Natural Human-Computer Interface Based on Gesture Recognition with YOLO to enhance user experience

MOMINA LIAQAT ALI

MIDDLE TENNESSEE STATE UNIVERSITY

OUTLINE

01
Introduction

02
Literature Review

03
Gesture Recognition

04
Natural HCI Design

05
Results

06
Conclusion & Future Work

01

INTRODUCTION

MIDDLE TENNESSEE STATE UNIVERSITY

INTRODUCTION

- Motivation
- Problem Statement

Motivation

Hand Tracking & Gesture Recognition

Enables computers to recognize and Respond to hand movements.

- Gained popularity during COVID-19.
- Demand for gesture recognition technologies is growing.
- Extend it to create virtual Engineering lab environment for Engineering students.

Problem Statement

Limitations of current techniques

Precision, Real-time responsiveness, adaptability and seamless Design.

- Precision:
  - To ensure reliable interaction by accurately interpreting hand movements.
- Real-time Responsiveness:
  - Timely response to optimize overall user experience.
- Restricted to few number of poses:
  - Current VR systems which are based on popular libraries like Unity offer restricted number of recognized poses and this too by involving third party plugins.

02

LITERATURE REVIEW

MIDDLE TENNESSEE STATE UNIVERSITY

Object Detection Algorithms

1. Single Stage Object Detectors

2. Two Stage Object Detectors
   - Region Proposals
   - Classification

BUT…

Computationally Expensive

Require large labeled data

Object Detection Algorithms

1 Single Stage Object Detectors

- No Region Proposal Stage
- Direct Prediction

2 Two Stage Object Detectors

- Region Proposals
- Classification

WHY YOLO?

Less Computation Cost

Real-time Performance

Grid based approach

Gesture Recognition

- Traditional Gesture Recognition Techniques
  - Hidden Markov Model (Chen et. al)
  - Orientation Histogram (Freeman et al.)
  - Finite State Machines (Hong et al.)
- Advanced Deep Learning Based Techniques
  - sEMG with CNN (Ozdemir et. al)
  - Depth camera with YOLOv3 (Yu et al.)
  - Kinetic Sensors with DNN (Tang et al.)

03

GESTURE RECOGNITION

MIDDLE TENNESSEE STATE UNIVERSITY

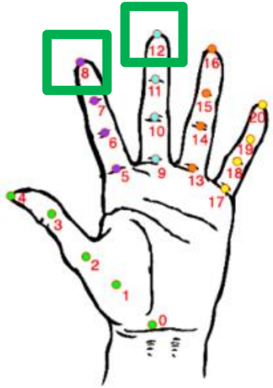
THREE – STEP HAND GESTURE RECOGNITION

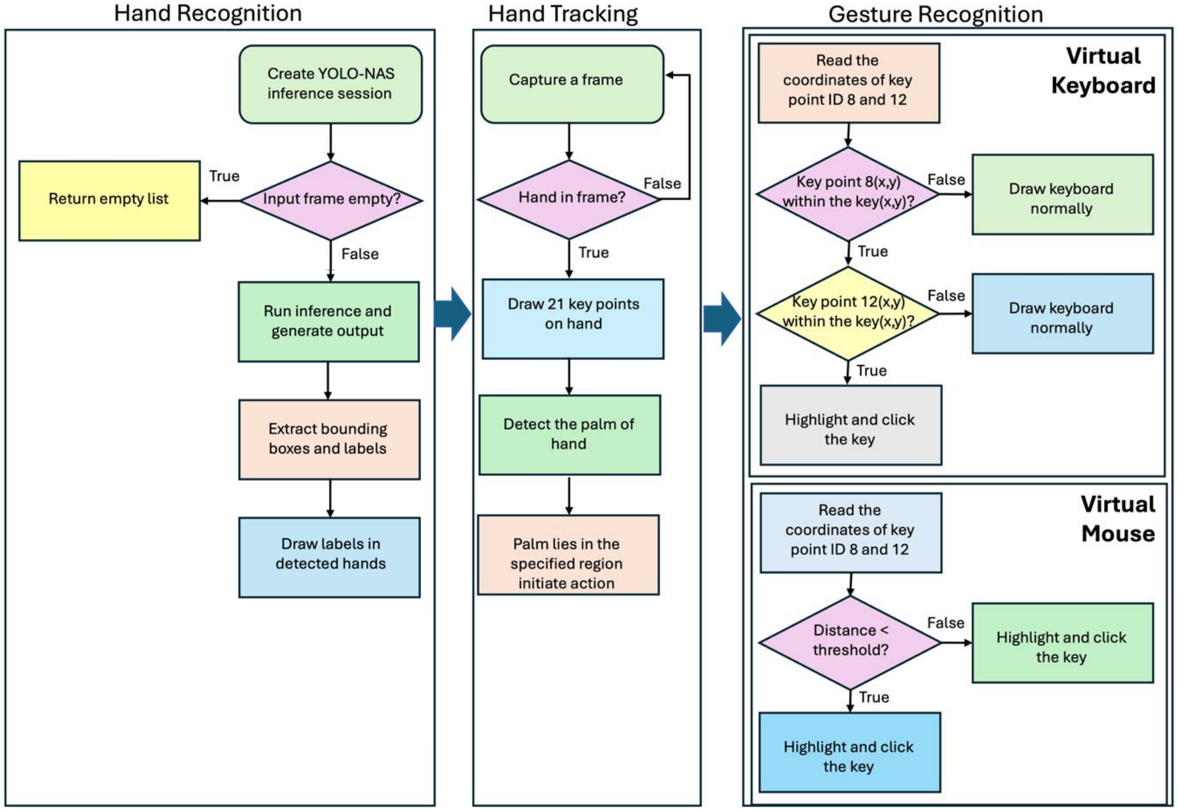
01 HAND RECOGNITION

02 HAND TRACKING

03 HAND GESTURE RECOGNITION

THREE – STEP PROCESS





- $Euclidean\ distance = \sqrt{(x_{12} - x_8)^2 + (y_{12} - y_8)^2}$

04

NATURAL HCI DESIGN

MIDDLE TENNESSEE STATE UNIVERSITY

GESTURE RECOGNITION IMPLEMENTATION

01 DATA COLLECTION & PRE-PROCESSING

02 GENERATING ANNOTATIONS

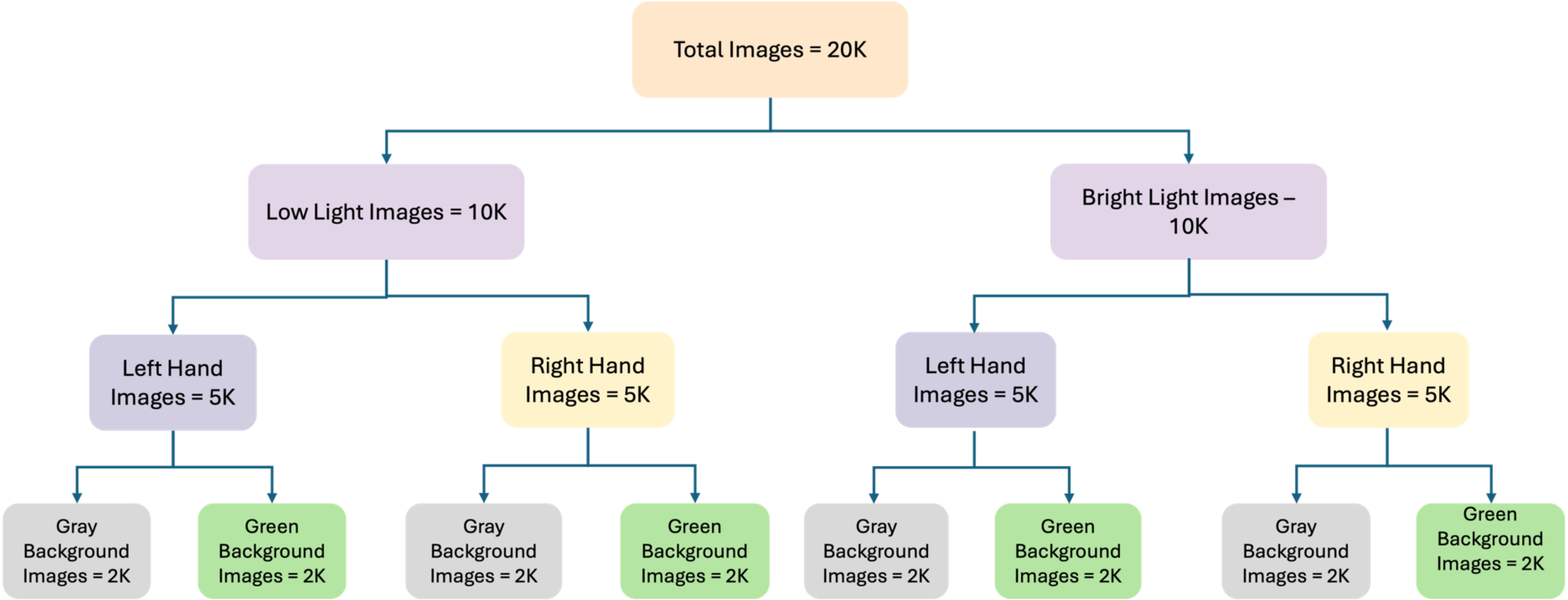
03 MODEL TRAINING & FINE TUNING

DATA COLLECTION & PRE-PROCESSING

- Gathered data using webcam.
- Dataset contains 20K images.
- Augmentation techniques like flipping and grayscale were used .
- All images were taken with green screen background in low light and bright light conditions.

Sample Images from Dataset

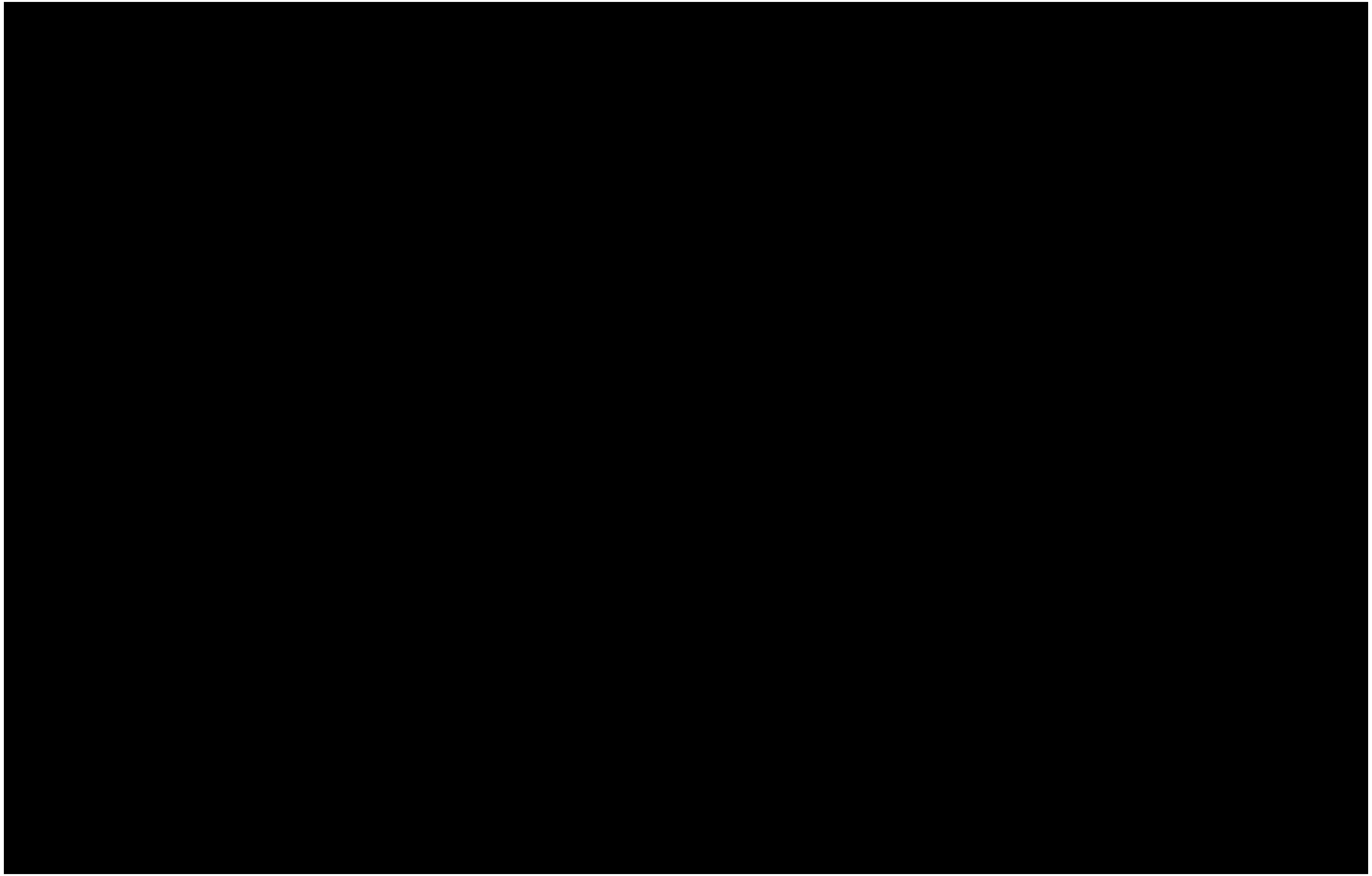
DATASET CONSTRUCTION TREE



MIDDLE TENNESSEE STATE UNIVERSITY

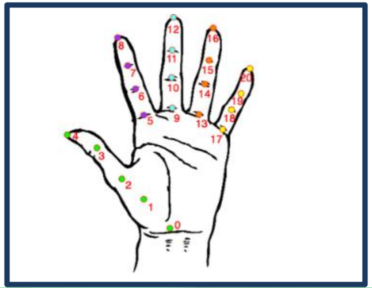
GENERATING ANNOTATIONS

Why not manual annotations?

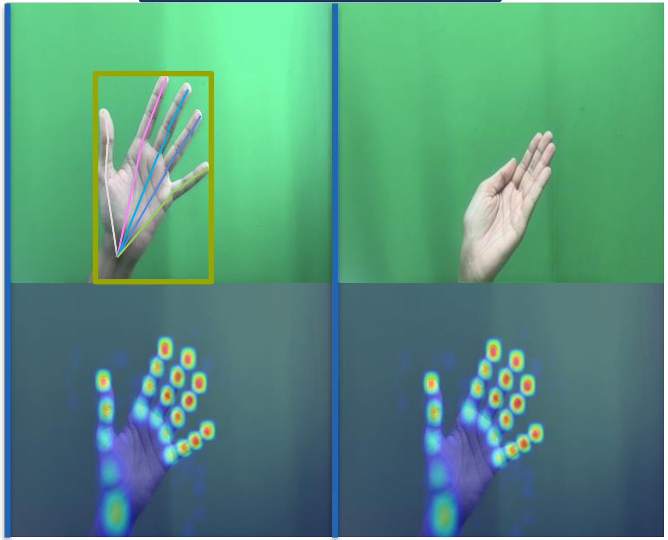


GENERATING ANNOTATIONS

- 21 key-points on human hand were annotated.
- RTMDet which is trained on hand datasets.
- RTMDet outperforms YOLO with 52.8% AP on COCO and 300+ FPS on an NVIDIA 3090 GPU.
- Used RTMDet-Nano for detection and RTMPose for posture estimation.
- Annotations were generated in json format.



HAND LANDMARKS

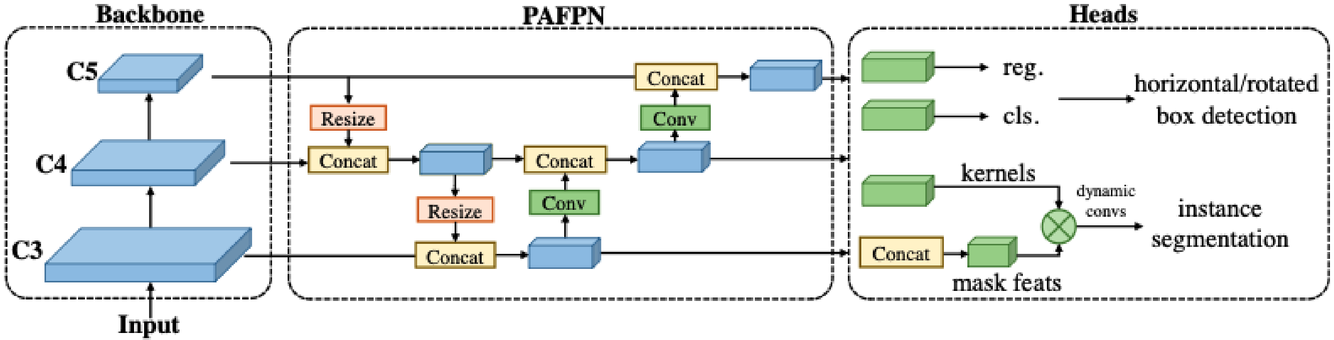


Original Image

Annotated Image

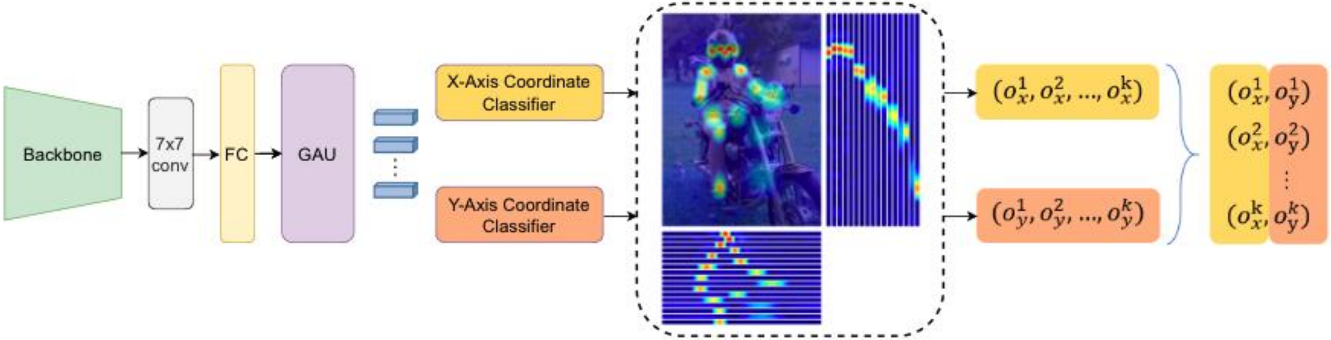
RTMDet

Lyu et. al. 2022

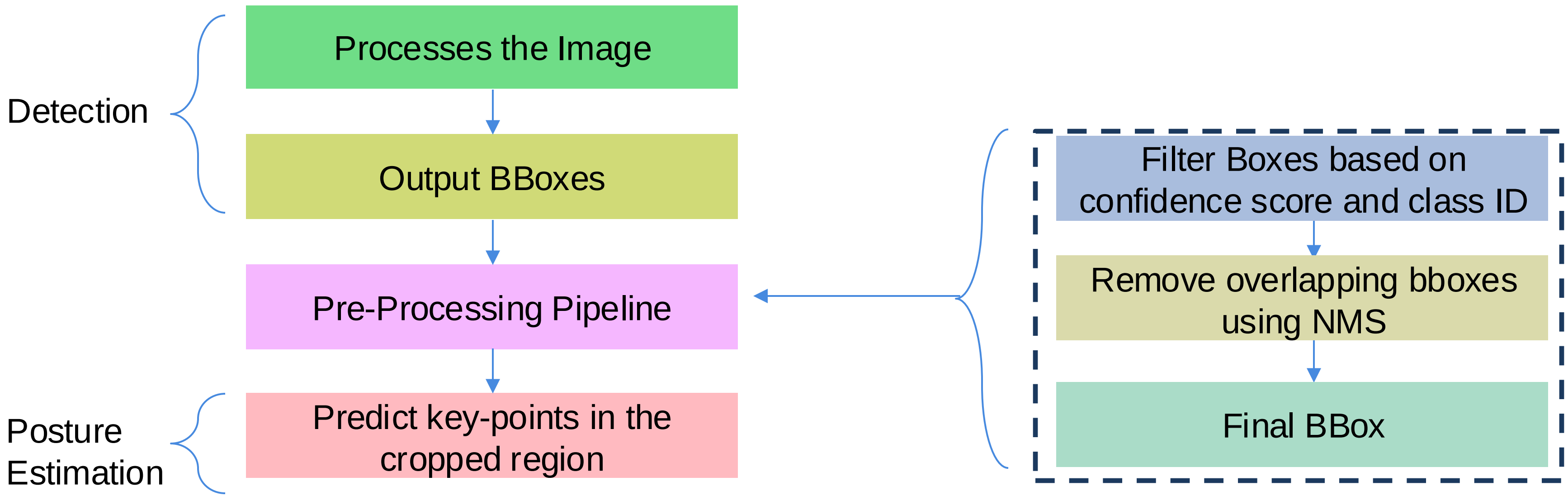


RTMPose

Jian et. al. 2023

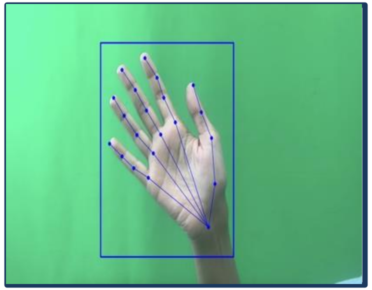


How RTMDet and RTMPose are combined?

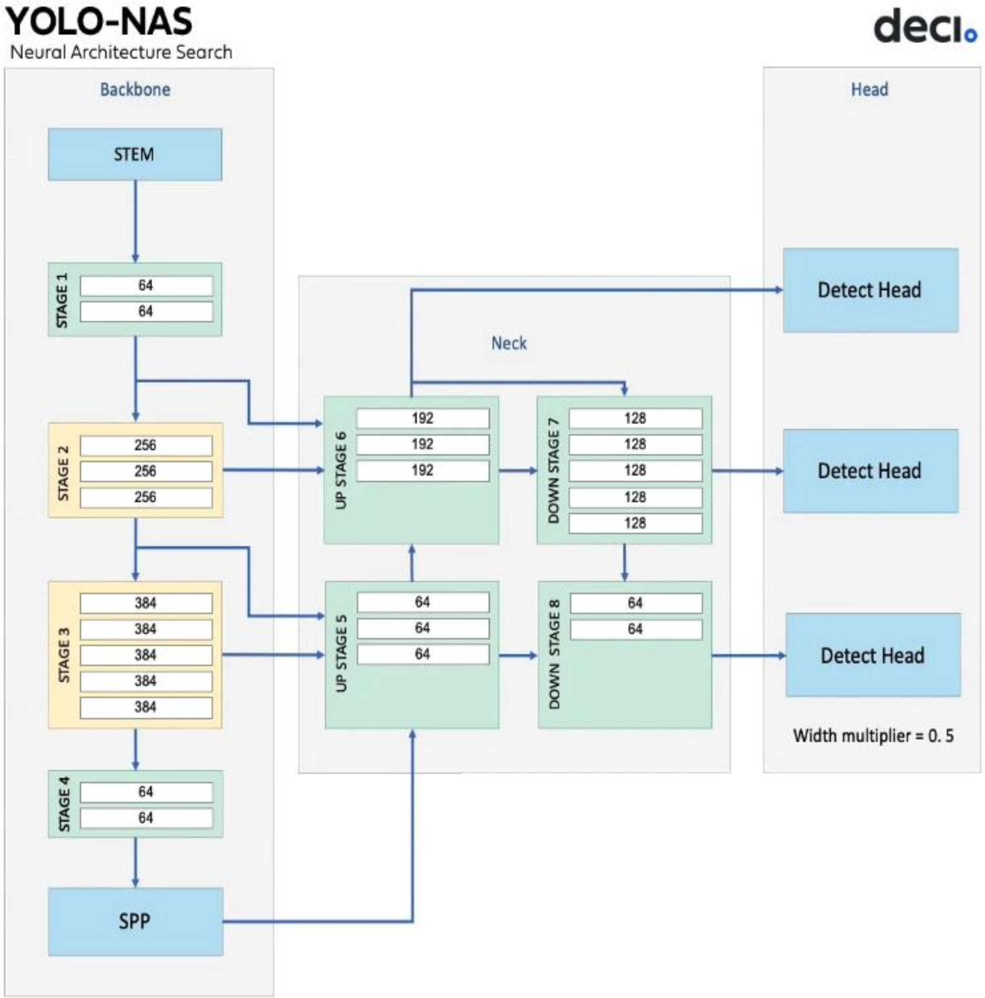


MODEL TRAINING & FINE TUNING

- Used YOLO-NAS Pose; a sibling model of YOLO-NAS.
- Famous model because of its capability of being a single-stage detector which makes it fast in real-time applications.
- YOLO-NAS Pose performs both detection and estimation of Pose in single pass.
- YOLO-NAS Pose is trained on COCO2017 Dataset.
- We fine-tuned the model on our dataset.



HAND LANDMARKS



MIDDLE TENNESSEE STATE UNIVERSITY

Training Configurations and System Details

Configurations	Value
Epochs	10
Learning Rate	0.001
Optimizer	AdamW
Batch size	32
Iterations per Epoch	439

Features	Details
CUDA Cores	7689 Cores
CPU Memory	24 GB @ 300 GBps
Compute Performance FP64	0.5 TFLOPS
Compute Performance FP32	30.3 TFLOPS
Architecture	NVIDIA Ada Lovelace

05

RESULTS

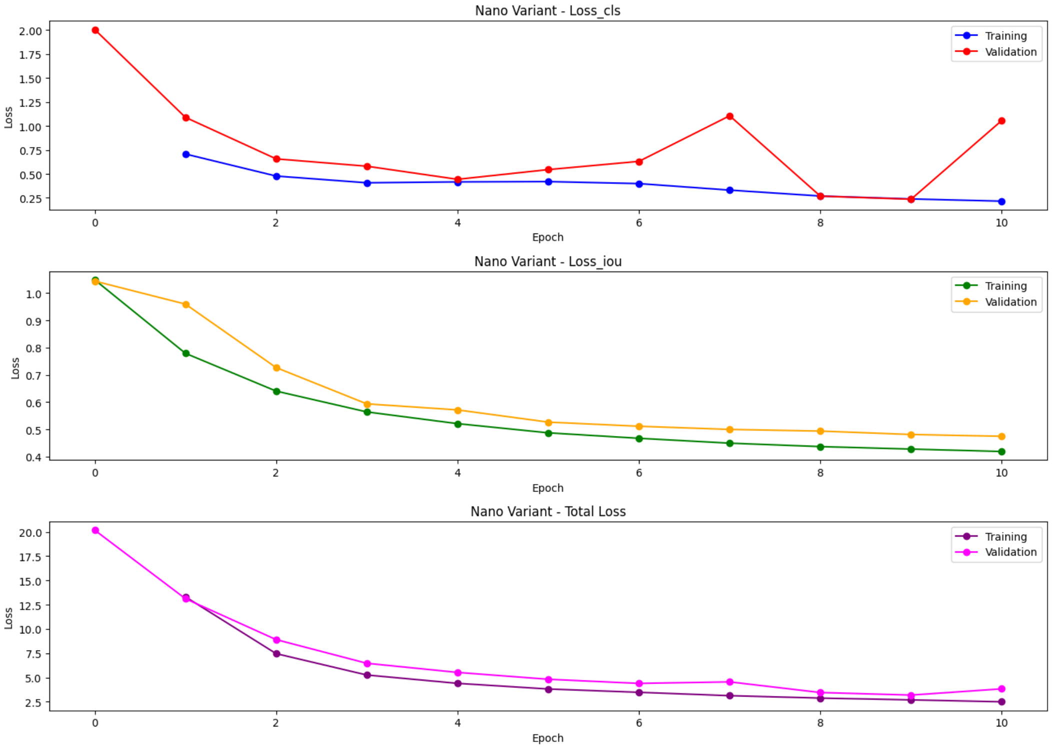
MIDDLE TENNESSEE STATE UNIVERSITY

RESULTS

- We created two instances of the dataset one had 5k images and named it as Dataset A and other had 20k images and names it as Dataset B.
- Each of these models were trained for 10 Epochs on both datasets A and B.
- Following metrics were used to evaluate the performance:
  - Average Precision (AP)
  - Average Recall (AR)
  - Total loss
  - classification loss
  - IOU loss

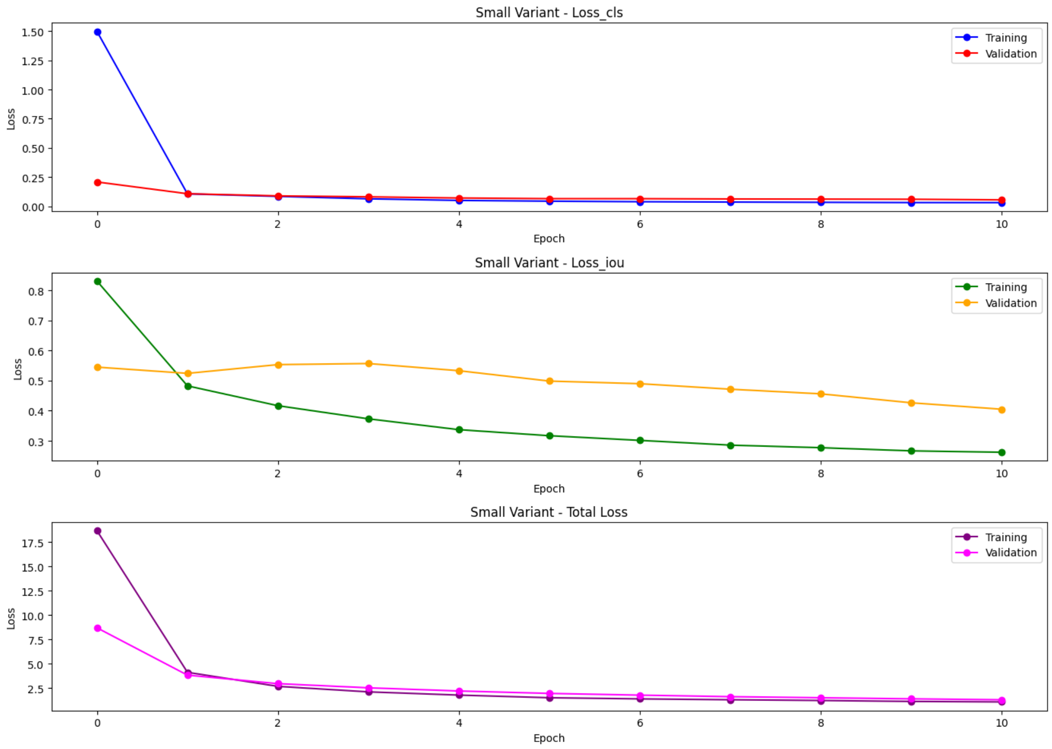
Model	No. of Parameters
Nano	9.9 million
Small	22.2 million
Medium	58.2 million

NANO MODEL - DATASET A



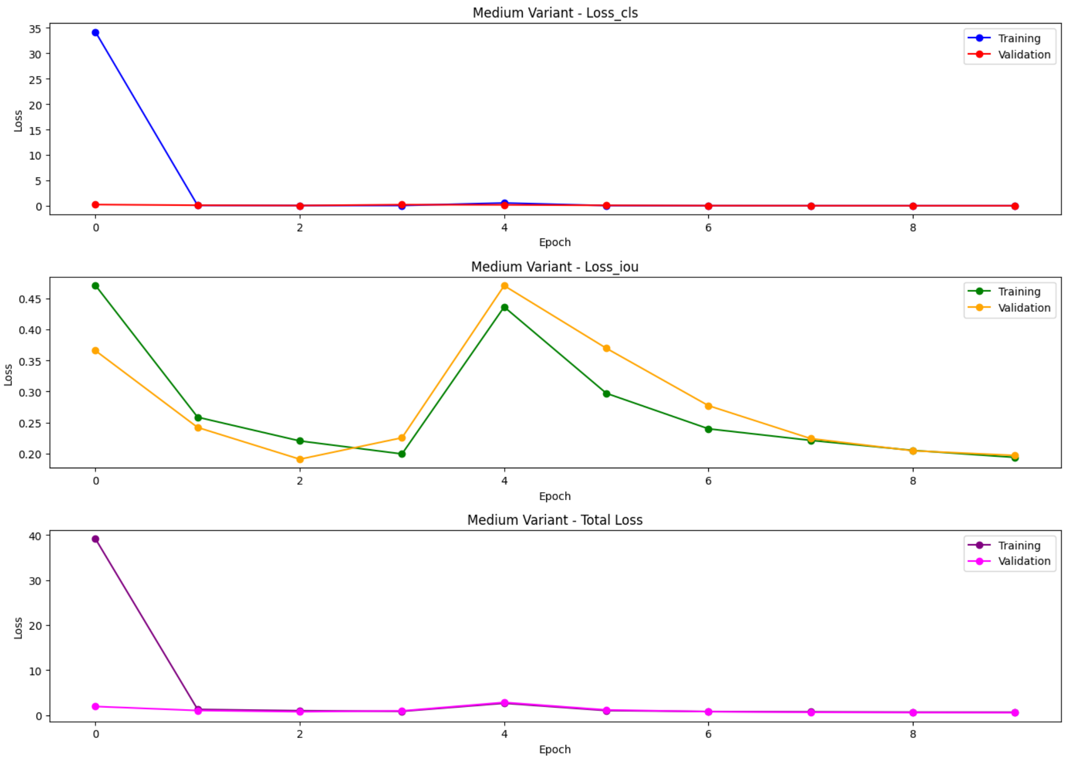
MIDDLE TENNESSEE STATE UNIVERSITY

SMALL MODEL - DATASET A

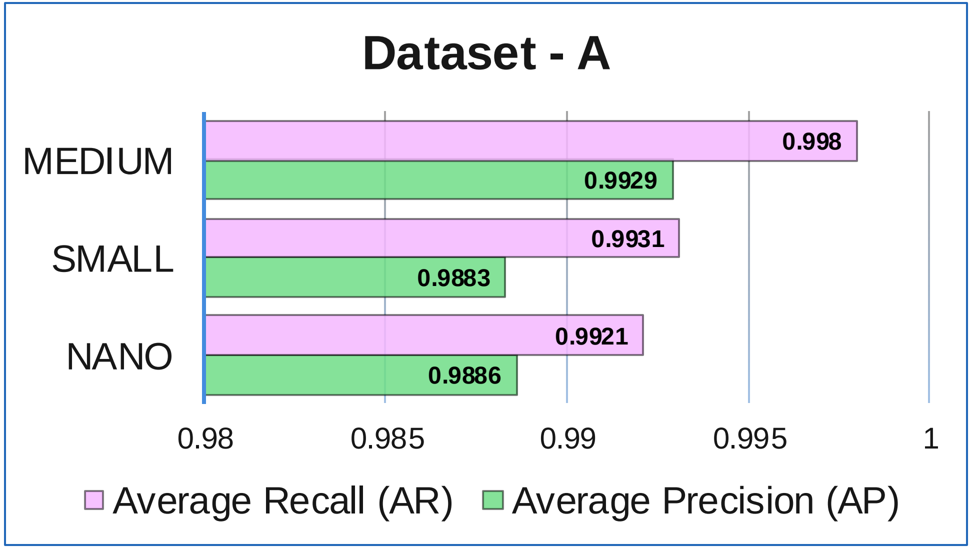


MIDDLE TENNESSEE STATE UNIVERSITY

MEDIUM MODEL - DATASET A

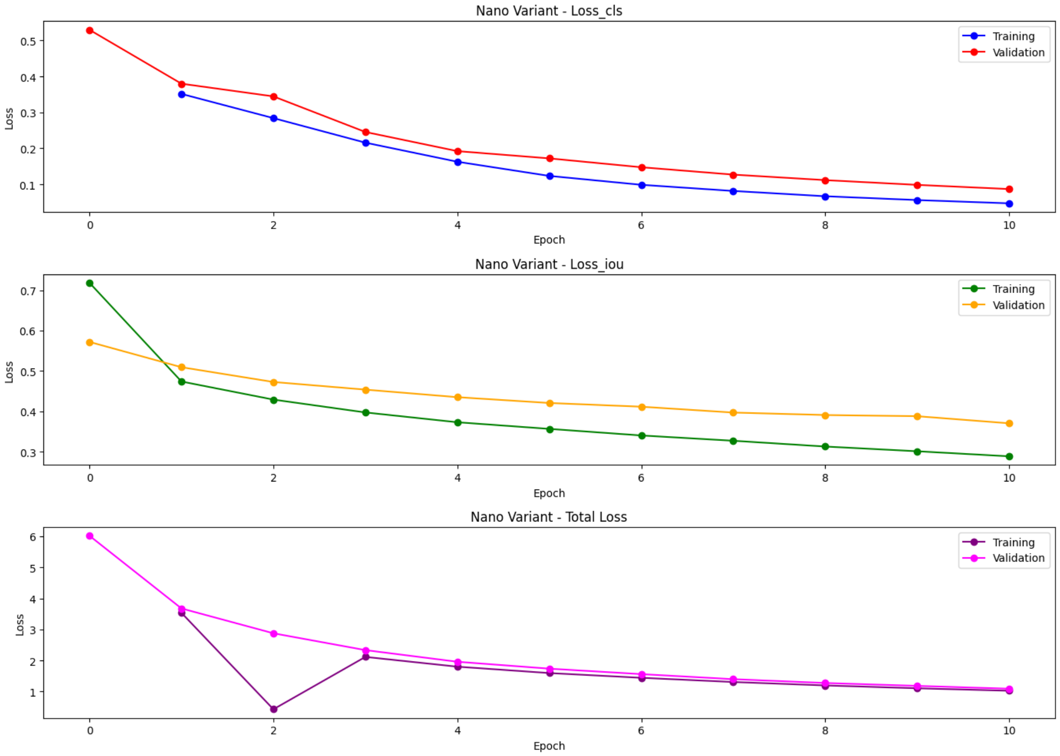


COMPARISON OF THREE VARIANTS OF THE MODEL ON DATASET - A

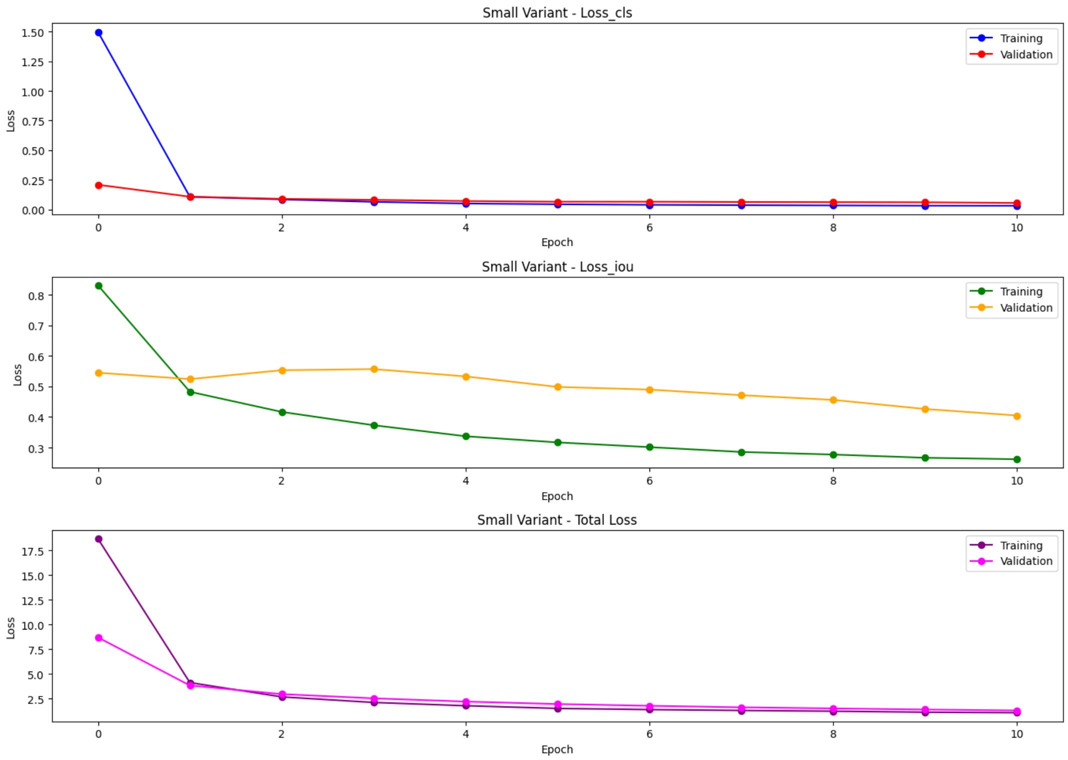


Model Variant	Average Precision (AP)	Average Recall (AR)
NANO	0.9886	0.9921
SMALL	0.9883	0.9931
MEDIUM	0.9929	0.998

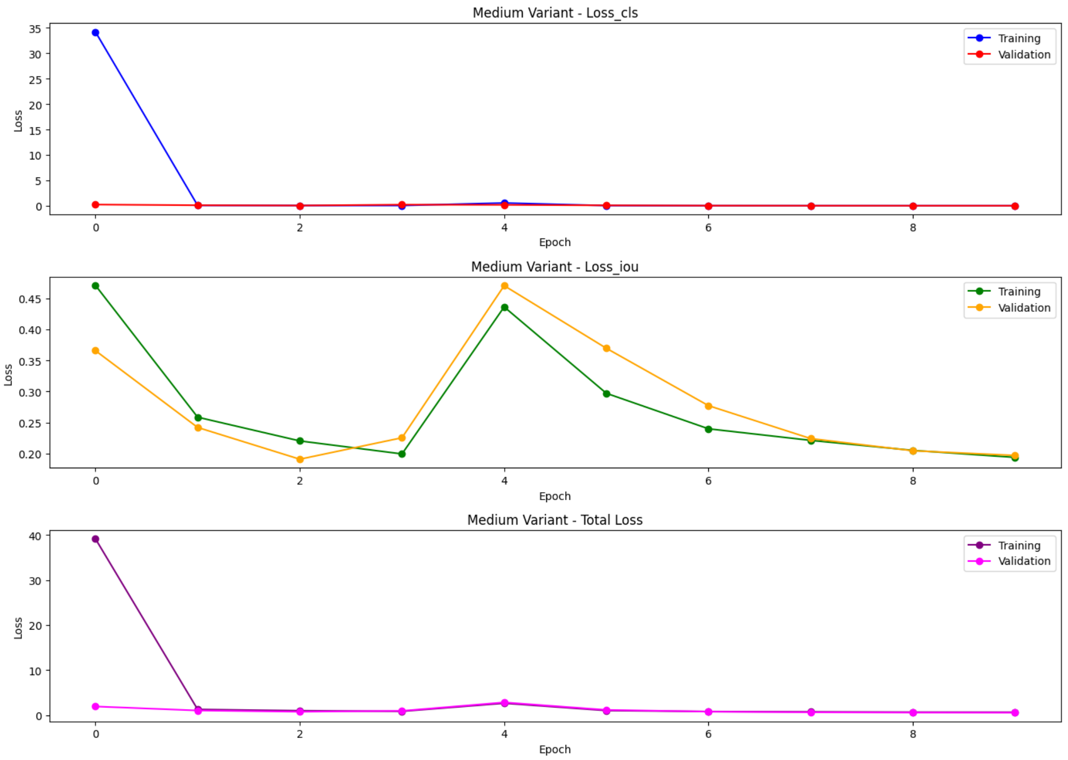
NANO MODEL - DATASET B



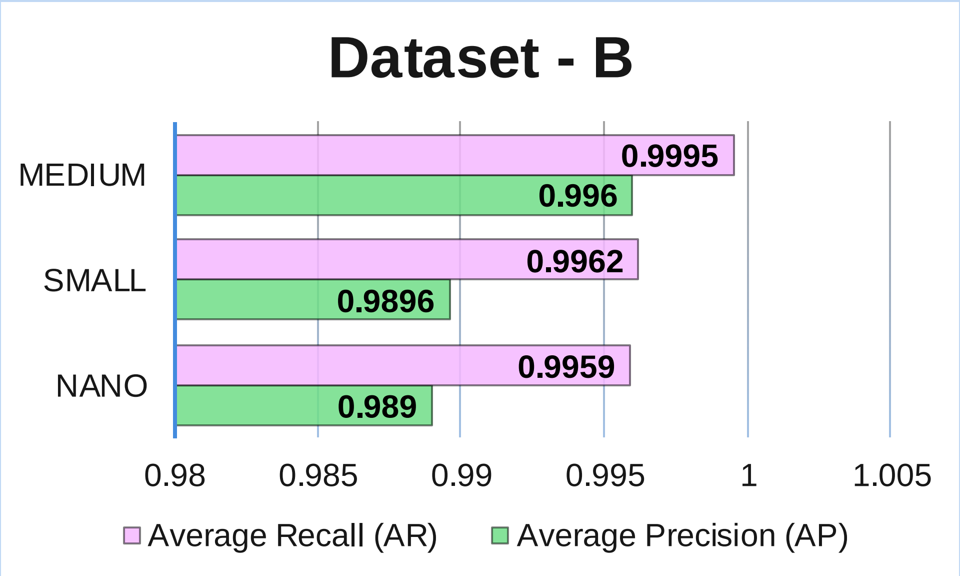
SMALL MODEL - DATASET B



MEDIUM MODEL - DATASET B



COMPARISON OF THREE VARIANTS OF THE MODEL ON DATASET - B



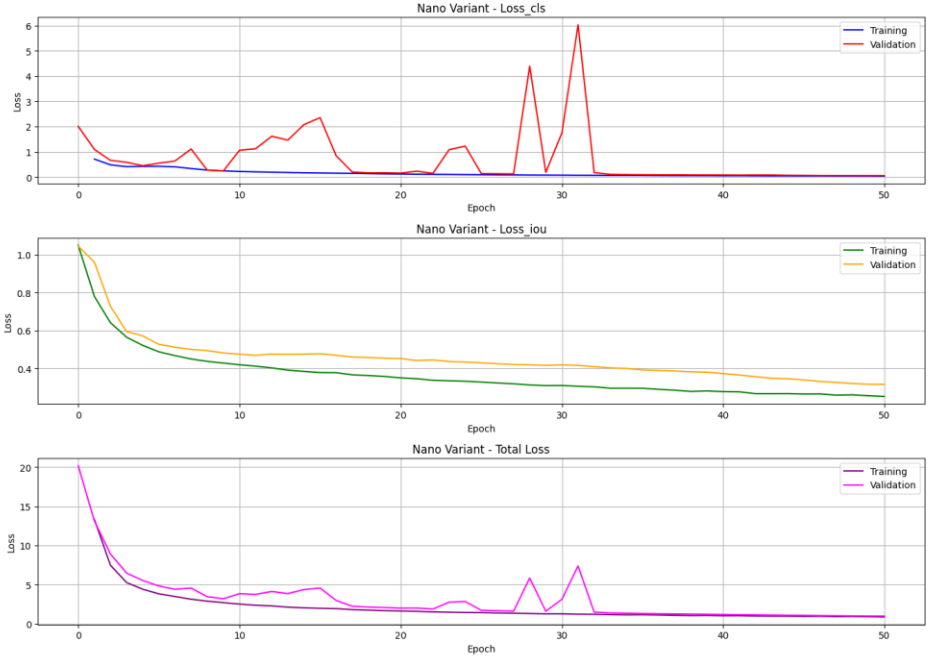
Model Variant	Average Precision (AP)	Average Recall (AR)
NANO	0.989	0.9959
SMALL	0.9896	0.9962
MEDIUM	0.996	0.9995

NANO VARIANT AS THE WINNER

- Since all the variants are showing almost same AP and AR so we choose the model that has least number of parameters.

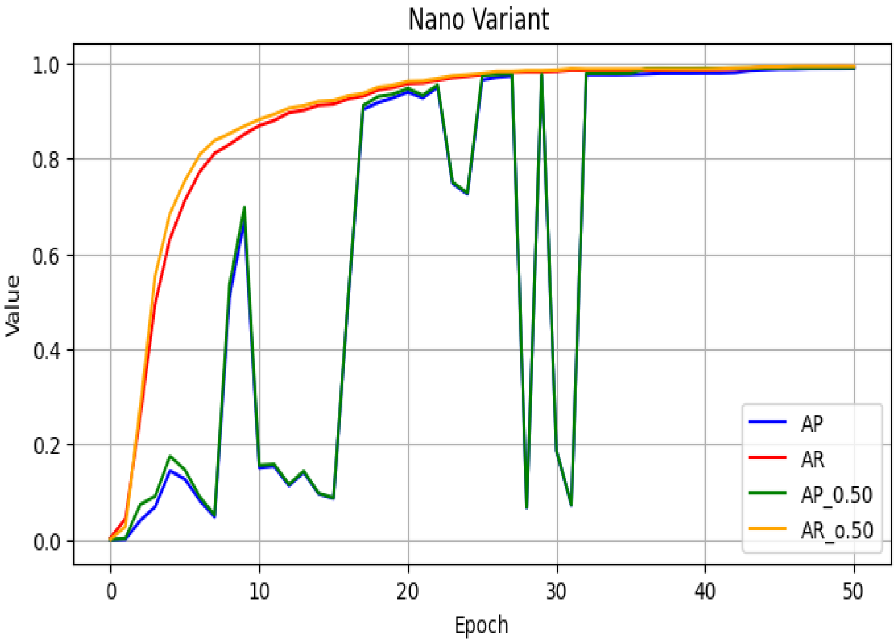
- From the results obtained on the datasets, nano variant is the best choice for real-time processing when we deploy it on edge devices.

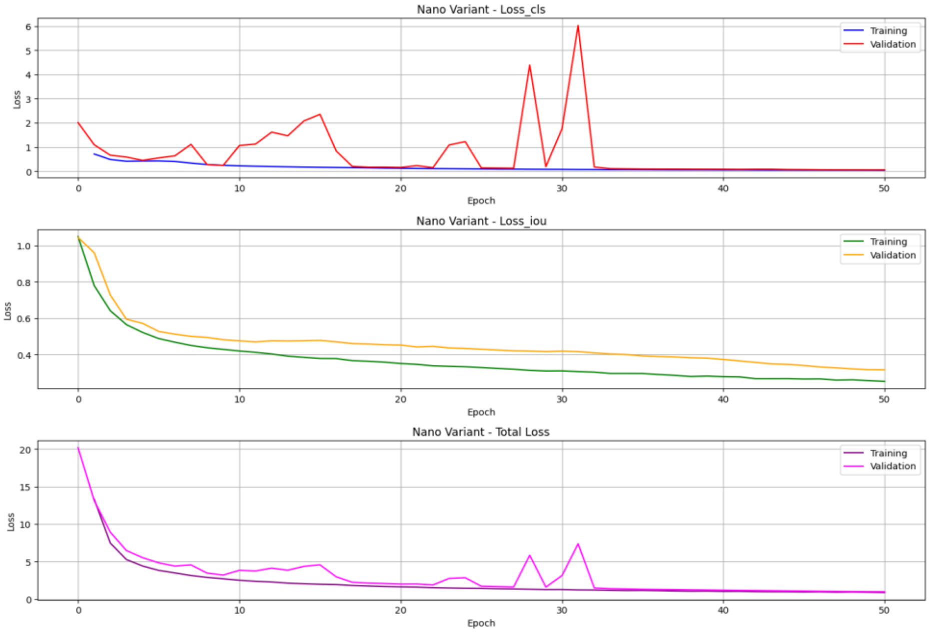
- Trained Nano variant for 50 Epochs.



MIDDLE TENNESSEE STATE UNIVERSITY

NANO VARIANT AS THE WINNER

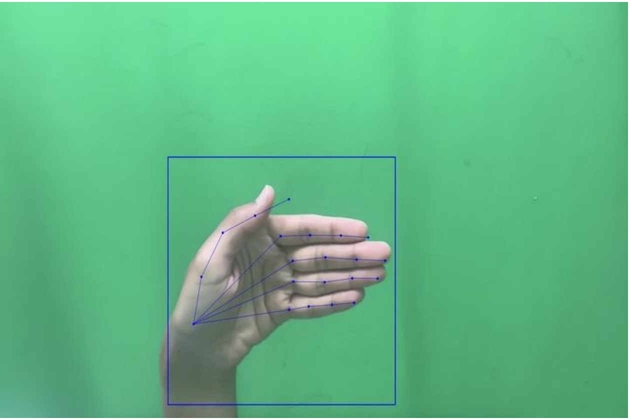
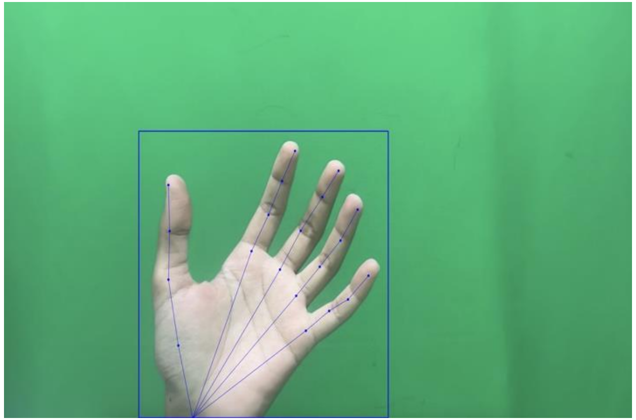
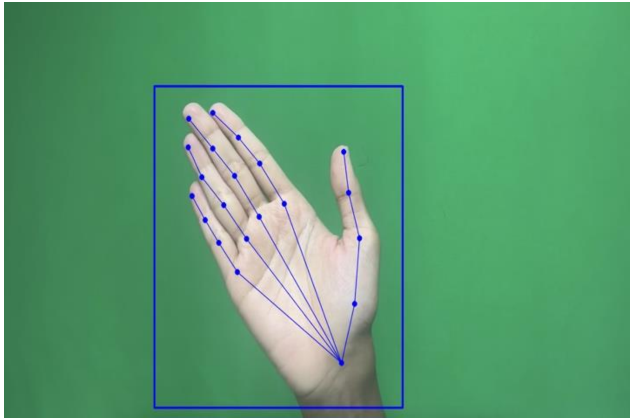
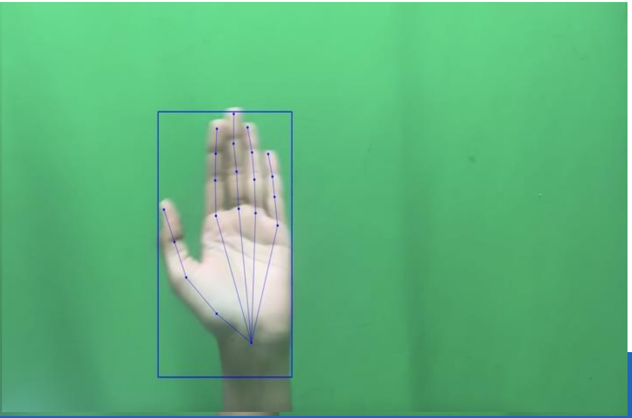
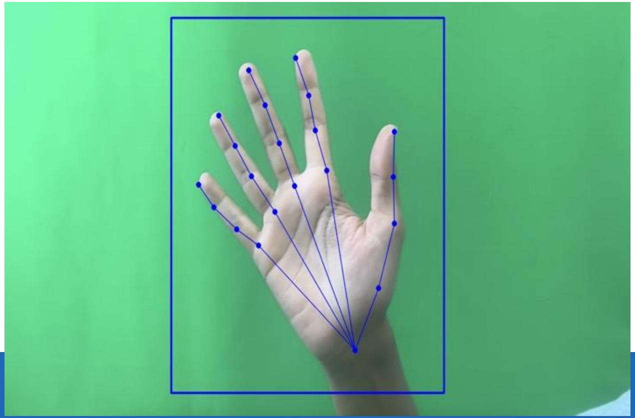
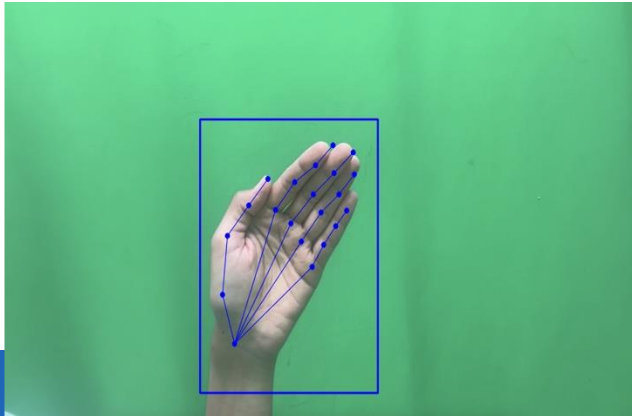




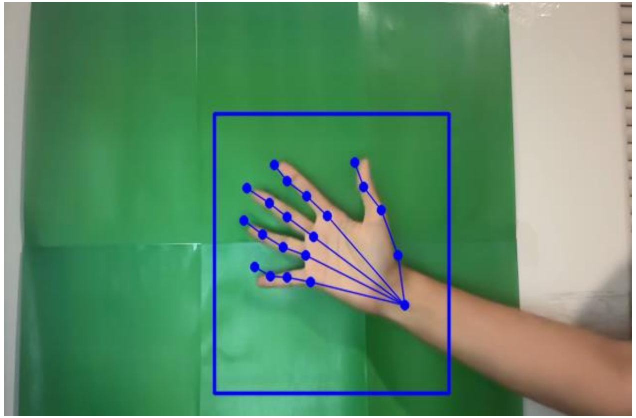
Model	AP	AR
Nano	0.9886	0.9921

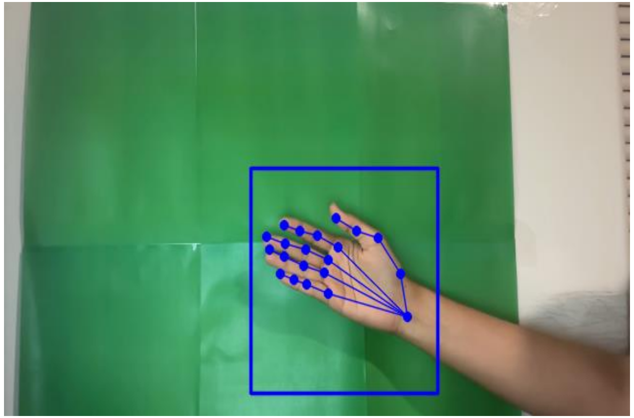
MIDDLE TENNESSEE STATE UNIVERSITY

Predictions made by model on Test Data

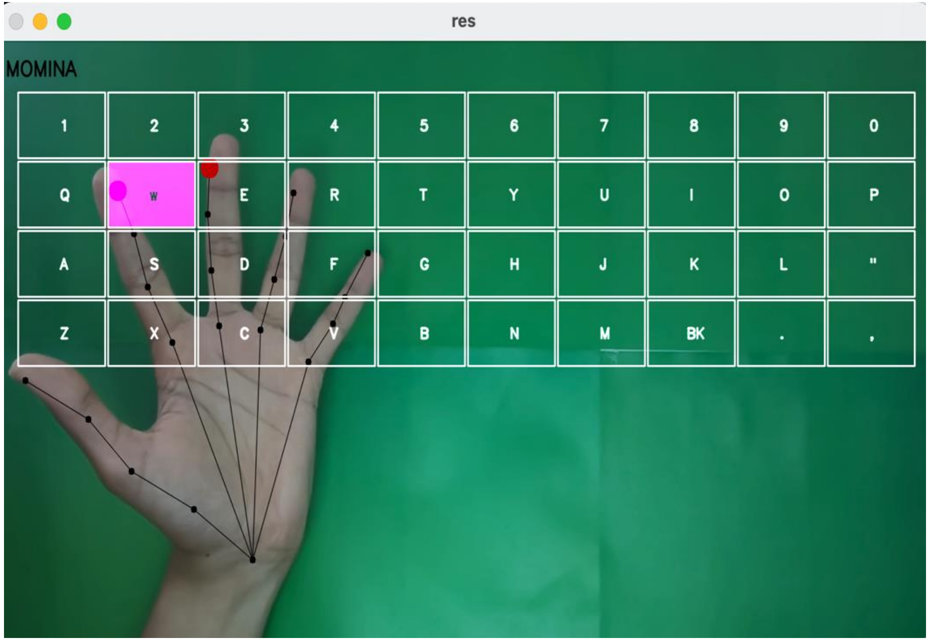
Predictions made by model on data with some degree of white background involved



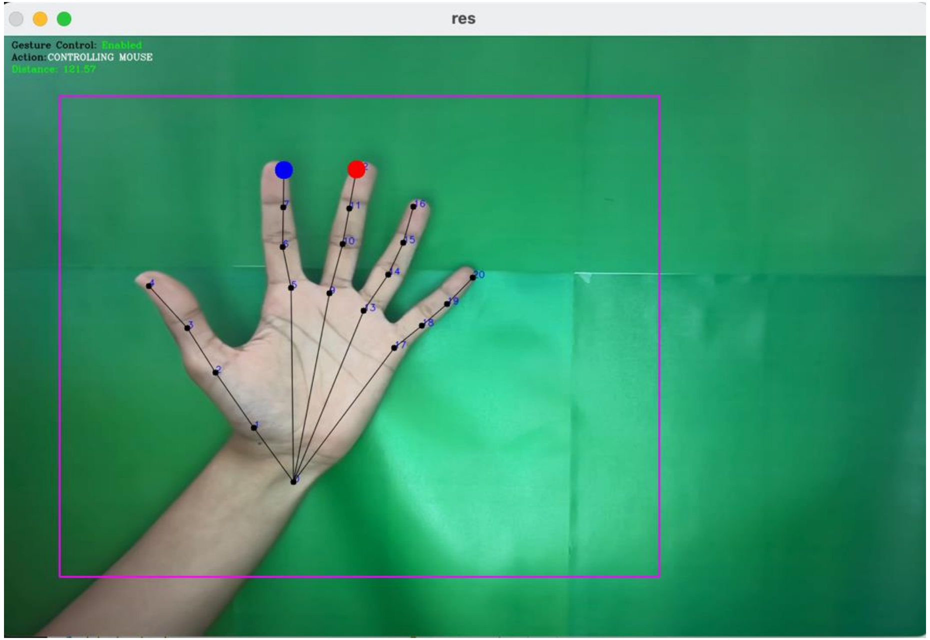


MIDDLE TENNESSEE STATE UNIVERSITY

Virtual Mouse and Keyboard in Action



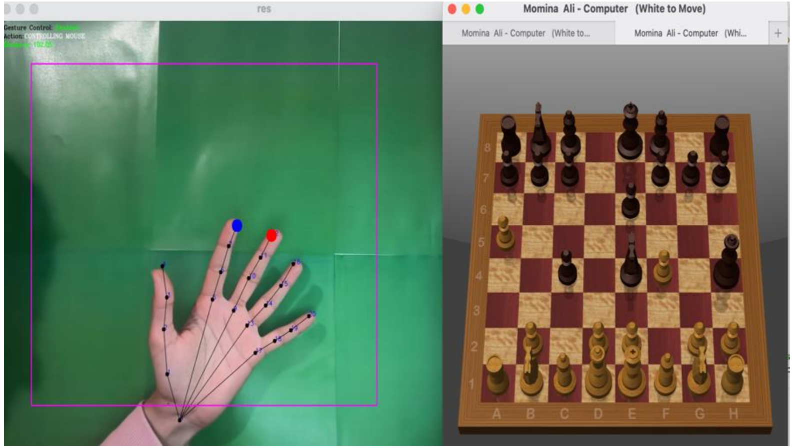
Virtual Keyboard



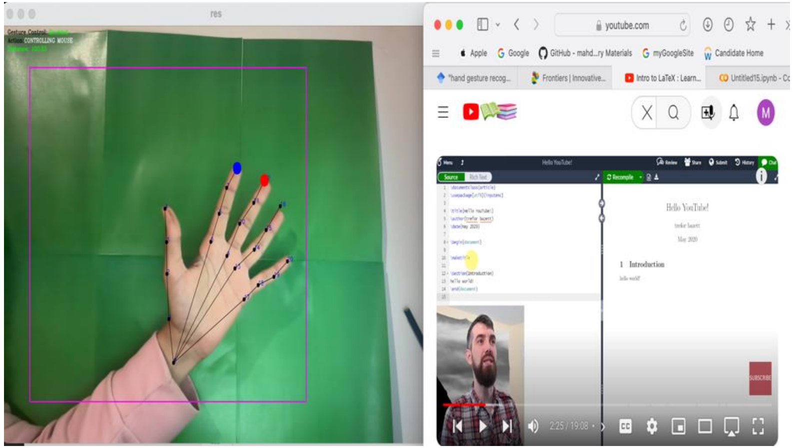
Virtual Mouse

MIDDLE TENNESSEE STATE UNIVERSITY

USE CASES



Virtual control for playing chess on computer



Remote control for media playback

06

CONCLUSION & FUTURE WORK

MIDDLE TENNESSEE STATE UNIVERSITY

CONCLUSION

FOCUS OF RESEARCH:

- Enhancing Human-Computer Interaction (HCI) in Virtual Reality (VR) by utilizing technology for gesture recognition and hand tracking

VIABILITY DEMONSTRATION:

- Use of YOLO models in a virtual Human Computer Interface demonstrates its practicality and increases user engagement in virtual reality settings.

REAL-WORLD APPLICATION:

- By connecting theory and practice, the use of virtual mouse and keyboard can greatly revolutionize the gaming and education industry.

LIMITATIONS

LESS DIVERSE DATA:

- We need to make dataset more diverse by including the images of backside of hand and involving other individuals after IRB approval.

HIGHLY DEPENDENT ON GREEN OR GRAY BACKGROUND:

- Currently, the system is high dependency on background color and if the background is changed the model gives false detections.

DELAYED RESPONSE TO THE REAL-TIME HAND GESTURE:

- The model is not really quick in detecting the posture and responding to it. **Solution: PTQ**

FUTURE WORK

SUSTAINED IMPROVEMENT:

- More precision and versatile hand gestures will be added.

IMPROVING VIRTUAL EXPERIENCE:

- To increase the frame-rate to give user a feeling of immersive control of mouse cloud interface will be deployed and parallel processing of the frames will be achieved.

VIRTUAL ELECTRICAL LAB ENVIRONMENT:

- We aim to extend this project to create a virtual simulation environment for the engineering students to use lab equipment virtually.

THANK YOU!