Toward Energy-Efficient Deep Neural Networks for Forest Fire Detection in an Image

Yali Wang, Chuulabat Purev, Hunter Barndt, Henry Toal, Jason Kim, Luke Underwood, Luis Avalo, Arghya Kusum Das
Computer Science Department
University Of Alaska Fairbanks
Fairbanks, United States

***Abstract*—Forest fires cause huge losses and are a serious problem facing many countries worldwide, including the USA, Canada, Brazil, Siberia, and Indonesia, to name a few. Automatic identification of forest fires in an image is thus an important field to research in order to minimize disasters while also helping in mitigation planning and designing rescue tactics. Artificial Intelligence technologies, especially deep neural networks, have emerged recently with promises to detect fires with better accuracy from an image. However, the massive energy consumption of deep neural networks thwarts their widespread adoption, especially when it comes to onsite detection of fire utilizing low-power devices such as those embedded in a drone or an artificial satellite.**

In this paper, we develop multiple deep neural network models such as a Convolutional Neural Network (CNN), a Deep Belief Network (DBN), an Auto Encoder (AEnc), and a U-net model to detect forest fires and systematically analyze their accuracy and energy consumption using IEEE FLAME data set which is openly available at IEEE data portal. After developing the models, we systematically pruned the models, retrained them, and analyzed their accuracy and energy consumption upon deployment. Our analysis shows that the CNN has the highest accuracy (almost 99%) on the validation data set, whereas the DBN model consumes the least amount of energy after deploying on both CPU and GPU. The trained models are deployed on a website for use. The source code can be found on GitHub (https://github.com/akdasUAF/ForestFireDetection).

***Index Terms*—Forest Fire, Fire Detection, Deep Learning, Deep Neural Network, Energy Consumption**

I. Introduction

In recent years, forest fires have become a serious hazard in many areas due to several environmental and climatic issues, such as global warming. According to the World Health Organization (WHO), between 1998 and 2017, approximately 6.2 million individuals were impacted by wildfires and volcanic events, leading to 2400 deaths globally due to suffocation, injuries, and burns [1]. Wildfires also simultaneously impact weather and the climate by releasing large quantities of pollutants and poisonous gases such as carbon dioxide, carbon monoxide, and fine particulate matter into the atmosphere. This results in various health issues, including respiratory and cardiovascular problems. Wildfires not only affect people's normal life and health but also greatly harm forests and land resources with a deleterious impact on world climate.

Accurate detection of forest fires has been a long-standing problem. Fire scientists and fire workers have been using satellite imaging and cameras to spot wildfires for years. However, the earlier image analysis and object recognition technologies for fire detection are far from producing optimal results. The disadvantage of color-based fire detection models is the high rate of false positives since, in most cases, monochrome information is insufficient for early and robust fire detection. [2]. The recent advances in the field of AI, specifically deep neural networks, presented enormous promises to detect wildfires accurately from an image [3]. In the current state-of-the-art research, scientists frequently send drones to scout wildfires [4]. The drones send the images captured to the ground station for further analysis. Finally, in the ground station, the scientists are utilizing powerful servers to detect fire in those images utilizing advanced AI-based techniques [5].

However, this method also has limitations. In most remote areas, the Internet is scarce, and the images, videos, or any other multimedia files cannot be sent to the ground station immediately for analysis delaying the response time [6]. Hence, an onsite fire detection technique would be tremendously helpful that can analyze the images using AI on the location of the fire and send the decisions to the ground station using relatively abundant communication resources such as SOS services. But the major challenge in implementing such an onsite fire detection system is the energy requirement. Deep neural network models are notorious for their high energy and resource consumption thwarting their use on low-power-consuming devices that can be embedded in a drone or a satellite [7].

In this paper, we took the initial steps toward developing energy-efficient deep neural network models that can accurately detect forest fires in an image while consuming substantially less amount of energy. We first developed four different deep neural network models, such as Convolutional Neural Network (CNN), Deep Belief Network (DBN), Auto Encoder (AEnc), and a U-Net, to detect forest fire in an image accurately. We trained our models utilizing the IEEE FLAME [8] data set, which is openly available on the IEEE data portal. Then, we systematically prune the deep neural network models to evaluate their accuracy and energy consumption

Identify applicable funding agency here. If none, delete this.

characteristics upon deployment. Our analysis shows that CNN achieves the maximum accuracy of almost 99% DBN shows the best energy efficiency compared to the other models. The DBN shows uniform energy consumption compared to CNN because of the topology of the neural network structure. DBN, which can also be described as a stacked Restricted Boltzmann Machine (RBM), has a simpler and uniform connection among its neurons at different layers whereas a CNN's neurons at different layers are sporadically connected during the training process. That is, by adding an RBM layer to a DBN always increase the number of connection, whereas adding another convolutional or dense layer to a CNN may result in a sparser network with less number of connections.

The rest of the paper is organized as follows. Section-II describes the related work prior to our current effort. Section-III briefly discusses the background of the tools and technologies that we used. Section-IV describes our deep neural network models in detail. In Section-V, we compare the accuracy and energy consumption of each model. This section also provides the links to our source code which is openly available to and proof-of-concept software product. Finally, Section-VI concludes our paper.

II. Related Work

A. Earlier Detection Methods

Traditionally, forest fires were mainly detected by human observation from fire lookout towers and involved only primitive tools, such as the Osborne fire Finder [9]; however, this was prone to human error. The automated methods of forest fire detection mainly use two different types of cameras, namely Red-Green-Blue (RGB) cameras and Infra-Red (IR) cameras [10] [2]. The traditional detection methods using RGB cameras combine features related to spatial, temporal, and texture characteristics of the fire site, such as the color of flame, the color of smoke, their motion determined through consecutive video frames, and so on [11]–[18]. These methods are mostly based on the principle of chromatic difference measurement among fire pixels and the background. However, these differences are measured utilizing different properties of the images such as intensity and saturation [12], modulated chrominance [14], [15], and lighting condition [16], among others.

B. Machine Learning-Based Detection Methods

Recently, machine-learning models have opened up new horizons in forest fire detection. For instance, logistic regression models can be used to estimate the probability of a fire occurring based on a variety of factors such as temperature, humidity, and wind speed [19]–[21]. Support Vector Machines (SVMs) can be used as the analysis method, then the Genetic Algorithm can be employed to compute the parameters of SVM to assess the forest fire susceptibility [22]. These machine-learning models analyze historical fire data and current environmental conditions to make predictions. When combined with real-time data from drone or satellite surveillance, these models can provide highly accurate early warnings and projections of fire spread. This information can be invaluable for forest management teams and emergency responders, helping them take swift action to prevent fires or limit their damage.

In this paper, we have trained deep learning models and compared their pros and cons in fire detection.

C. Deep Learning-based Detection Method

Deep learning, with its automatic feature detection techniques, has shown significantly superior performance compared to the traditional image analysis methodologies [23], [24]. CNN [25] has been the most popular choice regarding fire detection in an image. Paper [26] proposed a CNN a-based smoke detection algorithm based on the motion characteristics of smoke. [27] proposed a deep convolutional neural network for the detection and segmentation of fire pixels. Similarly, [3], [21], [25], [28]–[30] also used CNN to detect fire in an image or classify the images with or without fire. However, the CNNs vary in terms of different hyperparameters, such as the number of convolutional and dense layers, the number of neurons in each layer, the type of optimization technique, and so on. For example, [3] is inspired by complex VGG16 [31] and ResNet50 [32] whereas [25] and [28] is inspired by the Google Lenet [33] and Lenet-5 [24] respectively. [21] used Inception-V3 [34] whereas [30] used YOLO-V3 [35]. Like [30], [36], and [37] also used YOLO for fire detection. However, the first one used YOLO-V4, whereas the second one used YOLO-V5. [38] and [39] both used R-CNN [40] to detect fire.

Not only CNN but other deep neural network models have also been used in the literature that reported high accuracy for fire detection in an image. For example, [41] and [42] used DBN for fire detection. [43] and [44] used the Auto encoder-decoder model and U-net [45] which is an improved version of Auto Encoder model. [46] and [47] proposed a generative adversarial network (GAN) to detect forest fires. [48] proposed a Stacked Encoded-EfficientNet (SE-EFFNet) model. Paper [49] combined both YOLO-V5 and U-Net for fire detection.

Although many deep neural networks have been proposed for accurate forest fire detection, there is hardly any work done addressing energy consumption in this context. In our paper, we took the first step toward that. We developed the fundamental versions of many deep neural network models, including CNN, DBN, Auto Encoder, and U-Net, and then systematically analyzed the energy consumption profile of each of these models.

D. Energy Consumption in Deep Learning

Deep learning can facilitate decision-making during firefighting operations [50], especially with the advances in drone and artificial satellite technologies. To explore the full potential of deep neural networks and human-robot collaboration, it is important to embed the model in drones [4] that can detect fire. However, the high energy consumption of the deep neural network models has been a major impeding factor [51].

The paper [52] focuses its analysis on inference FLOPs (Floating Point Operations) required to process one input item. After they estimate how much energy is needed to perform one inference step with a given DNN and the GFLOPS per Watt for different GPUs. In this study [53], the computational performance of the network is assessed using an Arduino Nano 33 BLE Sense microcontroller. Measurements of flash usage, peak RAM usage, and inferencing time are conducted. The findings indicate that increasing the number of neurons in the dense layer results in elevated computational and storage demands, but does not yield substantial improvements in prediction performance. In our research, we used the PyJoules package to monitor energy consumption when referring to various deep-learning models to identify wildfire pictures and got the energy consumption of CPU, GPU, and RAPL in microjoules [54].

In our study, we took a more fundamental approach to understanding the energy consumption of deep neural network models. Instead of proposing any new TinyAI [55] model, we evaluated the energy consumption of multiple fundamental models to help the scientists to build better compression and pruning techniques in the future.

III. Technology Background

A. Tools and Technologies

In this section, we will give a brief overview of a typical deep-learning analytic pipeline as shown in Figure 1. All our models follow this fundamental pipeline. The detailed background of each specific model that we used in this study and our proposed architecture is discussed later in Section IV-B.

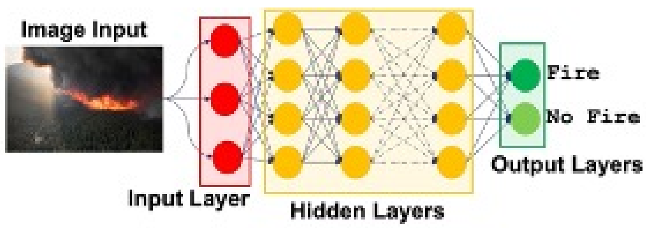


Fig. 1: Deep Learning Pipeline

AI is an umbrella term representing a range of techniques that allow machines to mimic or exceed human intelligence. When humans think, they sense what's happening in their environment, realize what those inputs mean, make a decision based on them, and then act. Artificially intelligent devices are in the early stages of beginning to replicate these same behaviors.

AI has made significant strides in recent years, particularly in the field of object detection, such as detecting a fire object in an image. Object detection is a computer vision technique that allows machines to identify and locate objects within images.

Among other AI techniques, artificial neural networks (ANN) showed the most accuracy when it comes to object detection. ANN is a method in AI that mimics the human brain more precisely by organizing multiple computational units known as neurons (the circles in Figure 1) in the form of layers, as shown in Figure 1. Each layer of neurons is capable of extracting important information (features) from the input data by applying some mathematical (specifically, linear algebra) tools and then passing the extracted information or features to the next layer. At the border level, there are three different types of layers in any ANN. First, an Input layer reads the raw input data, such as an image. Second, a series of hidden layers that extract features progressively from the input. Finally, an output layer gives the output, such as classifying the image.

Among different types of ANN, deep neural networks (DNN) emerged recently with the maximum promise towards object detection. DNNs are a special type of neural network that uses significantly many hidden layers (hence, the name deep) between its input and output layers to extract higher-level features from the raw input data progressively. Each layer applies different types of computational methods to its input systematically to extract different types of features. For example, in image processing, lower layers of the neural network may identify edges, while higher layers may identify the concepts relevant to a human, such as the presence of a fire object.

B. TensorFlow and Keras

TensorFlow and Keras are essential tools for training deep learning models for wildfire detection due to their key advantages [24]. TensorFlow provides a powerful and flexible framework for implementing and executing deep learning algorithms efficiently on various hardware platforms. It offers a wide range of pre-built functions and libraries specifically designed for deep learning tasks, enabling easier model development and deployment. Keras, on the other hand, serves as a high-level API built on top of TensorFlow, simplifying the process of building, training, and evaluating deep learning models. Its user-friendly interface and extensive documentation make it accessible for both beginners and experienced researchers, facilitating rapid prototyping and experimentation in wildfire detection.

C. pyJoules

pyJoules is a Python library that provides a convenient interface for measuring and monitoring energy consumption in software applications. It allows developers to measure the energy consumption of their code at different levels, such as the entire application, specific functions, or even individual code blocks. PyJoules can monitor the energy consumed by specific devices of the host machine, such as CPU and GPU, utilizing energy measurement capabilities provided by the hardware vendors, such as Intel's Running Average Power Limit (RAPL) interface, Nvidia Management Library, etc.

IV. Methodology

A. Data Description

To train, test, and validate our model, we used a set of RGB images from IEEE FLAME Dataset [8], which is openly available at IEEE Data Portal. The dataset has been released recently by the scientists of Northern Arizona University and Air Force Research Laboratory as a standard dataset for benchmarking. The developers of the dataset first conducted several fire spots to simulate wildfires. Then they used UAVs equipped with Zenmuse X4S and the Phantom 3 camera to record videos of those fire spots. Then, they extracted the image frames from those videos and finally, classified those images as *fire* and *no-fire*.

The video configuration was 1280 × 720 resolution and 29 frames per second (FPS). The H.264 codec was used for all the recordings. The captured videos were converted to frames based on the recorded Frames Per Second (FPS). The extracted images consist of both good and bad-quality images. The good-quality images are not blurry, and the terrain, trees, and most importantly, the fire spots can be clearly visualized. On the contrary, the bad-quality images are not only blurry but the fire spots in those images are too small or vague to be seen. Figure 2 shows examples of good and bad quality images containing a fire spot. We used both good and bad-quality images in our dataset to improve its robustness. We split the dataset into 80% for training, 10% for validation, and 10% for testing.

B. Model Description

1) Convolutional Neural Network: A Convolutional Neural Network (CNN) is a type of neural network architecture which is widely used for object recognition. It consists of multiple layers, including input, convolutional, pooling, and fully connected layers. The input layer receives the image data, and the convolutional layers apply filters to extract local features and capture spatial relationships. The pooling layers reduce the spatial dimensions of the feature maps, reducing the computation and enhancing translational invariance. The fully connected layers take the flattened feature maps and learn higher-level representations, ultimately producing the desired output. CNNs are trained through backpropagation, where the weights are adjusted to minimize the difference between predicted and target outputs.

Our CNN model is constructed as shown in Figure 3. This model consists of two halves. The dataset is imported from a given directory. The images from this dataset are labeled as 'Fire' and 'No_Fire.' The first half of the code has three pairings of a convolutional layer and a max pooling layer, which are used to extract the most relevant features from the input image. The second half consists of four dense layers with a decreasing number of nodes, which gradually reduces the dimensionality of the input until it reaches the final layer. The final layer uses sigmoid activation to determine whether there is a fire so that the network will output a single value between 0 and 1. The model is compiled with the Adam optimizer, binary cross-entropy loss as a binary classification task, and accuracy as the metric.

2) Deep Belief Network: Deep Belief Networks (DBNs) are probabilistic generative models that consist of multiple layers of restricted Boltzmann machines (RBMs). RBMs are unsupervised learning models that learn to extract useful features from input data. In a DBN, the bottom layer is treated as the visible layer, which receives the input data, and the subsequent layers are considered hidden layers. Each layer is trained in an unsupervised manner, with the hidden layer of one RBM serving as the visible layer for the next RBM. This layer-wise training enables the DBN to learn hierarchical representations of the input data, capturing increasingly complex features as it goes deeper [41]. After the unsupervised pretraining, a DBN can be fine-tuned using supervised learning techniques, making it suitable for tasks such as classification and feature learning. DBNs have been successfully applied to various domains, including image recognition, speech recognition, and recommendation systems [56].

We implemented the DBN using Restricted Boltzmann Machines (RBMs) with TensorFlow and Scikit-learn libraries. The model consists of three main layers. There are two RBM layers, each with 256 neurons, followed by a logistic regression layer for classification. The code reads a dataset of images, preprocesses the data, and trains a DBN model by stacking the two RBMs and the logistic regression layer. The generated model is saved to a joblib file. The model is constructed as shown in Figure 4.

In this implementation, the RBM layers are trained unsupervised, learning features from the input data. The logistic regression layer then leverages these learned features for classification. Using Scikit-learn's pipeline makes it easier to stack multiple layers and train the model in a more structured and modular manner.

3) Auto Encoder: An Auto Encoder is a deep neural network that consists of an encoder path and a decoder path. The encoder path maps the input data to a lower-dimensional latent space representation, and a decoder network reconstructs the original input data from the latent representation. Specifically, the encoding path captures the context and extracts high-level features through a series of down-sampling operations, such as convolutional and pooling layers. The decoding path employs up-sampling techniques, such as transposed convolutions. By training the Auto Encoder to minimize the reconstruction error between the input and the output, it learns to capture the essential features in the data.

Our Auto Encoder model is trained for anomaly detection by training the architectures using images without fire and checking if it can detect fires as an anomaly using the largest error pixels in the reconstructed images. The model is constructed as shown in Figures 5 and 6, it has an Encoder and a Decoder. The code first imports the required packages and functions. It defines two functions to import datasets. The first function imports images either with or without fire. The second one imports a different set of images with their corresponding masks for segmentation tasks. Then, the code constructs the Auto Encoder model, composed of an encoder and a decoder. The encoder compresses the input data, and the decoder reconstructs the input from this compressed representation. In the encoder part of our model, the Conv2D and MaxPooling2D layers, in essence, reduce the spatial dimensions of the input

(a) Example of a good-quality image containing a fire spot.

(b) Example of a good-quality image containing multiple fire spots.

(c) Example of a bad-quality photo containing vague fire spots.

Fig. 2: Photo examples from the IEEE FLAME Dataset.

while simultaneously increasing the depth to capture more complex features. The encoder gradually downsamples the input image while extracting important feature representations. The decoder part of the model does the opposite, gradually upsampling the compressed representation received from the encoder back to the original image size while attempting to recreate the original image. The upsampling is performed by Conv2D and UpSampling2D layers. The last layer of the decoder uses a sigmoid activation function, ideal for reconstruction as it outputs values between 0 and 1, which match the rescaled image pixel values [27].

4) U-Net: U-Net is an improved version of Auto Encoder. Like Auto Encoder, it also consists of an encoding path and a corresponding decoding path. However, an Auto Encoder often loses spatial information while passing the images through its encoding path, which may not be recovered during decoding. On the contrary, a U-Net's decoding path is shortcircuited with its encoding path to recover the spatial information by directly concatenating feature maps from the encoding path. These skip connections enable the network to combine both low-level and high-level features, which include the spatial information allowing for precise localization of objects during segmentation which can improve the accuracy.

Our U-Net model uses image similarity as a measure of success. The model is constructed as shown in Figures 7 and 8. The encoder of our U-Net code consists of Conv2D and MaxPooling2D layers. The decoder consists of UpSampling2D and Conv2D layers, with each Upsampling2D layer followed by a concatenation operation to combine its output with the output of the corresponding MaxPooling2D layer in the encoder. This allows the model to use the lower-resolution features learned at the corresponding stage of the encoder.

In the main function, the model is imported and evaluated on both the fire and no-fire datasets. The model's predictions are compared with the original images to calculate the mean squared error (MSE). If the MSE is above a certain threshold, it is considered an anomaly and highlighted in the image. After training, the model is evaluated on new images, and if the reconstruction error (MSE) is high, it flags those images as potential anomalies (fires).

V. Results and Discussion

A. Model Accuracy

Figure 9 compares the accuracy of different models. The CNN model performs exceptionally well, reaching an accuracy of 99.19% on the testing images. This is a very promising result and clearly demonstrates the power of the CNN architecture for image classification tasks. Our DBN model has an accuracy of 97.87%. Our U-Net model has an accuracy of 78.72%. Our Auto Encoder model has an accuracy of 63.8%, and this model can mark the fire location on the image.

B. Model Energy Consumption

After obtaining the trained models, we changed the layer structure of each model and retrained multiple models to monitor the energy consumption of each new model. For the CNN model, we made two primary changes: the number of Conv2D layer and MaxPooling2D layer pairs and the number of dense layers. For our DBN model, the number of RBM layers was changed while keeping the number of neurons at 256. Our Auto Encoder and U-Net models were varied by changing the number of encoders and decoders, each of which consisted of one Conv2D and MaxPooling2D layer pair. Figure 10 compares the GPU energy consumption of the model.

It is interesting to note that the CNN model, which shows the best accuracy, shows fluctuating characteristics for GPU energy consumption as shown in Figure 10a. As a matter of fact, energy consumption decreased with the increasing number of convolutional layers. The result is counterintuitive. However, we believe that this is due to the larger number of convolutional layers substantially decreasing the number of edges and thus reducing the complexity of the matrix multiplication operations. This means that the CNN model

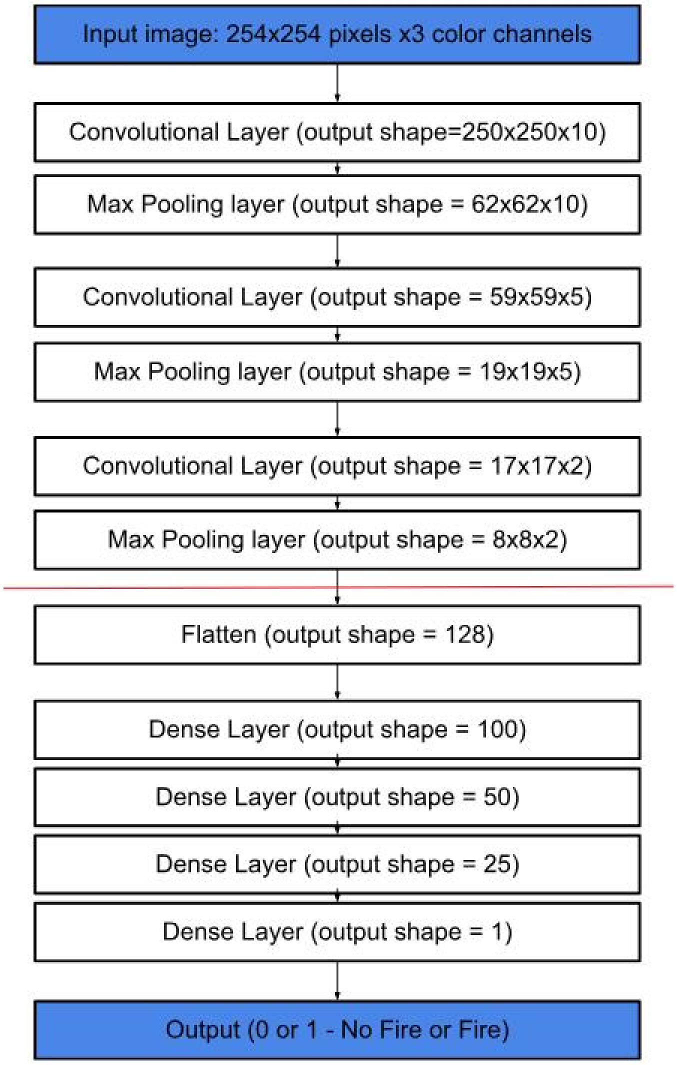


Fig. 3: CNN model architecture.

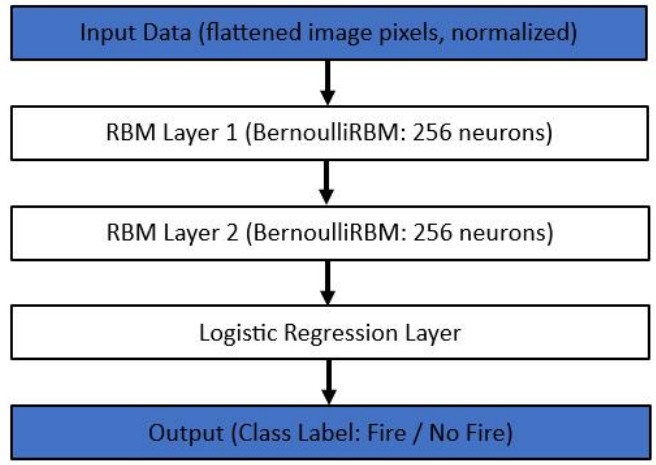


Fig. 4: Deep belief network model architecture.

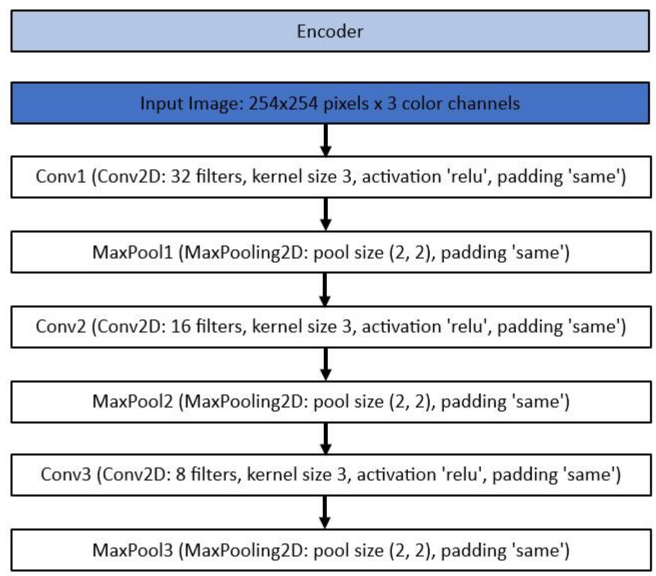


Fig. 5: Auto Encoder model architecture - encoder part.

configuration has ample potential for pruning and compression, making it an ideal choice for low-power devices.

The energy consumed by a DBN model (in Figure 10b) shows almost a linear increasing trend with an increase in the number of RBM layers. In a DBN, each RBM layer is stacked on top of another using a simple connection topology. The number of edges uniformly increases with the number of RBM layers. This means a complicated DBN model has limited scope for low-power devices. However, a simpler DBN can be an ideal choice for low-power devices as one RBM layer consumes half of the energy (2283.8 microjoules) compared to CNN architecture which consumes the lowest energy (4539.1 microjoules) among all the different CNN architectures we tested (shown in Figure 11).

Figure 12 compares the accuracy-to-energy trade off of different deep neural network models. We have divided the percentage accuracy obtained from the most accurate architecture of a particular type of deep neural network model with the energy consumed by that particular architecture. Clearly, DBN shows the best accuracy-to-energy ratio, which also proves its efficacy for onsite forest fire detection by embedding it in a low-power device.

C. Software Availability

Our source code and documentation are openly available through a public GitHub repository: https://github.com/akdasUAF/ForestFireDetection.git. A working prototype of the software is available at the website http://137.229.25.190:8000/. Although we are actively working and constantly updating the code, a user can test the accuracy of different deep neural network models with images from the IEEE Flame dataset. The code and the web interface are currently tuned for IEEE Flame's RGB dataset, including training, testing, and validation set. We are working

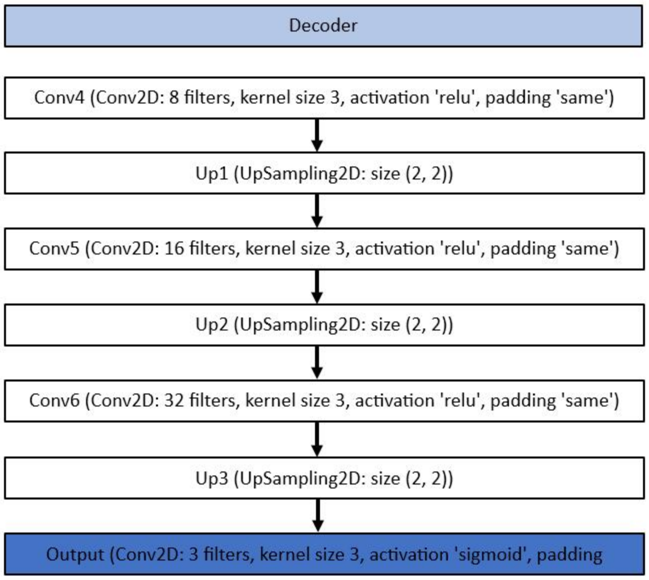


Fig. 6: Auto Encoder model architecture - decoder part.

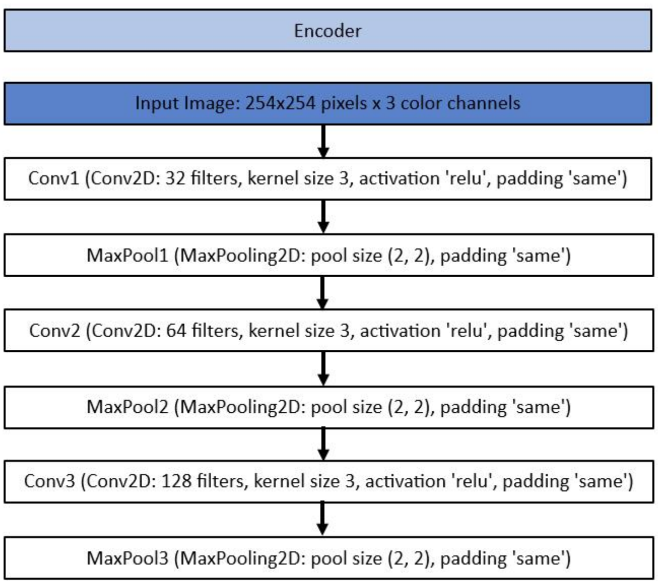


Fig. 7: U-Net model architecture - encoder part.

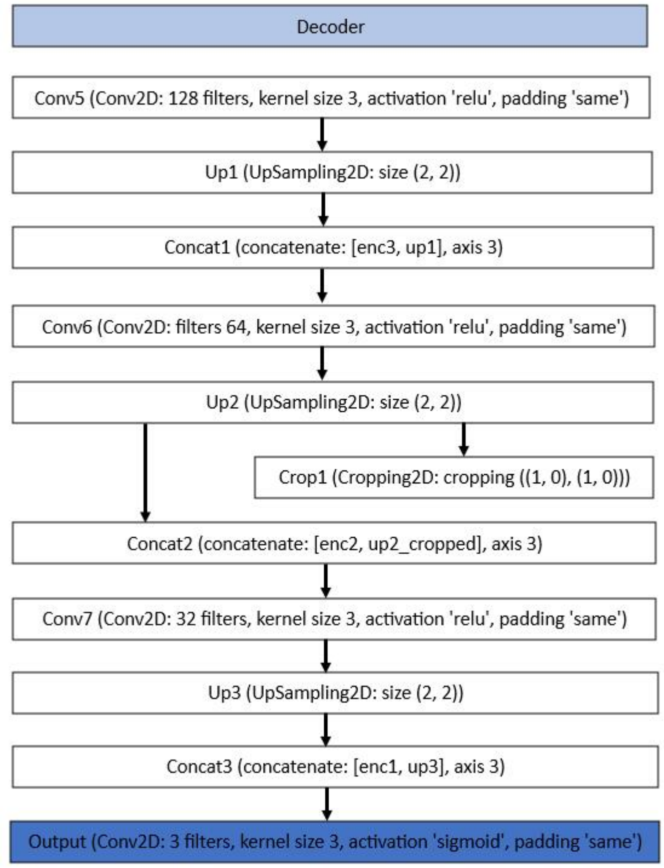


Fig. 8: U-Net model architecture - decoder part.

towards preparing a better version of the software that can identify fire in other datasets too.

VI. Conclusion and Future Work

As forest fires occur more frequently, we are actively exploring the onsite deployment of accurate and energy-efficient deep neural network models to identify wildfires, hoping to help reduce losses caused by forest fires and help emergency response departments such as firefighting to carry out rescue and disaster relief. We developed four fundamentally different deep neural network models to detect forest fires and compared their accuracy and energy consumption. Our CNN model shows an accuracy of 99.19% over the IEEE FLAME dataset, whereas our DBN model shows the best accuracy-to-energy-consumption ratio. Our systematic analysis clearly shows the promise of both models toward energy-efficient deployment on low-power devices such as drones and artificial satellites. We have also obtained a competitive accuracy using Auto Encoder and U-Net models. To help the scientists and provide easy access to these models for future research, we have developed an open-source and proof-of-concept software product with an easy-to-use web interface for fire detection (as shown in Figure 13).

We are actively working on improving our software product, model accuracy, and energy efficiency. While our end goal is to deploy the deep neural network model on a drone, our immediate future work is to develop an efficient pruning and compression technique for deep neural network models based on our current observations. We are also exploring the potential of GPU-based edge computing devices such as the Nvidia Jetson Nano, which recently emerged as an extremely energy-efficient yet very powerful computer to run neural networks for applications like image classification and object detection. Such devices have already shown enormous potential to be embedded in a drone. We believe that these GPU-based edge devices, together with our energy-efficient deep neural network model, can make a big impact on future forest fire mitigation and rescue techniques, which is a national priority, especially

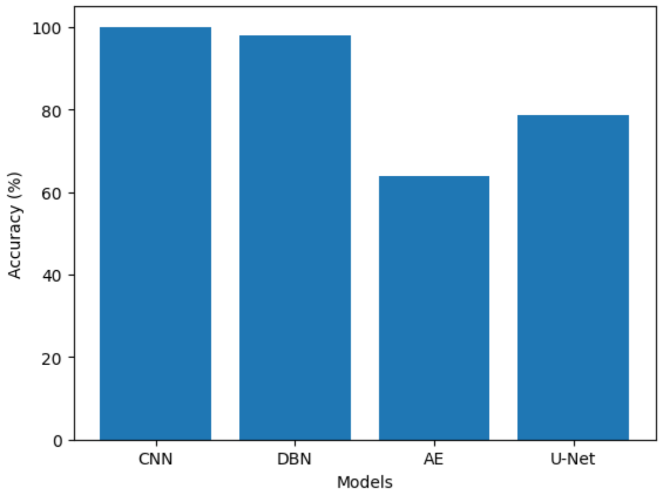


Fig. 9: Accuracy of different deep neural network models (higher is better)

during this period of climate change.

Acknowledgment

This work is supported in part by NASA Alaska EPSCoR Program (80NSSC22M0046) Research Infrastructure Development (RID) Award and UAF URSA (Undergraduate Research and Scholarly Activity) Innovative Technology and Education (ITE) Award.

VII. References

References

[1] “World health organization.” [Online]. Available: https://www.who.int/health-topics/wildfires/

[2] P. Barmpoutis, P. Papaioannou, K. Dimitropoulos, and N. Grammalidis, “A review on early forest fire detection systems using optical remote sensing,” *Sensors*, vol. 20, no. 22, p. 6442, 2020.

[3] J. Sharma, O.-C. Granmo, M. Goodwin, and J. T. Fidje, “Deep convolutional neural networks for fire detection in images,” in *Engineering Applications of Neural Networks: 18th International Conference, EANN 2017, Athens, Greece, August 25–27, 2017, Proceedings*. Springer, 2017, pp. 183–193.

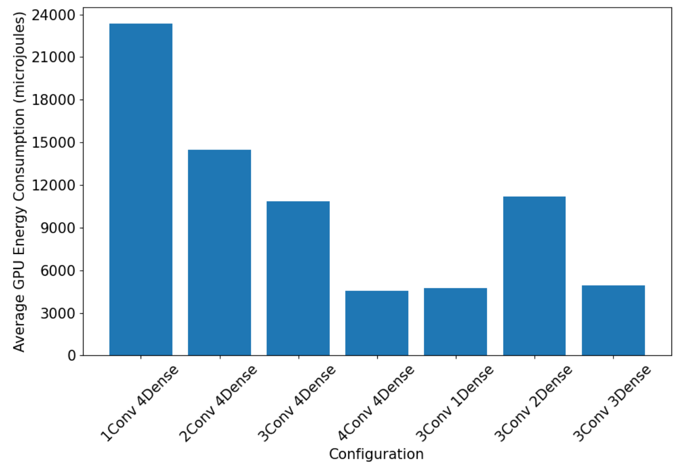
[4] M. S. Allauddin, G. S. Kiran, G. R. Kiran, G. Srinivas, G. U. R. Mouli, and P. V. Prasad, “Development of a surveillance system for forest fire detection and monitoring using drones,” in *IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2019, pp. 9361–9363.

[5] S. Biswas, R. Muttangi, H. Patel, and S. Prince, “Edge ai based autonomous uav for emergency network deployment: A study towards search and rescue missions,” in *2022 International Conference on Wireless Communications Signal Processing and Networking (WiSPNET)*. IEEE, 2022, pp. 268–272.

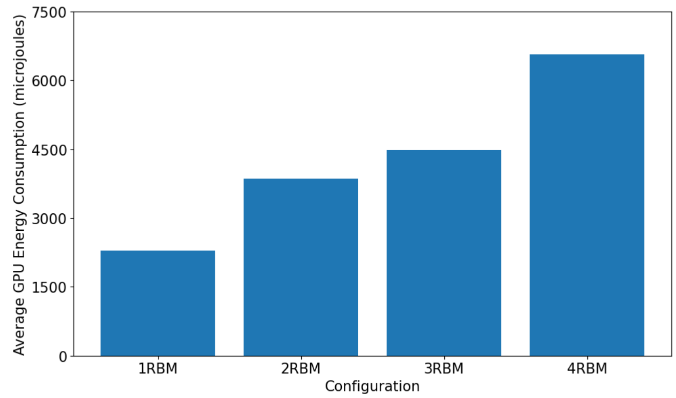
[6] M. M. Fouda, S. Sakib, Z. M. Fadlullah, N. Nasser, and M. Guizani, “A lightweight hierarchical ai model for uav-enabled edge computing with forest-fire detection use-case,” *IEEE Network*, vol. 36, no. 6, pp. 38–45, 2022.

[7] A. I. Abubakar, I. Ahmad, K. G. Omeke, M. Ozturk, C. Ozturk, A. M. Abdel-Salam, M. S. Mollel, Q. H. Abbasi, S. Hussain, and M. A. Imran, “A survey on energy optimization techniques in uav-based cellular networks: From conventional to machine learning approaches,” *Drones*, vol. 7, no. 3, 2023. [Online]. Available: https://www.mdpi.com/2504-446X/7/3/214

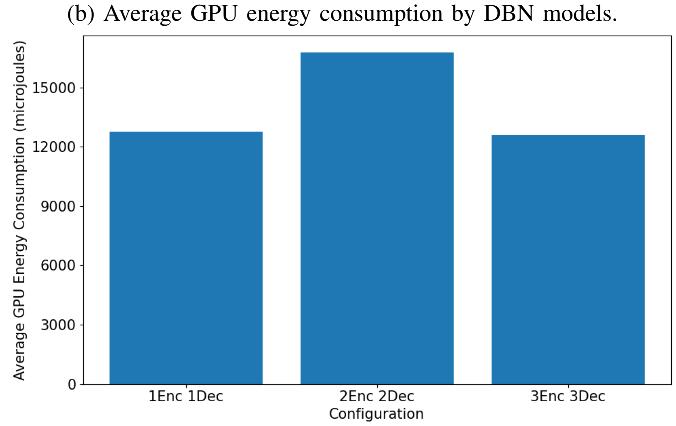
[8] A. Shamsoshoara, F. Afghah, A. Razi, L. Zheng, P. Fulé, and E. Blasch, “The flame dataset: Aerial imagery pile burn detection using drones (uavs),” *IEEE DataPort: New York, NY, USA*, 2020.



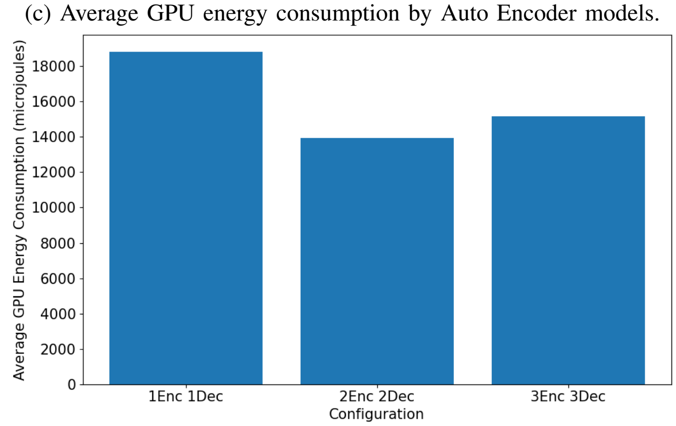
(a) Average GPU energy consumption by CNN models.



(b) Average GPU energy consumption by DBN models.



(c) Average GPU energy consumption by Auto Encoder models.



(d) Average GPU energy consumption by U-Net models.

Fig. 10: Average GPU energy consumption by different models (lower is better).

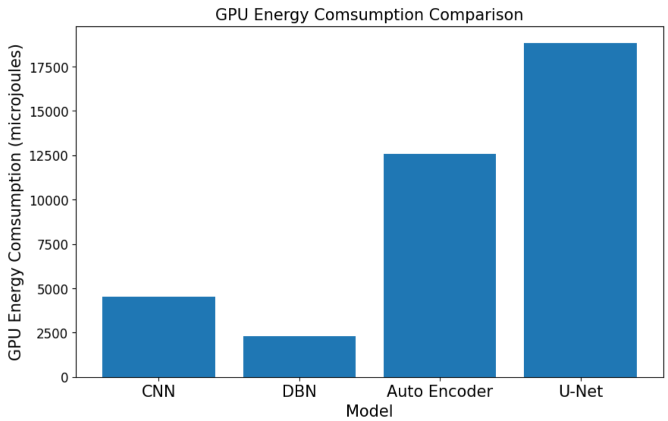


Fig. 11: GPU energy consumed by the most optimized architecture with the best accuracy results for different types of models (lower is better).

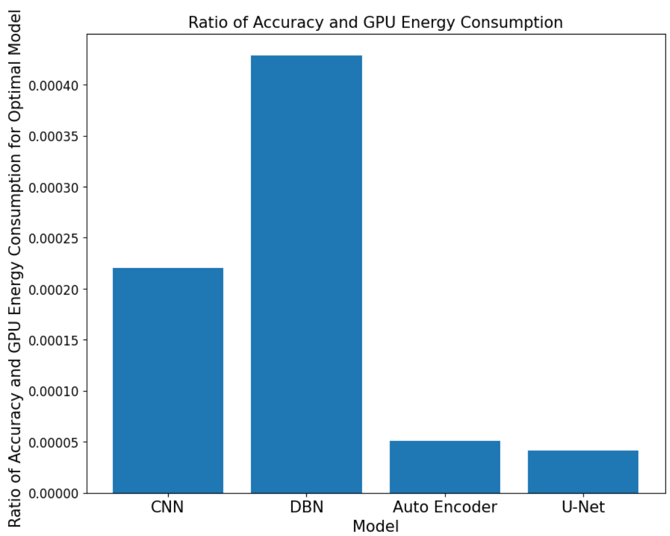


Fig. 12: Ratio of best accuracy and GPU energy consumption of the most optimized architecture of different types of models (higher is better).

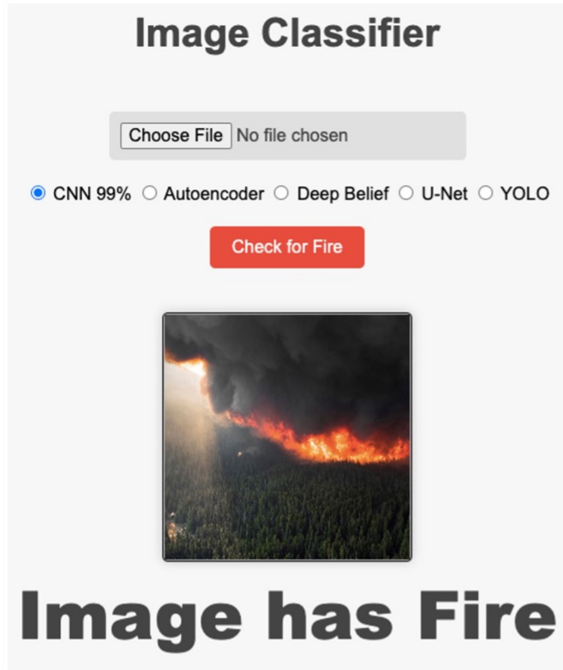


Fig. 13: Web interface.

[9] P. L. Guth, T. Craven, T. Chester, Z. O'Leary, and J. Shotwell, "Fire location from a single osborne firefinder and a dem," in *ASPRS 2005 Annual Conference Geospatial Goes Global: From Your Neighborhood to the Whole Planet*, 2005.

[10] A. E. Çetin, K. Dimitropoulos, B. Gouverneur, N. Grammalidis, O. Günay, Y. H. Habiboǧlu, B. U. Töreyin, and S. Verstockt, "Video fire detection–review," *Digital Signal Processing*, vol. 23, no. 6, pp. 1827–1843, 2013.

[11] K. Dimitropoulos, O. Gunay, K. Kose, F. Erden, F. Chaabene, F. Tsalakanidou, N. Grammalidis, and E. Çetin, "Flame detection for video-based early fire warning for the protection of cultural heritage," in *Progress in Cultural Heritage Preservation: 4th International Conference, EuroMed 2012, Limassol, Cyprus, October 29–November 3, 2012. Proceedings 4*. Springer, 2012, pp. 378–387.

[12] T.-H. Chen, P.-H. Wu, and Y.-C. Chiou, "An early fire-detection method based on image processing," in *2004 International Conference on Image Processing, 2004. ICIP'04.*, vol. 3. IEEE, 2004, pp. 1707–1710.

[13] V. Cantoni, R. Creutzburg, S. Levialdi, and G. Wolf, *Recent issues in pattern analysis and recognition*. Springer Science & Business Media, 1989, vol. 399.

[14] T. Celik and H. Demirel, "Fire detection in video sequences using a generic color model," *Fire safety journal*, vol. 44, no. 2, pp. 147–158, 2009.

[15] T. Celik, "Fast and efficient method for fire detection using image processing," *ETRI journal*, vol. 32, no. 6, pp. 881–890, 2010.

[16] G. Marbach, M. Loepfe, and T. Brupbacher, "An image processing technique for fire detection in video images," *Fire safety journal*, vol. 41, no. 4, pp. 285–289, 2006.

[17] I. Sfiligoi, D. Bradley, B. Holzman, P. Mhashilkar, S. Padhi, and F. Wurthwein, "2009 wri world congress on computer science and information engineering," 2009.

[18] H. Yamagishi and J. Yamaguchi, "Fire flame detection algorithm using a color camera," in *MHS'99. Proceedings of 1999 International Symposium on Micromechatronics and Human Science (Cat. No. 99TH8478)*. IEEE, 1999, pp. 255–260.

[19] F. Mohammadi, M. P. Bavaghar, and N. Shabanian, "Forest fire risk zone modeling using logistic regression and gis: an iranian case study," *Small-scale Forestry*, vol. 13, pp. 117–125, 2014.

[20] M. Bisquert, E. Caselles, J. M. Sánchez, and V. Caselles, "Application of artificial neural networks and logistic regression to the prediction of forest fire danger in galicia using modis data," *International Journal of Wildland Fire*, vol. 21, no. 8, pp. 1025–1029, 2012.

[21] K. Vani *et al.*, "Deep learning based forest fire classification and detection in satellite images," in *2019 11th International Conference on Advanced Computing (ICoAC)*. IEEE, 2019, pp. 61–65.

[22] M. A. I. Mahmoud and H. Ren, "Forest fire detection and identification using image processing and svm," *Journal of Information Processing Systems*, vol. 15, no. 1, pp. 159–168, 2019.

[23] S. T. Seydi, V. Saeidi, B. Kalantar, N. Ueda, and A. A. Halin, "Fire-net: A deep learning framework for active forest fire detection," *Journal of Sensors*, vol. 2022, pp. 1–14, 2022.

[24] V. E. Sathishkumar, J. Cho, M. Subramanian, and O. S. Naren, "Forest fire and smoke detection using deep learning-based learning without forgetting," *Fire ecology*, vol. 19, no. 1, pp. 1–17, 2023.

[25] K. Muhammad, J. Ahmad, I. Mehmood, S. Rho, and S. W. Baik, "Convolutional neural networks based fire detection in surveillance videos," *Ieee Access*, vol. 6, pp. 18 174–18 183, 2018.

[26] Y. Luo, L. Zhao, P. Liu, and D. Huang, "Fire smoke detection algorithm based on motion characteristic and convolutional neural networks," *Multimedia Tools and Applications*, vol. 77, pp. 15 075–15 092, 2018.

[27] M. A. Akhloufi, R. B. Tokime, and H. Elassady, "Wildland fires detection and segmentation using deep learning," in *Pattern recognition and tracking xxix*, vol. 10649. SPIE, 2018, pp. 86–97.

[28] S. Frizzi, R. Kaabi, M. Bouchouicha, J.-M. Ginoux, E. Moreau, and F. Fnaiech, "Convolutional neural network for video fire and smoke detection," in *IECON 2016-42nd Annual Conference of the IEEE Industrial Electronics Society*. IEEE, 2016, pp. 877–882.

[29] K. Muhammad, J. Ahmad, Z. Lv, P. Bellavista, P. Yang, and S. W. Baik, "Efficient deep cnn-based fire detection and localization in video surveillance applications," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 7, pp. 1419–1434, 2018.

[30] Z. Jiao, Y. Zhang, J. Xin, L. Mu, Y. Yi, H. Liu, and D. Liu, "A deep learning based forest fire detection approach using uav and yolov3," in *2019 1st International conference on industrial artificial intelligence (IAI)*. IEEE, 2019, pp. 1–5.

[31] J. Zhen-cun, W. Xiao-Jing, D. Zheng-xin, S. Yi-jie, and J. Wen-ping, "Research on fire detection of improved vgg16 image recognition based on deep learning," *Fire Science and Technology*, vol. 40, no. 3, p. 375, 2021.

[32] H. Jabnouni, I. Arfaoui, M. A. Cherni, M. Bouchouicha, and M. Sayadi, "Resnet-50 based fire and smoke images classification," in *2022 6th International Conference on Advanced Technologies for Signal and Image Processing (ATSIP)*. IEEE, 2022, pp. 1–6.

[33] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1–9.

[34] A. Biswas, S. K. Ghosh, and A. Ghosh, "Early fire detection and alert system using modified inception-v3 under deep learning framework," *Procedia Computer Science*, vol. 218, pp. 2243–2252, 2023.

[35] L. Zhao, L. Zhi, C. Zhao, and W. Zheng, "Fire-yolo: a small target object detection method for fire inspection," *Sustainability*, vol. 14, no. 9, p. 4930, 2022.

[36] R. Fan and M. Pei, "Lightweight forest fire detection based on deep learning," in *2021 IEEE 31st International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, 2021, pp. 1–6.

[37] U. Nepal and H. Eslamiat, "Comparing yolov3, yolov4 and yolov5 for autonomous landing spot detection in faulty uavs," *Sensors*, vol. 22, no. 2, p. 464, 2022.

[38] A. Jadon, M. Omama, A. Varshney, M. S. Ansari, and R. Sharma, "Firenet: a specialized lightweight fire & smoke detection model for real-time iot applications," *arXiv preprint arXiv:1905.11922*, 2019.

[39] B. Kim and J. Lee, "A video-based fire detection using deep learning models," *Applied Sciences*, vol. 9, no. 14, p. 2862, 2019.

[40] P. Barmpoutis, K. Dimitropoulos, K. Kaza, and N. Grammalidis, "Fire detection from images using faster r-cnn and multidimensional texture analysis," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 8301–8305.

[41] D. Sheng, J. Deng, W. Zhang, J. Cai, W. Zhao, and J. Xiang, "A statistical image feature-based deep belief network for fire detection," *Complexity*, vol. 2021, pp. 1–12, 2021.

[42] R. Kaabi, M. Sayadi, M. Bouchouicha, F. Fnaiech, E. Moreau, and J. M. Ginoux, "Early smoke detection of forest wildfire video using deep belief network," in *2018 4th International Conference on Advanced Technologies for Signal and Image Processing (ATSIP)*. IEEE, 2018, pp. 1–6.

[43] F. Yuan, L. Zhang, X. Xia, B. Wan, Q. Huang, and X. Li, "Deep smoke segmentation," *Neurocomputing*, vol. 357, pp. 248–260, 2019.

[44] D. Filatov and G. N. A. H. Yar, "Forest and water bodies segmentation through satellite images using u-net," *arXiv preprint arXiv:2207.11222*, 2022.

[45] J. Zhang, H. Zhu, P. Wang, and X. Ling, "Att squeeze u-net: A lightweight network for forest fire detection and recognition," *IEEE Access*, vol. 9, pp. 10 858–10 870, 2021.

[46] S. Cheng, J. Ma, and S. Zhang, "Smoke detection and trend prediction method based on deeplabv3+ and generative adversarial network," *Journal of Electronic Imaging*, vol. 28, no. 3, pp. 033 006–033 006, 2019.

[47] S. Aslan, U. Güdükbay, B. U. Töreyin, and A. E. Cetin, "Early wildfire smoke detection based on motion-based geometric image transformation and deep convolutional generative adversarial networks," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 8315–8319.

[48] Z. A. Khan, T. Hussain, F. U. M. Ullah, S. K. Gupta, M. Y. Lee, and S. W. Baik, "Randomly initialized cnn with densely connected stacked autoencoder for efficient fire detection," *Engineering Applications of Artificial Intelligence*, vol. 116, p. 105403, 2022.

[49] W. S. Mseddi, R. Ghali, M. Jmal, and R. Attia, "Fire detection and segmentation using yolov5 and u-net," in *2021 29th European Signal Processing Conference (EUSIPCO)*. IEEE, 2021, pp. 741–745.

[50] H. U. A. Tahir, A. Waqar, S. Khalid, and S. M. Usman, "Wildfire detection in aerial images using deep learning," in *2022 2nd International Conference on Digital Futures and Transformative Technologies (ICoDT2)*. IEEE, 2022, pp. 1–7.

[51] A. Canziani, A. Paszke, and E. Culurciello, "An analysis of deep neural network models for practical applications," *arXiv preprint arXiv:1605.07678*, 2016.

[52] R. Desislavov, F. Martínez-Plumed, and J. Hernández-Orallo, "Compute and energy consumption trends in deep learning inference," *arXiv preprint arXiv:2109.05472*, 2021.

[53] G. L. James, R. B. Ansaf, S. S. Al Samahi, R. D. Parker, J. M. Cutler, R. V. Gachette, and B. I. Ansaf, "An efficient wildfire detection system for ai-embedded applications using satellite imagery," *Fire*, vol. 6, no. 4, p. 169, 2023.

[54] C. Jean-Quartier, K. Bein, L. Hejny, E. Hofer, A. Holzinger, and F. Jeanquartier, "The cost of understanding—xai algorithms towards sustainable ml in the view of computational cost," *Computation*, vol. 11, no. 5, p. 92, 2023.

[55] K. Y.-W. Hong, H. S.-H. Chung, Y. Xiang, and A. W.-L. Lo, "Experimental assessment of a tiny ai-empowered output filter parameter extraction framework for digital power," *IEEE Access*, vol. 11, pp. 14 728–14 741, 2023.

[56] A.-r. Mohamed, G. Hinton, and G. Penn, "Understanding how deep belief networks perform acoustic modelling," in *2012 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2012, pp. 4273–4276.