Offline Signature Verification System

A BTP Report by

Tulluri Rakesh

Thorlikonda Sai Gopi Nadh

Veduparthi Sai Bhaskar

Roll Nos.:

S20190010182

S20190010180

S20190010188

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY SRI CITY

Date: 07-12-2022

Final Report

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY SRICITY

CANDIDATE'S DECLARATION

I hereby certify that the work which is being presented in the BTP entitled "*Offline Signature Verification*" in the partial fulfillment of the requirements for the award of the degree of B. Tech and submitted in the Indian Institute of Information Technology SriCity, is an authentic record of my own work carried out during the time period from January 2022 to December 2022 under the supervision of Prof. Vishwanath, Indian Institute of Information Technology SriCity, India.

The matter presented in this report has not been submitted by me for the award of any other degree of this or any other institute.

Signature of the student with date

Tulluri Rakesh, 7/12/2022
Thorlikonda Sai Gopi Nadh, 7/12/2022
Veduparthi Sai Bhaskar, 7/12/2022

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Signature of BTP Supervisor with date

(Prof. XYZ)

Offline Signature Verification System

ABSTRACT

In the era of growing technology, security is the major concern to avoid fake and forgeries. Signatures are also one of the most easily forgeable biometric identities when compared to other biometric features like thumb impression, face recognition etc. Nowadays, Signature verification is one of the most important features for checking the authenticity of a person.

Signature verification is the most generally used biometric to maintain human privacy. It is used in many areas as banking, access control, e-business etc. and equally important in financial transactions. Research has progressed greatly in the area of signature verification but still, it is hard to discriminate between genuine signatures and skilled forgeries.

So, there is a need to build an efficient verification system for offline signatures. In this report we have discussed several machine learning and deep learning techniques that are used for offline signature verification. The performance of these techniques are measured using three measures namely accuracy, false acceptance rate and false rejection rate. These performances are obtained from the CEDAR dataset.

Contents

1.INTRODUCTION

Signature verification system has two major categories as online and offline based on data acquisition techniques. The offline verification approach is often referred to as a static approach and online verification as a dynamic one. The latter method measures chronological information such as speed and pressure of a pen using a particular tool like a stylus or tablet. It has dynamic information which is captured in real-time as pen-tip coordinates, pressure etc. Offline method deals with static signature images which are obtained with an optical scanner by scanning a signature on paper. The verification system is more challenging due to the lack of strong features. Also, the discrepancies of the same person’s signature due to sickness, age or psychological factors make it quite difficult to distinguish between original and forged signatures hence provide inaccuracy in offline systems. It is also possible for a pretender to make an unforeseeable forgery of a signature. Also, it is much more difficult to solve a signature verification problem through offline approaches.

Despite all this, an offline system still has plentiful advantages as compared to an online system. In an offline verification system, there is no need for the presence of signer at verification phase because scanned signatures are processed which are already taken and stored in the database. Special devices or gadgets are not involved in it, so this system is suitable for the authentication of bank cheques, authorized documents, brochures and forms signed on a daily basis.

The Offline Signature Verification is categorized into writer-dependent verification and writer-independent verification. In writer-dependent verification, the set of features on which the models will be trained are different for each user. This approach uses several types of models such as support vector machine, k-nearest neighbors, and logistic regression. These are trained using the specific features for an individual and the one with best accuracy will be picked for each user. Whereas in writer-independent verification, a common set of features are used for every user and a common model will be trained on these features. This model will be used for further classification.

2.LITERATURE SURVEY

Generally for this type of problem there are some common techniques used in the fields of Machine Learning and Deep Learning. Some of them are Fast R-CNN, Faster R-CNN, Single Shot Detector, Yolo, Supervised SVM for object detection and Track R-CNN, Tractor++, Mean Shift are some of the object tracking methods. In the below survey there are some techniques used and were mentioned in the proposed work column in every table briefly for better understanding.

TABLE 1

Author	Title	Proposed Work	Drawbacks
Payal Mittal, Akashdeep Sharma, Raman Singh (2002)	Deep learning based object detection in low altitude UAV Datasets [1]	• Given detailed information about deep learning based object detection algorithms. • Provided comprehensive list of low-altitude UAV datasets. • Discussed about the popularity of UAV datasets over the years. • Compared the performance of different networks on the VisDrone dataset. • Evaluation metrics : mAP(mean Average Precision),AP	Dedicated to only low altitude UAV datasets

In this survey they have used the simple DL networks like VGG-16 and VGG-19 and compared the results based on the evaluation metrics like mean Average Precision, Average Precision and confusion matrix.

TABLE 2

Author	Title	Proposed Work	Drawbacks
Haijun Zhang Mingshan Sun Qun Li Linlin Liu Ming Liu Yuzhu Ji (2020)	An empirical study of multi-scale object detection in high resolution UAV images [2]	• Introduced MOHR dataset.Discussed about the features of their dataset. • Compared MOHR dataset with existing aerial datasets. • Performed state-of the-art approaches on MOHR and compared the results. • Evaluation metric : Average Precision(AP)	The proposed dataset(MOHR) only contains 10,631 images,90,014 instances.Lack of large dataset may result in poor performance.

This research paper is based on the creation of a new dataset MOHR. Some of the top researchers in the field of aerial imaging came together and did research and came up with a dataset and performed some state of the art object tracking and detection on the dataset and compared the results with existing datasets.

TABLE 3

Author	Title	Proposed Work	Drawbacks
Paraskevi Nousi, Ioannis Mademlis, Iason Karakostas, Anastasios Tefas, Ioannis Pitas (2019)	Embedded UAV REAL TIME Visual Object Detection and Tracking [3]	• Real time object detection and tracking. • Object detection: Single-Shot Detector (SSD), You Only Look Once (YOLO) algorithms	The authors didn't consider region based object detectors like Faster R-CNN into their experimentation

		based on convolutional Region Proposal Network. • Object tracking: Multithreaded KCF, light - weight version of Siam FClite.	

This research paper talks about real world object detection and tracking. They have some of the famous techniques like yolo (You only look once)and SSD(Single Shot Detection) and advanced deep learning models like Siam FCLite for object tracking.

3.PROBLEM STATEMENT

Offline signature verification system. This is a model which is used to differentiate original and forged signatures of an individual. Choose an efficient machine learning or Deep learning model to classify the signatures of an individual.

4.PROPOSED METHODOLOGY

4.1.Dataset

The CEDAR Signature is a database of off-line signatures for signature verification. This dataset consists of signatures of 55 individuals. Each of 55 individuals contributed 24 signatures thereby creating 1,320 genuine signatures. Some were asked to forge three other writers' signatures, eight times per subject, thus creating 1,320 forgeries. Each signature was scanned at 300 dpi gray-scale and binarized using a gray-scale histogram. Salt pepper noise removal and slant normalization were two steps involved in image preprocessing. The database has 24 genuines and 24 forgeries available for each writer.

So, this dataset is used to compare different machine learning and deep learning models that are built.

4.2.DATASET PREPROCESSING

A. Binarization

As we know there is no importance of color in which the signer is signing, that's why we convert the image from color image to grayscale image. Then for future calculation we have converted that image to binary image using threshold which is found using OTSU's method. In binarization a color image is converted into a binary image so as to make feature extraction easier because it is required to compare only two pixel values (colors). It allows us to reduce the amount of image information (removing color and background), so the output image is black-white. The black-white type of the image is much easier for further processing.

B. Thinning

The signer may use different pens at different times and that's why thickness of the signature may vary and so as to eliminate the thickness differences thinning is to be performed and it makes the image one pixel thick. It improves the accuracy rate and also the computational time is reduced.

C. Cropping an image

After the previous step it is required to locate the exact position of the signature in the image to perform signature verification, because signatures can be anywhere on the paper and they never start from the same position and neither do they terminate at the same position and they can sign in different angles and sizes. To solve this problem a solution has been proposed. And the solution is to scan the signature and find the smallest rectangular area covering the total signature. It is achieved using the bounding box property from regionprops command and after that the required portion is extracted.

D. Resizing the image

The system must be able to maintain the high performance regardless of the size of the given signature. It should be important that the system must be insensitive enough for the size of the signature image. So, The image matrix is rescaled to standard resolution which is 224 X 224 in this case.

4.3 Feature extraction and Classification

4.3.1.Manual feature extraction and classification using ML models

Some of the features are extracted manually from the signatures in the dataset. These features are used to classify the signatures. These features are given to machine learning models such as support vector machine, k-nearest neighbor and logistic regression for classification. And the performance of these models are measured using evaluation metrics such as accuracy, false acceptance rate and false rejection rate.

The features extracted are:

1. Height
2. Width
3. Aspect Ratio
4. Orientation
5. Number of connected components
6. Global centroid
7. Perimeter
8. Area

This is a user dependent model. Here, we have built a classifier for each user. Out of 24 genuine and 24 forged signatures of a user, 19 genuine and 19 forged signatures were used for training the model and remaining signatures are used for testing. So, we have obtained a total of 55 classifiers. And the average of evaluation metrics are mentioned below.

The results obtained from the above model are

Model	Accuracy	FAR	FRR
SVM	0.9136363636363637	0.0600	0.0181
KNN	0.8878787878787877	0.0900	0.0218
Logistic Regression	0.8924242424242423	0.0563	0.0400

4.3.2.Feature extraction using VGG16 and classification using ML models

The features are extracted from a neural network architecture called VGG-16. The obtained features are given to machine learning models for classification. The same set of machine learning models and evaluation metrics were used to compare the models.

4.4.OBJECT DETECTION

The Yolov5 model was used as an Object detector.

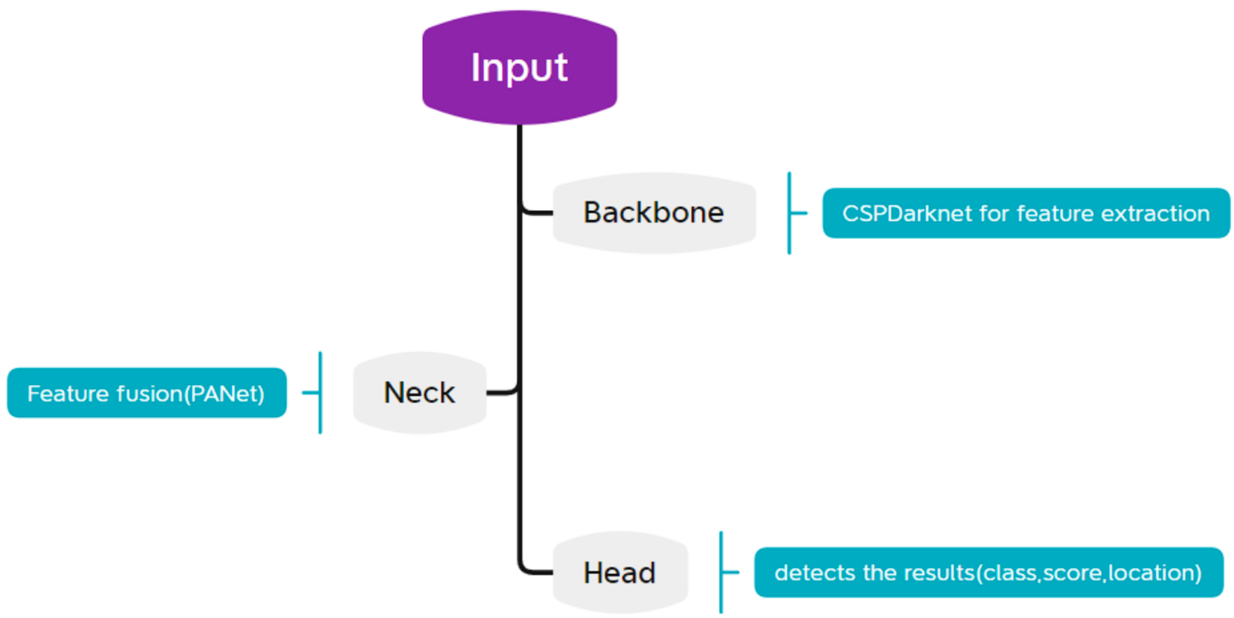

Fig.1. Yolov5 Architecture

4.5.Yolov5 Architecture

Model Backbone: The model backbone is used in object detection to extract the most important features. It is the richest and most distinctive one. In YOLOv5, the backbone used is CSPNet which stands for Cross Stage Partial Networks.

Model Neck: Model neck is used in object detectors to build feature pyramids in order to detect an object of different sizes and scales. YOLOv5 uses PANet, which stands for Path Aggregation Network.

Model Head: Head This is the last layer of YOLOv5, and it detects the results in the form of confidence score, size, and accuracy.

Take the preprocessed dataset and feed the data to the Yolov5 model and start training. While the training phase is over and the model was trained well you will be given weights. ”What are weights?” Weights Biases is directly integrated into YOLOv5, providing experiment metric tracking, model and dataset versioning, rich model prediction visualization, and more.

4.6.OBJECT TRACKING

Object tracking is a deep learning process where the algorithm tracks the movement of an object. In other words, it is the task of estimating or predicting the positions and other relevant information of moving objects in a video.

Object tracking usually involves the process of object detection. Here’s a quick overview of the steps: Object detection, where the algorithm classifies and detects the object by creating a bounding box around it. Assigning unique identification for each object (ID). Tracking the detected object as it moves through frames while storing the relevant information.

We have used DeepSORT as an Object Tracker.

4.7.DeepSORT

DeepSORT is a real-time,computer vision tracking algorithm for tracking objects while assigning an ID to each object. DeepSORT is an extension of the SORT (Simple Online Real Time Tracking) algorithm.SORT creates too many “ID switches” when the sight of objects is blocked.With the aid of a deep learning component that takes into account the visual characteristics of an object, DeepSORT enhances the motion model (Kalman filter).

4.8.DeepSORT Algorithm

Steps:

- Detection: YOLO is used to detect objects in the individual frames.

- Calculating the cost matrix: It assigns scores on how likely a detection belongs to a track.The cost matrix is a combination of the motion model(Kalman filter) and visual similarity(Deep Neural Network Embeddings).

-
    - -Kalman filter is an algorithm that will predict future positions based on current position.It predicts where the track will move(Prediction similarity is measured using Mahalanobis distance).

    - -Visual similarity is calculated using Cosine distance.

- Assignment problem: For each track that scores below a certain frame threshold, the algorithm assigns new detections.The detection that has the least cost is associated with the track.
    - -For all unmatched tracks, the algorithm runs an IoU(Intersection Over Union) matching.

DeepSORT Architecture

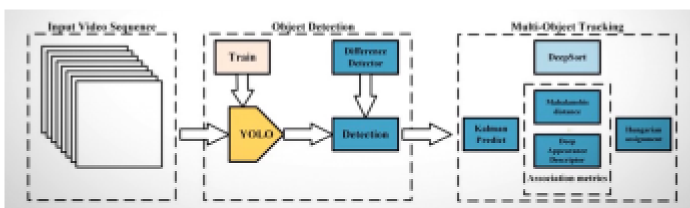

Fig.2. DeepSORT Architecture

Source: https://medium.com/augmented-startups/deepsort-deep-learning-applied-to-object-tracking-924f59f99104

4.9.Changes to the model and application build.

Adding NN layers to boost the performance: In the object tracking part in the Deep part of the DeepSort we have added an extra dense neural network layer and an extra pooling layer inorder to boost the result. The original model consists of the 3 neural network layers and no separate pooling layer, after addition of the custom neural network and pooling layers now the total layers the Deep part are 5 functional layers.

One of the applications implemented are for crowd control, traffic control and security purposes. It is an alert system where the alarm or alert system goes off. It is basically a counter where it counts the activity that’s happening when the system is on and does the following task given by the instructor at that very moment. Choose one of the functions at a time.

- Crowd management: when the amount of crowd exceeds the limit.
- Traffic management: When the no of vehicles reaches the count.
- Border security: Notices and notifies the user if there is any activity at the place it is actively securing.

Web application: Built a web application where one can upload a photo or video of the aerial content and can see the results stored in a file format in the local storage of the computer. This web application makes the work of the users easy by just clicking the button one can upload and see the models running quite efficiently and providing the results they want.
Local application: Built a local application where one can upload a photo or video or camera to upload or show the aerial content through camera and can see the results. This application is similar to the web application but works on a computer like a software.

5.RESULTS

Results of Yolov5

These are the results obtained from running the yolov5 object detection model.
We have trained the model on our dataset for 30 epochs.

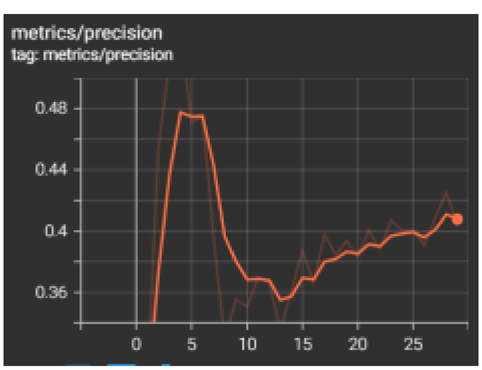

Fig. 3. Precision of Yolov5.

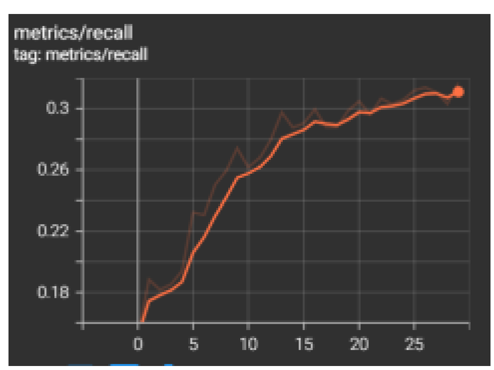

Fig. 4. Recall of Yolov5.

Fig.3 how much of the boundingbox predictions are correct- Precision
Fig.4 how much of the true boundingbox are correctly predicted-Recall

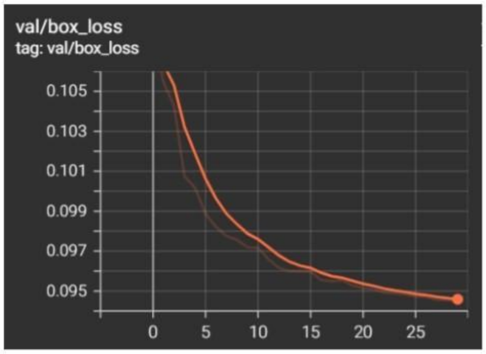

Fig. 5. body box identification loss of Yolov5.

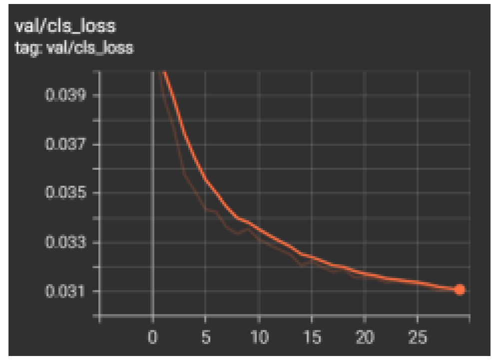

Fig. 6. class identification loss of Yolov5.

Fig.5 (Mean Squared Error)regression loss of boundingbox.
Fig.6 (Cross Entropy) classification loss.

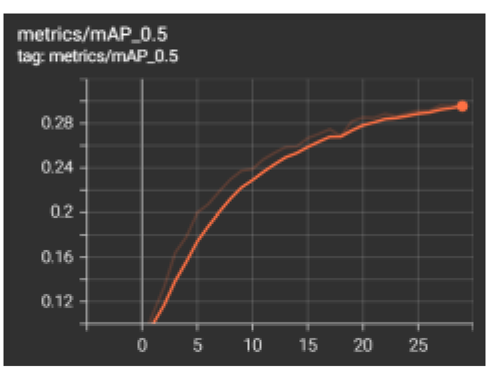

Fig. 7. mean absolute precission.

Fig. 8. final output with confidence scores associated with.

Fig.7 (mAP)Mean Average Precision at Intersection Over Union(IoU) threshold 0.5.
Fig.8 Object detection by yolov5 given an input image.

6.APPLICATIONS RESULTS

6.1.Results of the applications done

This application is for traffic control, crowd management and security purposes. There will be a line and a count associated with, when a car or pedestrian passes the line, it will increase the count of the item whichever crossed the line.

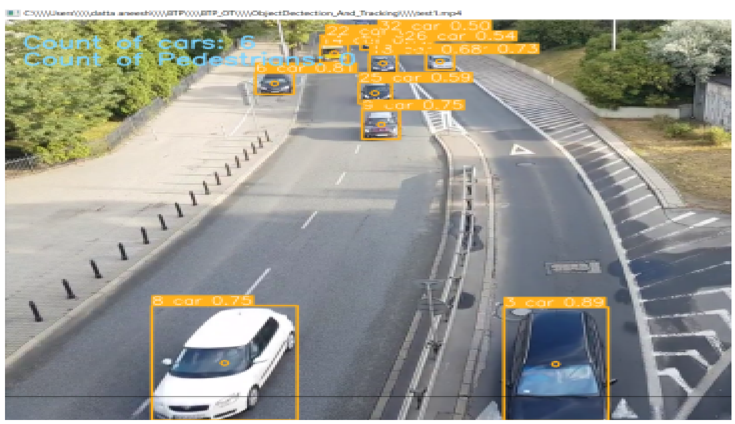

Fig. 9. output of the application, line with counting associated with it.

6.2.Result of the local application

Local application using PyQt5(GUI creation) and Computer Vision. Here one will be provided with three options: insert picture or insert video or open camera for live detection. One can click on a button and browse the picture or video and upload then it will give the result on the go.

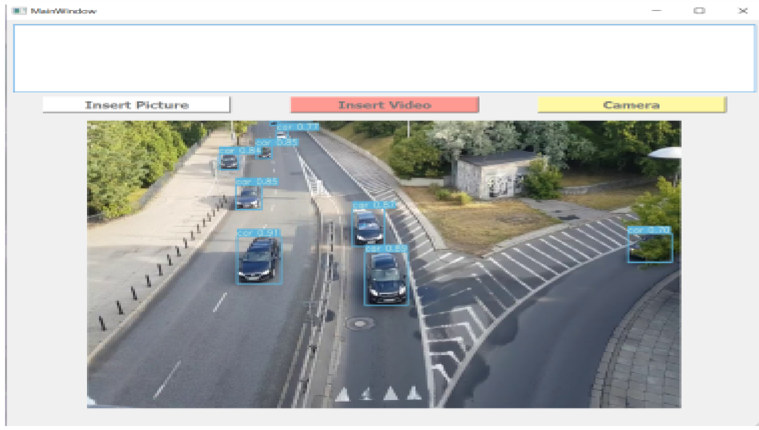

Fig. 10. picture of the local application.

6.3.Result of the web application

Web application using Streamlit. Here one will be provided with a browse option where one can insert pictures or insert video for live detection. Here when the photo is uploaded the instant detection will be happening and shows the result to the user and when the video is uploaded the detection part will take place and the result will be stored in the local system on completion.

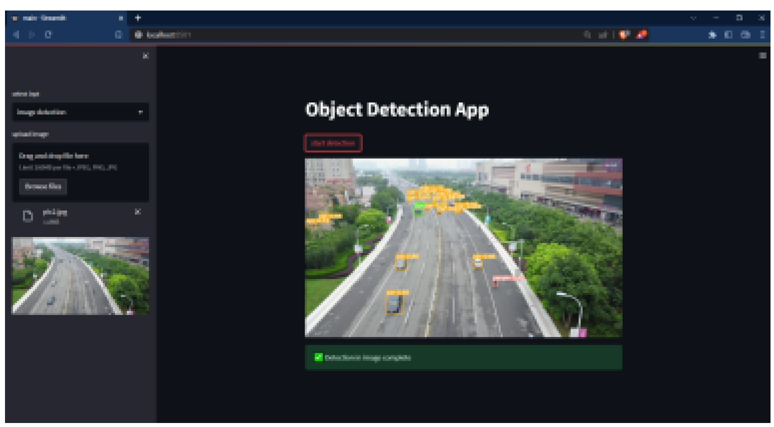

Fig. 11. picture of the web application.

7.LIST OF FIGURES

- Figure - 4 - how much of the true boundingbox are correctly predicted-Recall
- Figure - 5 - (Mean Squared Error)regression loss of boundingbox
- Figure - 6 - (Cross Entropy) classification loss
- Figure - 7 - (mAP)mean Average Precision at Intersection Over Union(IoU) threshold 0.5
- Figure - 8 - Object detection by yolov5 given an input image
- Figure - 9 - output of the application, line with counting associated with it
- Figure - 10 - picture of local application
- Figure - 11 - picture of web application

8.ACKNOWLEDGEMENTS

We would like to thank our mentor, Prof. Shathanaa for guiding us in this project and providing inputs and suggestions during evaluations and working.All team members have contributed equally,sharing the workload since the beginning.

9.REFERENCES

[1] Mittal, P., Singh, R., Sharma, A. (2020). Deep learning-based object detection in low-altitude UAV datasets: A survey. Image and Vision computing, 104, 104046.

[2] Zhang, H., Sun, M., Li, Q., Liu, L., Liu, M., Ji, Y. (2021). An empirical study of multi-scale object detection in high resolution UAV images. Neurocomputing, 421, 173-182.

[3] Iqbal, M. J., Iqbal, M. M., Ahmad, I., Alassafi, M. O., Alfakeeh, A. S., Alhomoud, A. (2021). Real-Time Surveillance Using Deep Learning. Security and Communication Networks, 2021.

[4] Micheal, A. Ancy; Vani, K.; Sanjeevi, S.; Lin, Chao-Hung (2020). Object Detection and Tracking with UAV Data Using Deep Learning. Journal of the Indian Society of Remote Sensing, (), –. doi:10.1007/s12524-020- 01229-x.

[5] Nousi, Paraskevi; Mademlis, Ioannis; Karakostas, Iason; Tefas, Anastasios; Pitas, Ioannis (2019).EMBEDDED UAV REAL-TIME VISUAL OBJECT DETECTION AND TRACKING , 708-713.

[6] Wu, Qingtian; Zhou, Yimin. Real-Time Object Detection Based on Unmanned Aerial Vehicle(2019) 574-579.

[7] Yu, H., Li, G., Zhang, W., Huang, Q., Du, D., Tian, Q., & Sebe, N. (2020). The unmanned aerial vehicle benchmark: Object detection, tracking and baseline. *International Journal of Computer Vision*, *128*(5), 1141-1159.

[8] Wu, H. H., Zhou, Z., Feng, M., Yan, Y., Xu, H., & Qian, L. (2019, June). Real-time single object detection on the uav. In *2019 International Conference on Unmanned Aircraft Systems (ICUAS)* (pp. 1013-1022). IEEE.

[9] Aradhya, H. R. (2019). Object detection and tracking using deep learning and artificial intelligence for video surveillance applications. *International Journal of Advanced Computer Science and Applications*, *10*(12).

[10] Jadhav, A., Mukherjee, P., Kaushik, V., & Lall, B. (2020, February). Aerial multi-object tracking by detection using deep association networks. In *2020 National Conference on Communications (NCC)* (pp. 1-6). IEEE.

[11] Pal, S. K., Pramanik, A., Maiti, J., & Mitra, P. (2021). Deep learning in multi-object detection and tracking: state of the art. *Applied Intelligence*, *51*(9), 6400-6429.

[12] Walambe, R., Marathe, A., & Kotecha, K. (2021). Multiscale object detection from drone imagery using ensemble transfer learning. *Drones*, *5*(3), 66.