DRONE DETECTION METHODS AND COMPARISON METRICS FOR AUDIO AND VISION MODALITIES

by

Furkan Oruç

B.S., Computer Education and Educational Technology, Boğaziçi University, 2019

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Computer Engineering
Boğaziçi University
2024

ACKNOWLEDGEMENTS

First of all, I would like to express my deepest gratitude to my thesis supervisor, Assist. Prof. Hüseyin Birkan Yılmaz, for his great contribution and availability to connect his graduate students. Also, this thesis is supported by the Boğaziçi University BAP under the grant number BAP-SUP-17862.

Another special appreciation will go to my research team and friends of our team in NetLab of Computer Engineering Department of Boğaziçi University. I am particularly thankful to team, Kerem Avcı, Yusuf Özben, and Ömer Şahin Albayram for their invaluable contributions during the real-time experiments. Also, I am thankful my girlfriend Zeynep for her enlightening support.

Lastly, I dedicate the thesis to our great library as one of the last theses before the reconstruction of Aptullah Kuran Library in Boğaziçi University.

ABSTRACT

DRONE DETECTION METHODS AND COMPARISON METRICS FOR AUDIO AND VISION MODALITIES

Drones are unmanned aerial vehicles with a significantly growing importance in science, technology, and defense systems. However, their existence and detection a complex issues in terms of risky situations and compromise security. In this thesis, a multi-modal drone detection system has been designed and evaluated using a real experimental sample. The drone detection procedure has two modality approaches: audio and optical modalities. While Yolov8, Long-short-term memory (LSTM), and Vision transformers are applied and evaluated for optic modality, Convolutional neural network (CNN) and LSTM are evaluated and applied for audio modality. After that, a fusion approach for the two modalities is evaluated and interpreted with the affection of different levels of darkness conditions (20, 40, 60, and 80 % of darkness respectively.) Experimental results indicate that precision-recall curves and ROC curves of different models for both modalities. The finding of results reflects insights for further research approaches in terms of the importance of the study with regard to using another modality or thermal infrared imaging. It also reflects the total prediction increase after fusion and darkness novelties. The challenge becomes when adding darkness to vision samples classifying the final detection results of modalities for each frame or sound snippet. The final results demonstrate that using logical methods different levels of samples from different darkness levels might affect the overall performance of probability detection for vision and audio modalities.

ÖZET

SES VE GÖRÜNTÜ MODALİTELERİ İÇİN DRONE TESPİT YÖNTEMLERİ VE KARŞILAŞTIRMA ÖLÇÜMLERİ

Dronlar, bilim, teknoloji ve savunma sistemlerinde önemli bir yere sahip olan insansız hava araçlarıdır. Ancak, tehlikeli durumlar ve güvenlik açısından varlıkları ve tespitleri karmaşık konulardır. Bu tez kapsamında, gerçek bir deneysel örnekle tasarlanmış ve değerlendirilmiş çok modallı bir drone tespit sistemi mevcuttur. Drone tespit prosedürü iki modalite yaklaşımına sahiptir: ses ve optik modaliteler. Optik modalite için Yolov8, uzun-kısa süreli bellek (LSTM) ve görüş dönüştürücüleri uygulanırken, ses modalitesi için Evrişimli Sinir Ağı (CNN) ve LSTM değerlendirilmiş ve uygulanmıştır. Daha sonra, iki modalitenin füzyon yaklaşımı farklı karanlık seviyelerinin etkisiyle (sırasıyla yüzde 20, yüzde 40, yüzde 60 ve yüzde 80 karanlık) değerlendirilmiş ve yorumlanmıştır. Deneysel sonuçlar, her iki modalite için farklı modellerin hassasiyet-geri çağırma eğrilerini ve ROC eğrilerini göstermektedir. Sonuçların bulguları, başka bir modalite kullanımının veya termal kızılötesi görüntülemenin önemi açısından daha fazla araştırma yaklaşımları için içgörüler yansıtmaktadır. Ayrıca, füzyon ve karanlık yeniliklerinden sonra toplam tahmin artışını da yansıtmaktadır. Meydan okuma, görüş örneklerine karanlık eklenmesiyle her kare veya ses parçacığı için modalitelerin son tespit sonuçlarını sınıflandırmak olmuştur. Sonuçlar, farklı karanlık seviyelerinden farklı örnekler için mantıksal yöntemler kullanıldığında, görüş ve ses modalitelerinin genel olasılık tespit performansının etkilenebileceğini göstermektedir.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS iii
ABSTRACT iv
ÖZET v
LIST OF FIGURES viii
LIST OF TABLES xi
LIST OF SYMBOLS xii
LIST OF ACRONYMS/ABBREVIATIONS xiv
1. INTRODUCTION 1
1.1. Problem Definition 1
1.2. Organization 2
2. LITERATURE REVIEW AND RELATED WORK 3
2.1. Optical And Acoustic Drone Detection Approaches 3
2.2. Day and Night Effect on Drone Detection 4
2.3. Acoustic Drone Detection Methods 5
2.4. Drone Detection with Multi-model and Transformers 6
3. EXPERIMENTAL SETUP 8
3.1. Vision Modality 11
3.2. Audio Modality 13
4. METHODOLOGY 14
4.1. Dataset 14
4.1.1. Dataset Directory Structure 15
4.1.2. Ground Truth Labeling 16
4.1.3. Synthetic Darkness Levels 17
4.1.4. Dataset Compositions 18
4.2. Data Preprocessing 19
4.2.1. Intersection Over Union (IoU) 19
4.2.2. Normalization 20
4.2.3. Feature Extraction 22

4.2.4. Data Splitting 23
4.3. Transfer Learning 23
4.4. Architectures of Vision Modality Models 24
4.4.1. Yolov8 24
4.4.2. Vision Transformers 26
4.4.3. LSTM (Long Short-Term Memory) 28
4.5. Architectures of Audio Modality Models 31
4.5.1. Convolutional Neural Networks (CNN) 31
4.5.2. LSTM (Long Short-Term Memory) 32
4.6. Fusion Strategies of Vision and Audio Modalities 33
5. EXPERIMENTAL RESULTS 36
5.1. Performance Metrics 36
5.2. Performance of Vision Models on CONNECT Dataset 37
5.2.1. Precision-Recall and ROC Curves 38
5.3. Effect of Darkness on Vision Models 39
5.3.1. Performance Under 20 % Darkness 41
5.3.2. Performance Under 40 % Darkness 41
5.3.3. Performance Under 60 % Darkness 42
5.3.4. Performance Under 80 % Darkness 42
5.4. Performance of Audio Models 44
5.5. Comparison of Two Modalities on CONNECT Dataset 45
5.6. Performance of Fusing AudioDNet and OpticDNet 47
5.7. Performance of Vision Models on CONNECT-M Dataset 49
6. DISCUSSIONS 53
6.1. Remarks 53
7. CONCLUSION 54
REFERENCES 55

LIST OF FIGURES

Figure 1.1. Mean deviations of average drone strikes that US military bases recorded over 31 days, adapted from [1]. 1

Figure 2.1. Proposed methodology, adapted from [2]. 6

Figure 3.1. Schematic map of the experimental area. 8

Figure 3.2. Designed setup outside. 9

Figure 3.3. Designed setup inside. 10

Figure 3.4. Image source 33U series color industrial camera structure. 11

Figure 3.5. DJI mavic air 2 drone and control mechanism. 12

Figure 3.6. Equipment for audio modality measurements. 13

Figure 4.1. Sample image from the dataset. 15

Figure 4.2. Directory tree of the entire dataset. 15

Figure 4.3. Creating new bounding boxes for annotation. 16

Figure 4.4. Sample detection result under 20 and 40 % darkness. 18

Figure 4.5. Sample detection result under 60 and 80 % darkness. 18

Figure 4.6. Ground truth (green) and predicted (red) bounding boxes. 20

Figure 4.7. Transfer learning architecture. 24

Figure 4.8. Standard yolo architecture, adapted from [3]. 25

Figure 4.9. Yolov8 architecture, adapted from [3]. 25

Figure 4.10. Example of predicted result with yolov8. 27

Figure 4.11. Vision transformer object detection architecture, adapted from [4]. 28

Figure 4.12. Example predicted result with vision transformer. 29

Figure 4.13. Example predicted result with vision transformer. 29

Figure 5.1. Precision recall and ROC curves under no additional darkness. . . 39

Figure 5.2. Four different level of darkness. 40

Figure 5.3. Precision recall and ROC curves under 20 % darkness. 41

Figure 5.4. Precision recall and ROC curves under 40 % darkness. 42

Figure 5.5. Precision cecall and ROC curves under 60 % darkness. 43

Figure 5.6. Precision recall and ROC curves under 80 % darkness. 43

Figure 5.7. ROC curve of CNN and LSTM. 44

Figure 5.8. Fusion method architecture. 48

Figure 5.9. Precision recall curve result of fusion method. 48

Figure 5.10. Precision recall and ROC curves on CONNECT-M dataset. 49

Figure 5.11. Precision recall and ROC curves (CONNECT-M trained). 50

Figure 5.12. ROC curve of fusion with EGC method and basic method. 52

LIST OF TABLES

Table 3.1. Inner side and yellow line coordinates. 9

Table 4.1. CNN layers of our approach and input-output shape. 32

Table 4.2. OR logic of ground truth (GT) values. 34

Table 5.1. Ground truth (GT) and vision models on CONNECT dataset. . . 38

Table 5.2. Confusion matrix for audio ground truth (GT) and audio DNet. . 45

Table 5.3. Number of frames for audio ground truth (GT) and audio DNet. . 46

Table 5.4. Confusion matrix for optic ground truth (GT) and opticDNet-yolo. 46

Table 5.5. Number of frames for optic ground truth (GT) and opticDNet-yolo. 46

Table 5.6. Multi-class confusion matrix for opticDNet-yolo and audioDNet on CONNECT dataset. 47

Table 5.7. Fused ground truth (GT) statistics. 47

Table 5.8. Ground truth (GT) and vision models on CONNECT-M dataset. . 51

LIST OF SYMBOLS

A	First area definition of bounding box
B	Second area definition of bounding box
C	Constant to reduce intensity
f	The model function
g	Model with added layers
$hatx_i^{(k)}$	Feature vector normalization of mini-batch
$I_{\text{dark}}(x, y)$	Original density
$L_{fine}(\theta')$	Loss function of fine-tuning
$L_{pre}(\theta)$	Loss function of pre-training
(m)	Index of the time frame
mu_β	Batch mean
N	Total number of samples
$P(m, \omega)$	Power spectrum over time
p	Probability
r	Random variable from Bernoulli distribution
$tanh(x)$	Tangent function
$(w(n))$	Windows function, hamming window
$\mathbf{W}$	Weight matrix
$\mathbf{Wx}$	Matrix multiplication
$\mathbf{b}$	Bias vector
x_i	Amplitude values
$x_i^{(k)}$	Feature value of each k
$(x(t))$	Signal function
$y^{(k)}$	Output of next layer
y_j	New task data point
α	First focal loss parameter
β	Adam optimizer

$\beta^{(k)}$	First learning parameter
γ	Second focal loss parameter
$\gamma^{(k)}$	Second learning parameter
μ_β	Calculated batch mean
ϕ	Activation function
σ_β^2	Mean variance
θ	Parameter of pre-training model
θ'	Fine-tuned parameters
ω	Angular frequency

LIST OF ACRONYMS/ABBREVIATIONS

2D	Two Dimensional
3D	Three Dimensional
AP	Average Precision
CNN	Convolutional Neural Network
CONNECT	Localizing Drone Signals and Remote Control Units
dB	Decibel
D-CNN	Deep Convolutional Neural Network
EGC	Equal Gain Combining
FP	False Positive
FPR	False Positive Rate
FPN	Feature Pyramid Network
Fps	Frame per Second
FN	False Negative
GAANet	Ghost Auto Anchor Network
GAM	Gradient Attention Module
GFCC	Gammatone Frequency Cepstral Coefficients
GiB	Gigabyte
HOG	Histogram of Oriented Gradients
Hz	Hertz
IR	Infrared
IoU	Intersection Over Union
kHz	Kilohertz
LSTM	Long-Short Term Memory
mAh	Miliamper-hour
MFCC	Mel Frequency Cepstral Coefficients
mAp	Mean Absolute Precision
MS COCO	Microsoft Common Objects in Content
PAN	Path Aggregation Network

RCNN	Region Based Convolutional Neural Network
ReLu	Rectified Linear Unit
RMS	Root Mean Square
ROC	Receiver Operator Characteristics
RNN	Recurrent Neural Network
SVM	Support Vector Machine
STFT	Short-Time Fourier Transform
SWIR	Short-Wave Infrared
TB	Terabyte
TF-Net	Tensorflow Network
TN	True Negative
TP	True Positive
TPR	True Positive Rate
UAV	Unmanned Aerial Vehicle
ViT	Vision Transformer
VR	Virtual Reality
YOLO	You Only Look at Once
ZCR	Zero-Crossing Rate

1. INTRODUCTION

1.1. Problem Definition

Drones are widely utilized, employed, and customized in military and civilian actions [1, 2]. We can observe that drones have variety in terms of their shape, size, and speed per hour [3,4]. Therefore, we might need a system or technological approach to observe and detect their movements during their flight [5]. However, defence systems suffer from malicious drone attacks that might cause explosion and other various types of vulnerabilities [6]. Observing small or medium drones can be challenging, especially at night and severe weather conditions such as foggy or stormy weather.

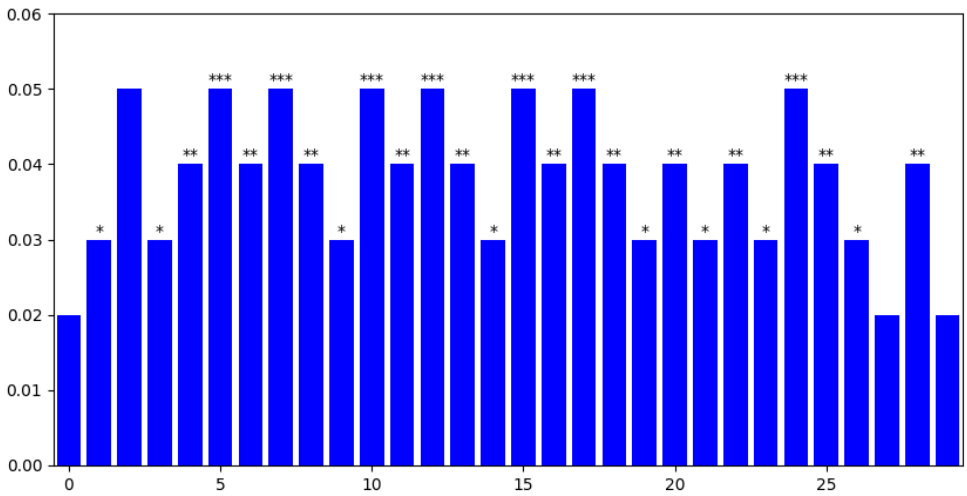


Figure 1.1. Mean deviations of average drone strikes that US military bases recorded over 31 days, adapted from [1].

Figure 1.1 illustrates average drone strike attacks that US military bases recorded in the Middle East per day over a month. Figure 1.1 also shows that almost every day of a month there is at least one drone strike that military bases cope with. Thus, intrusion detection systems are becoming popular and widespread based on the methods obtained from drone detection and drone localization. For the drone detection

systems, the important point is analyzing or understanding when a drone is approaching. A drone intrusion detection system has components such as information sources, communication links, system payload, and autonomous control basis [6]. We want to focus on information sources for a detection system among these components.

This thesis work will discuss and present the results of how we tried to get information about a flying drone using audio and vision modalities with our unique experiment setup. We also tried challenges such as adding darkness as an example of poor conditions. After data collection, we focused on drone detection algorithms with multi-modal data separately and data fusion strategies.

1.2. Organization

The thesis structure contains seven sections with few of subsections; introduction, literature review and related work, experimental setup, methodology, experimental results, discussions, and conclusion. Introduction part contains our motivation and the need behind this work. Literature review and future work mentions what kind of previous work has done so far. Furthermore, experimental setup shows how and what kind of technology what we used such as system architecture, microphones, computer devices, cameras and drone.

In the methodology section, our dataset and ground truth forming methods has been discussed first of all. Also, the data processing approaches we implemented (IoU, normalization, feature extraction and data splitting) for modelling has been presented. After that, the models such as Yolov8, transformers, LSTM (long-short term memory), and CNN (convolutional neural netrowk) that is been implemented for drone detection is demonstrated and explained with transfer learning and darkness features. Lastly, modality comparison schema and methods is discussed. In the results section, the results and performance of our method are shown with Precision and Recall Curves and ROC (Receiver operating characteristic) Curves. Lastly, probability comparison results are followed by a brief conclusion and future work discussions.

2. LITERATURE REVIEW AND RELATED WORK

2.1. Optical And Acoustic Drone Detection Approaches

It is known and an accepted issue that Drones which is also called UAVs (Unmanned Aerical Vehicles) have grown for a wider ranger uses day by day in terms of their popularity and effectiveness as Misbah et al. [7] stated. Although using drones in scince, constructions, and engineering is quite useful and has advantages widely, it may contain illegal use of UAVs such as terrorist attacks. Therefore, there are previous works that focus on detecting drones within various challengeable conditions to mitigate risks of attack and privacy. In this section, previous works related to our approach will be discussed and mentioned.

Firstly, Ding et al. [8] focuses on drone detection with camera and acoustic approaches. They implemented a audio and optical unmanned aerial vehicle 3D (3-Dimensional) trajectory exposure system to detect and track drones. As opposed to our work, Ding et al. [8] designed multipe sensor modules with lidar. Field experiments has been applied, detection range and 3D position accuracy has been estimated with their approach. As camera detection approach, SWIR (Short-wave infrared) cameras are utilized, as acoustic detection approach, microphone arrays are utilized.

Another approach that Svanström et al. [9] conducted is related to drone detection and tracking using thermal sensors for vision and acoustic sensors for audio detection. Yolov2 detector and classifier is implemented with the data gathered using IRcam (Infrared Camera) and VRCam (Virtual Reality Camera). Their experimental research part contains drone driving in Halmstad Airport in Sweden. In addition to our work, they conducted multiple detection procedures using airplane, bird, and helicopter detection and classification in addition to drones. As a result part of the work, they present average precision and recall for 0.91 and 0.87 for close distance, 0.89 and 0.80 for medium distance, and 0.82 and 0.48 for long distance respectively.

Kim et al. [10] collected image and audio data using two identical drones in the sky at an euclidean distance of 20 meters. As deep learning models, Yolov5 was used for computer vision and CNN was utilized for acoustic data. As feature extraction methods for audio classification, MFCC (Mel Frequency Cepstral Coefficients) is used. MFCC is defined as a non-linear mapping method of the original frequency [11]. Sigmoid and softmax activation functions are added unlike our CNN approach (ReLu (Rectified linear unit) activation function used in our methodology section.) Similar to ours, for the decision of fusion, 'OR' operation is used as the final result of CNN and Yolov5. At the end, they claim that after applying the OR function to the result of vision and acoustic, the accuracy is improved to 92.53 %.

Wang et al. [12] utilized latest version of Yolo which is Yolov8 for tiny drone detection. The dataset of authors of [12] contains tiny drone images with high resolution. In addition to regular Yolov8 structure, they added tiny-target detection head. As an improvement of feature fusion model, GAM (Gradient Attention Module) which is integrated into CNN architecture. Their baseline model with addition to Yolov8 improve precision performance by 11.9 %, recall performance by 15.2 %, and mAP (Mean Average Precision) by 9 %. GAM also reduced the number of parameters by 59.9 %, which makes it more suitable for engineering development.

2.2. Day and Night Effect on Drone Detection

With regard to detection under darkness, Khan et al. [13] proposed a drone detection network called GAANet (Ghost auto anchor network) under night and low visibility conditions using Yolov5. Then they claim that they implemented autoanchor calculation to block ghost convolution for their model architecture. Finally, experimental results show that Yolov5 plus their implementation of ghost convolution ,which is called GAANet, has higher overall precision and recall around 2.3 and 1.4 % respectively. Tf-net (Tensorflow network) is also another approach that authors of [7] proposed as a tiny feature network for drone detection under night time vision. Their network involves detect UAVs during the night by employing detection based

on IR(Infrared) images since it is a great challenge under complex background conditions. They simulated four different Yolov5 algoritms with different hyperparameter tuning approach which are called YOLOv5s, YOLOv5m, YOLOv5l and YOLOv5n respectively. In conclusion, they found out that TF-Net in terms of precision give better results which is an improved version of Yolov5s as the authors of [7] stated. The results are indicated as 95.7 % precision, 84 % mAp (Mean Absolute Precision), and 44.8 % IoU (Intersection over Union).

We used adding darkness level to our dataset to get more complex background and detection feature. In addition to usual sky images, Munir et al. [14] used rainy and foggy weather to make their dataset more complex in terms of background. Methods for benchmarking they used are Yolov8, Yolov5, and Faster R-CNN (Region based Convolutional Neural Network). Their interpretation shows that Faster-RCNN and YOLOv5 can detect the UAVs in the image despite the rainy conditions. As a result, UAV detection performance of the selected models with AP (Average Precision) are 69.3, 67.2 and 59.2 % for Yolov5, Yolov8 and Faster R-CNN respectively.

2.3. Acoustic Drone Detection Methods

Casabianca et al. [15] propose an audio drone detection method using various acoustic sound detection and classification methods. Their proposed methodology involves feature extraction using and comparing several methods such as Mel-frequency cepstral coefficients (MFCC), Gammatone cepstral coefficients (GTCC), and Zero-crossing rate (ZCR). After feature selection, they trained Support Vector Machine (SVM) for classification.

Audio acquisition is similar to our approach and they applied digitization to 44.11 kHz. In addition, since they worked with a large diverse database. They hold 20 % of the data for validation and used 10-fold as k in cross-validation. At the end, medium Gaussian SVM reaches out accuracy of 99.8 % recall and 99.9 % for precision. This approach that focuses on acoustic drone detection methods mentioned by the authors

of [15]. Their work involves both CNN and RCNN (Region-Based Convolutional Neural Network) approaches with late fusion methods using acoustic signals. In terms of the structure of the layers of the model, batch normalization and 2D convolutions has been used in addition to flatten, dense, and dropout layers. Their method has been evaluated according to an ensemble deep neural network. The weighted accuracy has achieved 94.7 %.

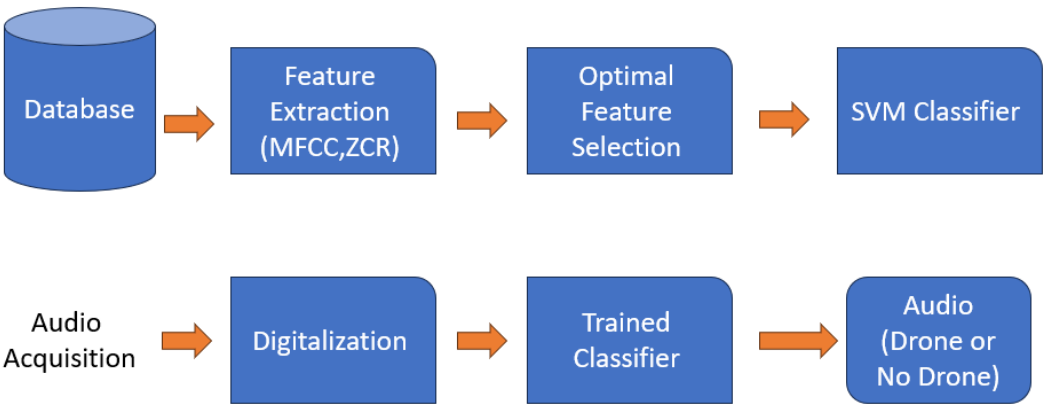


Figure 2.1. Proposed methodology, adapted from [2].

2.4. Drone Detection with Multi-model and Transformers

Phung et al. [16] developed a detection system for drones with quadcopter under complex background conditions such as occlusion. Similar to our work, our image dataset contains a typical effective background such as exit gate of the campus. They trained three different models (bileNetV2, EfficientDet-D3 and YoloV4) with 3150 drone images. As compared to detection results, Yolov4 gives the highest results with Average Precision (AP) = 34.63 %, following EfficientDet-D3 AP = 24.03 % and bileNetV2 AP = 20.78 %. They also claim the reason behind Yolov4 gives the best result is that they applied HOG (Histogram of Oriented Gradients) and interpolation method which has also been utilized in MS COCO dataset [17]. Zhang et al. [18] developed a transformer based YOLO architecture for object detection and although it does not only contain drones, the work uses both YOLO and vision transformers approaches.

Authors of [19] showed that transformers can also be used by Image datasets as (ViT) Vision Tranformers. Jamil et al. [20] stated that detecting and distinguishing malicious drones by using vision transformers outperforms CNN models. Their approach clearly demonstrates that position embeddings are applied to fixed-size drone images. Finally, general accuracy for different class of objects in their confusion matrix is 98.28 % when D-CNN (Deep Convolutional Neural Network) models are 93.50 %.

The last related work to our approach in terms of Yolo and transformers proposed by the authors of [21]. Authors observed that high target density, and a large proportion of small targets may result in high false alarm rates. Therefore, they designed a novel Yolov7 approach for multi-scale drone detection which is named as YOLO-ViT. They added feature extraction backbone network to reduce the complexity of the model. After that, K-means clustering method is utilized to arrange anchor boxes for detection. As the result section, they achieved reduction of number of parameters by 49.9 % with mAP improvement of 0.9 % as compared to original method with the calculated indicators of precision, recall, and F-1 score.

3. EXPERIMENTAL SETUP

In this section, the experimental setup that is used for the drone detection system will be explained. At the Computer Engineering Department of Boğaziçi University, we carried out our experiments in area inside North Campus of Boğaziçi University.

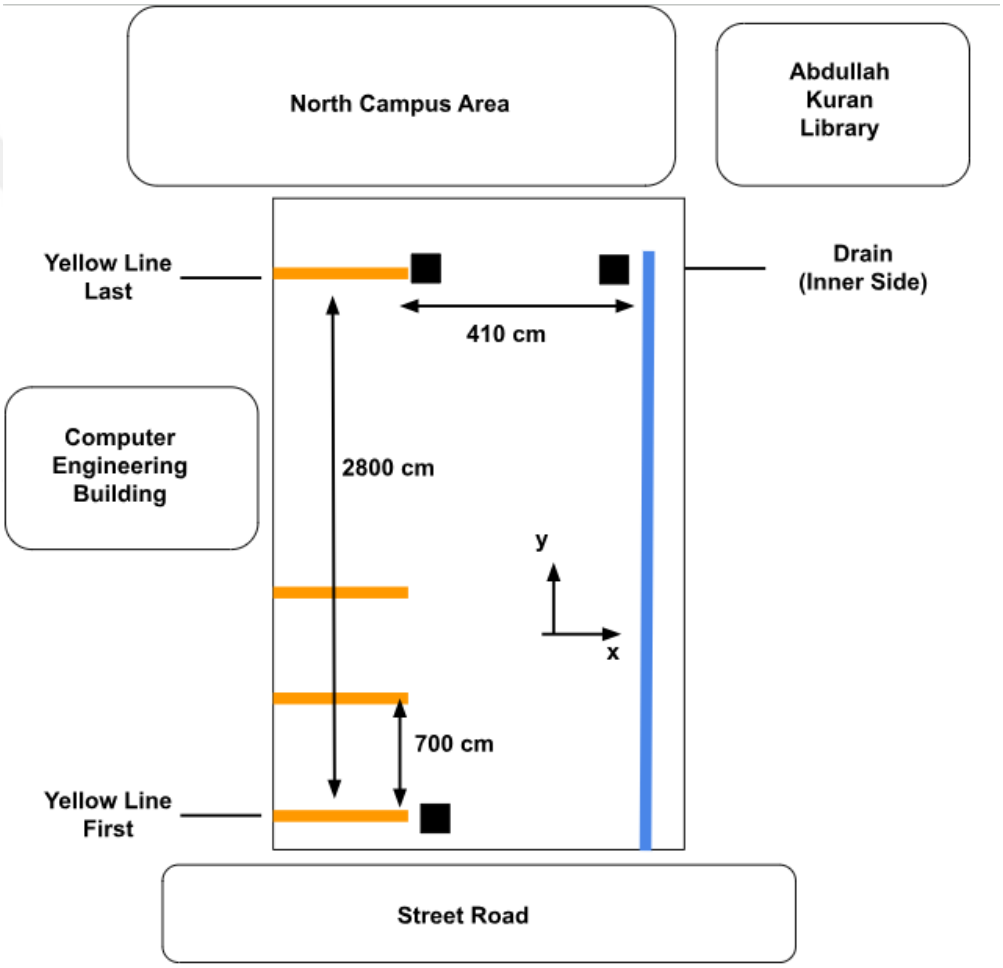


Figure 3.1. Schematic map of the experimental area.

Figure 3.1 shows rectangular area we used for experiments is area with yellow lines. We calculated the location of our setup according to that procedure. Firstly, we calculated the total distance between the first and last yellow lines, which is 2800 cm. Then, we observed that the distance between each line was 700 cm. After that, there

was a drain line through the right corner of the rectangular area. We detected it as the inner side. Finally, we located our setup place between the last yellow line and the end of the inner side, which is calculated to be a total of 410 cm. During the experiments, we drove the drone in different areas around the lines with up, down, left, right, back, and forward moves.

Table 3.1. Inner side and yellow line coordinates.

Label	**Latitude**	**Longitude**
Yellow Line - Last	41.0857753	29.0437393
Yellow Line - First	41.08570841	29.04412709
Drain (Inner Side)	41.08581119	29.04373023

Table 3.1 shows the coordinates of first and last yellow lines, in addition, the coordinates of drain line.



Figure 3.2. Designed setup outside.

Our setup with audio, vision and radio frequency modalities is set around the location we mentioned. According to Figure 3.2, there are three microphones and three Universal Software Radio Peripherals (USRP) devices for recording on the table. Since radio frequency modality has not been used in this part of the work.

In addition to that, on the right and left corners of the table, there are two cameras located parallel on top of the tripods. Also, three computers has been used; the first one is used for setup, the second and the third one are used for observing the left and right camera footage simultaneously. The distance between the closest point of the two tripods is 87 cm.



Figure 3.3. Designed setup inside.

In terms of another point of view, in Figure 3.3, we can clearly observe microphones are located in series towards to the end of the table as an inside approach. Other than the data collection setup, computer system features for the models for data analysis, data visualization, model learning, and testing are as follows:

- Hardware Model: Dell Precision 3600,

- Memory: 64 GiB,
- Processor: 12th Gen Intel Core i9-12900 x 24,
- Disk Capacity: 4.5TB,
- Graphics: NVIDIA GA102GL RTX A5000,
- OS Name and Type: Ubuntu 22.04.3 LTS, 64-bit.

3.1. Vision Modality

To be able to obtain drone footages, we use two identical cameras at the data collection setup. These cameras are placed on the right and the left sides of the table.



(a) Industrial camera. (b) Industrial camera with fixing device.

Figure 3.4. Image source 33U series color industrial camera structure.

Figure 3.4 demonstrates the camera we used during our experiments. The cameras have focal length size, fps (frame per second), and pixel size arrangements. We worked with Pixel Size: H: 3.45 µm, V: 3.45 µm, focal length size is 1.0 and 2.0 arranged with 30 fps recording and 35 mm lens thickness. Further features of our cameras are as follows:

- USB 3.0 Interface,
- 2/3 inch Sony CMOS Pregius IMX250 Sensor,
- 2,448×2,048 (5 MP) Resolution,

- Global Shutter,
- Trigger and I/O Inputs,
- 29x29x43 mm width, height, and length.



(a) DJI mavic air 2 drone. (b) Control of DJI mavic air 2.

Figure 3.5. DJI mavic air 2 drone and control mechanism.

Furthermore, as shown in Figure 3.5, the model and version of our drone is DJI Mavic Air 2 with its user control device and four propellers. The control device has three options: Cine, Normal and Sport modes.

We mostly completed our experiments in normal mode.In addition, the control device is used for up, down, left, right, back, and forward moves. The DJI Mavic Air 2 drone weighs 570 g and the diagonal distance is 302 mm. Also, the drone has its own gimbal with 3-axis (tilt, roll, pan). However, we haven't used it since we did not work with the drone's point-of-view images. The features of our drone for experiments are as follows:

- Battery Capacity: 2500 mA h,
- Max Flight Time: 34 minutes,
- Angular Velocity: 250 °/s (N Mode),
- Propellers: Quick release, low noise, folding.

3.2. Audio Modality

We have three identical microphones for audio modality creation. The model and version of the microphones are same and they are Behringer ECM8000. It is a condenser microphone designed for use with real-time analyzers. The sound spectrum we gained so far is up to 8192 Hz. Figure 3.6 indicates a microphone located with 45 degrees upwards to record the sound of the drone during experiments. In addition, a cable was used to connect with the sound recorder to obtain the voice for each microphone. The features of our microphone are as follows:

- Connector Type: XLR,
- Polar Pattern: Omnidirectional,
- Audio Sensitivity: 70 dB,
- Item weight: 6.4 Ounces,
- Phantom power from +15 to +48 V.



(a) Microphone. (b) Multi channel sound recorder.

Figure 3.6. Equipment for audio modality measurements.

4. METHODOLOGY

4.1. Dataset

Our unique dataset consists of two parts: Audio Dataset and Vision Dataset. Vision Dataset consists of total 28 videos ranging from five minutes to twenty minutes recorded with two different 30 fps camera described in Section 3. As a results, total of 518 700 frames were generated for the vision dataset. For each experiment, two videos were created at the same time, both left and right cameras were recording during the process from different angles. There are total of 14 experiments, thus, there are 28 different videos where the half of the videos are from right camera.

Concerning audio frames, it consists of 28 sound recordings divided into half, first and second microphones. The locations of the microphones are different for audio modality data gathering. For the synchronization, we ensured that the total number of frames will be the same. Therefore, the audio dataset also consists of 518 700 different audio samples or audio clips.

The images usually contain one drone in each frame with a complex background with external objects such as buildings, a gate, and trees. In addition, there may not be a flying drone in any image frame for vision modality. This method has been applied to eliminate simplicity and overfitting.

The sample image shows a randomly selected sample image with our drone located on the left corner of the gate. Here you may see that the environment is a challenging one for a drone detection task with having regular and irregular shapes at the background. Furthermore, the audio dataset contains sound waves with different range of decibels. Like vision modality, some audio snippets might not contain drone sound.



Figure 4.1. Sample image from the dataset.

4.1.1. Dataset Directory Structure

As mentioned before, the dataset consists of video and sound frames, divided into left and right camera, and first and second microphone recordings.

```
/
└─ Dataset/
   ├─ Vision Dataset/
   │  ├─ Left/
   │  │  ├─ LVision1.avi
   │  │  ├─ LVision2.avi
   │  │  ├─ ....
   │  │  └─ LVision14.avi
   │  └─ Right/
   │     ├─ RVision1.avi
   │     ├─ RVision2.avi
   │     ├─ ....
   │     └─ RVision14.avi
   └─ Audio Dataset/
      ├─ First/
      │  ├─ FirstMic1.wav
      │  ├─ FirstMic2.wav
      │  ├─ ....
      │  └─ FirstMic14.wav
      └─ Second/
         ├─ SecondMic1.wav
         ├─ ....
         └─ SecondMic14.wav
```

Figure 4.2. Directory tree of the entire dataset.

The Directory tree indicates that the dataset is separated into four parts: left camera image, right camera image, first microphone audio and second microphone

audio. Also, the video format has been recorded as AVI format; the sound has been recorded in WAV format and 30 frames were obtained per second.

4.1.2. Ground Truth Labeling

We used semi-automatic labeling to form the ground truth. We ran three models on the images and we manually checked the suspicious cases such as: drone disappearing in the frame when the previous and the following frames have drone, when a vision model claims to have a drone in the frame but the others are deciding the opposite. Therefore, the following approach is utilized to developed our labelling method.

```
function IsClose(box1, box2, threshold)
    center1 ← ((box1.x_min + box1.x_max)/2, (box1.y_min + box1.y_max)/2)
    center2 ← ((box2.x_min + box2.x_max)/2, (box2.y_min + box2.y_max)/2)
    distance ← √((center2.x − center1.x)² + (center2.y − center1.y)²)
    return distance < threshold
end function
function CreateNewBoundingBoxes(frm1_boxes, frm2_boxes, threshold)
    new_boxes ← empty list
    for each box1 in frm1_boxes do
        for each box2 in frm2_boxes do
            if IsClose(box1, box2, threshold) then
                new_box ← box2
                Append new_box to new_boxes
                break
            end if
        end for
    end for
    return new_boxes
end function
```

Figure 4.3. Creating new bounding boxes for annotation.

With regard to vision ground truth approach, we have an algorithm that will ease our task for data labeling. Figure 4.3 demonstrates an algorithmic approach for creating a new bounding box for data labelling. Firstly, 'IsClose' function calculates the euclidean distance between two bounding boxes labelled. If it returns true and the threshold is less than a certain one, 'CreateNewBoundingBoxes' function will generate new bounding boxes through each pair of bounding from nth and (n+1)th frames. Yuan et al. [22] refers to a similar approach. They calculate the Intersection Over Union (IoU) between two objects and if the area between them is under a certain threshold, they generate new bounding boxes for the next frame. More detailed explanation about IoU will be discussed in Section 4.2.1. At the end, after this approach if a bounding box labelled for an image, we called the frame True (1), otherwise we called it False (0).

Moreover, in terms of the audio ground truth approach, no special method was used for data annotation. In some of the experiments, no drone was flown, therefore there was no particular object sound. We decided to label those sound frames or sound snippets as False. However, the other experiments have a specific time range for drone landing and taking off. At this point, the first and last ten seconds, which means 300 sound frames at 30 fps, have been decided to be labeled as False and the remaining ones are annotated as True.

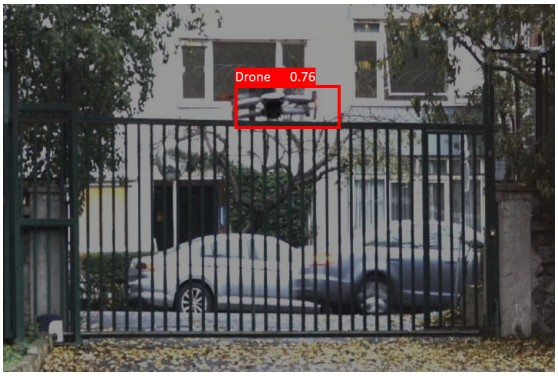
4.1.3. Synthetic Darkness Levels

After creating vision models, we wanted to apply different levels of darkness and test the models with that specified data. How darkness added and how it works will be discussed in Section 5.3. In addition to normal dataset, we tested three different model with four different darkness levels. Darkness methods applied are as follows:

- 20 % Darkness,
- 40 % Darkness,
- 60 % Darkness,
- 80 % Darkness.



(a) Detection with 20 % darkness.



(b) Detection with 40 % darkness.

Figure 4.4. Sample detection result under 20 and 40 % darkness.



(a) Detection with 60 % darkness.



(b) Detection with 80 % darkness.

Figure 4.5. Sample detection result under 60 and 80 % darkness.

Figure 4.4 and Figure 4.5 show that random detection results with bounding boxes. According to those examples, we can observe that the more darkness added the more the probability of detection is decreasing. In addition, some algorithms may show false positive approach as it is shown in the 60 % of darkness example in left part of Figure 4.5.

4.1.4. Dataset Compositions

We have total of six dataset with manipulation of adding darkness. First of all, we used CONNECT Dataset for all manipulations with no darkness addition and

darkness level added datasets are defined with four different levels in addition to normal CONNECT dataset. They are defined as follows:

- CONNECT-D20 Dataset : 20 % darkness addition of entire data,
- CONNECT-D40 Dataset : 40 % darkness addition of entire data,
- CONNECT-D60 Dataset : 60 % darkness addition of entire data,
- CONNECT-D80 Dataset : 80 % darkness addition of entire data.

Also, we are going to observe a new dataset with randomly selected darkness added frames. Darkness Added Dataset contains following features:

- 30 % of the Data : 20 % darkness addition,
- 30 % of the Data : 40 % darkness addition,
- 20 % of the Data : 60 % darkness addition,
- 20: % of the data : 80 % darkness addition.

Lastly, we called this dataset as CONNECT-M dataset. This dataset is more challenging than CONNECT dataset since it has mixture of darkened images. When the detection results are prone to errors due to darkness, we may rely on sound modality and fusing the results. Final results will be given in Section 5.6 of Experimental Results section.

4.2. Data Preprocessing

4.2.1. Intersection Over Union (IoU)

According to the authors of [23], IoU, also known as Jaccard Index, defined as a metric to compare the similarity between two boxes shapes. In this example, the shapes are described as bounding boxes and they are rectangular shapes that vision models calculated The formula for Intersection over Union (IoU) can be expressed as

$$IoU = \frac{|\mathbf{A} \cap \mathbf{B}|}{|\mathbf{A} \cup \mathbf{B}|}. \tag{4.1}$$

It shows the calculation of Intersection Over Union. Area of A represents the area of ground truth bounding box and area of B represents the area of predicted bounding box. IoU is calculated as a probabilistic value as intersection area of A and B over union area of A and B. Within our approach, IoU threshold has been decided as 0.5, the frame results is determined as True Positive (TP), if it is above this threshold. Otherwise, it is determined as False Positive (FP). More detail of calculating Precision-Recall and True Positive Rates - False Positive Rates will be discussed in Section 5.2.1.



(a) 1st example. (b) 2nd example.

Figure 4.6. Ground truth (green) and predicted (red) bounding boxes.

Figure 4.6(a) and Figure 4.6(b) demonstrate the example image data frames for ground truth (green bounding box) and predicted results (red bounding box). The results to calculate IoU for each frame are based on those generated boxes.

4.2.2. Normalization

Normalization is an essential step for object detection with both image and audio data.In the audio dataset, Amplitude or Peak normalization has been applied, McFee et al. [24] defines those values as an essential which makes the calculation of end-to-end sound signal values. Amplitude normalization and its steps can be described as follows:

(i) Firstly, the current amplitude level of the audio signal is calculated, which can be done by computing the Root Mean Square (RMS) with various constants and variables. Root Mean Square simply can be defined as

$$\text{RMS value} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} x_i^2}, \tag{4.2}$$

where x_i represents the amplitude values and N is the total number of samples.

(ii) Then we determine the target amplitude level for all audio samples.

(iii) Thirdly, we calculate the normalization Normalization factor can be defined as

$$\text{Normalization factor} = \frac{\text{Target RMS}}{\text{Current RMS}}, \tag{4.3}$$

where factor by dividing the target RMS value by the current RMS value of the audio signal.

(iv) Lastly, the normalization factor for entire audio signal has been applied and it can be expressed as

$$\text{Normalized audio} = \text{Audio} \times \text{Normalization factor}. \tag{4.4}$$

Batch normalization has been applied since we are working with deep learning models. Authors of [25] states that batch normalization's main principle is to gain stability and performance in terms of improving speed of training for neural networks. The batch normalization process by authors of [25] can be described as follows:

(i) Calculate Batch Mean and Variance: For each feature k, compute the mean μ_β and variance is calculated as

$$mu_\beta = \frac{1}{m}\sum_{i=1}^{m} x_i^{(k)}, \tag{4.5}$$

and σ_β^2 over the mini-batch is defined as

$$\sigma_\beta^2 = \frac{1}{m}\sum_{i=1}^{m}(x_i^{(k)} - \mu_\beta)^2. \tag{4.6}$$

(ii) Normalize: Normalize the feature vectors of the mini-batch. Normalized feature vectors can be calculated as

$$hatx_i^{(k)} = \frac{x_i^{(k)} - \mu_\beta}{\sqrt{\sigma_\beta^2 + \epsilon}}. \tag{4.7}$$

(iii) Scale and Shift: Then scale and shift the normalized values with learnable parameters $\gamma^{(k)}$ and $\beta^{(k)}$. Normalization of scale-shift can be calculated as

$$y_i^{(k)} = \gamma^{(k)}\hat{x}_i^{(k)} + \beta^{(k)}, \tag{4.8}$$

where $x^{(k)}$ is the input and μ_β is the calculated batch mean, σ_β^2 is the batch variance, and ϵ is the constant for improving stability and speeding up the performance. After normalization, we can observe that scaling and shifting is applied using parameters such as $\gamma^{(k)}$ and $\beta^{(k)}$. Also, $y^{(k)}$ is the output that goes into the next layer.

4.2.3. Feature Extraction

As an example of feature extraction part of our preprocessing steps, spectrogram has been applied to the audio dataset.

Authors of [26] clearly explain how spectrograms transform audio signals with a simply approach. The spectrogram is both visual representations of the frequency spectrum of a sound signal as Gong et al. [26] stated. The mathematical processes underlying these transformations are the Fourier Transform and Mel-Frequency conversion are two different ways to apply spectogram to audio signal frames.

The spectrogram is calculated using the Short-Time Fourier Transform (STFT). According to Gong et al. [26], the STFT is applied to the signal $x(t)$ and is defined as

$$STFT\{x(t)\}(m,\omega) = X(m,\omega) = \sum_{n=-\infty}^{\infty} x(n)w(n-m)e^{-j\omega n}, \tag{4.9}$$

where $x(n)$ is the time domain signal, $w(n)$ is the window function, which is typically a Hamming window, m is the index of the time frame, ω is the angular frequency. The magnitude squared of the Short-Time Fourier Transform (STFT) gives a result as the power spectrum and the result is calculated as

$$P(m,\omega) = |X(m,\omega)|^2, \tag{4.10}$$

where $P(m,\omega)$ is the representation of power spectrum over time.

4.2.4. Data Splitting

For the entire dataset, we splitted the data as 80 % of training, 10 % of validation and 10 % of testing. To be able to avoid overfitting and get better fine-tuning. We used different number indexed video and audio sets. As an instance, LVision1.avi and RVision1.avi has similar frames since they have recorded simultaneously. Within this approach, if a similar video set or audio set has been observed in training part, we used them to add different part of the dataset to the test part.

4.3. Transfer Learning

Yosinski et al. [27] stated that transfer learning is a machine learning technique in which a model has some improvement for a specified task. Our example here is object detection, and it is reused as the starting point for a model on a second task. As an example of transfer learning, we used the Pre-Trained HuggingFace Vision Transformers that used MS COCO dataset during the training phase (details are given in Section 4.4.2). This pre-trained model has already learned to recognize many features of images, in addition, our dataset is used in extra training phase to improve the drone detection performance.

Figure 4.7 indicates the difference between traditional machine learning approach and transfer learning approach. Authors of [28] defines traditional machine learning technologies as learning tasks from scratch.Our transfer learning approach has two aspects: pre-training and fine-tuning: Bert et al. [29] states that the model is firstly should be trained on a large dataset with a broad task using various functions and constants. It can be defined as

$$L_{pre}(\theta) = \frac{1}{N}\sum_{i=1}^{N} f(x_i, \theta), \tag{4.11}$$

where $L_{pre}(\theta)$ is the loss function during our pre training phase, θ are the parameters.

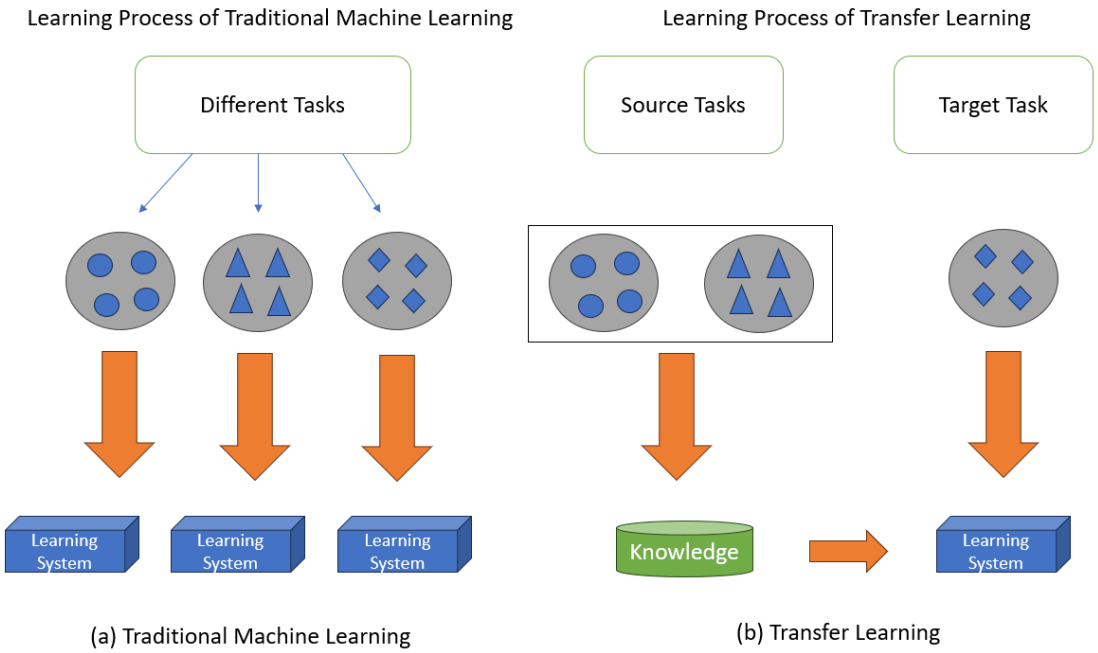


Figure 4.7. Transfer learning architecture.

Secondly, it is shown the adaption of pre-trained model to a specific task. In this situation, that equation provides extra layer and adjusting the parameters. The calculation can be defined as

$$L_{fine}(\theta') = \frac{1}{M}\sum_{j=1}^{M} g(y_j, \theta'), \tag{4.12}$$

where $L_{fine}(\theta')$ is the loss function during fine-tuning, θ' are the fine-tuned parameters, g is the model with added layers for the new task, y_j are the new task data points, and M is the number of samples for the new task.

4.4. Architectures of Vision Modality Models

4.4.1. Yolov8

There are other definitions and different parameters to be defined of YOLO algorithm. However, Yolov8 is the latest version of YOLO (You Only Look at Once) algorithm as of January 2024. Before getting into the details of Yolov8, we need to define what YOLO is and how it is structured with its components.

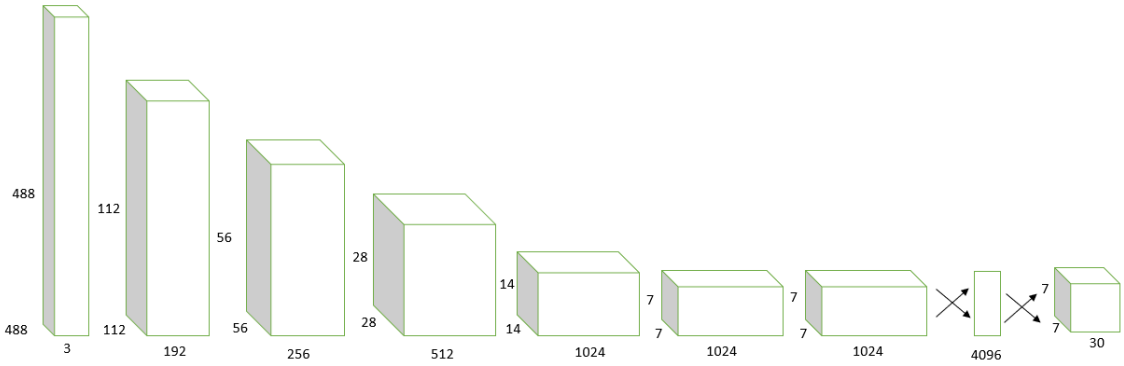


Figure 4.8. Standard yolo architecture, adapted from [3].

Figure 4.8 shows the fundamental Yolo Architecture. First model architecture consists of 24 layers and two connected layers.An average pooling layer is added to a fully connected layer as the backbone of the network as authors of [3] stated. In addition, Yolov8 architecture is demonstrated in Figure 4.9. Yolov8 uses Feature Pyramid Network (FPN) and Path Aggregation Network (PAN). These approaches also make the annotation process easier. Feature Pyramid Network is to reduce the spatial resolution of the input image with increased number. Input Image is an essential and first-coming parameter of Yolo Architecture [3].

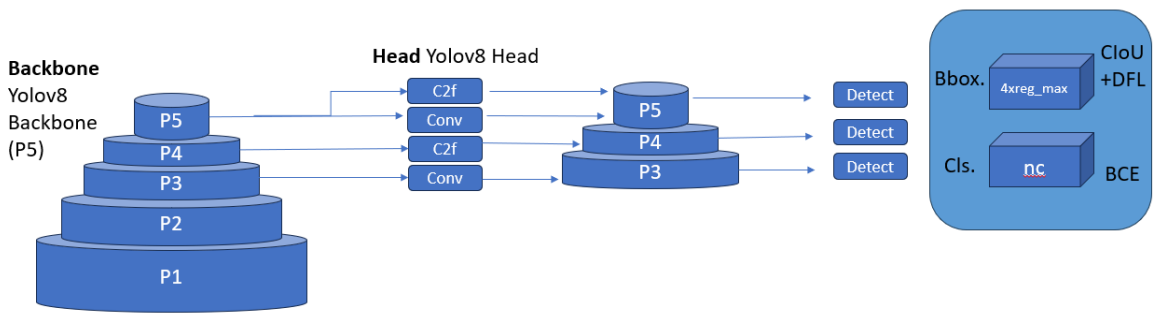


Figure 4.9. Yolov8 architecture, adapted from [3].

On the other hand, Path Aggregation Network system architecture utilizes different features from different layer levels throughout the network [30]. This method of

Yolov8 can capture small and tiny object detection performance which is quite useful for normalization, regularization and resolution challenge cases. We utilized Yolov8 since it performs effectively on resolution and size challenge frames. So that, following parameters have been used for model testing and parameter tuning:

- Input Image Size,
- Batch Size,
- Number of Epocs,
- Learning Rate,
- Anchor Boxes,
- Confidence Threshold.

Figure 4.10 illustrates the bounding box predictions made by our drone detection system, highlighting a specific instance where the confidence threshold is marked at 0.81. This level of confidence indicates a moderately reliable detection within this particular frame, underscoring the nuanced capabilities of the detection algorithm. It's critical to understand that this snapshot provides insight into the system's performance at a singular moment, rather than representing the overall accuracy and effectiveness of the detection mechanism. Comprehensive evaluation of the system's performance will be detailed in the experimental results section through the use of Precision-Recall and ROC (Receiver Operating Characteristic) curves. These statistical measures will offer a broader perspective on the system's capability to accurately identify drones under varying conditions, thereby providing a more complete picture of its operational efficacy.

4.4.2. Vision Transformers

Authors of [4] developed a method for multi-scale object detection using both words and objects using Vision Transformers. The official name of the approach is Open-Vocabulary Object Detection.

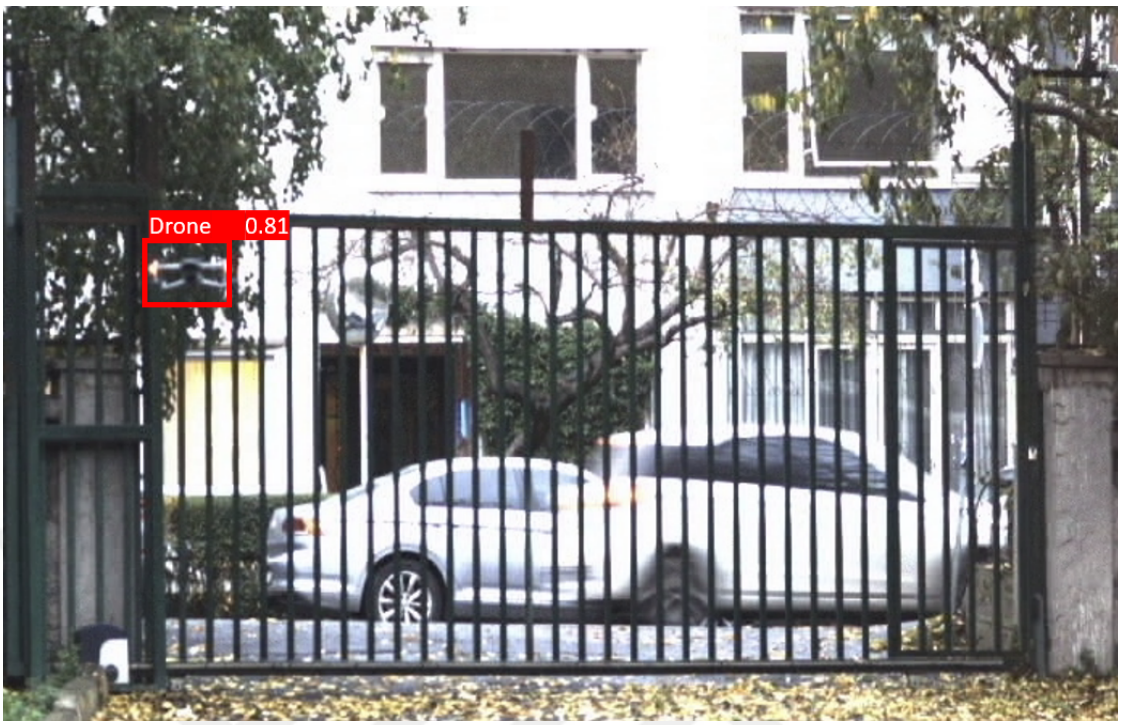


Figure 4.10. Example of predicted result with yolov8.

The model uses a standard computer vision transformer with image encoder and a text encoder. Token pooling and projection layer has been applied to obtain image embeddings classification. Within our approach, since we do not have image classification (only working with single object), we ignored this feature.

According to the ViT model, firstly, pretrained COCO model and text econder used and the output is an image-text pair.Then, it has been converted to an open-vocabulary object detection by removing token pooling to image encoder. With this approach, we used following text words to get object detection output. The word cluster that is used contains Drone, Dron, UAV, and Unmanned Aerial Vehicle.

In addition, detection output can be done any of four of them since it increases the probability of one-shot object detection. The hyper parameters used for this approach are follows:

- focal loss with $\alpha = 0.3$ and $\gamma = 2.0$,

- adam optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$,
- cosine learning rate,
- text encoder input length to 16 characters.

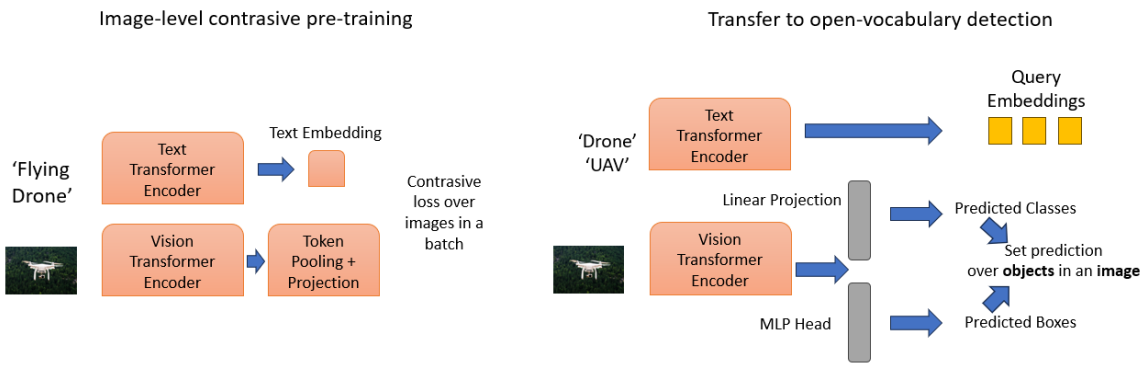


Figure 4.11. Vision transformer object detection architecture, adapted from [4].

Figure 4.12 shows a sample detection result of a random frame. However, although Figure 4.12 refers to a better results, it has also a different detection with a second bounding box coordinates as True Negative. The improvement of this part is based on a learning rate tuning approach.

4.4.3. LSTM (Long Short-Term Memory)

Long Short-Term Memory first developed by Hochreiter et al. [31]. They defined LSTM as truncating the gradient and LSTM can learn to bridge time laps minimally within special units. However, modern LSTM approaches with dropout layers, dense layers, and different kind of parameter features has been added over the time. Authors of [32] defines dropout layer as a regularization technique in neural networks to prevent the issue of overfitting.

Main idea is to drop random output features during training.Also, they define dense layer as a fully connected layer, where each input is connected to each output by a learnable weight [32].

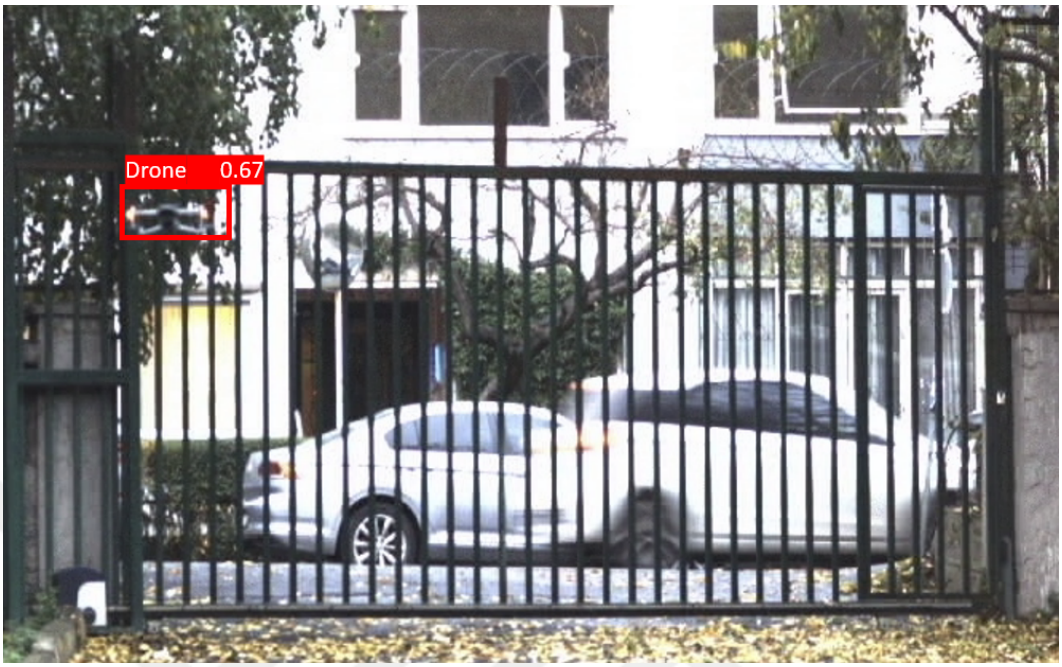


Figure 4.12. Example predicted result with vision transformer.

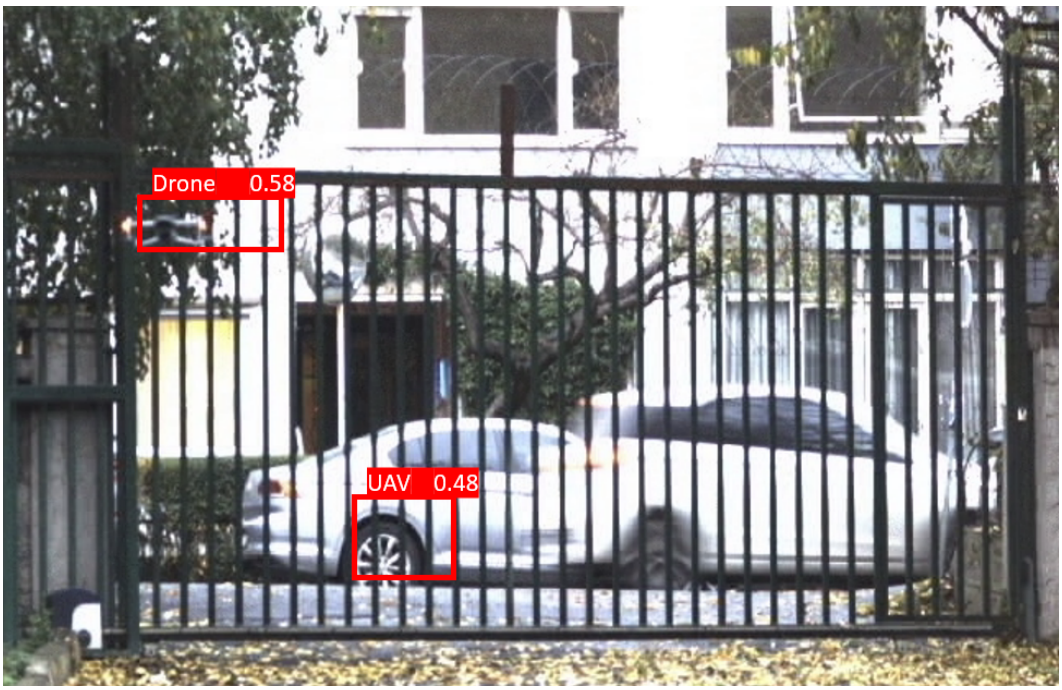


Figure 4.13. Example predicted result with vision transformer.

In addition, we can also observe that authors of [32] also state that given an input vector $\mathbf{x}$, the output y of a dense layer is computed as

$$y = \phi(\mathbf{W}\mathbf{x} + \mathbf{b}), \tag{4.13}$$

where $\mathbf{W}$ is the weight matrix, $\mathbf{b}$ is the bias vector, ϕ is the activation function.

Within this approach, the matrix multiplication $\mathbf{W}\mathbf{x}$ computes a weighted sum of the inputs, and then bias $\mathbf{b}$ is added.The mathematical approach of authors of [33] clearly explains the math behind dropout layers. Training phase is kept with neurons with a probability p or a dropped one with probability as $1 - p$. If a neuron's output is considered as x, and y as distributed in a neural network architecture after dropout layer (it could be improved by adding more than one), they are defined as

$$y = x \cdot r, \tag{4.14}$$

where r is a random variable from a Bernoulli distribution with probability p.vOur LSTM model for drone detection utilized following hyper parameters and values after tuning:

- Sequence length : 9,
- Number of layers : 5 , LSTM - Dropout - Dense - Dropout - Dense,
- Activation function: ReLu for LSTM and sigmoid for Dense layer,
- Optimizer : adam,
- Loss : binary crossentropy, item Hidden size: 256,
- Number of classes: 1,
- Learning rate: 0.0001,
- Batch size: 10,
- Number of epochs: 150.

It is recommended to closely monitor the model's performance due to the potential for overfitting given the depth of the network and the use of ReLu and sigmoid activations.

4.5. Architectures of Audio Modality Models

4.5.1. Convolutional Neural Networks (CNN)

As the authors of [34] stated, a Convolutional Neural Network (CNN) is a deep learning architecture that is utilized for analyzing image and audio datasets with convolution layers. It can be both used by video, sound and image classification. The architecture of a typical CNN consists of a sequence of layers and activation functions such as convolutional layer, activation layer, pooling layer, fully connected layer, and softmax layer. Thus, CNN architecture can perform differently with different combinations of layers.

CNN layer shape indicates total layers utilized and the input and output batch results according to filter values and kernel sizes. In terms of our CNN approach, five convolutional layers, flatten layer, and dense layer has been applied with ReLu activation function. Eventually, we concluded the model with a learning rate of 0.001 and a batch size of 128. In summary, our CNN architecture is characterized by a deliberate and strategic arrangement of layers and the selection of hyperparameters, all aimed at optimizing the model's performance for the task at hand by employing five convolutional layers followed by a flatten layer and a dense layer.

In summary, our described CNN architecture is systematically designed with a series of convolutional layers to extract features, followed by a flatten layer and a dense layer for classification. The selection of ReLU as the activation function, a learning rate of 0.001, and a batch size of 128 are strategic choices aimed at optimizing the model's performance for its intended task. A learning rate of 0.001 is chosen to ensure the model learns efficiently without overshooting the minimum of the loss function. Also, a batch size of 128 strikes a balance between training speed and model performance stability. The architecture leverages the depth of convolutional layers to process input data effectively, demonstrating a deliberate approach to constructing a CNN model that balances complexity with performance.

Table 4.1. CNN layers of our approach and input-output shape.

Layer	Input Shape	Filters	Kernel Size	Strides	Output Shape
Conv1	(batch, 60, 2, 2)	24	(6,2)	(1,1)	(batch, 60, 2, 24)
Conv2	(batch, 60, 2, 24)	24	(6,2)	(1,1)	(batch, 60, 2, 24)
Conv3	(batch, 60, 2, 24)	48	(5,1)	(2,2)	(batch, 30, 1, 48)
Conv4	(batch, 30, 1, 48)	48	(5,1)	(2,2)	(batch, 15, 1, 48)
Conv5	(batch, 15, 1, 48)	64	(4,1)	(2,2)	(batch, 8, 1, 64)
Flatten	(batch, 8, 1, 64)	–	–	–	(batch, 512)
Dense	(batch, 512)	–	–	–	(batch, 200)
Output - Dense	(batch, 200)	–	–	–	(batch, 5)

4.5.2. LSTM (Long Short-Term Memory)

Concerning audio samples, LSTM architecture also depends on an RNN (Recurrent Neural Network) schema and principles. According to Jeon et al., LSTM is used for audio binary classification (1 or 0) with replacing simple neurons with LSTM blocks [35]. This approach involves tanh input gate and output gate to compute the output activation. Finally, the output of the LSTM memory block is computed as a multiplication of these gates.

The hyperbolic tangent function referred as tanh function is given as

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}. \tag{4.15}$$

Authors of [32] state the final calculation with the derivative of tanh function (with sample variables and contants as input) which defined as

$$a = \tanh(z), \tag{4.16}$$

where $z = \mathbf{W}\mathbf{x} + b$, $\mathbf{W}$ is the weight matrix, $\mathbf{x}$ is the input vector to the neuron, and b is the bias term.

The final derivative of the `tanh` function, which is used in the backpropagation algorithm to calculate gradients, and is calculated as

$$\frac{d}{dx}\tanh(x) = 1 - \tanh^2(x). \tag{4.17}$$

Bi-directional LSTM-RNN (Recurrent neural network) with three layers and tanh function the best detection performance within this approach. Our LSTM model for audio classification and its hyperparameters are given as:

- Number of layers : 3,
- Activation function: Tanh ,
- Optimizer : adam,
- Loss : binary crossentropy,
- Number of classes : 1,
- Learning Rate 0.0005,
- Batch Size : 64,
- Number of Epochs : 100.

4.6. Fusion Strategies of Vision and Audio Modalities

After implementing the models described in previous section for vision and audio models, we used combined ground truth (OR-logic of ground truths) and the predicted output of the proposed detection method (OR-logic of OpticDNet-Yolo and Audio DNet) for performance evaluation. Also note that we analyzed two fusion strategies

for the vision models: Majority- and OR-logic. Therefore, we have two proposed methods: fusing Audio DNet with Optic DNet-Majority and fusing Audio DNet with Optic DNet-OR.

We evaluated a combined confusion matrix where the final decision of two modalities from ground truth and OpticDNet-Yolo and Audio DNet with a total of 16 different cases. Final results and more details are discussed and demonstrated in Section 5.5 and Section 5.6.

In addition to that, our final methodology for the ground truth part presents a method similar to a fusion principle. In terms of getting one simple ground truth with 1 or 0 for each frame, we utilized -OR logic- for first approach.

Table 4.2. OR logic of ground truth (GT) values.

Vision GT	Audio GT	Combined GT
1	1	1
1	0	1
0	1	1
0	0	0

First of all, an -OR Logic- approach has been applied for audio ground truth and optic ground truth. -OR Logic- takes a final result of each frame for audio ground truth and optic ground truth. Finally, it only gets False both ground truths' results are false, otherwise the final result become True.

Secondly, OpticDNet considers vision models according to the fusion strategy. If our frame-majority approach is used, fusion for Yolov8, vision transformer, and LSTM gets a final decision with majority method. The method results True if majority of detection decision for each frame is True. If the majority is False, the final decision also becomes false. If our frame-OR approach is used, fusion for Yolov8, vision transformer, and LSTM gets a final decision with OR-logic of model outputs. In terms of getting a

final combination result for audio modalities. Only CNN results are considered as final decision since it outweighs LSTM with regard to precision recall curve and ROC curve results.

In addition to OR-logic and Majority function, we propose another method defined as Vision-EGC (Equal Gain Combining). Han et al. [36] define equal gain combining as combining multiple received signals, or values with equal weights to improve over all quality. Although this technique is used in audio signal processing, for this work, it is used for both audio and vision modalities. Furthermore, as opposed to OR Logic and majority combining methods, equal gain combining method is defined as a soft combining method. The others could be considered as a hard combining method. The performance results and comparison of these methods will be discussed in results section. The equation for calculating the equal gain combining is defined as

$$\text{Probability Score } (\lambda_{thr}) = \alpha_1 c_{s1} + \alpha_2 c_{s2} + \alpha_3 c_{s3}, \tag{4.18}$$

where λ_{thr} is calculated confidence score value with the constants of α_1, α_2, and α_3 which are selected as 1/3 to get the equal gain. c_{s1}, c_{s2}, and c_{s3} are the confidence score values of models used in audio modality (CNN and LSTM) or vision modality (Yolov8, Vision Transformer, and LSTM). Apart from the the equal gain combining procedure, other methods are followed with the same approach. This method is applied for trained CONNECT-M dataset to compare the performance of the models.

5. EXPERIMENTAL RESULTS

5.1. Performance Metrics

As evaluation metrics, precision and recall are calculated for Precision-Recall Curve and True Positive Rates and False Positive Rates for ROC Curve. Precision is defined as the ratio between correctly detected objects and predicted positive observations [37]. Recall is the ratio between correctly predicted positive detection over total number of ground truth which is true positives and false negatives [37]. Precision is calculated as

$$\text{Precision} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Positives (FP)}}, \tag{5.1}$$

and recall is calculated as

$$\text{Recall} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Negatives (FN)}}, \tag{5.2}$$

where they also could be examined as a fraction with using TPs, FPs and FNs (false negatives).

However, when it comes to ROC curves, we calculate TPR and FPR as the ratio of True Positives and total ground truth of the cases where the true state is 1. Moreover, sensitivity is calculated with the same method as True Positive Rate. False Positive Rate can be calculated as the ratio of negatives that are wrongly detected or classified as positives to the total cases where the true state is 0. True Positive Rate is calculated as

$$\text{True Positive Rate (TPR)} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Negatives (FN)}}, \tag{5.3}$$

and False Positive Rate is calculated as

$$\text{False Positive Rate (FPR)} = \frac{\text{False Positives (FP)}}{\text{False Positives (FP)} + \text{True Negatives (TN)}}. \tag{5.4}$$

How precision, recall, TPR, and FPR are calculated in a step by step approach as follows: first of all, we create a tabular data with following columns; TP as 1 or

0, FP as 1 or 0, Accumulative TP (accumulative sum of total TP's until any frame n), Accumulative FP (accumulative sum of total FP's until any frame n). After that, general precision and recall and true positive rates and false positive rates are calculated for each row according to their unique formulas. As an example, a frame's precision result is calculated its TP value divied by cumulative sum of TP and FP. Within that approach, precision, recall, TPR, and FPR are calculated for each frame in the dataset as total 518 700 calculations for all four evaluation metrics. Finally, the results are visualized as Precision-recall curve and ROC Curve.

5.2. Performance of Vision Models on CONNECT Dataset

To get the results of optic modality, each frame in CONNECT Dataset is fed to a specified vision model (detector). If a drone image is detected in the frame according to the specified vision model, the detector outputs 1. If there is no detected drone image in the frame according to the specified vision model, the detector outputs 0. The findings demonstrates the comparison and classification of number of frames by ground truth and three different model results, which are YoloV8, Vision Transformers, and LSTM, respectively. For instance, there are $368\,442 + 82\,647 = 451\,089$ out of 518 700 (86.9%) frames in which the ground truth and all three models agree on output as 1 or 0. Error cases are shown with bold case for each model.

To quantify the performance and reliability of each detector, we constructed number results, which presents a comparative analysis of the classification outcomes against the ground truth for each frame. This ground truth serves as the benchmark for accuracy, representing the actual presence or absence of drone images in the frames.

Our analysis revealed that for a substantial portion of the dataset, specifically 451,089 out of the 518,700 frames (equivalent to an 86.9% agreement rate), there was a unanimous consensus between the ground truth and the predictions made by all three models, indicating a high level of accuracy in detecting drone images or confirming their absence. Through this detailed comparison and classification, we aim to provide

a nuanced understanding of how each model performs in the context of drone detection, shedding light on their respective strengths and weaknesses in processing optic modalities within the CONNECT Dataset.

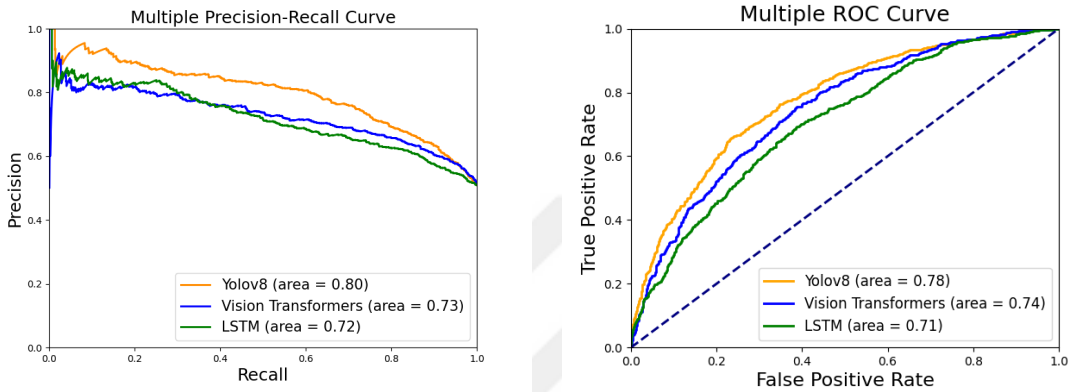
Table 5.1. Ground truth (GT) and vision models on CONNECT dataset.

GT	**YOLOV8**	**Vision Transformer**	**LSTM**	**Num. of Frames**
1	1	1	1	368 442
1	1	1	**0**	13 461
1	1	**0**	1	4256
1	1	**0**	**0**	18 725
1	**0**	1	1	2002
1	**0**	1	**0**	504
1	**0**	**0**	1	11
1	**0**	**0**	**0**	28 607
			Total (GT=1):	436 008
0	**1**	**1**	**1**	0
0	**1**	**1**	0	23
0	**1**	0	**1**	0
0	**1**	0	0	0
0	0	**1**	**1**	18
0	0	**1**	0	4
0	0	0	**1**	0
0	0	0	0	82 647
			Total (GT=0):	82 692
			Total (All):	518 700

5.2.1. Precision-Recall and ROC Curves

We evaluated three different vision models on CONNECT image dataset. The figures show that area under the ROC Curve and area under the Precision Recall curve

respectively. According to the results with True Positive Rates and False Positive Rates, Yolov8, Vision Transformers and LSTM have total area under the curve of 0.78, 0.75, and 0.71 respectively. In addition, with regard to Precision and Recall, the areas are calculated as 0.80, 0.73, and 0.72.



(a) Precision recall curve.

(b) ROC curve.

Figure 5.1. Precision recall and ROC curves under no additional darkness.

5.3. Effect of Darkness on Vision Models

As we mentioned earlier, an image can be represented as a matrix with pixel values on a 3D matrix value with red, green, and blue color components. In matrix representation, (x, y) represents the coordinate of each pixel with two different variables in the matrix [38]. We processed the images in the dataset to create darker images for mimicking the night scenarios. At this point each level will be shown one by one for a different visualization.

To darken the frames, the intensity has been reduced and can be calculated as

$$I_{\text{dark}}(x, y) = \max(I(x, y) - C, 0), \tag{5.5}$$

where $I_{\text{dark}}(x, y)$ is the original intensity and C is constant for to reduce the intensity. Since we have coloured images, we need to apply this operation three times with RGB

modules and R module is calculated as

$$R_{\text{dark}}(x, y) = \max(R(x, y) - C, 0), \tag{5.6}$$

G module is calculated as

$$R_{\text{dark}}(x, y) = \max(R(x, y) - C, 0), \tag{5.7}$$

and B module is calculated as

$$R_{\text{dark}}(x, y) = \max(R(x, y) - C, 0), \tag{5.8}$$

where R, G, and B represents red, green, and blue components of each pixels. This is a uniform darkening effect to be used in image processing [38].



(a) 20 % darkness.

(b) 40 % darkness.

(c) 60 % darkness.

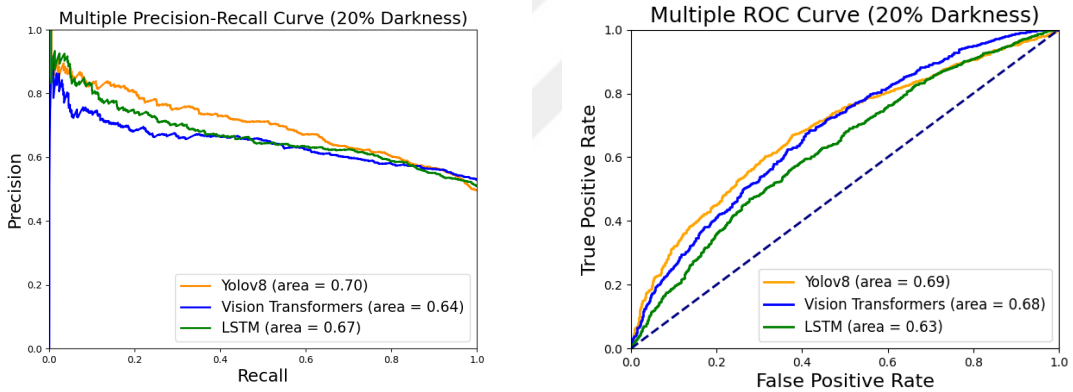
(d) 80 % darkness.

Figure 5.2. Four different level of darkness.

5.3.1. Performance Under 20 % Darkness

20% darkness figures indicate ROC Curves and Precision-Recall Curve of the following models with 20 % darkness added. Area under the curve for Yolov8, Vision Transformer, and LSTM are 0.69, 0.68, and 0.63, respectively.

With regard to normal conditions (no darkness scenario), we observe that there is a performance reduction for all three models. Similarly, In the precisio-recall curve figure, precision and recall curve of Yolov8, Vision Transformer, and LSTM show similar approach in terms of area under the curve. However, the performance of vision transformer is more closer to Yolov8 this time.



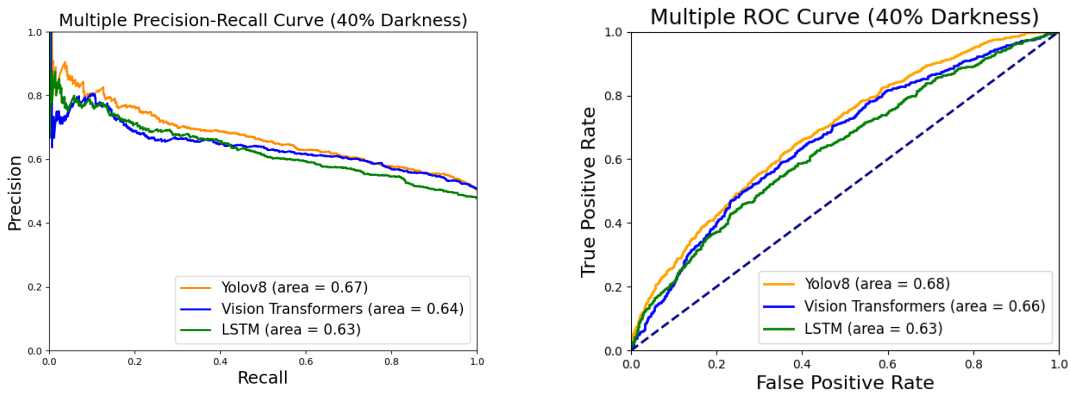
(a) Precision recall curve.

(b) ROC curve.

Figure 5.3. Precision recall and ROC curves under 20 % darkness.

5.3.2. Performance Under 40 % Darkness

40% darkness figure indicates ROC Curve and Precision-Recall Curves of the following models with 40 % darkness added. As an instance The C value in Equation (5.6) represents the darkness value to reduce the intensity of each frame. The areas under the curve are 0.68, 0.66 and 0.64 respectively. Unlike LSTM, Yolov8, and Transformers experienced a slight reduce with regard to area under the curve.



(a) Precision recall curve.

(b) ROC curve.

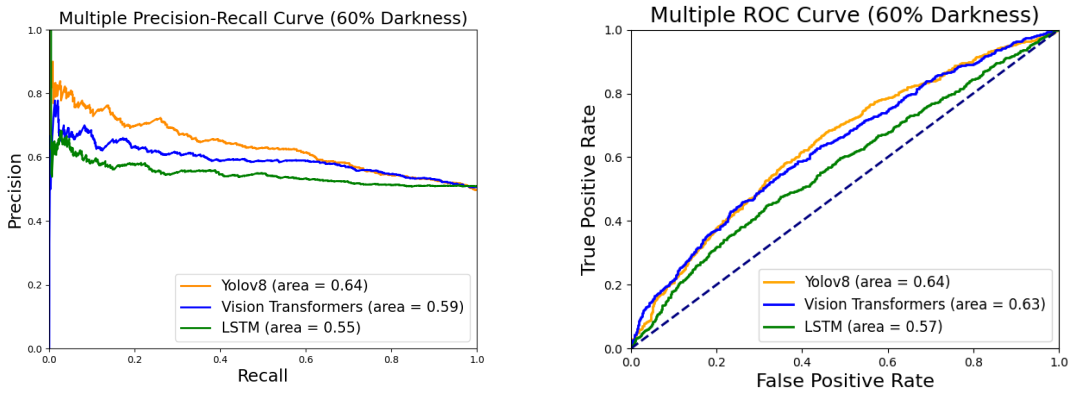
Figure 5.4. Precision recall and ROC curves under 40 % darkness.

5.3.3. Performance Under 60 % Darkness

60% percent figures represent area under the curves calculated with True Positive Rate and False Positive Rates are 0.64, 0.63 and 0.57 for Yolov8, Vision Transformers, and LSTM, respectively. In addition, the constant values C in Equation (5.5) and Equation (5.6) are determined as 0.6. It can be observed that LSTM showed steadier reduction as compared to Yolov8 and Transformers after experiencing 20 % more darkness. In terms of precision and recall values, the figure demonstrates that Vision Transformers showed a worse approach as compared to true positive rates and false positive rates which will quantify the parameters affection to a ViT model effectively.

5.3.4. Performance Under 80 % Darkness

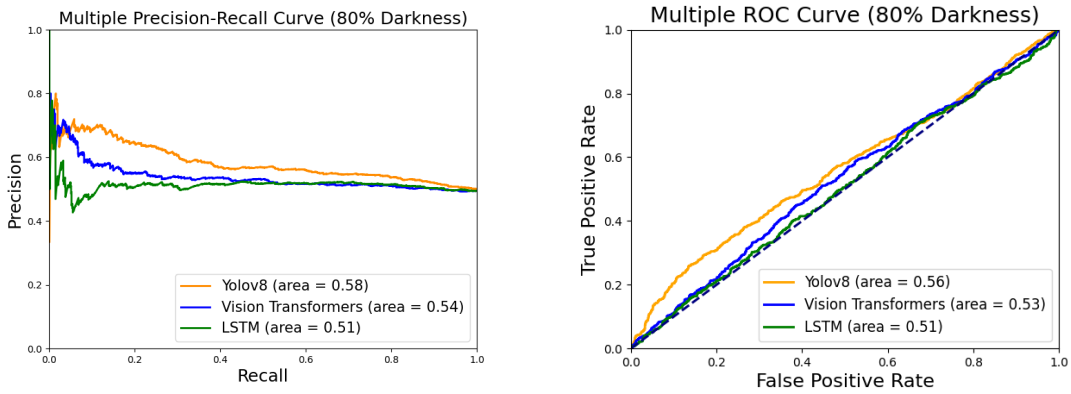
At this level of darkness , constant C value in Equation (5.5) is 0.8. Furthermore, ROC Curve results in 80% percent figure clearly shows that there is a slight decrease for all three deep learning models. Area under the curve for Yolov8, Vision Transformers and LSTM are 0.56, 0.53, and 0.51 respectively. Furthermore, AUC of three models in Precision Recall Curve are 0.58, 0.54, and 0.51 respectively.



(a) Precision recall curve. (b) ROC curve.

Figure 5.5. Precision cecall and ROC curves under 60 % darkness.

This results show that the more the darkness added, the worse the performance of true positive rates are getting. With the added darkness constant C equals 0.8, we can observe that detection results are more approaching to randomized model results. In conclusion with darkness, we can say that the more darkness added, the more performence of AUC decreases for three models with the exception of LSTM at certain constant C values. For the darkest samples, decreases of performance metrics are more observable.



(a) Precision recall curve. (b) ROC curve.

Figure 5.6. Precision recall and ROC curves under 80 % darkness.

5.4. Performance of Audio Models

To be able to compare results of vision modality, ROC Curve method has been used to clarify the performances of audio detection results. There are two methods used to clarify and compare the results and metrics of audio dataset.

According to Equation (5.3) and Equation (5.4), true positive rates and false positive rates has been applied for CNN and LSTM models. As mentioned, Audio dataset also contains 518 700 different time frames to be synchronized with image dataset.

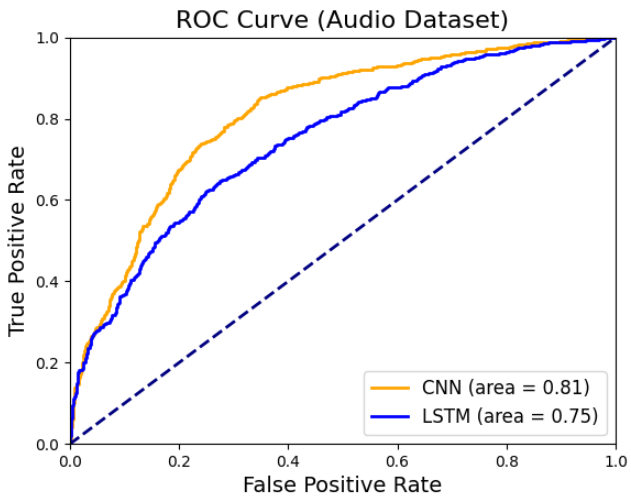


Figure 5.7. ROC curve of CNN and LSTM.

Figure 5.7 demonstrates the ROC Curve of two different models used in our unique dataset. According to True Positive Rate and False Positive Rate values, Area under the curve is 0.81 for CNN model structure and 0.75 for LSTM model structure. Unlike vision part, there are no further arrangement such as changing sound level to obtain ROC and precision-recall values and curves.

5.5. Comparison of Two Modalities on CONNECT Dataset

To obtain the insight from results, we need the compare the likelihood of presence of each frame.The probability scores in CONNECT image datasets is derived from the analysis of the identification of objects within a visual frame, while audio data is temporal and sequential. Therefore, this necessities makes algorithmic approaches interpretable for probability score calculation and comparison for each modality.

Table 5.2. Confusion matrix for audio ground truth (GT) and audio DNet.

		Audio DNet	
		0	1
Audio GT	0	0.461	0.539
	1	0.075	0.924

After getting the results of different models from two modalities, number of frames has been classified and used for evaluating the confusion matrix. Table 5.2 shows the probability scores of detection for Audio dataset ground truth and our audio model DNet (At this point, CNN has been applied). For example, 46.1 % of data of ground truth 0 (no detection) has also been detected negatively in the DNet model. Also, 53.9 % of the data of ground truth 0 (no detection) has been detected as 1 (true) in DNet model. For the detection made part of Audio Ground Truth which is 1, 7.5 % of the data has been detected as false and the remaning part which is 92.4 % has been detected as true with the same result of audio ground truth.

According to the GT vs DNet Table , the table shows the number of frames Audio Ground Truth and Audio DNet. Same values has been showed with real frame values in this table. For example, Total number of 119301 frames that ground truth detected as 0, 54983 of them has also been detected as zero by Audio DNet model and the remaining 64318 frame detected as true.

Table 5.3. Number of frames for audio ground truth (GT) and audio DNet.

		Audio DNet	
		0	1
Audio GT	0	54 983	64 318
	1	30 085	369 314

Moreover, frames and probabilities of GT vs OpticDNet show the related results with Optic Ground Truth and Our Optic DNet (Yolov8 has been chosen for this approach). Let's define it as OpticDNet-Yolo. As an instance, the table shows that 0.07 % of all true detections (1) has been detected false by OpticDNet-Yolo and remaining 0.97 part has also been detected true. Similarly, in Table 5.5, it is showed that the number of classified frames as a numeric value.

Table 5.4. Confusion matrix for optic ground truth (GT) and opticDNet-yolo.

		OpticDNet-Yolo	
		0	1
Optic GT	0	0.99	0.01
	1	0.07	0.93

Table 5.5. Number of frames for optic ground truth (GT) and opticDNet-yolo.

		Optic DNet-Yolo	
		0	1
Optic GT	0	82 669	23
	1	31 124	404 884

Fusion table indicates the number of frames according to a fusion of Table 5.5 and Table 5.3. For example, 43282 frame out of total frames are detected as negative for OpticDNet-Yolo model - AudioDNet model where the ground truth is also negative for both modalities.

Table 5.6. Multi-class confusion matrix for opticDNet-yolo and audioDNet on CONNECT dataset.

(Optic GT, Audio GT)	(OpticDNet-Yolo , AudioDNet)			
	00	01	10	11
00	43 282	0	1341	0
01	6203	67 302	0	31 495
10	3813	0	6013	0
11	0	92 308	23 406	178 407

5.6. Performance of Fusing AudioDNet and OpticDNet

To evaluate the fusion strategy, we first need to focus on the ground truth for both modalities as it is mentioned in Section 4.6. Then we will focus on the fusion strategy and its performance that is evaluated by using fused ground truth. Since the miss-detection (False Negative Rate) is critical in these kind of system, we have chosen -OR Logic- for fusing the ground truth values. Table fused ground truth statistics shows that the total 484 915 frames are chosen as true (1) in the fused dataset out of 518 700 frames of the CONNECT dataset, and density column as 0.936.

Table 5.7. Fused ground truth (GT) statistics.

	Num. of Frames	Density (Probability)
Fused GT=1	484 915	0.936
Fused GT=0	33 715	0.064
Total	518 700	1

Our fusion strategy is summarized in fusion method architecture. We are fusing AudioDNet and OpticDNet-Yolo outputs. For the audio modality, CNN's results has been considered as final model result.For the optic modality, majority method by Budak et al. [39] has been applied among three models and call it OpticDNet-Majority.

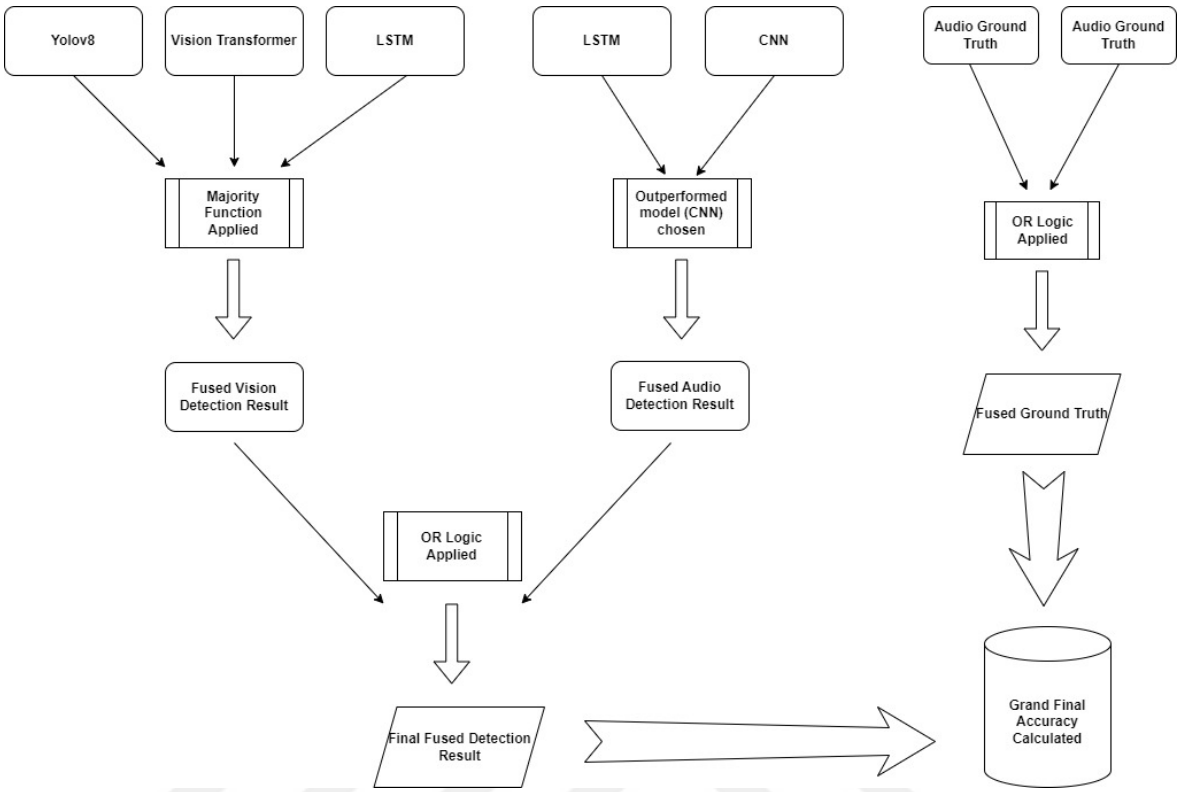


Figure 5.8. Fusion method architecture.

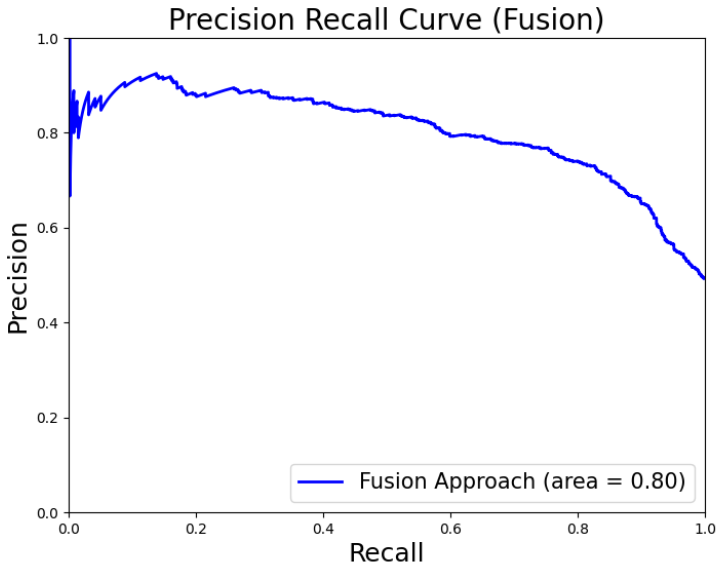
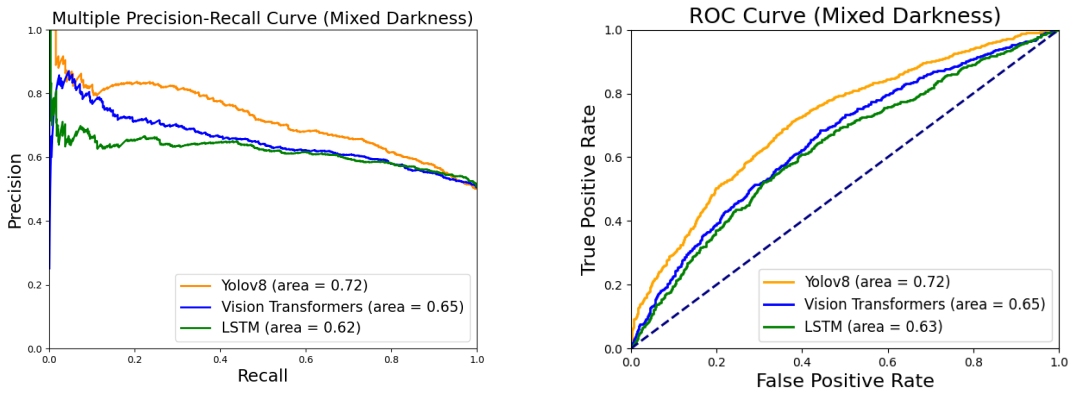


Figure 5.9. Precision recall curve result of fusion method.

5.7. Performance of Vision Models on CONNECT-M Dataset

According to fusion results, it shows that the precision recall result of final fusion approach reached Yolov8 result of CONNECT dataset. CONNECT-M dataset is a more challenging dataset compared to CONNECT dataset since it includes darkened images for some of the frames to simulate night conditions. Please recall that, 30% of the data is 20% darkened, 30 % of the data is 40% darkened, 20 % of the data is 60% darkened, and the remaining ones are 80% darkened. According to CONNECT-M results, various types of darkness levels affect the performance of the models. In addition the model performance might decrease at some points where the algorithm sees new image frame with new level of darkness other than training set. To mention more detail, while there is a frame a darkness added image chosen from a random video, another image with same darkness level is chosen from another subset (videos) of the dataset in order to handle issues such as overfitting.

However, to compare the improvement, we also trained the models on both CONNECT and CONNECT-M datasets. Firstly, we applied testing set of CONNECT-M to a CONNECT trained dataset. After that, the results of CONNECT-M training and CONNECT-M testing is demonstrated.

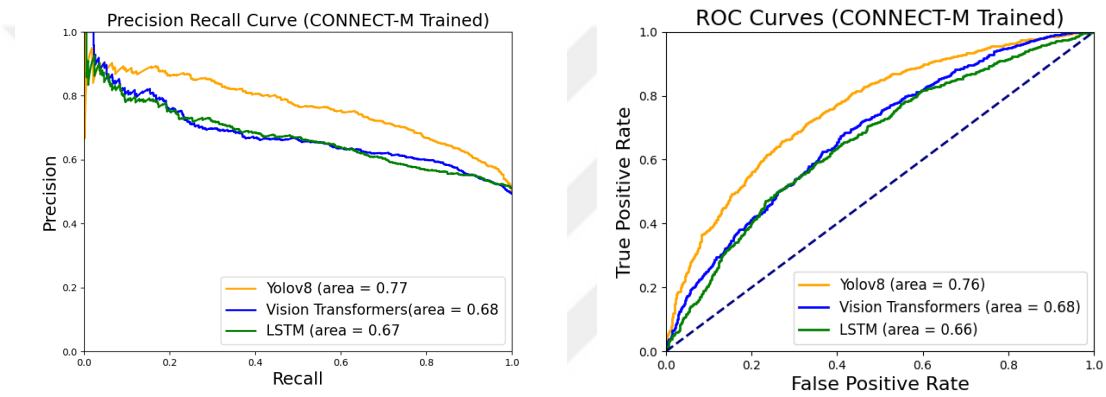


(a) Precision recall curve.

(b) ROC curve.

Figure 5.10. Precision recall and ROC curves on CONNECT-M dataset.

As it is explained in section 4.6, we applied three different models to a different dataset that different darkness levels added. As the main approach, train data is separated from the validation and test dataset which will present a different part of a video set than the test set. However, these results shows the preliminary results of only test data is used for CONNECT-M dataset. To be able to improve our results. We trained ours vision models (Yolov8, Vision Transformer, and LSTM) with CONNECT-M dataset too. Both training and test samples is chosen from CONNECT-M dataset.



(a) Precision recall curve.

(b) ROC curve.

Figure 5.11. Precision recall and ROC curves (CONNECT-M trained).

Table 5.11 shows precision-recall and roc curve results of three different vision models on CONNECT-M dataset trained. As compared to basic approach, we can clearly observe that there is a significant increase for Yolov8 and Vision Transformers. However, the performance increase of LSTM is less than other models. We can also point out that all models have an improvement when CONNECT-M train model comes into effect. Similar to performance table of CONNECT Dataset, Table 5.8 indicates that performance of vision models for CONNECT-M dataset. As an instance according to that table, there are total of 44 342 frames that Ground Truth and Yolov8 gives a True result and vision transformer and LSTM gives negative result for drone detection. Lastly, in addition to the methods in Figure 5.8, Vision-EGC is applied to obtain fused vision detection result on CONNECT-M dataset with threshold.

Fusion method comparison results show the final fusion result with true positive rate and false positive rated on CONNECT-M dataset and CONNECT dataset using equal gain combining method and basic fusion approach (OR Logic). It can be observed that for ground truth instances labeled as 0 (indicating the feature or object of interest is not present) no models falsely detected the feature in 82,613 frames. In some cases, one or two models incorrectly detected the feature when it was not present according to ground truth, as indicated by the numbers in respective cells.

Table 5.8. Ground truth (GT) and vision models on CONNECT-M dataset.

GT	**YOLOV8**	**Vision Transformer**	**LSTM**	**Num. of Frames**
1	1	1	1	276 004
1	1	1	**0**	41 761
1	1	**0**	1	10 264
1	1	**0**	**0**	44 342
1	**0**	1	1	473
1	**0**	1	**0**	712
1	**0**	**0**	1	6
1	**0**	**0**	**0**	62 446
			Total (GT=1):	436 008
0	**1**	**1**	**1**	0
0	**1**	**1**	0	46
0	**1**	0	**1**	8
0	**1**	0	0	0
0	0	**1**	**1**	0
0	0	**1**	0	25
0	0	0	**1**	0
0	0	0	0	82 613
			Total (GT=0):	82 692
			Total (All):	518 700

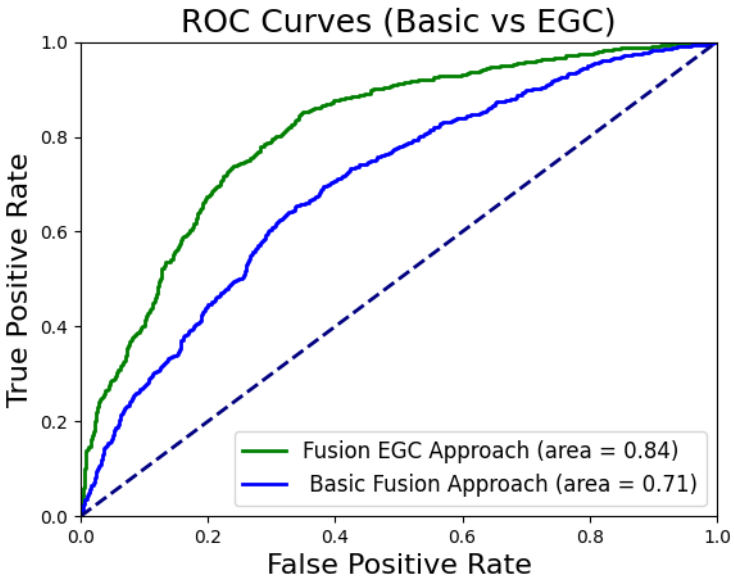


Figure 5.12. ROC curve of fusion with EGC method and basic method.

After getting the insights and values from binary results of different models, we can observe that the Fusion EGC Approach, represented by the solid green line, has an AUC of 0.84, indicating a good classification performance. It is closer to the top left corner, implying a higher true positive rate and a lower false positive rate.

However, if there is a significant difference between models in terms of confidence scores, the method will consider it also equally. Main advantage of equal gain combining is that all model performance can be benefited equally to obtain a certain threshold for detection. The development approach is based on same hyper parameter methods with different train set. At the end, increase of area under the curve can be interpreted and observed including train set difference of CONNECT and CONNECT-M.

6. DISCUSSIONS

6.1. Remarks

One of the critical remarks that we need to point out is that we can observe some of the models of deep learning methods might show less performance as compared to previous darkness level. For example, LSTM has less performed under 20% darkness than 40 % darkness. While area under the curves are 0.67 and 0.63 for CONNECT-D20 and CONNECT-D40 datasets respectively. This might lead to a discussion where somehow different parameter tuning methods of different model might be affected even the pixels of the frames. However, we observe that LSTM performed almost same when it comes to the ROC curve, it performed 0.63 in both 20 % of 40 % of darkness. Since recall and true positive rate are defined as the same values, false positive rate calculation might affect the area under the curve at a certain point.

Furthermore, after applying -OR logic- to the vision and audio ground truths, we observe total number of frames detected as true has increased significantly within the mathematical result of the logic. Also, we can make an inference on final fusion result of optic and audio modalities. During our experiments, we collected some videos with no drone flying, therefore we know that approximately 13 % of our total data must be detected as false. Final approach also showed this method reflect the increase of probability of detection since we got 24.3 % of total data as false and 75.7 % of total data as true.

In terms of comparison of -OR logic- and majority logic, we claimed that –OR Logic- performs better since it gets three out of four frames as True if any of them gets the drone detection as True. However, majority approach will consider the detection as false even if one outperformed approach significantly detected it as True as opposed to the other two optic modality decisions.

7. CONCLUSION

In this work, a comprehensive model performance comparison and fusion with darkness addition has been demonstrated with object detection capabilities using audio and optical modalities. The research integrated convolutional neural networks and long short-term memory (LSTM) deep learning methods to process and analyze audio datasets. In addition, Yolov8, vision transformers, and long short-term memory (LSTM) has been applied for image drone detection. The experimental results show that there are insights into the strengths and limitations when adding darkness for each modality. In addition, a novel approach which contains two aspects was proposed and evaluated. Firstly, we observed how getting one final result for ground truth, audio and optic modalities affect the performance of precision recall and roc curves. At the end, although various types of vision models are under the area under curve of 0.80, we observed 0.80 with the fusion approach of modalities and ground truths. Secondly, the thesis also shed light on the impact of environmental factors such as adding darkness on detection performance using multi-modal systems in real-world experimental applications. The last approach we conducted showed that although there are models with 60 and 80 % added darkness are below 0.6 accuracy in precision recall and roc curve results, we observed a score above 0.6 for each model with the total 40 % of data under 60 or more darkness.

In the future work, we could explore additional data such as infrared or thermal imaging. To improve detection performance in a more efficient way, we might conduct using a third modality which is radio frequency. The research not only conducted in this thesis but only contributes other works under CONNECT project funded by Computer Engineering Department of Boğaziçi University. Lastly, for each data frame of optic modality, we used single drone for detection. Multiple or swarm drone detection methods with latest further deep learning approaches may be conducted for the future work as a result of intelligence detection, surveillance, and monitoring systems.

REFERENCES

1. Saeed, L. and M. Spagat, "The Impact of Drone Warfare on Suicide Bombings in Pakistan", *Royal Holloway University of London Department of Economics*, Vol. 27, p. 2021, 2021.

2. Salman, S., J. Mir, M. T. Farooq, A. N. Malik and R. Haleemdeen, "Machine Learning Inspired Efficient Audio Drone Detection Using Acoustic Features", *International Bhurban Conference on Applied Sciences and Technologies (IBCAST)*, pp. 335–339, Islamabad, 2021.

3. Reis, D., J. Kupec, J. Hong and A. Daoudi, "Real-Time Flying Object Detection with YOLOv8", ArXiv:2305.09972, 2023.

4. Minderer, M., A. Gritsenko, A. Stone, M. Neumann, D. Weissenborn, A. Dosovitskiy, A. Mahendran, A. Arnab, M. Dehghani, Z. Shen, X. Wang, X. Zhai, T. Kipf and N. Houlsby, "Simple Open-Vocabulary Object Detection with Vision Transformers", ArXiv, 2022.

5. Moustafa, N., J. Slay and G. Creech, "Novel Geometric Area Analysis Technique for Anomaly Detection Using Trapezoidal Area Estimation on Large-Scale Networks", *IEEE Transactions on Big Data*, Vol. 5, No. 4, pp. 481–494, 2017.

6. Moustafa, N. and A. Jolfaei, "Autonomous Detection of Malicious Events Using Machine Learning Models in Drone Networks", *Proceedings of the 2nd ACM MobiCom Workshop on Drone Assisted Wireless Communications for 5G and Beyond*, pp. 61–66, London, 2020.

7. Misbah, M., M. U. Khan, Z. Yang and Z. Kaleem, "Tf-Net: Deep Learning Empowered Tiny Feature Network for Night-Time UAV Detection", *International Conference on Wireless and Satellite Systems*, pp. 3–18, Barcelona, 2023.

8. Ding, S., X. Guo, T. Peng, X. Huang and X. Hong, "Drone Detection and Tracking System Based on Fused Acoustical and Optical Approaches", *Advanced Intelligent Systems*, Vol. 5, No. 10, p. 2300251, 2023.

9. Svanström, F., C. Englund and F. Alonso-Fernandez, "Real-Time Drone Detection and Tracking with Visible, Thermal and Acoustic Sensors", *25th International Conference on Pattern Recognition (ICPR)*, pp. 7265–7272, Milan, 2021.

10. Kim, J., D. Lee, Y. Kim, H. Shin, Y. Heo, Y. Wang and E. T. Matson, "Deep Learning Based Malicious Drone Detection Using Acoustic and Image Data", *Sixth IEEE International Conference on Robotic Computing (IRC)*, pp. 91–92, California, 2022.

11. Liu, H., Z. Wei, Y. Chen, J. Pan, L. Lin and Y. Ren, "Drone Detection Based on An Audio-Assisted Camera Array", *Third International Conference on Multimedia Big Data (BigMM)*, pp. 402–406, California, 2017.

12. Wang, G., Y. Chen, P. An, H. Hong, J. Hu and T. Huang, "UAV-YOLOv8: A Small-Object-Detection Model Based on Improved YOLOv8 for UAV Aerial Photography Scenarios", *Sensors*, Vol. 23, No. 16, p. 7190, 2023.

13. Khan, M. U., M. Misbah, Z. Kaleem, Y. Deng and A. Jamalipour, "GAANet: Ghost Auto Anchor Network for Detecting Varying Size Drones in Dark", ArXiv:2305.03425 [cs], 2023.

14. Munir, A., A. J. Siddiqui and S. Anwar, "Investigation of UAV Detection in Images With Complex Backgrounds and Rainy Artifacts", *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 221–230, Hawaii, 2024.

15. Casabianca, P. and Y. Zhang, "Acoustic-Based UAV Detection Using Late Fusion of Deep Neural Networks", *Drones*, Vol. 5, No. 3, p. 54, 2021.

16. Phung, K. P., T. H. Lu, T. T. Nguyen, N.-L. Le, H.-H. Nguyen and V. P. Hoang, "Multi-Model Deep Learning Drone Detection and Tracking in Complex Background Conditions", *2021 International Conference on Advanced Technologies for Communications (ATC)*, pp. 189–194, Ho Chi Minh City, 2021.

17. Lin, T.-Y., M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár and C. L. Zitnick, "Microsoft COCO: Common Objects in Context", *Computer Vision–ECCV: 13th European Conference, Proceedings, Part V 13*, pp. 740–755, Zurich, 2014.

18. Zhang, Z., X. Lu, G. Cao, Y. Yang, L. Jiao and F. Liu, "ViT-YOLO: Transformer-Based YOLO for Object Detection", *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2799–2808, Montreal, 2021.

19. Dosovitskiy, A., L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale", ArXiv preprint arXiv:2010.11929, 2020.

20. Jamil, S., M. S. Abbas and A. M. Roy, "Distinguishing Malicious Drones Using Vision Transformer", *AI*, Vol. 3, No. 2, pp. 260–273, 2022.

21. Zhao, X., Y. Xia, W. Zhang, C. Zheng and Z. Zhang, "YOLO-ViT-Based Method for Unmanned Aerial Vehicle Infrared Vehicle Target Detection", *Remote Sensing*, Vol. 15, No. 15, p. 3778, 2023.

22. Yuan, D., X. Shu, N. Fan, X. Chang, Q. Liu and Z. He, "Accurate Bounding-Box Regression with Distance-IoU Loss for Visual Tracking", *Journal of Visual Communication and Image Representation*, Vol. 83, p. 103428, 2022.

23. Rezatofighi, H., N. Tsoi, J. Gwak, A. Sadeghian, I. Reid and S. Savarese, "Generalized Intersection Over Union: A Metric and A Loss for Bounding Box Regression",

Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 658–666, California, 2019.

24. McFee, B., C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg and O. Nieto, "Librosa: Audio and Music Signal Analysis in Python", *Proceedings of the 14th Python in Science Conference*, Vol. 8, pp. 18–25, Austin, Texas, 2015.

25. Ioffe, S. and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift", *International Conference on Machine Learning*, pp. 448–456, Lille, 2015.

26. Gong, Y., Y.-A. Chung and J. Glass, "Ast: Audio Spectrogram Transformer", ArXiv Preprint ArXiv:2104.01778, 2021.

27. Yosinski, J., J. Clune, Y. Bengio and H. Lipson, "How Transferable Are Features in Deep Neural Networks?", *Advances in Neural Information Processing Systems*, Vol. 27, pp. 1–9, 2014.

28. Pan, S. J. and Q. Yang, "A Survey on Transfer Learning", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 22, No. 10, pp. 1345–1359, 2009.

29. Kenton, J. D. M.-W. C. and L. K. Toutanova, "Bert: Pre-Training of Deep Bidirectional Transformers for Language Understanding", *Proceedings of NaacL-HLT*, Vol. 1, p. 2, 2019.

30. Li, Y., Q. Fan, H. Huang, Z. Han and Q. Gu, "A Modified YOLOv8 Detection Network for UAV Aerial Image Recognition", *Drones*, Vol. 7, No. 5, p. 304, 2023.

31. Hochreiter, S. and J. Schmidhuber, "Long Short-Term Memory", *Neural Computation*, Vol. 9, No. 8, pp. 1735–1780, 1997.

32. Goodfellow, I., Y. Bengio and A. Courville, *Deep Learning*, MIT Press, Cambridge, Massachusetts, 2016.

33. Nitish, S., "Dropout: A Simple Way to Prevent Neural Networks from Overfitting", *Journal of Machine Learning Research*, Vol. 15, p. 1, 2014.

34. LeCun, Y., Y. Bengio and G. Hinton, "Deep Learning", *Journal of Biomedical and Health Informatics*, Vol. 521, No. 7553, pp. 436–444, 2015.

35. Jeon, S., J.-W. Shin, Y.-J. Lee, W.-H. Kim, Y. Kwon and H.-Y. Yang, "Empirical Study of Drone Sound Detection in Real-Life Environment with Deep Neural Networks", *25th European Signal Processing Conference (EUSIPCO)*, pp. 1858–1862, Kos, Greece, 2017.

36. Han, J., X. Shan, H. Liu, J. Xiao and T. Huang, "Fuzzy Gain Scheduling PID Control of a Hybrid Robot Based on Dynamic Characteristics", *Mechanism and Machine Theory*, Vol. 184, p. 105283, 2023.

37. Powers, D. M., "Evaluation: from Precision, Recall and F-Measure to ROC, Informedness, Markedness and Correlation", ArXiv Preprint arXiv:2010.16061, 2020.

38. Szeliski, R., *Computer Vision: Algorithms and Applications*, Springer Nature, New York, 2022.

39. Budak, Ü., A. Şengür, A. B. Dabak and M. Cibuk, "Transfer Learning Based Object Detection and Effect of Majority Voting on Classification Performance", *International Artificial Intelligence and Data Processing Symposium (IDAP)*, pp. 1–4, Malatya, Turkey, 2019.

40. Müller, T. and B. Erdnüß, "Robust Drone Detection with Static VIS and SWIR Cameras for Day and Night Counter-UAV", *Counterterrorism, Crime Fighting, Forensics, and Surveillance Technologies III*, Vol. 11166, pp. 58–72, 2019.