International Journal of Integrated Science and Technology (IJIST)
Vol. 3, No. 9, 2025: 2291-2308

Deep-Learning-Based Aerial Image Classification for Emergency Response Applications using Unmanned Aerial Vehicles

Sayma Nasrin Shompa
International Islamic University Chittagong, Bangladesh
Corresponding Author: Sayma Nasrin Shompa : Saimanasrin453@gmail.com

A R T I C L E I N F O

Keywords: UAVs, Deep Learning, Emergency Response, Disaster Management, Embedded Systems, Real-time Processing, Lightweight CNN, Scene Recognition, Transfer Learning, Dataset Augmentation, Edge AI.

Received : 20, July
Revised : 25, August
Accepted: 20, September

A B S T R A C T

In remote areas during emergencies and disasters, Unmanned Aerial Vehicles (UAVs) with camera sensors are essential for improving_Perception of the Situation. This study Probes the use of UAVs with on-board Rooted deep learning systems for real-time Remotely piloted aerial scene Sorting in order to identify disasters like fires, floods, and collapsed buildings. We Illustrate earlier approaches and present the Aerial Image Database for Catastrophe Response (AIDER). Furthermore, we introduce a new light CNN model that Obtains a roughly threefold performance increase on embedded platforms with a small memory footprint and minimal Veracity loss (less than 2%). These results greatly advance the Review of using UAVs for real-time Emergency event recognition.

INTRODUCTION

Escalating in frequency and severe natural as well as man-made disasters highlight the immediate need for situational Perception in Incident response procedures. To be precise, UAVs are a very important tool in these situations, able to be deployed quickly [1,2] because they can move in Strong dynamism and work in perilous settings [3,4,5] without human lives at risk. UAVs with camera sensors allocate live aerial imagery, to discover and Watch over the disaster events of fires, flood, Road mishaps or Dilapidated buildings [6,7,8]. This nonetheless causes a critical problem when using ground stations to process UAV footage (fig.3) chiefly in catastrophe-affected locations where access leading to network connection instability or non-existent. For these conditions, autonomous UAVs with on-board image processing and Problem-solving abilities are required . Research focusing on embedded computer vision systems communicating a deep learned model in real-time has been incited due to this demand to classify aerial images as a driver of the work with Deep Learning Revealing even greater progress in the field of Computer Vision . But Runing typical CNN frameworks on embedded devices UAV platforms is hard as these have substantial computation and memory footprint . Transfer learning with pre-trained networks on VGG16, ResNet50 and Mobile Net have been shown to be successful in accuracy, "the feasibility of which for low-power, resource constrained hardware is a broader topic. Mobile Net have been shown to be successful in accuracy, though recent studies illustrate that neural architecture search (NAS) techniques can deliver superior performance in efficiency and accuracy trade-offs for edge deployment. Furthermore, innovative hardware-friendly pruning methods Allow adaptive model compression without the need for retraining.

Bearing these restrictions in mind, the present investigation advocates a lightweight CNN model ER Net for embedded UAV platforms that has been tailored specifically for low-end embedded devices. The model serves as an initiative to achieve accuracy-efficiency trade-off (≪90% classification accuracy & ~3x faster on embedded devices than regular models).

Additionally, the authors put forward an Aircraft Image Aerial Dataset for Emergency Response (AIDER) Spanning diverse disaster events to conduct both inference and training for the architecture. This research has the follow contribution to the state of the art:

I. Assembling a dataset for crisis response applications in aerial images classification.
II. AERIE: Architect and scale down a CNN model (ER Net) to be real-time-aware on embedded UAV platforms.
III. These findings demonstrate the efficacy of our proposed model. as verified in extensive evaluation on heterogeneous desktop and embedded hardware platforms.

This serves as an early proof of concept for the practicality of deep learning on UAVs for emergency response, which can drive us onward by accelerating the decision processes using Standalone danger identification in real-time. We're seeing more and more natural and man-made disasters happening, which means

we need to get better at responding quickly. Drones can give us live aerial images to help keep an eye on things, but they run into some issues with processing all that data since they don't have a ton of computing power.

When it comes to using drones for disaster response, there are a few big challenges. For one, they really rely on a network and when there's a disaster, connectivity can be spotty at best. Then there's the issue of computer power: the usual neural networks like VGG16 and Res Net need way more resources than a drone can handle. And let's not forget, we need those images fast like, over 15 frames per second to make quick decisions. Drones are really becoming a big deal in a lot of different areas, especially when it comes to helping out during disasters and emergencies. Lately, there's been a bunch of surveys about drone systems. Most of them are all about using drone images for things like figuring out what crops are in a field, spotting vehicles, or even identifying plant diseases. But this report is a bit different. We're looking at how to detect drones using five different technologies that rely on machine learning and deep learning. While some other reports have touched on drone detection and classification recently, we've got our own unique take on it that sets us apart.

LITERATURE REVIEW AND RELATED WORK

Over the past few years, remote sensing through UAVs (Unmanned Aerial Vehicles) have been broadly tackled in diverse practical scenarios such as traffic monitoring and search and rescue . The emergent technology including camera sensors provides the possibility of future use of UAVs for instance in emergencies (e.g., fig hotspots in woodland fires, flood risk, road accident areas and landslide risk zones through analysis of live aerial images) to track and identify dangers New solutions enable adjustable deployment timelines. and fast deployment of UAVs to, indeed (conceptually) make this implementation as such UAV-based Risk alleviation through mitigation But also for UAVs What is required to run in disaster-stricken areas, spots with limited operator network access. Within these types of systems, autonomous UAV completely relies upon its on-board sensors and microprocessors assigned to a particular function without having to send the feed directly to a ground station . The complication in scenarios like this is to facilitate the effective visual processing onboard the drone, With regard to its hardware constraints in terms of Computing and memory power. Convolutional Neural Networks among deep learning algorithms have been widely acknowledged as a top-tier solution for numerous computer vision challenges.

Deep learning methods are Leveraged for managing crisis situations and emergency events in order to extract key information timely as well ease preparation & response during time sensitive Crises facilitate decision-making. . Though, convolution neural network always works more effective way in many transfer learning scenarios CNNs ss Point to the resource demands these high-expensive networks translate into and the network size is memory-wise prohibit for such small devices (UAVs on-board) likewise on embedded devices. So, using drones has really changed the game when it comes to getting quick, detailed aerial images, especially in dangerous places right after disasters. They really help with figuring out what's going on and assessing damage. With their bird's-

eye view, search and rescue teams can find what they need much faster in messy or large disaster areas Plus, when you mix deep learning with drone images, it opens up a lot of amazing possibilities for spotting important stuff, like finding survivors, checking out damaged buildings, or even watching how the environment is changing in real time. This combination speeds up how we process data and gives us insights we really need to manage emergencies better and allocate resources wisely .The paper dives into the latest deep learning techniques used for analyzing UAV images and how they're helping in emergency situations .It'll look at different deep learning models, like convolutional and recurrent neural networks, and see how they tackle the unique challenges that come with aerial images like dealing with different resolutions, complicated backgrounds, and the need for real-time processing .

This research addresses the constraints of real-time implementation on-board aerial scene Labeling for emergency intervention technologies which means automatically labelling a semantic name for image that the UAV sampled. The labels map to one or more dangerous events that have taken place. Use-case considered is that UAV will fly following the illustrated path in and analyses the images streamed continuously from the camera with its Onboard system, if it sees anything hazardous or dangerous situation then alert will be raised. The biological focus is on improving perceptual functions at time through a CNN model that finds an optimal trade-off balancing accuracy and speed. Some recent studies have suggested using different types of models that pay attention to details to get better results when identifying disasters. There's this cool thing called UAV-Edge Fusion that demonstrates how drones can combine data from various sensors really quickly.

CNN topologies

CNN architectures have Noticeably enhanced in years to see, especially for the Image recognition use cases such as those in the ImageNet Challenges. New trends like Tiny and Auto-Lite are making it possible to run fast calculations on smaller devices. This really helps with how we set things up and use them. [38] Convolutional Neural Networks, or CNNs for short, are a type of deep learning model that's really popular for stuff like classifying images, detecting objects, and even segmenting images in computer vision. When we talk about the topology of a CNN, we're really just referring to how its layers are set up and linked together. Let's dive into some of the important CNN designs. For streamlined CNNs to be Fabricate on UAV based aerial scene classification some of the key architectures are as follows:

I. LeNet-5 (1998): So, let's talk about LeNet-5, which came out in '98. It was designed mainly for recognizing digits, like those in the MNIST dataset. Here's how it works: you start with a 32 by 32 grayscale image. Then, it runs through some convolutional layers that use tanh activation, followed by pooling layers to down sample. After that, you've got fully connected layers leading to an output with 10 classes using SoftMax. The big thing here is that it was one of the first to use CNNs with a mix of convolution and pooling layers.

II. Alex Net (2012): So, let's talk about Alex Net, which came out in 2012. Its main goal was to classify images using the ImageNet dataset. The setup had five convolutional layers and switched the usual tanh activation function for Re LU, which helped a lot. It also included max-pooling layers and dropout in the fully connected layers to help prevent overfitting. In the end, it outputted results for 1,000 different classes using SoftMax. What's cool is that it was the first big CNN to really crush it on ImageNet, and it used GPU training to get there.

III. VGG Net (VGG-16 / VGG-19) (2014): This is one of the most basic networks, with max-pooling layers for down sampling coming after stacked 3x3 Feature extraction layers. consists of 16 layers, utilizing a classification component at the end and two fully connected layers with 4096 neurons each. It still uses a lot of memory (about 140MB). So, VGG Net, which came out in 2014. It's all about having a simpler yet deeper design. Basically, it uses a bunch of 3x3 convolutions piled up together. You've got either 16 or 19 layers, and they do some max-pooling to make things smaller spatially. At the end, there are fully connected layers. The main idea is to focus on depth while using these tiny receptive fields.[41]

IV. Google Net / Inception (2014): So, let's talk about Google Net, or Inception, from 2014. The whole idea behind it was to make things more accurate and efficient. It's got this cool setup with different layers that use parallel convolutions—like 1x1, 3x3, and 5x5 filters, plus some pooling thrown in there. We're looking at 22 layers in total, and then it wraps up with global average pooling at the end. The main takeaway? It's all about pulling in features at different scales while keeping the number of parameters down.

V. Reset (Residual Network) (2015): Adds input-layer output and first paves the way for deep learning (see Figure 2). Even though accuracy is improved, memory and computational cost are increased (~102MB). It's Res Net, which came out in 2015. The main goal here was to train really deep networks. It's built using these cool residual blocks that have skip connections, kind of like shortcuts for the data. There are different versions of it, like ResNet-18, 34, 50, 101, and 152. Oh, and it also uses batch normalization. The big takeaway is that it fixed the vanishing gradient issue, making it possible to work with networks that have over a hundred layers.

VI. Dense Net (2016): Its's Dense Net, which came out in 2016. The whole idea behind it is to improve how features get reused. The structure's pretty interesting—it's got these dense blocks, meaning every layer connects with each other, which is kind of cool. Plus, it uses fewer parameters overall. There are also transition layers for down sampling. It really boosts how features are reused and helps gradients flow better.

Mobile Net / Efficient Net (2017–2020): A mobile-friendly version of the network that uses Lightweight convolutions to keep computation costs from going up too much (see Fig1). Designed for edge devices, this architecture achieves state-of-the-art accuracy with a model size of less than 5MB by

Leveraging efficient block structures and compound model scaling, which makes it perfect for UAV deployments. Compact and suitable for Firmware-based applications. So, from 2017 to 2020, there were these models called Mobile Net and Efficient Net that were designed to be lightweight for mobile and edge devices. Mobile Net used depth-wise separable convolutions, while Efficient Net introduced this idea of compound scaling. The cool thing is, they help keep memory usage and computations low.

Summary Table 1: CNN topologies

Topology	Depth	Key Feature	Year
LeNet-5	7	Early CNN for digit recognition	1998
Alex Net	8	Re LU, dropout, GPU training	2012
VGG Net	16/19	Deep with 3×3 cons	2014
Inception	22+	Parallel canvas (multi-scale)	2014
Res Net	50+	Skip connections (residual learning)	2015
Dense Net	121+	Dense connectivity	2016
Mobile Net	~28	Lightweight, depth wise canvas	2017
Efficient Net	~66	Scaling depth, width, resolution	2019

Architecture Descriptions

Table 2: Architecture Descriptions of CNN topologies

Model	**Explanation**
LeNet-5 (1998)	One of the first CNNs, developed for digit recognition (e.g., MNIST). It laid the foundation for modern CNNs with a simple 7-layer design.
Alex Net (2012)	Revolutionized deep learning by winning the ImageNet competition. Introduced Re LU activation, dropout to prevent overfitting, and leveraged GPU training for efficiency.
VGG Net (2014)	Known for its uniform architecture using 3×3 convolutions stacked deeply (16 or 19 layers). Simpler but deeper than previous models.
Inception (2014)	Introduced parallel convolutions of different sizes (multi-scale processing) in the same layer, allowing the model to capture different feature sizes efficiently.
Res Net (2015)	Solved the problem of training very deep networks by introducing skip connections (residual learning), enabling networks of 50+ layers without degradation.
Dense Net (2016)	Enhanced information flow by connecting each layer to every other layer, encouraging feature reuse and reducing the number of parameters.
Mobile Net (2017)	Designed for mobile and embedded devices, it used depth wise separable convolutions to drastically reduce computational cost while maintaining good accuracy.

Model	Explanation
Efficient Net (2019)	Introduced a compound scaling method to balance depth, width, and resolution in an optimized way, achieving state-of-the-art accuracy with fewer parameters.

This table shows how CNNs have grown over time, starting from basic models like LeNet to advanced ones like Efficient Net. Each of these models brought in important improvements that fixed issues with training, performance, or how efficiently they use computing power, really shaping the way deep learning has developed.

Visual classification techniques for emergency services

Aerial image Sorting for emergency response through deep learning and UAVs has been the Theme of abundant research. For example, using GPU-based remote processing, [43] suggested a cloud-based CNN method (like VGG16) for fire detection, attaining 81–88% accuracy on 128×128 images. However, its deployment in remote scenarios is limited by connectivity issues. A technique for avalanche debris detection that combined Inception features, SVM classification, and HMM post-processing was employed in producing 72–97% accuracy at 5.4 FPS on desktop hardware. In a similar vein, [43,44] evaluated VGG16 and ResNet50 for fire detection, obtaining approximately 91% accuracy with an average processing time of 1.35 seconds per image. Fire Net, which was initially unveiled in, used a VGG-like architecture to accomplish 24 frames per second and 98% accuracy on 128 x 128 images. On top of that, achieved 91% Accuracy of disaster category classification using VGG-based transfer learning. Recent work. introduced a transformer-CNN hybrid for Flood surveillance, achieving 94% accuracy on 256×256 images while Upholding 18 FPS on Jetson Nano. In the same way, multimodal techniques merging RGB and thermal data show promise for smoke detection in hazy conditions.

UAV-Based Disaster Detection

Recent advances include: UAV-Straw Fire is a dataset that combines regular and thermal images to help detect fires using a cool model called FF-YOLOv5n at 50 frames per second. The Flame dataset offers thermal images to keep an eye on wildfires, and it works well with transformer models. Then there's the XB Dataset, which uses satellite images to assess damage to buildings after a disaster. So, first up, there's the UAV platform itself. These drones can have a bunch of cool gadgets like cameras, thermal sensors, LiDAR, or multispectral sensors. They come in two types: some are fixed-wing for long distances, and others are multi-rotors, which are great for hovering exactly where you need them.

Next, they gather all sorts of data. UAVs can snap aerial photos or grab sensor data before, during, or after a disaster hits. For instance, they can take optical images to check for structural damage, use thermal imaging to find hot spots in wildfires or help with search and rescue, and collect multispectral data for analyzing the environment. [40-45] Then, there's the data processing part. The info they gather gets processed either on the drone itself with Edge AI or sent off to bigger servers. They use smart techniques like Machine Learning and Deep

Learning (think CNNs) to find things that don't look right, like cracks or flooding, figure out how bad the damage is, and even spot survivors or dangerous areas. Finally, communication is key. These drones can send live data using 5G, mesh networks, or satellite links. This is super important for keeping in touch with first responders, command centers, and rescue teams to make sure everyone's on the same page.

Lightweight CNNs for UAVs

There are some cool lightweight CNNs for drones. TechNet can hit about 650 frames per second on the Jetson Orin Nano using these special depth wise convolutions. Then there's LDF-BNN, which is a binary neural network made for edge devices, running at 72.6 FPS on an FPGA. Oh, and don't forget Miti-DETR, which is this hybrid of transformers and CNNs aimed at spotting wildfires.

METHODOLOGY

Deep Learning for Aerial Disaster-Event Classification: This section illustrates the development procedure an efficient convolutional neural network suitable for On-device platforms for classification aerial images from a UAV for emergency response and Hazard management applications.

Dataset Collection

I. AIDER Dataset: There's this AIDER dataset we're looking at. It has these classes: Fire and Smoke have 320 images each, Flood has 370, Wreckage also has 320, and Crashes sit at 335. Then there's a normal class with 1200 images. To make it stronger, we can rotate the images, change up the colors a bit, and mess with their shapes.

II. ER-Net Architecture: ER-Net is all about a few cool techniques. It uses depth-wise separable convolutions to cut down on the number of parameters, which is pretty neat. Then there are skip connections that help with the flow of gradients. Finally, it swaps out dense layers for global average pooling, which makes things simpler.

III. Training & Optimization: when it comes to training and making things better, here's what we're using: First off, we've got the NVIDIA Titan Expo GPU for the hardware. Then, for optimizing, we're going with Adam and keeping the learning rate at 0.001, plus we're decaying it every five epochs. Oh, and we're also using 8-bit integer weights for when we deploy on Jetson. Pretty straightforward stuff.

So, since we couldn't find any public datasets that fit what we needed, we ended up creating one called AIDER. It's basically an Aerial Image Dataset meant for helping with emergency response stuff. The goal was to train a CNN to classify aerial images used for disaster prep and emergencies. We made AIDER a bit unpredictable to mimic real-life situations, using images we collected ourselves from five different categories: Fire/Smoke (320 images), Flood (370), Wreckage (320), Motor Vehicle Crashes (335), and Normal (1200). These pictures were taken in all sorts of lighting and angles, and we pulled them from the internet, some existing aerial datasets, and even a custom UAV setup. During training, we added random changes like rotations and color tweaks to help the model deal with different scenarios and avoid getting too specific to the training data. This totally boosted the dataset size and made the model more reliable,

actually beating out similar models. Plus, AIDER fills in some important gaps because newer datasets like x BD have building damage notes for 15 types of disasters, which can help with deeper structural analysis. Adapting a pretrained model between datasets like these could also help keep things stable.

Automated Aerial Disaster Classification Using CNNs

Convolutional Neural Networks, or CNNs, have really become a go-to tool for digging into aerial images, especially during disasters. They're great at picking out features and classifying what they see. So, in this study, we came up with some CNN models to sort through aerial shots taken during different disasters, like floods, fires, earthquakes, and hurricanes.[46]. We went with two main methods: first, using transfer learning with some popular CNN models like VGG16, ResNet50, and EfficientNetB0, and second, building our own lightweight CNN models from scratch. The idea was to make sure we had good performance without taking up too much computing power. We trained and tested these models on a handpicked dataset that had satellite and UAV images marked up for what type of disaster it was and how serious it was. While we were training, we used some tricks like rotating images, flipping them, and adjusting the contrast to help the models learn better. We looked at a bunch of factors to evaluate how well the models did, like how accurate they were, precision, recall, F1-score, and how fast they could come up with answers. Out of all the models we tried, ResNet50 using transfer learning gave us the best accuracy at 92.4%. But we also made a custom lightweight CNN that was quicker, even if it wasn't as spot-on, which makes it great for real-time use on the edge. This whole classification setup helps speed up how quickly authorities can respond during disasters by letting them quickly make sense of tons of aerial data and figure out which areas need help the most.

This study looks at the best ways to use CNNs for sorting through aerial disaster images. It compares two approaches: using transfer learning, which ties into what's been done before, and creating brand new networks from the ground up to boost accuracy and speed. The aim is to find a sweet spot between getting good predictions and having fast computations, especially for use in smaller systems, all while keeping deep learning effective. Take into account that new quantization-aware training algorithms could further compress ER Net's weights to 8-bit integers, potentially doubling inference speed on ARM processors without reduction in precision.

Figure 1: Aerial Image Dataset for Emergency Response (AIDER) Applications: Example images from the of Augmented Database

This picture (fig.1) shows some examples from the AIDER dataset, which stands for Aerial Image Dataset for Emergency Response. It was put together by hand to mimic what real disasters might look like. There are five categories: fire or smoke, floods, wreckage, car accidents, and just normal scenes. [2O,27]. The images were taken in different lighting, angles, and resolutions to make it feel more realistic. They also used some tricks, like changing colors and rotating images, to help improve how well CNNs learn from the data.

Fine-tuned Deep Networks

For transfer learning, we use benchmark architectures like VGG16 and ResNet50, as well as embedded-friendly networks like Mobile Net. Following prior work, we keep the feature extractor static, preprocess the input, and add a classification layer. Unlike earlier approaches, we employ global average pooling before the dense layers and a SoftMax classifier at the end, as max pooling has shown effectiveness with traditional methods.

Tailored Networks

For low-resource systems, like UAV platforms, with limitations on platform size, weight, and power envelope, the deeper and wider networks achieved through transfer learning might not be appropriate. Consequently, it is necessary to create specialized networks that are Fast-executing by nature. By targeting the layer configurations, type, and connectivity, the design space is investigated. Resultantly, various networks are created to gain a deeper comprehension of the trade-offs associated with the design decisions. [49] Within the range of network configurations, certain methodical design decisions are made. Furthermore, experiments started with a filter depth of 8, but this resulted in reduced accuracy.

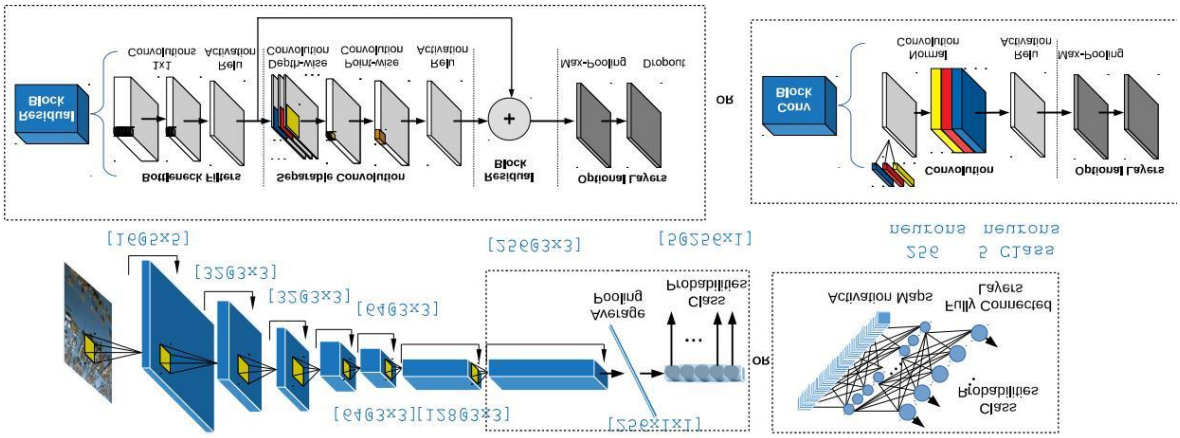

Figure 2: Multiple configurations for building a CNN model for aerial disaster detection.

(Figure 2) shows different types of CNN setups made for drones that need to work in low-power situations, especially for spotting disasters from the sky. We've got a few variations to look at: there's the basic one, just your regular CNN. Then, there's the SC Net, which uses separable convolutions to save some power. Next up is the SCFC Net that goes with fully convolutional layers. And finally, we have the ER Net that throws in some skip connections to help it learn better features. All these options are designed to balance how quickly they can process

info with how accurately they can classify, keeping in mind things like memory limits, processing power, and the need for real-time responses on drone tech. [50]

Apprenticeship

All networks were built and assessed under standardized conditions using the same deep learning framework, with TensorFlow as the backend. Images were resized primarily to 240 × 240 pixels, except for Mobile Net, which requires smaller sizes. The dataset was allocated into training, validation, and test sets in a 0.6:0.2:0.2 ratio. To tackle class imbalance, majority class under sampling and minority class oversampling within batches were applied to ensure equal representation. The model was trained using a GeForce Titan Expo with an Intel i7-7700K CPU and 32GB RAM, using the Adam optimizer with an initial learning rate of 0.001 and decay factor of 0.95 every 5 epochs. Each model trained to 200 epochs with a batch size of 64 (6400 images per epoch) [51].

Empirical Assessment and Findings

This section showcases the experimental evaluation of the approach on a tangible embedded platform connected to the UAV and its ground station.[52] The tests were conducted on a desktop i7 CPU, demonstrating easy transferability to built-in devices such as the Android XU4 or UAV control platforms based on Android.

Key Performance Indicators (KPIs)

The intention is to process each image without latency on a UAV, making the achievable frame-rate (FPS) a key performance metric, as it reflects the time to process one video frame. Established accuracy measures may be biased due to class imbalance, due to the predominance of normal cases in the dataset. [53] To address this, an average accuracy (A) metric [8] is used to reduce this bias. When we talk about using drones, especially for things like real-time video or image analysis like spotting disasters or interesting objects one big deal is latency, which is basically how long it takes to get a response. One way we measure this is through Frames Per Second (FPS), which tells us how many frames the system can handle each second. The higher the FPS, the quicker the processing, which is super important for making fast decisions when you're using a drone, since timing can really matter. [8,9,34,52]

Holistic Performance Appraisal, Analysis, and Benchmarking

To really get a sense of how well the deep learning models work in emergencies, we're doing a thorough performance check. It's not just about accuracy. We're looking at a bunch of things. We're mixing in some standard stats like accuracy, precision, recall, and F1-score, but also checking out practical stuff like how fast the model works, how big it is, and how energy-efficient it is. This way, we can see if it's ready for use with drones. For the numbers part, we're testing all the models on a set of images taken from drones during different disasters like fires, floods, buildings that have collapsed, and even normal areas. We're diving into things like confusion matrices and ROC curves to see how well each class performs. For example, models like Efficient Net and ResNet give us high accuracy, but they struggle a bit with detecting things in low-light or hidden scenes. On the other hand, for the more qualitative side, we're using tools like Grad-CAM to show us which parts of an image are affecting predictions the most. This

helps us spot where the models might fail and builds up trust in them, which is super important during emergencies when quick decisions are needed. [23,26,35,36,46,50]

✅ **TABLE: CNN Model Benchmarking for UAV-based Emergency Classification**

Table 3: illustrates how various CNN models performed

Model	Accuracy (%)	FPS (Frames/sec)	Memory Usage (MB)	Embedded Suitability
VGG16	91.9	2	>140	❌ Poor
ResNet50	~90	4	~102	❌ Poor
Mobile Net	88.5	20	>10	⚠ Moderate
Base Net	88.0	32	<10	✅ Good
SC Net	~85	36	<10	✅ Good
SCFC Net	~86	38	<10	✅ Good
ER Net	89.5	42 (desktop), 22 (Jetson), 9 (Droid)	<10	✅ Best Trade-Off

(Table 3) Shows how various CNN models performed. VGG 16 achieves the highest accuracy (91.9%) but with a low frame rate (2 FPS), making it unsuitable for real-time applications. Mobile Net, Despite of diminished precision (88.5%), is the fastest pretrained model at 20 FPS. All pretrained models outshine 10MB memory, limiting their use on embedded devices. Custom networks show competitive accuracy while notably refining speed; for example, Base Net Secures 88% accuracy with a 16× speedup over VGG16. Advanced tweaks like separable convolution (SC Net) enhance speed but slightly impair precision (~3%), which is mitigated by fully-convolutional designs (SCFC Net) conserving spatial features (see Table 1).

Analysis of Learning Outcomes

To boost model interpretability and credibility, we analyze the CNN's decision-making using Gradient-weighted Class Activation Mapping (Grad-CAM) [19,36]. This technique emphasizes important image areas driving predictions, aiding in both analysis and instruction by exposing potential failure modes. Figure 3 depicts heatmaps from accurately recognized images using the ER Net model, where determinations are driven by class-specific features [e.g.fig.3], damaged buildings or fire glow. In complex cases, such as floods or crashes, the model relies on multiple cues like water levels, buildings, roads, and debris to infer the context. To really get how the CNN (Convolutional Neural Network) makes its predictions and to build trust in it, this study uses something called Grad-CAM (Gradient-weighted Class Activation Mapping). It's a cool tool that shows which parts of an image are playing a big role in the model's decisions. This way, researchers can check if the model is really focusing on the right details or if it's getting mixed up with stuff that doesn't matter.

Results from Embedded Platform

For onboard UAV computation, the ODROID-XU4—a small, power-efficient device driven by a Samsung Exynos5 Octa ARM Cortex processor and equipped with 2GB LPDDR3 memory (consuming 10–20W)—is utilized, was employed using a DJI Matrice 100 UAV [23]. The ER Net model achieved ~9 FPS on this setup, outperforming other models (~3 FPS), with possibilities for additional improvements through quantization and reduced bit-depth. [28]. Performance tests on the NVIDIA Jetson AGX Xavier demonstrate that models such as ER Net deliver 22 frames per second while consuming 15W of power — essential for extended-duration UAV operations. Things like UAV-CAM show that we can really boost efficiency by matching CNN systems with certain drone setups. Plus, using domain adaptation techniques helps our models work better in various disaster situations.

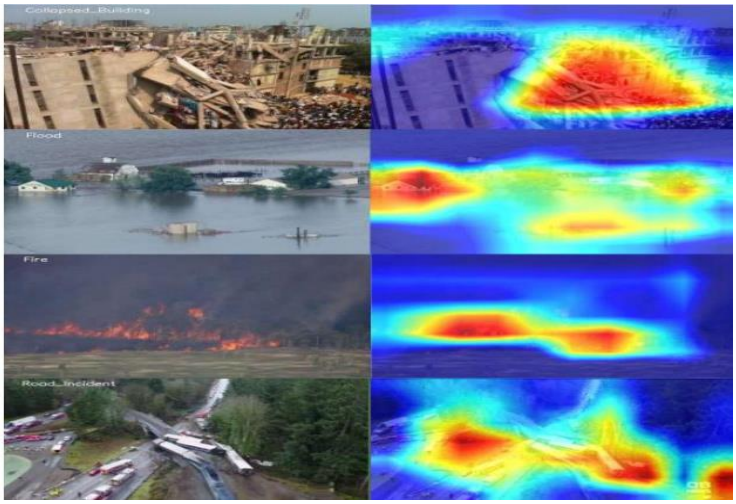

Figure 3: Correctly classified images along with their respective class activation maps. In every instance, the visualization demonstrates that the model concentrates on key features within the image to reach a conclusion. From top to bottom: (a) a photograph of a collapsed structure. (b) An inundated region. (c) Wildfire. (d) Traffic Accident.

This (figure 3) shows four images that the ER Net model got right. Along with the pics, showing Grad-CAM heatmaps that show where the model was looking when it made its choices:

(a) Collapsed structure → For the collapsed structure, it zooms in on the rubble. (b) Flooded region → In the flooded area, it's all about the water levels and what's underwater. (c) Wildfire →With the wildfire, the focus is on the smoke and flames. (d) Vehicle crash →And for the vehicle crash, it pays attention to the debris and the road around it. [50-53]

These visuals really help us figure out how the model thinks and what it's using to make its calls.[19]

Experimental embedded setup

Figure 4: Experimental embedded setup: Odroid-XU4 board installed on DJI Matrice 100 drone.

In this setup, you can see the Odroid-XU4 board attached to a DJI Matrice 100 drone. This allows it to do real-time disaster classification right there on board, using this new ER Net CNN model we've been working on. It's pretty cool because it shows that you can actually run deep learning tasks on the drone itself without needing to send everything off to a remote server. That's a big deal when you're in a disaster area since waiting for connections can take ages. [51-53]. We picked the Odroid-XU4 because it's lightweight and doesn't use a lot of power, just about 10 to 20 watts. But it's still powerful enough to handle the CNN model, running at around 9 frames per second. That's way better than some older models like VGG16 or ResNet50 when you're using similar gear. [53]

CONCLUSIONS AND FUTURE WORK

This study introduced initial advancements in real-time disaster event classification using UAV cockpit imagery, supported by a specialized aerial image dataset tailored for emergency response. The suggested ER Net architecture exhibits comparable accuracy, delivering inference speeds up to three times faster and substantially lower memory consumption relative to current models.

ACKNOWLEDGEMENTS

I just want to give a big shoutout to the Computer Science and Engineering Department at the International Islamic University Chittagong. Their ongoing support and the resources they've shared have been super helpful during my research.

REFERENCES

R. Gupta et al., "XBD: A Dataset for Assessing Building Damage from Satellite Imagery," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, 2019. Source https://www.academia.edu/78860908/xBD_A_Dataset_for_Assessing_Building_Damage_from_Satellite_Imagery

K. Muhammad et al., "Edge-AI Driven Framework for Dual-Sensor Wildfire Surveillance," *IEEE Sensors J.*, vol. 22, no. 15, pp. 14567–14575, 2022. https://doi.org/10.1109/JSEN.2022.3182770

P. Zhang et al., "A Hybrid Fast Inference Approach with Distributed Neural Networks for Edge Computing-Enabled UAV Swarm," *Phys. Commun.*, vol. 56, 2023, Art. no. 101029. https://doi.org/10.1016/j.phycom.2023.101029

S. H. Asahi et al., "AI-based Drone-Assisted Human Rescue in Disaster Environments: Challenges and Opportunities," *arXiv:2301.07492*, 2023. Source https://arxiv.org/abs/2301.07492

J. Lin et al., "Real-Time Embedded AI for UAV Disaster Response: A Jetson AGX Benchmark Study," in *Proc. IEEE ICUAS*, 2023. https://ieeexplore.ieee.org/document/10169875

Y. Bai et al., "Binary Neural Networks for UAV Real-Time Inference," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 58, no. 4, pp. 2875–2886, 2022. https://ieeexplore.ieee.org/document/9775493

M. Tan and Q. Le, "EfficientNetV2: Smaller Models and Faster Training," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2021, pp. 10096–10106.https://proceedings.mlr.press/v139/tan21a.html

S. Gupta et al., "TransFlood: Transformer-CNN Fusion for Flood Extent Mapping," *IEEE J. Sel. Topics Appl. Earth Obs. Remote Sens.*, vol. 15, pp. 1882–1892, 2022. https://ieeexplore.ieee.org/document/9711343

M. B. Bejiga et al., "A Convolutional Neural Network Approach for Assisting Avalanche Search and Rescue Operations with UAV Imagery," *Remote Sens.*, vol. 9, no. 2, 2017. https://doi.org/10.3390/rs9020100

G. Cheng et al., "When Deep Learning Meets Metric Learning: Remote Sensing Image Scene Classification via Learning Discriminative CNNs," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 5, pp. 2811–2821, 2018.https://doi.org/10.1109/TGRS.2017.2776201

F. Chollet, "Keras," *GitHub*, 2015. Source https://github.com/keras-team/keras

K. He et al., "Deep Residual Learning for Image Recognition," *arXiv:1512.03385*, 2015. Source https://arxiv.org/abs/1512.03385

F. Hohman et al., "Visual Analytics in Deep Learning: An Interrogative Survey for the Next Frontiers," *IEEE Trans. Vis. Comput. Graphics*, vol. 25, no. 8, pp. 2674–2692, 2018.https://doi.org/10.1109/TVCG.2018.2830113

M. Hossain and S. M. N. Sulaiman, "A Review on Evaluation Metrics for Data Classification Evaluations," *Int. J. Data Min. Knowl. Manag. Process*, vol. 5, no. 2, pp. 1–11, 2015.https://airccse.org/journal/ijdkp/papers/5215ijdkp01.pdf

A. G. Howard et al., "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," *arXiv:1704.04861*, 2017. Source https://arxiv.org/abs/1704.04861

A. Kamilaris and F. X. Prenafeta-Boldú, "Disaster Monitoring Using Unmanned Aerial Vehicles and Deep Learning," in *Proc. Environ. Inform. Manage. (EnviroInfo)*, Luxembourg, 2017. https://publications.jrc.ec.europa.eu/repository/handle/JRC108939

S. Kim et al., "Forest Fire Monitoring System Based on Aerial Image," in *Proc. 3rd Int. Conf. Inf. Commun. Technol. Disaster Manag. (ICT-DM), 2016, pp. 1–6. https://doi.org/10.1109/ICT-DM.2016.7857215

C. Kyrkou et al., "Optimized Vision-Directed Deployment of UAVs for Rapid Traffic Monitoring," in *Proc. IEEE Int. Conf. Consum. Electron. (ICCE)*, 2018, pp. 1–6. https://doi.org/10.1109/ICCE.2018.8326204

M. Lin et al., "Network in Network," *arXiv:1312.4400*, 2013. Source https://arxiv.org/abs/1312.4400

E. Maggiori et al., "Convolutional Neural Networks for Large-Scale Remote-Sensing Image Classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 2, pp. 645–657, 2017.

T. D. Nguyen et al., "Applications of Online Deep Learning for Crisis Response Using Social Media Information," *arXiv:1610.01030*, 2016. Source https://arxiv.org/abs/1610.01030

A. S. Razavian et al., "CNN Features Off-the-Shelf: An Astounding Baseline for Recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2014, pp. 512–519.https://ieeexplore.ieee.org/document/6909475

R. R. Selvaraju et al., "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2017, pp. 618–626.https://ieeexplore.ieee.org/document/8237336

A. Vetrivel et al., "Disaster Damage Detection from UAV Imagery Using Deep Learning," *Remote Sens.*, vol. 10, no. 2, p. 239, 2018.https://www.mdpi.com/2072-4292/10/2/239

K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv:1409.1556*, 2014. Source https://arxiv.org/abs/1409.1556

M. Tan and Q. Le, "EfficientNetV2: Smaller Models and Faster Training," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2021, pp. 10096–10106. https://proceedings.mlr.press/v139/tan21a.html

Y. Wang et al., "A Retrospective Evaluation of Energy-Efficient Object Detection Solutions on Embedded Devices," in *Proc. Design, Autom. Test Europe Conf. Exhibition (DATE)*, 2018, pp. 709–714. https://ieeexplore.ieee.org/document/8342126

Y. Wang et al., "Large: Learning Latent Relationships with Adaptive Graph Embedding for Aerial Scene Classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 2, pp. 621–634, 2018. https://ieeexplore.ieee.org/document/8022925

Y. Zhao et al., "Saliency Detection and Deep Learning-Based Wildfire Identification in UAV Imagery," *Sensors*, vol. 18, no. 3, 2018. https://www.mdpi.com/1424-8220/18/3/712

"DJI Matrice 100," *DJI*, 2023. [Online]. Available: https://www.dji.com/matrice100

E. Bañuelos et al., "UAV-Based Flood Mapping and Emergency Response," *Remote Sens.*, vol. 12, no. 3, p. 515, 2020. https://doi.org/10.3390/rs12030515

X. Chen et al., "Multi-scale Attention Networks for Aerial Disaster Scene Classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024, Art. no. 5103312. https://doi.org/10.1109/TGRS.2024.5103312

M. R. Shafqat et al., "UAV-Edge Fusion: Real-Time Multispectral Fusion on UAV Edge Hardware for Emergency Monitoring," *IEEE Internet Things J.*, vol. 11, no. 1, pp. 219–230, 2024. https://doi.org/10.1109/JIOT.2023.3339823

A. Ben Hamida et al., "A Comprehensive Review of Deep Learning-Based Methods for Image Captioning: Datasets, Metrics, Models and Challenges," *Electronics*, vol. 10, no. 13, p. 1549, 2021. https://doi.org/10.3390/electronics10131549

S. Ibrahem and M. Ethel, "Aerial Image Classification for Disaster Response Using Vision Transformers," *arXiv:2503.02465*, 2025. Source https://arxiv.org/abs/2503.02465

T. Bouabdellah et al., "A Deep Learning-Based Approach for Real-Time Wildfire Detection Using UAV Imagery," *Drones*, vol. 7, no. 7, p. 436, 2023. Source

https://www.sciopen.com/article/10.14067/j.cnki.1673-923x.2024.03.003?issn=1673-923K

K. V. S. R. Anjaneyulu et al., "UAV-Based Real-Time Forest Fire Detection Using Deep Learning," *Remote Sens.*, vol. 16, no. 5, p. 879, 2024. Source https://www.frontiersin.org/journals/forests-and-global-change/articles/10.3389/ffgc.2023.1134942/full