Convolutional Neural Networks: Comparative Fire Detection Using Deep Learning Algorithm

Meju Moses
mosemej@gmail.com

Ekoko Ujerekre
ekokoujerekre@gmail.com

Chiadika Dominic Mario
anwumario@gmail.com

Abstract

Recent studies have showcased significant accomplishments in leveraging computer vision and deep learning techniques, significantly contributing to the realm of fire detection. Among these techniques, computer vision and AI-driven approaches, including static and dynamic texture analysis. A major safety problem for both buildings and public areas is fire detection. The reliability and coverage of traditional approaches depending on sensor networks are constrained. Convolutional Neural Networks, also known as CNNs, are a subset of artificial neural networks used in deep learning and are frequently employed for object and picture recognition and categorization. Thus, Deep Learning uses a CNN to identify items in a picture. There is a significant scarcity of publicly available datasets for fire detection in images, the factors that influenced our choices to use the Flame Vision dataset from Kaggle for this research work development are sizes, annotations, and balanced classes. The Flame Vision dataset consists of 8600 complete high resolution photographs. A purposeful and intentional divide has been made within this collection: 5000 photographs are devoted to exhibiting diverse fire scenarios, and the remaining 3600 images show scenes without any fires. This paper suggests that CNN and transfer learning models using VGG-16 offer the most promising results for the task of wildfire image classification. These models not only yield higher accuracy but also demonstrate a more balanced trade-off between precision and recall compared to traditional ANN.

Introduction

Statistics released by the Korean National Fire Agency revealed that 40,030 fires occurred across South Korea in 2019, resulting in 284 deaths and 2219 injuries (Cheong-mo, Y. 2020). Furthermore, an average of 110 fires and 0.8 fire-related fatalities occurred daily, leading to KRW 2.2 billion in fire property damages (Cheong- mo, Y. 2020). In 2020, more than 50 people were killed in a fire after a 33-story tower block building completely burned down in Ulsan, and a warehouse blaze occurred in Incheon, both incidents in major Korean cities (Seok-min, 2020).

Accidental fires can ignite with frightening unpredictability and spread uncontrollably in seconds. Early identification and prevention of the circumstances of these risks help to avoid unexpected fires and keep people safe. Recent studies have showcased significant accomplishments in leveraging computer vision and deep learning techniques, significantly contributing to the realm of fire detection. Among these techniques, computer vision and AI-driven approaches, including static and dynamic texture analysis as discussed by Opoulos, et al in 2015, convolutional neural networks (CNNs) as demonstrated by Alikhujaev, et al in 2020, and the utilization of 360 - degree sensors as explored by Barmpoutis, et al in 2020, have found widespread application in fire detection scenarios.

Meju, M., Ekoko, U. & Chiadika, D. M. (2026). Convolutional Neural Networks: Comparative Fire Detection Using Deep Learning Algorithm. Academy Journal of Information and Communication Technology (AJICT), 1(1), 1-15. https://ajict.mydspg.edu.ng

Aims, Objective and Scope

The research methodology employed in this paper can be summarized as follows:

1. Wildfire images acquisition using image dataset from (Flame Vision – Kaggle.com)

2. Wildfire images augmentation such as shearing, zooming and horizontal flipping to artificially grow the dataset. Also, image resizing from 512x512 to 64x64 pixels for computational efficiency and image normalization by rescaling the pixels form a scale of 0255 to a scale of 0-1 in order to speed-up the convergence while training.

3. Wildfire images training and validation using ANN, CNN and VGG-16, after which test images are either classified as fire or no-fire.

4. Performance analysis using the following statistical criteria namely accuracy, precision, recall and f1-score.

The scope of this research work is limited to simulation and modeling of deep learning algorithms for the detection and classification of wildfire images as fire and no-fire. The dataset used is the wildfire images are the Flame Vision dataset form Kaggle.com. The deep learning algorithms are ANN, CNN and VGG-16. This research work didn't cover the hardware that has to do with the design, construction and implementation of this research work into CCTV-based fire detection and alarm system.

Theoretical Background

According to numerous studies (Bixby et al. 2015; Ryu et al. 2018; McWethy et al. 2019; Tymstra et al. 2020), fire is a crucial ecological phenomenon that has complicatedly shaped landscapes, biodiversity, terrestrial ecosystems, and the atmosphere environment. A major safety problem for both buildings and public areas is fire detection. The reliability and coverage of traditional approaches depending on sensor networks are constrained (Rahardjo, et al. 2020).

Fire Detection and Classification

Fire detection systems are designed to discover fires early in their development when time will still be available for the safe evacuation of occupants. Early detection also plays a significant role in protecting the safety of emergency response personnel. Property loss can be reduced and downtime for the operation minimized through early detection because control efforts are started while the fire is still small.

The Need for Advanced Fire Detection

The limitations of traditional fire detection systems are evident in their coverage and reliability. An early fire detection system can save lives by alerting building occupants to the presence of a fire and allowing them to evacuate safely. Early warning alerts can notify a select group of people of a potential issue in the making. This buys valuable time to act on safety protocols and extinguish the hot spot before turning into a threatening hazard (MoviTHERM, 2018).

Computer Vision and Deep Learning

Computer vision (CV) is the scientific field which defines how machines interpret the meaning of images and videos. Computer vision algorithms analyze certain criteria in images and videos and then apply interpretations to predictive or decision-making tasks. Deep learning has been a game changer in the field of computer vision. It's widely used to teach computers to "see" and analyze the environment similarly to the way humans do.

Convolutional Neural Networks, also known as CNNs, are a subset of artificial neural networks used in deep learning and are frequently employed for object and picture recognition and categorization. Thus, Deep Learning uses a CNN to identify items in a picture. CNNs are being used extensively in a variety of tasks and activities, including speech recognition in natural language processing, video analysis, and difficulties with image processing, computer vision, and self-driving car obstacle detection. Due to their major contribution to these rapidly developing and expanding fields, CNNs are widely used in deep learning.

Artificial Neural Network (ANN)

An input layer, hidden layers, and an output layer are common components of an artificial neural network. ANNs draw inspiration from the structure of the human brain. Artificial neurons, or nodes in ANNs, collect inputs, process them, and deliver the result as output, much like a neuron in the brain does when it distributes information throughout the body. The input is the image. The input layer accepts arrays of the picture pixels as input.

There may be several hidden layers in ANNs that use mathematics to extract features from the image. Their construction is driven by the brain's visual cortex, which is made up of alternating layers of basic and sophisticated cells. Although there are many different types of ANN designs, they all generally comprise convolutional and pooling (or subsampling) layers that are organized into modules.

Fig 2.4 shows an active Artificial Neural Network which is the basic building block of every deep learning network or algorithm.

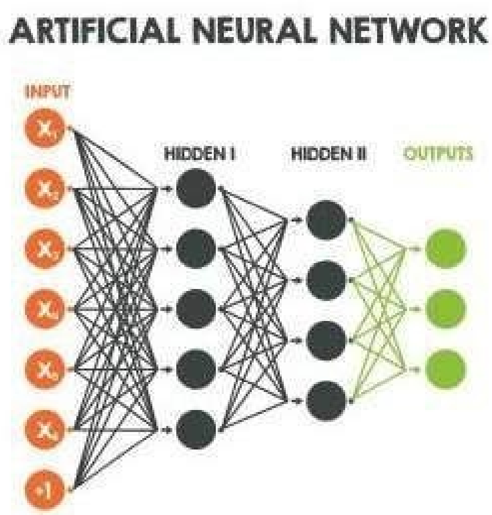

Fig.2.4: The architecture of basic Artificial Neural Network (ANN)

Convolutional Neural Network (CNN)

A CNN typically has three layers: a convolutional layer, a pooling layer, and a fully connected layer.

Convolution Layer: The majority of CNN's computations are handled by the convolution layer, which is its core. Between a kernel (a collection of teachable parameters) and a particular region of the receptive field, it conducts a dot product. The kernel matches the image's depth, covering all of its channels, including RGB, despite being shorter and wider than the image in height and width.

Pooling Layer: The pooling layer reduces the representations spatial size and processing requirements by condensing the network's output in particular locations by averaging neighbouring outputs. There are other pooling functions, but the most popular one is max pooling, which identifies the neighborhood with the highest output.

Fully-connected layer: The Fully Connected (FC) layer consists of the weights and biases along with the neurons and is used to connect the neurons between two different layers. These layers are usually placed before the output layer and form the last few layers of a CNN Architecture (Dharmaraj, 2022).

Transferred Learning with VGG16

VGG16, short for "Visual Geometry Group 16," is a specific type of Convolutional Neural Network (CNN) . VGG16 is composed of several layers, and while it follows the general structure of a CNN, which includes Convolutional layers, pooling layers, and fully connected layers, it is particularly known for its deep architecture featuring 16 weight layers.

Convolutional Layer:

Unlike CNN architectures that use a variety of kernel sizes, VGG16 exclusively uses 3x3 convolutional kernels, which are small but effective at creation of features from images. Each kernel operates over all the channels of its receptive field, capturing intricate patterns and details.

Pooling Layer:

Similar to other CNNs, VGG16 incorporates pooling layers specifically designed for down sampling the spatial dimensions of the input volume. These layers are usually of the max-pooling type. By picking the maximum value from a neighborhood of values, max-pooling ensures that the most prominent features are retained while reducing the computational complexity for the upcoming layers.

Fully-connected Layer:

VGG16 ends with three fully-connected (FC) layers. The first two have 4096 nodes each, and the last one corresponds to the number of classes for classification and usually contains 1000 nodes for a 1000-way classification problem like ImageNet. These layers play the role of integrating the features learned by the previous layers to make the final prediction.

Research Methodology

Fire Dataset Acquisition

There is a significant scarcity of publicly available datasets for fire detection in images, the factors that influenced our choices to use the Flame Vision dataset from Kaggle for this research work development are sizes, annotations, and balanced classes. The Flame Vision dataset consists of 8600 complete high-resolution photographs. A purposeful and intentional divide has been made within this collection: 5000 photographs are devoted to exhibiting diverse fire scenarios, and the remaining 3600 images show scenes without any fires. This particular distribution highlights the significance of distinguishing between fire and non-fire events in addition to ensuring that the models have a wide and varied set of data points to learn from. The dataset is neatly segregated into training, validation, and test sets as 80%, 10% and 10% respectively.

Table 3.1: Dataset Distribution

Training Data:	Validation Data:	Test Data:
Total: 6,800 images Fire: 3,600 No fire: 3,200	Total: 900 images Fire: 700 No fire: 200	Total: 900 images Fire: 700 No fire: 200

From (Sharma et al., 2017), there are images that are difficult for models to classify, and they are: Red coloured houses, sunsets, and vehicles with bright lights with different shades of yellow and red etc.

The Flame Vision dataset, although primarily designed for aerial fire detection, was adapted for our fire detection research.

Fire Image Augmentation

The raw dataset could not include all conceivable cases or variations. Then, controlled distortions to the images using augmentation methods including shearing, zooming, and horizontal flipping, were carried out in other to artificially grow the dataset or to increase the number of fire-scene in the images in the dataset.

Image Data Generators

Image Data Generators are indispensable tools when managing large volumes of image data. They not only streamline numerous preprocessing tasks, transforming potential challenges into smooth operations but also handle crucial tasks. These include scaling pixel values for normalization, implementing augmentation techniques, and batching images cohesively. As a result, during training sessions, data is consistently fed into the model in optimized batches, ensuring more streamlined and effective training cycles.

Artificial Neural Network (ANN) Model Architecture

The neural network designed for fire classification from the images consists of:

- **Input Layer**: Consists of 64x64 pixels with three channels (RGB), making up a total of 12,288 input neurons.
- **Hidden Layers**: Two dense hidden layers. The first one contains 128 neurons and the second one 64 neurons. Both utilize the ReLU activation function.
- **Output Layer**: A single neuron employing the sigmoid activation function, yielding a probability score.

Activation Function

ReLU (Rectified Linear Unit) was chosen as the activation function for the hidden layers due to its efficiency and capability to deal with the vanishing gradient problem. For the output layer, the sigmoid activation function was employed, providing a probability score between 0 and 1, indicative of the two classes (fire and no fire).

Training and Compilation:

Optimizer: Adam

Loss Function: Binary Cross entropy

Metrics: Accuracy

Epochs: 10

Convolutional Neural Network (CNN) Model Architecture

The CNN model developed for fire classification from the images has the following structure:

- **Input Layer**: Accepts images of dimension 64x64x3, representing the width, height, and RGB channels of the images.
- **Convolutional Layer 1**: This layer contains 32 filters (or kernels), each of size 3x3, coupled with the ReLU activation function. Given the input image size of 64x64x3, the resulting feature maps have dimensions of 62x62x32.

- **Pooling Layer 1**: Following the first convolutional layer, a max-pooling operation is applied with a 2x2 filter, reducing the spatial dimensions of the feature maps by half (i.e., 31x31x32).
- **Convolutional Layer 2**: Another set of 32 filters of size 3x3 is applied, again with the ReLU activation function, resulting in feature maps of 29x29x32.
- **Pooling Layer 2**: A subsequent max-pooling operation is applied with a 2x2 filter, further reducing the spatial dimensions of the feature maps (i.e., 14x14x32).
- **Flattening**: The 2D feature maps are transformed into a 1D vector.
- **Fully Connected Layer**: This dense layer contains 128 neurons, each activated by the ReLU function.
- **Output Layer**: A single neuron with the sigmoid activation function, yielding a probability score.

Activation Function

ReLU (Rectified Linear Unit) is used as the activation function for both the convolutional and fully connected layers because of its efficiency and ability to mitigate the vanishing gradient problem. The output layer, on the other hand, employs the sigmoid activation function to provide a probability score between 0 and 1, representing the two classes (fire and no fire).

Training and Compilation:

- Optimizer: Adam
- Loss Function: Binary Cross entropy
- Metrics: Accuracy
- Epochs: 10

Transfer Learning with VGG-16 Model Architecture

The VGG-16 architecture was utilized as a foundational model to classify fire from images. This architecture consists of:

- **VGG-16 Base Model**: The model employs a pre-trained VGG-16 network that has been trained on the ImageNet dataset. Given the provided input shape, the model accepts images of dimension 64x64x3. The top layers of the VGG-16 model are excluded (i.e., the fully connected layers that perform the classification in the original VGG16 model).
- *Frozen Layers*: All the layers in the VGG16 base model have been frozen, meaning they will not be updated during training. This is to preserve the features learned from the ImageNet dataset.
- **Flatten Layer**: To transform the 2D feature maps into a 1D vector for the dense layers.
- **Fully Connected Layer**: Comprises 256 neurons and utilizes the ReLU activation function.
- **Output Layer**: A single neuron that employs the sigmoid activation function, which provides a probability score indicating the presence or absence of fire.

Feature Extraction Method:

- The VGG-16 model, pre-trained on the ImageNet dataset, is employed for feature extraction. This model is designed with several convolutional and pooling layers which capture hierarchical patterns in images. By using this pre-trained model and excluding its top fully connected layers, we extract features from our images that are then fed into the custom dense layers for classification. The inclusion of the Flatten layer transforms these 2D spatial features into a 1D vector which can be input into the dense layers.

Activation Functions:

- **ReLU (Rectified Linear Unit)**: Used in the fully connected layer of 256 neurons. Chosen due to its efficiency in handling non-linearity and mitigating the vanishing gradient problem.
- **Sigmoid**: Utilized in the output layer to yield a probability score between 0 and 1, indicative of the two classes (fire and no fire).

Training and Compilation:

- **Optimizer**: Adam
  - **Loss Function**: Binary Cross Entropy
  - **Metrics**: Accuracy
- **Epoch:** 10

Performance Evaluation Metrics

The performance evaluation metrics for this research work is based on the following statistical analysis namely Accuracy, Precision, Recall and F1-Score.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (3.1)$$

$$Precision = \frac{TP}{TP + FP} \quad (3.2)$$

$$Recall = \frac{TP}{TP + FN} \quad (3.3)$$

$$F1 - Score = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (3.4)$$

Specification of System

The computational experiments were executed on Google Colab's GPU T4, specifically the NVIDIA Tesla T4. This GPU, with its Turing architecture and 16 GB GDDR6 memory, offers up to 8.1 TFLOPS of single-precision performance. Such specifications ensure efficient training and validation processes, especially for deep learning models.

Results and Discussion

Fire Dataset

Figure 4.1 shows the different classes (Fire and No-Fire) of images in the Flame Vision dataset used for this research work.

(b) No-Fire Image (a) Fire Image

Figure 4.1: Sample of different classes (Fire and No-Fire) of images in the Flame Vision dataset

The original images in the Flame Vision dataset were resized from 512x512 pixels to 64x64 pixels. The image normalization helped in rescaling the pixels from a scale of 0-255 to a scale of 0-1, this helps to speed up the convergence while training. Setting a shear range of 0.2 slants the images along their axis by a maximum number of 20%. Setting a zoom range of 0.2 magnifies or reduces the size of the content of the images by up to 20% i.e the images are zoomed in or zoomed out up to a level of 20%, Horizontal flip mirrors the images along the vertical axis, it flips the images horizontally like a mirror.

Image Augmentation

Figure 4.2 show the original and augmented images. The augmented images show the manipulated original images.

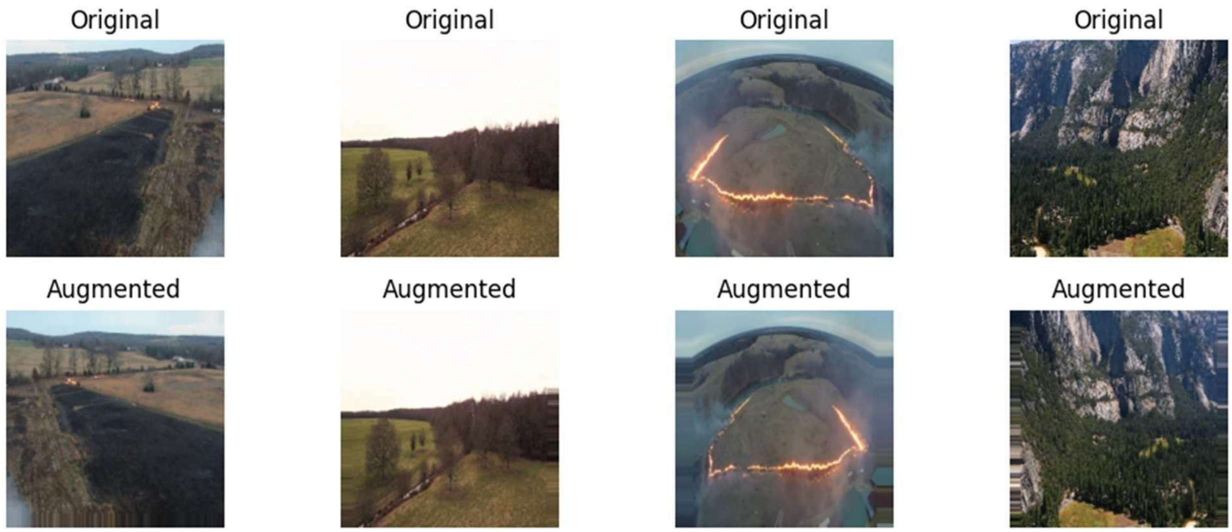

Figure 4.2: Original vs Augmented Images

Convolutional Neural Network

The training log for CNN is shown in Table 4.2, which shows the loss, accuracy, validation loss and validation accuracy and the visualization plot are shown in Figure 4.6 and 4.7. Figure 4.6 shows the CNN model accuracy for the training and validation while Figure 4.7 shows the CNN model loss for the training and validation. Figure 4.8 shows the confusion matrix of the CNN model.

Table 4.1.3: Training log for CNN

Epoch 1/10

213/213 [===========] - 81s 349ms/step - loss: 0.3530 - accuracy: 0.8487 - val_loss: 0.3884 val_accuracy: 0.9056 Epoch 2/10

213/213 [===========] - 64s 300ms/step - loss: 0.1813 - accuracy: 0.9307 - val_loss: 0.1942 val_accuracy: 0.9333 Epoch 3/10

213/213 [===========] - 62s 292ms/step - loss: 0.1376 - accuracy: 0.9481 - val_loss: 0.1668 val_accuracy: 0.9422 Epoch 4/10

213/213 [===========] - 62s 293ms/step - loss: 0.0871 - accuracy: 0.9706 - val_loss: 0.4263 val_accuracy: 0.7589 Epoch 5/10

213/213 [===========] - 62s 292ms/step - loss: 0.0667 - accuracy: 0.9753 - val_loss: 0.4882 val_accuracy: 0.8933 Epoch 6/10

213/213 [===========] - 62s 292ms/step - loss: 0.0432 - accuracy: 0.9853 - val_loss: 0.0610 val_accuracy: 0.9911 Epoch 7/10

213/213 [===========] - 62s 293ms/step - loss: 0.0399 - accuracy: 0.9869 - val_loss: 0.5712 val_accuracy: 0.7300 Epoch 8/10

213/213 [===========] - 63s 294ms/step - loss: 0.0326 - accuracy: 0.9906 - val_loss: 0.2901 val_accuracy: 0.9456 Epoch 9/10

213/213 [===========] - 63s 298ms/step - loss: 0.0273 - accuracy: 0.9906 - val_loss: 0.4483 val_accuracy: 0.8622 Epoch 10/10

213/213 [===========] - 63s 294ms/step - loss: 0.0264 - accuracy: 0.9921 - val_loss: 0.2960 val_accuracy: 0.8989

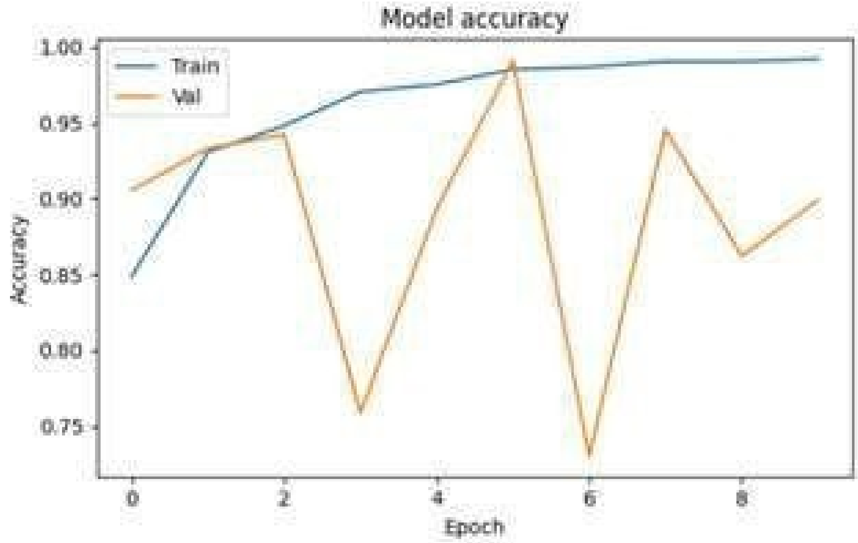

Figure 4.1.3 : CNN model accuracy for training and validation

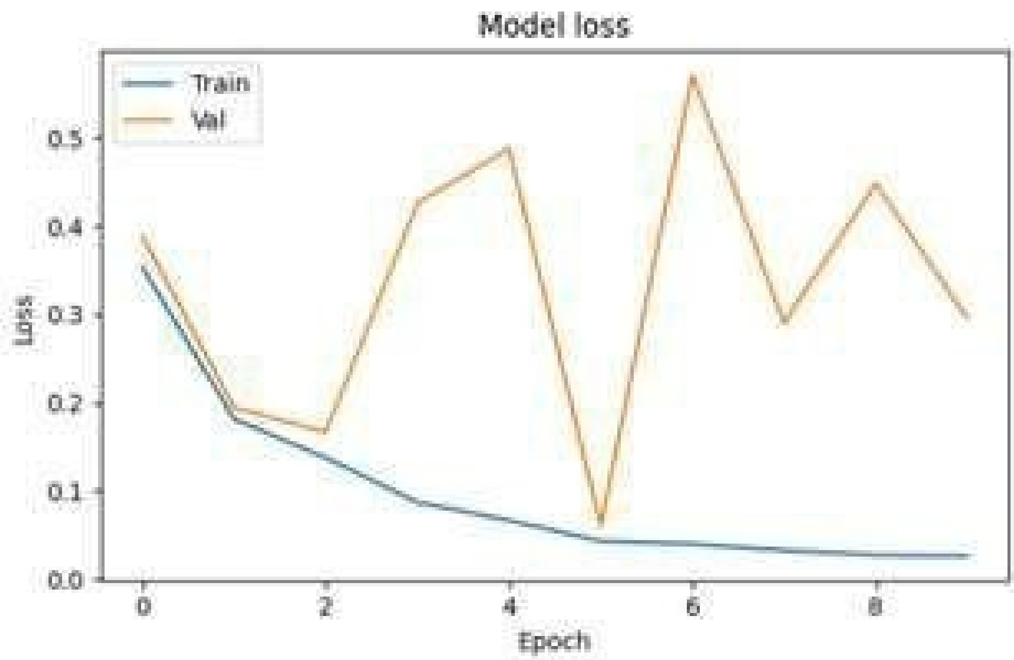

Figure 4.1.4: CNN model loss for training and validation

It is observed from Table 4.1.3 and Figures 4.1.3 to 4.1.4 that the accuracy increased from 0.0.8487 to 0.9921 with a decreased in the loss from 0.3530 to 0.2640 with increased in the number of epochs. This shows the effectiveness of the CNN model. This shows the effectiveness of the CNN model.

Figure 4.1.5: Confusion Matrix for CNN Model

The confusion matrix for the CNN model in Figure 4.8 shows that 189 images are true negatives (detected as true images of no-fire), 11 images as false negatives (detected as images of no-fire when they are images of fire), 16 images as false positives (detected as images with fire when the images of no-fire) and 684 as true positives (detected as true images of fire).

Transfer Learning with VGG-16

The training log for VGG-16 is shown in Table 4.3, which shows the loss, accuracy, validation loss and validation accuracy and the visualization plot are shown in Figure 4.9 and 4.10. Figure

4.9 shows the VGG-16 model accuracy for the training and validation while Figure 4.10 showsthe VGG-19 model loss for the training and validation. Figure 4.11 shows the confusion matrix of the VGG-16 model.

Table 4.1.6: Training log for VGG-16

Epoch
1/10

213/213 [===========] - 87s 358ms/step - loss: 0.1876 - accuracy: 0.9297 - val_loss: 0.1748 val_accuracy: 0.9389 Epoch 2/10

213/213 [===========] - 69s 326ms/step - loss: 0.0730 - accuracy: 0.9766 - val_loss: 0.0930 val_accuracy: 0.9544 Epoch 3/10

213/213 [===========] - 62s 293ms/step - loss: 0.0514 - accuracy: 0.9824 - val_loss: 0.1301 val_accuracy: 0.9656 Epoch 4/10

213/213 [===========] - 62s 291ms/step - loss: 0.0344 - accuracy: 0.9884 - val_loss: 0.1218 val_accuracy: 0.9511 Epoch 5/10

213/213 [===========] - 62s 292ms/step - loss: 0.0343 - accuracy: 0.9878 - val_loss: 0.1542 val_accuracy: 0.9656 Epoch 6/10

213/213 [===========] - 62s 291ms/step - loss: 0.0246 - accuracy: 0.9921 - val_loss: 0.1825 val_accuracy: 0.9467 Epoch 7/10

213/213 [===========] - 63s 296ms/step - loss: 0.0215 - accuracy: 0.9926 - val_loss: 0.0868 val_accuracy: 0.9689 Epoch 8/10

213/213 [===========] - 63s 298ms/step - loss: 0.0127 - accuracy: 0.9963 - val_loss: 0.0534 val_accuracy: 0.9778 Epoch 9/10

213/213 [===========] - 63s 295ms/step - loss: 0.0142 - accuracy: 0.9954 - val_loss: 0.0666 val_accuracy: 0.9722 Epoch 10/10

213/213 [===========] - 62s 293ms/step - loss: 0.0185 - accuracy: 0.9943 - val_loss: 0.0902 val_accuracy: 0.9722

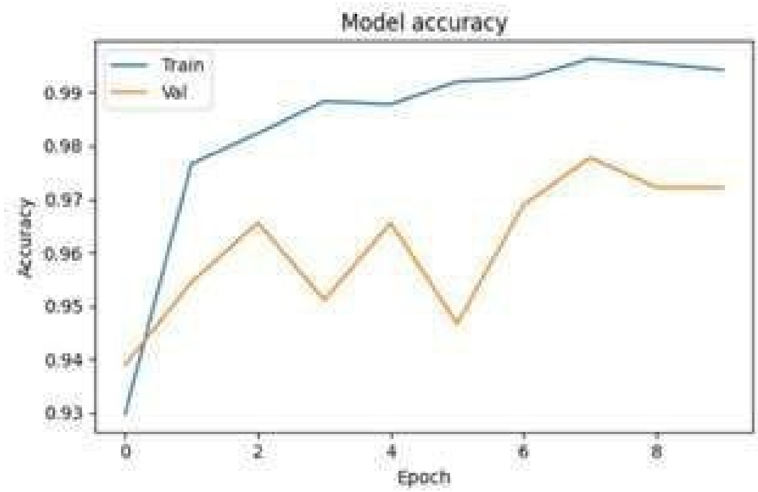

Figure 4.9: VGG-16 model accuracy for training and validation

Conclusion and Recommendation

In conclusion, This paper suggests that CNN and transfer learning models using VGG-16 offer the most promising results for the task of wildfire image classification. These models not only yield higher accuracy but also demonstrate a more balanced trade-off between precision and recall compared to traditional ANN.

Recommendations

The following recommendations are suggested for the improvements on the research work in the future.

1. It is recommended that we create custom Coupled-Circuit Television (CCTV) dataset for this kind of study in order to have better features extracted that will be more representative of reallife scenarios for CCTV system integrations.
2. Future research could focus on the real-world implementation of these models, considering computational constraints and evolving fire patterns.

References

Adurkar,A., P. Patel, M. Sabnis and P. Patil,. (2022). Fire Detection using HSV Color Picker: A

Review. 5th International Conference on Advances in Science and Technology (ICAST), (pp. 450-455). Mumbai, India

Alikhujaev, Y., Abdusalomov, A., & Cho, Y. I. (1241). Automatic Fire and Smoke Detection Method for Surveillance Systems Based on Dilated CNNs. *Atmosphere*, *11*.

Allison, R., Johnston, J., Craig, G., and Jennings, S. (2016). Airborne Optical and thermal Remote Sensing for Wildfire Detection and Monitoring. Sensors. 16 (8), 1310. doi:10.3390/s16081310

Avazov, K., Mukhiddinov, M., Makhmudov, F., & Cho, Y. I. (2021). Fire detection method in smart city environments using a deep-learning-based approach. *Electronics*, *11*(1), 73. https://doi.org/10.3390/electronics11010073

Barmpoutis, P., Papaioannou, P., Dimitropoulos, K., and Grammalidis, N. (2020). A Review on Early forest Fire Detection Systems Using Optical Remote Sensing. Sensors. 20 (22), 6442. doi:10.3390/s20226442

Ba, R., Chen, C., Yuan, J., Song, W., & Lo, S. (2019). SmokeNet: Satellite smoke scene detection using convolutional neural network with spatial and channel-wise attention. *Remote Sensing*, *11*(14), 1702. https://doi.org/10.3390/rs11141702

Barmpoutis, P., Stathaki, T., Dimitropoulos, K., & Grammalidis, N. (2020). Early fire detection based on aerial 360-degree sensors, deep convolution neural networks and exploitation of fire dynamic textures. *Remote Sensing*, *12*(19), 3177. https://doi.org/10.3390/rs12193177Barry. (2017, December 21). *The four different types of fire detectors*. A Total Solution. https://www.atotalsolution.com/blog/fire-protectionthe-four-types-of-fire-detectors/

Brown, L. E., Palmer, S. M., Johnston, K., & Holden, J. (2015). Vegetation management with fire modifies peatland soil thermal regime. *Journal of Environmental Management*, *154*, 166–176. https://doi.org/10.1016/j.jenvman.2015.02.037

Chollet, F.: Keras (2015)

Cheong-mo, Y. (2020, January 6). *Number of fires, fire deaths fall in 2019*. Yonhap News Agency;

Yoo Cheong-mo. https://en.yna.co.kr/view/AEN20200106008000315

closed circuit television (CCTV). (2011, March 22). Whatis.com; TechTarget.

https://www.techtarget.com/whatis/definition/closed-circuit-television-CCTV

Contributor, T. (2011, March 22). *closed circuit television (CCTV)*. Whatis.com; TechTarget.

https:// www.techtarget.com/whatis/definition/closed-circuit-television-CCTV

Coen, J. L., and Schroeder, W. (2013). Use of Spatially Refined Satellite Remote Sensing Fire Detection Data to Initialize and Evaluate Coupled Weather-wildfire Growth Model Simulations. Geophys. Res. Lett. 40 (20), 5536–5541. doi:10.1002/2013gl057868

Craig Schroll, B. Y. R. (2007). *Fire detection and alarm systems: A brief guide -- occupational health & safety*. Occupational Health & Safety.

https://ohsonline.com/Articles/2007/12/Fire-Detection-and-Alarm-Systems-A-Brief-Guide.aspx

Chen, C., Cheng, Y., H. Bai, C. Mou and Y. Zhang,. (2019). Research on Image Fire Detection Based on Support Vector Machine. 9th International Conference on Fire Science and Fire

Protection Engineering (ICFSFPE), (pp. 1-7). Chengdu, China

Deep learning for computer vision: The abridged guide. (n.d.). Run.Ai. Retrieved August 14, 2023, from https://www.run.ai/guides/deep-learning-for-computer-vision

Dharmaraj. (2022, June 1). *Convolutional neural networks (CNN) — architecture explained*. Medium. https://medium.com/@draj0718/convolutional-neural-networks-cnnarchitectures-explained-716fb197b243

Duggal, N. (2020, October 21). *What is data processing: Cycle, types, methods, steps and examples*. Simplilearn.com; Simplilearn. https:// www.simplilearn.com/what-is-data-processing-article

Frizzi, S., Kaabi, R., Bouchouicha, M., Ginoux, J.M., Moreau, E., Fnaiech, F.:Convolutional neural network for video fire and smoke detection. In: IECON 2016–42nd Annual Conference of the IEEE Industrial Electronics Society, pp. 877–882, October 2016

Gaur, A., Singh, A., Kumar, A., Kumar, A., &Kapoor, K. (2020). Video flame and smoke-based fire

Geeks.com(2023)/Accessed online at https:// www.geeksforgeeks.org/tag/opencv/

Giglio, L., Boschetti, L., Roy, D. P., Humber, M. L., and Justice, C. O. (2018). The Collection 6 MODIS Burned Area Mapping Algorithm and Product. Remote sensing Environ. 217, 72–85. doi:10.1016/j.rse.2018.08.005Guede-Fernández, F., Martins, L., de Almeida, R. V., Gamboa, H., & Vieira, P. (2021). A deep learning based object identification system for forest fire detection. Fire, 4(4), 75

Justice, C., Giglio, L., Korontzi, S., Owens, J., Morisette, J., Roy, D., et al. (2002). The MODIS Fire Products. Remote sensing Environ. 83 (1-2), 244–262. doi:10.1016/s00344257(02)00076-7

Kaku, K. (2019). Satellite Remote Sensing for Disaster Management Support: A Holistic and Staged Approach Based on Case Studies in Sentinel Asia. Int. J. Disaster Risk Reduction. 33, 417–432. doi:10.1016/j.ijdrr.2018.09.015Leblon, B., Bourgeau-Chavez, L., and SanMiguel-Ayanz, J. (2012). Use of Remote Sensing in Wildfire Management. London:

Sustainable development-authoritative and leading edge content for environmental management, 55–82.

Li, Z., J. Liu, C. Guo, S. Li, L. Li and L. Xu, . (2020). Design of Smoke Detection and Alarm System Based on LoRa. Chinese Automation Congress (CAC) . Shanghai, China

Maksimovic, M., Vujovic, V., Miloevic V. and B. Pericic,. (2014). Evaluating the Optimal Heat Detector Deployment for Fire Detection. 2014 International Conference on Engineering and Telecommunication. Moscow, Russia.

Logunova, I. (2022, August 5). *Deep Learning Applications for Computer Vision*. Serokell Software Development Company; Serokell. https://serokell.io/blog/deep-learning-forcomputer-vision

Manolis Loukadakis, José Cano and Michael O'Boyle (2108). Accelerating Deep Neural Networks on Low Power Heterogeneous Architectures.Martin, R. (2023). closed-circuit television. In *Encyclopedia Britannica*.

McWethy, D. B., Schoennagel, T., Higuera, P. E., Krawchuk, M., Harvey, B. J., Metcalf, E. C., Schultz, C., Miller, C., Metcalf, A. L., Buma, B., Virapongse, A., Kulig, J. C., Stedman, R. C., Ratajczak, Z., Nelson, C. R., & Kolden, C. (2019). Rethinking resilience to wildfire. *Nature Sustainability*, *2*(9), 797–804. https://doi.org/10.1038/s41893-019-03538

Mishra, M. (2020, August 26). *Convolutional neural networks, explained*. Towards Data Science. https://towardsdatascience.com/convolutional-neural-networks-explained-9cc5188c4939

Morisette, J. T., Giglio, L., Csiszar, I., and Justice, C. O. (2005). Validation of the MODIS Active Fire Product over Southern Africa with ASTER Data. Int. J. Remote Sensing. 26 (19), 4239–4264. doi:10.1080/01431160500113526

MoviTHERM, B. (2018). *5 benefits of a fire detection system*. MoviTHERM. https://movitherm.com/knowledgebase/benefits-of-a-fire-detection-system/

MVD Prasad, N. Sai Kiran, Ch Sumanth, T V V R N Sri Harsha, Sk Hasane Ahammad. (2021). *Video surveillance-based security system using OpenCV and Arduino uno*.

Opoulos, K., Barmpoutis, P., & Grammalidis, N. (2015). Spatio-temporal flame modeling and dynamic texture analysis for automatic video-based fire detection. *IEEE Trans. Circuits Syst. Video Technol*, *25*, 339–351.

Pefipresa. (2022, July 12). *▷ Types of Fire Detectors · How they work · Classes and models ▷*. Pefipresa; Pefipresa S.A. https://www.pefipresa.com/en/blog-fp/types-of-fire-detectors-how-they-work/Rahardjo, H. A., & Prihanton, M. (2020). The most critical issues and challenges of fire safety for building sustainability in Jakarta. *Journal of Building Engineering*, *29*(101133), 101133. https://doi.org/10.1016/j.jobe.2019.101133

Ryu, J., Cho, M., Kim, J. S., & Lee, C.-W. (2018). A study on criteria for firefighting service allocation in Korea. *Korean Policy Sciences Review*, *22*(3), 47–73. https://doi.org/10.31553/kpsr.2018.09.22.3.47

Seok-min, O. (2020, October 9). *Fire continues to burn at tower block in Ulsan; 88 taken to hospital*. Yonhap News Agency; Oh Seok-min. https://en.yna.co.kr/view/AEN20201009001400325

Sharma, J., Granmo, O.C., Goodwin, M., & Fidje, J.T. (2017). Deep Convolutional Neural Networks for Fire Detection in Images. In: G. Boracchi, L. Iliadis, C. Jayne, & A. Likas (eds) Engineering Applications of Neural Networks. EANN 2017. Communications in Computer and Information Science, vol 744. Springer, Cham. https://doi.org/10.1007/978- 3-319-65172-9_16

Srividhya, S., Sindhu, S., Sornalakshmi, K., Hemavathi D. and V. Rajaram,. (2022). Fire Detection System using Deep Learning Techniques. 2022 4th International Conference