Int. J. Advanced Networking and Applications
Volume: 17, Issue: 02, Pages : 6873-6884 (2025) ISSN: 0975-0290
https://doi.org/10.35444/IJANA.2025170210

Smart Aerial Image Classification using Deep Learning for Drone-Based Emergency Monitoring

Amira Hassan Abed
Faculty of Computers & Information Systems,
Egyptian Chinese University, Cairo, Egypt

----------ABSTRACT----------

DL can provide cutting-edge accuracy for technologies that utilize remote sensing including unmanned aerial vehicles (UAVs), especially enhance the remote detection abilities in responding to emergencies along with catastrophe control activities. In particular, "UAVs" that have sensors for cameras are able to function online difficult accessing disaster-affected places, process images, also notify people in the event of various catastrophes like collapsed buildings, floods, or fire. The goal is to lessen the detrimental impact of disasters on the natural world and the human population as quickly as possible. DL integration introduces high processing demands that prevent the implementation of DNN for different cases where low-latency restrictions on inference are required for making critical decisions in real-time. Consequently, our interest of this work is the efficient classification of aerial photos taken while a UAV is in flight for emergency response and surveillance purposes. Specifically, an Aerial Photos Collection for response to emergencies systems is proposed and a comparison between many current algorithms has been done. That processing is used to propose EmergencyRes, a CNN design. With lower than 1% accuracy decrease in comparison to the most advanced techniques, it can achieve up to 20x greater performance than currently available models and operate efficiently on low-power existing platforms. It can also handle multi-resolution features utilizing atrous convolutions.

Keywords: UAVs, video & image processing, emergency monitoring, drones, Deep learning (DL) and convolutional neural networks.

Date of Submission: July 20, 2025 Date of Acceptance: August 27, 2025

I. INTRODUCTION

The "UAVs" and drones have recognized as a remotely sensing system for many real-world uses, including surveillance of traffic [1], rescuing efforts [2], accurate farming [3], and satellite image processing [4]. New "UAV" applications, including the ability to identify, track, as well as evaluate both proactive and passive dangers and risks during event images (for instance, flames observes in forests, floods danger, roads crashes, and landslip prone areas), are made possible by recent advances in technology, like the inclusion of camera sensors [5]. Furthermore, because of their compact size, "UAVs" may be quickly deployed, allowing them to be present for crucial actions that enhance resource management and enhance assessment of risks, reduction, and avoidance [2]. However, there are several limitations which have to be addressed since a "UAV" must operate in disaster-affected areas where expensive equipment is missing and remote access to cloud services might not be feasible to create. As a result, greater independence is needed to guarantee immediate evaluation and efficiency in operation. "UAVs" that operate autonomously can rely only on their own sensors and microprocessors to complete tasks without transmitting their data to a base station. Moreover, "UAVs" independent functioning — which combines automated on-board visual processing with path planning algorithms— permits them to more swiftly occupy a greater region [6]. On the other hand, because of the "UAVs'" limited payload capacity restricted computing capacity and low-power limits provide unique hurdles for on-board processing. Therefore, a critical factor in allowing autonomous "UAVs" to identify threats at incident scenes instantly is the computing effectiveness of the underlying computer vision system. CNNs, one of the popular DL techniques, have demonstrated outstanding results in several applications and are widely acknowledged as a major solution for various computer applications [7]–[10]. Therefore, there are numerous advantages to employing DL techniques in disaster mitigation and emergency response applications. These advantages include the ability to quickly retrieve vital information, improve readiness and response in urgent situations, and facilitate the processes of decision-making [11]. Previous research has shown that transfer learning, which uses a pre-trained CNN for the features extraction and a number of layers stated on above for carrying out the tasks of classification, can help DL approaches outperform conventional techniques for machine learning with features that have been handcrafted [12].

Although CNNs are getting more and more efficient at classification tasks, their interpretation rate on integrated minimal power consumption systems, such as ones "on-board UAVs", is limited by their highly computational cost [13]. This can be particularly true if you have to use the same platform to do numerous vision activities. These considerations (privacy, latency, or working in remote locations having minimal or no bandwidth) make locally integrated computing near the sensor more desirable for numerous applications than cloud analytics. [14, 15] Furthermore, specifically designed tiny networks can offer the degree of precision as well as efficiency required for specialized applications in which there are computational

limitations and sparse data availability. In addition to their computing effectiveness, they undergo training iterations more quickly and are easier to update over-the-air. Therefore, a UAV becomes a highly appealing and logical alternative to typical techniques when it is equipped with a tiny CNN which considered efficient in processing at edge for classifying the aerial scene on board. The goal here, as discussed in this article, is to automatically classify aerial images that the UAV records into a semantic label [16]. When it comes to emergency response softwares, these designations match actual dangers or hazards. To improve readiness and offer quick situational awareness, a device like this can be installed on UAVs for continual tracking and inspection. This particular use-case involves a UAV that follows a predefined way and analyses the becoming frames from the camera continuously, via its embedded platform. It then sounds an alert for any potentially hazardous situations or hazards it recognizes. Our aim in this study concentrates on enhancing immediate time perceptual abilities in such events by developing an CNN-based system which provides the best combination of precision and performance, and is capable of running on hardware that is incorporated on the UAV. By proposing a portable CNN constructed using depth-wise atrous convolutions, called EmergencyRes, the initial effort in [17] has been greatly improved upon. This CNN is able to get many characteristics in a computationally effective path, controlling to support a suitable compromise across precision [18] and rapidity for the challenge of responding to emergency tracking via UAVs. In addition, the dataset has been greatly expanded, a thorough comparison with other models is conducted, and additional optimizations are suggested to enhance efficiency for certain scenarios. The following is a summarization of this article's primary contributions.

1) A specialized database with more photos than current datasets, intended for the use of aerial image1 categorization in emergency response.

2) A brand-new, computationally effective CNN (called EmergencyRes) is presented, combining multiple resolution depth-wise convolutions to deliver near-recent accuracy (95.7%) and more than 20× quicker processing times, making it appropriate for low-cost, low-power devices.

3) Examine and compare the effects of various CNN designs on the job of classifying aerial scenes of disasters and assessing their accuracy, rate of deduction, along with storage.

4) A real trial setup made up of a Smartphone base station, an embedded computer platform, and a UAV with two analyzing choices was used to evaluate the various models.

This is how the article is organized. In Part II, the architectures of CNN is presented, along with an overview of earlier research in the field of aerial picture categorization for disaster relief and emergency response. In Part III, the methods utilized to create the suggested network and the dataset that was created for the purpose of classifying aerial images of catastrophes are described in depth. In Part IV, several models and methodologies are analyzed and evaluated using an actual practical UAV platform. The final part offers closing thoughts and explores potential areas for further research in this field.

II. Related Work and Background

A. CNN Design Methodologies and Architectures

CNNs are methods of machine learning with roots in biology that could possibly be trained to carry out a range of recognition of images, authentication, and categorization tasks. Stochastic gradient descent and back-propagation are used for training each of the layers which make up CNNs to learn hierarchical representations of visual data. The deep learning community has put out a number of architectures in an effort to improve upon previous efforts in image categorization. A CNN's structure usually consists of a feature extractor stage and a classification layer. Here are a few of the most significant architectures.

VGG16 [17]: When removing CNN elements from photos, the VGG network has grown in popularity. This specific network has 16CONV/FC layers and is aesthetically pleasing in its simplicity. The final structure is only two 4096-neuron fully-connected layers and three × three convolutional layers stacked over upon each other, with two pooling layers lowering the size of the feature maps by a factor of two. The final thick layer is equal to the total number of classes employed in the entire classification. The VGGNet has several drawbacks, including higher evaluation costs and a high parameter and memory use.

ResNet [19]: it introduces the concept of residual learning to train even more complex CNNs. Essentially, the network learns residuals by propagating and adding the input of a convolution layer to its output after the operation. But the precision it gains comes at the expense of running times and memory requirements.

The primary contribution of the design is seen in Inception [20] and Xception [21], where several convolution filters are combined to create a multilayer feature extractor. Before being sent into the following layer, the outcomes from these filtration at the identical network layer gets arranged across the dimension of the channel. In the network, this module is known as the inception module. The Inception V3 design is derived from the work of [20], who suggested modifications to the original Inception module in order to increase accuracy even further. At 96 MB, the weights of Inception V3 are less than those of VGG and ResNet, but they are still deemed excessively huge for embedded applications. In an effort to reduce the complexity of computation, a more optimized version of the Inception family known as Xception was created, which included the concept of separable convolutions. A depth-wise separable convolution is created by separating a convolution and applying a separate filter to each channel. Point-wise convolutions are then used for combining the resulting sets of data.

MobileNet [22]: The MobileNet series of algorithms achieves the latest in performance at a lower computing cost by utilizing the concept of separable convolutions. In essence, instead of the more intricate inception module, the structure is a simplified variant of the Xception network which performs just one filter at every input. As a result, optimizing and parameterizing this approach for mobile apps is simple. The creators of the most recent version of MobileNets [22] use neural search methods to

better optimize the design of the model, particularly for semantically segmenting tasks.

SqueezeNet [23]: This CNN is made to be efficient with parameters. It makes use of a fire module to blend the output from several convolutional layers. Additionally, it has a bottleneck layer which lowers the input feature map's dimensionality, which lowers computing costs.

ShuffleNet [24]: This method lowers the network's parameter count by employing many filter groups, each of which focuses on a distinct channel collection from the input feature map. It then makes use of a technique known as channel shuffling to relate the various channels together.

EfficientNet [25]: To improve accuracy and efficiency, a novel CNN scaling method is presented in this paper that evenly scale every dimension of depth, breadth, and resolution in a logical way. This method works with any backbone architecture, including ResNets and MobileNets. Furthermore, an optimized architecture is automatically found through the usage of the auto-ML framework. All things considered, EfficientNets have been shown to outperform bigger models like VGG with orders of magnitude less parameters and FLOPS. However, only cloud-centric processing systems were included in the evaluation of EfficientNets.

B. Image Categorization for Disaster and Emergency-Response

This section describes some pertinent works that address the aerial picture classification problem for responding to emergencies and disaster management. A few of these works also focus on using UAVs for remote sensing. Over time, a variety of techniques have been put forth to identify different types of disasters in images. [26] These include pixel-level classification performed by image processing based on thresholds, models based on Gaussian mixtures that need to be empirically tuned, as well as support vector machines, which are deemed to be inefficient for applications that operate in real time. [27, 28] The research community is now looking at CNNs and DL's potential for various image processing tasks due to their success in these areas. These methods were originally put out for use with ground robots, as in the work in [29], and then they were applied to the interpretation of aerial photos [30]. Because of its greater classification accuracy and generalization skills, deep learning has become a popular method for classifying aerial images for responding to emergencies and disaster management applications. The authors of [31] suggest using "UAVs" to detect fires using a deep learning method based in the cloud. The detection uses a bespoke CNN that is trained to distinguish between fire and non-fire pictures with a resolution of 128 x 128 and has a structure akin to that of VGG16. The algorithm is run on a workstation equipped with a GPU Titan Xp GPU once the video feed from a UAV is transmitted to it. In [32], an approach for locating items of interest among avalanche debris uses a pre-trained inception network for extraction of features and an SVM with linearity for classification. They provide a picture segmentation methodology that serves as a preparation method to split the image in portions via a sliding window. The basis for this technique is the difference in colour between the object of interest and the background. Additionally, they employ post processing to improve a classifier that makes employing hidden Markov models in terms of performance. With a clock speed of 3 GHz and 8GB RAM, the program runs on a desktop computer rather than an embedded device averaging 5.4 frames per second (FPS) for 224 by 224 pictures. The range of accuracy was 72%–97%.

In a similar vein, [33]'s study focuses on deep learning applications for fire detection. To train fire detection systems, two pre-trained CNNs—VGG16 and Resnet50—are specifically employed and contrasted as foundation architectures. To test the classification accuracy, fully connected layers are added to the architectures once the features are extracted. For a bespoke database, the various models have an average accuracy of 91% and a median computation time of 1.35 s using an Intel GTX 820 GPU. The study in reference [34] suggests a method for using a UAV platform to identify wildfires. The overall method consists of a CNN named FireNet with 15 layers that recognizes fire in 128 × 128 resolution photos. Its design is comparable to the VGG16 network and includes 8 convolutional, 4 maximum pooling, and 2 fully interconnected layers. Alongside it is a region suggestion approach that uses higher resolution photos to extract image areas for neural network classification. The system was trained on an Intel 840M GPU; for 128 x 128 resolution pictures, and without accounting for the cost of the region proposal and area selection algorithms, the overall accuracy is 98%, as well as the average performance is 24 frames per second on the specific GPU platform. In terms of application domain, the work of [35] may be the most pertinent as it trains a deep neural network (DLN) to categories aerial photographs into one of five groups that correlate to natural disasters. A fully connected network is layered on top of the VGG network, which serves as the foundation for feature extraction, in order to carry out transfer learning for the new job. For a bespoke test set, an accuracy of 91% is attained, and processing a 224 x 224 picture takes less than 3 seconds on average. The goal typically focuses on increasing the independence and actual processing capabilities of the UAV on-board system by adding extra limitations to it, even if the application is comparable. Using DCNN algorithms as the primary classifying strategy, previous systems for aerial scene in managing emergencies and catastrophe with UAVs have demonstrated very significant results, according to the literature study. For the purpose of classifying a particular event, they mostly employ pre-trained networks that are already in place and adapt through transfer learning. The primary computational platform used to remote processing the UAV data on GPUs is desktop-class PCs. But in other cases, connection problems and communication delay might make these systems perform worse, requiring the UAV to have more autonomy and on-board computing power [36]. Additionally, the computational constraints of embedded devices make it impractical to employ current methods designed for desktop-class computers. The assessment of current methods for the challenge of classifying aerial image for responding to emergencies, as well as the launch of an effective CNN appropriate for embedded systems (like UAVs that is capable of on-board classification for

several disasters), constitute this article's contribution to the state-of-the-art.

II. DL for Classifying Aerial Disaster Events

The method for creating an efficient CNN which would be applied for classifying aerial photos taken by a UAV for emergencies response applications is described in this section. In particular, it describes the process of gathering the training set for this task, the many networks utilized for comparison and analysis, both through transfer learning of pre-trained networks and new networks, and the decisions taken in their design. Lastly, the specifics of the training procedure are given.

A. Collection of Datasets

Gathering an appropriate dataset is the initial step in training a CNN for aerial image classification applications related to emergency response and disaster management. As far as we are aware, there isn't a publicly accessible and extensively utilized dataset designed for emergency response purposes. Because of this, a special database known as AIDER is created specifically for this purpose. In order to produce the dataset, all photos for four catastrophic events—fire, flood, collapsed building, and traffic accidents—as well as one class for the normal scenario, had to be manually gathered. Images with comparable visual characteristics, like smoke and flames, are grouped together. Although conceivable, a more precise categorization is left for later research. Using the terms ""UAV" or "Drone" and an incident like "Fire," "Earthquake," "Highway accident," etc., the aerial images for above events are collected from a variety of open accessed datasets. Before being trained, images are first standardized despite having varying sizes. Every picture underwent a human inspection process to ensure that the event of interest was present and centred inside the image to prevent its removal from the image view during any geometric alterations during augmentation. The multiple catastrophic occurrences were recorded during the data gathering procedure at varying resolutions and in varied lighting and perspective conditions. In order to accurately represent real-world situations, the dataset is unbalanced, with a greater proportion of photos belonging to the no-risk class. [37] Naturally, it may affect on the training to be more difficult, but throughout the training, a suitable technique is used to counter this, and it will be covered in depth in the upcoming parts. [38] Since the environment might affect the UAV's operating circumstances, it's critical that the dataset includes a variety of "clean" and "clear" picture types. Furthermore, gathering data may be costly and time-consuming. Therefore, before adding each image to the batch for training, a number of random augmentations are performed to it in a probabilistic manner in order to further improve the dataset. [39] These include specifically picture manipulations including lighting adjustments, colour transferring, blending, grinding, and shadowing, as well as geometric transformations like rotations, translations, cropping, and zooming on the horizontal axis. [40] Furthermore, sample pairing is utilized to combine photos and improve the training set even more [41]. To prevent the network from capturing the augmentation qualities as a dataset characteristic, each transformation is done with a random probability that is carefully calibrated to guarantee that not all photos in a training batch are changed. [42] Fighting over-fitting and raising training size variability are the goals of all these modifications in order to boost generalization capability. In general, over 17× more data were gathered than in previous research that take into account multiclass issues (e.g., [43]). While there are less photos in the dataset than in popular benchmarks like ImageNet and CIFAR, it is more harder to come across such real-life situations and obtain sufficient data. Therefore, augmentation techniques were applied in order to further improve the original dataset. We believe that this designates AIDER (Table I) as a useful supplementary data source for creating and comparing data-driven approaches for applications related to emergency response and catastrophe monitoring.

Table I: AIDER cases

Classes	Dataset		
	Training	validating	testing
Building collapsing	480	180	280
Fires	500	190	290
Flooding	480	180	280
Accidents of traffic	480	180	280
Normal	2900	1100	2100

B. CNNs for Classifying Airborne Disasters

In order to determine the optimal CNN structure for classifying aerial images, two distinct methodologies were used to create several networks. Investigating the trade-offs between these networks' accuracy and performance is the method' main goal. Initially, the networks described in Section II-A are trained by transfer learning, which is consistent with the approach taken in previous studies. Moreover, particularly for this job, new network structures have been developed and trained from scratch. The latter method is justified by the fact that it enables the design decisions that result in faster, more accurate networks while maintaining the accuracy of bigger networks, making them more efficient on embedded systems.

1) Transfer Learning Networks: Various networks (VGG16, ResNet50, FireNet, MobileNets, and SqueezeNet are used for transfer learning. Many related works have also made use of most of these networks. For

Int. J. Advanced Networking and Applications
Volume: 17, Issue: 02, Pages : 6873-6884 (2025) ISSN: 0975-0290
https://doi.org/10.35444/IJANA.2025170210

each of these networks, the feature extraction portion is frozen once the input picture has undergone all required preprocessing. A classification layer is then added on top, in a manner similar to previous research. Unlike previous efforts, a softmax classification layer is applied at the conclusion of the process after a global (per feature-map) average-pooling layer has been applied before the dense layers. Average pooling has been demonstrated to function just as well as traditional techniques, while also reducing the number of parameters and the ensuing computing and storage demands. Because the networks utilized in published research for this purpose include fully connected layers, the pre-trained models that were utilized for evaluation are therefore by nature more effective with respect to of memory and operations.

2) Custom Networks: according to how closely the dataset resembles the large-scale dataset being utilized (such as ImageNet), the pre-trained networks have varying degrees of sensitivity. Transfer learning can lead to larger and deeper networks, but they might not be appropriate for resource-constrained systems like UAV platforms, which have restrictions on platform size, weight, and power envelope. Modifying the architecture of current networks is also cumbersome since it necessitates repeating the pre-training on a big dataset, which is computationally expensive. To get over the aforementioned restrictions, it is necessary to create specialized networks that are computationally efficient by nature. [45] By concentrating on the layer designs, type, along with connection, the design space is investigated. As a result, several networks are created to help comprehend the trade-offs associated with the design decisions. Several consistent design decisions are taken for each of the various network setups. An architecture that uses the fusion of features from atrous convolutions with varying degrees of dilatation as its fundamental building block is devised, and these designs are mostly applied to segmentation tasks. They are used here to more effectively learn characteristics at several resolutions simultaneously, enabling UAV applications that encounter interesting locations at different sizes. With the same resolution and fewer parameters, the suggested architecture enables customizable grouping of the multi-scale context data. We next go into depth about the atrous convolutional feature fusion block.

C. Atrous Convolutional Feature Fusion (ACFF)

By controlling the dilation rate, which establishes the distance between the kernel points, atrous convolutions, also known as dilated convolutions [46], may efficiently increase their receptive field without adding more parameters. As a result, they can collect and convert pictures at various resolutions. As a result, it may be applied to provide further context to the model. The proposed block computes several such ACFF (U_d) across various dilation rates d. In order to decrease computational complexity, every atrous convolution is factored into depth-wise convolution, which applies a single convolutional kernel per input channel to accomplish lightweight filtering. [47] The various traits are then combined by some kind of fusion. The natural tendency is to use the various dilation rates since variations in object/region resolution might cause one approach to reveal details that another would have overlooked. It is significant to notice that each path learns unique weights wd that can be more beneficial; the weights are not shared across pathways. The fact that atrous convolutions need the same amount of calculations and parameters at every resolution is another benefit of employing them. Typically, each atrous convolution operates on the same feature map but at a different spatial resolution, beginning with a dilation rate of 1 and a filter size of 3x3 (i.e., no spacing) and increasing to 3, which is equivalent to a 7x7 receptive field (Figure I).

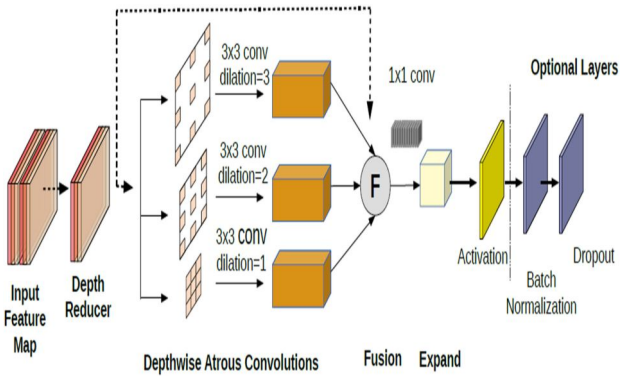

Fig. I: ACFF Structure [48]

This allows for different kernels to effectively look at the same area with fewer parameters used. Reducing the channel dimensions as well as the spatial size of feature maps is a crucial component of optimized CNNs. For this reason, the input feature map channels are separated in half before the atrous convolutions. Because of this, atrous convolution can have several branches without noticeably affecting performance. A hyper-parameter that may be adjusted further based on requirements is the depth reduction factor. A bottleneck layer is employed to reduce the overall amount of pathways after resizing without changing the feature maps' spatial scale. The atrous convolutional characteristics at different dilation rates have to be blended in order for the unit to learn across representations from a wide practical receptive field. After the fusion procedure, channel characteristics are combined nonlinearly and projected onto a higher dimensional space using 1×1 convolutions and activation. An activation and batch normalization come next for every atrous convolutional block.

$$\mathsf{U_d} = \sum_{i=1}^{M}\sum_{j=1}^{N} \mathsf{X}(m + d \times i, n + d \times j) \odot \mathbf{w_d}(i,j) \tag{1}$$

$$\mathsf{Z} = F_{\substack{1\le m\le M \\ 1\le n\le N \\ 1\le d\le N_d}}(\{\mathsf{U_d}(m,n)\}) \tag{2}$$

$d \in [1, \ldots, Nd]$.

D. Selections for Macro-architecture Design

Int. J. Advanced Networking and Applications
Volume: 17, Issue: 02, Pages : 6873-6884 (2025) ISSN: 0975-0290
https://doi.org/10.35444/IJANA.2025170210

A low-computational-complexity deep neural network that is appropriate for embedded systems is constructed using the ACFF macro-block as a foundation. **Table II** shows the total network structure that was designed using the following options.

1) Lower First Layer Cost: Because the first layer is applied to the whole image, it usually has the highest computational cost. As a result, only sixteen filters with a 3 × 3 spatial resolution are chosen. Here, low-level picture characteristics are more effectively captured by using a conventional convolution layer. Two strides are employed in order to minimize the calculations.
2) Early Down-sampling: All of the earliest layers undergo down-sampling. Stride and max-pooling layers are combined in an attempt to minimize information loss caused by aggressive striding, while the spatial resolution is still decreased. Empirical research revealed that reducing the size of feature maps in later phases led to a decline in accuracy; hence, the first four layers undergo down-sampling.
3) Canonical Architecture: A pyramid-shaped architecture is used for the CNN configuration in order to maintain representational expressiveness. This implies that as the depth of the feature maps increases, the spatial resolution of each layer is gradually reduced. Thousands of filtering at each layer are common in big networks, however this adds a significant burden for embedded systems. As a result, the first layer encounters Sixteen filters, and the number of filters increases from there, although never goes over 256, indicating the last layer before the categorization section.
4) Fully Convolutional Architecture: By eschewing the usage of fully linked layers, a quick and easy method is used to drastically lower the number of parameters and computing expense. Rather, the channels are reduced to the number of classes using a channel-wise 1 × 1 convolution. The per class feature maps are then summarized using a global average pooling operation and supplied to a softmax layer.
5) Capped Leaky ReLU: Because capped ReLU is resilient when used to low-precision computing, it is employed, like to [17], as the primary nonlinearity throughout the network. To be more precise, the output has an upper constraint of 255, which may be roughly represented by the integer 8 b. This particular ReLU variation does not take use of quantization, but it makes the network more receptive to future enhancements. Additionally, there are two ways that the modified capped leaky ReLU can function. The tiny percentage of gradient is allowed to flow throughout the training phase, but it is wiped out during the inference phase to allow for quantization if needed.
6) Regularization: Beyond augmentation to prevent over-fitting, extra regularization approaches are also included because of the dataset's smaller size in comparison to databases like ImageNet. Specifically, depth reduction techniques, 1 × 1 convolutions, and atrous convolutions are employed before batch-normalization [49]. The layers with the most parameters are followed by a dropout layer with a rate of 0.2. Similar to previous studies, it was discovered that L2 weight regularization, using a 5 × 10−4 parameter, was useful in accelerating the network's learning process and reducing error rates.
7) Network Depth: Large amounts of data are a prerequisite for deep networks, which are required to construct robust representations. Additionally, the computational cost of very deep networks is significant. Based on these two considerations, it was empirically discovered that a network size of six ACFF blocks had been adequate to reach accuracy levels comparable to the state-of-the-art; expanding it resulted in increased computing costs but no discernible accuracy gains.
8) Ignore Connection Fusion: The aggressive down sampling procedure may result in the loss of a substantial quantity of information. The input feature map is additionally incorporated into the fusion process in order to retain part of the original data.

The foundational EmergencyNet architecture depicted in **Table II** is constructed from the aforementioned combinations. In order to assess and contrast the tradeoffs, several approaches of fusing together the feature maps have been tested alongside base implementations of conventional convolution, depth-wise convolution networks, and spatially-separable convolutions. Section IV-B presents these findings.

Table II: EmergencyRes Model Structure

Layer	Size of outputs	Filters numbers	Stride
Input images	240*240		
Convolution	120*120	16	2
ACFF-Block-1	120*120	64	1
Max-pooling-1	60*60		2
ACFF-Block-2	60*60	96	1
Max-pooling-2	30*30		2
ACFF-Block-3	30*30	128	1
Max-pooling-3	15*15		2
ACFF-Block-4	15*15	128	1
ACFF-Block-5	15*15	128	1
ACFF-Block-6	15*15	256	1
Convolution	15*15	5	1
Global-pooling	5		
Softmax	5	5	

E. Testing and training

For equal circumstances and an equitable comparison throughout the inference phase, every network is tested and grown using the same training framework. All models are available through the Keras DL framework, which uses TensorFlow as the backend. When feasible, every network should utilize the same picture size (with the exception of the MobileNet V1 and V2 models, which need a smaller image size). As a result, a picture is first shrunk to the proper size based on the network before being enhanced and added to the batch (240 × 240 pixels is the default size that's usual for training CNNs). Although using higher picture sizes might result in slower inference times, this study does not examine the image size space; instead, the network architecture is the main emphasis. Dividing the set of data into training, validation, and testing sets is the first stage in the training process. [50] The training set receives the majority of the data, with the remaining data being divided in a ratio of 0.5, 0.2, and 0.3 amongst the other two sets. The Normal class is overrepresented in the dataset because, as was previously indicated, it is the dominant class. This is a reflection of actual

Int. J. Advanced Networking and Applications
Volume: 17, Issue: 02, Pages : 6873-6884 (2025) ISSN: 0975-0290
https://doi.org/10.35444/IJANA.2025170210

circumstances, but if left unchecked, it may cause issues in which the network overfits and labels everything as belonging to the majority class.
Under sampling the majority class and oversampling the minority classes in the same batch are done simultaneously to prevent problems caused by the imbalanced dataset. In order to do this, the steps listed in Algorithm 1 are followed to choose an equal number of photos from each class to create a batch, ensuring that every case is represented equally. This procedure is favoured in part because most of the photos were taken from movies. There's a reliance between pictures in close temporal proximity even with minimally sampled footage. Therefore, throughout the training process, it is less common to come across visually comparable photos thanks to random sampling. A GeForce Titan Xp, an Intel i7 − 7700K CPU, and 32GB of RAM were used to train each network. For training, a cosine learning rate annealed plan was employed along with the conventional Adam optimization technique [51]. Given that there were 300 epochs overall and a starting learning rate of 0.1, the learning rate LR at time t is determined using the formula in (3). Additionally, label smoothing with an epsilon of 0.1 is done during training. With a batch size of 64 and 60 iterations, each epoch produces 3840 enhanced photos in 1total.

$$LR(t) = 0.5 * (1 + \cos(((t * \pi) / (300)))) * 0.1. \quad (3)$$

IV. Results of the Experimental Evaluation

Results from the approach's testing on a real embedding system are shown in this part along with the experimental evaluation of the suggested technique. Initially the produced data is used to confirm the enhanced performance over the current networks, and the learning process' qualitative outcomes are shown. Additionally, real tests have been carried out in two distinct environments. First, there is aboard embedding processing, in which every calculation is done on a device that is embedded inside the UAV platform's limited resource pool. Second, processing that is done remotely, using an Android tablet to operate the UAV while transmitting the footage it has recorded to a controller ground station. Because this is typical in actual time streaming software in which the camera produces each picture repeatedly, it is important to note that the evaluation stage happens with a batch size of one. This is because the major interest is in the speed at which just one picture can be processed.

A. Measures of Performance

The resultant frame-rate, or FPS, attained by each model is a crucial performance indicator for real-time applications since it is inversely related to the amount of time required to analyze a single image frame from a video or camera stream. Furthermore, a simple accuracy measure (percentage of correctly classified examples), that has been employed within other publications, might be less suited to the exact issue regarded during this research since the typical scenario is typically much more frequent than other classes, given that previous variation over classes is significantly non-uniform. The F1 score is used as the learning performance parameter instead of this to prevent bias in our results. Here is a summary of the important metrics.
1) Frames-Per-Second (FPS): The pace that a machine learning algorithm can process receiving camera pictures, with ti being the amount of time it takes to process one picture.

$$\mathrm{FPS}_{\mathrm{model}} = \frac{1}{\frac{1}{N_{\mathrm{test\_samples}}} \times \sum_{i=1}^{N_{\mathrm{test\_samples}}} (t_i)}. \quad (4)$$

2) Mean F1 Score (F1): it measures the average accuracy for every class, scaled using the quantity of test cases for every label, represented by the symbol |Ci|. The balance across recall and accuracy is expressed by the F1 score. Generally speaking, having low false positives and low false negatives corresponds to having a high F1 score. This solves the issue of label imbalance and yields a performance metric that is better suitable for comparing different classes.

$$\overline{F1}_{\mathrm{model}} = \frac{2}{N_{\mathrm{classes}}} \times \sum_{i=1}^{N_{\mathrm{classes}}} \frac{\mathrm{prec}_i \times \mathrm{sens}_i}{\mathrm{prec}_i + \mathrm{sens}_i}. \quad (5)$$

B. Comparison and Analysis of the General Performance

The bespoke algorithms with various fusion strategies with the primary architect shown in Section III-B2 have been contrasted with networks made up of depth-wise and regionally separable convolutions and vanilla CNN representations of regular convolution networks. **Table III** summarizes the findings of comparing the various networks based on factor size that influences the model's learning complexity, accuracy, and memory for storing the model weights.

TABLE III: models comparison

Models	Factors	Memory	F1 score
Standard convolutional	720123	3.12	94.4
Depthwise-Separable	96112	0.401	92.01
Spatially-Separable	629320	2.688	93.61
Max-Fusion	91021	0.409	95.8
Average-Fusion	91021	0.409	94
Add-Fusion	91021	0.409	96
Concatenate Fusion	223501	0.975	95.3

In comparison to other models, the typical CNN requires more memory because to its greater parameter count. Among the conventional networks with the greatest cpu and parameter count requirements, it obtained the best accuracy. These requirements are lessened by the depth wise-separable convolution network, but the accuracy reduction is significant. The accuracy is increased by the regionally separated network, although more parameters are required. Conversely, the suggested networks using various fusion techniques offer better or comparable accuracy. The small number of parameters makes them memory- and computationally-efficient, and the combination of many features from various kernel sizes results in a notable improvement in accuracy. Among the ACFF-based networks, the maximal and add fusion models performed superior for this specific sample. This is

Int. J. Advanced Networking and Applications
Volume: 17, Issue: 02, Pages : 6873-6884 (2025) ISSN: 0975-0290
https://doi.org/10.35444/IJANA.2025170210

explained by the reality that they place greater focus on certain spatial places that aid in learning as opposed to average fusion, which may diminish some significant characteristics and suppress information. Furthermore, they do not result in an increase in memory or processing demands, as concatenation fusion does.

The primary design of EmergencyRes, which is utilized for evaluation compared to different networks in addition to results from experiments, is chosen from the top-performing networks with add fusion. **Table IV** presents an overview of the outcomes for every network. In order to quantify complexity as a platform-independent metric, comparisons are also done for these networks with regard to FLOPS. First, it is noted that VGG16 leads all other pre-trained models in terms of accuracy, with an F1 accuracy score of 96.4%, followed closely by ResNet50 with 96.1%. This is consistent with previous research utilizing these networks, which revealed accuracy levels of 81% − 98% for various scenarios and purposes. However, because to their extremely high computing and storage needs, these networks are not suited for applications that operate in real time or systems with limited resources. The most recent generation, V32, earns an accuracy rating 95.3% and is the most accurate of the MobileNet series of models. It also has the least FLOPs among the pre-trained models. But it needs an order substantial greater in RAM and variables. Even though both of the MobileNet editions (V1 and V2) had lesser picture sizes (224 × 224), their FLOP needs remain greater.

EfficientNet3's complex architecture allows for an impressive accuracy of 96.0%. Nevertheless, compared to EmergencyNet, there are more parameters and memory needs. Some networks need more FLOPs with more parameters and don't offer accurate enough results. This research makes it evident that, if one wants improve every area of design, it is worthwhile to look at custom-made solutions for limited applications. Compared to the most comparable structures, EmergencyNet delivers equivalent performance with less FLOPs and variables. It achieves 95.7% accuracy on the F1 score, extremely similar to pre-trained network techniques, and requires just 360 KB of RAM, 30% fewer than MobileNetV3. In regard to accuracy, EmergencyNet performs well than more modern models like EfficientNet and MobileNetV3, but with a lot fewer parameters. Further evidence of EmergencyNet's effectiveness in drastically lowering memory requirements for using classification of images over actual time incorporated rescue applications comes from its noticeably smaller model size when compared to other models. This also qualifies within the chip storage on low-power structures with memory limitations and for more specialized computing platforms like FPGAs that can restrict on-chip storage.

Table IV: models comparison

Models	Factors	Storage	F1 Score	Flops
EmergencyRes	91201	0.428 MB	96.7%	59
ResNet50	24224601	98.3 MB	97.2%	4644
VGG16	14972520	60.28 MB	97.6%	17831
Xception	21442416	86.47 MB	96.5%	428
MobileNet V1	3587628	14.3 M	96%	561
MobileNet V2	2696318	11.2 MB	96.2%	281
MobileNet V3	3132148	13.4 MB	96.4%	71
SqueezeNet	701340	3.6 MB	92.6%	855
ShuffleNet	4393536	17.6 MB	92%	562
EfficientNet	4489896	18.1 MB	97.1%	435
FireNet	5346971	5.7 MB	91%	1688

Using EmergencyNet's confusion matrix in Figure II, we can make some inferences about the task's complexity and identify some areas that may benefit from further investigation. Reducing the bulk of mistakes associated with correctly identifying photos as belonging to the normal class can lead to several areas of improvement. It's for being anticipated because there are certain photos where the occurrence doesn't take up much space in the image and the angle might not be perfect. It is standard procedure to process many crops from the images supplied for the purpose of making up such instances. On the other hand, this has detrimental effects on time for processing. Additionally, it is noted as the traffic incident class yielded the least accuracy generally. This is due to the fact that in certain photos, the "incident" was unclear due to the point of view, or the label was unclear in certain instances. **Fig. III** provides some intriguing categorization results examples. Notably, similar behaviours were true for all networks, supporting previous research suggesting additional pertinent data is required for CNNs to perform well in challenging scenarios.

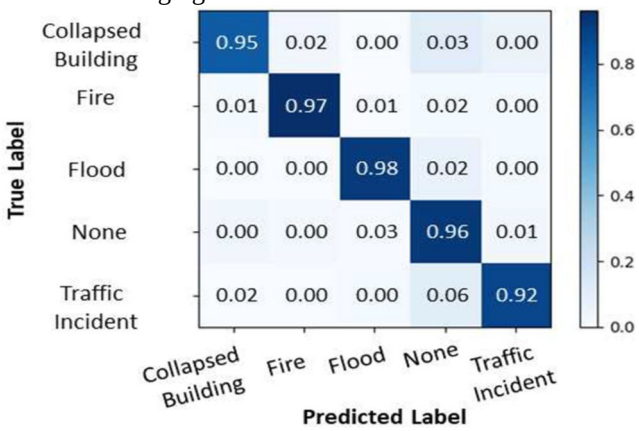

Fig. II: EmergencyNet confusion matrix.

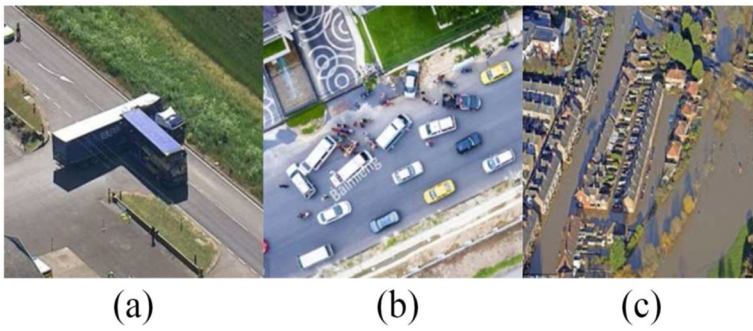

Fig. III: Events classification.

C. Evaluating Learning Qualitatively

The following subsection delves further into the elements and areas of the picture that affect the neural network's predictions along with how it learned to respond for different scenarios to determine a classification. To do

Int. J. Advanced Networking and Applications
Volume: 17, Issue: 02, Pages : 6873-6884 (2025) ISSN: 0975-0290
https://doi.org/10.35444/IJANA.2025170210

this, the method described in [33] is applied to identify the regions of the picture which elicit the largest activity. The actions of the maps of features are averaged and adjusted up to match the dimension of the map in the layer above using this method in each layer. The mean map from the layer below is then multiplied by the up-scaled averaged map of a higher level. Once the input is attained, these stages are repeated. Additionally, data on gradients coming towards the final layer of convolution instead of activating result can be employed in the gradient-weighted class activation mapping (Grad-CAM) technique [34] to visualize the input areas that are significant for class-specific predictions. Which areas of the input picture influence most to the network's output are displayed by these mask visualizations? Several successfully identified photos are shown in the illustration III, along with the regions that correlate to the CNN's greatest activations (namely, the EmergencyNet framework). It is clear from both methods that the CNN considers significant indicators while making decisions, such as the reddish-yellow glow fire in **Figure IV.** The heat-maps consistently demonstrate that the catastrophic event-related pixels dominate the significance of the categorization outcome. This suggests that the network does acquire the ability to identify significant occurrences for scenarios of emergency.

Fig. IV: the red–orange glow of fire images

D. Results of Embedded Applications

The assessment of EmergencyRes in actual use scenarios is shown in this section. Targeted are edge devices with constrained computing capacity and energy overhead, such as mobile ARM-based devices with 10–150 MFLOPs. Along with other cutting-edge networks, the planned network EmergencyRes's processing capacity and ensuing frame-rate are measured. The focus is on two distinct situations. For the purpose of processing the input image, the UAV controller is connected to a tablet that serves as the mobile ground station. The first involves an on-board processing platform. Due to their ease of deployment, both solutions can be used for remote monitoring in emergency situations.

1) UAV aboard Processing: A platform with a quad-core ARM Cortex-A57 can be utilized for on-board processing; this processor is strong and energy-efficient, and it has a tiny form factor that makes it ideal for UAVs. In comparison to other state-of-the-art models that can only attain 9 frames per second, the EmergencyRes model obtains 25 frames per second on this platform. This makes it considerably more practically usable. It's crucial to remember that MobileNetV3's lower FLOPs are not immediately converted into greater efficiency on the particular CPU-based experimentation architecture. This highlights even more how important it is to optimize while keeping the unique features of the computing architecture in consideration. All in all, a 2 – 20× accelerates is shown in comparison to previous models. But more benefits might be made by using bits compression also normalization approaches, for which EmergencyRes has proven appropriate. This would allow it to operate at even faster frame rates and enhance CPU architecture efficiency.

2) Analysis on "UAV Mobile Ground Station": A somewhat traditional approach is also used for evaluation, in which the UAV controller is linked to a tablet that serves as a mobile control station and receives the picture from the UAV camera. With this configuration, the weight and power of the UAV platform are not affected as the video stream may be processed directly on the tablet. Modern networks can only provide frame rates of about 10 FPS; the EmergencyRes network is able to provide greater frame rates of 19 FPS. Observe that interference between the UAV and the controller may be seen in real-world scenarios, which has a detrimental effect on the visual task's dependability. Therefore, the former scenario may be a better choice.

E. Discussion and Additional Improvements

The EmergencyRes network can produce accuracy on par with cutting-edge models at a quarter of the amount of storage and compute required, which is a promising outcome overall. It is still feasible to make changes, especially if the computer platform permits them. EmergencyRes's wide margins enable scientists to analyze photos with even better quality. Specifically, by fusing the effective model with partitioning techniques that divide bigger pictures into smaller sections, each model is able to handle separately. More specifically, it is feasible to provide comparable frame-rates for higher quality pictures by merging them into one batch using the use of parallel processing capabilities [52]. Assuming that successive frames in a video are associated with respect to their semantic contents might significantly enhance total accuracy in classification for streams of video. It is feasible to determine the degree of similarity among several frames by measuring the distance between the output vectors before to the classification tier. This allows one to weight the predictions made by previous frames so as to flatten out and eliminate flickering in the classification. This straightforward yet efficient method was able to lessen predicting flicker in the tests we performed films. It was discovered that the characteristic of the fusion design offered strong representation capability with fewer parameters. Therefore, utilizing the latest developments in computerized modeling search—which have been applied more and more in recent studies [23], [26]—would be advantageous as a natural progression towards enhancing the integration of features architecture.

The study has shown a few challenging scenarios that need more investigation to produce better answers. Sub-classing can be useful, and the traffic accident scenario, for instance, can encompass a wide range of circumstances. A significant obstacle that needed to be addressed was the complexity of data collection and gathering. In situations when obtaining a large amount of data online is difficult, it may be useful to investigate the latest developments in generative models to determine the joint probability distribution for every class and produce new, lifelike synthetic data. Last but not least, there is a chance to incorporate EmergencyRes with algorithms which identify individuals and vehicles along with other methods (like thermal cameras) to produce substantially improved contextual awareness. This may serve as a useful instrument to feed disaster reconstruction alongside emergency-management applications, particularly when combined with spatial applications to getting the identified events.

CONCLUION

This paper serves as a foundational research for emergency response applications utilizing "on-board UAV" analysis. The design and deployment of an effective deep learning system for autonomously identifying and categorizing catastrophic situations in real-time from a UAV's onboard have been analyzed. The suggested method offers a suitable balance between complexity, inference speed, and accuracy, and it may be applied as a foundation for use-cases with related limitations. The experimental investigation verifies the effectiveness of the suggested approach given that EmergencyRes offers accuracy that is on par with or superior to current models, is up to 20x quicker, and uses an order of magnitude less memory. Furthermore, an exclusive aerial pictures collection for emergency response applications will be provided, providing researchers with additional tools to enhance the current models. With the goal to improve on current models and methodologies and increase community knowledge regarding these applications, the dataset is going to be continuously updated along with enlarged using more photos and classes.

REFERENCES

[1]. C.Kyrkou, S.Timotheou, P.Kolios,T. Theocharides, and C. G. Panayiotou, “Optimized vision-directed deployment of UAVs for rapid traffic monitoring,” in Proc. IEEE Int. Conf. Consum. Electron., Jan. 2023, pp. 1–6.

[2]. Abed, A. H. (2024). The Applications of Deep Learning Algorithms for Enhancing Big Data Processing Accuracy. International Journal of Advanced Networking and Applications (IJANA), 16(02), 6332-6341.

[3]. D. Murugan, A. Garg, & D. Singh, “Development of an adaptive approach for precision agriculture monitoring with drone and satellite data,” IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens., vol. 10, no. 12, pp. 5322–5328, Dec. 2023.

[4]. Abed, A. H., Nasr, M., & Sayed, B. (2020). The principle Internet of Things (IoT) security techniques framework based on seven levels IoT’s reference model. In Internet of Things—Applications and Future: Proceedings of ITAF 2019 (pp. 219-237). Singapore: Springer Singapore.

[5]. Abed, A. H., Nasr, M., & Saber, W. (2019). The future of internet of things for anomalies detection using thermography. International Journal of Advanced Networking and Applications, 11(3), 4298-4304.

[6]. Abed, A. H., Rizk, F. H., Zaki, A. M., & Elshewey, A. M. (2024). The Applications of Digital Transformation Towards Achieving Sustainable Development Goals: Practical Case Studies in Different Countries of the World. Journal of Artificial Intelligence and Metaheuristics (JAIM), 7(01), 53-66.

[7]. Abdelgwad, M., Abed, A., & Bahloul, M. (2022). Authenticated Diagnosing of COVID-19 Using Deep Learning-Based CT Image Encryption Approach. Future Computing and Informatics Journal, 8(2), 31-58.

[8]. Abed, A. H. (2025). Classifying Blood Cancer from Blood Smear Images using Artificial Intelligence Algorithms. Int. J. Advanced Networking and Applications, 16(05), 6602-6614.

[9]. Abdel-Elaah, H. F., & Abed, A. H. (2024). Measuring the Activities of Human Resources systems based IoT using Machine Learning Techniques. Al-Ryada Journal for Computational Intelligence And Technology (ARJCIT), 1(1), 65-85.

[10]. Abed, A. H. (2025). Leveraging Diabetes Prediction using the Deep Learning-based Hybrid ANN-CNN Architecture. International Journal of Advanced Networking and Applications, 16(6), 6683-6690.

[11]. Fathy, H., & Abed, A. H. (2024). The Evaluation of Electronic Human Resources (eHR) Management based Internet of Things using Machine Learning Techniques. ALRYADA Journal For Computational Intelligence and Technology, 1(1), 65-85.

[12]. Attia, M., & Abed, A. H. (2024, March). A comprehensive investigation for Quantifying and Assessing the Advantages of Blockchain Adoption in Banking industry. In 2024 6th International Conference on Computing and Informatics (ICCI) (pp. 322-331). IEEE.

[13]. Y. Wang, L. Zhang, X. Tong, F. Nie& J. Mei, ‘Large: Learning latent relationships with adaptive graph embedding for aerial scene classification,” IEEE Trans. Geosci. Remote Sens., vol. 56, no. 2, pp. 621–634, Feb. 2021.

[14]. Abed, A. H. (2025). The Techniques of authentication in the Context of Cloud Computing. Int. J. Advanced Networking and Applications, 16(04), 6515-6522.

[15]. Amira, H., Ahmed, A., & Mohamed, M. (2024). Authentication in cloud computing environments. Multicriteria Algorithms with Applications, 5, 59-67.

[16]. C. Kyrkou & T. Theocharides, “Deep-learning-based aerial image classification for emergency

response applications using unmanned aerial vehicles," in Proc. IEEE Conf. Comput. Vision Pattern Recognit.Workshops, Jun. 2023, pp. 517–525.

[17]. K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in Proc. Int Conf. Learn. Representations, 2020.

[18]. Abed, A. H., & Shaaban, E. M. (2021). Modeling Deep Neural Networks for Breast Cancer Thermography Classification: A Review Study. International Journal of Advanced Networking and Applications, 13(2), 4939-4946.

[19]. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proc. IEEE Conf. Comput Vision Pattern Recogn., Las Vegas, NV, 2022, pp. 770–778.

[20]. C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z.Wojna, "Rethinking the inception architecture for computer vision," in Proc. IEEE Conf. Comput. Vision Pattern Recognit., Jun. 2021, pp. 2818–2826.

[21]. F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in Proc. IEEE Conf. Comput. Vision Pattern Recogn., Honolulu, HI, 2022, pp. 1800–1807.

[22]. A. G. Howard et al., "Mobilenets: Efficient convolutional neural networks for mobile vision applications," 2023, arXiv:1704.04861.

[23]. F. N. Iandola, M. W. Moskewicz, K. Ashraf, S. Han, W. J. Dally, and K. Keutzer, "Squeezenet: AlexNet-level accuracy with 50x fewer parameters and <1MB model size," 2023, arXiv:1602.07360.

[24]. N. Ma, X. Zhang, H.-T. Zheng, & J. Sun, "ShufflenetV2: Practical guidelines for efficient CNN architecture design," in Proc. IEEE Conf. Comput. Vision, Cham: Springer International Publishing, 2022, pp. 122–138.

[25]. M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in Proc. 36th Int.Conf. Mach. Learn., Long Beach, California, USA, vol. 97, Jun. 9–15, 2023, pp. 6105–6114.

[26]. Abed, A. H. (2024). Enhancing Big Data Processing Performance using Cutting-Edge Deep Learning Algorithms. ALRYADA Journal For Computational Intelligence and Technology, 1(1), 39-53.

[27]. Abed, A. H., & Ebrahim, M. A. (2024). Blood Cancer Detection from Blood Smear Images using Machine Learning Algorithms. SciNexuses, 1, 160-173.

[28]. Abed, A. H. (2025). Machine Learning (ML) Algorithms for Diagnosing Blood Cancer in Blood Smear Images. Journal of Computers and Digital Business, 4(2), 64-75.

[29]. L. Li, K. Ota, M. Dong, and W. Borjigin, "Eyes in the dark: Distributed scene understanding for disaster management," IEEE Trans. Parallel Distrib. Syst., vol. 28, no. 12, pp. 3458–3471, Dec. 2022.

[30]. V. Mnih and G. Hinton, "Learning to label aerial images from noisy data," in Proc. 29th Int. Conf. Int. Conf. Mach. Learn, 2020, pp. 203–210.

[31]. Hassan A., M. Nasr, L. A. Elhamid, and L. El-Fangary, "Applications of IoT in Smart Grids Using Demand Respond for Minimizing On-peak Load," *International Journal of Computer Science and Information Security*, vol. 19, no. 8, 2021.

[32]. P. Petrides, C. Kyrkou, P. Kolios, T. Theocharides, and C. Panayiotou, "Towards a holistic performance evaluation framework for drone-based object detection," in Proc. Int. Conf. Unmanned Aircr. Syst., Jun. 2023, pp. 1785–1793.

[33]. S. Kim,W. Lee,Y.-s. Park, H.W. Lee, and Y.T.Lee, "Forest fire monitoring system based on aerial image," in Proc. 3rd Int. Conf. Inf. Commun. Technol. Disaster Manag., Dec. 2023, pp. 1–6.

[34]. J. Sharma, O.-C. Granmo, M. Goodwin, and J. T. Fidje, "Deep convolutional neural networks for fire detection in images," in Eng. Appl. Neural Netw., G. Boracchi, L. Iliadis, C. Jayne, and A. Likas, Eds. Cham, Switzerland: Springer, 2022, pp. 183–193.

[35]. A. Kamilaris and F. X. Prenafeta-Bold, "Disaster monitoring using unmanned aerial vehicles and deep learning," in Proc. EnviroInfo Disaster Manag. Resilience Public Saf. Workshop, Sep. 2022.

[36]. L.-C. Chen, Y. Zhu, F. Schroff, and H. Adam, "Encoder decoder with atrous separable convolution for semantic image segmentation," in Proc. IEEE Comput. Vision, Cham: Springer International Publishing, 2023, pp. 833–851.

[37]. A. Hassan Abed, & M. Nasr (2022). Business Intelligence (BI) Significant Role in Electronic Health Records-Cancer Surgeries Prediction: Case Study. International Journal of Advanced Networking and Applications, 13(6), 5220-5228.

[38]. Abed, A. H., Nasr, M., & Abd Elhamid, L. (2021). A conceptual framework for minimizing peak load electricity using Internet of Things. International Journal of Computer Science and Mobile Computing, 10(8), 60-71.

[39]. Farhan, M. S., Abd Ellatif, M., & Abed, A. H. (2017). The success implementation CRM model for examining the critical success factors using statistical data mining techniques. International Journal of Computer Science and Information Security, 15(1), 455.

[40]. Abed, A. H. (2025). Artificial Intelligence-Driven Pharmaceutical Research: A Comprehensive Analysis of Applications and Challenges. Journal of computers and digital business, 4(1), 37-44.

[41]. Elshewey, A. M., Tawfeek, S. M., Alhussan, A. A., Radwan, M., & Abed, A. H. (2025). Optimized deep learning for potato blight detection using the waterwheel plant algorithm and sine cosine algorithm. Potato Research, 68(1), 1-25.

[42]. Farhan, M. S., Abed, A. H., & Abd Ellatif, M. (2018). A systematic review for the determination and classification of the CRM critical success factors supporting with their metrics. Future Computing and Informatics Journal, 3(2), 398-416.

[43]. Abed, A. H., Shaaban, E. M., Jena, O. P., & Elngar, A. A. (2022). A comprehensive survey on breast cancer thermography classification using deep neural network. Machine Learning and Deep Learning in Medical Data Analytics and Healthcare Applications, 169-182.

[44]. Abed, A. H. (2021). Internet of Things (IoT) Technologies for Empowering E-Education in Digital campuses of Smart Cities. International Journal of Advanced Networking and Applications, 13(2), 4925-4930.
[45]. Elshewey, A. M., Abed, A. H., Khafaga, D. S., Alhussan, A. A., Eid, M. M., & El-Kenawy, E. S. M. (2025). Enhancing heart disease classification based on greylag goose optimization algorithm and long short-term memory. Scientific Reports, 15(1), 1277.
[46]. Elshewey, A. M., Selem, E., & Abed, A. H. (2025). Improved CKD classification based on explainable artificial intelligence with extra trees and BBFS. Scientific Reports, 15(1), 17861.
[47]. Abed, A. H. (2022). Deep learning techniques for improving breast cancer detection and diagnosis. International Journal of Advanced Networking and Applications, 13(6), 5197-5214.
[48]. Abdel-Elaah, H. F., & Abed, A. H. (2025). The Contribution of Artificial Intelligence to Addressing the Global Goals for Sustainable Development. Journal of computers and digital business, 4(1), 45-55.
[49]. N. S., Shehata, & A. Hassan, (2020). Big Data with column oriented NOSQL database to overcome the drawbacks of relational databases. International Journal of Advanced Networking and Applications, 11(5), 4423-4428.
[50]. A. H. Abed,& M. Nasr, (2019). Diabetes disease detection through data mining techniques. International Journal of Advanced Networking and Applications, 11(1), 4142-4149.
[51]. Abed, A. H. (2020). Recovery and concurrency challenging in big data and NoSQL database systems. International Journal of Advanced Networking and Applications, 11(4), 4321-4329.
[52]. S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in Proc. 32nd Int. Conf. Int. Conf. Mach. Learn., vol. 37, ser. ICML15.JMLR.org, 2021, pp. 448–456.