International Journal of Data and Network Science 9 (2025) ***–***

Contents lists available at GrowingScience

International Journal of Data and Network Science

homepage: www.GrowingScience.com/ijds

Employing CNN mobileNetV2 and ensemble models in classifying drones forest fire detection images

Dima Suleiman[a], Ruba Obiedat[a], Rizik Al-Sayyed[a], Shadi Saleh[b], Wolfram Hardt[b] and Yazan Al-Zain[c*]

[a]*King Abdullah II School for Information Technology, The University of Jordan, Amman, 11942, Jordan*
[b]*Department of Computer Engineering, Chemnitz University of Technology Chemnitz, Germany*
[c]*Department of Industrial Engineering, The University of Jordan, Amman, 11942, Jordan*

C H R O N I C L E

Article history:
Received: August 20, 2024
Received in revised format: September 25, 2024
Accepted: October 10 2024
Available online: October 10, 2024

Keywords:
Forest fire detection
Drone imagery
MobileNetV2
Ensemble learning
DeepFire dataset
Transfer learning

A B S T R A C T

In recent years, the adoption of advanced machine learning techniques has revolutionized approaches to solving complex problems, such as identifying occurrences of forest fires. Among these techniques, the use of Convolutional Neural Networks (CNNs) combined with ensemble methods is particularly promising. To investigate the feasibility of detecting fires using video streams from Unmanned Aerial Vehicles (UAVs), the lightweight CNN architecture MobileNetV2 was utilized for real-time detection. Several experiments were conducted on the DeepFire dataset, which comprises an equal number of images with and without fire, to evaluate MobileNetV2's performance. Notably, the architecture's linear bottlenecks and the efficient use of inverted residuals ensure high accuracy without compromising on feature extraction capabilities. For a comprehensive assessment, MobileNetV2 was benchmarked against other models, including DenseNet121, EfficientNetV2S, and VGG16. Accuracy was enhanced by averaging predictions through methods such as voting or summing results. As documented in the literature, MobileNetV2 consistently outperforms other architectures in computational efficiency and provides an excellent balance between efficiency and the quality of learned features over multiple epochs. This study underscores the suitability of MobileNetV2 for real-time applications on drones, particularly for the detection of forest fires in resource-constrained environments. The results show that MobileNetV2 achieves the highest accuracy (0.994), sensitivity (0.994), and specificity (0.998) among the tested models, with low standard deviations across all metrics. In contrast, EfficientNetV2S exhibited the lowest accuracy and sensitivity, both at 0.779, with a specificity of 0.829. The ensemble (Sum) method achieved an average accuracy of 0.989, sensitivity of 0.989, and specificity of approximately 0.988. Therefore, MobileNetV2 not only delivers the highest accuracy and stability but also demonstrates that the choice of ensemble method significantly affects the results.

1. Introduction

According to the World Health Organization (WHO) (WHO, 2022), National Interagency Fire Center (NIFC) (NIFC, 2022), and (Lee et al., 2017) wildfires are responsible for 10 thousand deaths and injuries yearly. The number and intensity of fires are rising year by year due to climate change and other factors (Davis & Shekaramiz, 2022), National Wildland Fire Situation (NWFS) reported 2765 fires in 2023 report (NWFS, 2023). In addition to the threat to human life, wildlife and ecosystems, a massive financial loss is caused by the forests' fires, which damage millions of hectares of lands, produce air pollution and destroy

* Corresponding author.
E-mail address y.alzain@ju.edu.jo (Y. Al-Zain)

ISSN 2561-8156 (Online) - ISSN 2561-8148 (Print)

doi: 10.5267/j.ijdns.2024.10.004

infrastructure, homes and wildlife (Mousavi & Ilanloo, 2023). These fires are caused by many factors, most of them related to human activities (Davis & Shekaramiz, 2023; Guede-Fernández et al., 2021). Forests cover around 29% of the earth with 4 billion hectares' land (Oom & Pereira, 2013; Seydi et al., 2022), they present an essential element for many creatures, providing food, fuel and shelter for species. Further, it is important for producing fresh air, preventing soil erosion and reducing global temperature (Davis & Shekaramiz, 2023; Jonnalagadda et al., 2024). An early detection of forest fires can help firefighters to respond quickly, minimize the amount of damage caused by these fires and most significantly save lives (Seydi et al., 2022). Wildfires are usually detected by many sensors such as temperature, flame and smoke detectors (Ghali et al., 2022; Jonnalagadda et al., 2024). Nowadays traditional fire detection tools are replaced by more advanced tools based on computer vision techniques, Unmanned Aerial Vehicles (UAVs) play an important role in detecting these fires (Mousavi & Ilanloo, 2023; Ghali et al., 2022).

UAVs, usually named as drones, are aircraft without pilots, they can be considered as flying robots, which can be controlled manually or by using special software with the help of sensors and Global Positioning System (GPS) (Aswini et al., 2021, Xiao, 2023). The usage of drones is accelerating these days because of the wide variety of applications that can be applied in (Aswini et al., 2021). The nature of UAVs makes them a proper tool for many applications as they have high speed, are lightweight and small, making them capable of accessing remote areas (Mousavi & Ilanloo, 2023). Furthermore, UAVs are strong tools, need no pilots and require low operational and maintenance cost, they can be equipped with cameras and sensors to detect heat, flame and capture images with precise and flexible spatial resolution which are hard to be obtained by fixed satellite orbits (Mousavi & Ilanloo, 2023; Bhatnagar et al., 2020). Thus, UAVs can be used in detecting and overcoming many natural disasters including flood detection, fire detection, volcano estimation and human rescue activities (Mousavi & Ilanloo, 2023). Combining artificial intelligence, computer vision and Deep Learning (DL) with the availability of powerful graphics processing units (GPUs) and drones encourage their usage in early fire detection systems (Davis & Shekaramiz, 2023). Computer vision using deep learning undertakes three main tasks namely, image classification, object detection and image segmentation. Image classification is applied to find out which objects exist in an image or video or whether an object presents in an image or not. On the other hand, object detection combines image classification and localization, where it identifies objects in an image or a video and specifies their position in that image utilizing bounding boxes. Further, image segmentation analyzes the images at a lower level by dividing the image into regions. The aim is to identify the useful areas based on the user's interest for further analysis like classification and object detection. Unlike classification and object detection, image segmentation provides pixel-by-pixel outlines of the objects (Osco et al., 2021).

Object detection is one of the main tasks in DL and computer vision that involves identifying objects in images or real time videos. Object detection can be classified under two main research categories; General Object Detection and Application based Detection. General Object Detection, which was applied in (Aswini et al., 2021), is used to identify different objects in the surrounding environment such as, fires, vehicles, animals, people, plants and others and used in different applications including disasters detections, autonomous driving and medical feature detection. Whereas application-based detection is used in certain applications such as face detection and line detection (Aswini et al., 2021). Object Detection research started with a traditional object detection approach and evolved to DL algorithms. DL has two approaches; a single-stage approach using YOLO different versions and a two-stage approach with Region-Based Convolutional Neural Networks (RCNN) (Aswini et al., 2021). Recently, DL gives promising results in the field of fire detection and there is a crucial need to improve the accuracy of wildfire detection and reduce the false prediction, which can lead to catastrophic consequences (Idroes et al., 2023). DL techniques can be used in detecting fires based on the fire's color and geometric features such as height, width and angle. It usually uses the images captured by drone's cameras and sensors as input for the model in order to locate the fire and determine its shape (Ghali et al., 2022, Jonnalagadda et al., 2024). Several researchers adapt DL and computer vision approaches in early fire detection. Typically, they follow these main steps: first, the data is collected from the UAVs' cameras and sensors; then onboard microchips with learning algorithms process this data, after that the data is transmitted to ground equipment for additional processing. Finally, the authorities will be notified if any action is required (Jonnalagadda et al., 2024).

Many studies utilize DL and image vision in early fire detection, for instance (Seydi et al., 2022) applied a Fire-Net DL framework on Landsat-8 satellite data taken from different regions, where active forest fires are heavily reported. The study combines the optical (Red, Green, and Blue) and thermal images from satellite data. The proposed approach was compared with the traditional machine learning (ML) algorithms and proved its superiority with 97.35% accuracy, besides proving its ability to strongly detect small fires. In the same context, Ghali et al. (2022) adapted a DL approach to detect wildfire by combining EfficientNet-B5 and DenseNet-201 models using aerial images. The study employed a deep CNN model (EfficientSeg) and two vision transformers (TransUNet and TransFire) for image segmentation to obtain precise results regarding the shape of fire and fire regions. The proposed ensemble model proves its ability in wildfire classification and segmentation by applying DL and vision transformers on the chosen flame dataset. They successfully achieved impressive results with 85.12% classification accuracy and 99.9%, 99.82% F1-score for semantic segmentation. In addition, Shamta and Demir (2024) studied the use of DL surveillance systems for early fire detection utilizing the images captured by drone's cameras. The study examined YOLOv8 and YOLOv5 for object detection used in identifying wildfires, besides using NN-RCNN to classify whether an image contained a fire or not and compare it with YOLOv8 classification results. The results showed that YOLOv8 and CNN-RCNN gave the same result with 96% accuracy for classification, while YOLOv8 and YOLOv5 obtained 89% object detection accuracy.

In this paper, four transfer learning pretrained CNN models are used, namely; DenseNet121, MobileNetV2, EfficientNetV2S, and VGG16. In addition, two ensembles learning are utilized: ensemble majority voting and ensemble sum. Experiments are conducted on a DeepFire dataset consisting of 1900 images. The rest of this research is organized as follows: Section 2 covers literature review, the methodology is described in Section 3 while the description of the dataset and the explanations of experiments and results are discussed in Section 4. Finally, Section 5 presents the conclusion.

2. Literature review

Recently, transfer learning has been applied to drone-based forest fire detection since it provided significant results. Pre-trained Convolutional Neural Networks (CNNs) such as DenseNet, MobileNet and ResNet have been used to implement fire detection models that can identify fire in UAV images efficiently. EfficientNet-B5 and DenseNet-201 transfer learning CNN models were combined to generate an ensemble learning model to detect wildfire using aerial images (Ghali et al., 2022). Even more, the precise fire regions were identified by using the EfficientSeg model and two transformers visions called TransUNet and TransFire. Experiments were made using 48,010 images of the FLAME dataset. The generated classification model outperformed most state of art models where it achieved 85.12% accuracy. On the other hand, segmentation models achieved F1-score of 99.9% and 99.82% for TransUNet and TransFire respectively. Furthermore, Ghali et al. fire detection model can extract finer details in aerial images in addition to addressing the complexity of background and small areas containing fire.

FFireNet, a novel CNN-based model for forest fire detection was used in another research (Khan et al., 2022a). The FFireNet model combined Internet of Things and Artificial Intelligence (AI) technologies, emphasizing their role in environmental monitoring, which has influenced wildfire patterns. Strengths of their work include its robust performance metrics, detailed comparative analysis, and the development of a specialized dataset for forest fire detection. However, limitations such as potential variability in dataset image quality, a single false negative, and the need for further validation in diverse conditions suggest areas for improvement and further research. The experimental results showed a notable accuracy of 98.42%, with precision and recall rates of 97.42% and 99.47%, respectively. This superior performance compared to existing methods highlights its potential for early fire detection, which is essential for mitigating large-scale wildfires.

On the other hand, particle swarm optimization (PSO) technique and five transfer learning models including MobileNet, ResNet, AlexNet, VGGNet, and GoogLeNet with the help of drones network were used to develop a plan of fire quenching (Manoj & Valliyammai, 2023). PSO technique was used to detect the shortest and best path to fire quenching plan. This was achieved by using shortest path algorithms on a list of water bodies' locations near to fire location. In case of fire, once the leading drone reaches the nearest point to fire, it communicates with other nearby drones located near the water bodies, which in turn perform the fire quenching task. GoolgeNet-TL technique provided the best results in terms of accuracy and F1-score, with values of 96% and 97%, respectively. Moreover, a PSO-based Federated Learning (FL) strategy was presented to address the major problem of forest fires that seriously destroy infrastructure and human lives (Supriya et al., 2023). Some of the difficulties encountered by conventional ML and DL techniques include managing enormous amounts of multidimensional data, transmission delay, communication lags, processor power limitations, and privacy issues. Federated Learning (FL) provides a way to reduce processing overhead and ensure privacy while processing large amounts of data efficiently. However, a major problem with FL is its high communication overhead caused by the model weights being transferred between clients. A combination between FL with the PSO method was used to lessen this. Using spatial data trends, this hybrid approach seeks to improve response times to forest fires. The PSO-enabled FL framework fared better than the conventional federated average model in addressing data imbalance, lowering communication costs, and enhancing network efficiency when tested using multidimensional forest fire image data from Kaggle. With a prediction accuracy of 94.47%, the suggested model showed promise as an essential part of creating early warning systems for forest fires.

An ensemble of CNNs in conjunction with a staged YOLO model was used to provide an early wildfire and smoke detection solution (Bahhar et al., 2023). Given the destructive nature of forest fires, the suggested architecture integrated several CNN architectures for two computer vision tasks: detection and classification, with the goal of improving detection efficiency. Here's how the two-stage pipeline operates: Step 1: A CNN finds anomalies within the frame. Stage 2: The YOLO architecture locates the smoke or fire if an anomaly is found. By applying transfer learning, the classification model produced remarkable outcomes with a 0.95 F1-score, 0.99 accuracy, and 0.98 sensitivity. The detector model also functioned well, with a mean average precision (mAP) of 0.76 mAP for the combined model and 0.85 mAP at a 0.5 threshold for smoke detection. The F1-score for the smoke detection model was 0.93. Although there were certain obstacles to overcome, like the scarcity of high-quality real-world UAV-captured images of fire and smoke, the suggested DL pipeline produced promising experimental outcomes and has the potential to be used in early wildfire detection systems.

In addition, a fire detection and geolocalization two-stage framework was used in (Choutri et al., 2023). Experiments were performed on a large dataset consisting of more than 12,000 images compiled from several resources related to scenes of fire. The regions of interest in images were surrounded by bounding boxes labeled using one of three values, namely; fire, non-fire, and smoke. In addition, YOLO-NAS model was trained on the collected dataset to perform both fire detection and localization. Finally, stereo vision was used to locate fires where the Pixhawk microcontroller was employed in drones. The macro average precision of YOLO-NAS was 71% while the model achieved 68% F1-score. On the other hand, Shamta and Demir (2024) presented an early forest fire detection system by using images captured from cameras placed on a four-rotor UAV. The fire related data was displayed by ground station interface. Moreover, onboard NVIDIA Jetson Nano was used as real time hardware that contains an embedded DL algorithm to help UAV in detecting forest fire. Furthermore, CNN-RCNN network was created to classify whether an image contains a fire or not while the performance of two object detection methods including YOLOv8 and YOLOv5 was examined for forest fire detection. Experiments showed that the accuracy for YOLOv8 and YOLOv5 object detection were 96% and 89%, respectively, while it was 96% for CNN-RCNN fire images classification.

The problem of real-time wildfire identification in contexts with limited computational resources was discussed in (Tsalera et al., 2023), considering the increasing frequency of wildfires as a result of drought and climate change. Lightweight CNNs like SqueezeNet, ShuffleNet, and MobileNetv2, as well as more sophisticated ResNet-50 were used. Also, to recreate realistic settings and conduct cross-dataset comparisons, the authors utilized a number of datasets, including the Forest-Fire and Fire-Flame datasets in addition to third-party photos. Furthermore, lightweight networks are preferred to control operating costs and computing resources. Even more, the contextualization was investigated via ResNet-18 picture semantic segmentation, with an emphasis on identifying components associated with energy infrastructures. The identification findings demonstrated a 96% classification accuracy and good cross-dataset performance.

Early detection of forest fire was achieved using the Forest Defender Fusion system (Ibraheem et al., 2024). High detection accuracy was gained by providing drones energy consumption regulations from Forest Fusion System. In addition to using Enhanced Consumed Energy-Leach protocol (ECP-LEACH) alongside the VGG16 Intermediate Fusion model to enhance the accuracy. Furthermore, the FLAME2 dataset was used in experiments where the system achieved an accuracy value of 99.86%.

Zheng et al. (2024) introduced a modified deep convolutional neural network model (MDCNN) for recognizing and localizing forest fires in video imagery using a deep learning-based approach. To enhance the model for fire image recognition, the authors employed transfer learning. Furthermore, the imprecise flame detection was addressed by integrating the deep CNN with a novel feature fusion algorithm. A diverse training dataset of fire and non-fire images was used to fine-tune the model, improving detection accuracy. The MDCNN model achieved a low false alarm rate (0.563%), a false positive rate (12.7%), a false negative rate (5.3%), a recall rate (95.4%), and an overall accuracy of 95.8%. The results showed significant improvement in flame recognition accuracy, demonstrating the model's strong generalization ability.

The temperature and smoke sensors used in conventional fire detection systems have limitations in terms of response time, range, and environmental compatibility. These sensor systems need to be maintained on a regular basis and are also expensive. Therefore, a fire detection system based on images from security cameras and the DenseNet 201 algorithm was presented in (Muhammad & Alrikabi, 2024). A DL-based computer vision method, namely a CNN, to enhance fire detection and lower false alarms was used. With an average predicted accuracy of 98% on the dataset, the system can identify fires as soon as they start by using security camera footage. It is an affordable and practical substitute for sensor-based systems due to its great precision.

3. Methodology

An ensemble method of pretrained CNN transfer learning models was utilized on fire detection images captured by Drones. Majorly, four pretrained models were used including (DenseNet121 (Huang et al., 2017), MobileNetV2 (Sandler et al., 2018), EfficientNetV2S (Tan, M., & Le, Q, 2021) and VGG16 (Simonyan, 2014)) as shown in Fig. 1. Using ensemble methods with transfer learning pretrained models accomplished significant performance in fire detection problems. In this research, to enhance the performance of fire detection classifiers, the decision of the four pretrained models was merged using majority voting. Moreover, MobileNetV2 provides significant results in the majority of experiments.

3.1 Transfer Learning (TL) and pretrained Convolutional Neural Network Models for fire detection images

Transfer learning (TL) is a ML method that is used to generate a model for a certain task and then reuse the same generated model for another task as a pretrained model. Usually, it is trained on large amounts of data. TL is used widely in DL and neural network applications since it provides several advantages such as minimizing training time, reducing data requirements, and enhancing generalization. In this research, four pretrained TL models were utilized in detecting fires including DenseNet121, MobileNetV2, EfficientNetV2S, and VGG16.

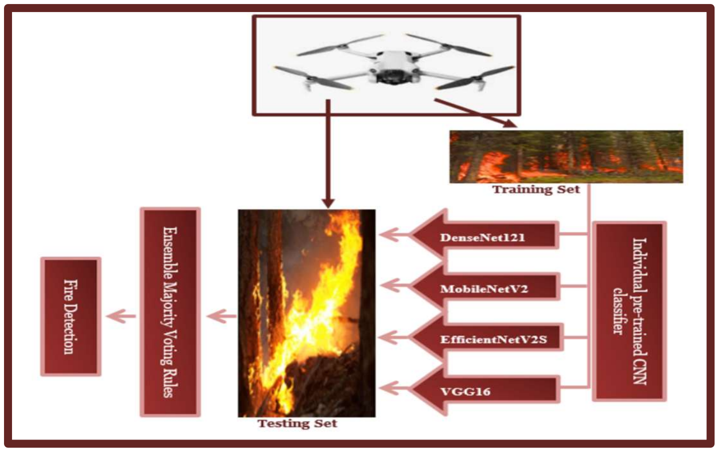


Fig. 1. DenseNet121, MobileNetV2, EfficientNetV2S, VGG16, and their Ensemble (majority voting) and Ensemble(sum)

3.1.1 Pretrained DenseNet121

DenseNet121 is a pretrained CNN model that focuses on maximizing the flow of information between the tiers of network in addition to keeping the training of DL easier by making short connections between all layers. In DenseNet121, every layer is connected to all layers below it, such that if there are four layers then the first layer is connected to second, third and fourth layers and the second layer is connected to the third and fourth layers and finally the third layer is connect to the fourth layer as illustrated in Fig. 2. It is worth mentioning that the model is called DenseNet121 since it consists of Dense Blocks of 121 layers.

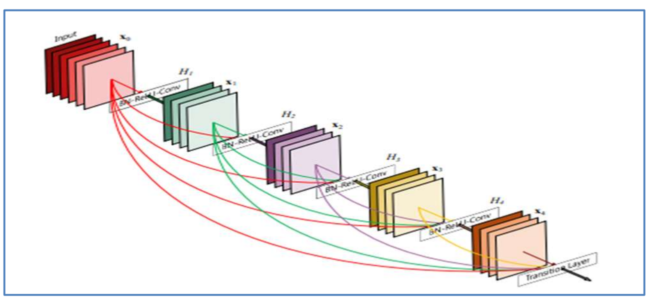


Fig. 2. DenseNet121 Architecture (Huang et al., 2017)

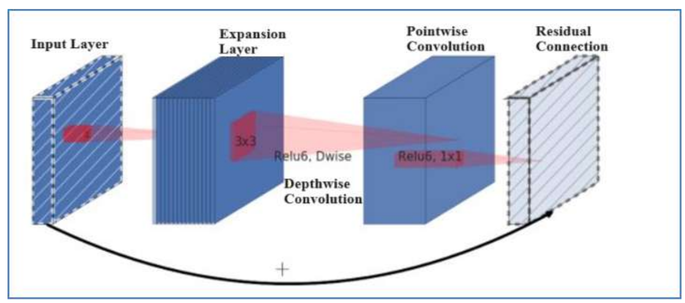


Fig. 3. MobileNetV2 architecture (Sandler et al., 2018)

3.1.2 Pretrained EfficientNetV2S

One variant of EfficientNet architecture is called EfficientNetV2S, which maintains high performance and at the same time makes sure that the efficiency of computation is emphasized, i.e., keeps balance between efficiency and performance. In Efficient-NetV2S, to keep high accuracy and at the same time reduce the overhead of computations, Fused- Mobile Inverted Bottleneck Convolution (MBConv) block is used in addition to using MBConv blocks. In Fused-MBConv, both pointwise and depthwise convolutions are combined into a single convolution. Furthermore, EfficientNetV2S can increase the performance by using compound scaling balance between the network's resolution, depth, and width.

As shown in Fig. 4, EfficientNetV2S architecture consists of 45 layers divided as follows: one Input layer, one 3 × 3 convolutional layer, ten Fused-MBConv layers, thirty MBConv, and one layer for Convolutional, Pooling, Fully Connected (FC), and Output layers. For more details, the first image is fed to the model through the input layer. After that, the 3 × 3 convolutional layer extracts the initial features from the image, followed by the Fused-MBConv1 3 × 3 convolutional layer. Furthermore, two separate Fused-MBConv4 3 × 3 followed, where each one of them consists of four layers. At the middle of EfficientNetV2S, after Fused-MBC layers there are three blocks of MBConv layers. The first block of MBConv consists of six 3 × 3 convolutional layers, the second block consists of nine MBConv 3 × 3 convolutional layers, and the last one consists of fifteen 3 × 3 convolutional layers. More complex features can be extracted using MBConv layers. In order to minimize the dimensionality, the kernel of the 1 × 1 convolutional layer is deployed after all MBConv layers. The last two layers are pooling and FC, where the average pooling layer is used in the pooling layer to reduce the dimensionality of features map into a single vector. Moreover, the FC layer makes the final classification based on the features extracted from previous layers.

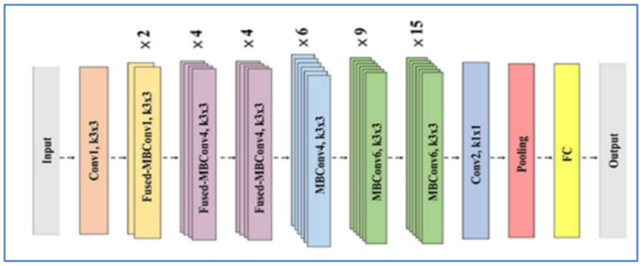


Fig. 4. EfficientNetV2S architecture (Tan, M., & Le, Q, 2021)

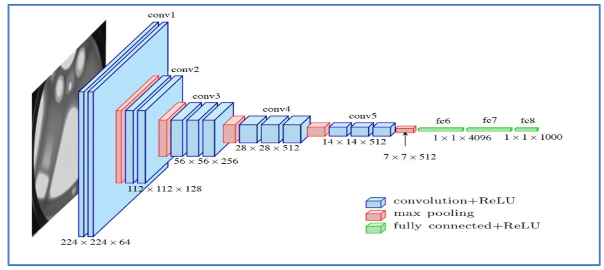


Fig. 5. VGG16 architecture (Nash et al., 2018)

3.1.3 Pretrained EfficientNetV2S

One variant of EfficientNet architecture is called EfficientNetV2S, which maintains high performance and at the same time makes sure that the efficiency of computation is emphasized, i.e., keeps balance between efficiency and performance. In EfficientNetV2S, to keep high accuracy and at the same time reduce the overhead of computations, Fused- Mobile Inverted Bottleneck Convolution (MBConv) block is used in addition to using MBConv blocks. In Fused-MBConv, both pointwise and depthwise convolutions are combined into a single convolution. Furthermore, EfficientNetV2S can increase the performance by using compound scaling balance between the network's resolution, depth, and width.

As shown in Fig. 4, EfficientNetV2S architecture consists of 45 layers divided as follows: one Input layer, one 3 × 3 convolutional layer, ten Fused-MBConv layers, thirty MBConv, and one layer for Convolutional, Pooling, Fully Connected (FC), and Output layers. For more details, the first image is fed to the model through the input layer. After that, 3 × 3 convolutional layer extracts the initial features from the image, followed by the Fused-MBConv1 3 × 3 convolutional layer. Furthermore, two separate Fused-MBConv4 3 × 3 followed, where each one of them consists of four layers. At the middle of EfficientNetV2S, after Fused-MBC layers there are three blocks of MBConv layers. The first block of MBConv consists of six 3 × 3 convolutional layers, the second block consists of nine MBConv 3 × 3 convolutional layers, and the last one consists of fifteen 3 × 3 convolutional layers. More complex features can be extracted using MBConv layers. In order to minimize the dimensionality, the kernel of the 1 × 1 convolutional layer is deployed after all MBConv layers. The last two layers are pooling and FC, where the average pooling layer is used in the pooling layer to reduce the dimensionality of features map into a single vector. Moreover, the FC layer makes the final classification based on the features extracted from previous layers.

3.1.4 Pretrained VGG16

VGG16 is one version of VGG-Net pretrained CNN models, which was presented in (Simonyan, 2014). VGG16 can classify images into one of 1000 classes, where it takes a 224 × 224 pixels color image as an input and returns a vector of size 1000 filled with the probability of belonging the image to each class. VGG16 architecture consists of 16 layers where 13 of them are convolutional layers and the rest 3 layers are fully connected as shown in Fig. 5.

3.2 Transfer Learning Used

Ensemble method of four pretrained CNN transfer learning models was used to detect fires. CNN pretrained models including DenseNet121, MobileNetV2, EfficientNetV2S, and VGG16 use Drones fire detection images training dataset, which are fed as input to four models. After training, the classifiers are evaluated using testing dataset. Once the classifiers produce the output, 8-D feature vector (i.e., two probabilities are generated from each individual and its softmax) is generated from the output probability. The number of probabilities is based on the number of classes, which are Fire and No-fire. In this research, ensemble majority voting and ensemble (sum) is used to produce the final decision of detecting the fire as illustrated in Figure 1. Each one of the four pretrained CNN models generates an output (predicted class) of the testing images, after that, the ensemble (majority voting) method is used to tally the votes for the Fire class and the No-fire class. Accordingly, the class that has the maximum number of votes will be the winner.

Additionally, in the ensemble (sum), the final decision of the class is determined by choosing the maximum likelihood of sum. In this case, the sum is calculated by adding the posterior probability outputs of each classifier t for each class j of test image I $P_{jt}(I)$. The ensemble sum equation is shown in Eq. (1), where T is the number of classifiers.

$$P(I) = \max_{j=1toc} \sum_{t=1}^{T=4} P_t^j(I) \quad (1)$$

3.2 Resources used

All the experiments are conducted using the Tesla GPU on Google Colaboratory or "Colab". In addition, the dataset is uploaded on the google drive where the code is written using Python programming language. Finally, TensorFlow, Keras API are used.

4. Dataset, experiments, results, and discussion

4.1 Dataset

The experiments are performed using DeepFire dataset (Khan & Khan, 2022b). DeepFire is a specialized balanced dataset created using UAV for detecting forest fires. It consists of 1900 colored images divided into two classes (Fire and No-Fire) where 1900 images are divided into training and testing dataset. Training dataset consists of 1520 images partitioned into two classes where each class consists of 760 images. On the other hand, testing data consists of 380 images divided into two equal size classes. In conclusion, the dataset consists of 950 fire images and 950 no-fire images where the 950 images are partitioned into 760 images for training and 190 images for testing in each class as shown in Table 1. A sample of utilized images are displayed in Fig. 6.

Table 1
Dataset Specifications

	Fire	No-fire	Total
Training	760	760	1520
Testing	190	190	380
Total	950	950	1900

To gain ideal classifiers and to enhance the training and testing process, the original dataset was subjected to several preprocessing steps including removing undesirable objects from the images such as firefighters and fire trucks. Further, all irrelevant parts were cropped from the images. It is worth noting that all the images had the same resolution and a size of 250 × 250 pixels.

Fig. 6. A sample of images

4.2 Experiment Setup and resources

The value of hyperparameters of the four pretrained models that are used to generate the proposed fire detection classifiers are the same. Such that, in all experiments, batch size value was 32, the learning rate was lr=0.000001 where the value is too small to reduce the speed of model learning. Furthermore, stochastic gradient descent (SGD) with momentum was used for training the models. Experiments were made using five values of epochs; 5, 10, 15, 20, and 25. In addition, the loss function was the cross entropy (CE), which estimates the distance between the probability vector of the ground truth table (T) of the one-hot-encoded and prediction likelihood vector (E). The equation of CE is illustrated in Eq. (2).

$$CE(E,T) = -\sum_{t=1}^{T} T_t \log E_t \tag{2}$$

Moreover, all utilized pretrained model contains dropout layer to prevent overfitting during training. In this research, dropout value was set to 0.3, which is the typical value in DL models.

4.3 Evaluation Criteria

Confusion matrix was used to calculate the evaluation criteria of the proposed classifiers. Confusion matrix values are filled after testing the proposed model using testing the dataset with respect to the number of images related to predicted and actual classes. The structure of confusion matrix presented in Figure 7 composed of True Positive (TP) and False Positive (FP). TP refers to the set of images that are classified as fire and contain fire, while FP refers to the set of images that are classified as fire and do not contain fire. On the other hand, True Negative (TN) refers to the set of images that are classified as nofire and contain nofire, while False Negative (FN) refers to the set of images that are classified as nofire and they do not contain nofire. By using confusion matrix, three evaluation measures are used in this research to evaluate the performance of the proposed classifiers including accuracy, sensitivity, and specificity.

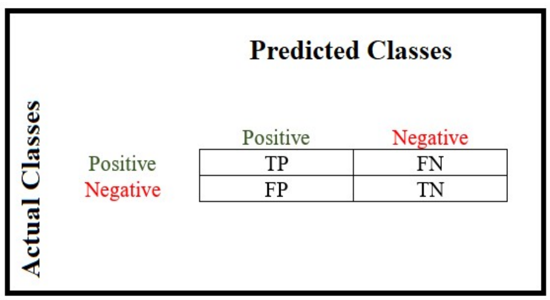


Fig. 7. Confusion Matrix

a) Average classification accuracy: Accuracy is calculated by aggregating the probability of correctly classifying the image into two classes. Where TP and TN mean that the images are correctly classified. Accuracy is measured using Eq. (3) as follows:

$$Acc = \frac{1}{M}\sum_{j=1}^{M} \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

where the number of independent runs is represented by M.

b) Average classification sensitivity: Sensitivity or recall are interchangeably used. It represents the proportion of predicted class; in this paper it is considered as the probability of predicting fire images correctly. The equation of calculating sensitivity is described in Eq. (4):

$$sensitivity = \frac{1}{M}\sum_{j=1}^{M} \frac{TP}{TP + FN} \tag{4}$$

As shown from Eq. (4), the value of sensitivity is between [0, 1] where 0 means the worst possible classification and 1 means the best classification. Note that the overall value is multiplied by 100% to gain percentage

c) Average classification Specificity: Unlike sensitivity, specificity calculates the percentage of the correctly classified negative classes. In fire detection problem, it is
the probability that an image not containing fire is correctly identified and classified as belonging to the 'Nofire' class.

Specificity is computed as shown Eq. (5):

$$Specificity = \frac{1}{M}\sum_{j=1}^{M}\frac{TN}{TN+FP} \tag{5}$$

5. Results and discussion

The experimental results retrieved from using four CNN TL models, ensemble majority voting, and ensemble sum over drone's forest fire detection images are discussed in this section. In this research, four pretrained CNN TL models are used including: DenseNet121, MobileNetV2, EfficientNetV2S, and VGG16 in addition to using ensemble majority voting and ensemble sum of the individual pretrained models. The generated models are trained using five values of epochs; 5, 10, 15, 20, and 25 to show the effect of using several values of epochs on accuracy of the predicted model. Moreover, the generated models are evaluated in terms of accuracy, sensitivity (recall) and specificity. All measurements are illustrated on Table 2 to Table 6 for various values of epochs 5, 10, 15, 20, and 25, respectively. The worst values of accuracy, sensitivity, and specificity are acquired when testing the EfficientNetV2S model over all epochs values. After EfficientNetV2S, the next worst values are achieved using VGG16.

The lowest value of accuracy was obtained when testing EfficientNetV2S using 5 epochs where the value was 70%. On the other hand, the highest value of 82% was achieved using 10 epochs. Moreover, the lowest value of accuracy of testing VGG16 was 81% using 5 epochs while the highest value achieved was 91% using 15 epochs. It can be concluded that using 5 epochs to train all models is not enough to get reasonable accuracy and other measures. Furthermore, in most experiments, using more than 20 epochs is unnecessary, as performance often deteriorates with 25 epochs in many models.

Additionally, it's worth noting that accuracy and sensitivity often yield the same values in most experiments. Accuracy measures the percentage of correct predictions, while sensitivity (recall) measures the proportion of actual fire cases that are correctly identified. There are several reasons why accuracy and recall might yield the same values: 1) balanced dataset where the number of fire and no-fire images is the same in either training or testing. 2) High performance, where most of the models predict the classes of images correctly. In this case, the values of FP and FN are low resulting in high accuracy and high recall. 3) Binary classification of a balanced dataset results in very similar values of accuracy and recall. Furthermore, experiments on drones' forest fire detection images using individual pretrained CNN models including DenseNet121, MobileNetV2, EfficientNetV2S, and VGG16 and ensemble sum and ensemble majority voting show that MobileNetV2 gained the highest accuracy and outperformed other models as shown in Figs. 8-27. The reason is related to the Inverted Residuals and Linear Bottlenecks architecture of MobileNetV2, which provides high performance and makes the model generalize well over several areas. Moreover, the architecture of MobileNetV2 makes it well-suited for real-time image applications in drones and less prone to overfitting. Additionally, the small size of the dataset and its features facilitate the extraction of features using MobileNetV2. The highest accuracy of 99.4% was achieved using 20 epochs as illustrated in Fig. 21(a). Initially, the training accuracy started at a low value, but after the second epoch, it began to increase significantly, indicating that the model is learning effectively. The same is true for testing accuracy, a rapid increase can be noticed leading to good generalization model.

The highest values of DenseNet121 of 98.8% was achieved using 20 and 25 epochs as displayed in Fig. 20(a) and Fig. 24(a). Training and testing started early and increased sharply to reach high accuracy. During accuracy increase, some fluctuations took place without adversely affecting the accuracy. Moreover, although training accuracy sometimes slightly exceeds testing accuracy, the overall performance of the model is strong, and it generalizes well. Also, regarding the DenseNet121 and MobileNetV2, there was no evidence of overfitting and both models converge early within a few epochs.

Two ensemble Learning methods are used including ensemble Sum and ensemble Voting to merge the result of the four CNN pretrained TL. The results using Ensemble Sum indicate that the accuracy is slightly lower than MobileNetV2, which achieves 98.9% with 15 and 20 epochs. However, it remains higher than that of all other models, as shown in Table 4 and Table 5. On the other hand, ensemble Voting achieved an accuracy of 96.9% using 15 epochs, which is slightly lower than MobileNetV2 and ensemble Sum as indicated in Table 4.

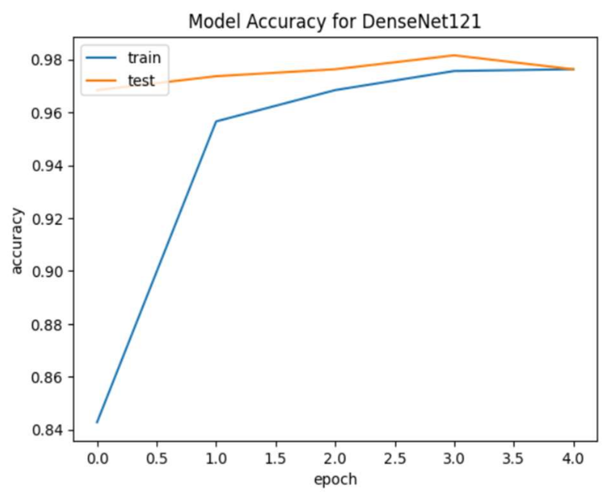
Overall, in this research, ensemble sum and ensemble majority voting of pretrained CNN transfer learning models provide significant accuracy, sensitivity, and specificity when generating fire detection classifiers. However, individual MobileNetV2 outperforms ensemble methods and individual DenseNet121, EfficientNetV2S, and VGG16. Additionally, the individual models' performance surpasses the ensemble performance when consistent.

5.1 Detailed Results

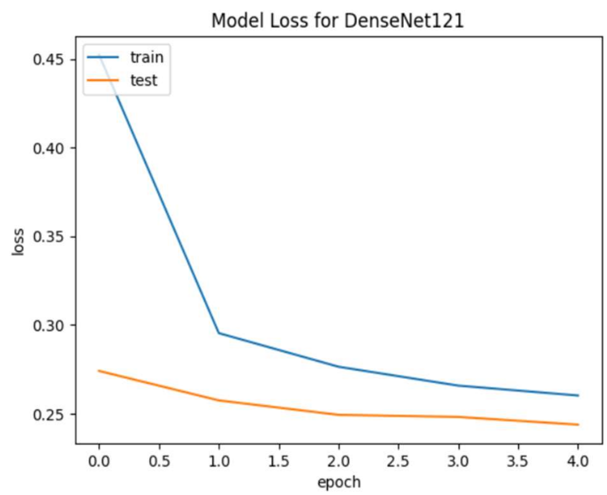
A set of experiments was conducted to compare accuracy, sensitivity, specificity, and ensemble of four CNN models when the epochs were changed from 5 to 25 with a step of 5. To ensure reproducibility of the results, the same experiment was repeated 10 times. Table 2 presents the results at 5 epochs.

Table 2
Accuracy, Sensitivity and Specificity of the Densenet121, MobileNetV2, EfficientNetV2S and Ensemble Models for Drones Forest Fire Detection Images at 5 epochs.

Model used	Accuracy		Sensitivity		Specificity	
	AVG	STD	AVG	STD	AVG	STD
DenseNet121	0.977	0.006	0.977	0.006	0.972	0.007
MobileNetV2	0.986	0.003	0.986	0.003	0.993	0.005
EfficientNetV2S	0.696	0.008	0.696	0.008	0.748	0.024
VGG16	0.813	0.005	0.813	0.005	0.803	0.094
Ensemble (Sum)	0.989	0.258	0.989	0.258	0.989	0.470
Ensemble (Majority voting)	0.937	1.58	0.937	1.58	0.995	0.333

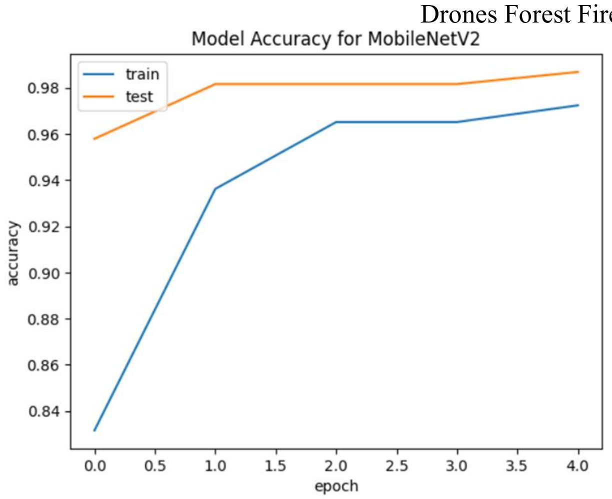


a)

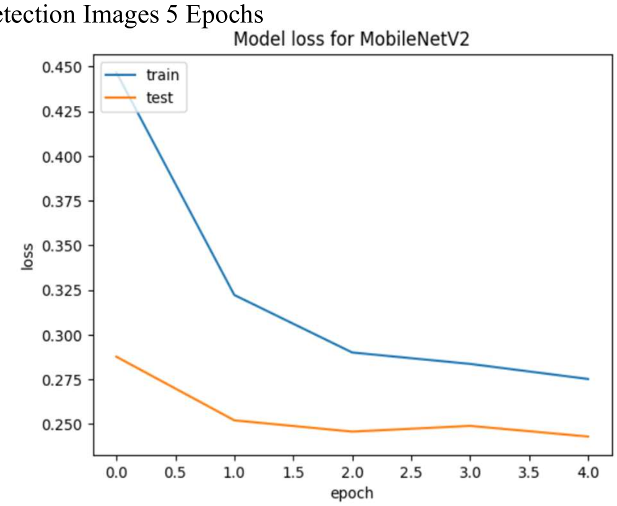


b)

Fig. 8. Learning curves for (a) training and testing accuracy, and (b) training and testing loss of the Densenet121 for Drones Forest Fire Detection Images 5 Epochs

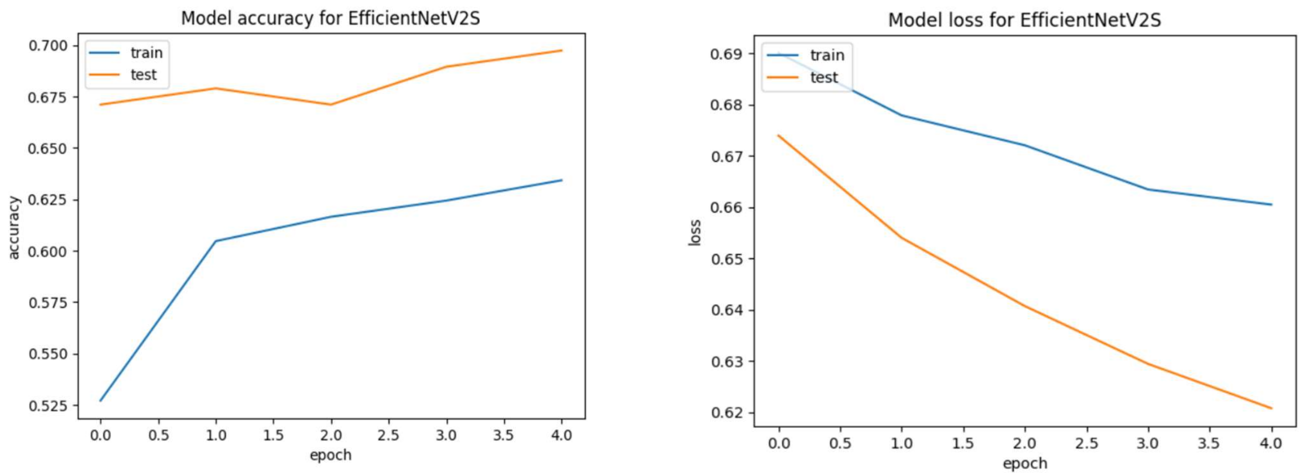


a)



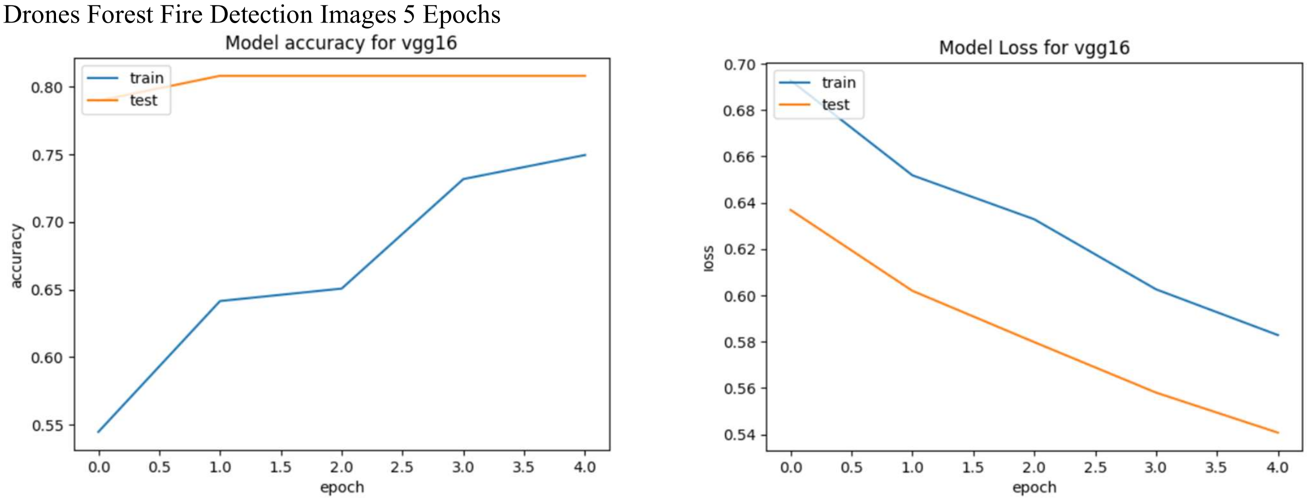
b)

Fig. 9. Learning curves for (a) training and testing accuracy, and (b) training and testing loss of the MobileNetV2 for Drones Forest Fire Detection Images 5 Epochs



a) b)

Fig. 10. Learning curves for (a) training and testing accuracy, and (b) training and testing loss of the EfficientNetV2S for Drones Forest Fire Detection Images 5 Epochs



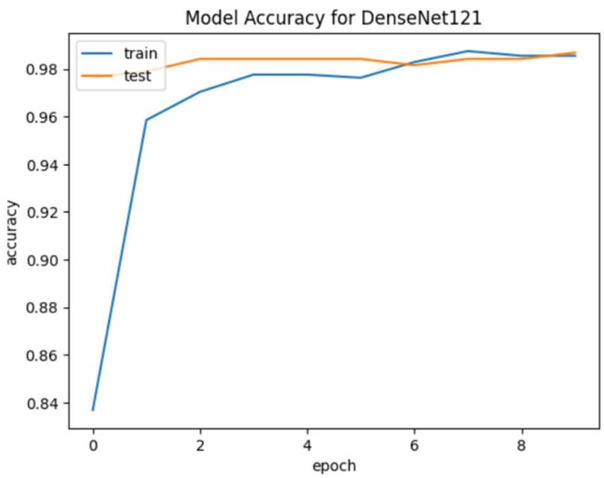
a) b)

Fig. 11. Learning curves for (a) training and testing accuracy, and (b) training and testing loss of the VGG16 for Drones Forest Fire Detection Images 5 Epochs

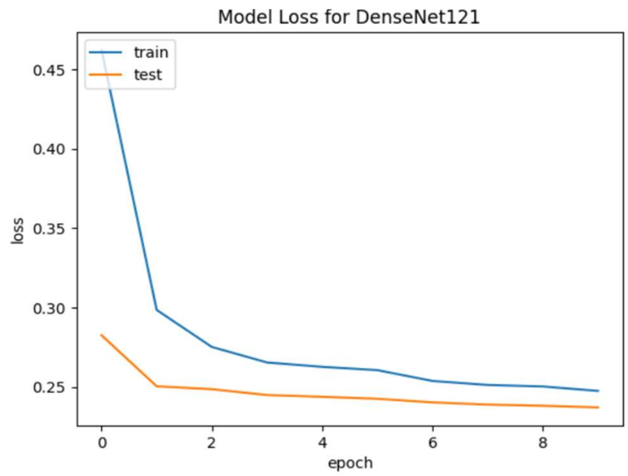
Over 5 epochs (rounds), MobileNetV2 registers an accuracy of 98.6%, sensitivity of 98.6% and specificity of 99.3%; while EfficientNetV2B has the lowest accuracy (69.6%), sensitivity (69.6%) and specificity (74.8%). The 'ensemble (Sum)' model is characterized by both high accuracy (98.9%) and sensitivity (98.9%), but there is significant variation between these two measures. On the other hand, ensemble Majority voting has a slightly lower accuracy at 93.7%, however, it has a higher specificity reading at 99.5% with variance being evident in this model as well. The same set of experiments was repeated over 10 epochs and results were recorded in Table 3.

Table 3
Accuracy, Sensitivity and Specificity of the Densenet121, MobileNetV2, EfficientNetV2S and Ensemble Models for Drones Forest Fire Detection Images at 10 epochs

Model used	Accuracy		Sensitivity		Specificity	
	AVG	STD	AVG	STD	AVG	STD
DenseNet121	0.983	0.004	0.983	0.004	0.983	0.005
MobileNetV2	0.991	0.002	0.991	0.002	0.997	0.004
EfficientNetV2S	0.724	0.008	0.724	0.008	0.796	0.015
VGG16	0.825	0.021	0.825	0.021	0.756	0.072
Ensemble (Sum)	0.986	0.387	0.986	0.387	0.986	0.537
Ensemble (Majority voting)	0.953	1.071	0.953	1.071	0.995	0.332

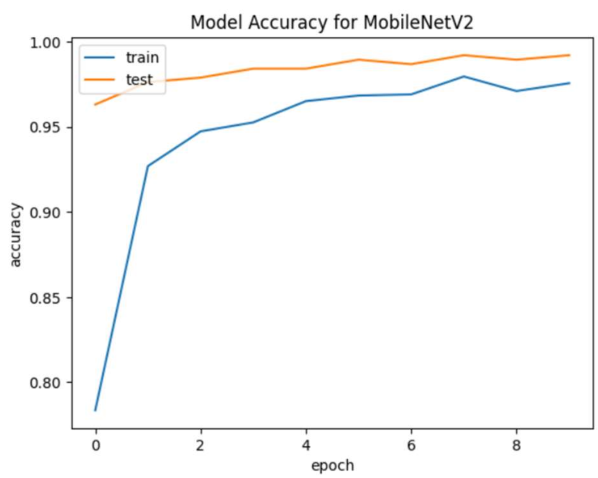


a)

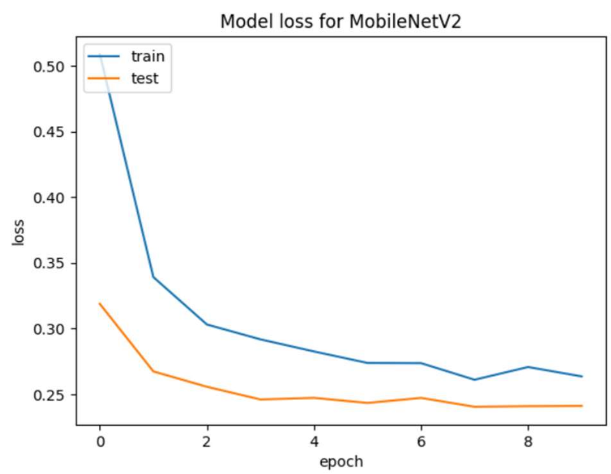


b)

Fig. 12. Learning curves for (a) training and testing accuracy, and (b) training and testing loss of the Densenet121 for Drones Forest Fire Detection Images 10 Epochs



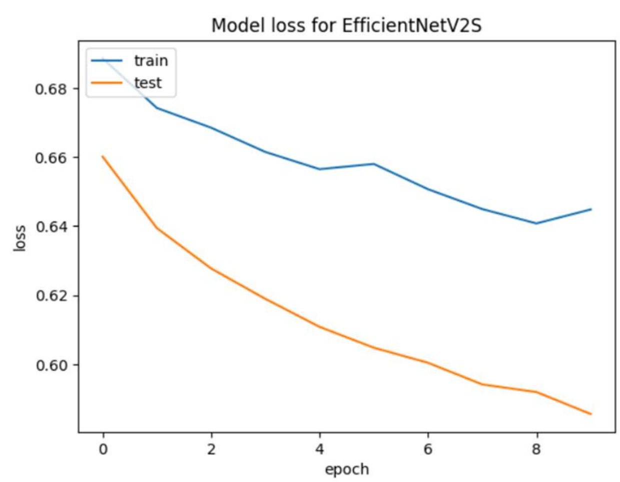
a)



b)

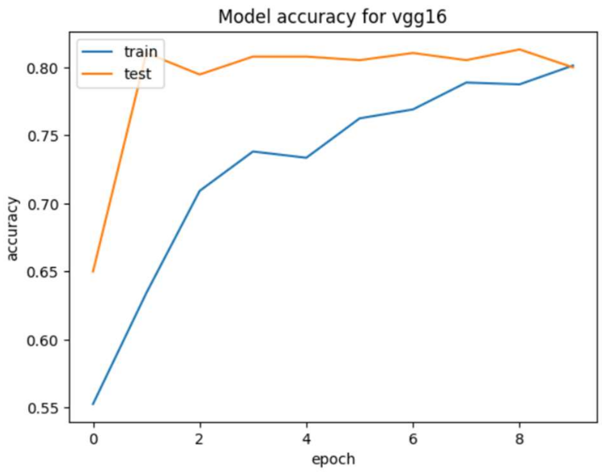
Fig. 13. Learning curves for (a) training and testing accuracy, and (b) training and testing loss of the MobileNetV2 for Drones Forest Fire Detection Images 10Epochs

a)

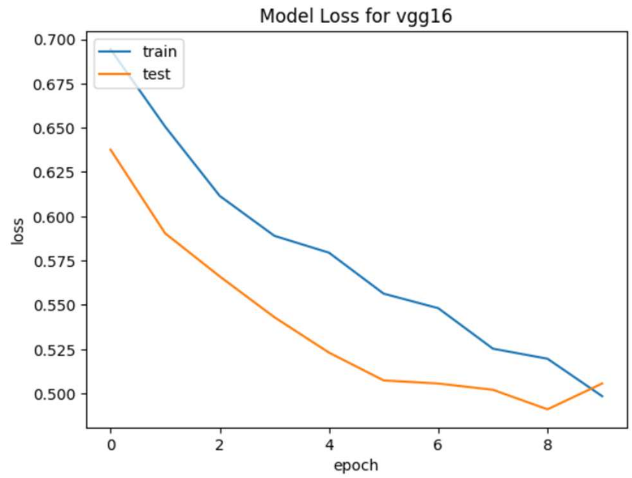


b)

Fig. 14. Learning curves for (a) training and testing accuracy, and (b) training and testing loss of the EfficientNetV2S for Drones Forest Fire Detection Images 10 Epochs



a)



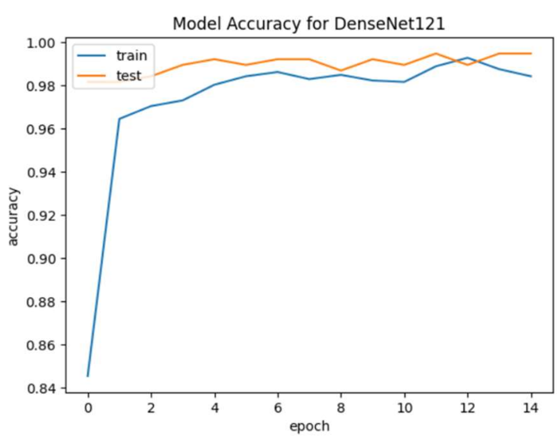
b)

Fig. 15. Learning curves for (a) training and testing accuracy, and (b) training and testing loss of the VGG16 for Drones Forest Fire Detection Images 10 Epochs

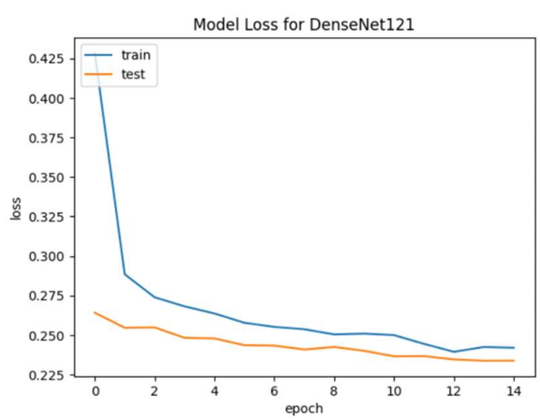
Over 10 epochs, MobileNetV2 achieved a 0.991 score in accuracy and sensitivity across 10 epochs, with a specificity of 0.997. EfficientNetV2S stays the least effective, showing an accuracy and sensitivity of 0.724. The ensemble (Sum) retained its strong performance scoring 0.986 in accuracy and sensitivity, although it shows some fluctuations. The ensemble (Majority voting) revealed a lower accuracy at 0.953 but a high specificity (0.995), with noticeable changes in its results. The same set of experiments was repeated over 15 epochs and results were recorded in Table 4.

Table 4
Accuracy, Sensitivity and Specificity of the Densenet121, MobileNetV2, EfficientNetV2S and Ensemble Models for Drones Forest Fire Detection Images at 15 epochs

Model used	Accuracy		Sensitivity		Specificity	
	AVG	STD	AVG	STD	AVG	STD
DenseNet121	0.987	0.005	0.987	0.005	0.988	0.005
MobileNetV2	0.993	0.0031	0.993	0.002	0.998	0.002
EfficientNetV2S	0.761	0.029	0.761	0.029	0.809	0.038
VGG16	0.908	0.011	0.908	0.01	0.8945	0.029
Ensemble (Sum)	0.989	0.105	0.989	0.105	0.991	0.211
Ensemble (Majority voting)	0.969	0.387	0.969	0.387	0.996	0.394

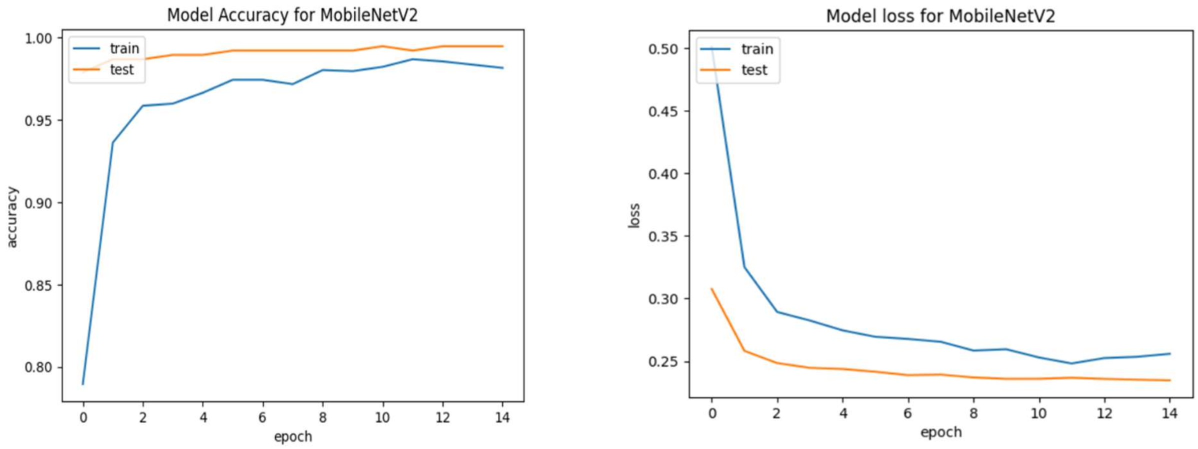


a)



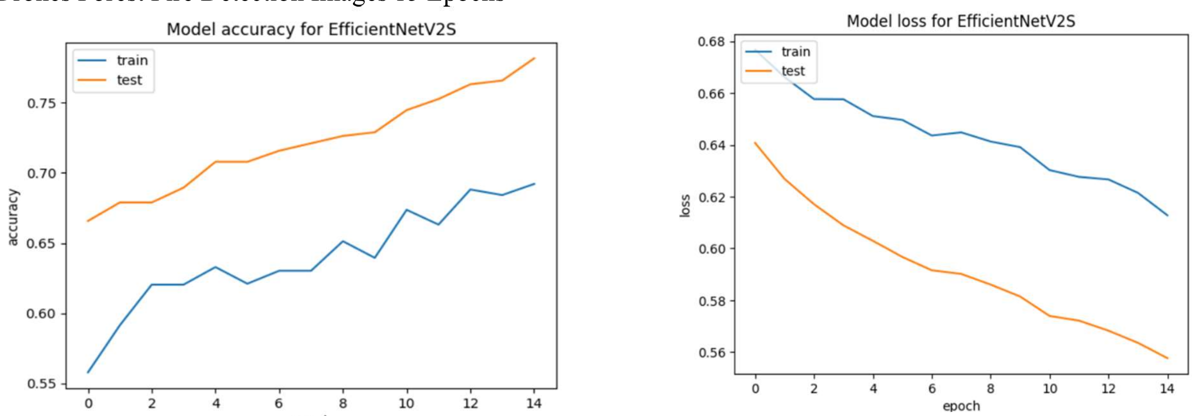
b)

Fig. 16. Learning curves for (a) training and testing accuracy, and (b) training and testing loss of the Densenet121 for Drones Forest Fire Detection Images 15 Epochs



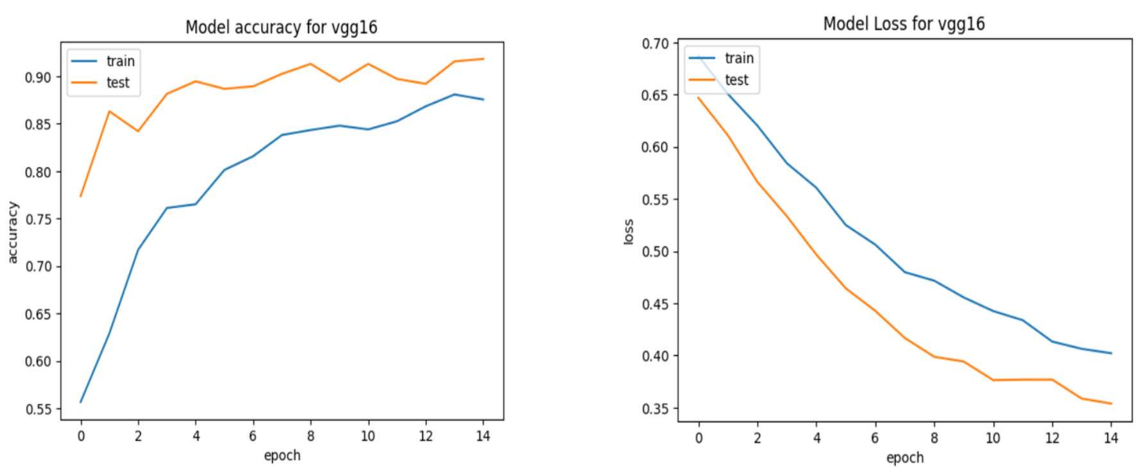
a) b)

Fig. 17. Learning curves for (a) training and testing accuracy, and (b) training and testing loss of the MobileNetV2 for Drones Forest Fire Detection Images 15 Epochs



a) b)

Fig. 18. Learning curves for (a) training and testing accuracy, and (b) training and testing loss of the EfficientNetV2S for Drones Forest Fire Detection Images 15 Epochs



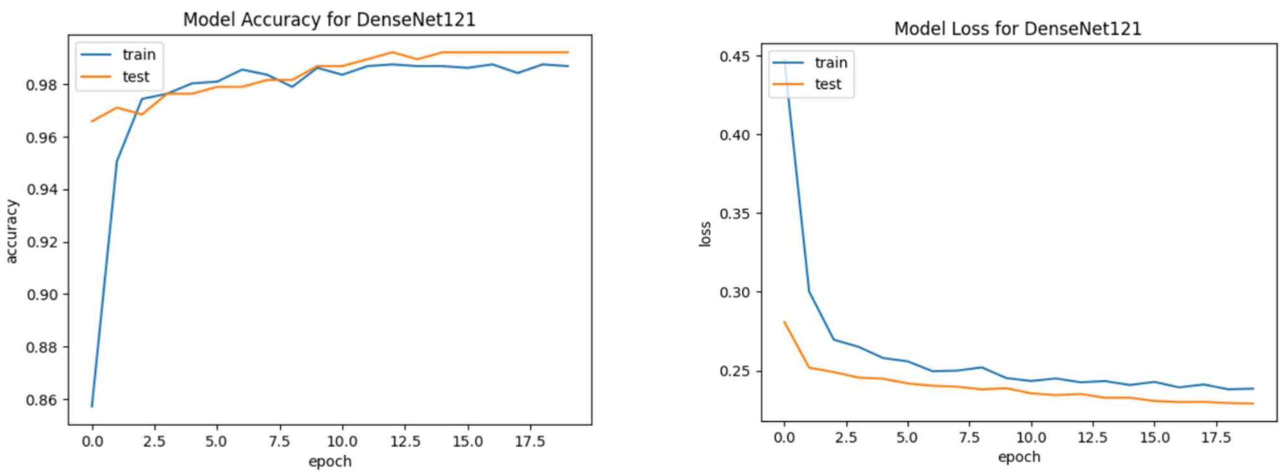
a) b)

Fig. 19. Learning curves for (a) training and testing accuracy, and (b) training and testing loss of the VGG16 for Drones Forest Fire Detection Images 15 Epochs

Over 15 epochs, MobileNetV2 revealed an accuracy of 0.993, sensitivity of 0.993, and specificity of 0.998. EfficientNetV2S still shows the worst performance with an accuracy of 0.761. The ensemble (Sum) model gives high accuracy and sensitivity (0.989) but with some variability, while the ensemble (Majority voting) showed an accuracy of 0.969 and high specificity (0.996), with noticeable variability. To maintain consistency, the set of experiments was repeated over 20 epochs, and the results are recorded in Table 5.

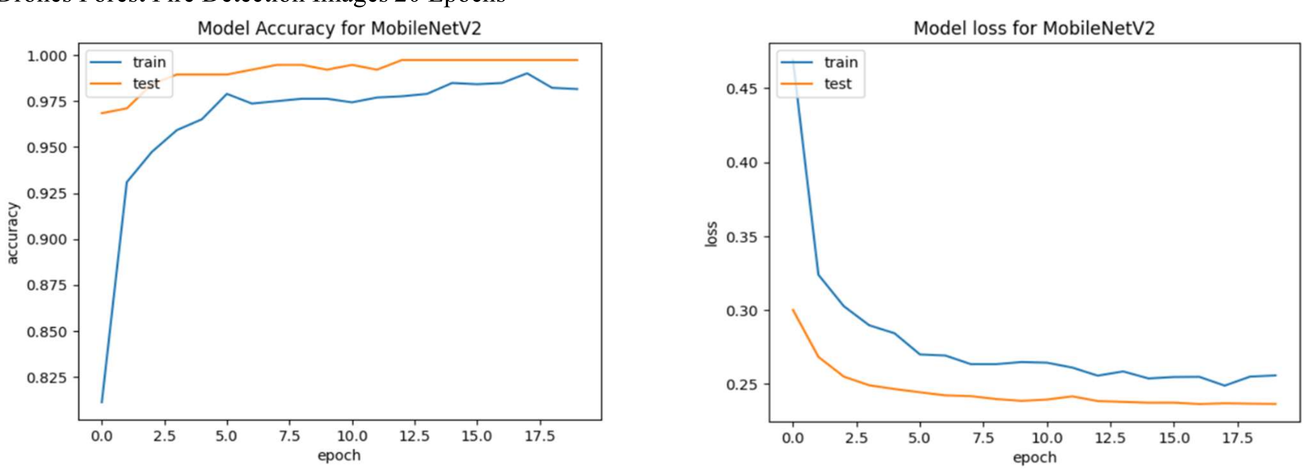
Table 5
Accuracy, Sensitivity and Specificity of the Densenet121, MobileNetV2, EfficientNetV2S and Ensemble Models for Drones Forest Fire Detection Images at 20 epochs

Model used	Accuracy		Sensitivity		Specificity	
	AVG	STD	AVG	STD	AVG	STD
DenseNet121	**0.988**	0.004	0.988	0.004	0.992	0.006
MobileNetV2	**0.994**	0.003	0.994	0.003	0.998	0.004
EfficientNetV2S	0.779	0.009	0.779	0.009	0.829	0.031
VGG16	0.862	0.021	0.862	0.021	0.829	0.083
Ensemble (Sum)	**0.989**	0.105	0.989	0.105	0.988	0.211
Ensemble (Majority voting)	0.958	0.349	0.958	0.349	0.995	0.330



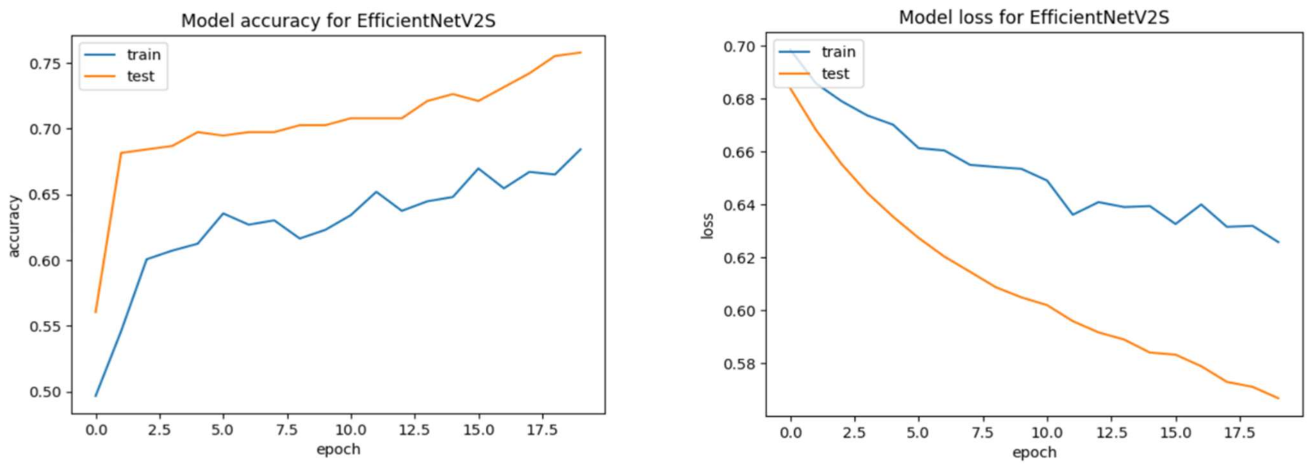
a) b)

Fig. 20. Learning curves for (a) training and testing accuracy, and (b) training and testing loss of the Densenet121 for Drones Forest Fire Detection Images 20 Epochs



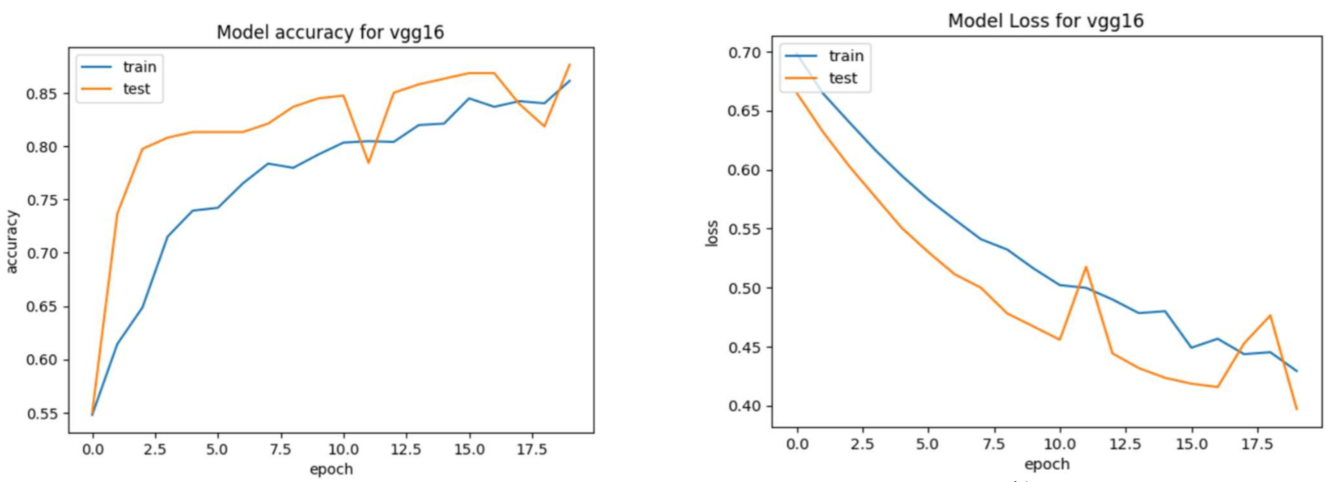
a) b)

Fig. 21. Learning curves for (a) training and testing accuracy, and (b) training and testing loss of the MobileNetV2 for Drones Forest Fire Detection Images 20 Epochs



a) b)

Fig. 22. Learning curves for (a) training and testing accuracy, and (b) training and testing loss of the EfficientNetV2S for Drones Forest Fire Detection Images 20 Epochs



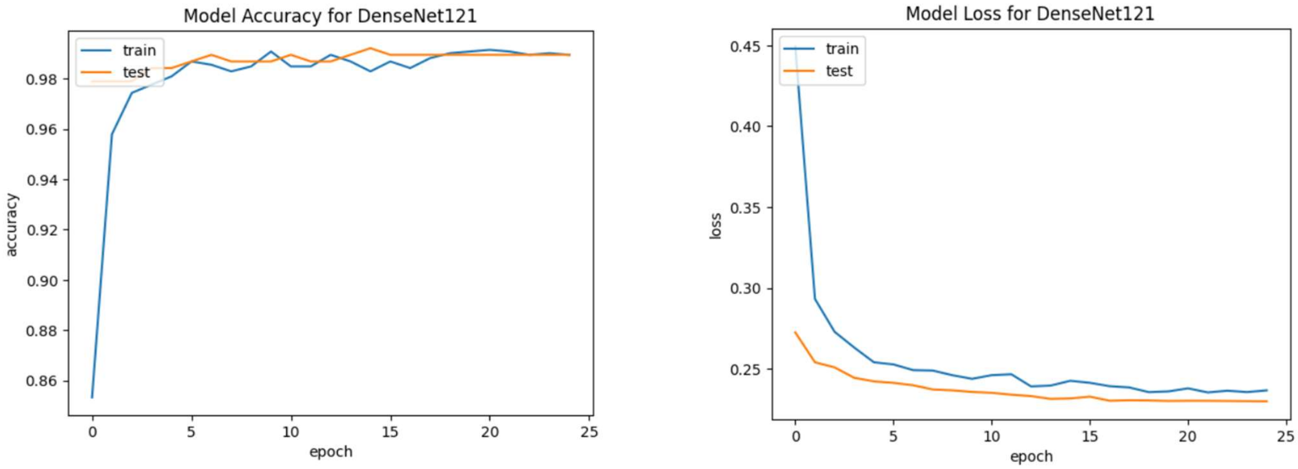
a) b)

Fig. 23. Learning curves for (a) training and testing accuracy, and (b) training and testing loss of the VGG16 for Drones Forest Fire Detection Images 20 Epochs

Over 20 epochs, MobileNetV2 revealed an accuracy of 0.994, sensitivity of 0.994, and specificity of 0.998. EfficientNetV2S performed less well with an accuracy of 0.779. For the ensemble, accuracy is 0.989 and sensitivity is 0.989, with good specificity at 0.988, but some variability. For the ensemble by majority vote, accuracy was 0.958, specificity was high at 0.995, and variability was considerable. To draw a conclusion on the recorded results, the same set of experiments was conducted lastly over 25 epochs and results were recorded in Table 6.

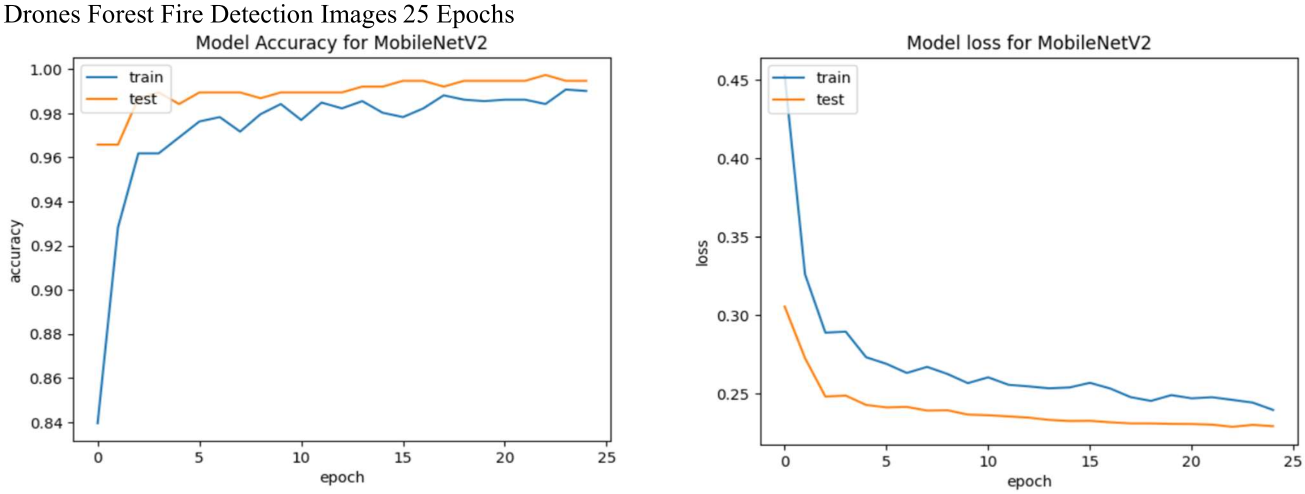
Table 6
Accuracy, Sensitivity and Specificity of the Densenet121, MobileNetV2, EfficientNetV2S and Ensemble Models for Drones Forest Fire Detection Images 25 Epochs

Model used	Accuracy		Sensitivity		Specificity	
	AVG	STD	AVG	STD	AVG	STD
DenseNet121	0.988	0.002	0.988	0.002	0.989	0.004
MobileNetV2	0.992	0.002	0.992	0.002	0.994	0.005
EfficientNetV2S	0.809	0.010	0.809	0.010	0.819	0.270
VGG16	0.869	0.012	0.869	0.012	0.828	0.014
Ensemble (Sum)	0.988	0.121	0.988	0.121	0.986	0.258
Ensemble (Majority voting)	0.968	0.704	0.968	0.704	0.944	0.437



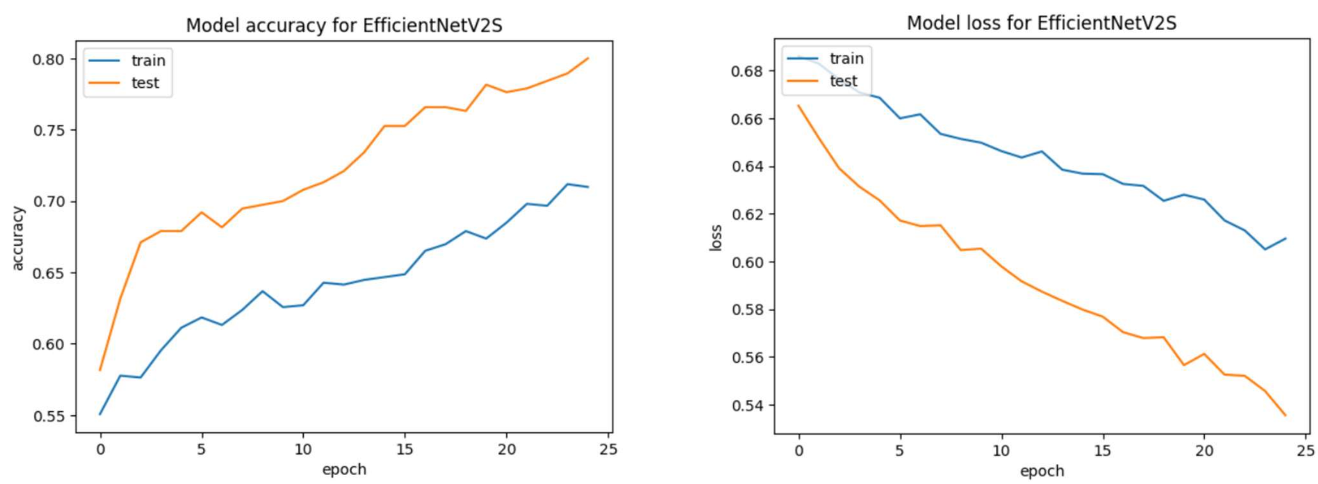
a) b)

Fig. 24. Learning curves for (a) training and testing accuracy, and (b) training and testing loss of the Densenet121 for Drones Forest Fire Detection Images 25 Epochs



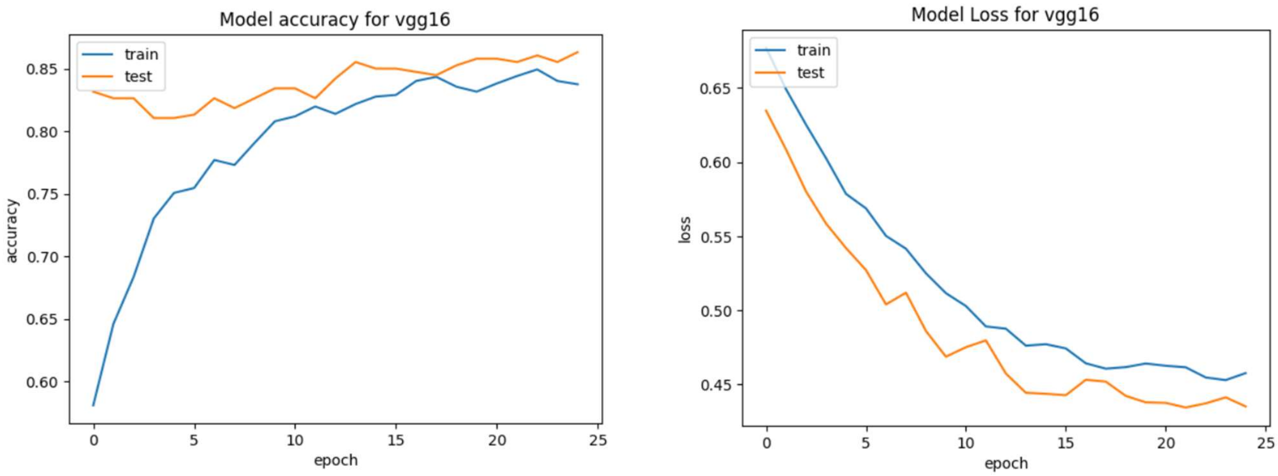
a) b)

Fig. 25. Learning curves for (a) training and testing accuracy, and (b) training and testing loss of the MobileNetV2 for Drones Forest Fire Detection Images 25 Epochs



a) b)

Fig. 26. Learning curves for (a) training and testing accuracy, and (b) training and testing loss of the EfficientNetV2S for Drones Forest Fire Detection Images 25 Epochs



a) b)

Fig. 27. Learning curves for (a) training and testing accuracy, and (b) training and testing loss of the VGG16 for Drones Forest Fire Detection Images 25 Epochs

Over 25 epochs, a very good performance was achieved with MobileNetV2: accuracy of 0.992, sensitivity of 0.992, and specificity of 0.994. EfficientNetV2S showed the worst performance in terms of accuracy, at 0.809. In the case of the ensemble Sum model, very high values of accuracy, 0.988, and sensitivity, 0.988, with good specificity, 0.986, but rather high variability, have been reached. The ensemble by majority vote had an accuracy of 0.968 and specificity of 0.944, thus showing some variability in performance.

6. Conclusion and future work

For the classification of drone-captured forest fire detection images, MobileNetV2 has shown the highest performance during all different epochs from 5 to 25, attaining peak accuracy, sensitivity, and specificity. These metrics improved from 98.6% accuracy, 98.6% sensitivity, and 99.3% specificity at 5 epochs, to 99.4% accuracy, 99.4% sensitivity, and 99.8% specificity at 20 epochs, and still strong but a little lower at 25 epochs, with 99.2% accuracy, 99.2% sensitivity, and 99.4% specificity. In contrast, EfficientNetV2S has an accuracy and sensitivity that ranges between 69.6% and 80.9%, while specificity is from 74.8% to 81.9%. For the ensemble Sum model, very good performance was obtained, with accuracy and sensitivity up to 98.9%, in addition to generally good specificity. It has high specificity, reaching 99.5%, but lower accuracy and sensitivity than MobileNetV2. It also demonstrates high variability in performance. In general, MobileNetV2 performed better than the other models in every epoch, while EfficientNetV2S was the worst, and the ensemble models were good but highly variable in their performance. To sum up, MobileNetV2 surpasses other models in accuracy, sensitivity, and specificity for all epochs, with DenseNet121 also doing well. EfficientNetV2S always has the worst performance. Ensemble methods give good results but can vary significantly.

Acknowledgement

This work is a part of the Smart.Eco Project no. 9046-255050-53 funded by the German Academic Exchange Service (DAAD).

References

Aswini, N., & Uma, S. V. (2021, September). Drone Object Detection Using Deep Learning Algorithms. In *2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA)* (pp. 1187-1192). IEEE.

Bahhar, C., Ksibi, A., Ayadi, M., Jamjoom, M. M., Ullah, Z., Soufiene, B. O., & Sakli, H. (2023). Wildfire and smoke detection using staged YOLO model and ensemble CNN. *Electronics*, *12*(1), 228.

Bhatnagar, S., Gill, L., & Ghosh, B. (2020). Drone image segmentation using machine and deep learning for mapping raised bog vegetation communities. *Remote Sensing*, *12*(16), 2602.

Canadian Wildland Fire Information System, Natural Resources Canada. https://cwfis.cfs.nrcan.gc.ca/report (accessed on 24 August 2024).

Choutri, K., Lagha, M., Meshoul, S., Batouche, M., Bouzidi, F., & Charef, W. (2023). Fire detection and geo-localization using uav's aerial images and yolo-based models. *Applied Sciences, 13*(20), 11548.

Davis, M., & Shekaramiz, M. (2023). Desert/forest fire detection using machine/deep learning techniques. *Fire, 6*(11), 418.

Ghali, R., Akhloufi, M. A., & Mseddi, W. S. (2022). Deep learning and transformer approaches for UAV-based wildfire detection and segmentation. *Sensors, 22*(5), 1977, 10.3390/s22051977.

Guede-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.

Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4700-4708).

Ibraheem, M. K. I., Mohamed, M. B., & Fakhfakh, A. (2024). Forest Defender Fusion System for Early Detection of Forest Fires. *Computers*, *13*(2), 36.

Idroes, G. M., Maulana, A., Suhendra, R., Lala, A., Karma, T., Kusumo, F., ... & Noviandy, T. R. (2023). TeutongNet: A fine-tuned deep learning model for improved forest fire detection. *Leuser Journal of Environmental Studies*, *1*(1), 1-8.

Jonnalagadda, A. V., & Hashim, H. A. (2024). SegNet: A segmented deep learning based Convolutional Neural Network approach for drones wildfire detection. *Remote Sensing Applications: Society and Environment, 34*, 101181.

Khan, A., Hassan, B., Khan, S., Ahmed, R., & Abuassba, A. (2022a). DeepFire: A novel dataset and deep transfer learning benchmark for forest fire detection. *Mobile Information Systems, 2022*(1), 5358359.

Khan, S., & Khan, A. (2022b). Ffirenet: Deep learning based forest fire classification and detection in smart cities. *Symmetry*, *14*(10), 2155.

Lee, W., Kim, S., Lee, Y. T., Lee, H. W., & Choi, M. (2017, January). Deep neural networks for wild fire detection with unmanned aerial vehicle. In *2017 IEEE international conference on consumer electronics (ICCE)* (pp. 252-253). IEEE.

Manoj, S., & Valliyammai, C. (2023). Drone network for early warning of forest fire and dynamic fire quenching plan generation. *EURASIP Journal on Wireless Communications and Networking, 2023*(1), 112.

Mousavi, S., & Ilanloo, A. (2023). Nature inspired firefighter assistant by unmanned aerial vehicle (UAV) data. *Journal of Future Sustainability, 3*(3), 143-166.

Muhammad, S. S., & Alrikabi, J. M. (2024). Fire Detection by using DenseNet 201 algorithm and Surveillance Cameras images. *Journal of Al-Qadisiyah for Computer Science and Mathematics, 16*(1), 81-91.

Nash, W., Drummond, T., & Birbilis, N. (2018). A review of deep learning in the study of materials degradation. *npj Materials Degradation, 2*(1), 37.

National Interagency Fire Center (NIFC), (2022). https://www.nifc.gov/fire-information/nfn, (Access on 24 August 2024).

Oom, D., & Pereira, J. M. (2013). Exploratory spatial data analysis of global MODIS active fire data. *International Journal of Applied Earth Observation and Geoinformation, 21*, 326-340.

Osco, L. P., Junior, J. M., Ramos, A. P. M., de Castro Jorge, L. A., Fatholahi, S. N., de Andrade Silva, J., ... & Li, J. (2021). A review on deep learning in UAV remote sensing. *International Journal of Applied Earth Observation and Geoinformation, 102,* 102456.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4510-4520).

Seydi, S. T., Saeidi, V., Kalantar, B., Ueda, N., & Halin, A. A. (2022). Fire-Net: A Deep Learning Framework for Active Forest Fire Detection. *Journal of Sensors, 2022*(1), 8044390.

Shamta, I., & Demir, B. E. (2024). Development of a deep learning-based surveillance system for forest fire detection and monitoring using UAV. *Plos one, 19*(3), e0299058.

Simonyan, K. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

Supriya, Y., & Gadekallu, T. R. (2023). Particle swarm-based federated learning approach for early detection of forest fires. *Sustainability, 15*(2), 964.

Tan, M., & Le, Q. (2021, July). Efficientnetv2: Smaller models and faster training. In International conference on machine learning (pp. 10096-10106). PMLR.

Tsalera, E., Papadakis, A., Voyiatzis, I., & Samarakou, M. (2023). CNN-based, contextualized, real-time fire detection in computational resource-constrained environments. *Energy Reports, 9*, 247-257.

Wildfires, World Health Organization. Available online: https://www.who.int/health-topics/wildfires, (Access on 24 August 2024).

Xiao, S. (2023). Convolutional neural network for UAV image processing and navigation in tree plantations based on deep learning. *Nonlinear Engineering, 12*(1), 20220299.

Zheng, S., Zou, X., Gao, P., Zhang, Q., Hu, F., Zhou, Y., ... & Chen, S. (2024). A forest fire recognition method based on modified deep CNN model. *Forests, 15*(1), 111.