TITLE: **Real-time Bird tracking system, which tracks the birds' movement and monitors their behavioral insights.**

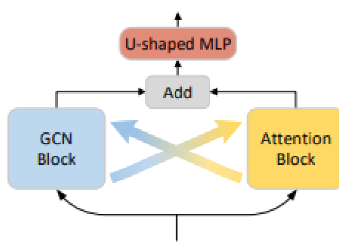
Ashish Aman (M22MA002)
Bikash Dutta (D22CS051)
Prashant Gautam (M22CS057)

INTRODUCTION:

In the current era of digitalization, technology has made too much advancement in the field of Machine learning and deep learning. And when we talk about deep learning, we must remember the models based on real-time scenarios that use computer vision. Here, we will develop one such model which will track the movement of birds in real-time.

WORKFLOW:

1. We tried to approach the problem statement by estimating the real time movement of bird by implementing a new framework that interweaves a graph-based convolutional network (GCN) and an attention-based network (ABN) on Human 3.6M dataset to simultaneously model the spatial relationships between body joints and learn the key features from the input image.
2. We implemented the above pose estimation of module, which contains two main components: the Interweaved Graph and Attention module (IGA) and the U Shaped MLP module (uMLP), on **Human 3.6M.**



3. On implementation of IGANet ,we achieved qualitative results of the proposed IGANet method on the Human3.6M dataset. The figure displays side-by-side comparisons between the ground truth 3D poses (in blue) and the predicted 3D poses (in red) for various actions performed by different people.
4. Transfer learning was applied to the 3D human pose estimation model to adapt it for the task of estimating 3D poses of birds by implementing as – obtained a labeled dataset of 3D bird poses made by us,Preprocess the labeled dataset ,Fine-tune the weights of the pre-trained IGANet model on the bird pose dataset and finally evaluating the fine-tuned model on a validation set of bird poses to ensure that it can accurately estimate the 3D poses of birds.But we did not got the results from it.
5. Hence we created the dataset of bird which contains the frames of 4 videos(train + test) and labeled the key points- Beak, head, left leg, right leg, right wing, left wing, Body and tail for each frame and generated the key points of their body structure for 66 frames manually.

6. After that we trained the model using ResNet50 using the above generated datasets on the platform DeepLabCut which is an efficient tool for the pose estimation based on deep neural networks that achieves excellent results with minimal training data (typically 50-200 frames).
7. Results that we got were satisfactory but not that much accurate ,as keypoints dislocated from the body of the bird most of the time in their real-time live tracking.
8. Next we have shifted to the YOLO V8n framework in which activities of birds were made as a detection part, where it tracks the 4 activities of birds in real time displayed under bounding boxes.
9. Dataset :- We have taken the Bird dataset Caltech-UCSD Birds-200-2011 and tried to annotate the dataset using the online tool makesense.ai with 4 activities which is labeled as 0-Swimming, 1-Flying, 2-Walking, 3-Resting on 1083 training and 120 validation images.
10. We have trained the YOLO v8n model of Ulltralytics using a train_data.yaml file in which image size of 640, batch of 10 img were passed and run for 100 epochs.
11. Next we have predicted the models performance on 3 videos and Results were much better than the earlier implementation in real time which monitors the four activities: Swimming, flying, walking and resting.
12. After that we also implemented the Faster R-CNN ,which is a popular object detection algorithm that was proposed in 2015. It builds upon the earlier R-CNN and Fast R-CNN architectures and achieves significant speed improvements while maintaining or improving detection accuracy.
13. Here we have generated the same annotations which were used in the YOLO v8n and generated the .xml extensions of each image.
14. The main differences between Faster R-CNN and YOLOv8n is the approach they take to object detection. Faster R-CNN uses a two-stage approach where it first generates region proposals and then classifies those proposals. YOLOv8n, on the other hand, is a single-shot detector that generates object predictions directly from the input image.
15. Faster R-CNN typically uses a ResNet or VGG network as the backbone, while YOLOv8n uses a darknet architecture.

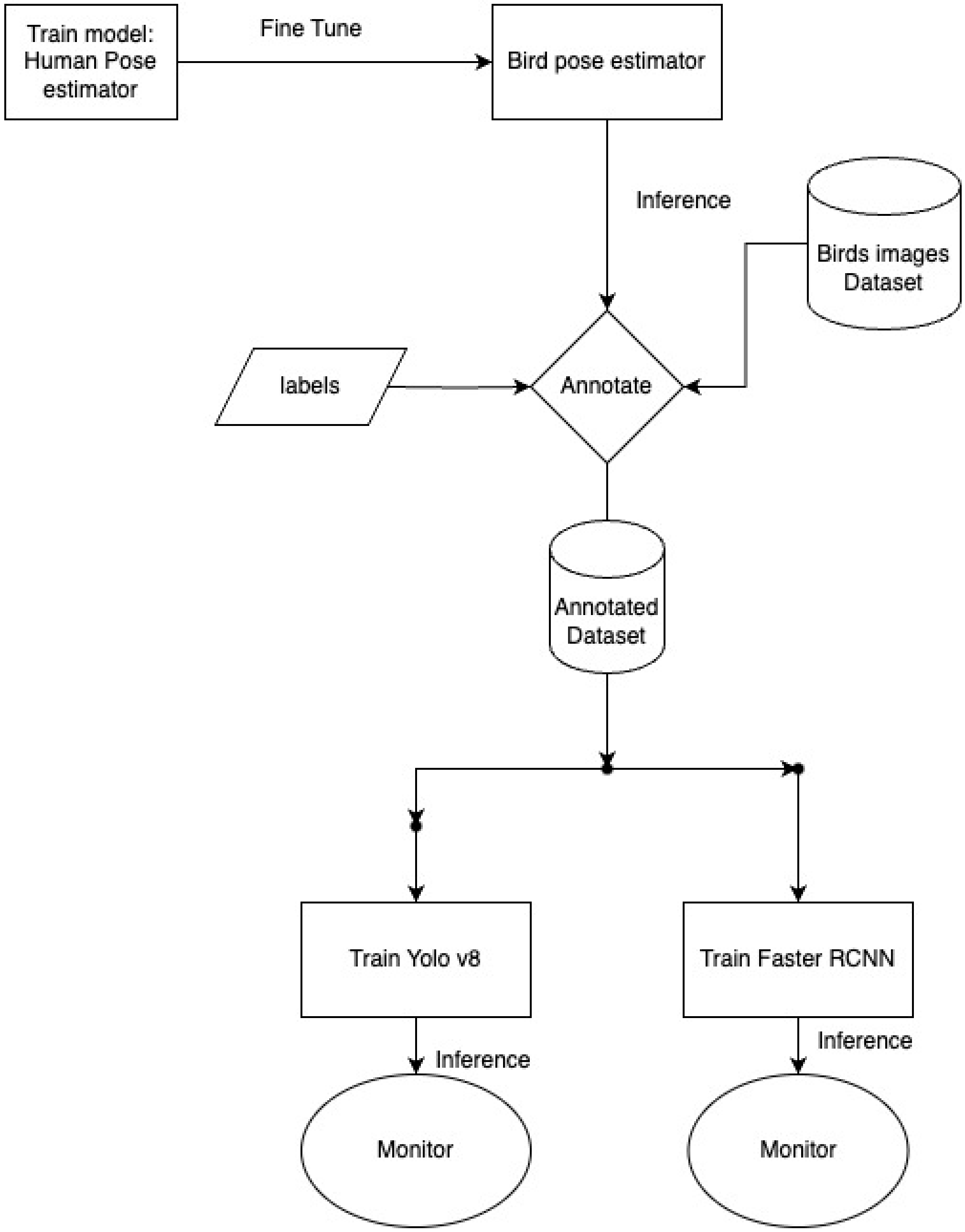


Fig: Flow chart of our complete experimental methodology and workflow.

RESULTS:

1. Visualization Result of 3D- Human Pose Estimation

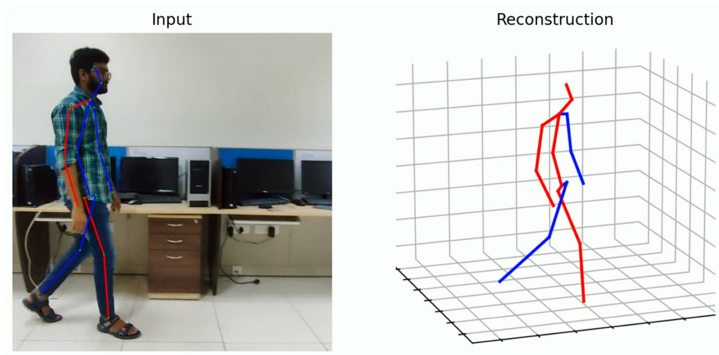


Fig1. The figure displays side-by-side comparisons between the ground truth 3D poses (in blue) and the predicted 3D poses (in red) for various actions performed by different people.

1. Snaps of Videos of Real time tracking of ResNet50 model- DeepLabCut.

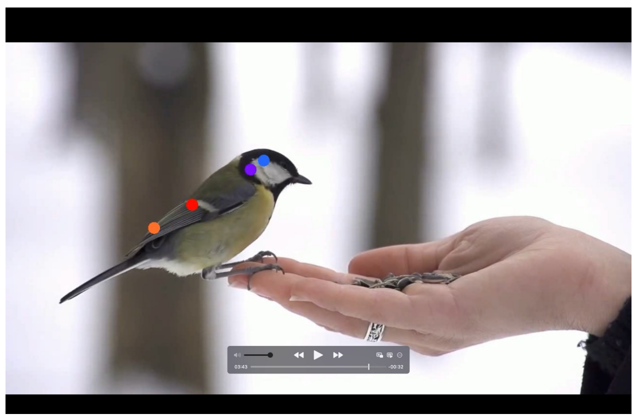
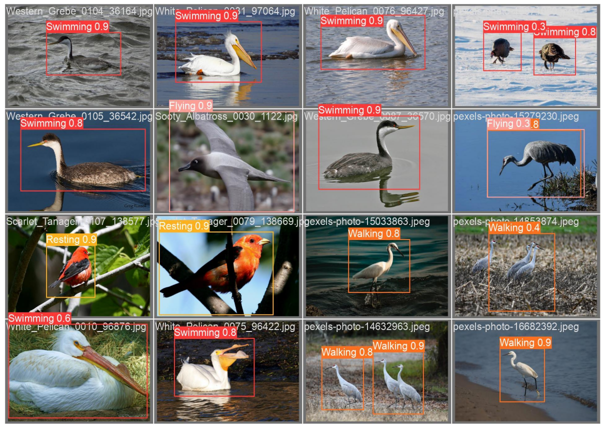


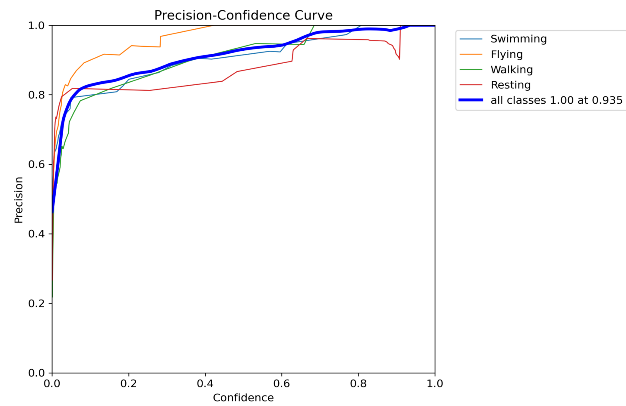
Fig2. The figure displays the detected key points of the bird.

2. YOLO V8n results and evaluation metrics on 1083 training and 120 validation images:

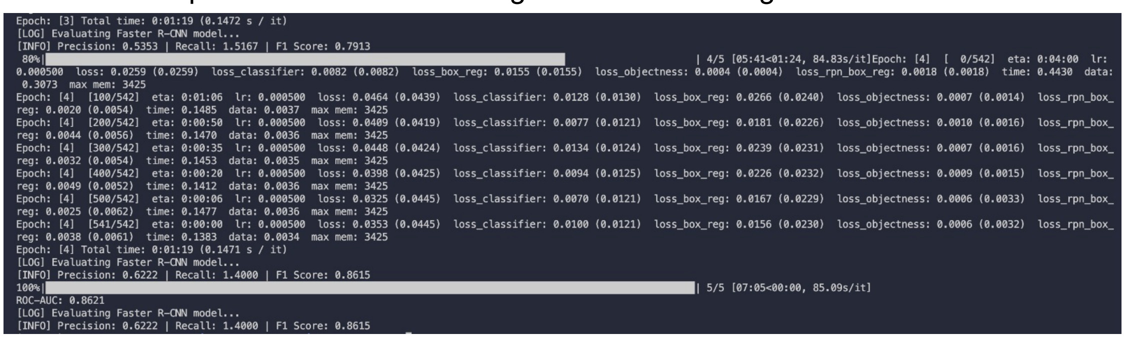
PERF METRICS	SCORE
Precision	0.971
Recall	0.86
mAP(50 % IoU)	0.955
mAP(100 % IoU)	0.663
Memory Usage	1.42GB

3. Graphs and Snaps of Videos of Real time tracking of YOLO V8n framework.





4. Graphs and Snaps of Videos of Real time tracking of F-RCNN framework.-[unsuccessful]
   We have implemented it but bounding box is not showing in the video.



```
Epoch: [3] Total time: 0:01:19 (0.1472 s / it)
[LOG] Evaluating Faster R-CNN model...
[INFO] Precision: 0.5353 | Recall: 1.5167 | F1 Score: 0.7913
 80%|                                   | 4/5 [05:41<01:24, 84.83s/it]Epoch: [4]  [  0/542]  eta: 0:04:00  lr:
0.000500  loss: 0.0259 (0.0259)  loss_classifier: 0.0082 (0.0082)  loss_box_reg: 0.0155 (0.0155)  loss_objectness: 0.0004 (0.0004)  loss_rpn_box_reg: 0.0018 (0.0018)  time: 0.4430  data:
 0.3073  max mem: 3425
Epoch: [4]  [100/542]  eta: 0:01:06  lr: 0.000500  loss: 0.0464 (0.0439)  loss_classifier: 0.0128 (0.0130)  loss_box_reg: 0.0266 (0.0240)  loss_objectness: 0.0007 (0.0014)  loss_rpn_box_
reg: 0.0020 (0.0054)  time: 0.1485  data: 0.0037  max mem: 3425
Epoch: [4]  [200/542]  eta: 0:00:50  lr: 0.000500  loss: 0.0409 (0.0419)  loss_classifier: 0.0077 (0.0121)  loss_box_reg: 0.0181 (0.0226)  loss_objectness: 0.0010 (0.0016)  loss_rpn_box_
reg: 0.0044 (0.0056)  time: 0.1470  data: 0.0036  max mem: 3425
Epoch: [4]  [300/542]  eta: 0:00:35  lr: 0.000500  loss: 0.0448 (0.0424)  loss_classifier: 0.0134 (0.0124)  loss_box_reg: 0.0239 (0.0231)  loss_objectness: 0.0007 (0.0016)  loss_rpn_box_
reg: 0.0032 (0.0054)  time: 0.1453  data: 0.0035  max mem: 3425
Epoch: [4]  [400/542]  eta: 0:00:20  lr: 0.000500  loss: 0.0398 (0.0425)  loss_classifier: 0.0094 (0.0125)  loss_box_reg: 0.0226 (0.0232)  loss_objectness: 0.0009 (0.0015)  loss_rpn_box_
reg: 0.0049 (0.0052)  time: 0.1412  data: 0.0036  max mem: 3425
Epoch: [4]  [500/542]  eta: 0:00:06  lr: 0.000500  loss: 0.0325 (0.0445)  loss_classifier: 0.0070 (0.0121)  loss_box_reg: 0.0167 (0.0229)  loss_objectness: 0.0006 (0.0033)  loss_rpn_box_
reg: 0.0025 (0.0062)  time: 0.1477  data: 0.0036  max mem: 3425
Epoch: [4]  [541/542]  eta: 0:00:00  lr: 0.000500  loss: 0.0353 (0.0445)  loss_classifier: 0.0100 (0.0121)  loss_box_reg: 0.0156 (0.0230)  loss_objectness: 0.0006 (0.0032)  loss_rpn_box_
reg: 0.0038 (0.0061)  time: 0.1383  data: 0.0034  max mem: 3425
Epoch: [4] Total time: 0:01:19 (0.1471 s / it)
[LOG] Evaluating Faster R-CNN model...
[INFO] Precision: 0.6222 | Recall: 1.4000 | F1 Score: 0.8615
100%|                                   | 5/5 [07:05<00:00, 85.09s/it]
ROC-AUC: 0.8621
[LOG] Evaluating Faster R-CNN model...
[INFO] Precision: 0.6222 | Recall: 1.4000 | F1 Score: 0.8615
```

Fig: Training Results for fine tuning Faster-RCNN our custom dataset on pretrained coco dataset.

VIDEO LINKS:

Dataset:Human 3.6M

Dataset: Key Points Annotation for DeepLabCut

Dataset:Train_data of Annotated Bird image for YOLOv8n

Dataset:Train_data of Annotated Bird image for FRCNN

Video for 3D Human POSE Estimation

Real time tracking of Birds using DeepLabCut

Real time tracking of Birds using YOLO V8n

FUTURE SCOPE:

1. The bird tracking system's coverage and precision can be expanded through integration with other technologies like drones, satellites, or ground sensors. For instance, the precision of bird tracking algorithms can be enhanced by collecting high-resolution photos and videos of birds in flight using drones.

2. The real time tracking of the bird will help researchers gain a better understanding of bird migration patterns, habitat use, and behavior.
3. The tracking system can be modified to follow several bird species, allowing scientists to examine the dynamics of avian communities in response to climatic shifts and habitat alteration.
4. The bird tracking can be extended to any creature,thereby we can track and safeguard and also open further research on other species

LIMITATIONS:

1. The model which is deployed gives the result but it's not that accurate on which we can be relied upon and do our research to collect data.
2. In the YOLO v8n model, during activity transitioning the model confuses and detects wrong activities,which can create miscalculations if it is applied on real world applications.
3. In the DeepLabCut model, the key points get shifted outside the body of birds ,thus integration with other tracking algorithms will give wrong data.

REFERENCES:

1. https://pytorch.org/tutorials/intermediate/torchvision_tutorial.html
2. https://paperswithcode.com/dataset/human3-6m
3. https://ultralytics.com/
4. https://arxiv.org/pdf/2304.14045.pdf
5. https://www.makesense.ai/
6. www.mackenziemathislab.org/deeplabcut
7. https://github.com/tensorflow/tfjs-models/tree/master/posenet

Deep learning Key-points:

1. **Through DeepLabCut we have trained the ResNet 50 model on a custom bird dataset.**
2. **We have tried Transfer Learning to fine tune the Bird tracking model.(CURRENTLY INCOMPLETE)**
3. **YOLO uses a deep convolutional neural network (CNN) architecture, primarily designed for object detection, and its architecture is optimized for processing images in real-time.**
4. **Here YOLO Version is V8 nano version is used which is fastest in feature fusing ,and integrating anchoring box which helps in detecting smaller objects ,also computation time and model size is less.**

5. **The 3D human pose estimation model uses Attention mechanism to learn the weighted combination of the keypoint embeddings, which learn the global spatial and temporal dependencies between the keypoints.**
6. **We have tried to implement Faster RCNN on the annotated dataset but detection was not successful.**