“Integrating YOLOv8 and Media-Pipe for Accurate Pose Detection and Replication in Unity 3D Environments”

Raja Ali Akhtar, Hira Yaseen, and Tahira Mahboob*
*Corresponging author

Abstract

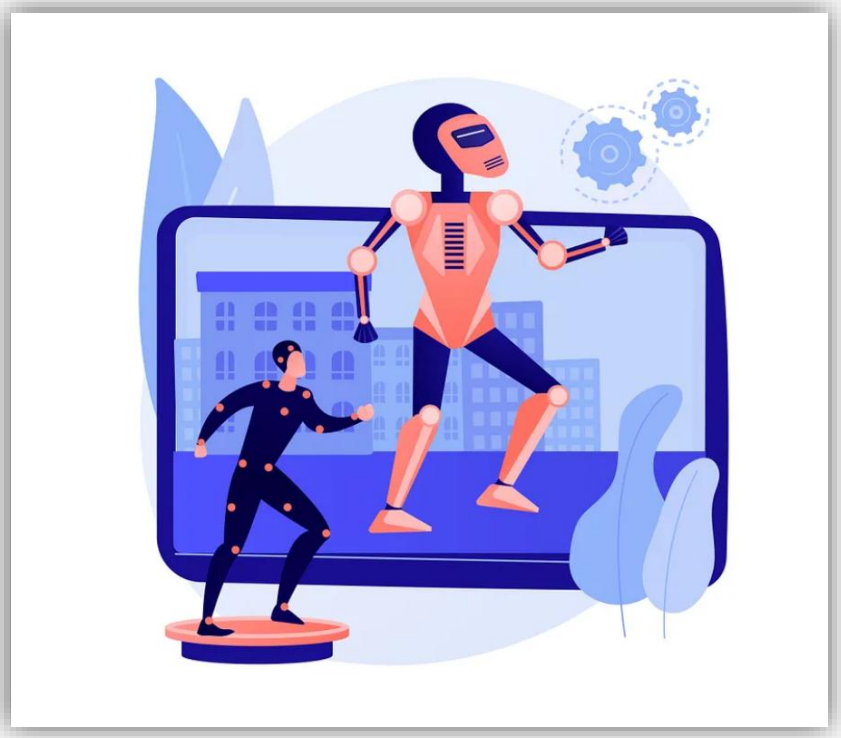
- This research addresses the challenge in virtual character animation: seamless motion replication by recording human movements.
- Fine-tuning YOLOv8 for real-time pose detection using a custom dataset.
- Comparative analysis of fine-tuned YOLOv8 and pretrained MediaPipe model in OpenCV for pose detection.
- Integration with Unity 3D for automatic motion replication.
- Applications: virtual reality experiences, gaming, motion capture.

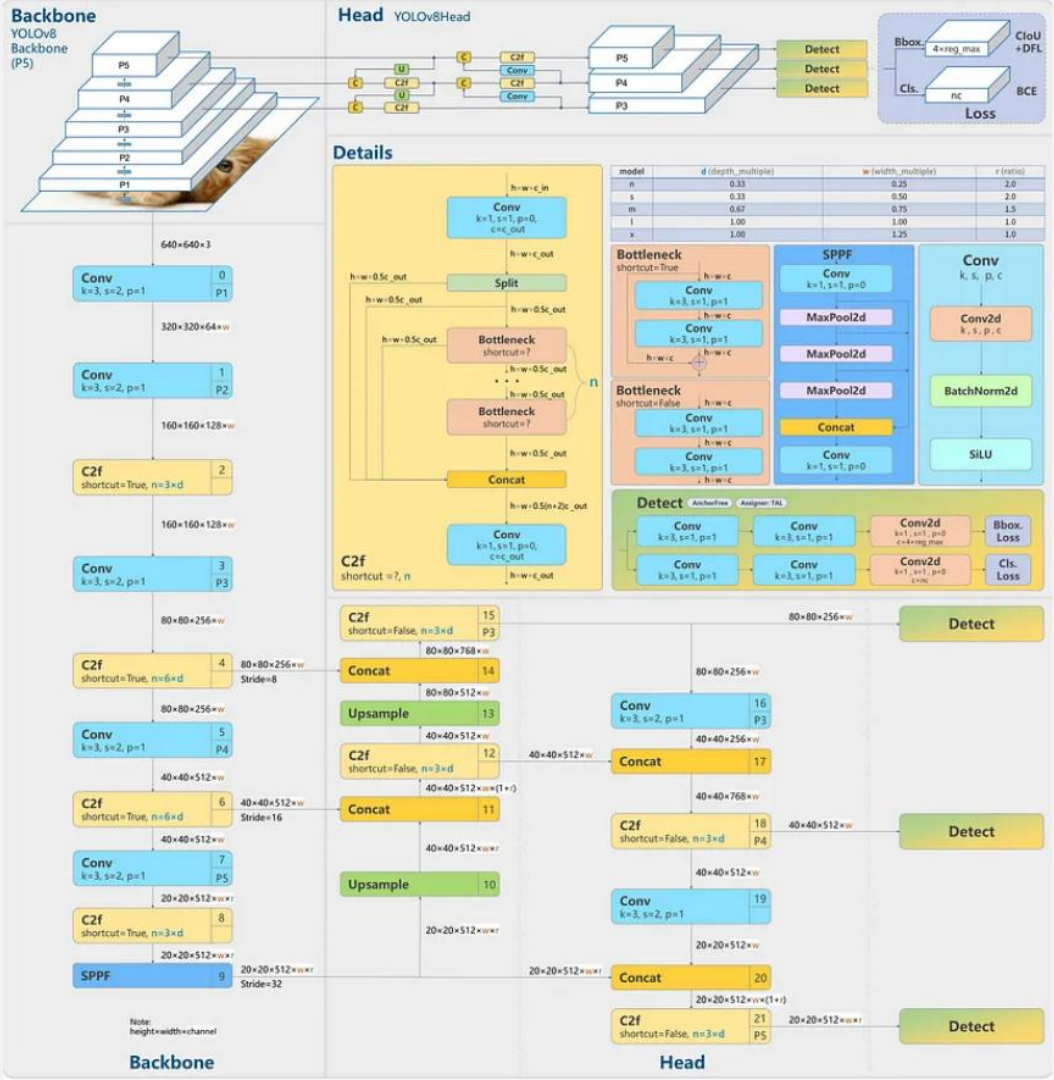
Introduction

- Advancements in computer vision and VR have heightened interest in automating 3D character animation.
- Traditional approaches (manual keyframing, motion capture setups) are time-consuming and costly.
- This research introduces a deep learning-based method for real-time posture identification and replication.
- Objectives: Fine-tune YOLOv8, compare with MediaPipe, integrate into Unity 3D for seamless animation.

Related Work

- Markerless Motion Capture: Extracting human motion without markers/sensors.
- Pose Estimation and Tracking: Estimating 2D/3D human poses and tracking motion.
- Motion Retargeting and Synthesis: Transferring motion between characters/generating new sequences.
- Deep Learning for Motion Generation: Learning and generating realistic motion patterns.
- Applications in Animation and VR: Enhancing animation, VR, and gaming experiences.
- Evaluation Metrics and Benchmarks: Assessing pose accuracy, temporal coherence, realism.





System Architecture

Backbone

- CSPDarknet53: 53 convolutional layers
- Enhances information flow with cross-stage partial connections

Neck

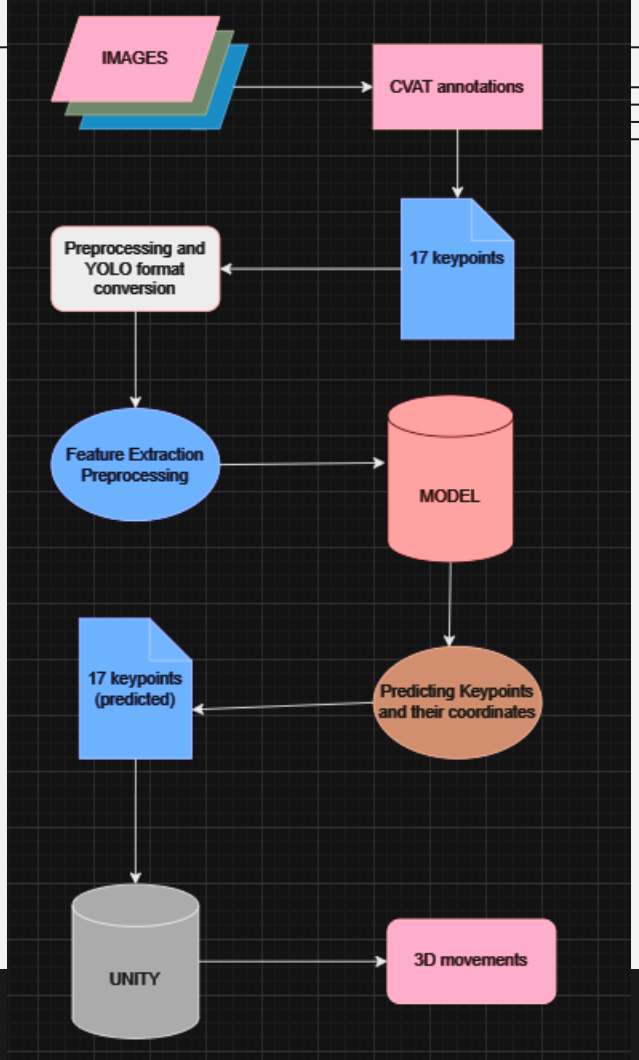
- Merges feature maps
- Uses Spatial Pyramid Pooling (SPP) and Path Aggregation Network (PAN)

Head

- Generates final predictions
- Multiple convolutional and fully-connected layers
- Detects objects at various scales

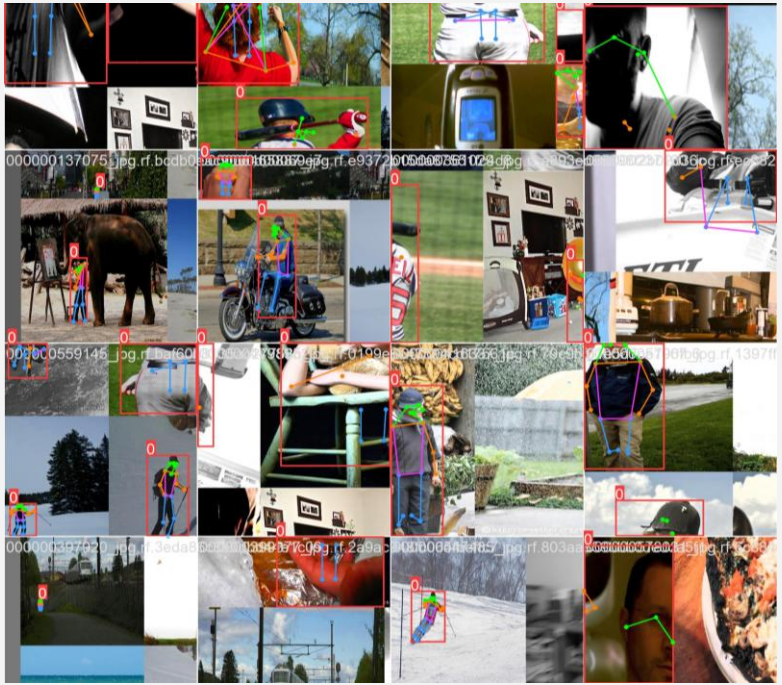
Process flow

- Data Collection
- Annotation
- Preprocessing and Format Conversion
- Feature Extraction and Preprocessing
- Model Training
- Prediction
- Integration with Unity



Data Collection

- Custom dataset combining COCO 2017 and OChuman datasets.
- Approximately 3000 images with single subjects in various poses.



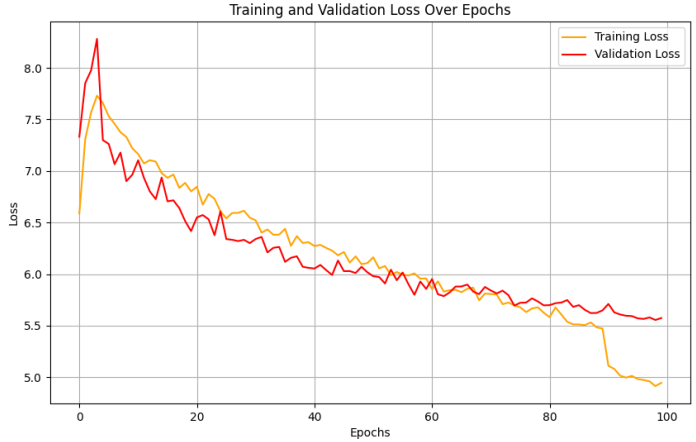
Data Annotation and Preprocessing

- Manual annotation using Computer Vision Annotation Tool.
- Preprocessing with RoboFlow: orientation correction, resizing to 640x640, contrast adjustment.
- Augmentation: 10% of training dataset augmented with grayscale and flipped images.
- Dataset split: 70% training, 15% validation, 15% testing.

Model Training

D. Model Training

The Yolo V8 accepts the data only in the form of YOLO txt format. So I converted my COCO format to YOLO format. With the following configurations,, the model was trained. `EPOCH 84/84 Batchsize=16 imgsz=640` Other parameters are `weight decay = 0.005 , momentum=0.937 , learning rate=0.01.` Other model properties are `250 layers , 329540 parameters , 3295470 gradients , 9.3 Gflops , 397 pretrained weights.`The dataset split looked like this :`Train=2368 , Val=500 , Test =500.` The optimiser used in model training is `optimizer = ADAM Weights.`The model took 10.29 hours to train.



Performance Evaluation

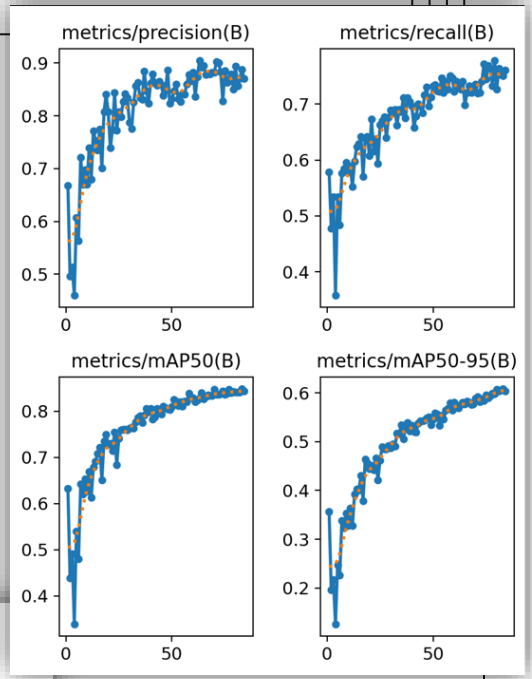
Precision and Recall Metrics:

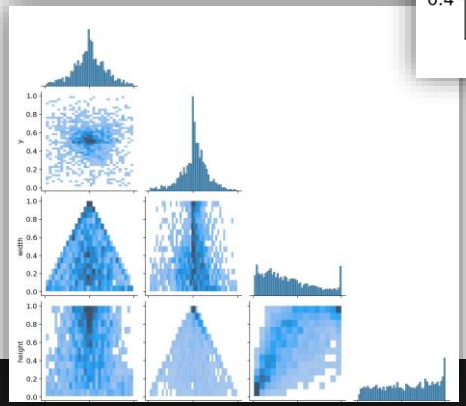
- Top-left: Precision (B) graph
- Top-right: Recall (B) graph

Mean Average Precision (mAP):

- Bottom-left: mAP50 (B) graph
- Bottom-right: mAP50-95 (B) graph

- Label Distribution and Correlation:
- Top histograms: Distribution of x, y, width, height
- Scatter plots: Correlation among x, y, width, height





Operating System & Software Versions

- AMD Ryzen 3600 processor, Nvidia GTX 1660 Super GPU.
- Maximum thermal envelope (TDP): 65W.
- Library and Software Versions:
  Python 3.12 and 3.8 (for cv2).
  Libraries: Ultralytics YOLOv8, CvZone, PyTorch, TensorFlow.
- Development Tools: PyCharm (Python), Unity 3D 2024, Microsoft Visual Studio Community (C#).

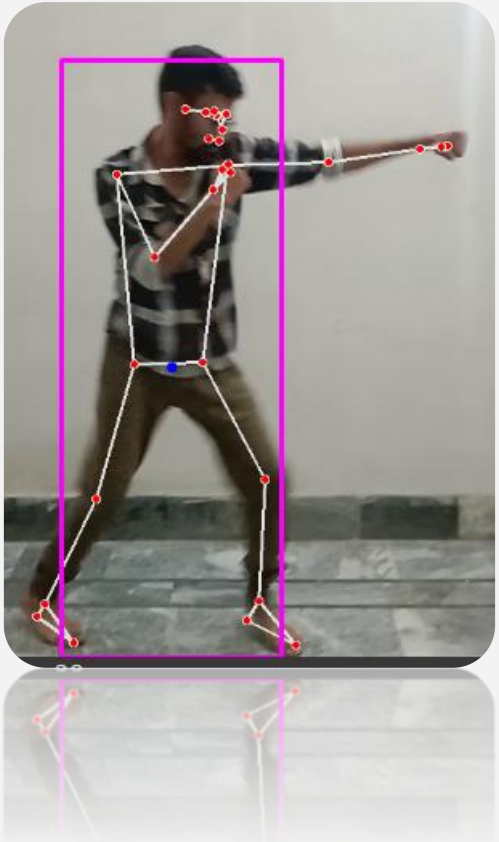
Model Comparison

YOLO V8 Vs Mediapipe

Metric	YOLOv8	Mediapipe
Accuracy	60 to 70%	75 to 80%
Speed	15-45 FPS	30-60 FPS
Size	50-200 MB	5-15 MB

Conclusion

- This research presents a novel approach for virtual character animation using YOLOv8 and MediaPipe.
- Fine-tuned YOLOv8 model demonstrates promising results in real-time pose detection.
- Integration with Unity 3D enables automatic motion replication for interactive applications.
- Applications in VR, gaming, and motion capture.
- Contribution to advancing computer vision and virtual character animation.



Future Work /Application

- Enhance model accuracy and performance.
- Expand dataset for diverse poses.
- Applications in interactive storytelling, immersive content creation, VR experiences, and gaming.

References:

- [1] Davies, T., Taylor, J., & Morten, T. (2017). "Markerless motion capture using multiple color cameras." IEEE Transactions on Circuits and Systems for Video Technology, 27(4), 783-797.
- [2] Cao, Z., Simon, T., Wei, S. E., & Sheikh, Y. (2017). "Realtime multi-person 2D pose estimation using part affinity fields." CVPR.
- [3] Lee, Y., Liu, M. Y., & Grauman, K. (2019). "MoCoGAN: Decomposing motion and content for video generation." CVPR.
- [4] Holden, D., Saito, J., & Komura, T. (2017). "Deep learning of human character motion." ACM Transactions on Graphics (TOG), 36(4), 1-16.
- [5] Magnenat-Thalmann, N., & Thalmann, D. (2004). "The Making of Virtual Humans." Springer Science & Business Media.
- [6] Loper, M., Mahmood, N., & Black, M. J. (2014). "MoSh: Motion and shape capture from sparse markers." ACM Transactions on Graphics (TOG), 33(6), 1-14.