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

Gesture Recognition

03

Natural HCI Design

04

Application in Virtual Electrical Power Lab

05

Conclusion & Future Work

01

INTRODUCTION

MIDDLE TENNESSEE STATE UNIVERSITY

INTRODUCTION

- Hand Tracking & Gesture Recognition
- Human Computer Interaction in Virtual Reality
- Challenges
- YOLO Based Solution

MIDDLE TENNESSEE STATE UNIVERSITY

INTRODUCTION

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.
- Applications go beyond education to industries like automobile and healthcare.

INTRODUCTION

HCI in Virtual Reality

VR Systems usually consist of 5 elements and three layers.

- VR system consists of:
  - VR Engine
  - Software & Database
  - Input/Output Devices
  - Users
  - Tasks
- VR system unfolds across:
  - System Layer
  - Middle Layer
  - Application Layer

INTRODUCTION

Challenges

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.
- **Adaptability & seamless Design:**
  - Maintaining platform compatibility and user-friendliness while smoothly integrating hand tracking into a variety of applications

INTRODUCTION

YOLO

YOLO based gesture recognition system.

- For accurate gesture recognition, we used YOLO architecture.
- Using the object detection feature of YOLO to accurately recognize and comprehend hand gestures in a variety of settings.

MIDDLE TENNESSEE STATE UNIVERSITY

01

LITERATURE REVIEW

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

Pose Estimation Efficient Frontier | COCO | 4th Generation Intel Xeon CPU

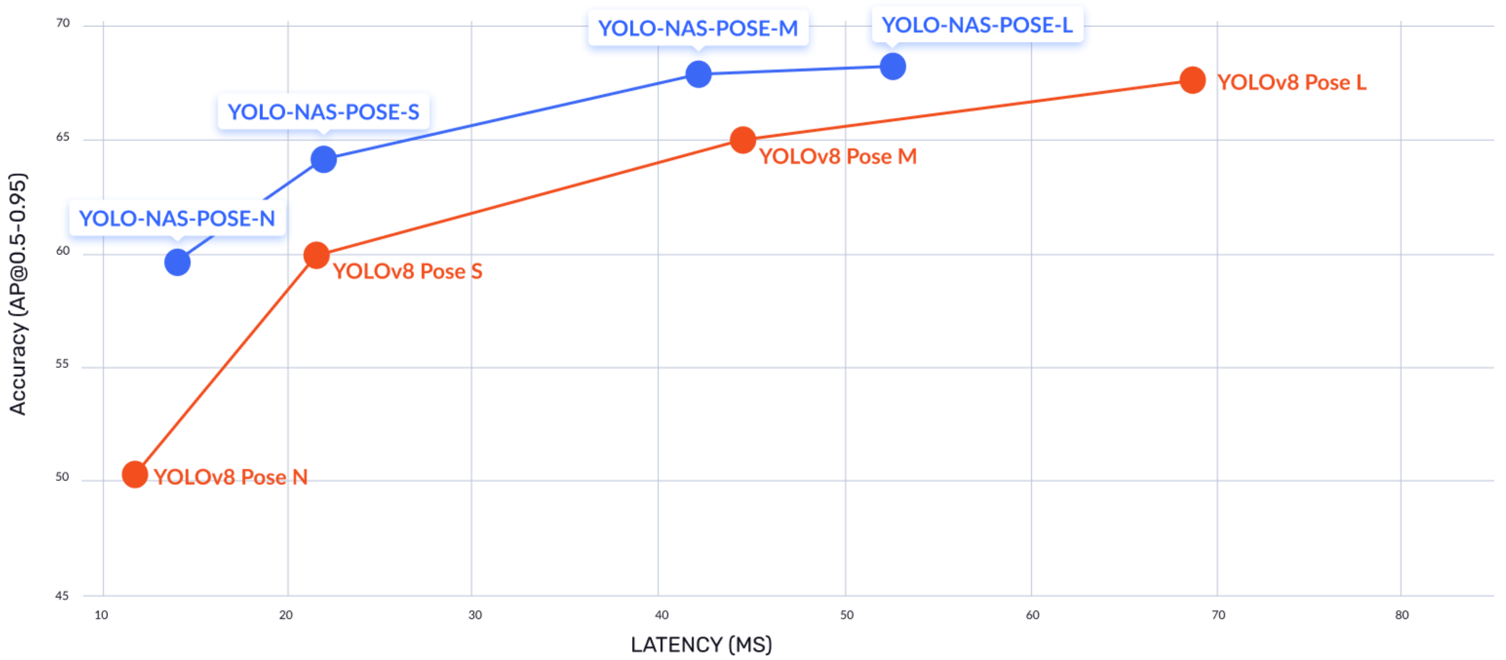


Image Credits: https://www.linkedin.com/pulse/8-community-created-content-get-started-yolo-nas-pose-deciai-omguc/

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.)
  - Transfer Learning (Savas et al.)

Posture Estimation

- Traditional Gesture Recognition Techniques
  - sEMG with CNN (Wang et. al)
  - Kinetic Sensors with DNN (Tang et al.)
  - DNN with Residual Connections (Bonab et al.)

02

GESTURE RECOGNITION

MIDDLE TENNESSEE STATE UNIVERSITY

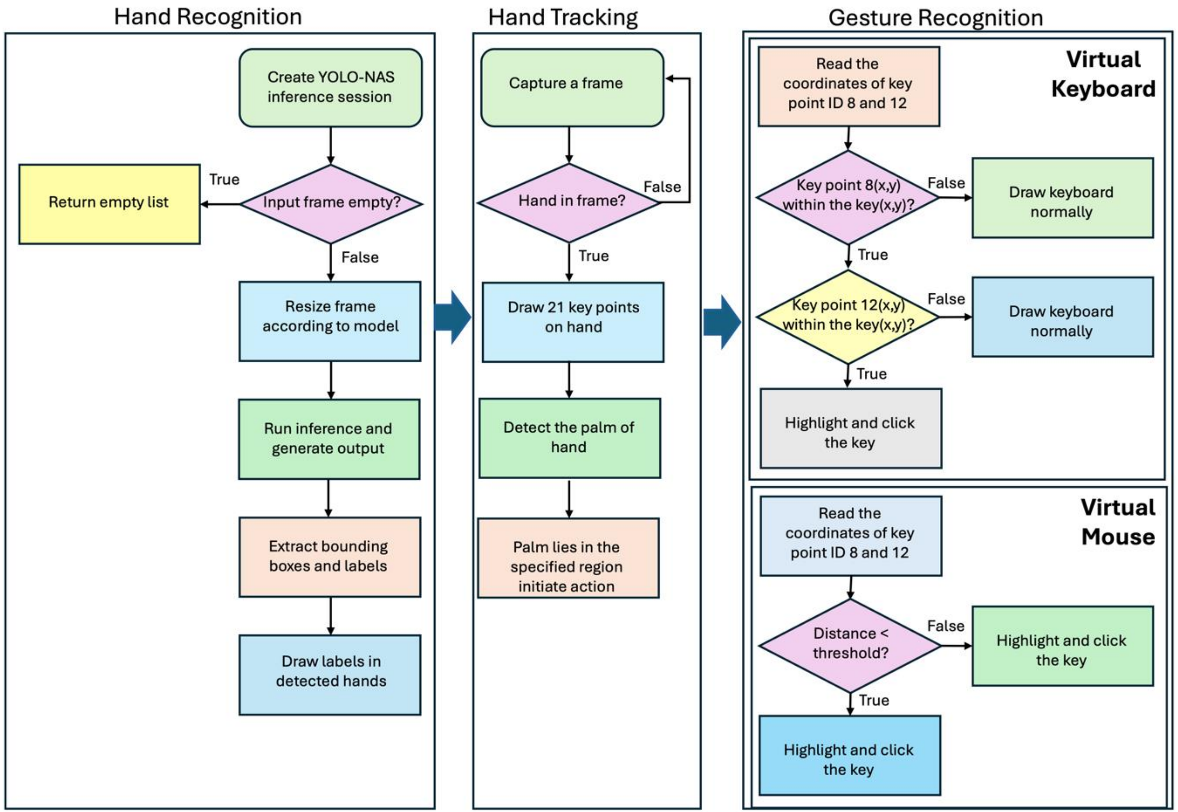
THREE – STEP HAND GESTURE RECOGNITION

01 HAND RECOGNITION

02 HAND TRACKING

03 HAND GESTURE RECOGNITION

THREE – STEP PROCESS



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

MIDDLE TENNESSEE STATE UNIVERSITY

03

NATURAL HCI DESIGN

GESTURE RECOGNITION IMPLEMENTATION

01 DATA COLLECTION & PRE-PROCESSING

02 GENERATING ANNOTATIONS

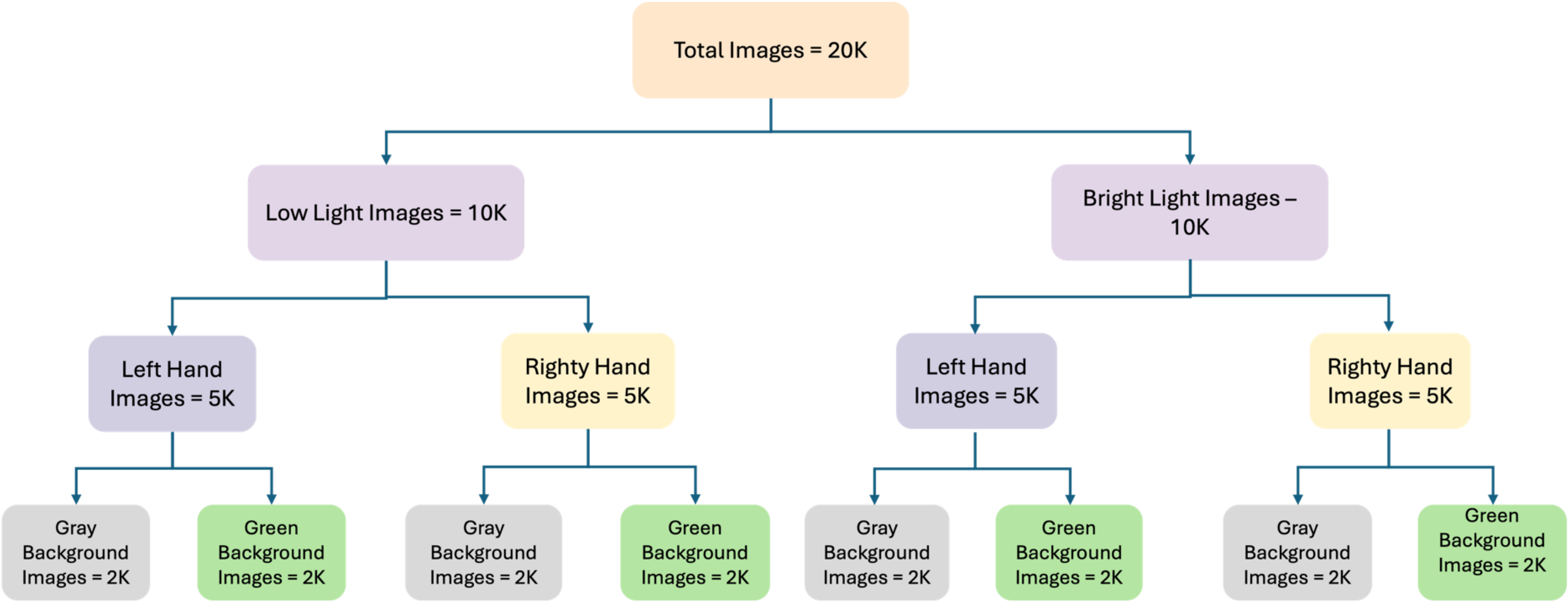
03 MODEL TRAINING & FINE TUNING

DATA COLLECTION & PRE-PROCESSING

- Gathered data using webcam.
- Each image was of 2666x1488 pixels.
- 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.

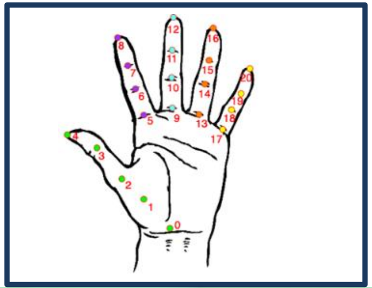
Sample Images from Dataset

DATASET CONSTRUCTION TREE

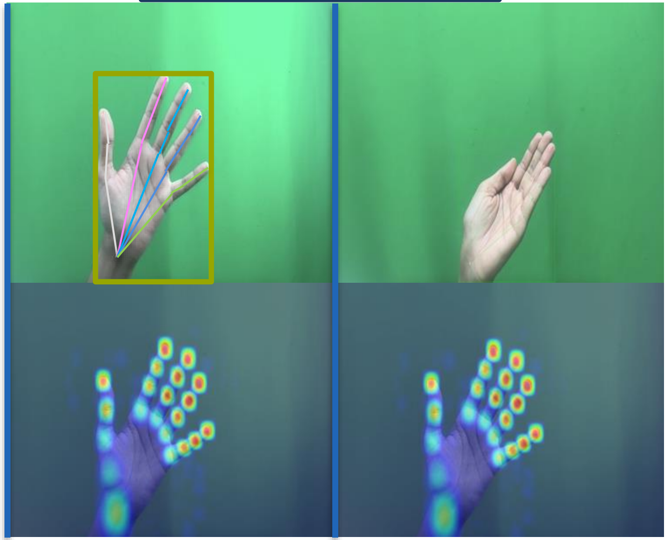


GENERATING ANNOTATIONS

- Why not manual annotation?
- 21 key-points on human hand were annotated.
- MMPose uses RTMDet which is trained on 4 different 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 converted to json format.



HAND LANDMARKS

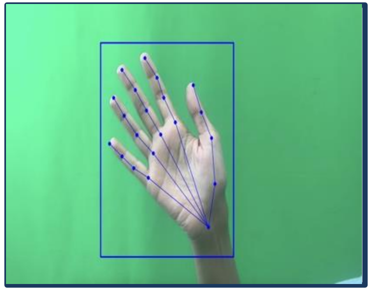


Original Image

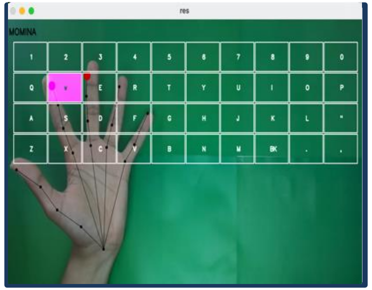
Annotated Image

MODEL TRAINING & FINE TUNING

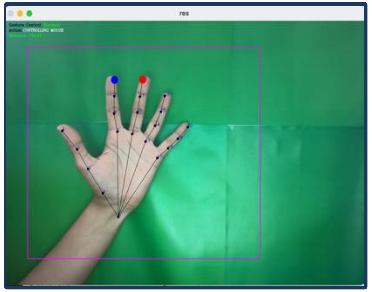
- Used YOLO-NAS Pose; a sibling model of YOLO-NAS.
- Famous model because of its capability of being a single-stage detector which makes it fast in real-time applications.
- YOLO-NAS Pose performs both detection and estimation of Pose in single pass.
- YOLO-NAS Pose is trained on COCO2017 Dataset.
-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



HAND LANDMARKS



KEYBOARD CONTROL



MOUSE CONTROL

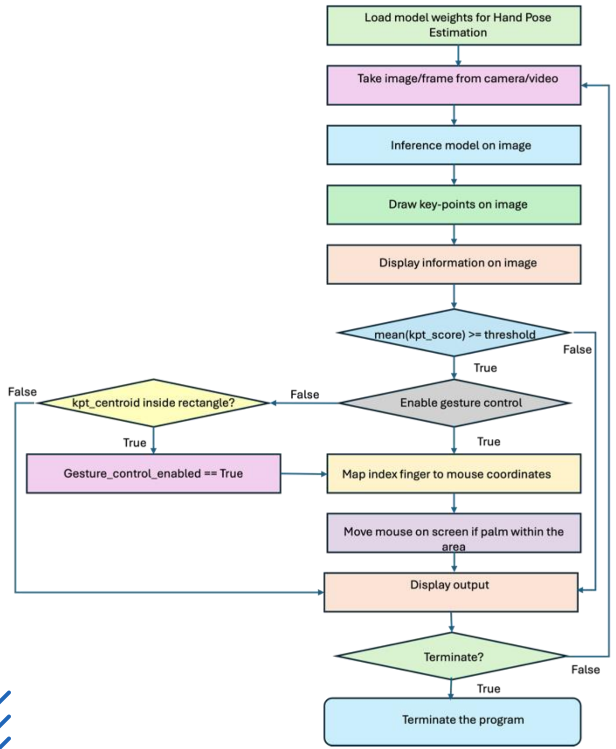
Model and System Details

- $r = max(\frac{640}{h}, \frac{640}{w})$
- $Updated\_height = h * r$
- $Updated\_width = w * r$
- $UpdatedKeypoint_i = originalKeypoint * r$
- $UpdatedBBox_i = originalBBox_i * r$

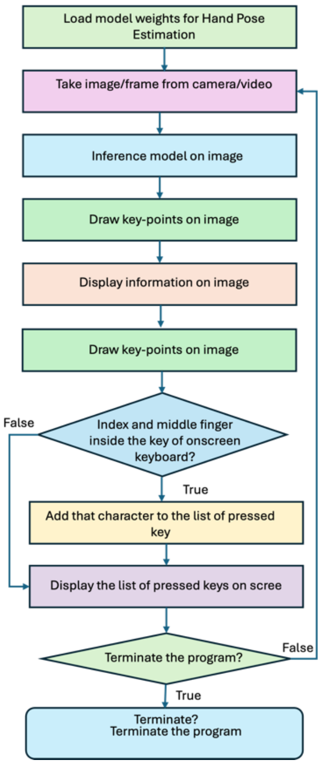
Configurations	Value
Epochs	10
Learning Rate	0.001
Optimizer	AdamW
Weight Decay	0.000001
Batch size	32
Iterations per Epoch	439

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

VIRTUAL MOUSE



VIRTUAL KEYBOARD



04

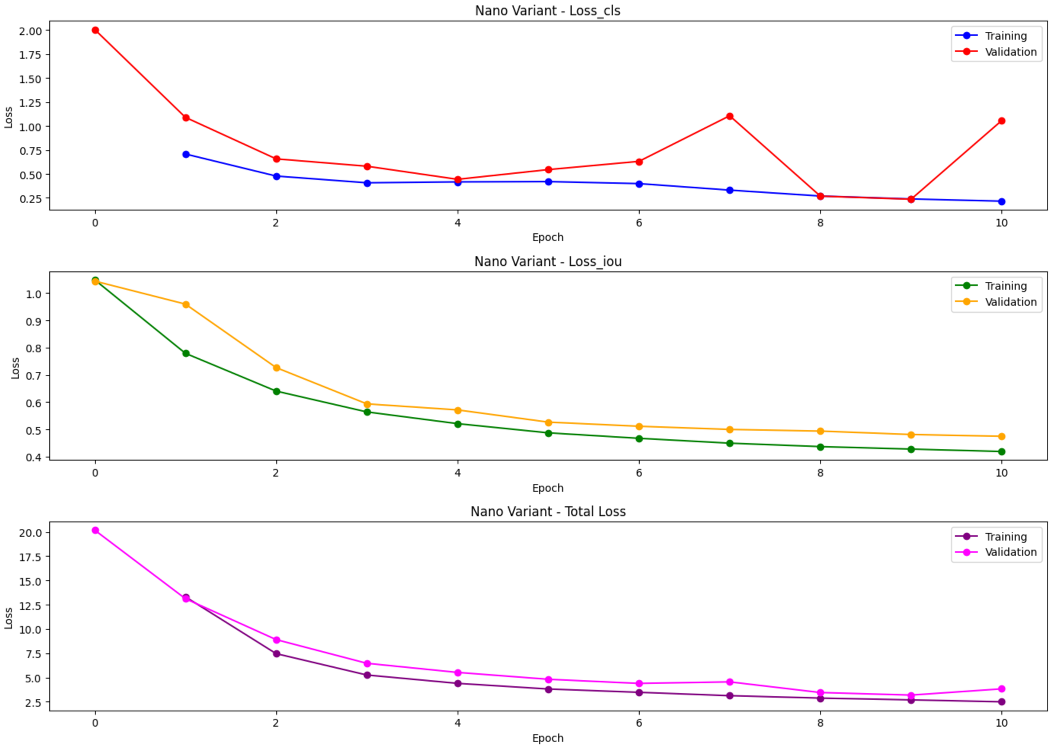
RESULTS

RESULTS

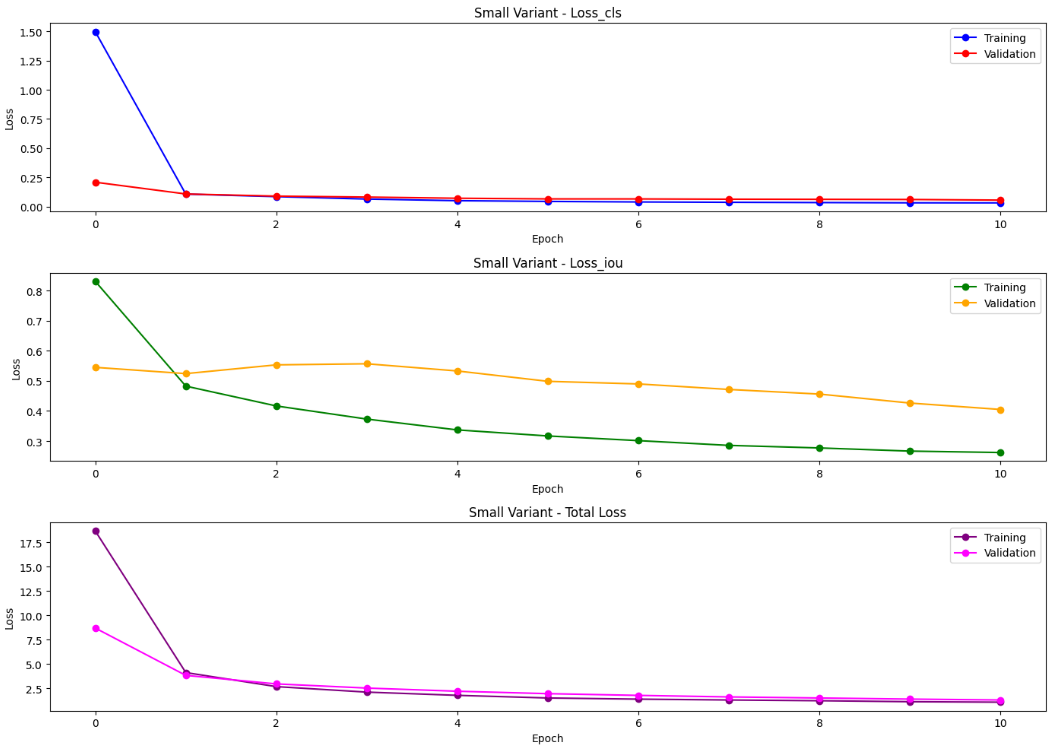
- We created two instances of the dataset one had 5k images and named it as Dataset A and other had 20k images and names it as Dataset B.
- Each of these models were trained for 10 Epochs on both datasets A and B.
- We monitored AP and AR on each epoch and computed the Total loss, classification loss and IOU loss

Model	No. of Parameters
Nano	9,901,483
Small	22.2 million
Medium	58.2 million

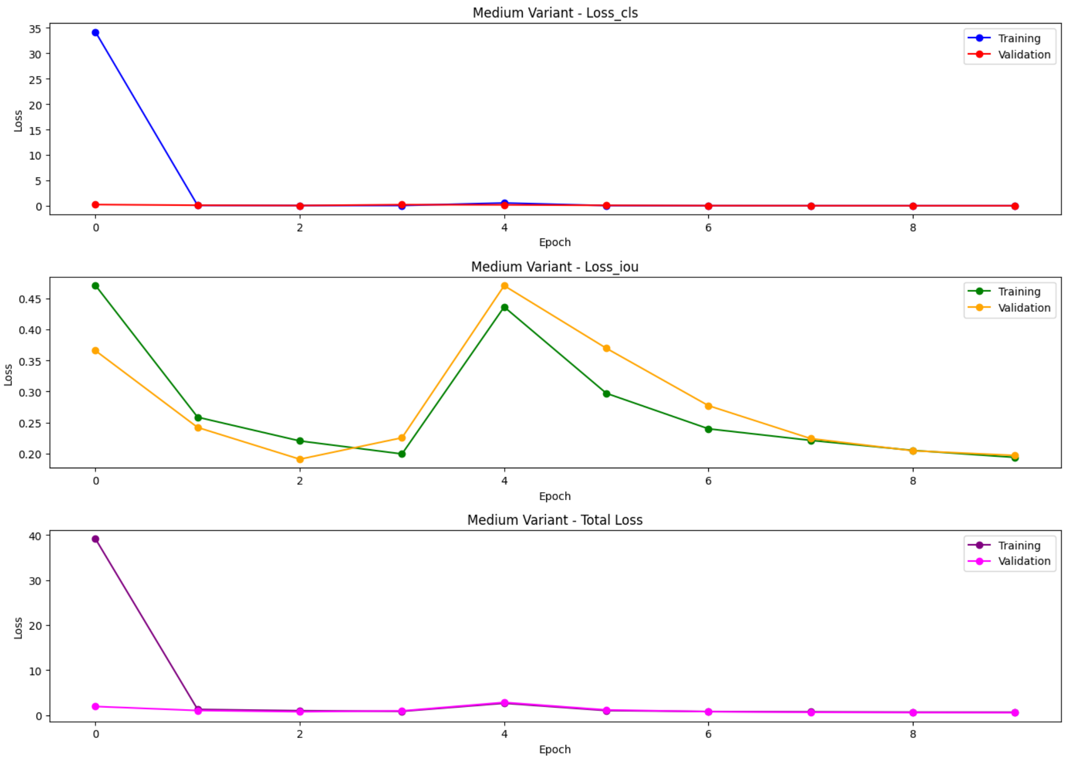
NANO MODEL - DATASET A



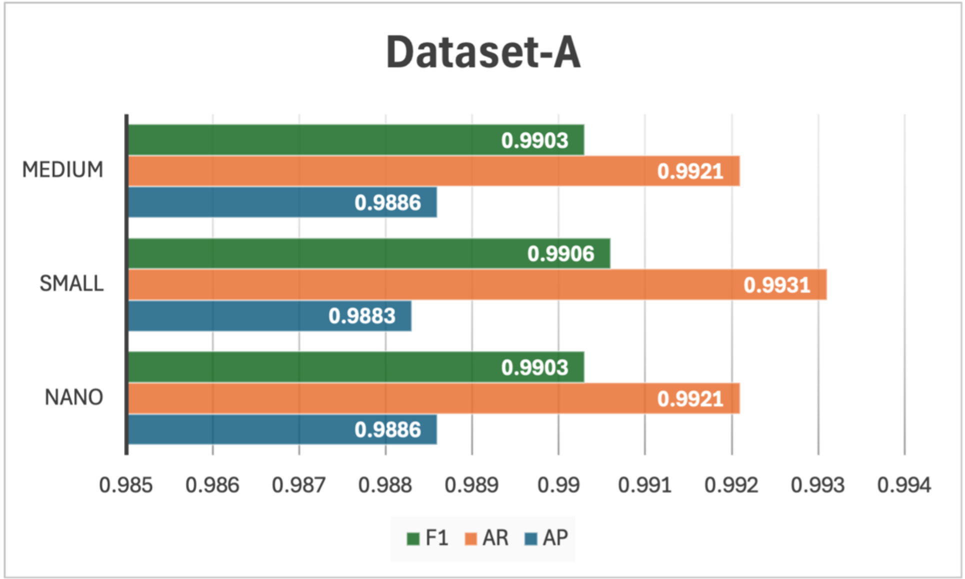
SMALL MODEL - DATASET A



MEDIUM MODEL - DATASET A

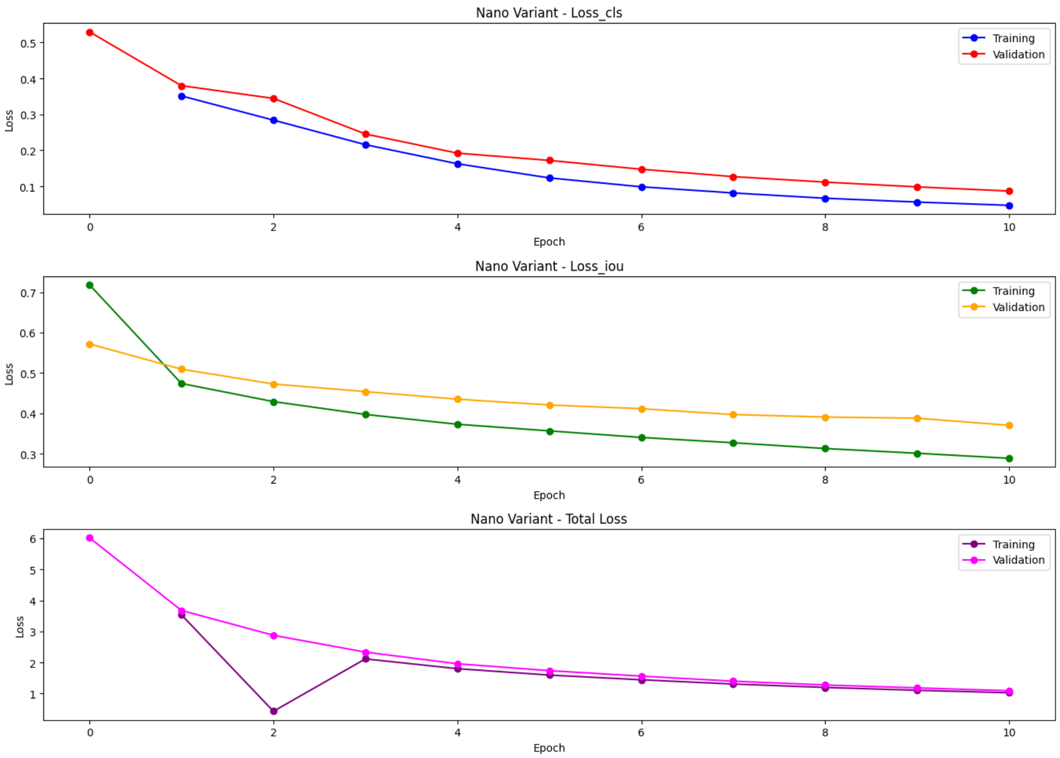


COMPARISON OF THREE VARIANTS OF THE MODEL ON DATASET - A

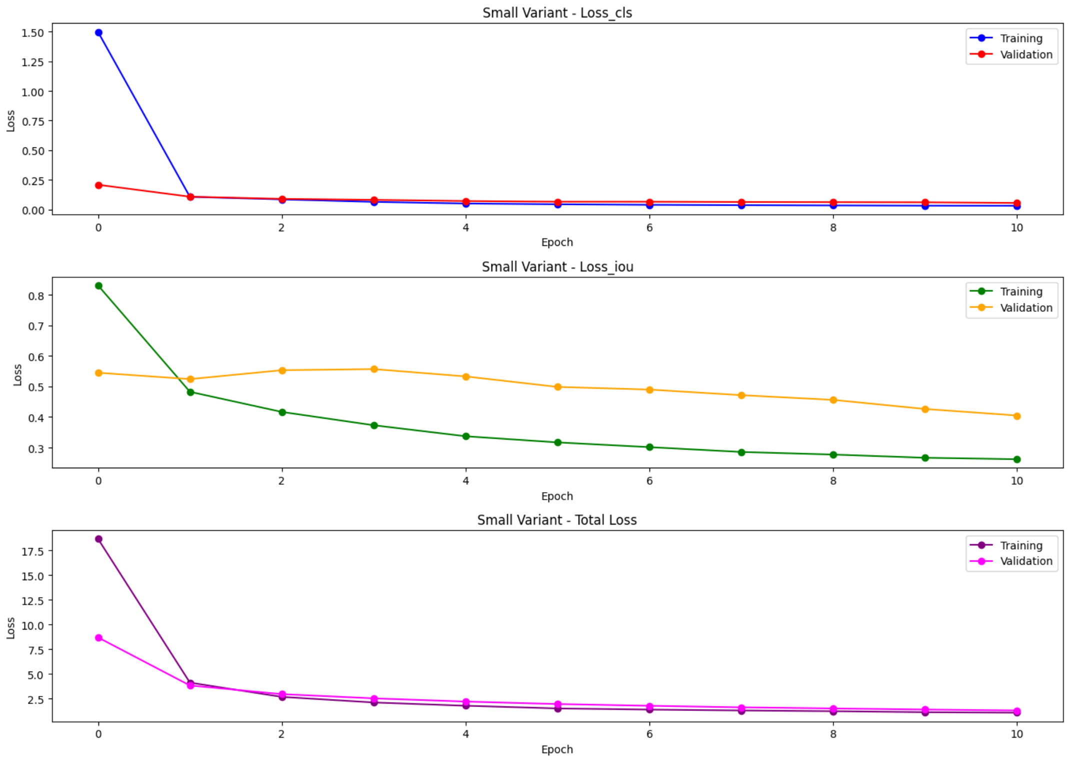


Model Variant	AP	AR	F1
NANO	0.9886	0.9921	0.9903
SMALL	0.9883	0.9931	0.9906
MEDIUM	0.9886	0.9921	0.9903

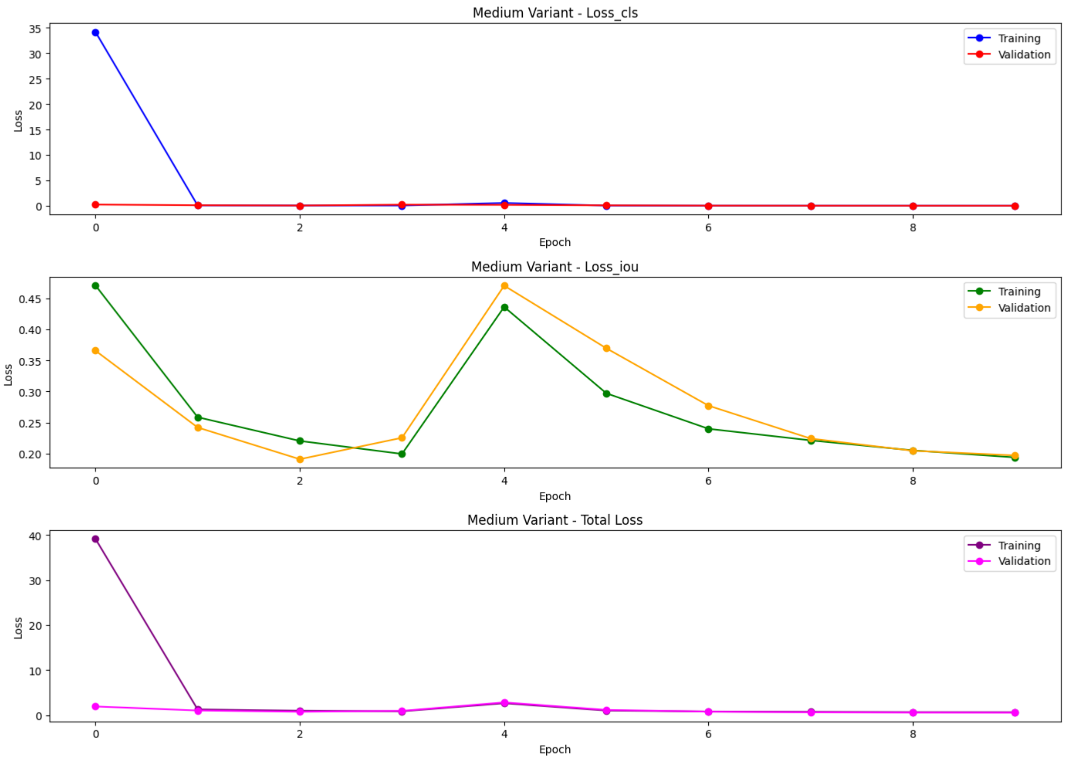
NANO MODEL - DATASET B



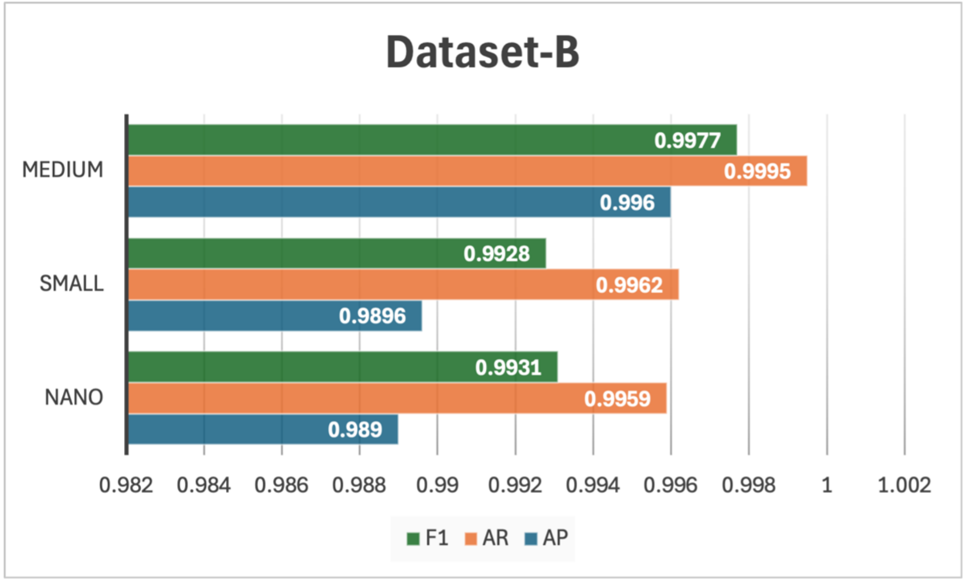
SMALL MODEL - DATASET B



MEDIUM MODEL - DATASET B



COMPARISON OF THREE VARIANTS OF THE MODEL ON DATASET - B

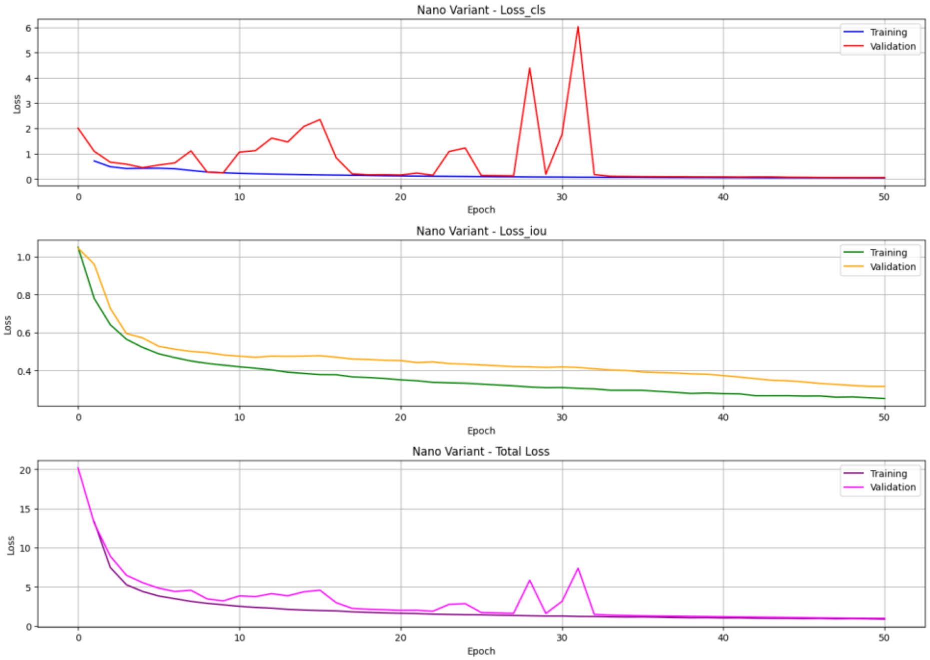


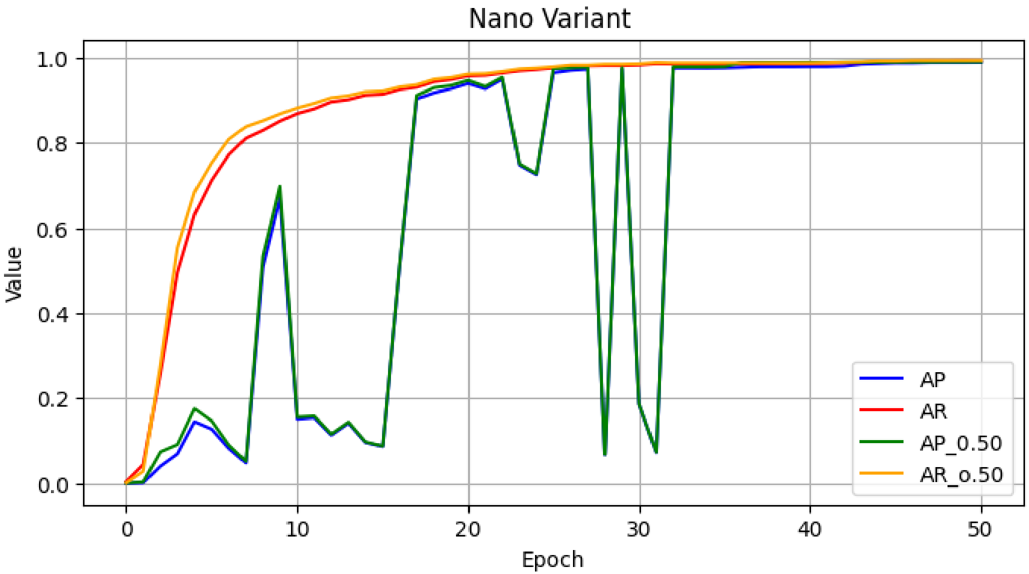
Model Variant	AP	AR	F1
NANO	0.989	0.9959	0.9931
SMALL	0.9896	0.9962	0.9928
MEDIUM	0.996	0.9995	0.9977

NANO VARIANT AS THE WINNER

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

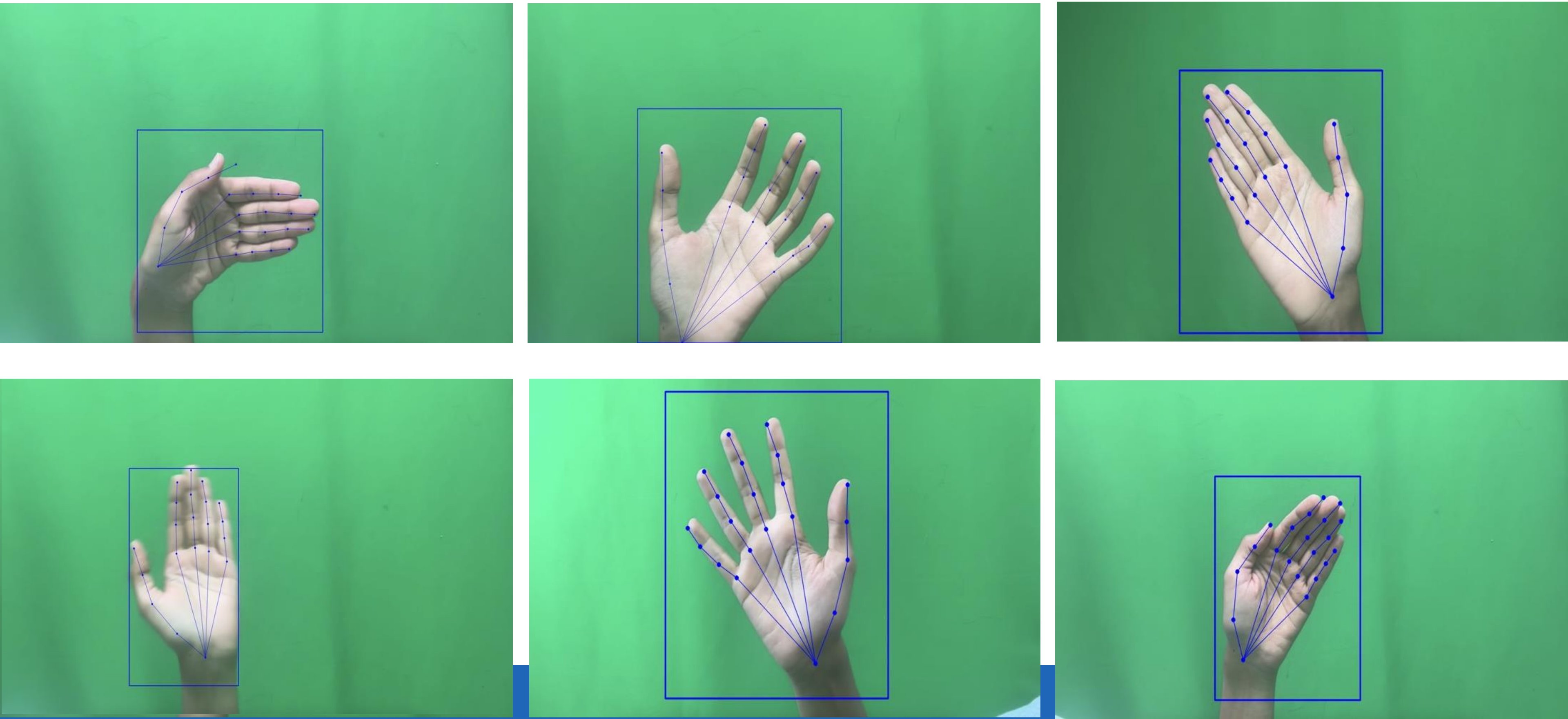
- We then trained Nano variant for 50 Epochs and we saw clear results of less overfitting when considering AP and AR metrices.



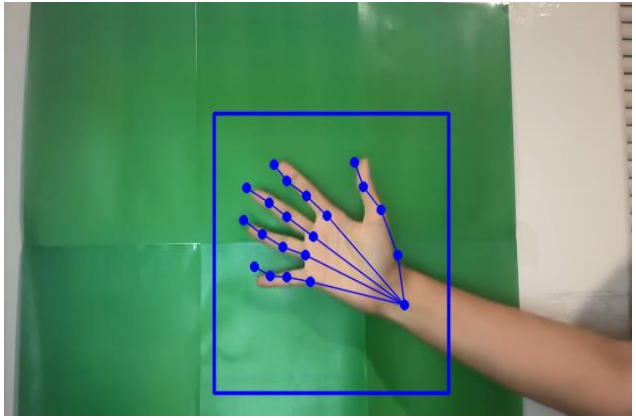


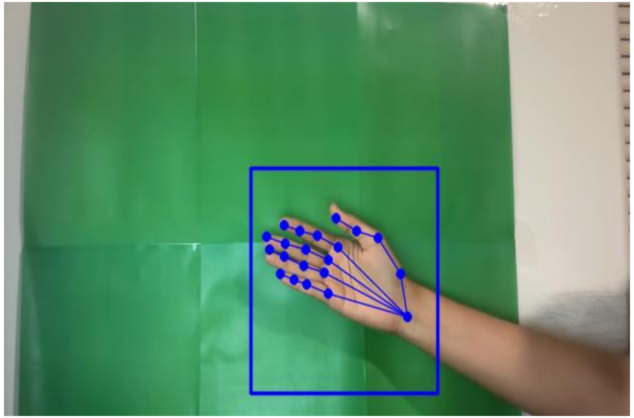
Model	AP	AR	F1
Nano	0.9886	0.9921	0.9903

Predictions made by model on Test Data

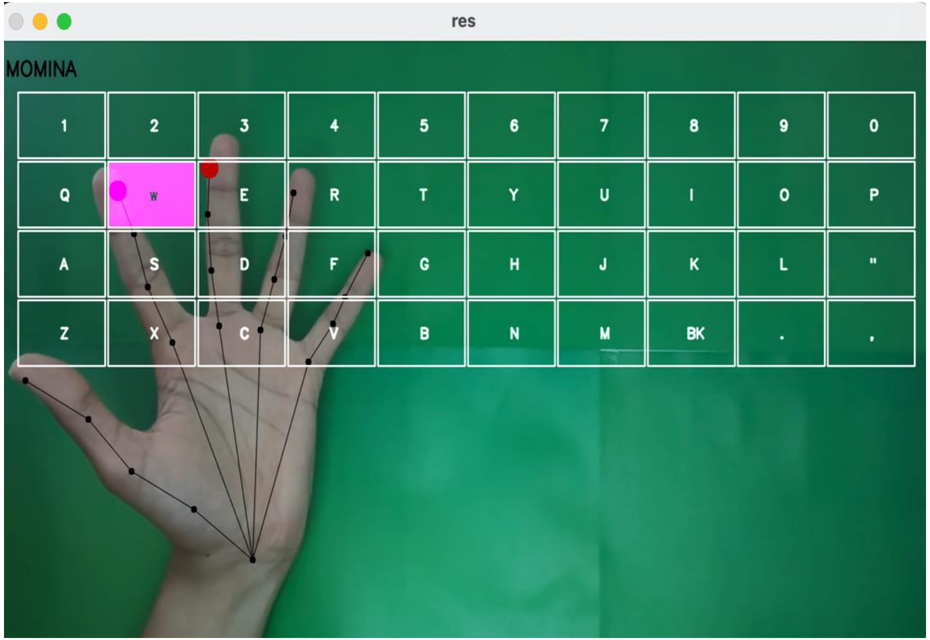


Predictions made by model on data with some degree of white background involved

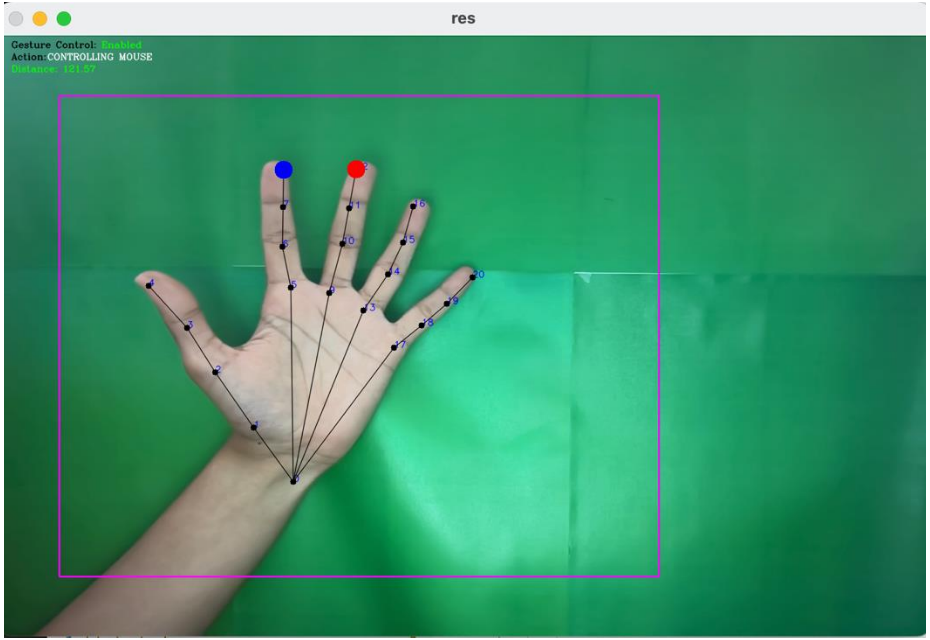




Virtual Mouse and Keyboard in Action

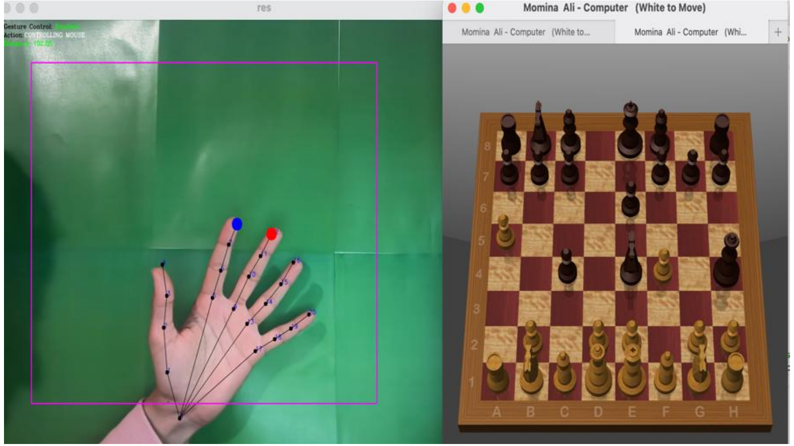


Virtual Keyboard

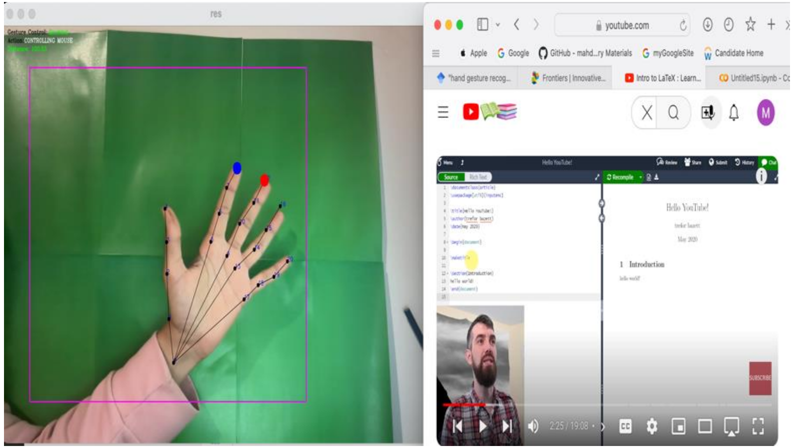


Virtual Mouse

USE CASES



Virtual control for playing chess on computer



Remote control for media playback

04

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.

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 made better and parallel processing of the frames will be achieved.

MIDDLE TENNESSEE STATE UNIVERSITY

THANK YOU!