-aided energy-efficient information collection in Internet of underwater things networks. *IEEE Internet Things J.* **2021**, *9*, 1775–1789. [[CrossRef](#)]
33. Llugsi, R.; El Yacoubi, S.; Fontaine, A.; Lupera, P. Comparison between Adam, AdaMax and Adam W optimizers to implement a Weather Forecast based on Neural Networks for the Andean city of Quito. In Proceedings of the 2021 IEEE Fifth Ecuador Technical Chapters Meeting (ETCM), Cuenca, Ecuador, 12–15 October 2021; pp. 1–6.
34. Syed, D.; Abu-Rub, H.; Ghayeb, A.; Refaat, S.S.; Houchati, M.; Bouhali, O.; Bañales, S. Deep learning-based short-term load forecasting approach in smart grid with clustering and consumption pattern recognition. *IEEE Access* **2021**, *9*, 54992–55008. [[CrossRef](#)]
35. Fradi, M.; Zahzah, E.h.; Machhout, M. Real-time application based CNN architecture for automatic USCT bone image segmentation. *Biomed. Signal Process. Control.* **2022**, *71*, 103123. [[CrossRef](#)]

36. Halgamuge, M.N.; Daminda, E.; Nirmalathas, A. Best optimizer selection for predicting bushfire occurrences using deep learning. *Nat. Hazards* **2020**, *103*, 845–860. [[CrossRef](#)]
37. Sevilla, J.; Heim, L.; Ho, A.; Besiroglu, T.; Hobbhahn, M.; Villalobos, P. Compute trends across three eras of machine learning. *arXiv* **2022**, arXiv:2202.05924.
38. Liu, J.; Sun, Y.; Gan, W.; Xu, X.; Wohlberg, B.; Kamilov, U.S. Sgd-net: Efficient model-based deep learning with theoretical guarantees. *IEEE Trans. Comput. Imaging* **2021**, *7*, 598–610. [[CrossRef](#)]
39. Mei, S.; Chen, X.; Zhang, Y.; Li, J.; Plaza, A. Accelerating convolutional neural network-based hyperspectral image classification by step activation quantization. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 1–12. [[CrossRef](#)]
40. Shu, W.; Cai, K.; Xiong, N.N. A short-term traffic flow prediction model based on an improved gate recurrent unit neural network. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 16654–16665. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.