Drone Detection in Edge Devices: An Efficient CNN-based Model with Knowledge Distillation

Maryam Lawan Salisu
Dept. of computer Science
Federal University Dutse
Jigawa, Nigeria
marylsgumel@gmail.com

Farouk Lawan Gambo
Dept. of Cyber Security
Federal University Dutse
Jigawa, Nigeria
farouk4142@gmail.com

Aminu Musa
Dept. of computer Science
Federal University Dutse
Jigawa, Nigeria
musa.aminu@fud.edu.ng

Aminu Aliyu Abdullahi
Dept. of computer Science
Federal University Dutse
Jigawa, Nigeria
aminuabdullahisumaila@gmail.com

***Abstract*- Drones have become prevalent nowadays, with many applications including security, public safety, agriculture, and environmental monitoring security. However, as with any other disruptive technology, there exists a risk. Flying drones in unauthorized airspace could compromise security which poses a threat to national security. Therefore, it is imperative for efficient and real-time drone detection mechanisms to ensure security and safety. Deep learning models on the other hand have proven to be accurate in object detection. Unfortunately, deep learning models are not practical to be deployed on edge devices due to the resource demanding nature of the models. This paper presents an approach to drone detection using deep learning techniques, specifically tailored for edge devices using knowledge distillation. We collected an open source dataset named "Drone vs. Birds" obtained from Kaggle and used Convolutional Neural Network (CNN) with Knowledge Distillation {KD) for the detection. The proposed model achieved acceptable accuracy of 92% and Mean Average Precision (mAP) of 88%. However the proposed model is lightweight making it efficient for deployment on edge devices. .The findings demonstrate the feasibility of real-time drone detection on edge despite the resource constraints of edge devices**

Keywords: Drones, Object detection, CNN, Deep learning

I. INTRODUCTION

The proliferation of Unmanned Aerial Vehicles (UAVs), commonly known as drones, has brought about significant advancements in various industries due to their versatility, mobility, and potential to revolutionize tasks previously considered labor-intensive [2]. Drones are remotely controlled or autonomous aircraft, ranging in size from small quadcopters to huge fixed-wing aircraft [1] [2]. They have a variety of sensors and technology, including GPS, cameras, and communication systems, which enables them to carry out a variety of duties [1], [2].

Drones find applications across numerous sectors, including surveillance, agriculture, search and rescue, filmmaking, and package delivery [2]. Their ability to access remote or hazardous areas and capture high-resolution imagery has made them indispensable tools in these domains [2]. The rapid growth of drones has opened up new possibilities for these industries. Agriculturists employ drones to optimize crop management through aerial analysis, while film makers harness their capabilities for breathtaking cinematic shots [2]. Additionally, drones play an integral role in search and rescue operations, enabling access to hard-to-reach terrains. However, these very attributes that make drones indispensable also raise pertinent concerns regarding privacy, security, and safety [1],[2]. The chances for misuse and unauthorized drone flights in restricted airspace highlight the need for effective drone detection systems [10].

Machine Learning (ML) is referred to as systems changes which performs tasks that are associated with artificial intelligence [17]. Machine learning techniques have shown promise in addressing the security challenges associated with drones. Deep Learning's ascent within the realm of machine learning technology has redefined the boundaries of what machines can learn and accomplish [6]. Deep learning algorithms have demonstrated exemplary performance in detection of complex image learning problems [18, 19, 20].

Researchers have explored the use of Convolutional Neural Networks (CNNs) for drone detection, leveraging their ability to automatically learn features from images [3]. However, these models often suffer from computational complexity, making real-time detection infeasible, especially on resource-constrained devices [3].

To tackle this issue, this paper proposes a novel approach by introducing a knowledge distillation based real-time drone detections model for efficient drone detection. Knowledge distillation is employed to transfer the knowledge from a complex model to a simplified lightweight model, enabling accurate detection while minimizing computational demands [4]. This distilled model maintains a balance between accuracy and efficiency, making it well-suited for drone detection applications. By employing this technique, the paper aims to contribute to the development of effective and efficient solutions to the growing concern of drone detection in various contexts.

This paper presents a novel approach to address the pressing issue of drone detection in resource-constrained environments. The primary objectives of this research are to develop a highly efficient Convolutional Neural Network (CNN) model tailored for real-time drone detection and to employ knowledge distillation techniques for model compression and deployment on edge computing platforms.

The paper makes a significant contribution to the field by introducing an innovative model architecture that strikes a balance between accuracy and computational efficiency, enabling practical deployment of drone detection systems in edge environments. Additionally, the application of knowledge distillation enhances the model's lightweight properties, further advancing the feasibility of edge-based drone detection solutions, with promising implications for a wide range of security and surveillance applications.

The rest of this paper is divided into Literature Review, where Existing research in drone detection is presented in Section II, the Methodology- methods and techniques used, the dataset collection, research design, data analysis etc. to conduct this research is explained in Section III, "Result and Discussion" where the research findings are presented and interpreted is discussed in Section IV, and finally Summary and Conclusion, here the concise overview of the main points and final insights of the finds are presented in Section V.

II. LITERATURE REVIEW

This section explains some basic terminologies and concepts used in this paper which include drone, their types and application. It also presents the realm of deep learning in object detection. A survey on existing research in drone detection is also presented which has witnessed the integration of machine learning techniques, particularly CNNs, to address the challenge of accurate and real-time detection. Finally, a brief overview of Knowledge distillation is presented to help toward understanding the concept used in this paper

A. Drones, Their Types and Applications

Drones or Unmanned Aerial Systems (UAV - Unmanned Aerial Vehicle or UAS - Unmanned Aerial Systems) are the aircrafts, which are able to fly without a pilot and passengers on board [1]. Drone Controlling is performed remotely by radio waves or autonomously (with a predetermined route). Drones do not have a specific size or type of a drive. They are often equipped with accessories used for surveillance and monitoring, in the form of the optoelectronic heads [1]. Drones have garnered immense attention due to their diverse applications across various domains. Drones come in various types, ranging from quadcopters, fixed-wing UAVs, to hybrid models. Each type serves specific purposes, with quadcopters excelling in agile maneuvering and hovering, while fixed-wing UAVs are optimal for long-range surveillance and mapping applications.

The applications of drones span a wide spectrum of industries. In agriculture, drones equipped with multispectral cameras enable precision farming through crop health analysis [2]. In disaster management, drones assist in search and rescue missions by accessing hard-to-reach locations [1, 2].

Filmmakers harness drones for cinematic aerial shots, while logistics companies explore drone delivery systems [2]. These applications highlight the versatility and potential of drones across diverse sectors.

B. Deep Learning for object Detection

Deep Learning's ascent within the realm of machine learning technology has redefined the boundaries of what machines can learn and accomplish [6]. With its capacity to learn intricate patterns from vast and complex datasets, deep learning has catalyzed transformative changes across domains. As research continues, deep learning is poised to advance further, addressing its challenges and unlocking novel capabilities that will shape the future of technology and human interaction.

Drone technology often involves object detection, a critical task for applications like surveillance, monitoring, and tracking [1, 7]. Detecting objects, such as vehicles, pedestrians, or even other drones, is essential for ensuring safe and efficient operations. Object detection is typically achieved through computer vision techniques, with Convolutional Neural Networks (CNNs) emerging as a prominent approach. CNNs excel at learning hierarchical features from images, enabling accurate object localization.

C. Knowledge Distillation

Knowledge distillation is a technique wherein knowledge from a complex model, often referred to as the teacher model, is transferred to a simpler model, the student model. This process involves training the student model to mimic the outputs of the teacher model. Knowledge distillation serves various purposes, including model compression, transfer learning, and improving the generalization of smaller models [8].

The primary motivation behind knowledge distillation in drone detection on edge devices is model compression. Large neural networks with millions of parameters are computationally expensive and memory-intensive, which is a significant limitation on edge devices with limited resources. In knowledge distillation two models called teacher and students are trained. The student model is a shallow CNN model which retains the performance of a larger teacher model and tends to have faster inference times, making it suitable for real-time applications like drone detection, where real-time decisions are essential. However, the model may lose certain fine-grained information, the student model is dependent on a competent teacher model. Knowledge distillation is susceptible to architectural changes between teacher and student models.

D. Related Works

B. Taha et al. [3] offer a comprehensive review of machine learning's role in drone detection and categorization, noting its relevance due to growing drone presence. They cover radar, visual, acoustic, and radio-frequency methods, observing potential in individual efforts but highlighting the need for standardized specifications and reference datasets for meaningful comparisons in this evolving field. Also, F Mahdavi et al. [9] addresses drone detection using a fisheye camera. It compares CNN, SVM, and nearest neighbor methods, showing CNN's superior accuracy (95%) in contrast to SVM (88%) and nearest neighbor (80%).

M. Wu et al. [13] Implements YOLOv2. Created a semi-automatically labeled dataset using the KCF tracker. Modifications were made to the YOLOv2 model's structure and anchor box parameters. Training occurred on proprietary and public datasets, demonstrating the system's ability to achieve real-time, accurate drone detection.

A. Sharma and N. Jain et al. [4] discuss a new method, the tiny-YOLO v4 algorithm which stands out for its fast speed and accurate detection. Adding a stereo triangulation-based depth module enables 3D positioning. The researchers provide pre-trained models, simulation code, and a custom dataset. However, future plans involve extending the method for long-range drone detection using UAV-Yolo.

Another work of [5] [10] discusses the YOLOv4 and YOLOv5 algorithms. The models were trained on drone and bird datasets and evaluated using various metrics including mAP, FPS, precision, recall, and F1-score. YOLOv4 has mAP of 74.36%, Precision of 0.95, recall of 0.68, and F1-score of 0.79. On the other hand, YOLOv5 was more efficient with F-1 score of 93.63%, precision of 96.2%, mAp of 95%, recall of 91.2%, and accuracy of 95.5%.

Though the collection of drones, birds and Aircraft datasets is not easy to obtain, [12] uses YOLOv4 with Darknet as backbone and achieves accuracy of 98.3% with dataset collected from three global and certified datasets in aircraft detection systems and the result was a dataset containing all cases. On the other hand, [15] uses novel computer vision framework "convet" and yield 99.69% AP.

Nevertheless, [11] uses YOLOv5 with a manual dataset of 5460 samples and achieves highest accuracy of 99.6%, sensitivity of 99.80% and F1 score of 99.69%. Overall, the study contributes to an advanced drone detection system with impressive results. However, the study was limited to YOLO implementation only since various object detection algorithms require datasets to be labeled in certain formats, which is time consuming. Unfortunately, there is no room for comprehensive comparison with other state-of-the-art drone detection systems, limiting the understanding of system applicability. Furthermore, the study lacks discussion on encountered challenges, warranting investigation into practical feasibility and robustness.

Also, [24] focuses on the detection and identification of four-rotor, six-rotor, and fixed-wing UAVs in low-altitude, complex settings with high precision and light weight. By integrating deep learning target detection with superresolution feature improvement, a lightweight UAV identification model is built and field-tested for verification, resolving the YOLOv4 network's shortcomings for small UAV target recognition by improving the network's accuracy with the addition of large-scale detection layers, anchor improvements, and enhanced small target images. The experimental validation shows a 7.1% increase in mean average accuracy (mAP) to 86.7%, as well as increased precision and recall.

Most of the previous studies on UAV identification relied on various feature-extraction techniques, which are computationally expensive. Therefore, [14] proposes an end-to-end deep-learning-based model to detect and identify UAVs based on their radio frequency (RF) signature, and yields an overall accuracy, precision, sensitivity, and F1-score of 97.53%, 98.06%, 98.00%, and 98.00% respectively.

Additionally, [23] provide a unique drone classification model based on the proposed thin pattern and an iterative neighborhood component analysis (INCA) feature selector based on ADr sounds. To demonstrate the general success of the provided model, 16 classifiers were utilized in five categories: decision tree (DT), discriminant (D), support vector machine (SVM), k closest neighbor (kNN), and ensemble classifiers (EC). Using Fine kNN, classification accuracy was calculated to be 99.72%. The suggested model outperformed previous ADr detection/classification models in comparison.

This literature review underscores the significance of drones and their multifaceted applications. Object detection plays a pivotal role in enabling various drone tasks, and the integration of machine learning techniques, especially CNNs, has shown promise in this domain. Knowledge distillation emerges as a viable solution to address the computational challenges associated with real-time drone detection. The synthesis of these elements forms the basis for the proposed paper's investigation into an efficient CNN-based drone detection model using knowledge distillation, offering a novel approach to optimize the balance between accuracy and computational efficiency in drone detection systems.

III. METHODOLOGY

This paper explores the application of knowledge distillation to enhance the efficiency of CNN-based drone detection models. By transferring the insights of a complex CNN-based model to a simpler version. The proposed methodology consists of the following steps: dataset acquisition and preprocessing, knowledge distillation, model development and evaluation. This approach addresses the computational limitations that hinder real-time detection on devices with restricted processing power.

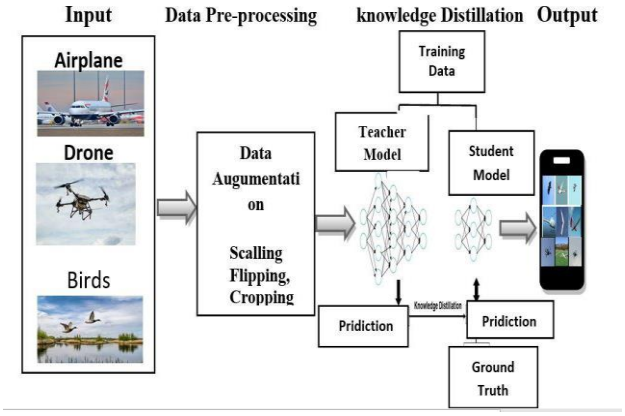

Fig .2: Proposed Methodology

A. Dataset Acquisition and Preprocessing

The dataset used in this experiment is an open source dataset named "Drone vs. Birds" dataset obtained from Kaggle [16], a popular platform for data science and machine learning enthusiasts. The dataset is designed purposely for image classification to identify whether it is a Bird or Drone image in the sky. Originally, the dataset contains two directories with a total number of 828 images of Birds and Drones combined. The dataset has 400 images of birds and 428 drones respectively. Additional 194 images of the airplane are collected using various preprocessing techniques such as data augmentation, flipping and randomly sourcing over the internet and added to the dataset to create a third airplane class in the dataset. This is

necessary to ensure the class balance among the class of objects under investigation.

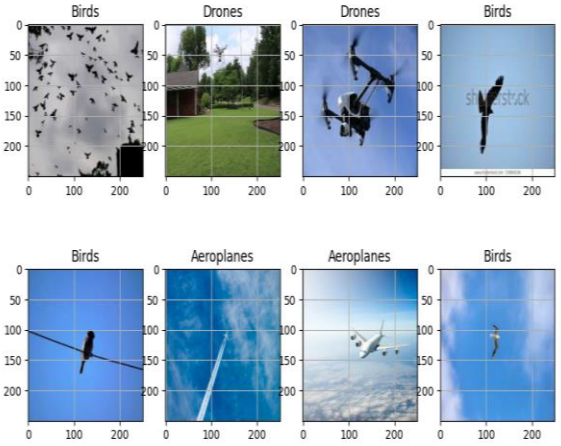

Fig: 1 sample dataset images [16]

B. Knowledge Distillation

Knowledge distillation is a technique of transferring knowledge from a complex, pre-trained model (the teacher) to a simpler, more lightweight model (the student) [22]. This technique enables the student model to achieve the generalization and reliability of the teacher model while maintaining lower computational requirements. The teacher model in the context of drone identification is often a deep neural network, such as a convolutional neural network (CNN) that has been trained on a huge dataset of varied drone images.

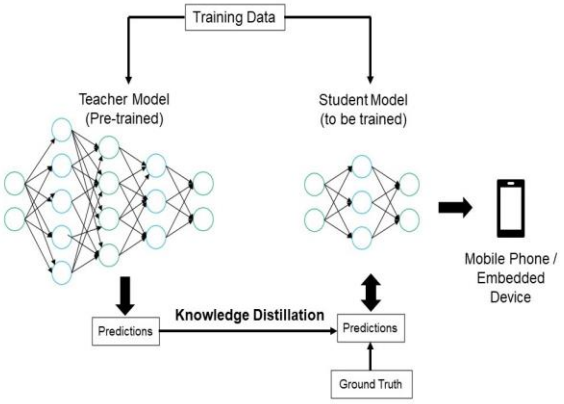

Figure 3: knowledge distillation technique

C. Model Development

To build a shallow student image classification model that retains the accuracy of a pre-trained teacher model while lowering computational requirements of complex CNN-based models for edge device adoption the concept of knowledge distillation is utilized [21]. The teacher model employed in this paper is a mobilenet_v2 [15]. A deep pre-trained CNN model trained on imagenet and obtained a state-of- the-art accuracy of 91.06% on top 5k accuracy.

The model used has 51 convolutional layers, 3.4M trainable parameters. Even though mobilenet was proposed to allow deployment of CNN model on edge devices, it is still big for edge devices with limited resources. Therefore, in this paper, mobilenet was employed and fine-tuned as a teacher model.

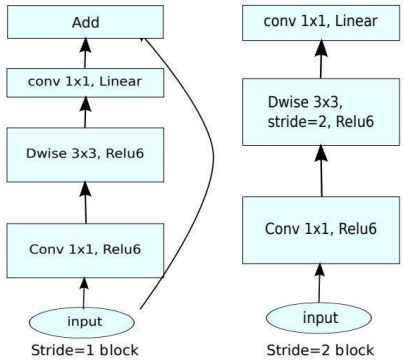

Fig. 4: Mobilenet Architecture [15]

The student model is a simplified CNN built from scratch. The student model has fewer layers and trainable parameters compared to the teacher model. The student model is optimized for efficiency and low computing complexity. The model takes the dataset as input and soft labels, which are the logits of the last layer from the teacher model as the distilled knowledge. The student model is trained on the soft labels in the training phase I, with temperature T. Where T is an integer greater than 1, distillation loss is obtained by computing a kullberg-lavenberg on the cross-entropy loss. In the training phase II, the student model is trained on the ground truth and temperature T=1 until student loss is obtained. Finally, the sum of the two losses is computed to obtain the general loss which is shown in Fig. 4

The model hyper-parameters were tuned several times, experimenting with different values for learning rate and number of layers until a desired result was obtained. The best result is obtained by selecting three Conv2d layers followed by two maxpooling layers, dropout layers, one batch normalization layer, one flattens and dense layers. The best learning rate is 0.001.

```
Model: "student"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 125, 125, 32)      896

 activation (Activation)     (None, 125, 125, 32)      0

 max_pooling2d (MaxPooling2D  (None, 125, 125, 32)     0
 )

 dropout_1 (Dropout)         (None, 125, 125, 32)      0

 conv2d_1 (Conv2D)           (None, 63, 63, 64)        18496

 activation_1 (Activation)   (None, 63, 63, 64)        0

 max_pooling2d_1 (MaxPooling  (None, 32, 32, 64)       0
 2D)

 dropout_2 (Dropout)         (None, 32, 32, 64)        0

 conv2d_2 (Conv2D)           (None, 16, 16, 128)       73856

 activation_2 (Activation)   (None, 16, 16, 128)       0

 batch_normalization (BatchN  (None, 16, 16, 128)      512
 ormalization)

 dropout_3 (Dropout)         (None, 16, 16, 128)       0

 flatten (Flatten)           (None, 32768)             0

 dense_2 (Dense)             (None, 3)                 98307

=================================================================
Total params: 192,067
Trainable params: 191,811
Non-trainable params: 256
```

Fig.5 shows the architecture of the student model.

The process of distilling knowledge from the teacher model to student model can best be described mathematically in the equations below.

$$\text{TM} = yi(x/t) = \frac{e^{\frac{z^{i}(x)}{t}}}{\sum_j e^{\frac{z^{j}(x)}{t}}} \quad (1)$$

Where t is a parameter called temperature and Z*i*: is the logit of the input x.

The teacher model is trained with T>1 to obtain soft labels.

In student training phase 1, the model is trained to predict soft labels from the teacher model with softmax activation function and T with the same value as that of teacher training.

To obtain Distillation loss, the loss from training phase 1 is computed using equation 2.

Dist Loss =

$$\frac{i}{m}\sum_{i=1}^{c} 2t2\ ^{2}\alpha D_{kl}(p^{j}, q^{j}) - (1-\alpha)\sum_{i=1}^{c}(yi)^{j} log(1 \quad (2)$$

Such that: m is the batch size, P: is the soft labels from teacher network, q: is the softmax scores of student model, Dkl: is the Kullback- Liebler (KL) divergence between p and q and α: is the relative importance of the teacher's control.

In student training phase II, the student model is trained on ground truth with softmax activation function and temperature T=1. The student loss is obtained using equation 3.

$$\text{SM loss} = yi(x/t) = \frac{e^{\frac{z^{i}(x)}{t}}}{\sum_j e^{\frac{z^{j}(x)}{t}}} \quad (3)$$

With t=1.

Finally general loss of the model can be computed by taking the sum of the two losses as shown in equation 4.

$$\text{General loss} = \alpha\ (SM\ loss) + (1-\alpha) Dist\ loss \quad (4)$$

The entire proposed model framework can better be explained diagrammatically as depicted in Fig.5.

The proposed knowledge distillation approach uses the knowledge gained by a bigger instructor model (teacher model) to train a more efficient student image classification model. This method strikes a balance between model size and accuracy, making it appropriate for resource-constrained applications while maintaining competitive performance.

D. *Experimental Setup*

To implement the experiment proposed in this paper, a python programming language is used with Tensorflow machine learning framework. The experiments are carried out on Google Collaboratory (Colab) with GPU T4 support, 107 GB hard disk and a 12 GB RAM virtual machine.

E. Evaluation Metrics

Accuracy is utilized in this work, being one of the most popular evaluation metrics used for evaluating classifier performance of a balance class. Mean Average Precision (mAP) metric is also used to evaluate object detection models, which enable us to monitor the performance of the student model
Accuracy = (TP + TN) / (P+N),
where P and N are total positive and negative classes.
Accuracy tells what proportion of negative and positive classes are correctly classified.

Formula

$$mAP = \frac{1}{n}\sum_{k=1}^{k=n} AP_k$$

AP_k = the AP of class k

$n = the\ number\ of\ classes$

Mean Average Precision (mAP) values are calculated over recall values from 0 to 1.
Training Time
Inference time
Number of parameters

IV. RESULT AND DISCUSSION

The results of the experiments were recorded in this section of the paper, various graphs were plotted with different values of accuracy against epoch for the trainer model and proposed model (see Figure 6a, 6b, 7a and 7b). Also the model size, training time, number of parameters of the teacher model, student model and proposed model were compared and presented in Table 1. The average performance metrics across the 10 experiments ran are recorded.

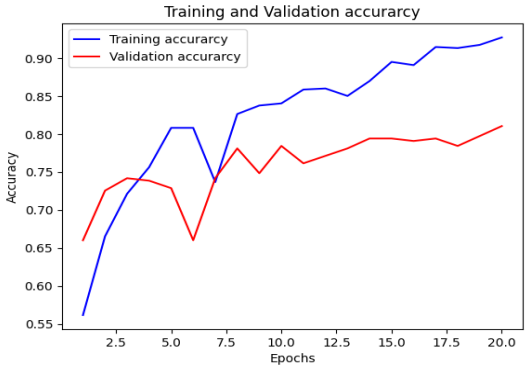

Fig. 6a Accuracy of teacher model

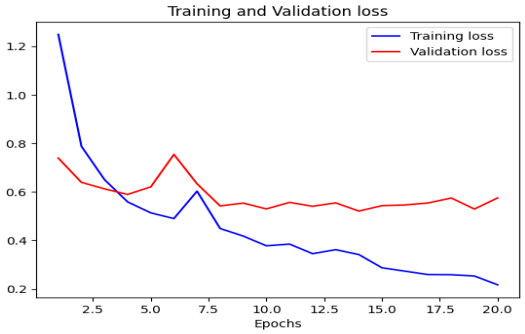

Fig. 6b Loss of teacher model

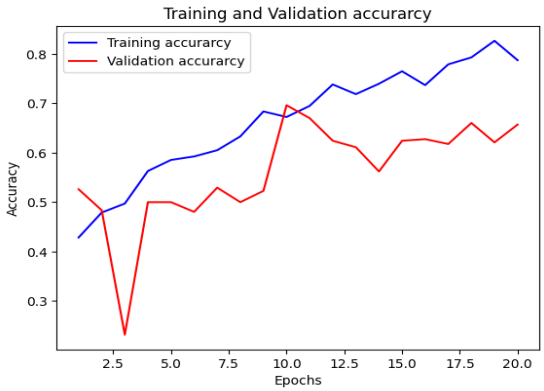

Fig. 7a Accuracy of proposed model

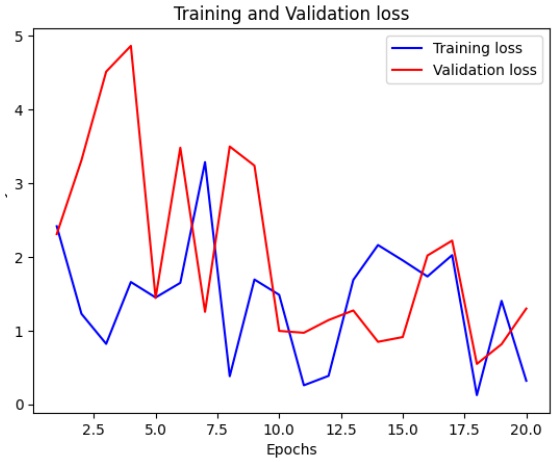

Fig. 7b Loss of proposed model

Table 1: Performance comparison of teacher, student and proposed model.

Models	**Accuracy (%)**	**Model size**	**Training time**	**No. of parameters**
Teacher	0.9272	17mb	1m20s	2,915,395
Student	0.6667	787kb	1m 1s	192,067
Proposed model	0.8263	787kb	1m2s	192,067

It can be observed that, KD technique used has the potential of building an efficient model that can detect flying drones. The proposed model also has fewer parameters as compared to the teacher model. This implies that it will require less storage space as compared to the teacher model. The proposed model recorded an accuracy of 82.6% which is far better than the student model, these findings suggest that the proposed model can match teacher level accuracy while keeping the computational requirement low.

The performance achieved by the proposed model is realistic to be deployed on edge devices efficiently and can be used to detect drones accurately. To ascertain how good our proposed model is through comparison with existing literature is however limited to only accuracy as other studies have not evaluated their model using mAP and model size. To make comparison regarding model size is still an open issue. However, the model size of our model verifies the effectiveness and feasibility in terms of computational complexity.

The model was further compared with existing drone detection models found in the literature, and the results are presented in Table II.

Table II: Comparison of result with existing models

Authors	Accuracy	Precision	Recall	Model size
K S Bhavishya et. al.	0.955	0.962	0.912	-
S. O. Ajakwe	0.996	0.997	0.998	-
S. S. Alam *et al.*	0.975	0.980	0.980	-
Proposed Model	0.826	0.87	0.81	786 kb

V. Conclusion and Future Work

This paper presents a convolutional neural network -based model with knowledge distillation technique for drone detection. Previous attempts to produce a lightweight model of drone detection suffer from performance/compression tradeoff. Efforts made to reduce the complexity and computing resources demand of the existing CNN-based drone detection models to produce a lightweight model result in reduced performance. However, the proposed model maintains a balance between accuracy and compression. This paper addressed the problem of high resource utilization by existing drone detection models by proposing a lightweight model of drone detection for edge devices using knowledge distillation techniques. The proposed models have proven to be lightweight, efficient and accurate when tested, Therefore, the model can be efficiently deployed on edge devices despite the resource constraints of the edge devices environment.

It is noteworthy that one of the major limitations of this work, is that the model cannot detect multiple instances of drones in a single image, as such cannot be used for real-time detection. As stated in the introduction, this is research in progress. In future, the work will be extended by collecting more dataset to propose a lightweight real-time- based drone detection using YOLO-v3 and knowledge distillation.

ACKNOWLEDGEMENTS

We acknowledge TETFund National Research Fund (TETF/DR&D-CE/NRF2021/SETI/ICT/00133)-2021 intervention for its support.

REFERENCES

[1] P. Kardasz and J. Doskocz, “Drones and Possibilities of Their Using,” Journal of Civil & Environmental Engineering, vol. 6, no. 3, 2016, doi: https://doi.org/10.4172/2165-784x.1000233.

[2] R. Hossain, “A Short Review of Drone Technology,” International Journal of Mechatronics and Manufacturing Technology , vol. Volume 7, no. Issue 2.

[3] B. Taha, Student Member,Ieee), and A. Shoufan, “Machine Learning-Based Drone Detection and Classification: State-of-the-Art in Research.,” IEEE Access, vol. VOLUME 7, 2019.

[4] A. Sharma, N. Jain, and M. Kothari, “Lightweight Multi-Drone Detection and 3D-Localization via YOLO. ,” Feb. 2022, Available: arxiv:2202.09097v1 [cs.CV]

[5] S. Singha and B. Aydin, “Automated Drone Detection Using YOLOv4,” Drones, vol. 5, no. 3, p. 95, Sep. 2021, doi:https://doi.org/10.3390/drones5030095.

[6] A. Chahal and P. Gulia , “Machine Learning and Deep Learning,” International Journal of Innovative Technology and Exploring Engineering, vol. 8, no. 12, pp. 4910–4914, Oct. 2019, doi: https://doi.org/10.35940/ijitee.l3550.1081219.

[7] Y. Amit, “Object Detection.,” University of Chicago Pedro Felzenszwalb, Brown University Ross Girshick, Facebook AI Research.

[8] G. Hinton, O. Vinyals , and J. Dean, “Distilling the Knowledge in a Neural Network,” Mar. 2015, Available: arxiv:1503.02531v1 [stat.ML]

[9] F. Mahdavi and R. Rajabi, “Drone Detection Using Convolutional Neural Networks. ,” 6th Iranian C6th Iranian Conference on Signal Processing and Intelligent Systems (ICSPIS)., 2020.

[10] K. Bhavishya, K. Reddy, G. Vamsi, and I. Krishnachaithanya, “Drone Detection and Surveillance,” 2023.

[11] S. O. Ajakwe, V. U. Ihekoronye, D.-S. Kim, and J. M. Lee, “DRONET: Multi-Tasking Framework for Real-Time Industrial Facility Aerial Surveillance and Safety,” Drones, vol. 6, no. 2, p. 46, Feb. 2022, doi: https://doi.org/10.3390/drones6020046.

[12] A. Naseri, N. Hussein, and M. Ali, “Detection of drones with YOLOv4 deep learning algorithm,” Int. J. Nonlinear Anal. Appl, vol. 13, no. 2008–6822 (electronic), pp. 2008–6822, 2022, doi: https://doi.org/10.22075/ijnaa.2022.27556.3648.

[13] M. Wu, W. Xie, X. Shi, P. Shao, and Z. Shi, “Real-Time Drone Detection Using Deep Learning Approach,” Machine Learning and Intelligent Communications, pp. 22–32, 2018, doi: https://doi.org/10.1007/978-3-030-00557-3_3.

[14] S. S. Alam et al., “RF-Enabled Deep-Learning-Assisted Drone Detection and Identification: An End-to-End Approach,” Sensors, vol. 23, no. 9, p. 4202, Apr. 2023, doi: https://doi.org/10.3390/s23094202.

[15] M. Sandler, A. W. Howard, M. Zhu, Andrey Zhmoginov, and L.-C. Chen, “MobileNetV2: Inverted Residuals and Linear Bottlenecks,” arXiv (Cornell University), Jan. 2018, doi: https://doi.org/10.48550/arxiv.1801.04381.

[16] H. Walia, “Birds vs Drone Dataset,” www.kaggle.com. https://www.kaggle.com/datasets/harshwalia/birds-vs-drone-dataset (accessed Sep. 06, 2023).

[17] M., Mohri, A., Rostamizadeh, & A. Talwalkar, “Foundations of machine learning. Cambridge, Mass: Cambridge University Press”.2012.

[18] F.L. Gambo1, et al. “Performance Comparison of Convolutional and Multiclass Neural Network for Learning Style Detection from Facial Images” EAI Endorsed Transactions on Scalable Information Systems. 2021

[19] F.L Gambo et al. “Prediction of learning style using facial expressions with convolutional neural network”. International Journal of Information Processing and Communication; 2020. 9 (special issue), 389-399.

[20] F. L. Gambo, G. M. Wajiga,& E. J. Garba, ‘A Conceptual Framework for Detection of Learning Style from Facial Expressions using Convolutional Neural Network’, in 2019 2nd International Conference of the IEEE Nigeria Computer Chapter (NigeriaComputConf), 2019, pp. 1–5.

[21] A. Musa, M. Hamada, and M. Hassan, “A Theoretical Framework Towards Building a Lightweight Model for Pothole Detection using Knowledge Distillation Approach,” SHS Web of Conferences, vol. 139, p. 03002, 2022, doi: https://doi.org/10.1051/shsconf/202213903002.

[22] A. Musa, M. Hassan, M. Hamada, and F. Aliyu, “Low-Power Deep Learning Model for Plant Disease Detection for Smart-Hydroponics Using Knowledge Distillation Techniques,” Journal of Low Power Electronics and Applications, vol. 12, no. 2, p. 24, Apr. 2022, doi: https://doi.org/10.3390/jlpea12020024.

[23] E. Akbal, A. Akbal, S. Dogan, and T. Tuncer, “An automated accurate sound-based amateur drone detection method based on skinny pattern,” Digital Signal Processing, vol. 136, p. 104012, May 2023, doi: https://doi.org/10.1016/j.dsp.2023.104012.

[24] Y. Dong, Y. Ma, Y. Li, and Z. Li, “High-precision real-time UAV target recognition based on improved YOLOv4,” Computer Communications, vol. Volume 206, Apr. 2023, doi: https://doi.org/10.1016/j.comcom.2023.04.019.