2024 IEEE 9th International Conference for Convergence in Technology (I2CT)
Pune, India. Apr 5-7, 2024

A Comparative Assessment of CNN-Sigmoid and CNN-SVM model for Forest Fire Detection

Samta Gaur[1],J. Sathish Kumar[2] Shailendra Shukla [3]
Computer Science and Engineering Department
Motilal Nehru National Insttitute of Technology Allahabad, Prayagraj, India
[1]samta.2022p14@mnnit.ac.in, [2]sathish613@mnnit.ac.in and [3] ss@mnnit.ac.in

2024 IEEE 9th International Conference for Convergence in Technology (I2CT) | 979-8-3503-9447-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/I2CT61223.2024.10543925

***Abstract*—Forest fires present a notable danger to the environment, wildlife, and human lives. Early detection of forest fire is crucial to prevent their spread and minimize damage. The timely detection and response to forest fires are paramount to mitigating their destructive impact. Yet, conventional methods like visual inspection and ground patrols are labor-intensive, time-consuming, and demand extensive coverage. Recently, machine learning algorithms for image detection have emerged as an effective approach for detecting forest fire using forest fire images taken from drone. This study aims to evaluate the effectiveness of Convolution Neural Network with Sigmoid (CNN-Sigmoid) and Convolution Neural Network with Support Vector Machine (CNN-SVM) model to detect forest fire (on Forest Fire Dataset). Through our experimental result we have found that the extraction using CNN and classification using SVM has improved the training accuracy (99.87%) and validation accuracy (97.11%) of CNN-SVM model compared to CNN-Sigmoid. This observation was substantiated by the CNN-SVM achieving a test accuracy of 96.84%, while the CNN-Sigmoid attained a test accuracy of 95.79%.**

***Index Terms*—Forest Fire Detection, Convolution Neural Network (CNN), Support Vector Machine (SVM), CNN-SVM, CNN-Sigmoid, Forest Fire Dataset.**

I. Introduction

Forests are essential natural resources, offering diverse benefits such as shielding against sandstorms, ecological stability, and water conservation. Forests are often referred to as the ”earth’s lungs,” given their role in purifying the air, generating oxygen, and mitigating carbon dioxide levels [1]. Besides, forests provide habitats and protect crops. Yet, increasing forest fires due to dry conditions and human activities endanger these ecosystems. Unlike common fires, wildfires disrupt the natural balance, spurred by factors like rising temperatures and lightning [2]. The Uttar Pradesh State Disaster Management Authority (UPSDMA) reports that every year, rural regions of the state witness the destruction of numerous homes due to fires, particularly during the summer season. Urban areas are not exempt from fire incidents either [3]. Shockingly, over 36% of India’s forest cover is deemed prone to frequent forest fires, with nearly 4% being classified as extremely prone and 6% as highly fire-prone [4]. This recurring devastation results in the loss of valuable forest resources, including stored carbon in biomass. Such fires pose substantial threats to the environment, human lives, and property.

Consequently, forest fire prevention has gained prominence on governmental agendas, leading to increased efforts in developing and implementing automated monitoring and fire detection systems to safeguard these vital ecosystems. The timely detection and response to forest fires are paramount to mitigating their destructive impact. However, traditional approaches such as visual examination and ground patrols require a significant amount of labor, consume a lot of time, and necessitate comprehensive coverage.

Therefore, researchers have delved into the use of machine learning techniques to automate the forest fire detection process. They have explored employing machine learning algorithms to autonomously detect forest fires from images. These algorithms possess the capability to scrutinize vast image datasets, identifying patterns indicative of a forest fire.

Hence, the focus of this paper is the study of machine learning algorithms to enhance the accuracy and efficiency of forest fire detection. This approach facilitates early fire detection and swift response, ultimately leading to superior environmental and human life protection. The key contributions of this paper are as follows:

- An evaluation and comparison of CNN-Sigmoid and CNN-SVM for forest fire detection.
- Utilization of forest fire images to train and assess these models, gauging their performance based on metrics like accuracy and validation accuracy.
- Exploration of Support Vector Machines (SVMs) as a classification layer within the CNN-Sigmoid model to enhance prediction accuracy.

The subsequent sections of the paper are structured in the following manner. The II section delves into the analysis of relevant studies, while the subsequent section III outlines our proposed approach. Following that, the IV section examines and discusses our study’s findings. The final section, designated as the V section, offers the concluding remarks of the work.

II. Related Work

This section explains some of the prior research in this area that is relevant to the proposed work. Khan et al. [1] introduced a dataset with fire and non-fire images, demonstrating improved forest fire detection through a novel CNN training approach using transfer learning. This highlights transfer learning's efficacy for machine learning-based fire detection, particularly valuable to researchers in the field. Bouguettaya et al. [5] surveyed the integration of Unmanned Aerial Vehicles (UAVs) and deep learning-based computer vision in wildfire detection. Their study covered UAV types, sensors, and algorithm outlines, emphasizing the potential to enhance detection accuracy and speed for rapid wildfire prevention and mitigation responses. Kinaneva et al. [6] advocated for using drones and AI to achieve swift and precise forest fire detection, underscoring the preemptive potential of this approach for disaster impact minimization. They critically assessed traditional methods and addressed challenges like refining AI algorithms and integrating drones into existing systems. Allauddin et al. [7] proposed a drone-based system for real-time monitoring and accurate detection of forest fires. The paper assessed current detection methods and highlighted challenges such as optimizing flight paths, image processing, and communication systems for immediate data transfer. Agarwal et al. [8] presented experimental results utilizing UAV-captured thermal images, achieving high accuracy and low false alarms. Employing classifiers and transfer learning, their approach demonstrated superior accuracy and speed in forest fire detection compared to traditional methods. Abdusalomov et al. [9] introduced a method using the Detectron2 model, attaining high accuracy and low false alarms in detecting forest fires from aerial images. Their approach surpassed traditional methods with enhanced accuracy and efficiency. Avazov et al. [10] developed an AI and IoT-based method achieving 98% accuracy in forest fire detection using thermal images. They also devised a real-time IoT alert system for authorities to be promptly notified of detected fires. Dampage et al. [11] introduced a system combining wireless sensor networks and machine learning for enhanced forest fire detection. Their real-time approach analyzing sensor data achieved a notable accuracy of 95.12% in identifying forest fires, showcasing its potential for accuracy and speed improvement.

III. Proposed Work

This section discusses the methodology and datasets utilized in this work. The Forest Fire dataset is used to train with different machine-learning algorithms. We also added testing images to check the result of each algorithm used.

A. Dataset

The dataset utilized for our project primarily originates from the Forest Fire dataset. This collection encompasses images sourced from various online platforms, obtained by employing search criteria like forest fire, wildfire on mountains, forest, and green hills. The dataset has been divided into two distinct classes: fire and no-fire [12]. The category denoted as ”fire class” encompasses imagery depicting forests and hills with observable flames or flames accompanied by clouds of smoke. Conversely, the ”no-fire class” is constituted of visuals featuring forests and mountains of greenery, taken from a range of perspectives [1].

For a binary classification goal, we categorize Forest Fire dataset images into fire and no-fire classes. The dataset includes diverse landscape images, aiding the model in discerning fire presence. The approach uses 1898 images, split 80:20 for training data and data for testing. The training set has 749 fire and 749 fire-free forest images, while the testing set has 190 images for each class.

TABLE I
DATASET SPLITING

Dataset	Training	Testing	Total
Fire	759	190	949
No-Fire	759	190	949
Total	1518	380	1898

B. Normalization

Normalization is a crucial data preprocessing method that involves adjusting data values to a particular range or distribution. This technique is vital for enhancing the performance and convergence of various machine learning algorithms.

Normalization ensures data consistency and comparability across different features or samples. It prevents features with larger value ranges from disproportionately influencing the model and overshadowing the significance of features with narrower value ranges.

C. Data Augmentation

Data augmentation is a technique of expanding a training dataset by applying alterations to existing data samples. Shearing, zooming, and horizontal flipping are popular data augmentation techniques. Shearing alters an image's form, zooming alters its scale, and horizontal flip turns it horizontally. We may increase the model's robustness by using these changes to generate fresh, comparable training samples but not identical to the original data.

The method used in this study involves loading the images in batches of 32. The image paths for the dataset are then retrieved and passed through a label encoder. Following this, the images are resized to 150 x 150 pixels and preprocessed before being collected into batches and passed through the feature extractor.

D. Methods used in detecting Forest Fire

Detecting forest fires poses inherent challenges due to expansive coverage, rugged terrains hindering sensor placement, and unpredictable weather conditions in remote locations. These factors significantly complicate the creation of automated algorithms for early fire detection. Moreover, the efficacy of machine learning algorithms hinges on abundant data for achieving heightened detection precision.

In response, various solutions based on machine learning algorithms have been developed for classifying forest fires. To enhance classification accuracy further, we advocate for a hybrid model solution, combining Convolutional Neural Networks with Support Vector Machines (CNN-SVM). This integration aims to establish an effective forest fire detection system that capitalizes on the strengths of both techniques.

1) Convolutional Neural Network with Sigmoid (CNN-Sigmoid): Convolutional neural network model (CNN) is highly sophisticated machine learning algorithm that are primarily utilized for image processing and classification tasks. They are capable of identifying patterns and extracting meaningful features from complex datasets such as images, and are made up of various layers, including convolutional layers, pooling layers, and dense layers as shown in the Fig. 1. The input layer takes the image data, and the convolutional layer extracts essential features using a set of filters. Activation functions, like ReLU, introduce non-linearity, and max pooling reduces spatial dimensions while retaining key features. The flattened output is then passed through a fully connected layer, where more complex relationships are learned. Subsequently, another activation function, like ReLU, introduces non-linearity again. Finally, the output layer with two neurons and the Sigmoid activation assigns probabilities to each class, enabling the model to make binary predictions (Fire, No Fire). Regularization techniques such as dropout or batch normalization can be incorporated to enhance the network's performance and prevent overfitting. To train a CNN model, labeled data is used to adjust the weights and biases of the filters in the convolutional layers through backpropagation, which enables the CNN to learn how to recognize patterns in images and make precise predictions.

After that the classification is performed by minimizing the loss using the Adam optimizer, with a decision threshold of 0. Images with a prediction result less than or equal to 0 are predicted to contain a fire, while images with a prediction result greater than 0 are predicted to not contain a fire.

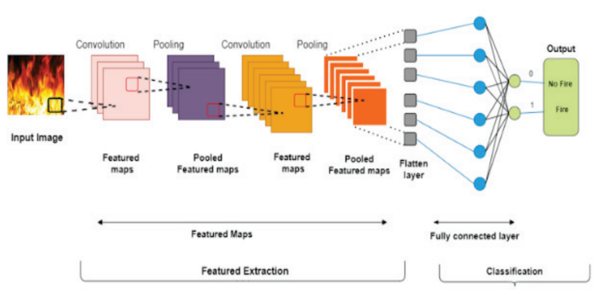

Fig. 1. A typical Convolutional Neural Network

To enhance the accuracy of the plain CNN model, SVM is employed in the final classification layer.

2) Support Vector Machine (SVM): SVM is a popular type of classifier that can be used to separate data into different classes based on a boundary called a hyperplane. SVM work by mapping the input data to a high-dimensional feature space and determining the optimum hyperplane $f(w,x) = wx + b$, where "w" is the weight, "b" is the bias, that best separates the classess in a given dataset, with features $x \epsilon R^m$ [13].

SVM derives the parameter w by solving an optimization equation, as stated in Equation 1.

$$min \frac{1}{p} w^T w + C \sum_{i=1}^{p} max(0, 1 - y_i'(w^T x_i + b)) \quad (1)$$

In the given context, where w^T represents the Manhattan norm (L1 norm), C denotes the penalty parameter (chosen either arbitrarily or through hyper-parameter tuning), y' stands for the true label, and $w^T x + b$ represents the predictive function, Equation 1 takes the form of L1-SVM with the conventional hinge loss. Its differentiable alternative, L2-SVM (as expressed in Equation 2), yields a more consistent outcome [14].

$$min \frac{1}{p} \|w\|_2^2 + C \sum_{i=1}^{p} max(0, 1 - {y_i}'(w^T x_i + b))^2 \quad (2)$$

where, $\|w\|_2$ signifies the Euclidean norm (L2 norm), and the method involves using the squared hinge loss.

3) Proposed CNN-SVM model: The combination of CNN and SVM for image classification tasks results in a machine learning model known as a CNN-SVM model.

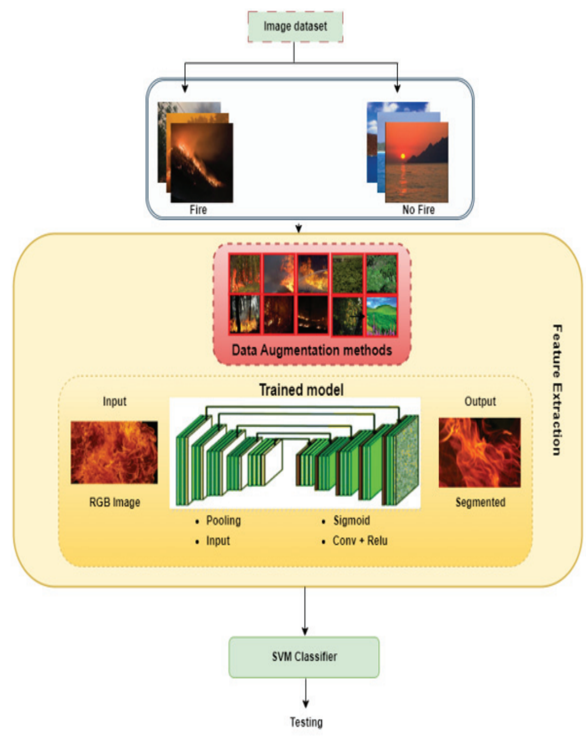

Fig. 2. Represents the architecture of CNN-SVM

The CNN-SVM model's structure is quite similar to that of a standard CNN-Sigmoid model, except for the last layer, which has been adjusted to perform SVM classification. The architecture of the CNN-SVM model is depicted in Fig 2. CNN-Sigmoid model is already explained above. For CNN-SVM model last layer need to be changed. The last layer includes a "kernel regularizer" parameter that uses the L2 norm and a linear activation function, similar to the final output layer. The L2 norm, also referred to as ridge regularization, is used to impose a penalty on the weights in the SVM classifier's kernel. This penalty term is added to the objective function, discouraging the presence of large weights in the model's parameters. By incorporating the L2 regularizer, the complexity of the model is effectively controlled. This regularization technique plays a crucial role in mitigating overfitting, which occurs when the model becomes excessively tailored to the training data and fails to generalize well to unseen data. And the primary objective of the linear SVM classifier is to perform binary classification by identifying the optimal hyperplane that separates the classes in the feature space. This algorithm assumes a linear decision boundary and actively searches for a straight line or plane that effectively distinguishes the classes. Establishing the linear decision boundary entails the maximization of the margin, representing the gap between the hyperplane and the closest data points from both classes. And lastly, we optimized the model using the Adam optimizer and set the loss function to "Hinge". The hinge loss function is a widely used component in SVM models for training the classifier. It evaluates the gap between the predicted scores and the true labels, penalizing errors in classification. Minimizing the hinge loss encourages the SVM model to achieve accurate classifications. Specifically, the loss is minimized when the predicted scores are significantly distanced from the decision boundary, promoting a broader separation margin between the different classes. The Flowchart of CNN-SVM model is shown in Fig 3.

For image classification the CNN-SVM model employs a decision threshold of 0 to determine whether an image is indicative of a forest fire. If the final prediction score is equal to or below 0, the image is classified as containing a forest fire, while a prediction score above 0 denotes that the image does not contain a forest fire.

IV. Experimental Analysis And Results

The proposed model is built with TensorFlow and the Keras framework, trained and tested on Google Colab. Subsequently, the effectiveness of the hybrid CNN-SVM model is appraised by evaluating the specified performance metric.

A. Evaluation Parameter

We have compared our proposed CNN-SVM model with CNN-Sigmoid model for the purpose of forest fire detection. For a more comprehensive analysis of the models, we delve into evaluating the predictions generated by both the CNN-Sigmoid model and the CNN-SVM model. The subsequent Table II outlines the quantities of false positives (F_p), referring non-fire images that were inaccurately classified as fire images.; true positives (T_p), denoting fire images correctly predicted as fire images; true negatives (T_n), representing the count of fire images accurately predicted as non-fire; and false negatives (F_n), which correspond to non-fire images erroneously predicted as non-fire by each model. Both the CNN-SVM and CNN-Sigmoid models adopted a threshold value of 0 for their predictions.

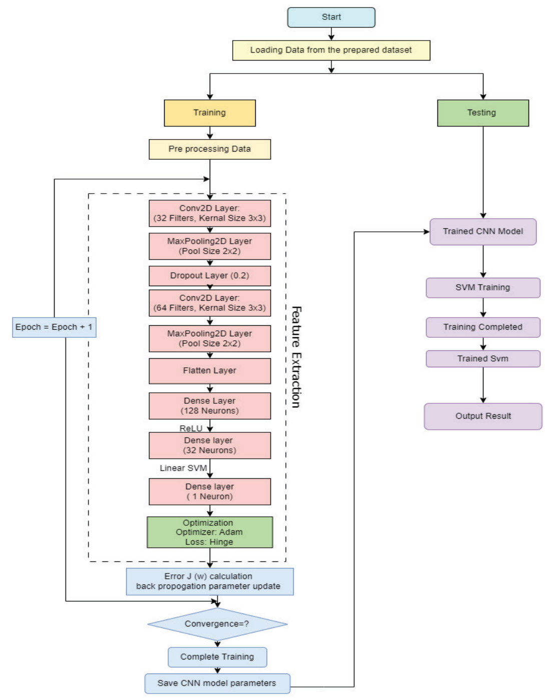

Fig. 3. Flowchart of CNN-SVM

TABLE II
Table of Predicted Values

Prediction Values	CNN-Sigmoid	CNN-SVM
True Positives	185	188
True Negatives	179	180
False Positives	5	2
False Negatives	11	10

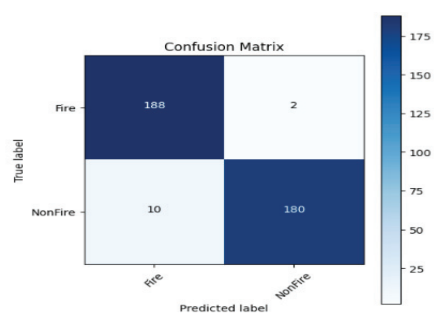

Fig. 4. Represents the confusion metrics of CNN-SVM

Upon examining the table II it is evident that the overall number of false predictions is lower in the CNN-SVM model when compared to the CNN-Sigmoid model.

For a thorough analysis, the performance of the CNN-Sigmoid model and CNN-SVM model is evaluated using the following metrics.

1) Accuracy: Accuracy quantifies the alignment between a model's predictions and the actual outcomes, represented as the ratio of accurately predicted instances to the total instances in a dataset. It's computed using the formula in Equation 3:

$$Accuracy = \frac{T_p + T_n}{T_p + T_n + F_p + F_n} \quad (3)$$

2) Misclassification Rate: Misclassification rate, quantifies the proportion of instances in a dataset that are incorrectly classified by the model. It's essentially the opposite of accuracy.

Mathematically, the misclassification rate is calculated as shown in Equation 4:

$$MisclassificationRate = \frac{F_p + F_n}{T_p + T_n + F_p + F_n} * 100 \quad (4)$$

A lower misclassification rate indicates that the model is making fewer mistakes and is performing better.

3) Precision: Precision evaluates the ratio of accurately predicted positive instances out of all instances predicted as positive by the model. Mathematically, precision is calculated as(Equation 5):

$$Precision = \frac{T_p}{(T_p + F_p)} \quad (5)$$

A high precision indicates that the model is accurately identifying positive cases and minimizing false alarms. However, high precision might come at the expense of missing some true positive cases, which is where recall comes into play as a complementary metric.

4) Recall: Recall focuses on the proportion of actual positive instances that were successfully predicted as positive by the model.

Mathematically, recall is calculated as shown in Equation 6:

$$Recall = \frac{T_p}{(T_p + F_n)} \quad (6)$$

It ensures that the model is capable of capturing as many actual positive cases as possible, even if it means a slightly higher rate of false positives.

5) F1 Score: The F1 Score serves as an assessment of a test's accuracy. In mathematical terms, it is derived as the harmonic mean of precision and recall, as demonstrated in Equation 7:

$$F1 - Score = \frac{2 * Precision * Recall}{Precision + Recall} \quad (7)$$

With values ranging from 0 to 1, the F1 Score provides an indication of the model's performance, with higher values representing better overall performance.

To obtain a more in-depth understanding of the results, we utilize equation 3 to calculate the accuracy values. Furthermore, we employ equation 4 to calculate the misclassification rates for both models.

The CNN-Sigmoid model achieves an test accuracy of 95.79%, while the CNN-SVM model exhibits an test accuracy of 96.84%.

The CNN-Sigmoid model has a misclassification rate of 4.21%, whereas the CNN-SVM model achieves a lower misclassification rate of 3.16%. These results indicate that the CNN-SVM model surpasses the performance of the CNN-Sigmoid model.

The CNN-Sigmoid and CNN-SVM model's other performance metrics, calculated using equations 5, 6, and 7, are as follows(shown in Table III):

TABLE III
TABLE OF EVALUATED PARAMETERS

Parameter	CNN-Sigmoid	CNN-SVM
Precision	0.9737	0.9895
Recall	0.9439	0.9495
F1-Score	0.9586	0.9691

B. Result Analysis

The CNN-Sigmoid model underwent training on the forest fire dataset for 20 epochs with 38 steps per epoch. The training accuracy of the model after training was found to be 0.9848, while the validation accuracy stood at 0.9658. Fig 5 illustrates the variation in training accuracy values and validation accuracy values concerning the number of epochs, while Fig 6 depicts the variation in validation loss and training loss values concerning the number of epochs.

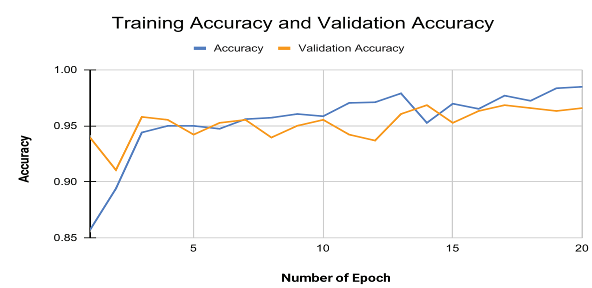

Fig. 5. Graph depicting the accuracy of the CNN-Sigmoid model over number of epochs

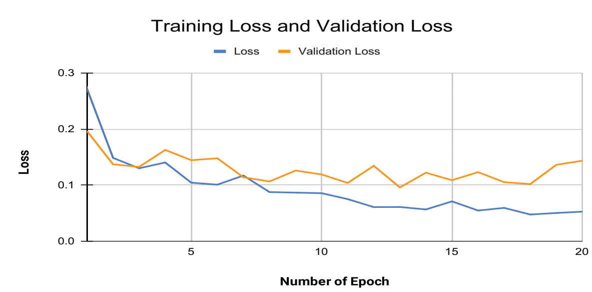

Fig. 6. Graph depicting the Loss of the CNN-Sigmoid model over number of epochs

To enhance the accuracy of the CNN-Sigmoid model, the CNN-Sigmoid model was modified by incorporating a SVM in the final classification layer. The CNN-SVM model was trained on the forest fire dataset for a total of 20 epochs, with 38 steps per epoch. Fig 7 illustrates the variation in training accuracy value and validation accuracy value concerning the number of epochs, while Fig 8 depicts the variation in validation loss value and training loss value concerning the number of epochs. The CNN-SVM model attained a training accuracy of 99.87% and a validation accuracy of 97.11%. Upon examining Fig 5 and Fig 7, it is evident that the training accuracy and validation accuracy of the CNN-SVM model outperforms that of the CNN-Sigmoid model.

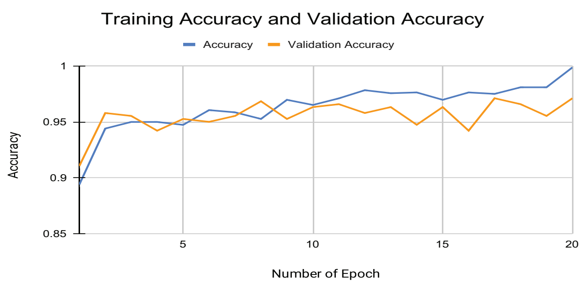

Fig. 7. Graph depicting the accuracy of the CNN-SVM model over number of epochs

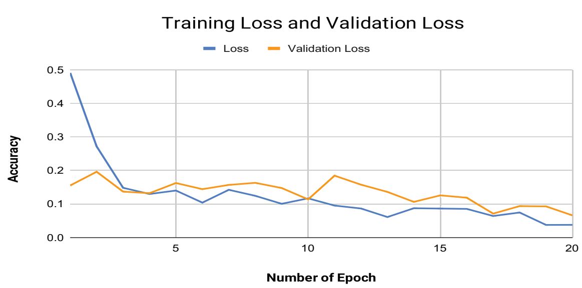

Fig. 8. Graph depicting the Loss of the CNN-SVM model over number of epochs

V. Conclusion

This study compares the effectiveness of the CNN-Sigmoid and CNN-SVM methods for forest fire detection using the Forest Fire dataset. The proposed CNN-SVM model demonstrates superior binary classification performance by optimizing decision boundaries effectively and enhancing class distinction using the SVM component. This outperforms the CNN-Sigmoid model, which employs a sigmoid activation function. The CNN-SVM model achieved higher training (99.87%) and test (96.84%) accuracies than the CNN-Sigmoid model (98.88% and 95.79% respectively). It also showed higher precision (98.95%), recall (94.95%), and F1-Score (96.91%) values. The CNN-SVM model's misclassification rate was 3.16%, while the CNN-Sigmoid model's rate was 4.21%. The results of the study indicated that the CNN-SVM had a lower misclassification rate as compared to the CNN-Sigmoid model, demonstrating that incorporating Support Vector Machine (SVM) in the final layer can enhance accuracy. The study concluded that the CNN model with SVM classifier was better suited to the forest fire dataset and had the potential to be extended to multi-classification for identifying different types of disasters.

Acknowledgments

The work presented here is performed as part of the research project entitled 'Internet of Things (IoT) and Machine Learning (ML) based Smart UAV-Assisted Emergency System for Disaster Management', which is supported and funded by the Council of Science and Technology (CST), Government of UP, India (Project and Sanction No. 1199 and CST/D/1219).

References

[1] Khan, A., Hassan, B., Khan, S., Ahmed, R., and Abuassba, A. DeepFire: A novel dataset and deep transfer learning benchmark for forest fire detection. Mobile Information Systems, 2022.

[2] Yuan, C., Liu, Z., Zhang, Y.: Vision-based forest fire detection in aerial images for firefighting using uavs. In proceedings of International Conference on Unmanned Aircraft Systems (ICUAS), IEEE, pp. 1200–1205, 2016.

[3] U.P.S.D.M.A.:https://upsdma.up.nic.in/stateprofile.html

[4] India, F.S.: https://fsi.nic.in/forest-fire-activities?pgid=forest-fire-activities

[5] Bouguettaya, A., Zarzour, H., Taberkit, A. M., and Kechida, A. A review on early wildfire detection from unmanned aerial vehicles using deep learning-based computer vision algorithms, Signal Processing,Vol 190, pp. 108309, 2022.

[6] Kinaneva, D., Hristov, G., Raychev, J., and Zahariev, P. (2019, May). Early forest fire detection using drones and artificial intelligence. In proceedings of 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO),IEEE ,pp. 1060-1065, 2019.

[7] Allauddin, M. S., Kiran, G. S., Kiran, G. R., Srinivas, G., Mouli, G. U. R., and Prasad, P. V. (2019, July). Development of a surveillance system for forest fire detection and monitoring using drones. In proceedings of IEEE International Geoscience and Remote Sensing Symposium (IGARSS), IEEE, pp. 9361-9363, 2019 .

[8] Agarwal, P., and Jha, G. Forest fire detection using classifiers and transfer learning. In proceedings of International Conference on Robotics, Automation and Artificial Intelligence (RAAI), IEEE ,pp. 29-33, 2021.

[9] Abdusalomov, A. B., Islam, B. M. S., Nasimov, R., Mukhiddinov, M., and Whangbo, T. K. . An improved forest fire detection method based on the detectron2 model and a deep learning approach, Sensors, Vol 23 No.3, pp. 1512, 2023.

[10] Avazov, K., Hyun, A. E., Sami S, A. A., Khaitov, A., Abdusalomov, A. B., and Cho, Y. I. Forest Fire Detection and Notification Method Based on AI and IoT Approaches, Future Internet, Vol. 15 No. 2, pp. 61, 2023.

[11] Dampage, U., Bandaranayake, L., Wanasinghe, R., Kottahachchi, K., and Jayasanka, B. Forest fire detection system using wireless sensor networks and machine learning, Scientific reports, Vol. 12 No. 1, pp. 46, 2022.

[12] Akyol, K.: A comprehensive comparison study of traditional classifiers and deep neural networks for forest fire detection, Cluster Computing, pp. 1–15, 2023.

[13] Agarap, A. F. An architecture combining convolutional neural network (CNN) and support vector machine (SVM) for image classification. arXiv preprint arXiv:1712.03541, 2017.

[14] Tang, Y. Deep learning using linear support vector machines. arXiv preprint arXiv:1306.0239, 2013.