Wireless Personal Communications (2023) 130:1085–1105
https://doi.org/10.1007/s11277-023-10321-7

Using Neural Networks to Detect Fire from Overhead Images

Lukas Kurasinski[1] · Jason Tan[1] · Reza Malekian[1]

Accepted: 15 February 2023 / Published online: 14 March 2023

Abstract
The use of artificial intelligence (AI) is increasing in our everyday applications. One emerging field within AI is image recognition. Research that has been devoted to predicting fires involves predicting its behaviour. That is, how the fire will spread based on environmental key factors such as moisture, weather condition, and human presence. The result of correctly predicting fire spread can help firefighters to minimise the damage, deciding on possible actions, as well as allocating personnel effectively in potentially fire prone areas to extinguish fires quickly. Using neural networks (NN) for active fire detection has proven to be exceptional in classifying smoke and being able to separate it from similar patterns such as clouds, ground, dust, and ocean. Recent advances in fire detection using NN has proved that aerial imagery including drones as well as satellites has provided great results in detecting and classifying fires. These systems are computationally heavy and require a tremendous amount of data. A NN model is inextricably linked to the dataset on which it is trained. The cornerstone of this study is based on the data dependencieds of these models. The model herein is trained on two separate datasets and tested on three dataset in total in order to investigate the data dependency. When validating the model on their own datasets the model reached an accuracy of 92% respectively 99%. In comparison to previous work where an accuracy of 94% was reached. During evaluation of separate datasets, the model performed around the 60% range in 5 out of 6 cases, with the outlier of 29% in one of the cases.

Keywords Neural networks · Fire detection · Datasets · Accuracy

✉ Reza Malekian
reza.malekian@mau.se

Lukas Kurasinski
lukasz.kurasinski@gmail.com

Jason Tan
jason.tbz@hotmail.com

[1] Department of Computer Science and Media Technology, Malmö University, 205 06 Malmö, Sweden

1 Introduction

The use of artificial intelligence (AI) is increasing in our everyday applications. One emerging field within AI is image recognition, where the system can detect and recognise the contents of an image. A lot of research that has been devoted to predicting fires involves predicting its behaviour. That is, how the fire will spread based on environmental key factors such as moisture, weather condition, and human presence. The result of correctly predicting fire spread can help firefighters to extinguish it faster, control it to minimise the damage, deciding on possible actions, as well as allocating personnel effectively [1]. Furthermore, forest fires may not always be caused by specific weather conditions but also through deforestation for agricultural land. Recent research on fire detection for already-burning flames focuses on identifying the fire early to prevent it from spreading further [1], as well as helping first responders contain it. Fire detection revolves around finding proper methods of identifying fires in an image. Satellites that orbit our planet are equipped with different imagery sensors, it is common that a specific sensor has a designated method corresponding to the system [2].

Utilising neural networks (NN) for active fire detection has proved to be exceptional in classifying smoke [2, 3] and being able to separate it from similar patterns such as clouds, ground, dust, and ocean. Despite the promising results of using neural networks, deep learning is still a relatively young field where larger datasets are uncommon [2]. Moreover, the datasets are often strongly related to a specific model, resulting in a less robust system when using different models. Larger neural networks such as AlexNet and ResNet often produce accuracies around the 80% mark, other networks like GoogLeNet can reach accuracies around 95%. Training these architectures, however, is often computationally heavy [4]. In Sect. 2 the background of the research is explained. Section 3 describes the previous work related to fire detection and their corresponding methods and results. In Sect.4 the motivation behind the study is brought forth along with the research questions. In 5 the applied methodology of the research is explained in finer detail. The results and are presented in Sect. 6, furthermore, results of the schema is analysed and evaluated in 7. Section 8 covers the discussion of the results. Finally, Sect. 9 concludes the paper and future research is proposed.

2 Background

Traditional fire detection and monitoring comprised mostly of watchtowers which require personnel to be on the constant lookout in fire-prone areas. A fire lookout is required to stand guard and report smoke or fire, which are then relayed to emergency services [5]. Other approaches in fire management involve aircrafts such as helicopters to monitor the spread. Further research involves the Internet of Things (IoT) devices such as wireless sensors which would function as an alarm when a fire had been ignited [6]. These devices would notify emergency services when smoke was present. Albeit, such alarm systems require practical testing. The use of satellite images in fire assessment is not uncommon. These images, however, require a human inspection as the images suffer from low-resolution [5]. The aforementioned systems can detect fire with some limitations in, for example, being static, only covering specific areas and having slow response times [7].

Table 1 Segmentation results from [5]

Dataset	Precision (%)	Recall (%)	AUC (%)	F1-Score (%)	Sensitivity (%)	Specificity (%)	IOU (%)
Image Segmentation	91.99	83.88	99.85	87.75	83.12	99.96	78.17

Table 2 Results achieved from [5]

	Performance	
Dataset	Loss	Accuracy (%)
Test set	0.7414	76.23
Validation set	0.1506	94.31
Training set	0.0857	96.79

Recently, advances in fire detection using NN has proved that aerial imagery including drones as well as satellites has provided great results in detecting and classifying fires [4]. These systems are computationally heavy and require a tremendous amount of data. Commonly, a neural network model is closely related to the dataset it is being trained on. Thereby, highly data-dependent. The data dependencies of these models have formed the foundation of this work and future work of previous research [2].

3 Related Work

In this section previous work on the topic of fire detection and the results achieved from the corresponding work are summarised. Shamsoshoara et al. in [5] uses aerial imagery and monitoring systems helping first responders to mitigate fire quicker. With accurate data and proper fire detection models, the fire's behaviour can be predicted thereby helping first responders contain the spread [5].

3.1 Aerial Imagery Pile Burn Detection Using Deep Learning: The FLAME Dataset

The scope of this project is to study the dependencies of fire detection within images. In [5] Shamsoshoara et al. detects forest fires in videos and images. Despite this, the techniques used in the work is deemed reasonable because the authors are using segmentation masking as well as image classification without segmentation for fire detection. The authors describes it as"*to accurately localize and extract the fire regions from the background*". Using image segmentation Shamsoshoara et al. got a precision of 91.99%, recall of 83.88%, and an F1-score of 87.75%. The results in further detail can be seen in Table 1. Another approach was to use thermal images, which could also be very helpful in fire detection.

In terms of fire classification, the Xception network is used, as it is a binary classification. The model classifies images according to one of two classes, those showing an image of a fire and those not showing an image of a fire. The network is trained on the FLAME dataset 5.5.1, which is split into 80% training and 20% validation. The authors perform augmentation of the images to create new frames and prevent bias due to the unbalanced number of images in the classes [5]. The model had a training accuracy of 96.79%,

validation accuracy of 94.31%, and finally, a testing accuracy of 76.23%. The results can be seen in Table 2.

3.2 Active Fire Detection Landsat-8 Imagery: A Large-Scale Dataset and a Deep-Learning Study

Pereira et al. [2] used Landsat-8 images and produced excellent results with the help of NN. However, the issue regarding proper dataset and model dependencies on specific datasets were also highlighted. In the work, three variations of the U-Net is being used, the standard U-Net, an alternative U-Net that replaces the input layer with a 3-channel image, and finally, a lighter version of the U-Net architecture. The light version reduces the number of layers by a factor of 4, i.e. the first layer in the classic U-Net consists of 64 layers, in the light version this would correspond to 16 layers. Furthermore, Pereira et al. use different segmentation techniques, in general, the different segmentation techniques proposed in the work recognise the fire presence on a similar level. Although, on a pixel level, there seems to be some difference in sensitivity. The results from [2] can be seen in Table 3. The results using U-Net architecture ranged from 80-90% in precision. Recall ranging from 86–99% recall and an F1-Score of 80–90%.

3.3 Wildfire Segmentation Using Deep Vision Transformers

Similarly to the aforementioned articles, Ghali et al. [7] use a deep learning-based approach consisting of segmentation masks. According to the authors, their work consists of the first study to use Transformers as a part of a forest fire task. Their method is a hybrid CNN-Transformers model, the model proposed has had state-of-the-art performance within the medical field. The experimental results showed an F1-score of 96–97.7%, as the accuracy, precision, and recall is not included it can not be compared to the other work presented

Table 3 Results from [2]

Mask		CNN Architecture	P	R	IoU	F
Schroeder et al.		U-Net(10c)	86.8	89.7	78.9	88.2
		U-Net(3c)	89.8	88.8	80.7	89.3
		U-Net-Light(3c)	90.8	86.1	79.2	88.4
Murphy et al.		U-Net(10c)	93.6	92.5	87.0	93.0
		U-Net(3c)	89.1	97.6	87.2	93.2
		U-Net-Light(3c)	92.6	95.1	88.4	93.8
Kumar-Roy		U-Net(10c)	84.6	94.1	80.3	89.1
		U-Net(3c)	84.2	90.6	77.5	87.3
		U-Net-Light(3c)	76.8	93.2	72.7	84.2
Intersection	Schroeder et al.	U-Net(10c)	84.4	99.7	84.2	91.4
	Murphy et al.	U-Net(3c)	93.4	92.4	86.7	92.9
	Kumar-Roy	U-Net-Light(3c)	87.4	97.3	85.4	92.1
Voting	Schroeder et al.	U-Net(10c)	92.9	95.5	89.0	94.2
	Murphy et al.	U-Net(3c)	91.9	95.3	87.9	93.6
	Kumar-Roy	U-Net-Light(3c)	90.2	96.5	87.3	93.2

Table 4 Results from [7]

Model	Back bone	Input Resolution	Fi-Score (%)
TransUNet	Res50-ViT	224*224	97.5
TransUNet	Res50-ViT	512*512	97.7
TransUNet	ViT	224*224	94.1
TransUNet	ViT	512*512	94.8
MedT	simple CNN-Transformer	224*224	95.5
MedT	simple CNN-Transformer	256*256	96.0

above. The results can be seen in Table 4. In the Table, the two different frameworks' results are shown. The input resolution of the image and their F1-score is shown. The TransUNet got an F1-score of 97.7% and the MedT model got an F1-score of 96.0%. Both of the models can accurately minimise classification error in fire images. The segmentation algorithm managed to perform better than manual annotation, being able to correctly separate fire from background under special conditions. It was able to distinguish fire and background when the image consisted of smoke and in different weather conditions. Similarly to other work, it was able to locate smaller fire patches [7]. The downside to the large model is the long training time and being computationally heavy.

3.4 Attention Based CNN Model for Fire Detection and Localization in Real-World Images

In [3] Majid et al. propose an attention-based convolutional neural network. Based on state-of-the-art models' performances, the model that performed the best was chosen. In this case, the authors chose the EfficientNetB0 model which had an overall better performance than the comparison models. In Fig. 1 the models' performances are summarised. To achieve better performances the Majid et al. utilised transfer learning of the pre-trained model. Using EfficientNetB0 was the most ideal choice since it was significantly more efficient and also lightweight [3]. The best results were obtained over 20 epochs, as well as adding slight modifications to the architecture such as adding a dropout of 0.2 between the dense layers.

During an evaluation of the model, the authors calculate precision, recall, f-score, and accuracy. Over 20 epochs, the model achieved 95.4% accuracy, 91.77% precision, 97.61% recall and, 94.76% f-score. The model performed the best in accuracy and f-score in comparison to GoogLeNet, VGG16, and ResNet50. However, coming in third place behind

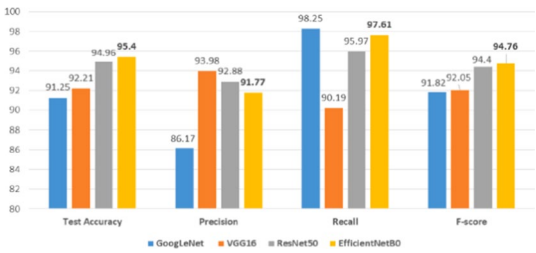

Fig. 1 Results from model comparison in [3]

VGG16 and ResNet in precision metric and second place behind GoogLeNet in the recall. The EfficientNetB0 model was chosen since it had far fewer parameters and still performed reasonably well according to the authors [3].

A common factor in all of the aforementioned work is the lack of public datasets. Datasets are either private, require special access, removed, or had to be purchased. Moreover, in [2] Pereira et al. stress the dependency of the dataset corresponding to the model's performance.

4 Goals

The use of models which are dependent on a specific dataset is a common issue in research. The majority of the previous work's positive result is believed to be closely related to either the model used or the data at hand. Therefore, this work aims to study whether a given model is dependent on the dataset it is trained on. For this reason, as stated in [2], the work herein involves investigating the hypothesis that a model trained on specific sample data can perform similarly when applied to another dataset. By evaluating an ML model's performance on several datasets, a deeper understanding of the relationship between these two elements is expected. The primary focus of this work is on the dataset's influence on the model's performance.

4.1 Research Questions

Based on the aim of the research, the following research questions are formulated:

1) Are models dependent on specific datasets?
2) Can a model trained on one dataset be successfully used with different dataset?

    a) What makes a dataset more effective?

5 Method

The use of a convolutional neural network in image classification has gained attention in previous years. With the help of GPUs, the processing capabilities it allows has improved, parallel computations and improved techniques have led to better models. Previous architectures and techniques involved RGB channels to detect objects within images. The RGB method compares pixels in the image, in the case of fire detection, the sun and sun-rays can affect the outcome of the classification [2, 5]. Common solutions to fire detection include models such as U-Net [8] and variations of U-Net, which require segmentation as well as LeNet [9], AlexNet [10], and ResNet [11]. However, the LeNet, AlexNet, and ResNet models provide significant loss in accuracy (20% less) [4]. In this case, the Xception model was considered from [5]. This was due to the fact that scarcity of datasets that include masks for segmentation during training.

The work herein has an experimental approach aimed towards gaining insight into a model's dependency on the data used for training and validation. In the experimental environment, the model is trained on the FLAMES dataset from IEEE Dataport. Furthermore,

validating the results from [5] which can be seen in Table 2. The model that is created is thereafter studied closer when predicting other data points as well as trained on newer datasets. By recreating the initial model, a clear picture of performance can be created. Furthermore, study how a trained model performs on a different dataset.

5.1 Xception Architecture

Commonly to neural networks, the Xception model is constructed by three main sections, the input layer, the hidden layers, and the output layer. The input layer receives an image of a given size, in this case, the image is scaled to $254 \times 254 \times 3$ [5]. In Fig. 2 the Xception architecture is visualised.

Since the task is to determine whether the image contains fire or not the output function returns the probability of the image containing a fire or no fire respectively. To calculate the probability of the image containing a fire, the activation function in the output layer is a Sigmoid function [5]. The sigmoid function ranges from 0-1 based on the probability of the image belonging to each class. For each epoch, a model version for the corresponding epoch is saved. In the case of 40 epochs, 40 models will be saved in order to evaluate which model performed best and thereafter, use that model for classification.

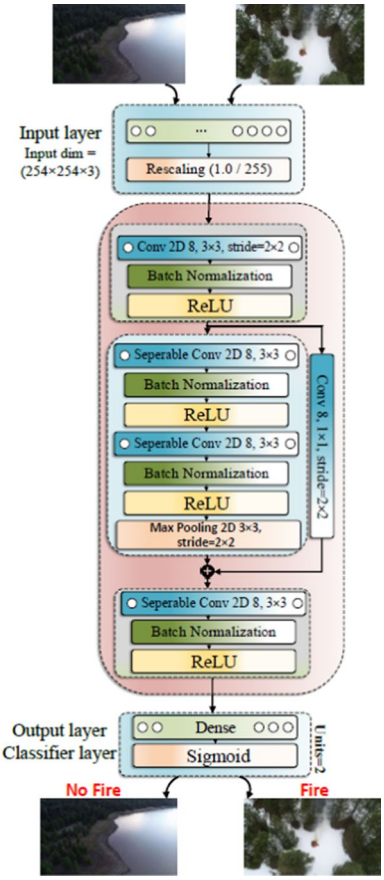

Fig. 2 Xception architecture [5]

5.2 Setup

The models are created on Google Colab due to the lack of hardware required for large computational tasks such as image processing and object detection. Once the complete architecture has been recreated, it is trained and tested on the same dataset as that proposed in [5]. The model is tested on the base dataset to establish comparison metrics. These metrics will function as benchmarks when comparing the model with other datasets and previous work.

5.3 Limitations and Validity Threats

5.4 Expected Results

The output is expected to be identical when using the same model and dataset, while the same results may not be achieved when using different datasets. The working hypothesis is that the dataset on which the model is trained and not necessarily the model architecture itself, causes the model to perform poorly on different datasets. The hypothesis is examined by training the model on two distinct datasets (FLAME and NASA) and cross-validating it on all datasets herein. In addition, an insight into what type of images should be used for the model to be robust, versatile, and immune to the changes of the input images is expected to be gained as well.

5.5 Datasets

In total three different datasets are tested. Two of which the model was trained on individually. Later these two models are cross-validated on the rest of the datasets. Datasets are balanced between classes, except for the FLAME dataset. A summation of datasets, used can be seen in Table 5

5.5.1 FLAME Dataset

The first dataset is the FLAME dataset [12], which consists of aerial imagery of fire patches in the forest, as presented in Fig. 3. The data is collected with the help of drones which has recorded footage and thereafter converted the frames into images. The dataset is 1.18GB in size and contains around 25000 images with fire and 15000 images without fire. The amount of images containing fire versus no fire differs; how this affects the model is explained in 8. In the current setup, training with this dataset took about 2,5–3 h over 40 epochs. To replicate the work as accurately as possible, 40 epochs was chosen since it was also used in the work by Shamsoshoara et al. in [5]. Moreover, this dataset was chosen as it was used by the state-of-the-arts and comparison purposes between them. The main characteristic of this data set is the vast amount of images it provides. With this dataset, a claim of - ”More is better” is examined.

Fig. 3 Example of the FLAME dataset, image containing fire (above) and no fire (below) respectively

5.5.2 Kaggle-"NASA"dataset

This dataset was used in the NASA Space Apps Challenge 2018 [13] as part of the team's data for fire detection. The dataset from Kaggle consists of images of fire in forest settings as well as normal looking images of forests [14]. This dataset is smaller than the FLAME 5.5.1 dataset (200 pictures of "fire", and 200 pictures of "no fire"), but it is also different. Images are not aerial and do not represent a wide scenery-like picture. This dataset is also used to train a model. Characteristics of this dataset are that images represent some form of fire or setting relatively close to the ground. With this dataset, a question is posed, how well a model trained on aerial images will perform when subjected to images of different resolutions. An example of an image from this dataset can be seen in Fig. 4.

5.5.3 "GitHub" Dataset

The dataset found on GitHub [15] is a collection of different fire datasets combined into one. It offers a large number of images containing a fire in different settings ranging from

Fig. 4 Example of the closer scenery,"NASA" dataset images for both classes, "Fire", and "No Fire"

Fig. 5 Example of the more versatile, "GitHub" dataset images for both classes, "Fire", and "No Fire"

Table 5 A summary of datasets used

	Dataset	Size/Class	Characteristic
1	FLAME	15000	aerial, scenery
2	NASA	2000	ground level, closer
3	GitHub	1200	ground level, versatile

forest fires to buildings on fire. The dataset consists of 1200 images for each class. While the images not containing fire consists of sunset images, forest images, images of office spaces, and cities etc. [15]. The dataset offers a wide range of different images, which could help prevent overfitting from the model. This is the most versatile dataset of all herein. The question posed here is, is it more beneficial to train a model on a smaller but more diverse dataset. An example of an image from this dataset can be seen in Fig. 5.

6 Results

In this section, the achieved results are presented and compared to the other work. Moreover, the model's performance on a different dataset is shown. A model's performance is calculated based on precision, recall, and accuracy. Additionally, a confusion matrix of the true positive (TP), true negative (TN), false positive (FP), and false negative (FN) is presented.

6.1 Evaluation

In terms of classification, statistical measures are being used in form of precision, recall, and accuracy. These results are calculated based on the confusion matrix presented herein. A confusion matrix shows how to model classifies images, thus highlighting when the model is confused by certain images.

A confusion matrix displays the actual labels over the predicted labels. Based on the model's predictions the items are divided into one of the four groups (TP, TN, FP, and FN).

- True positive: are images predicted to contain fire which are in fact fire images i.e., a fire image which is correctly predicted.
- True negative: are images predicted to contain no fire which are in fact no fire images i.e., an image which does not contain any fire is correctly predicted.
- False positive: are images where the model predicted that the image contained fire, while in fact it did not contain any fire. That is, an incorrect fire prediction.
- False negative: are images where the model predicted that the image contained no fire, however, there was fire present in the image. That is, an incorrect no fire prediction.

How these statistical methods are calculated can be seen below.

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

$$Recall = \frac{TP}{(TP + FN)} \tag{2}$$

$$Precision = \frac{TP}{(TP + FP)} \tag{3}$$

6.2 Model Performances Comparison

In total the Xception model was trained on two different datasets on two different separate occasions, hence, two tests were done. One of the tests consists of testing the model when trained on a different dataset. The second test consists of testing the outcome from [5], here the model is trained on the same dataset and comparison is made on how well it can predict images in a separate dataset. The summarising of the results can be seen in the Table 6.

6.2.1 Training Model on FLAME Dataset

The model was trained on the FLAME dataset. Naturally, the best performing model is the one saved towards the end of the training session. The model reached an accuracy of around 97% and a loss of just below 0.1. This is in line with the results presented in [5] and in Table 2. The validation accuracy is high at almost 100% accuracy and a loss very close to zero. (Figs. 6 and 7).

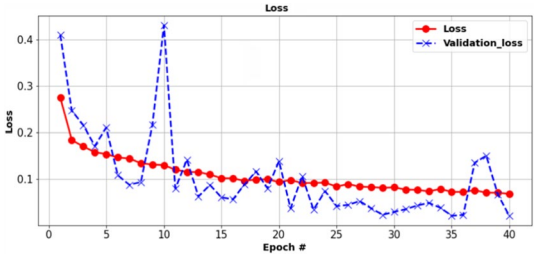

Fig. 6 Model training: Loss over 40 epochs on FLAME dataset

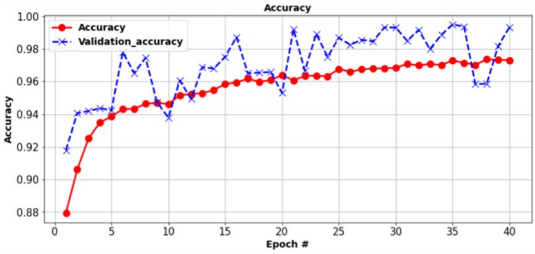

Fig. 7 Model training: Accuracy over 40 epochs on FLAME dataset

6.2.2 Training Model on NASA Dataset

This model is trained on the NASA dataset. As can be seen in Figs. 8, 9 the best model is not the one from the last epoch but the second to last. For this reason, the model from epoch 39 is chosen for the comparison. The training accuracy is 81% and the training loss is 0.36 for the model created in epoch 39. The chosen model's validation accuracy reached 92%, with a loss of 0.18. The loss rate does not seem to be elevated as that in the previous dataset, therefore the model does not seem to be overfitted, therefor it is considered a better choice for the comparison.

6.2.3 Performance: FLAME Model with GitHub Dataset

When evaluated on the GitHub dataset, the FLAME model achieved an accuracy of 60%, with a precision of 98%, recall 20%, and loss of 53. The results also show TP of 246 cases, FP of 4 cases, TN of 1231 cases, and FN of 989 cases. As illustrated in the confusion matrix in Fig. 10, and Table 7.

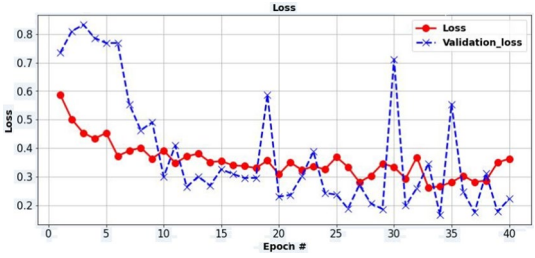

Fig. 8 Model training: Loss over 40 epochs on NASA dataset

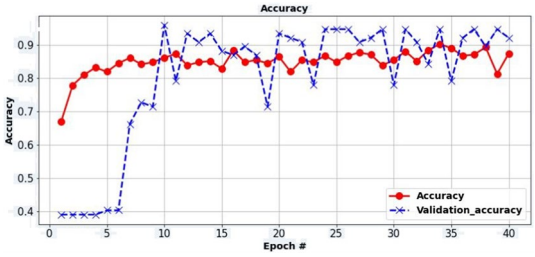

Fig. 9 Model training: Accuracy over 40 epochs on NASA dataset

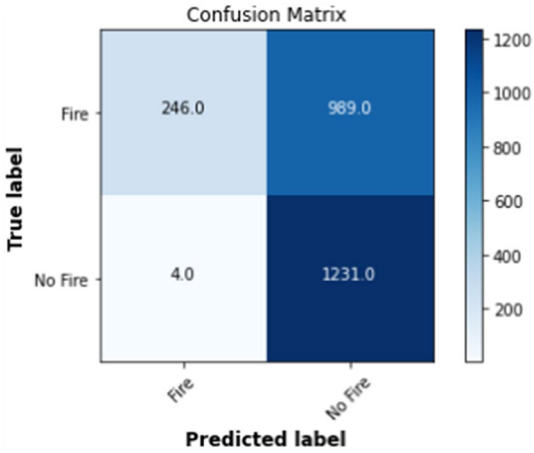

Fig. 10 GitHub dataset evaluated on model trained on FLAME dataset

6.2.4 Performance: FLAME Model with NASA Dataset

Fig. 11 consists of a confusion matrix covering the performance of the model trained on the FLAME dataset. These results are also summarised in Table 7. When evaluated on the NASA dataset; the model reached an accuracy of 61.88%, precision of 98.33 %, recall of 24.18%, and a loss of 33.01. Moreover, this resulted in 59 cases of TP, one (1) case of FP, 243 cases of TN, and 185 cases of FN. The NASA dataset is noticeably smaller than the FLAMES dataset.

6.2.5 Performance: NASA Model with FLAME Dataset

In Fig. 12, the evaluation results from the NASA model's prediction on images in the FLAME dataset is presented. The model achieved an accuracy of 29.66%, a precision of 20.40%, recall of 32.02%, and a loss of 0.90. Out of the large dataset containing about 39 000 images, 4598 were TP, 17937 FP, 7082 TN, and finally, 9759 FN. Results are shown in the confusion matrix Fig. 12, and summarised in Table 7.

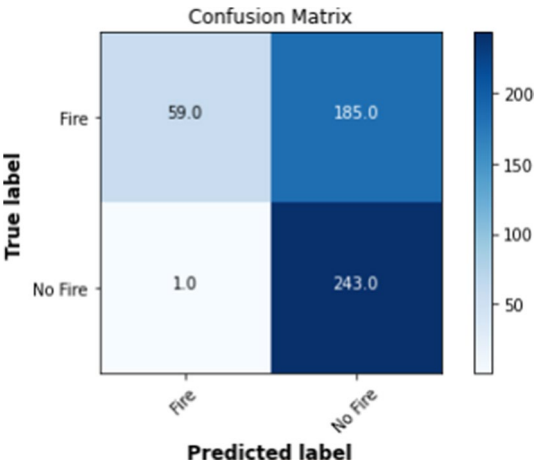

Fig. 11 NASA dataset evaluated on the model trained on FLAME dataset

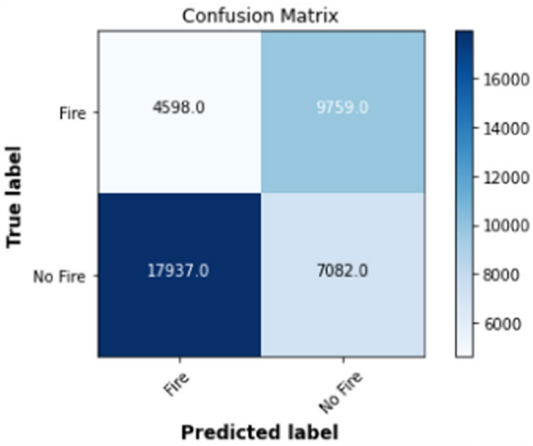

Fig. 12 FLAMES dataset evaluated on model trained on the NASA dataset

6.2.6 Performance: NASA Model with GitHub Dataset

When evaluating the GitHub dataset, the NASA model achieved an accuracy of 67%, precision of 82%, recall of 44%, and loss of 0.67. The results also show TP of 547, FP of 117, TN of 1118, and FN of 688. The confusion matrix can be seen In Fig. 13, and Table 7.

The validation results are summarised in Table 6. The Table shows the results from each corresponding model and the dataset it is tested on. The model column consists of the dataset for which the Xception model has been trained on. The dataset column, is the data for which the model has been tested on. Finally, the accuracy and loss is shown. In Table 7, normalised results of confusion matrices are shown in a form of metrics precision and recall.

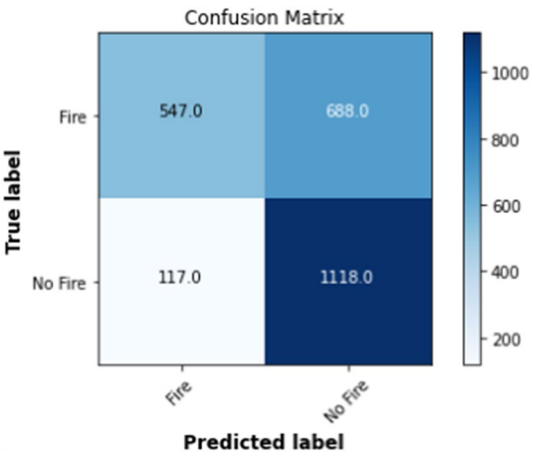

Fig. 13 GitHub dataset evaluated on model trained on NASA dataset

Table 6 A summary of validation results

	Model	Dataset	Accuracy	Loss
1	FLAME	FLAME	99%	0.02
2	FLAME	NASA	62%	33
3	FLAME	GitHub	60%	53.5
4	NASA	NASA	92%	0.18
5	NASA	FLAME	29%	0.90
6	NASA	GitHub	67%	0.67

Table 7 A summary of a confusion matrix precision and recall metrics

	Model	Dataset	Precision	Recall
1	FLAME	NASA	98%	24%
2	FLAME	GitHub	98%	20%
3	NASA	FLAME	20%	32%
4	NASA	GitHub	82%	44%

7 Analysis

In this section the results from 6 is analysed, the model's performances is analysed compared to each other.

7.1 Model Analysis

When looking at the model performance itself, it can be seen that both models performed best on their base datasets. FLAME model achieved 99% accuracy on the FLAME dataset, and NASA model 92% on the NASA dataset. Contrarily, FLAME model was not as good as NASA model on NASA dataset, and NASA model was not as good as FLAME model on FLAME dataset. NASA model achieved slightly better results on the third the GitHub dataset, with accuracy 67%. Overall, the FLAME model performed better than NASA model in general. Its worst accuracy performance was around 60%, and it was consequently as low on both NASA, and GitHub datasets. In NASA models case the situation is not as uniform. Although the model did perform at a slightly better 67% on the GitHub dataset, it struggled with the FLAME dataset achieving only 29%.

In general, the FLAME model seems to be performing better from an accuracy standpoint. This performance comes with the cost of time to train the model. FLAME model although more successfully overall, took about 2,5 h to finish, while slightly worse in some cases NASA model took only about 20 min.

7.2 Model Training Analysis

During training, the FLAME model is trained over 40 epochs, where for every epoch a model is saved. As can be seen in Figs. 6, 7, the model reached the region of maximal values after 40 epochs, and there is no reason to train it further. The best performing model was saved at the

last epoch. From the training results above the model trained on the FLAME dataset (herein denoted as the FLAME model) has a high training accuracy in line with the work by Shamsoshoara et al. [5] however, during validation the accuracy is somewhat high in comparison to the original work, reaching about 99% accuracy and 0.07 loss. The accuracy and loss achieved during training point towards an overfitted FLAME model. This is further shown in the results when the model is compared to other datasets.

The best performing model for the NASA dataset (herein, the NASA model) occurred in the 39th epoch. As can be seen in Figs. 8, 9, the model reached the region of maximal values after 40 epochs as well. The model trained on the NASA dataset achieved an accuracy of 86% and a loss of 0.36 during training. During validation, the model achieved an accuracy of about 92% and a loss of 0.26. This suggests that the model is not as overfitted as the aforementioned FLAME model and should therefore perform better on different datasets. However, by studying the graphs the validation loss has spikes that occur irregularly. Moreover, concerning the loss spikes, there are dips in the validation accuracy on the same epochs, suggesting high volatility in training.

7.3 Loss

The model trained on the NASA dataset has a reasonable low loss. While the FLAME model has a very low loss which as aforementioned could point toward an overfitted model. While the solution could be to use a more diverse dataset, in the case of the FLAME model this was intentionally left out since the task was to replicate the work in [5] in order to investigate the model's performance during classification on different datasets.

Keeping in mind that the FLAME model was overfitted with an exceptionally low loss rate. During validation, this point was proven. The model trained on the FLAME dataset reached a much larger loss rate of 33 and 53.5. The fast-growing loss rate further backs the theory of the model being overfitted. As for the model trained on the NASA dataset, based solely on the loss during training and validation, it is difficult to evaluate whether the model is over-or underfitted. It has an overall consistent low loss rate across the different datasets in comparison the the FLAME model.

7.4 Accuracy

During training the FLAME model produce accuracies in line with the initial study. However, during validation the model reached 99% which is not too far from the 94% presented in the original work [5]. Moreover, when tested on different datasets, the model had a consistent accuracy around 60%. This shows that despite reaching accuracies in the high 90% it could still perform above 50% when exposed to different images. While the NASA model was somewhat similar, it could not perform as well when exposed to different datasets. The dataset from GitHub showed similar accuracies for both models. Although, the NASA model could outperform the FLAME model with about 7 percentage points on this set. However, only having an accuracy of 29% when exposed to the FLAME dataset, is below par.

7.5 Confusion Matrix Analysis

For each case during the evaluation a confusion matrix is created. In the confusion matrix the model's classification can be seen in more detail. After calculating the precision and

recall, a comparative analysis is performed. Generally, in the case of fire detection in images, it is more beneficial to have models performing with lower precision and higher recall than the other way around. It is more acceptable to have more ”false alarms” than to miss fires altogether. Looking at the results summarised in Table 7, it can be seen that the FLAME model performs poorly under this assumptions. FLAME model is substantially more inclined to misclassify images to show no fire, than to not recognise them at all. High precision, and low recall for both NASA, and GitHub datasets for this model are evidence of that. The NASA model, although performing relatively well on the GitHub dataset, does not perform at all on the FLAME dataset.

7.6 Datasets Analysis

The models are trained on two datasets, FLAME model on the FLAME dataset, and the NASA model on the NASA dataset. After training and calculating results, both models are evaluated on the additional two datasets as seen in Table 6. Datasets differ from each other in several ways, potentially influencing the result. The first noticeable difference in the datasets is their sizes. FLAME data set is the largest with about 15000 data-point for each class, GitHub with 1200 data-points for each class, and NASA 200 data-points for each class. It is not clear from the results if such significant differences in sizes influence models performance, since both models performed better on their base set. It is worth mentioning though, that FLAME model performed at a relatively high level in the set it was trained on, and almost equally worse on the other two dataset. NASA model on the other hand, after performing relatively well on its base set, could not achieve equally good results on the other two sets. NASA model achieved slightly better results of 7% higher than the FLAME model on the GitHub dataset, although on FLAME dataset it completely failed (with 30% accuracy).

Another significant difference in datasets is their content. FLAME dataset is composed of homogeneous aerial images of landscapes. NASA is composed of ”closer to the ground” images, and GitHub dataset although with similarly framing as NASA dataset has more versatile scenes. Looking at the results, and taking into consideration the characteristics of the sets, it seems that the model’s performance, in general, is strongly connected to the type of dataset it is trained on. FLAME model trained on a larger yet uniformed dataset achieved satisfactory results but struggled with other smaller, and significantly different images. The same can be concluded for the NASA model. In this case, as in previous, the model achieved good results on its specific dataset (NASA dataset), and the similar in appearance GitHub dataset. It struggled on the other hand, with a totally different FLAME set.

8 Discussion

In this section, the results from the analysis 7 is being discussed. Looking closer at the possible reasons for an overfitted model, generalised datasets, and answering the research questions in Section 4.1. Overall the models performed similarly on the GitHub dataset, which was the only dataset that had no dedicated model and was only meant for benchmark evaluation. The surprising low accuracy from the NASA model on the FLAME dataset could have its origin in that the FLAME dataset only consists of aerial drone images. Moreover, since the dataset consists of video frames converted into images, many of the

images are similar which might be the reason for incorrect predictions. Considering a video recording of one minute with 20 frames per second, the collection footage in the location would produce 1200 images. Which could explain the difficulty in prediction by the NASA model. On the other hand, any accuracy below 50% in terms of binary classification is worse than guessing blindly. Furthermore, this might also explain the overfitted FLAME model.

8.1 Overfitting

The first research question is fundamental, but very important when it comes to successful model training. Are models dependent on specific data (RQ1)? If not what factors should be taken into consideration when choosing a model so that the training results are acceptable no matter the datasets used. When looking at the results 6, and the analysis 7, a case could be made of the choice of a dataset strongly influencing the model's performance in general (even on different datasets). FLAME model, for example, performed outstandingly on its dataset with great accuracy (97%), and loss (0.1). However, when applying it to other datasets, not only does the accuracy constantly drop with a rate of 30%, but the loss value skyrockets as well. In this situation, accuracy strongly decreasing, and loss value increasing is a case of overfitting. Although the FLAME model is trained on the biggest dataset, it is unable to learn features in a manner that would prevent overfitting. Contrarily, NASA model although trained on a smaller dataset, and significantly dropping accuracy on others, did not similarly elevate the loss value. NASA dataset, although significantly smaller did not cause the model to overfit to such a degree as in the case for FLAME. To answer RQ1, a conclusion could be reached that a model is dependable on a specific dataset. It is the dataset used in the training that can make a model robust and versatile, or can lock it in a specific pattern present in that dataset. To answer RQ2, when choosing a dataset for successful fire detection, it should strongly be taken into consideration the possibility of a model not being able to predict fires outside of the scope of the data used in training. The dataset should not only be sufficiently large but also versatile as to prevent overfitting.

8.2 Generalisation

The RQ2 is, is it possible to train a model on one dataset and use it successfully on a different one. The short answer seems to be, it depends. It is a question of models ability to generalise well enough that the output is correct whatever the input. Is it possible to train a model which can detect, isolate, and recognise specific features so well, that no matter the input it will always find these features if they exist. In theory, if all of the possible situations would be represented in a dataset, it should be possible. Although such a dataset would be enormous, and the training time would take weeks, months or even years to finish even on supercomputers. For this reason, a question is, is it possible to have a smaller more feasible set of data points to train on, that would trigger a good generalisation in a model? When looking at the results 6, and the analysis 7 one thing stands out. A model trained on a larger set (FLAME) was able to achieve not only better results while training, but also when generalising on different sets. On the other hand, NASA model was not able to do that. The question remains, what if all of the datasets would be lumped together and additional data augmentation was applied to them. Would that lead to even better results? Maybe, but the problem of training time returns. For these reasons, it is sufficient to say that for good model generalisation (RQ2.1), the dataset should be versatile enough, but in

constraints of some subjects. On one hand broad enough to be robust, and applicable to different situations, and on the other tight enough to still have a focus on the problem at hand. For a successful fire detection with one model, and with different datasets, it is important to have, not necessarily the biggest dataset possible, but one that broadly covers the scenarios of interest. The question of having a ”super” dataset covering every possibility, is a question for a future study.

9 Conclusion

In this work, a comparison study of the model's ability to effectively detect fire on several different datasets is examined. After looking at the results and the analysis, a conclusion is made that data influences model's capability to be applicable to different sets. The choice of data is important, as it shapes models capacity to generalise and potentially is the cause for overfitting. Data should not be scarce, nor too big. It should be versatile but in the scope of the area of interest. The choice of a model, although important, is not as important as choosing the dataset to train on. Many different successful models have been produced in the past. A problem arises when a model is meant to be used on different datasets. This ability is directly connected to the choice of the initial dataset. For having a robust model being able to detect fires from different data sources, it is critical to choose the initial training with consideration to these results. Possible future work could be to normalise the confusion matrix to provide a more even comparison between larger and smaller datasets, and adding a third“GitHub model”to the comparison. Moreover, finding or constructing datasets that are similar rather than a mixed set of images.

Acknowledgements The authors would like to thank Professor Per Jönsson from the Department of Materials Science and Applied Mathematics at Malmö University for his valuable advice on the formulation of the project concept.

Author Contributions All authors contributed to the study conception and design. Material preparation, data collection and analysis were performed by LK and JT. The first draft of the manuscript was written by LK, JT, RM, and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Funding Open access funding provided by Malmö University. The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.

Data Availability The datasets generated during and/or analysed during the current study are available online in the http://webshare.mah.se/aj3678

Declarations

Conflict of interest The authors have no relevant financial or non-financial interests to disclose.

References

1. Abid, F. (2020). A survey of machine learning algorithms based forest fires prediction and detection systems. *Fire Technology, 57*, 559–590. https://doi.org/10.1007/s10694-020-01056-z
2. de Almeida Pereira, G. H., Fusioka, A. M., Nassu, B. T., & Minetto, R. (2021). Active fire detection in landsat-8 imagery: A large-scale dataset and a deep-learning study. *ISPRS Journal of Photogrammetry and Remote Sesing, 178*, 171–186. https://doi.org/10.1016/j.isprsjprs.2021.06.002
3. Majid, S., Alenezi, F., Masood, S., Ahmad, M., Gundüuz, E. S., & Polat, K. (2022). Attention based cnn model for fire detection and localization in real-world images. *Expert Systems with Applications, 189*, 116114. https://doi.org/10.1016/j.eswa.2021.116114
4. Dutta, S., & Ghosh, S. (2021). Forest fire detection using combined architecture of separable convolution and image processing. In *2021 1st International conference on artificial intelligence and data analytics (CAIDA)* (pp. 36–41). https://doi.org/10.1109/CAIDA51941.2021.9425170.
5. Shamsoshoara, A., Afghah, F., Razi, A., Zheng, L., Fule, P. Z., & Blasch, E.(2020). Aerial imagery pile burn detection using deep learning: The FLAME dataset, CoRR, vol. abs/2012.14036, arXiv: 2012. 14036. [Online]. Available: https://arxiv.org/abs/2012.14036.
6. Vincente, J., & Guillemant, P. (2002). An image processing techniques for automatically detecting fire. *International Journal of Thermal Science*, 1113–1120.
7. Ghali, R., Akhloufi, M. A., Jmal, M., Souidene Mseddi, W., & Attia, R. (2021). Wildfire segmentation using deep vision transformers. *Remote Sensing, 13*(17), 3527.
8. Wang, Z., Yang, P., Haotian, L., Zheng, C., Yin, J., Tian, Y., & Cui, W. (2022). Semantic segmentation and analysis on sensitive parameters of forest fire smoke using smoke-unet and landsat-8 imagery. *Semantic Sensing, 14*, 1113–1120. https://doi.org/10.3390/rs14010045
9. Farhan, R., Al-Jumaili, M., & Nezar, I. S. (2019). Fire detection using convolutional deep learning algorithms, REVISTA (pp. 36–41). DOI: /https://doi.org/10.4206/aus.2019.n26.2.53/www.ausrevista.com.
10. Khan, M., Irfan, M., & Sung, W. B. (2018). Convolutional neural networks based fire detection in surveillance videos. In *special section on multimedia analysis for internet-of-things*, vol. 6. IEEE.
11. Kim, Y.-J., & Kim, E.-G. (2018). A study on fire detection using faster cnn and resnet. In *International lnformation lnstitutee*, vol. 21. 1nternational lnformation lnstitute (pp. 173–180).
12. Shamsoshoara, A., Afghah, F., Razi, A., Zheng, L., Fule, P., & Blasch, E. (2020). The flame dataset: Aerial imagery pile burn detection using drones (uavs). [Online]. Available: https://doi.org/10.21227/qad6-r683.
13. Ahmed Saied A. M., Ahmed Atef G. Osman, Hebatullah Mostafa S. M., Ahmed Abdel-Aziz S., Gamal Eldin, A. M. (2018). Available: https://2018.spaceappschallenge.org/challenges/volcanoes-icebergsand-asteroids-oh-my/real-time-fire-app/teams/the-faze/members/.
14. Saied, A. (2018). Fire dataset, [Online]. Available: https://www.kaggle.com/phylake1337/fire-dataset.
15. Visweswaran, Fire-detection-dataset. (2020). [Online]. Available: https://github.com/VISWESWARAN1998/Fire-Detection-Dataset

Lukas Kurasinski is a final year master's student in Applied Data Science in the Department of Computer Science and Media Technology at Malmö University. His research focuses on applied data science with applications in data and image processing.

Jason Tan is a final year master's student in Applied Data Science in the Department of Computer Science and Media Technology at Malmö University. His research focuses on applied data science with applications in data and image processing.

Reza Malekian is a Full Professor and Deputy Head of Department in the Department of Computer Science and Media Technology at Malmö University, Sweden. Dr. Malekian is also an Extraordinary Professor in the Department of Electrical, Electronic and Computer Engineering at the University of Pretoria, South Africa, where he previously led the Advanced Sensor Networks research group and received the Vice-Chancellor and Principal's Exceptional Young Researcher's award. He is also nominated for the Tage Erlander prize, an award by the Royal Swedish Academy of Science for outstanding scientific research in natural sciences and technology.