SafeSpace MFNet: Precise and Efficient MultiFeature Drone Detection Network

mahnoor dil [1], misha urooj khan [1], Muhammad Zeshan Alam [1], Farooq Alam Orakzai [1], Zeeshan Kaleem [2], and Chau Yuen [1]

[1]Affiliation not available
[2]COMSATS University Islamabad

October 30, 2023

Abstract

Unmanned air vehicles (UAVs) popularity is on the rise as it enables the services like traffic monitoring, emergency communications, deliveries, and surveillance. However, the unauthorized usage of UAVs (a.k.a drone) may violate security and privacy protocols for security-sensitive national and international institutions. The presented challenges require fast, efficient, and precise detection of UAVs irrespective of harsh weather conditions, the presence of different objects, and their size to enable SafeSpace. Recently, there has been significant progress in using the latest deep learning models, but those models have shortcomings in terms of computational complexity, precision, and non-scalability. To overcome these limitations, we propose a precise and efficient multiscale and multifeature UAV detection network for SafeSpace, i.e., \textit{MultiFeatureNet} (\textit{MFNet}), an improved version of the popular object detection algorithm YOLOv5s. In \textit{MFNet}, we perform multiple changes in the backbone and neck of the YOLOv5s network to focus on the various small and ignored features required for accurate and fast UAV detection. To further improve the accuracy and focus on the specific situation and multiscale UAVs, we classify the \textit{MFNet} into small (S), medium (M), and large (L): these are the combinations of various size filters in the convolution and the bottleneckCSP layers, reside in the backbone and neck of the architecture. This classification helps to overcome the computational cost by training the model on a specific feature map rather than all the features. The results show significant performance gain even for unseen feature maps with minimal loss in accuracy. Results show a significant reduction in training parameters, inference, and increased pattern in FPS and GFLOPs for \textit{MFNet} compared to YOLOv5s. \textit{MFNet-M} performance evaluation in terms of precision, recall, mean average-precision (mAP), and IOU increased around 1.8\%, 2.2\%, 0.9\%, 1.7\% compared to YOLOv5s. Furthermore, \textit{MFNet-M} achieves the best performance with 96.8\% precision, 88.4\% recall, 95.9\% mAP, and 51.1\% IoU for UAV detection. The dataset and code are available as an open source: github.com/ZeeshanKaleem/MultiFeatureNet.

SafeSpace **MFNet**: Precise and Efficient **MultiFeature** Drone Detection **Network**

Mahnoor Dil, Misha Urooj Khan, Muhammad Zeshan Alam, Farooq Alam Orakazi, Zeeshan Kaleem, *Senior Member, IEEE*, Chau Yuen, *Fellow, IEEE*

***Abstract*—Unmanned air vehicles (UAVs) popularity is on the rise as it enables the services like traffic monitoring, emergency communications, deliveries, and surveillance. However, the unauthorized usage of UAVs (a.k.a drone) may violate security and privacy protocols for security-sensitive national and international institutions. The presented challenges require fast, efficient, and precise detection of UAVs irrespective of harsh weather conditions, the presence of different objects, and their size to enable SafeSpace. Recently, there has been significant progress in using the latest deep learning models, but those models have shortcomings in terms of computational complexity, precision, and non-scalability. To overcome these limitations, we propose a precise and efficient multiscale and multifeature UAV detection network for SafeSpace, i.e., *MultiFeatureNet* (*MFNet*), an improved version of the popular object detection algorithm YOLOv5s. In *MFNet*, we perform multiple changes in the backbone and neck of the YOLOv5s network to focus on the various small and ignored features required for accurate and fast UAV detection. To further improve the accuracy and focus on the specific situation and multiscale UAVs, we classify the *MFNet* into small (S), medium (M), and large (L): these are the combinations of various size filters in the convolution and the bottleneckCSP layers, reside in the backbone and neck of the architecture. This classification helps to overcome the computational cost by training the model on a specific feature map rather than all the features. The results show significant performance gain even for unseen feature maps with minimal loss in accuracy. Results show a significant reduction in training parameters, inference, and increased pattern in FPS and GFLOPs for *MFNet* compared to YOLOv5s. *MFNet-M* performance evaluation in terms of precision, recall, mean average-precision (mAP), and IOU increased around 1.8%, 2.2%, 0.9%, 1.7% compared to YOLOv5s. Furthermore, *MFNet-M* achieves the best performance with 96.8% precision, 88.4% recall, 95.9% mAP, and 51.1% IoU for UAV detection. The dataset and code are available as an open source: github.com/ZeeshanKaleem/MultiFeatureNet.**

***Index Terms*—Birds, Multi-scale Detection, MultiFeatureNet, UAV Detection, YOLOv5s.**

I. Introduction

The market for unmanned aerial vehicles (UAVs, a.k.a drones) was valued at USD 10.72 billion in 2019, and by 2027, expected to grow to USD 25.13 billion [1], and this predicted surge is due to the rising need for automation and accelerated developments in technology. Due to their low price and ease of use, UAVs are employed for numerous tasks like surveillance, healthcare, animal tracking, and disaster response as shown in Fig. 1. But drones can contravene the security protocols of many national institutions by entering sensitive security areas. Any unauthorized organization or any individual carrying explosive and chemical materials could commit these breaches [2]. Therefore, drone misuse endangers and compromises public safety and security, and also, UAVs flying at low speeds upon collision in mid-air could result in aerial accidents. Those problems highlight the importance of automated drone detection technology, which can avert unnecessary drone interventions and quickly detect and deactivate unknown drones.

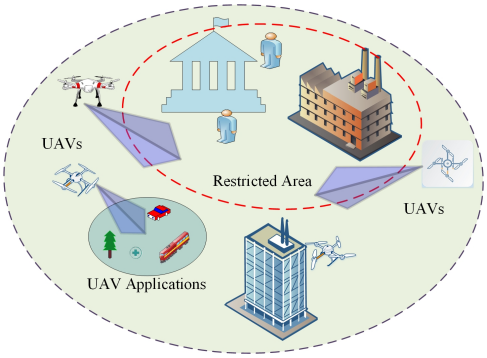


Fig. 1: UAV's applications and its implications in security-sensitive areas. UAVs have applications in numerous fields ranging from performing complex tasks for military to delivering foods in homes [2].

Kaleem *et al.* mentioned that drone neutralization approaches have improved ineffectiveness, but they rely on expensive and specialized equipment [3]. Therefore, high-performance and low-cost hardware-based drone detection systems are required. Usually, there are two ways of detecting UAVs: The first one is ground-to-air detection (GAD), in which cameras are installed on the ground to identify flying UAVs, and the second way is air-to-air detection (AAD), in which a flying UAV uses onboard cameras to identify other flying UAVs. In many GAD activities, ground cameras are immobile or move passively, whereas the background of target UAV shots is a bright or cloudy sky. A flying UAV in an AAD deployment may observe the target UAV from the top

Mahnoor Dil, Misha Urooj Khan, Zeeshan Kaleem, Farooq Alam Orakazi are with the Department of Electrical and Computer Engineering, COMSATS University Islamabad, Wah Campus, Pakistan. (e-mail: {noorijazhussain, mishauroojkhan, zeeshankaleem, farooqorakzai}@gmail.com)

Muhammad Zeshan Alam is with the Department of Computer Science, Brandon University, Canada. (e-mail: alamz@brandonnu.ca)

Chau Yuen is with the Engineering Product Development Pillar, The Singapore University of Technology and Design, Singapore. (e-mail: yuenchau@sutd.edu.sg)

This work was supported by the Higher Education Commission (HEC) Pakistan under the NRPU 2021 Grant#15687.

or side view perspectives. As a consequence, the background of the target UAV would have complex scenes such as urban or natural settings. As the onboard camera is flying dynamically, the appearance of the target UAV, such as its form, scale, and color, may change significantly [4], and this method results in less performance compared to GAD.

According to Lykou *et al.* in [5], around 6% commercial UAV detection systems designed on acoustic sensors, 26% are using radio frequency (RF), 28% are radar-based, and 40% adopted visual sensors for detection. Mostly those sensors adopted for drone detection and classification are: radar (radio detection and ranging on several different frequency bands for both active and passive sensing) [6], cameras for the visible spectrum sensing [4], cameras detecting thermal or infrared (IR) emissions [7], LIDAR (light detection and ranging) [8], microphones/acoustic sensors for acoustic vibration detection [2], and RF [9], [10] monitoring sensors for the detection of radio signals [11].

UAV detection methods using acoustics and RF sensing usually have higher costs or low range and accuracy of detection [6]. Contrarily, camera-based detection using visible images does not face these difficulties due to their high resolution, which is the main reason for their popularity in classification and object recognition. However, their utilization has several challenges, including light shifting, occluded sections, and crowded backgrounds, which necessitate the research of an effective detection method. The existing literature lacks significant research on UAV detection using thermal and IR cameras in challenging weather conditions [7]. Moreover, sensor fusion [8], [12] is one of the hottest open research areas that could further improve UAV detection accuracy.

A. Challenges and limitations

Advanced hardware with excellent accelerated abilities and deep learning technologies have made accurate and robust UAV detection possible. Convolution neural network (CNN)-the most basic deep learning model classifies UAVs by using visual and acoustic information [11], [13] and have improved more feature extraction technique than traditional object recognition algorithms. Complex deep learning models, such as YOLO (you only look once) [13], have excellent object recognition precision and speed compared to the basic models. They are also faster than region-based techniques due to their simplified architectural design. The proposed research methodologies in the existing literature faced the following challenges during drone detection and classification;

- **UAVs vs. birds classification:** UAVs are physically similar to birds, which generates the false alarm for birds during the UAVs identification stage.
- **Crowded backgrounds:** The inability to accurately segregate UAVs from the background makes UAVs detection difficult when present in a dense /cluttered background having clouds, flames, mist, sun, and smoke in the sky.
- **Different-size UAVs:** It's quite exigent to train a deep learning model which is sensitive to different-size UAVs [8].

B. Contributions

In this paper, we propose a precise and efficient multi-feature and multi-scale UAV detection network, i.e., SafeSpace *MultiFeatureNet* (MFNet), an improved version of the popular object detection algorithm YOLOv5s [14]. In MFNet, we perform multiple changes in the backbone and neck of the YOLOv5s to focus on the various small and ignored features required for accurate and fast UAV detection shown in Fig. 2. Moreover, we also conclude that *If we train a deep learning model on one particular feature set, then it can be equally beneficial for unseen features with minimal loss in accuracy and precision*. The key contributions are summarized here:

- Baseline YOLOv5s [14] takes an input image and extracts informative features by using a neck block that helps YOLOv5s generalize well on object scaling and identification of the object at different sizes. To further improve the feature sensitivity and kernel scale-ability of the YOLOv5s model, we modify the kernel size (KS) in the backbone and neck to change the size of extracted feature maps. This alteration would help the model to perform well on unseen data with improved precision. The proposed MFNet connections are the same as YOLOv5s with the difference in kernel sizes and their adjustment in the layers.
- Its a challenge to find which type of extracted features from feature pyramids would be the best for UAV detection in challenging backgrounds with complex conditions. We address this challenge by evaluating three different versions of MFNet: small (S), medium (M), and large (L). MFNet-S extracts small feature maps with the input grid size of the Conv and Bottleneck CSP layers set to 256×256 and 128×128, with a stride of 1 and 2. The proposed MFNet-M extracts medium feature maps as we set KS in the backbone to 512×512 and 256×256 in the neck. However, the MFNet-L generates larger feature maps than the previous ones as we fix the KS for the head and backbone to 1024×1024 and 512×512, respectively. This feature-based extraction provides optimized results for extracted feature size because it requires fewer parameters and a GPU requirement.
- Although MFNet is trained only on a single-size feature map, it performs well on images containing other-sized feature maps with minimal loss and good confidence scores.
- We consider only the famous drones vs. birds binary-class classification problem with challenging backgrounds. We also tested the trained model on kite images and videos (i.e., no drones and birds, and can be any image except these two) with our trained classifiers performed well with no misclassification.
- We train MFNet on a dataset with multiple environmental backgrounds such as fog, rain, sunlight, water, forest, and mountains images with complex scenarios in an image such as; the number of birds or drones, single bird or drone, varying drone size, multiple types of drones (General Atomics MQ-9 Reaper, Dassault nEUROn, DJI phantom), and different kinds of birds (eagle, parrot, etc.).

- MFNet extracts a minimal number of parameters and gradients compared to the *CT-Net-Middle* [15], *YOLOv5s* [15], *Improved YOLOv5s* [16] *Fine-tuned YOLOv5x [14] YOLOv5s* [17], *SAG-YOLOv5s* [18], *TransVisDrone* [19]. It proves the increased learning ability of MFNet for UAVs and birds and significantly decreases the false detection and missed alarms with improved precision, recall, and mAP compared to the available literature.

II. Literature Review

Recently, Li *et al.* in [1] adopted a software-defined radio (SDR) for the detection and classification of different types of jamming attacks on UAVs. They trained the conventional machine learning algorithms with a 35% false alarm (FA), while with deep learning models around 0.03% FA to perform spectrogram-based classification. The machine learning (ML) framework considering acoustic sensors protected institutional security from amateur drones. Mel-frequency and linear predictive cepstral coefficients with support vector machines (SVM) achieved 96.7% accuracy [2]. Zheng *et al.* in [4] utilized the monocular cameras to perform AAD on micro UAVs, which proved vital for vision-based swarm and malicious UAV detection. They introduced Det-Fly, a novel dataset containing 13,000 images of a flying target UAV in multiple circumstances. Detailed experimental evaluation of eight deep-learning systems resulted in the highest accuracy of 82.4%.

Authors in [6] used a hybrid synthetic framework with deep features for robust UAV classification and detection. They used acoustic, image/video, and wireless RF signals as system inputs. As UAV indoor flights place greater emphasis on stability and localization accuracy, Gerwen emphet al. in [8] used a flexible sensor fusion platform. The deployed system targeted the influence of multiple sensors on 3D indoor location accuracy when faced with different-sized UAVs. In [11], the authors adopted RF signals and spectral-based audio characteristics with SVM for drone identification and classification. They adjusted the spectral feature parameters for drone signal categorization. The achieved results reduced computation with improved classification performance.

Alsanad *et al.* improved YOLOv3 by using dense connecting modules and multiple-scale detection. The improved model was trained on drone images with a 70:30 ratio and achieved 95.60% accuracy, 0.36 mAP, and 60 FPS [13]. The authors presented the LIDAR-assisted UAV detection scheme in [20] to detect and track various types of drones with varied sizes. The experimental results proved that the LIDAR detected multiple UAVs at different ranges. In [21], SqueezeNet was used to extract multiple features from an RF dataset for a range of signal-to-noise ratio (SNR) [5-30]dB. The results showed significant improvement compared to the conventional machine learning classifiers like k-nearest neighbors (KNN) and SVM. Moreover, Wisniewski *et al.* in [22] adopted CNN to spot drones in real-time anti-UAV demonstration videos. They varied the parameters like model orientation, backdrop graphics, and textures via domain randomization in synthetic images. This proposal saved the time spent on individual drone labeling and provided a pixel-level mask of the drone's position.

Similarly, the authors proposed an online drone-based target detection system using an adaptive motion planner and feature pyramid feedback. The designed system was evaluated in various environmental conditions like fog, day, night, and high and low altitudes and eliminated the influence of the dynamic background. The proposed system proved that it had less detection duration and precise target detection in complex environments [23]. Moreover, Dai*et al.* trained 700 drone images on a pre-trained YOLOv5s model and used it as a sensor to calculate the in-front drone's relative position [24]. Result-level fusion-based 2D-CNN for binary classification with audio signals was adopted in [25]. They extracted the Log-Mel spectrogram and Mel frequency cepstral coefficients and achieved the highest average accuracy of 93.5% by the fusion of these two features as summarized in Table I.

Authors in [26] proposed DIAT-RadSATNet containing modules from SqueezeNet and MobileNet for multi-class classification. They summarized the effects of different dimension filters on computing cost, multiscale kernels, UAV targets, and the impact of down-sampling on classification accuracy. Moreover, Elsayed *et al.* presented a visual drone detection method that relied on videos with a uniform background. Its detection phase leverages the CNN classifier's background removal algorithm, while its tracking phase handles the missed detection tracks [27]. Global-local feature-enhanced network (GLF-Net) with a multiscale feature fusion module was proposed in [28] to extract the affective features of UAVs in complex backgrounds. They considered the model to have three main modules: a feature recombination module, a local feature extraction, and a global feature extraction. Their highest detection accuracy was 86.52% mAP on the RO-UAV dataset. Ye *et al.* proposed convolution–transformer network (CT-Net) by integrating an attention-enhanced transformer block and a feature-enhanced multi-head self-attention for low-altitude object detection. The achieved mAP score of 0.966 on small objects was superior to the baseline YOLOv5 [15]. Object detection algorithms during training, specifically for UAV detection, faced problem in hierarchical feature extraction for multilevel representations from pixel to high-level semantic features. Many hidden factors of input data were tangled through multilevel nonlinear mappings, which reduced model's expressive capability on real-time and unseen images. Some algorithms also faced problems when dealing with high-dimensional data having multi-class problem.

After reviewing the existing literature, we highlighted limitations: datasets used for training contained only one type of drone image and ignored challenging weather conditions or complex environments. The details of the number of samples in each class of dataset were missing and neglected the data augmentation techniques to balance the imbalanced classes. Moreover, they also ignored the model's training time and evaluation metrics like mAP, IoU, etc., which are necessary to decide the computational resources required to train the datasets. These limitations and challenges have motivated us to present the scheme, which can improve precision, efficiently target multiscale UAVs, and provide balanced results in challenging weather conditions.

TABLE I: Literature review

Reference + Publication Year	Dataset	Problem Statement	Achieved Results
Li *et al* 2022 [1]	Self-collected dataset using B210 SDR from National Instruments and GNURadio	Jamming detection and classification in UAVs	92.20% with 1.35% false-alarm in real noisy environment
Anwar *et al* 2019 [2]	Self-collected dataset in real noisy environment	Sound-based amateur drone detection	96.7% accuracy in real noisy environment
Zheng *et al* 2021 [4]	Self-collected dataset named as Det-fly	Air-to-air visual detection of micro-UAVs	82.4% average precision.
McCoy *et al* 2022 [6]	Publicly available RF, image and audio dataset	Multi-modal UAV classification	Precision: 91.84% F1: 92.78%
Gerwen *et al* 2022 [8]	IMU, sonar, SLAM camera based dataset	Sensor fusion based indoor drone positioning	Average 6 ultra-wideband (UWB) anchors 3D error: 10.7cm
Kılıç *et al* 2021 [11]	DroneRF dataset	Drone classification	4 class average accuracy:98.67% 10 class average accuracy:95.15%,
Alsanad *et al* 2022 [13]	Extracted 5000 drone images from online videos	Drone detection	96% average precision, 95.60% accuracy
Dogru *et al* 2022 [20]	Sparse Lidar measurements	Drone detection	88% of the environment is detected
Medaiyese *et al* 2021 [21]	Self-collected RF dataset	UAV Detection	98.9% accuracy at 10 dB SNR
Wisniewski *et al* 2022 [22]	Self-generated synthetic dataset	Drone classification	92.4% accuracy, 88.8% precision, 88.6% recall, 88.7% F1-score
Wang *et al* 2022 [23]	Online UAV dataset	Online target detection system	12.26 s/frame is achieved by YOLOv4 during pre-processing of 1280 × 720 image
Dai *et al* 2022 [24]	700 drone images	Platooning control of drones	Trained model can process images at 15 FPS
Dong *et al* 2022 [25]	Self-collected sound dataset	Drone detection	94.5% accuracy
Kumawat *et al* 2022 [26]	Radar-based self-collected dataset	Small UAV targets detection and classification	97.1% detection and 97.3% classification
Elsayed *et al* 2022 [27]	Drone vs bird challenge dataset	Visual drone detection	90.51% precision, 64.55% recall, 74.56% f1 score
Proposed Methodology 2022	Roboflow	Birds vs drone detection	96.8% precision, 90.4% recall, 95.9% f1 score, 96% mAP, 62.7% IoU for UAV detection (MFNet-M)

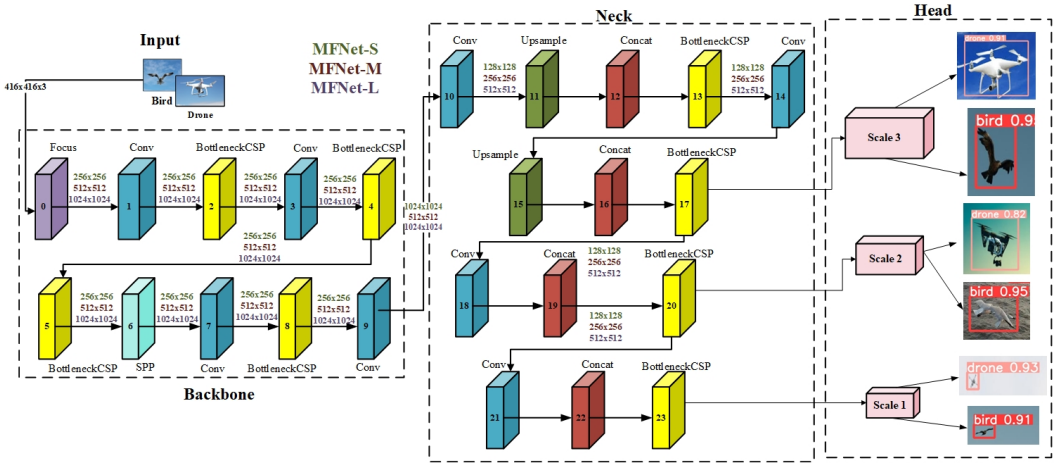


Fig. 2: Proposed **SafeSpace MFNet**. Takes input in fixed size of 416×416, extract feature maps by using backbone block and fused together in head block. Then using head layer performed final prediction and sketched bounding boxes.

TABLE II: Number of filters and extracted features of the models.

	YOLOv5s		MFNet-S		MFNet-M		MFNet-L	
Layer	**Kernel size**	**Features**	**Kernel size**	**Features**	**Kernel size**	**Features**	**Kernel size**	**Features**
0:Focus	64	3520	**256** (↑**4**×)	**14080** (↑**4.3**×)	**512** (↑**8**×)	**28160** (↑**8.6**×)	**1024** (↑**16**×)	**56320** (↑**17.3**×)
1:Conv	128	18560	**256** (↑**2**×)	**147712** (↑**7.95**×)	**512** (↑**4**×)	**590336** (↑**31.80**×)	**1024** (↑**8**×)	**2360320** (↑**127.17**×)
2:Bottleneck CSP	128	19904	128	**24000** (↑**1.20**×)	**512** (↑**4**×)	**313088** (↑**15.72**×)	128	**48576** (↑**2.44**×)
3:Conv	256	73984	256	73984	**512** (↑**2**×)	**590336** (↑**7.97**×)	**1024** (↑**4**×)	**295936** (↑**4**×)
4: Bottleneck CSP	256	161152	256	161152	**512** (↑**2**×)	**641792** (↑**3.98**×)	256	**210304** (↑**1.30**×)
5:Conv	512	295424	**256** (↓**0.5**×)	**147712** (↓**0.5**×)	512	**590336** (↑**1.99**×)	**1024** (↑**2**×)	**590848** (↑**2**×)
6:Bottleneck CSP	512	641792	512	**609024** (↓**0.94**×)	512	641792	512	**707328** (↑**1.1**×)
7:Conv	1024	1180672	**256** (↓**0.25**×)	**295168** (↓**0.25**×)	**512** (↓**0.5**×)	**590336** (↓**0.5**×)	1024	1180672
8:SPP	1024	656896	**256** (↓**0.25**×)	**41344** (↓**0.062**×)	**512** (↓**0.5**×)	**164608** (↓**0.25**×)	1024	656896
9:Bottleneck CSP	1024	1248768	1024	**1052160** (↓**0.84**×)	**512** (↓**0.5**×)	**313088** (↓**0.25**×)	1024	1248768
10:Conv	512	131584	**128** (↓**0.25**×)	**32896** (↓**0.25**×)	**256** (↓**0.5**×)	**33024** (↓**0.25**×)	512	131584
11:Upsample	0	0	0	0	0	0	0	0
12:Concat	1	0	1	0	1	0	1	0
13:Bottleneck CSP	512	378624	**128** (↓**0.25**×)	**36288** (↓**0.095**×)	**256** (↓**0.5**×)	**111488** (↓**0.29**×)	512	378624
14:Conv	256	33024	**128** (↓**0.5**×)	**4224** (↓**0.127**×)	256	**16640** (↓**0.50**×)	**512** (↑**2**×)	**66048** (↑**2**×)
15:Upsample	0	0	0	0	0	0	0	0
16:Concat	1	0	1	0	1	0	1	0
17:Bottleneck CSP	256	95104	**128** (↓**0.5**×)	**28096** (↓**0.29**×)	256	**111488** (↑**1.17**×)	**512** (↑**2**×)	**345856** (↑**3.63**×)
18:Conv	256	147712	**128** (↓**0.5**×)	**36992** (↓**0.25**×)	256	147712	**512** (↑**2**×)	**590336** (↑**3.99**×)
19:Concat	1	0	1	0	1	0	1	0
20:Bottleneck CSP	512	313088	**128** (↓**0.25**×)	**24000** (↓**0.07**×)	**256** (↓**0.5**×)	**95104** (↓**0.30**×)	512	**378624** (↑**1.20**×)
21:Conv	512	590336	**256** (↓**0.5**×)	**36992** (↓**0.062**×)	**256** (↓**0.5**×)	**147712** (↓**0.25**×)	512	590336
22:Concat	1	0	1	0	1	0	1	0
23:Bottleneck CSP	1024	1248768	**128** (↓**0.125**×)	**24000** (↓**0.019**×)	**256** (↓**0.25**×)	**95104** (↓**0.076**×)	**512** (↓**0.5**×)	**378624** (↓**0.30**×)
24:Detect	[2, [P3, P4, P5]]	18879	[2, [P3, P4, P5]]	**4095** (↓**0.21** ×)	[2, [P3, P4, P5]]	**8127** (↓**0.43** ×)	[2, [P3, P4, P5]]	**16191** (↓**0.85**×)

III. Proposed MFNet Architecture

A. MFNet blocks description

The proposed MFNet has four main blocks: input, backbone, neck, and head as shown in Fig. 2. The *input block* transfers spatial information to the channel dimension on the input images for faster inference with no mAP penalty. It also handles the data preparation by employing mosaic data augmentation (MDA) [29] and adaptive image filling (AIF) [29] methods. MDA enables the MFNet to learn object detection at a smaller scale than usual object detection algorithms, which is advantageous in training and reduces the requirement for mini-batch size. AIF adjusts picture cropping and aspect ratios according to context, allowing MFNet to adapt a wide range of visual patterns. MDA and AIF combine adaptive anchor frame computation into the input for adaptability to varied datasets, so it can automatically determine the initial anchor frame size.

The *backbone* retrieves feature maps of various sizes from the input image by using cross-stage partial network (CSP) [30] and spatial pyramid pooling (SPP) [31]. Here, we adopted *BottleneckCSP* block that reduces the computation and inference time. SPP extracts three-scale feature maps to improve detection accuracy. For the *neck block*, *feature pyramid network (FPN)* [32] is used, which extracts semantic qualities from top to lower hierarchy, whereas path aggregation network (PAN) [33] extract localization features from lower to top order. These two structures collaborate to strengthen the features acquiring capability from multiple network levels by fusion, helping in increasing detection capabilities even more. The last block named as *head* performs the final detection. MFNet is able to conduct detection at three scales, which are obtained by downsampling the dimensions of the input image by 32, 16, and 8, respectively. The first detection is made after the 17th layer feature map, the second detection is performed after the 20th layer and third and final detection is formed at

the 23rd layer feature map.

B. Anchor boxes

An anchor box is a predetermined collection of bounding boxes with a specific height and breadth. These boxes are chosen based on the object sizes in training datasets to capture the scale and aspect ratio of various object classes. The anchor boxes for MFNet are 8×8 (P3), 16×16 (P4), and 32×32 (P5) for the identification of smaller, medium, and large objects. After a detailed evaluation of the results, we notice that the larger feature maps have smaller anchor boxes because if feature maps are repeatedly down-sampled, there is a possibility of losing small-sized objects. Therefore, we adopt large feature maps and smaller anchor boxes to detect the small-sized objects in an image. When an object's center falls into a grid cell, the output neurons associated with that cell are responsible for creating the object's (x, y), width, height, objectness score, and class. It helps to distinguish different-sized elements in the same image at different scales.

C. Loss functions and target prediction

To optimize MFnet, we utilize adaptive moment estimation (Adam) optimizer [34] and loss functions introduced by YOLOv5 [15] named as *class loss* (L_{cls}), *objectness loss* (L_{obj}), and *localization loss* (L_{loc}). Adam computes the decaying averages of past mt and the past squared gradients of vt.

$$m_1 = \gamma_1 \times m_{1(t-1)} + (1 - \gamma_1) \times \nabla_t, \tag{1}$$

$$m_2 = \gamma_2 \times m_{2(t-2)} + (1 - \gamma_2) \times \nabla_t^2, \tag{2}$$

where m_1 is the first moment (the mean), m_2 is the second moment (the uncentered variance) of the gradients, γ_1 is the exponential decay rate for the m_1, γ_2 is the exponential decay rate for m_2, and ∇_t is the gradient of the current mini-batch. These estimates update the parameter θ using the following equation as

$$\theta_t = \theta_{(t-1)} - \frac{\eta}{\sqrt{\hat{m}_2 + \in}} + \hat{m_1}. \tag{3}$$

η represents the learning rate, and the feature map of all MFNet models is represented by $g(f(I_x, y, t))$ having ZxZ grids in each cell, have B bounding boxes where prediction losses are applied. The L_{obj} can only consider during the presence of an object in a particular cell C.

$$L_{obj} = \sum_{l=0}^{Z^2} \sum_{m=0}^{B} 1_{obj}^{lm} (C_l - \hat{C_l}) + \lambda_n \sum_{l=0}^{Z^2} \sum_{m=0}^{B} 1_{obj}^{lm} (C_l - \hat{C_l}), \tag{4}$$

where 1_{obj}^{lm} has a value of 1 if the m-th bounding box in cell l contains the object. λ_n is loss coefficient. L_{cls} is computed using a square of the error between predicted conditional class probability $P_l(c)$ and ground-truth $P_l(c)$ for cell l.

$$L_{cls} = \sum_{l=0}^{Z^2} 1_{obj}^{lm} \sum_{c \in C} (P_l(c) - \hat{P}_l(c)), \tag{5}$$

Here, L_{loc} returns the difference between the predicted and true bounding boxes, and is computed as

$$\begin{aligned} L_{loc} = \sum_{l=0}^{Z^2} \sum_{m=0}^{B} 1_{obj}^{lm} \left[(x_l - \hat{x}_l)^2 + (y_l - \hat{y}_l)^2 \right] + \\ \lambda_{cd} \sum_{l=0}^{Z^2} \sum_{m=0}^{B} 1_{obj}^{lm} \left[(\sqrt{w_l} - \sqrt{\hat{w}_l})^2 + (\sqrt{h_l} - \sqrt{\hat{h}_l})^2 \right]. \end{aligned} \tag{6}$$

$(\hat{x}_l, \hat{y}_l)$ and (x_l, y_l) represent the top-left corner coordinates of the predicted and ground truth bounding box, respectively. Whereas $(\hat{w}_l, \hat{h}_l)$ and (w_l, h_l) are the width and height of predicted and ground truth bounding box, respectively. Therefore, the total loss for the MFNet model is

$$L_t = \lambda_1 \times L_{cls} + \lambda_2 \times L_{obj} + \lambda_3 \times L_{loc}. \tag{7}$$

These are the MFNet losses: L_{obj} is the confidence of object existence calculated using binary cross-entropy loss. Similarly, L_{loc} is the bounding box regression loss calculated using mean squared error, and L_{cls} is the classifying loss calculated using cross-entropy. Here, λ_1, λ_2 and λ_3 are loss coefficients.

MFNet calculates the target coordinates and target frame size of the bounding box with a specific grid size. The network predicts 4 coordinates for each bounding box, t_x, t_y, t_w, t_h. If the cell is offset from the top left corner of the image by (cx, cy) and the bounding box prior has width and height pw, ph and σ is the sigmoid activation function then the predictions correspond to:

$$b_x = (2 \cdot \sigma(t_x) - 0.5) + c_x \tag{8}$$

$$b_y = (2 \cdot \sigma(t_y) - 0.5) + c_y \tag{9}$$

$$b_w = p_w \cdot (\sigma(t_w))^2 \tag{10}$$

$$b_h = p_h \cdot (\sigma(t_h))^2 \tag{11}$$

D. MFNet working procedure

MFNet processes the entire image using a single neural network, divides it into grids and predicts bounding boxes with probabilities for each grid cell. The predicted probability weights the bounding boxes as it provides predictions after only one forward propagation pass through the neural network. Lastly, the max suppression algorithm ensures that the MFNet identifies each object once. The network topology in Fig. 2 is the same for all MFNet models; whereas the main difference is the change in the kernel sizes of the backbone and head, which in turn varies the number of extracted features, training parameters, gradients, and GFLOPs. We propose MFNet to compute the optimum size of feature maps that provides the best and fast detection results for flying birds and UAVs in challenging weather conditions. The anchor sizes, model depth multiple (i.e., 0.33), and layer channel multiple (i.e., 0.50) are the same for all MFNet models. For comparison, the applied kernel sizes and the extracted features of each layer of the proposed MFNet-S, M, L, and baseline YOLOv5s summarized in Table II with differences are highlighted.

TABLE III: Hyper-parameters qualitative analysis

Input size	Epochs	Batch size	Layers	Learning rate	Momentum	Weight Decay
416×416	120	32	232	0.01	0.937	0.0005

IV. Dataset and Implementation

UAVs are challenging to detect because of their closeness to birds in terms of radar cross-section (RCS), moderate velocities, and low flying altitudes. Generally, birds are misidentified as UAV targets by the drone surveillance system resulting in an unnecessarily high incidence of false reports and lowering the efficacy of the surveillance technique. To overcome this, we consider the birds vs. UAV detection problem to improve the system's precision and mAP compared to the existing schemes.

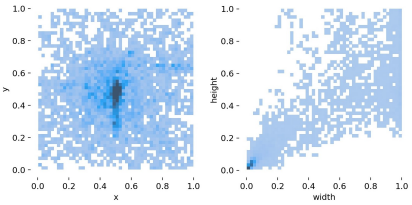
To train the proposed MFNet architecture, we gathered around **5105 images** of UAVs and birds from the publically available open-source datasets provided on the Roboflow [35]. It includes various types of UAVs like multi-rotor (tri, quad, Hexa, and octa-copter), single rotor, fixed wings, and various types of birds. The images also contain different-sized (small, medium, and large) birds and UAVs. We target the detection of birds and UAVs problem, but we want our models to be sensitive to multi-size and multi-types of birds and UAVs. The final output image only gives information about the presence of a bird and drone with a bounding box and confidence level, and not any information about its type or size.

Among 5105 images, we consider 2605 images of UAVs and 2500 images of birds. Moreover, we further divide the (2605, 2500) images of UAVs and birds into (850, 833) small-sized, (855, 833) medium-sized, and (900, 834) large-sized drone images, respectively. To fulfill the muti-type UAV and birds criteria, these small-, medium-, and large-sized images contain different drone models and different types of birds (white stork, crane, rüppell's vulture, eagle, bar-tailed godwit, and common blackbird). Moreover, 5105 images contain eight backgrounds: clear sky, cloudy, sunny, fog, rainy, water, mountains, and forest. We have approximately 325 UAV images per background (e.g., cloudy background) and 312 birds images per background.

For dataset pre-processing, we resize the images to a dimension of 416 × 416 pixels. Contrast enhancement is applied to get a fixed range of intensity values to reduce the intensity spread, training time, and exponential increase in computational resources. All the datasets are pre-annotated and imported in the *Roboflow- YOLO Darknet TXT format* for training the YOLOv5s and the proposed MFNet models. We split the total 5105 pre-processed images into 4340 training images **(85%)**, 510 validation images **(10%)**, and 255 test images **(5%)**. Fig. 3(a) shows the ground truth images with respective bounding boxes and classes.

We plot the distribution of the drone sizes and positions in the dataset in Fig. 3(b). The diversity of drone size is a substantial problem for YOLOv5 models to identify and classify tiny drone objects against an ambient background. Therefore, the input images of the dataset used for training contained images with multiple region proposals of various sizes for drone objects. All experiments are separately run on

(a) Ground-truth images of the annotated dataset



(b) Dataset Distribution

Fig. 3: Graphical representation of the merged dataset.

the *Google Colab* environment with an NVIDIA Tesla T4 GPU and 12GB RAM.

The hyper-parameters for all models are set at the same values to have a fair comparison with the state-of-the-art as given in Table IV. The learning rate specifies the model alteration rate with the predicted error each time the model weights are adjusted. Therefore, it is difficult to calculate the optimal weights since a small number may lead to a prolonged training time, and high values may result in an inconsistent training process. Hence, we set an initial learning rate (lr0) of 0.01 for SGD and the Adam optimizer. The momentum is adjusted at 0.937 to accelerate learning in low-curvature directions while remaining steady in high-curvature directions. We used a weight decay of 0.0005 to minimize overfitting by penalizing heavy weights. This selection increases optimizer convergence by; promoting lower weights, training efficiency and also minimizes the time it takes to converge on a response. We set the warm-up epoch to 3.0 with an initial warm-up momentum and initial bias of 0.8 and 0.1, respectively. These warm-up parameters lessen the predominant impact of the early training instances on the optimizer. It enables the computation of the exact gradients from the start, whereas more epochs are required without it to obtain optimal convergence. classes.

V. Performance Evaluation

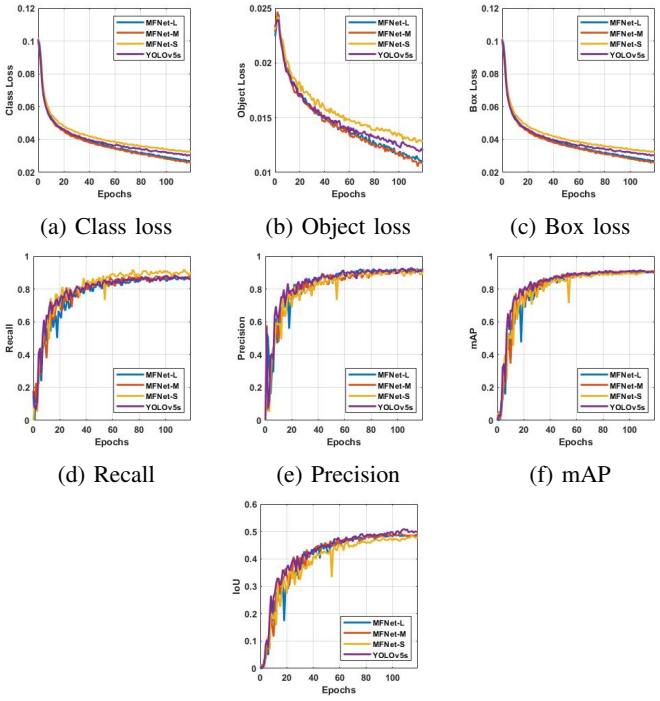
We evaluate the proposed MFNet architecture detection performance by using the evaluation metrics; precision (P), recall (R), mean accuracy precision (mAP), and intersection over Union (IoU), respectively, and mathematically represented as;

$$P = \frac{TP}{TP + FP} \tag{12}$$

$$R = \frac{TP}{TP + FN} \tag{13}$$

$$mAP = \frac{1}{N} \sum_{c=1}^{N} \left(\frac{TP}{TP + FP} \right) \tag{14}$$

$$IoU = \frac{\text{GroundTruth} \cap \text{DetectedBox}}{\text{GroundTruth} \cup \text{DetectedBox}} \tag{15}$$



(a) Class loss (b) Object loss (c) Box loss

(d) Recall (e) Precision (f) mAP

(g) IoU

Fig. 4: MFNet and YOLOv5s performance comparison under various metrics.

These metrics computation requires four attributes; true positive (TP), true negative (TN), false positive (FP), and false negative (FN). We apply the classification loss shown in Fig. 4(a) to train the classifier head and to find the targeted object type. Here, YOLOv5s shows an increased classification loss compared to MFNet-M/L, which proves its high detection accuracy in unknown scenarios. Moreover, we compare the **objectness loss** for YOLOv5s and MFNet in Fig. 4(b) during the training session, which shows MFNet-M/L have high precision values as of achieving low objectness loss compared to YOLOv5s. Similarly, **box loss** of YOLOv5s has higher values than MFNet-S/M/L during the training session shown in Fig. 4(c), which proves MFNet-S/M/L excellent capability to locate an object's center point with a predicted bounding box. Moreover, we plot the recall, precision, mAP, and IoU results shown in Fig. 4(d)-(f) over 120 epochs for both models that show an increasing trend which proves the correctness of the trained model.

In Table IV, the proposed MFNet-M achieves 92.3% average P, 88.4 % R, 91.5% mAP, and 51.1 % average IoU compare to YOLOv5s model. We achieve significant gains for other MFNet models as well. UAV has the highest precision of 96.8% for MFNet-M models, while birds achieved the highest precision of 87.7% with MFNet-M. That proves that MFNet-M is most sensitive to multi-sized flying birds and other targeted objects in complex background conditions.

TABLE IV: Evaluation metrics of trained models.

Class	Precision (%)	Recall (%)	mAP (%)	IoU (%)
MFNet-S				
Bird	86.7(↑ 0.9)	83.1(↑ 0.6)	86.3(↑ 0.8)	37.4(↓ 0.1)
UAV	95.2(↓ 0.8)	90(same)	95.4(↓ 0.2)	60.8 (↓ 0.5)
Average	91 (↑ 0.5)	86.6 (↑ 0.4)	90.8 (↑ 0.2)	49.1 (↓ 0.3)
MFNet-M				
Bird	**87.7** (↑ 1.6)	**86.4**(↑ 3.9)	**87.1**(↑ 1.6)	**39.4**(↑ 1.9)
UAV	**96.8**(↑ 0.8)	**90.4**(↑ 0.4)	**95.9**(↑ 0.3)	**62.7** (↑ 1.4)
Average	**92.3** (↑ 1.8)	**88.4**(↑ 2.2)	**91.5** (↑ 0.9)	**51.1** (↑ 1.7)
MFNet-L				
Bird	87.4(↑ 2.3)	82.1(↓ 0.4)	86.1(↑ 0.6)	36.8 (↓ 0.7)
UAV	95.8(↓ 0.2)	88.2(↓ 7.2)	95.6(↓ 0.4)	61.1(↓ 0.2)
Average	91.6 (↑ 1.1)	85.2(↓ 1)	90.9(↑ 0.3)	49 (↓ 0.4)
YOLOv5s				
Bird	85.1	82.5	85.5	37.5
UAV	96	90	95.6	61.3
Average	90.5	86.2	90.6	49.4

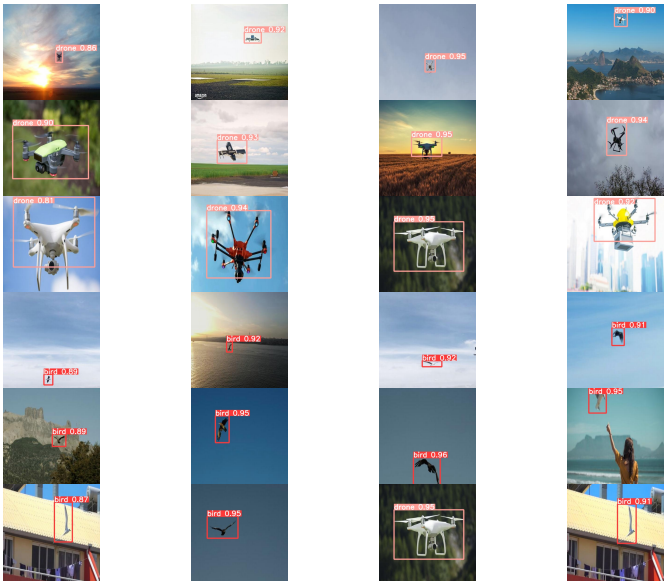
A. Impact of varying network attributes on extracted features and model detection performance

Table II gives detailed insight into the effect of changing input network attributes on model outputs. We select the kernel sizes in MFNet-S backbone and neck to 128 and 256, respectively, which extracts 61.52% fewer features (i.e., 2,789,023) compared to baseline YOLOv5s 7,249,215 features. Similarly, when MFNet-M Kernel sizes of backbone and neck changed to 512 and 256, respectively, it reduced the features by 27.92% (i.e., 5,223,231) compared to YOLOV5s. However, the opposite trend is noticed for MFNet-L when Kernel sizes in backbone and neck changed to 1024 and 512, respectively, which increased the 41% features (i.e., 10,222,111) of MFNet-L compared to YOLOv5s, which increased its resource consumption.

One of the key finding of this research is that we extract the best suitable feature map for differentiating drones from birds. Results prove that MFNet-M achieved the best results for the conv and BottleckCSP layers with 512×512 and 256×256 filters, respectively. These filters covered the complete attributes of small-sized targets and half the features of large-sized objects. Although MFNet-S/M/L is trained on one specific-sized feature map, they all performed well on multi-sized drones and bird images upon testing. Results displayed in Fig. 5 - Fig. 8 show that MFNet-M is the best-performing model as it detected small, medium, and large-sized drones with 95% accuracy. Moreover, MFNet-M achieves a detection accuracy of 92%, 96%, and 95% for small, medium, and large-sized birds, respectively, which shows that MFNet-M performance is the most consistent among all models. We conclude that even training images on one particular-sized feature map can detect birds and drones effectively upon testing.

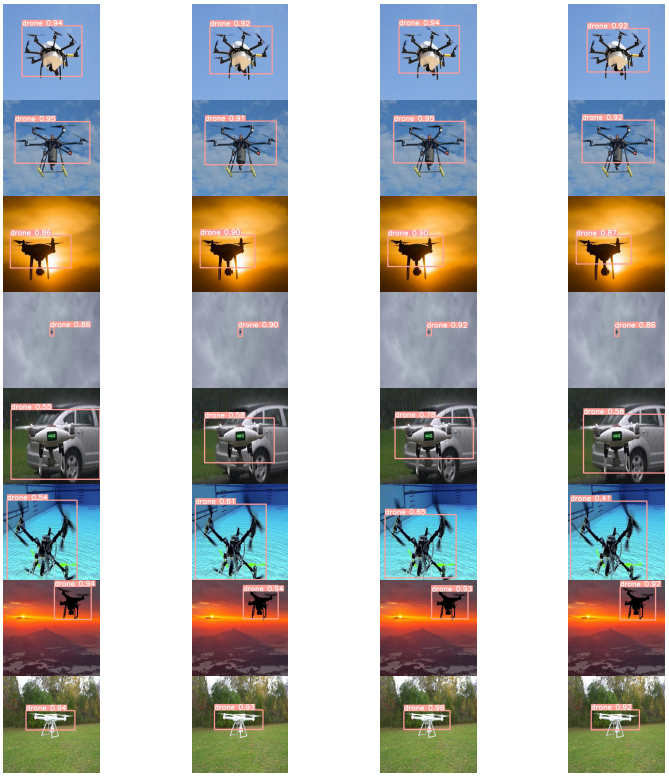
B. Image attributes affecting detection

We evaluate key aspects of the images such as environmental background, target scales, and other challenging conditions on the detection performance. The performance of MFNet is affected by many aspects such as insufficient training and different parameters, we take the detection accuracy of these algorithms as the criteria for a fair evaluation.



(a) YOLOv5s (b) MFNet-S (c) MFNetM (d) MFNet-L

Fig. 5: Detection results with respect to target scale (Top to bottom) for small-sized, medium-sized, and large-sized drones and birds.



(a) YOLOv5s (b) MFNet-S (c) MFNetM (d) MFNet-L

Fig. 6: UAV detection performance for proposed MFNet and baseline under challenging environmental conditions (Top to bottom) like clear sky, cloudy conditions, sunlight, foggy weather, Rainy situation, waterfall, mountains, and forest.

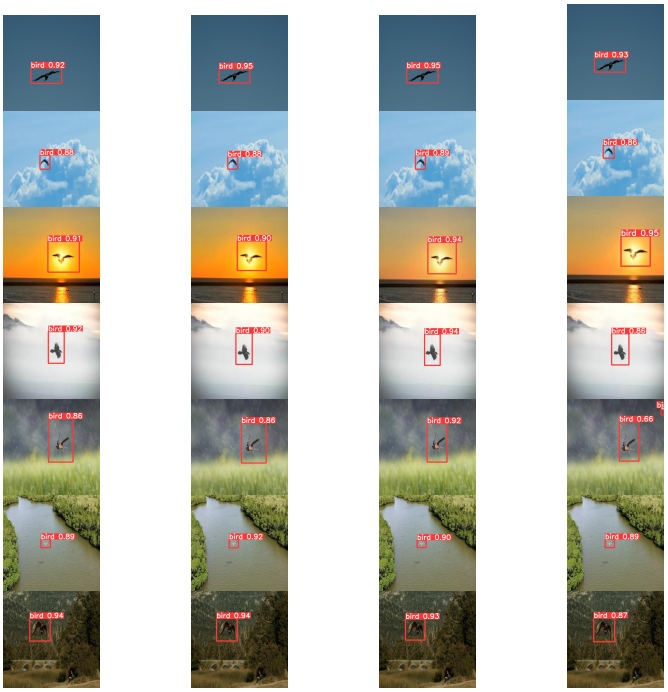
1) Target scales: The image size of the target (small, medium or large) greatly influences the model's detection performance. We plot the detection results of UAVs and birds as the target on all the selected models in Fig. 5 and Fig. 6. Results show that MFNet-M has the highest detection accuracy (95%) for all sized UAV targets. MFNet-S and MFNet-M scored the highest detection accuracy of 92% for small-sized bird targets. MFNet-M had the highest detection accuracy of 96% on medium-sized bird targets, while MFNet-S and MFNet-M scored equally (95%) on large-sized bird targets. Thus, MFNet-M proved an excellent choice for detecting birds and drone targets of all sizes in challenging conditions.

2) Impact of varying environment backgrounds on detection: In Fig. 6 and Fig. 7, we compare the detection performance of the proposed MFNet in different conditions with challenging backgrounds for UAVs and birds, respectively. YOLOv5s and MFNet-M achieve the highest detection accuracy of 94% on UAV images in clear skies and cloudy weather conditions. Similarly, MFNet-S and MFNet-M scored the highest detection accuracy of 90% for UAV images with sunny conditions. MFNet-M achieves a detection accuracy of 92% for UAVs with foggy backgrounds, 78% for UAVs with a rainy situation, 65%for UAVs flying with water backgrounds, and 96% for green forest backgrounds. YOLOv5s and MFNet-M achieved 94% for UAVs flying in hilly areas.

For bird detection, MFNet-S and MFNet-M performed best by achieving 95% accuracy with clear sky, whereas YOLOv5s and MFNet-M have around 86% detection accuracy in rainy and hilly areas and 94% in forests. Moreover, MFNet-M performed best with a cloudy sky, sunny, and foggy conditions by achieving an accuracy of 89%, 94%, and 94%, respectively. We conclude that the rainy condition and the cloudy plus rainy conditions are the most challenging for UAV and bird detection, respectively.

3) Multiple targets detection in a single scene with challenging conditions: Multiple UAVs (swarms) are attracting the attention of domestic, commercial, and military users because they can improve performance. However, such deployment requires robust collision avoidance and detection technologies in overcrowded airspace, which proves the necessity of multi-target detection with high precision in challenging conditions. The performance of the proposed MFNet is tested for multi-targets in the scene with a challenging environment, as shown in Fig. 8, where we achieve more accuracy for multi-birds and multi-UAVs detection using MFNet-S.

We saved all the trained models' weight files as .pt extension. To make sure that our trained models detect drones and birds and avoid misclassification with other similar objects (like a kite). We tested the trained networks for the scenario when neither drones nor birds were in the images. We can input any image and videos for testing the trained model for verification. Therefore, we gave images containing kites in eight different backgrounds to the models. As we set the proposed model confidence level (threshold) to 40% (i.e., if the proposed model detects with this much confidence), it only gives the output. Otherwise, it generates no bounding box by avoiding the misclassification of kites as birds or drones.



(a) YOLOv5s (b) MFNet-S (c) MFNetM (d) MFNet-L

Fig. 7: Bird detection performance for proposed MFNet and baseline under challenging environmental conditions (Top to bottom) like clear sky, cloudy conditions, sunlight, foggy weather, Rainy situation, waterfall, mountains, and forest.

However, YOLOv5s identified a kite as a bird in a cloudy background.

C. Computational complexity and inference time for UAV detection

Inference rate plays an essential role in real-time system deployment and improving the UAV detection speed. Commercial and racing UAVs can fly around 50–70 mph and 150 mph, respectively. Therefore, even a one-second delay translates to a flying distance of 22m to 66m, which raises a serious security threat. The computational interdependence qualities of features can improve the detection time, which helped us to achieve 0.8-0.9 ms detection time. MFNet-S has the lowest inference time of 8.7 msec with 114 FPS on Tesla T4 GPU shown in Table V, which proves that model is the best for real-time UAV fast detection.

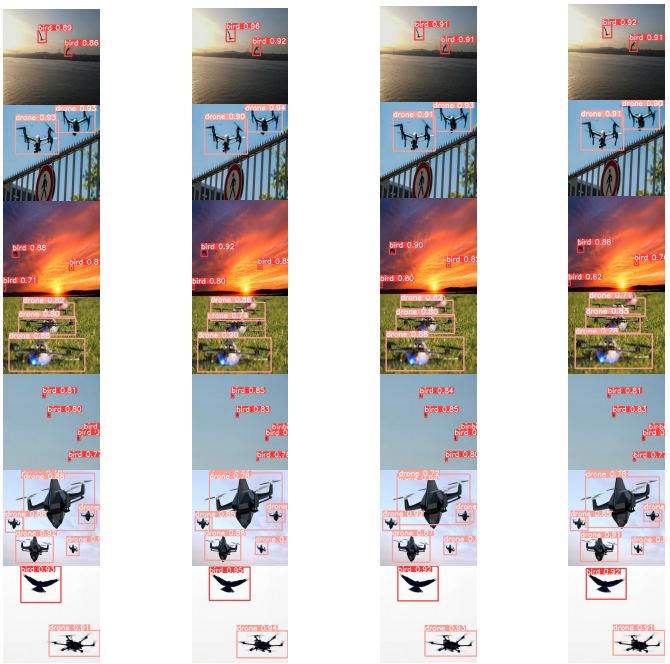
The system's computational complexity depends on training time, extracted features, trained model size, and GFLOPs. The training time indicates the amount of time required for training data to pass through forward and backpropagation of the model during the training phase. MFNet-S has a minimum training time of around 2.182 hours and a model size of 5.9 MB. MFNet-L has the GFLOPs of 157.6G, a training time of 4.072 hours, 10.8 million parameters, and a trained model size of 20.8 MB. However, YOLOv5s took 2.226 hours, 7.2 million parameters, a model size of 14.8 MB, and 16.7 GFLOPs as shown in Fig. 9. Thus, the computational complexity of the MFNet-L due to the highest GFLOPs and extraction of increased feature maps needs more memory requirement and GPU capacity.

TABLE V: Proposed models performance on unseen test dataset.

Model	Pre-process (msec)	Inference (msec)	NMS/image (msec)	FPS
MFNet-S	0.3	**8.7** (↓ 1.5)	**0.8** (↓ 0.1)	**114.94** (↑17.86)
MFNet-M	0.3	9.9 (↓ 0.6)	**0.8** (↓ 0.1)	101.01 (↑ 3.93)
MFNet-L	0.3	17.7 (↓ 0.8)	**0.8** (↓ 0.1)	56.49 (↓ 40.59)
YOLOv5s	0.3	10.3	**0.9**	97.08

D. Tradeoff in IoU and precision performance

The results prove that the average precision and mAP of all MFNet models have significantly increased compared to YOLOv5s. We achieved the average precision, recall, mAP, and average IoU of 92.3%, 88.4%, 91.5%, and 51.1% by MFNet-M, respectively. Moreover, we notice the increase in average precision by 1.8%, average recall by 2.2%, mAP by 0.9%, and IoU by 1.9% compared to YOLOv5s as summarized in Table IV. MFNet-M can accurately predict the TP cases for UAV detection as it achieves the highest precision, recall, mAP, and IoU. However, no improvement is noticed in the recall by using MFNet-S for UAV detection. Similarly, the



(a) YOLOv5s (b) MFNet-S (c) MFNetM (d) MFNet-L

Fig. 8: Multiple-targets (birds or drones) detection performance in a single scene with challenging environmental conditions.

TABLE VI: Comparison with the state-of-the-art schemes for UAV detection.

Model	Dataset	Input size	Precision (%)	Recall (%)	mAP @0.5 (%)	FPS	Parameters (million)	GFLOPS	CPU/GPU
CT-Net-Middle [15]	26062 images	640×640	N/G	N/G	95.1	78	14.67	41.7	NVIDIA GeForce GTX1080TI
YOLOv5s [15]	26062 images	640×640	N/G	N/G	91.7	156	7.20	17	NVIDIA GeForce GTX1080TI
Improved YOLOv5s [16]	1259 images	640×640	93.54	91.09	94.82	N/G	N/G	9.19	NVIDIA GeForce RTX 3050
Fine-tuned YOLOv5x [14]	1359 images	416×416	94.7	92.05	94.1	N/G	N/G	N/G	NVIDIA RTX2070
YOLOv5s [17]	3600 images	512×512	77	79	N/G	5.69	N/G	N/G	Tesla K80
SAG-YOLOv5s [18]	11,286 images	416×416	97.3	95.5	97.6	13.2	8.8	3.3	NVIDIA GeForce RTX 2070 SUPER
TransVisDrone [19]	2500 images	1280×1280	92	91	95	24.6	N/G	N/G	NVIDIA Jetson Xavier NX
Proposed MFNet-S	5105 images	416×416	95.2	90	95.4	114.95	2.7	18.9	NVIDIA Tesla T4
Proposed MFNet-M	5105 images	416×416	96.8	90.4	95.9	101.01	5.2	75.3	NVIDIA Tesla T4
Proposed MFNet-L	5105 images	416×416	95.8	88.2	95.6	56.49	10.2	157.6	NVIDIA Tesla T4

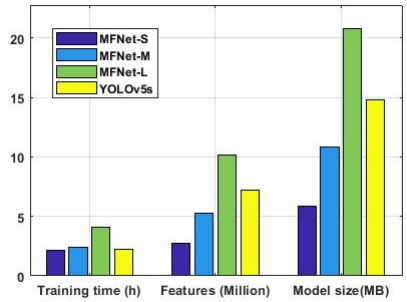


Fig. 9: Computational complexity of the trained models.

average recall of the MFNet-L model decreased by 1%, while the average IoU for MFNet-S and MFNet-L decreased by 0.3% and 0.4%, respectively. Therefore, there is always a tradeoff between average IoU and precision.

E. Comparison with the state-of-the-art schemes

We compare our results with the latest available literature that adopted the YOLOv5s model for UAV detection in order to have a fair comparison. In [16] Liu *et al.*, performed multi-rotor drone detection by training improved YOLOv5s on 1259 drone images with 16 batch sizes, 150 epochs on a workstation equipped with an AMD Ryzen 9 5900HS and 16 GB RAM. The authors have improved the performance of YOLOv5s by replacing the model backbone with Efficientlit and the head with adaptive spatial feature fusion to achieve 93.54% precision, 91.09% recall, and 94.82% mAP with 9.19 million features for UAV detection. However, the proposed MFNet-M gets 3.26% and 1.08% higher precision and mAP, respectively, for UAV detection with 3.99 million less trained features. In [14], the transfer learning method combined with YOLOv5s for unauthorized UAV detection. The model was trained on 1359 drone images and achieved 94.7% precision, 92.5% recall, and 94.1% mAP for UAV detection after training the dataset on 100 epochs. The proposed MFNet-M achieves 2.1% higher precision and 1.8% mAP greater than [14]. Similarly, Hai *et al.* in [17] trained 3600 drone images on YOLOv5s, with no changes in the model on NVIDIA Tesla K80 by using Google Colab with 32 batch size, 300 epochs, and 512×512 input image size. It took a total training time of around one day and 3 hours, and they achieved UAV detection accuracy of 92% at a drone size of 95×65 with 17.6 ms of inference time. In comparison to [17], MFNet-S and MFNet-M took around 8.9 msec and 7.7msec less than [17] YOLOv5s inference time. The training time taken to train 3600 images was 24.818 hours greater than the proposed MFNet-S trained on 5105 images, 24.582 hours greater than MFNet-M, and 22.928 hours greater than the MFNet-L model. We have summarized this performance comparison in Table VI with existing state of the art models available in the latest literature for UAV detection.

VI. Conclusion

In this paper, we proposed the novel SafeSpace MultifeatureNet (MFNet) architecture that significantly improved the precision and mAP in UAV detection compared to YOLOv5s. To successfully implement the proposed architecture and test its validity in challenging weather conditions, we gathered the existing five datasets of birds and UAVs from the literature to verify its performance on three MFNet variants. All algorithms' detection performance was rigorously examined and analyzed with varying environmental backgrounds (i.e., weather conditions) and target scales. Proposed MFNet-small, MFNet-medium, and MFNet-large successfully detected and

identified UAVs in 0.8msec with the highest UAV detection precision of 95.2%, 96.8%, and 95.8%, respectively, compared to YOLOv5s and the existing state-of-the-art schemes. For the time being MFNet can only identify birds and drones, to further improve it we to train all the models for multi-class classification and identification.

REFERENCES

[1] Y. Li, J. Pawlak, J. Price, K. Al Shamaileh, Q. Niyaz, S. Paheding, and V. Devabhaktuni, "Jamming Detection and Classification in OFDM-Based UAVs via Feature- and Spectrogram-Tailored Machine Learning," *IEEE Access*, vol. 10, pp. 16859–16870, 2022.

[2] M. Z. Anwar, Z. Kaleem, and A. Jamalipour, "Machine Learning Inspired Sound-Based Amateur Drone Detection for Public Safety Applications," *IEEE Transactions on Vehicular Technology*, vol. 68, pp. 2526–2534, Mar. 2019.

[3] Z. Kaleem and M. H. Rehmani, "Amateur drone monitoring: State-of-the-art architectures, key enabling technologies, and future research directions," *IEEE Wireless Communications*, vol. 25, pp. 150–159, Apr. 2018.

[4] Y. Zheng, Z. Chen, D. Lv, Z. Li, Z. Lan, and S. Zhao, "Air-to-air visual detection of micro-UAVs: An experimental evaluation of deep learning," *IEEE Robotics and Automation Letters*, vol. 6, pp. 1020–1027, Apr. 2021.

[5] G. Lykou, D. Moustakas, and D. Gritzalis, "Defending Airports from UAS: A Survey on Cyber-Attacks and Counter-Drone Sensing Technologies," *Sensors*, vol. 20, pp. 1–40, June 2020.

[6] J. McCoy, A. Rawal, D. B. Rawat, and B. M. Sadler, "Ensemble Deep Learning for Sustainable Multimodal UAV Classification," *IEEE Transactions on Intelligent Transportation Systems*, 2022.

[7] Y. Sun, B. Cao, P. Zhu, and Q. Hu, "Drone-based RGB-Infrared Cross-Modality Vehicle Detection via Uncertainty-Aware Learning," *IEEE Transactions on Circuits and Systems for Video Technology*, 2022.

[8] J. V. V. Gerwen, K. Geebelen, J. Wan, W. Joseph, J. Hoebeke, and E. De Poorter, "Indoor Drone Positioning: Accuracy and Cost Trade-Off for Sensor Fusion," *IEEE Transactions on Vehicular Technology*, vol. 71, pp. 961–974, Jan. 2022.

[9] H. Fu, S. Abeywickrama, L. Zhang, and C. Yuen, "Low-Complexity Portable Passive Drone Surveillance via SDR-Based Signal Processing," *IEEE Communications Magazine*, vol. 56, pp. 112–118, Apr. 2018.

[10] Y. Sun, S. Abeywickrama, L. Jayasinghe, C. Yuen, J. Chen, and M. Zhang, "Micro-Doppler Signature-Based Detection, Classification, and Localization of Small UAV with Long Short-Term Memory Neural Network," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, pp. 6285–6300, Aug. 2021.

[11] R. Kılıç, N. Kumbasar, E. A. Oral, and I. Y. Ozbek, "Drone classification using RF signal based spectral features," *Engineering Science and Technology, an International Journal*, vol. 28, p. 101028, Apr. 2022.

[12] J. Xie, J. Yu, J. Wu, Z. Shi, and J. Chen, "Adaptive switching spatial-temporal fusion detection for remote flying drones," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 7, pp. 6964–6976, 2020.

[13] H. R. Alsanad, A. Z. Sadik, O. N. Ucan, M. Ilyas, and O. Bayat, "YOLO-V3 based real-time drone detection algorithm," *Multimedia Tools and Applications*, vol. 81, pp. 26185–26198, July 2022.

[14] N. Al-Qubaydhi, A. Alenezi, T. Alanazi, A. Senyor, N. Alanezi, B. Alotaibi, M. Alotaibi, A. Razaque, A. A. Abdelhamid, and A. Alotaibi, "Detection of Unauthorized Unmanned Aerial Vehicles Using YOLOv5 and Transfer Learning," *Electronics*, vol. 11, p. 2669, Aug. 2022.

[15] T. Ye, J. Zhang, Y. Li, X. Zhang, Z. Zhao, and Z. Li, "CT-Net: An Efficient Network for Low-Altitude Object Detection Based on Convolution and Transformer," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, 2022.

[16] B. Liu and H. Luo, "An Improved YOLOv5 for Multi-Rotor UAV Detection," *Electronics*, vol. 11, p. 2330, Jul. 2022.

[17] V. T. Hai, V. N. Le, D. Q. Khanh, N. P. Nam, and D. V. Sang, "Multi-size drone detection using YOLOv5 network," *Journal of Military Science and Technology*, pp. 142–148, June 2022.

[18] D. Zarpalas, A. Dimou, A. Schumann, L. Sommer, A. Fascista, Y. Lv, Z. Ai, M. Chen, X. Gong, Y. Wang, and Z. Lu, "High-Resolution Drone Detection Based on Background Difference and SAG-YOLOv5s," *Sensors*, vol. 22, p. 5825, Aug. 2022.

[19] T. Sangam, I. R. Dave, W. Sultani, and M. Shah, "TransVisDrone: Spatio-Temporal Transformer for Vision-based Drone-to-Drone Detection in Aerial Videos," *arXiv preprint arXiv:2210.08423*, 2022.

[20] S. Dogru and L. Marques, "Drone Detection Using Sparse LIDAR Measurements," *IEEE Robotics and Automation Letters*, vol. 7, pp. 3062–3069, Apr. 2022.

[21] O. O. Medaiyese, M. Ezuma, A. P. Lauf, and I. Guvenc, "Wavelet transform analytics for RF-based UAV detection and identification system using machine learning," *Pervasive and Mobile Computing*, vol. 82, p. 101569, June 2022.

[22] M. Wisniewski, Z. A. Rana, and I. Petrunin, "Drone Model Classification Using Convolutional Neural Network Trained on Synthetic Data," *Journal of Imaging*, vol. 8, p. 218, Aug. 2022.

[23] Q. Wang, J. Gu, H. Huang, Y. Zhao, and M. Guizani, "A Resource-Efficient Online Target Detection System With Autonomous Drone-Assisted IoT," *IEEE Internet of Things Journal*, vol. 9, pp. 13755–13766, Aug. 2022.

[24] X. Dai and M. Nagahara, "Platooning control of drones with real-time deep learning object detection," *Advanced Robotics*, 2022.

[25] Q. Dong, Y. Liu, and X. Liu, "Drone sound detection system based on feature result-level fusion using deep learning," *Multimedia Tools and Applications*, pp. 1–23, June 2022.

[26] H. C. Kumawat, M. Chakraborty, and A. Arockia Bazil Raj, "DIAT-RadSATNet-A Novel Lightweight DCNN Architecture for Micro-Doppler-Based Small Unmanned Aerial Vehicle (SUAV) Targets' Detection and Classification," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, 2022.

[27] M. Elsayed, A. S. Mashaly, M. Reda, and A. S. Amein, "Visual Drone Detection In Static Complex Environment," *13th International Conference on Electrical Engineering (ICEENG)*, pp. 154–158, 2022.

[28] T. Ye, W. Qin, Y. Li, S. Wang, J. Zhang, and Z. Zhao, "Dense and Small Object Detection in UAV-Vision Based on a Global-Local Feature Enhanced Network," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, 2022.

[29] P. Wang, J. Xiao, K. Kawaguchi, and L. Wang, "Automatic Ceiling Damage Detection in Large-Span Structures Based on Computer Vision and Deep Learning," *Sustainability*, vol. 14, Mar. 2022.

[30] C. Y. Wang, H. Y. Mark Liao, Y. H. Wu, P. Y. Chen, J. W. Hsieh, and I. H. Yeh, "CSPNet: A New Backbone that can Enhance Learning Capability of CNN," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, vol. 2020, pp. 1571–1580, Nov. 2019.

[31] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8691 LNCS, pp. 346–361, June 2014.

[32] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature Pyramid Networks for Object Detection," Dec. 2016.

[33] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path Aggregation Network for Instance Segmentation," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 8759–8768, Mar. 2018.

[34] I. S. Isa, M. S. A. Rosli, U. K. Yusof, M. I. F. Maruzuki, and S. N. Sulaiman, "Optimizing the Hyperparameter Tuning of YOLOv5 for Underwater Detection," *IEEE Access*, vol. 10, pp. 52818–52831, 2022.

[35] Roboflow, "Roboflow: Give your software the power to see objects in images and video," *Roboflow Universe*, 2021. visited on 2022-11-02.

Mahnoor Dil received the BS and MS degree in Electrical Engineering with specialization in Computer from COMSATS University Islamabad, Wah Campus Pakistan. She is currently working as a Research Assisant at COMSATS University Islamabad, Wah Campus under HEC funded project to develop UAV detection system using computer vision models.

Misha Urooj Khan is working as a research assistant at the COMSATS University Islamabad, Wah campus. She received her master's degree in electronics engineering with a specialization in electronic system design from UET Taxila. She is the Chairperson of the Community of Research and Development (CRD) and got the first position in the open house and job fair in 2019. She has 25 publications in topics related to bio-medical engineering, machine learning, deep learning, audio processing, and computer vision.

Muhamad Zeshan Alam received his B.S. degree in Computer Engineering from COMSATS University, Pakistan, M.S. degree in Electrical and Electronics Engineering from the University of Bradford, UK, and Ph.D. In Electrical Engineering and Cyber-Systems from Istanbul Medipol University, Turkey. He worked at the University of Cambridge as a post-doctoral fellow where his work focused on computer vision and machine learning models. He recently joined Brandon University, Canada, as an assistant professor while also working as a Computer Vision Consultant at Vimmerse INC. His research interests include immersive videos, computational imaging, computer vision and machine learning modeling.

Farooq Alam Orakzai is as an Assistant Professor in the Electrical and Computer Engineering Department, COMSATS University Islamabad, Wah Campus. He published several research articles. His current research interest include 5G technologies, unmanned aerial vehicles (UAVs), computer vision, cognitive radio networks, wireless mobile communication, wireless sensor networks, and optical fiber communication.

Zeeshan Kaleem [Senior Member, IEEE] is serving as an Assistant Professor at COMSATS University Islamabad, Wah Campus. He consecutively received the National Research Productivity Award (RPA) awards from the Pakistan Council of Science and Technology (PSCT) in 2017 and 2018. He won the Higher Education Commission (HEC) Best Innovator Award for the year 2017, and there was a single award all over Pakistan. He is the recipient of the 2021 Top Reviewer Recognition Award for IEEE TRANSACTIONS on VEHICULAR TECHNOLOGY and published over 100 technical papers and patents.

Chau Yuen (S'02-M'06-SM'12-F'21) received the B.Eng. and Ph.D. degrees from Nanyang Technological University (NTU), Singapore, in 2000 and 2004, respectively. He received the IEEE Asia Pacific Outstanding Young Researcher Award in 2012 and IEEE VTS Singapore Chapter Outstanding Service Award on 2019. Currently, he serves as an Editor for the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, IEEE System Journal, and IEEE Transactions on Network Science and Engineering. He is a Distinguished Lecturer of IEEE Vehicular Technology Society.