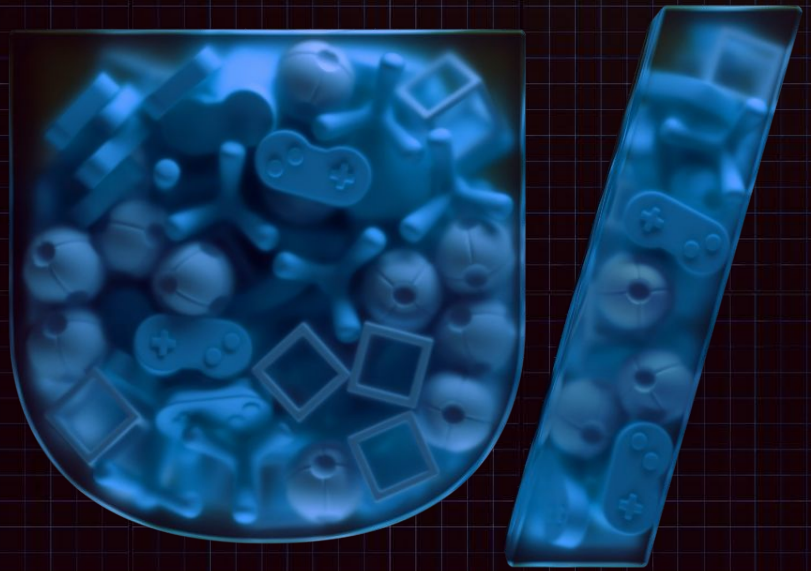
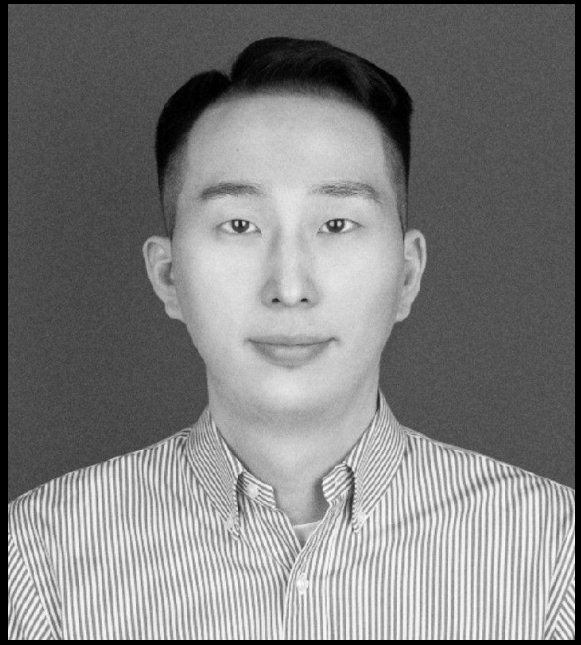


Real-time AI Motion Tracking for XR and Mobile devices

Unity Technologies / Sky Kim

Sky Kim



Senior Software Engineer, Unity
Engine Support Team in APAC
AI and Simulation Technical Support

→ Fun fact: Avid Hackathon participant
→ Contact: sky.kim@unity3d.com

Agenda

- → Advancements in Motion Tracking Technology
- → Motion Tracking in Mobile Environments
- → Motion Tracking in XR Environments
- → Creating a Gesture Recognition Model with Your Own Data

Advancements in Motion Tracking Technology

History of Motion Tracking

Before 2010: Marker-based optical motion capture, IMU sensor-based motion capture
→ Nintendo Wii, Xsens IMU

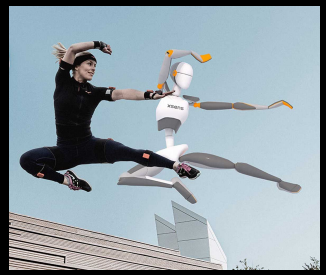
2010–2015: Markerless tracking using RGB-D sensors
→ Microsoft Kinect, Leap Motion

2015–2020: Deep learning-based real-time pose estimation
→ ARKit(Apple), ARCore(Google), OpenPose(CMU)

2020–2025: AR/VR era, AI-sensor fusion
→ Meta Quest, Apple Vision Pro, Mocopi(Sony), Move.ai ...

Post-2025: Popularization of AI glasses and XR-based spatial computing, robotics applications
→ Meta Orion

Nintendo Wii (2006)



Xsens (2008)

Kinect (2011)

OpenPose (2017)

mocopi (2023)

Meta Orion (?)

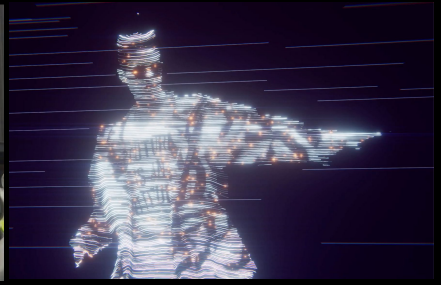
Applications of Motion Tracking

Entertainment (Film, Gaming)
→ AI motion capture used in films like Avatar, The Batman, and media art
→ Character motion generation in games

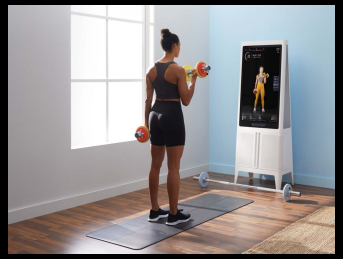
Sports and Medical Rehabilitation
→ Smart fitness services for home training (e.g., Tempo)
→ Rehabilitation exercise games using Kinect
→ AI-based gait analysis, posture correction, and monitoring

Metaverse and Digital Humans
→ Real-time avatar representation in VR social platforms (e.g., VRChat), virtual YouTubers (Vtubers)
→ Real-time facial expression and gesture mirroring through avatars

Avatar: The Way of Water



Media Art (Keijiro)



Tempo

Neofect: Smart Glove

VR Chat

VTuber

BREAKOUT SESSION

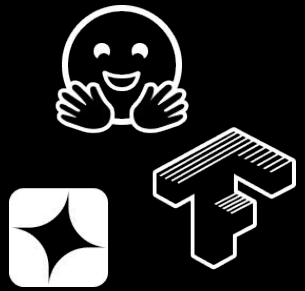
Motion Tracking in Mobile Environments

UNITE/

Environments

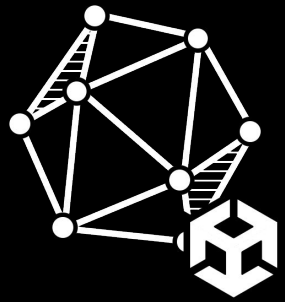
- → Unity 6000.0.40f1 (URP)
- → Sentis 2.1.2
- → Devices
  - — Samsung Galaxy S24+
  - — Meta Quest 3

Running AI Models with Sentis



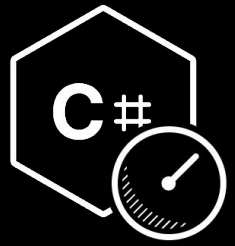
Convert selected AI model to ONNX format

→



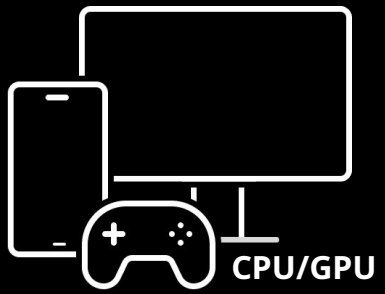
Import into Unity and optimize the model

→



Write inference code

→



Deploy to runtime platforms

(Desktop, Console, Mobile, WebGL, WebGPU)

MediaPipe

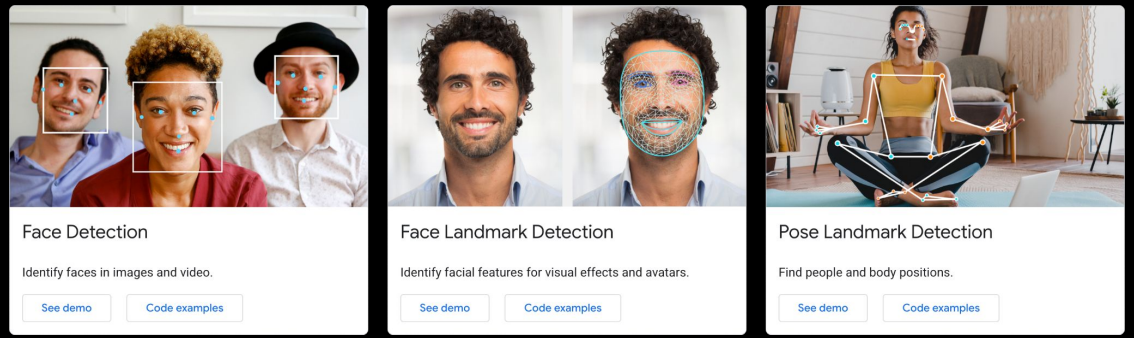
Google MediaPipe provides pre-trained models for various tasks in Vision, NLP, and Audio domains (Apache License)

→ **Vision examples**: Object Detection, Image Classification, Hand Landmark, Hand Gesture Recognition, Image Segmentation, Interactive Segmentation, Face Detection, Face Landmark Detection, Pose Landmark Detection, Image Embedding

→ **Text examples**: Text Classification, Text Embedding, Language Detection

→ **Audio examples**: Audio Classification

— https://ai.google.dev/edge/mediapipe/solutions/examples



Importing MediaPipe Models into Sentis

→ Select a suitable task from the MediaPipe models and download the .task file (click Latest)

- https://ai.google.dev/edge/mediapipe/solutions/vision/face_landmarker

Model bundle	Input shape	Data type	Model Cards	Versions
FaceLandmarker	FaceDetector: 192 x 192 FaceMesh-V2: 256 x 256 Blendshape: 1 x 146 x 2	float 16	FaceDetector FaceMesh-V2 Blendshape	Latest

→ Unzip the task file to extract TFLite models

- unzip face_landmarker.task
- face_detector.tflite, face_landmarks_detector.tflite, face_blendshapes.tflite 생성

→ Convert to ONNX format

- pip install tflite
- python -m tf2onnx.convert --opset 15 --tflite model.tflite --output model.onnx

→ Be sure to review the Model Card to understand the model specifications and limitations


Face Tracking
BlazeFace (Short Range) + FashMesh-V2 + Blendshape-V2

Face Tracking Overview

→ Face landmark detection guide

— https://ai.google.dev/edge/mediapipe/solutions/vision/face_landmarker

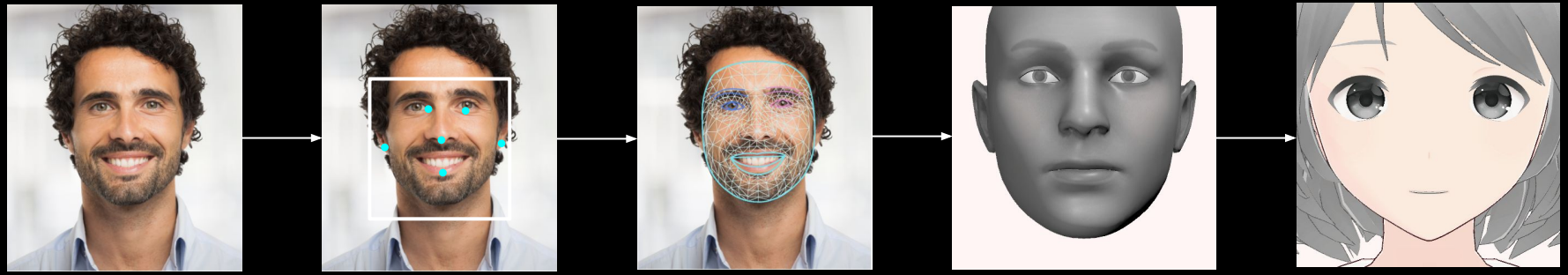


Image (RGB)

Face Region Detection (BlazeFace)

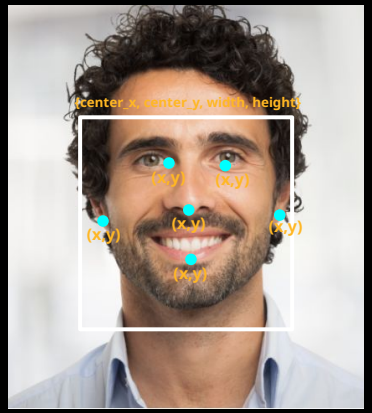
Facial Landmark Detection (FaceMesh)

Blendshape Extraction (Blendshape)

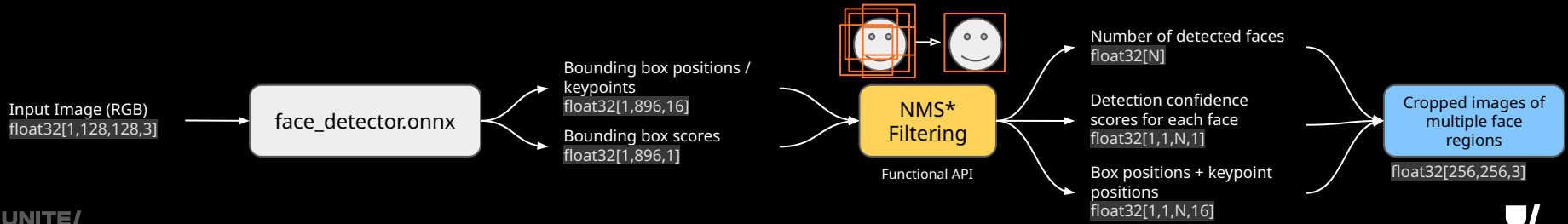
Character

MediaPipe BlazeFace (Short Range)

→ https://storage.googleapis.com/mediapipe-assets/MediaPipe BlazeFace Model Card (Short Range).pdf



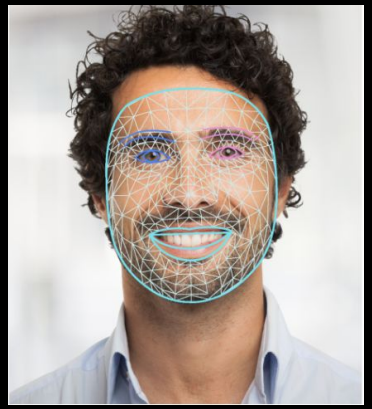
	Description
Details	Extract multiple face bounding boxes and keypoints from images optimized for smartphone front-facing cameras
Architecture	SSD (Single Shot Multibox Detector) like with a custom encoder (CNN)
Model File	face_detector.onnx (418KB)



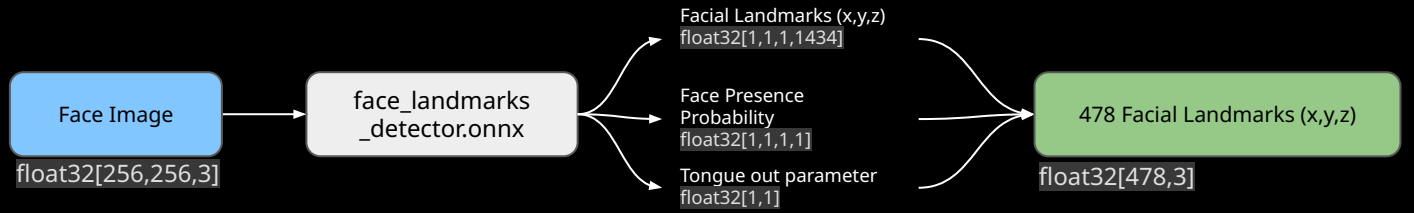
*NMS (Non-Maximum Suppression): A technique used to prevent duplicate detections by keeping only the most confident prediction among overlapping bounding boxes.

MediaPipe FaceMesh

→ https://storage.googleapis.com/mediapipe-assets/Model Card MediaPipe Face Mesh V2.pdf

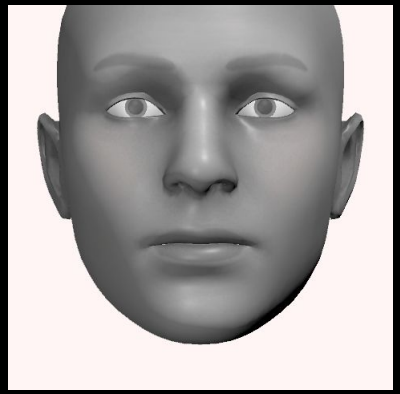


	Description
Details	Real-time prediction of 478 3D facial landmarks from a single frame captured by a smartphone front-facing camera
Architecture	MobileNetV2-like with customized blocks for real-time (CNN)
Model File	face_landmarks_detector.onnx (4.9MB)

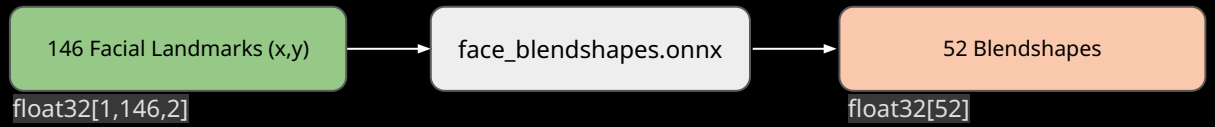


MediaPipe Blendshape V2

→ https://storage.googleapis.com/mediapipe-assets/Model Card Blendshape V2.pdf



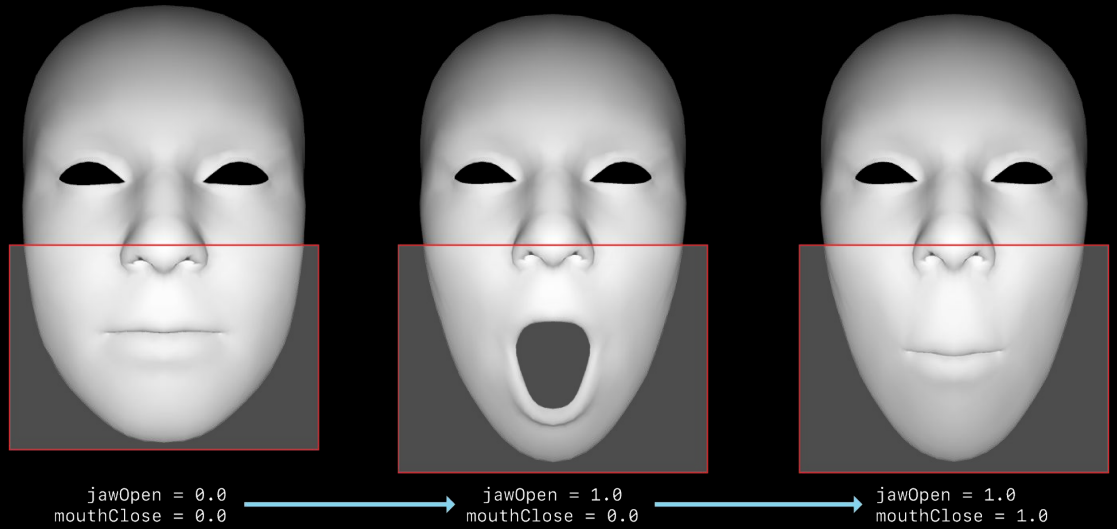
	Description
Details	Real-time prediction of 52 ARKit blendshapes from 146 facial landmarks using FaceMesh
Architecture	MLP-Mixer (CNN)
Model File	face_blendshapes.onnx (1.9MB)



ARKit 52 Blendshapes

→ Apple ARKit 52 blendshapes

- Brow, Cheek, Eyes, Jaw, Mouth, Nose, …
- https://developer.apple.com/documentation/arkit



browDownLeft
browDownRight
browInnerUp
browOuterUpLeft
browOuterUpRight
cheekPuff
cheekSquintLeft
cheekSquintRight
eyeBlinkLeft
eyeBlinkRight
eyeLookDownLeft
eyeLookDownRight
eyeLookInLeft
eyeLookInRight
eyeLookOutLeft
eyeLookOutRight
…


Pose Tracking
BlazePose Pose Detector + BlazePose GHUM 3D + Retargeting

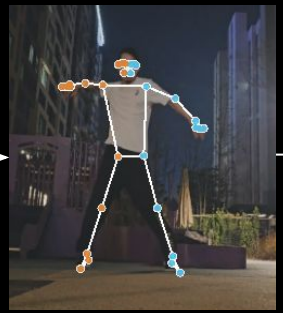
Pose Tracking Overview

→ Pose landmark detection guide

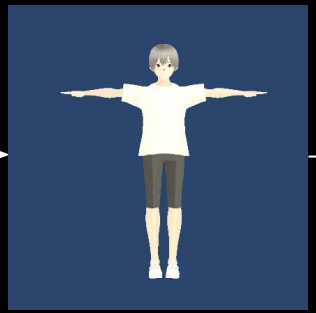
— https://ai.google.dev/edge/mediapipe/solutions/vision/pose_landmarker

Image (RGB)

Body Region Detection (Pose Detector)



Body Landmark Extraction (GHUM 3D)



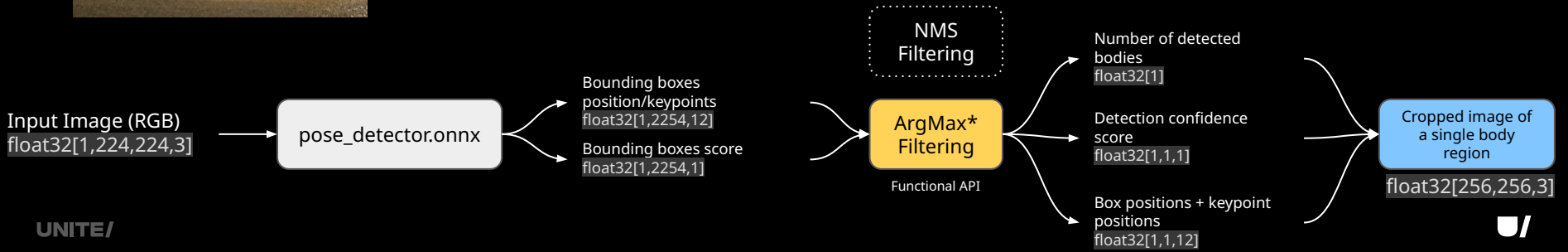
Retargeting

Character

MediaPipe BlazePose Pose Detector

https://storage.googleapis.com/mediapipe-assets/Model Card BlazePose GHUM 3D.pdf

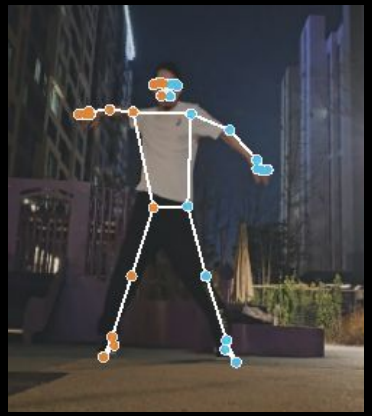
	Description
Details	Extract body region bounding boxes and confidence scores from mobile or webcam video images
Architecture	SSD (Single Shot Multibox Detector) like with a custom encoder (CNN)
Model File	pose_detector.onnx (15.1MB)



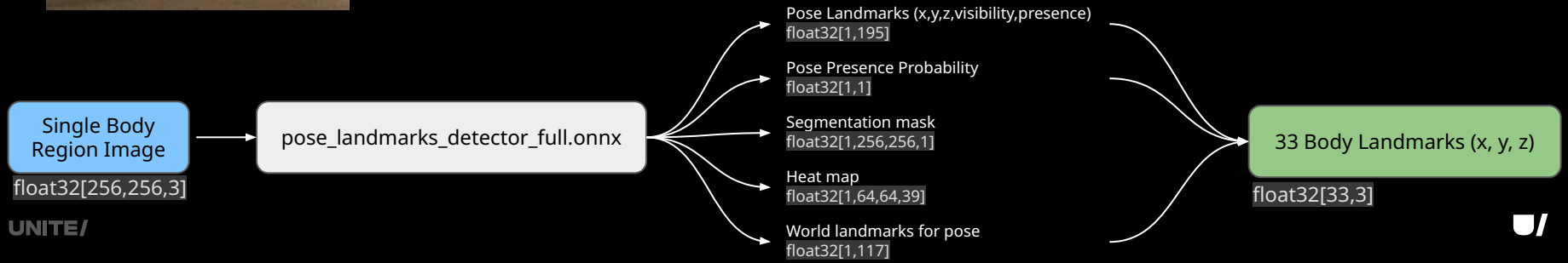
*ArgMax: An operation that returns the index of the element with the highest value within a given array or function

MediaPipe BlazePose GHUM 3D

https://storage.googleapis.com/mediapipe-assets/Model Card BlazePose GHUM 3D.pdf



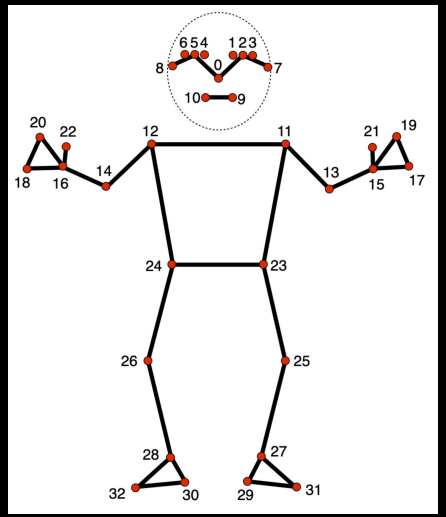
	Description
Details	Real-time extraction of 33 full-body 3D landmarks from a single image
Architecture	MobileNetV2-like with customized blocks for real-time performance (CNN)
Model File	pose_landmarks_detector_full.onnx (12.8MB)



Retargeting and Inverse Kinematics (IK)

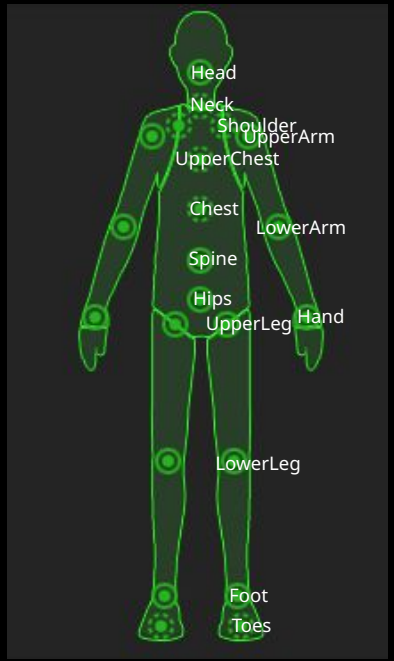
→ Retargeting from BlazePose 33 points 3D landmark to Unity Humanoid Bone

→ Calculate inverse kinematics for every bone



0 - nose
1 - left eye (inner)
2 - left eye
3 - left eye (outer)
4 - right eye (inner)
5 - right eye
6 - right eye (outer)
7 - left ear
8 - right ear
9 - mouth (left)
10 - mouth (right)
11 - left shoulder
12 - right shoulder
13 - left elbow
14 - right elbow
15 - left wrist
16 - right wrist
17 - left pinky
18 - right pinky
19 - left index
20 - right index
21 - left thumb
22 - right thumb
23 - left hip
24 - right hip
25 - left knee
26 - right knee
27 - left ankle
28 - right ankle
29 - left heel
30 - right heel
31 - left foot index
32 - right foot index

Head
LeftEye
RightEye
Neck
Spine
Hips
RightUpperArm
RightLowerArm
RightHand
RightThumbIntermediate
RightMiddleProximal
LeftUpperArm
LeftLowerArm
LeftHand
LeftThumbIntermediate
LeftMiddleProximal
RightUpperLeg
RightLowerLeg
RightFoot
RightToes
LeftUpperLeg
LeftLowerLeg
LeftFoot
LeftToes



Hand Tracking

Hand Detector + Hand landmarker + Gesture Embedder + Gesture Classifier

Hand Tracking Overview

→ Hand landmarks detection guide

— https://ai.google.dev/edge/mediapipe/solutions/vision/hand_landmarker

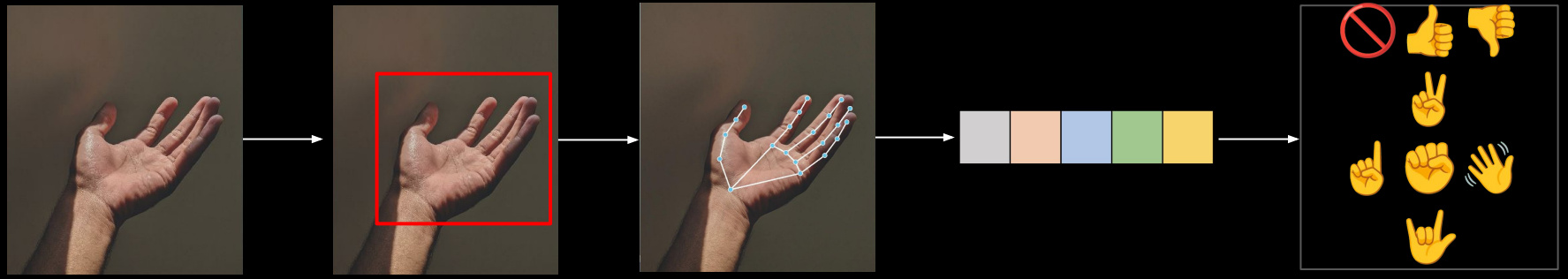


Image (RGB)

Hand Region Detection (Hands Detector)

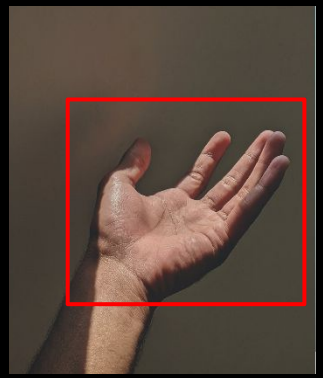
Hand Landmark Extraction (Hand Tracker)

Hand Embedding Tensor Generation (Gesture Embedding)

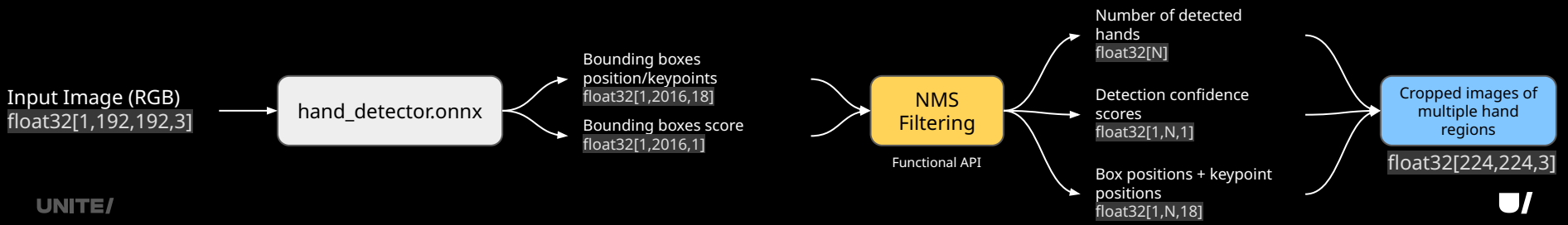
Hand Gesture Recognition (Gesture Classifier)

MediaPipe Hand Detector

https://storage.googleapis.com/mediapipe-assets/Model Card Hand Tracking (Lite_Full) with Fairness Oct 2021.pdf

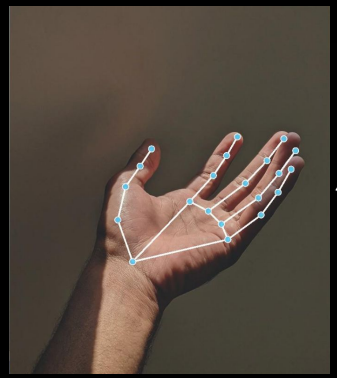


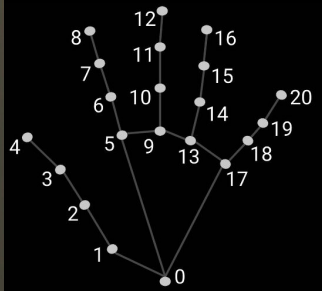
	Description
Details	Extract hand region bounding boxes and confidence scores from mobile or webcam video images
Architecture	SSD (Single Shot Multibox Detector) like with a custom encoder (CNN)
Model File	hand_detector.onnx (4.6MB)



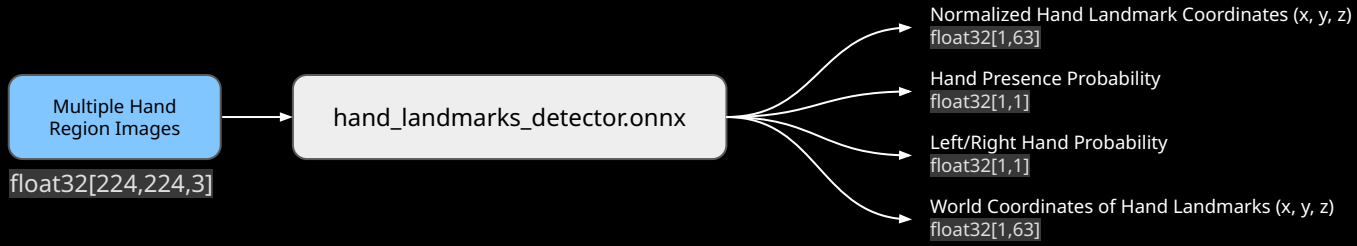
MediaPipe Hand Tracker

https://storage.googleapis.com/mediapipe-assets/Model Card Hand Tracking (Lite_Full) with Fairness Oct 2021.pdf





	Description
Details	Real-time extraction of 21 hand 3D landmarks from a single image
Architecture	Regression model (CNN)
Model File	hand_landmarks_detector.onnx (10.9MB)



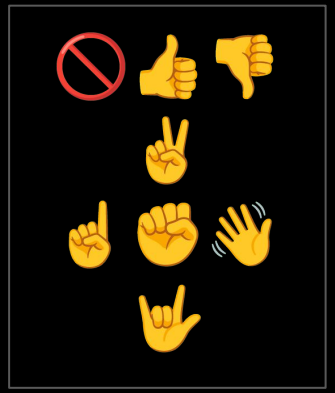
MediaPipe Gesture Embedding

https://storage.googleapis.com/mediapipe-assets/gesture_recognizer/model_card_hand_gesture_classification_with_faireness_2022.pdf

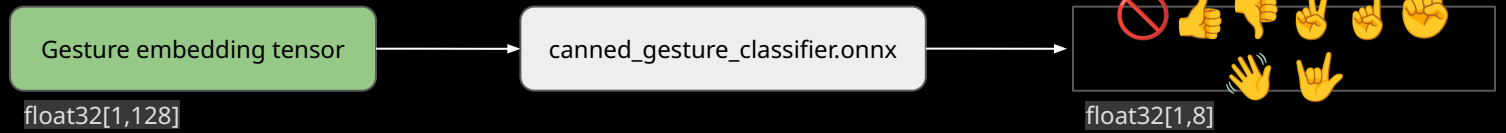
	Description
Details	Generate a 1×128 embedding tensor from 21 three-dimensional hand landmarks
Architecture	Fully Connected Neural Network with residual blocks
Model File	gesture_embedder.onnx (546KB)

MediaPipe Gesture Classification

https://storage.googleapis.com/mediapipe-assets/gesture_recognizer/model_card_hand_gesture_classification_with_faireness_2022.pdf
https://ai.google.dev/edge/mediapipe/solutions/customization/gesture_recognizer



	Description
Details	Calculate the probabilities of 8 gestures using a 128-dimensional embedding tensor
Architecture	Classification model (Fully Connected Neural Network)
Model File	canned_gesture_classifier.onnx (6KB)




Right
Victory
0.8893496


Multi-Person Tracking
Yolo-Pose

Multi-Person Tracking Overview

→ Ultralytics Yolo Pose Estimation

— https://github.com/ultralytics/ultralytics/blob/main/docs/en/tasks/detect.md

BlazePose

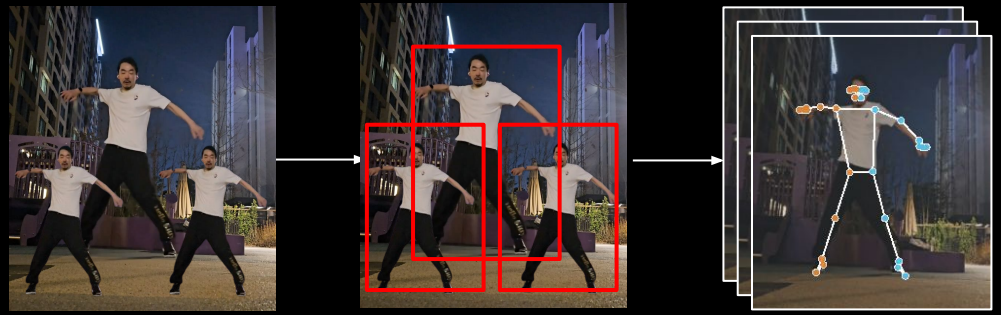


Image (RGB)

Body Region Detection (Pose Detector)

3D Body Landmark Extraction (GHUM 3D)

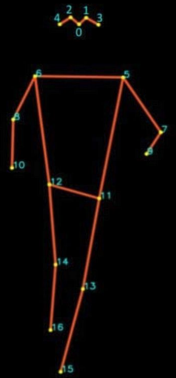
Yolo-Pose

Image (RGB)

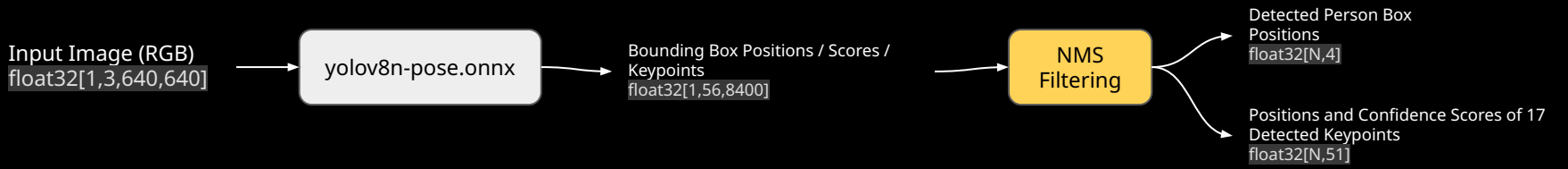
2D Body Landmark Extraction (Yolo-pose)

Yolov8n-Pose

https://huggingface.co/Ultralytics/YOLOv8



	Description
Details	Real-time simultaneous extraction of full-body bounding boxes and 17 2D pose landmarks
Architecture	Backbone, Neck, Head (CNN architecture)
Model File	yolov8n-pose.onnx (13.5MB)



3 people

Top 3 Sentis Optimization Tips for

Set the appropriate Backend Type for your model and environment

→ CPU: Burst
→ GPUCompute: Compute shader
→ GPUPixel: Pixel shader

Use Async/Await to prevent blocking the main thread

→ ReadbackAndClone()
→ await ReadbackAndCloneAsync()

Utilize ScheduleIterable to split and execute layers across multiple frames

→ ScheduleIterable
→ Schedule.MoveNext()

→ https://docs.unity3d.com/Packages/com.unity.sentis@2.1/api/Unity.Sentis.BackendType.html?q=backend
→ https://docs.unity3d.com/Packages/com.unity.sentis@2.1/manual/read-output-async.html
→ https://docs.unity3d.com/Packages/com.unity.sentis@2.1/manual/split-inference-over-multiple-frames.html

Motion Tracking in XR Environments

Meta Quest 3: Body Tracking

→ Face Tracking: Camera-based (Quest Pro only), Voice-based (supported on Quest 3/3S/Pro)

→ Body Tracking: 3-Point vs. IOBT (Inside-Out Body Tracking) with Generative Legs (supported on Quest 3/3S/Pro)

→ Eye Tracking: Camera-based (Quest Pro only)

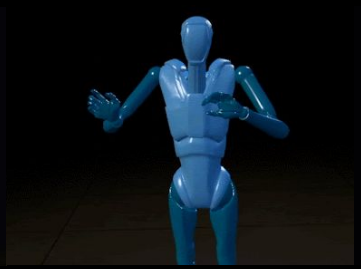
→ Hand Tracking: XR Hands (Unity Package)

- **Movement SDK**: https://github.com/oculus-samples/Unity-Movement
- **XR Hands**: https://docs.unity3d.com/Packages/com.unity.xr.hands@1.5/manual/index.html

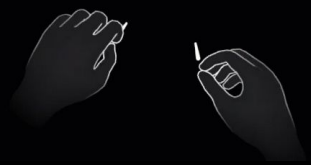
Face Tracking

3-Point vs. IOBT

Body Tracking (Upper body)



Body Tracking (Full body)



Hand Tracking

(Demo) Full body Tracking Demo

→ Meta Quest 3 + Movement SDK

→ Import Mixamo Character (https://www.mixamo.com)

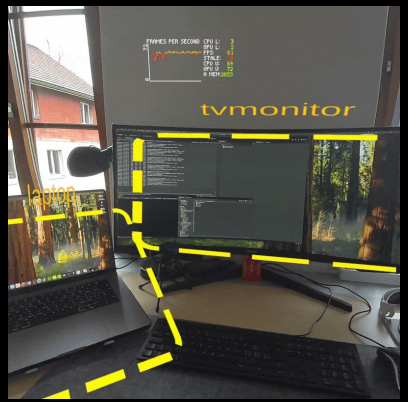
→ GameObject > Movement Samples > Body Tracking > Animation Rigging Retargeting (fullbody)

(my nephew's posture)

Meta Quest 3: Passthrough Camera API

→ Allow camera access on Meta Quest

- Meta Quest 3/3s, HorizonOS: v74 or higher
- Frame Rate: 30fps / Image latency: 40-60ms
- Available resolutions per eye: 320x240, 640x480, 800x600, 1280x960
- Image processing with Sentis is supported
- https://github.com/oculus-samples/Unity-PassthroughCameraApiSamples

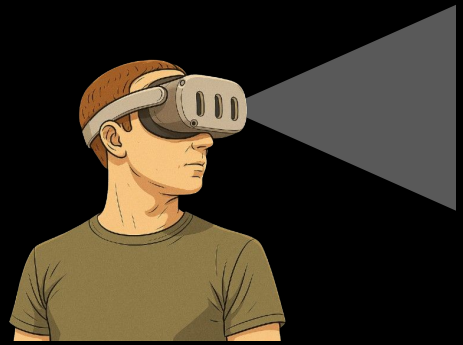


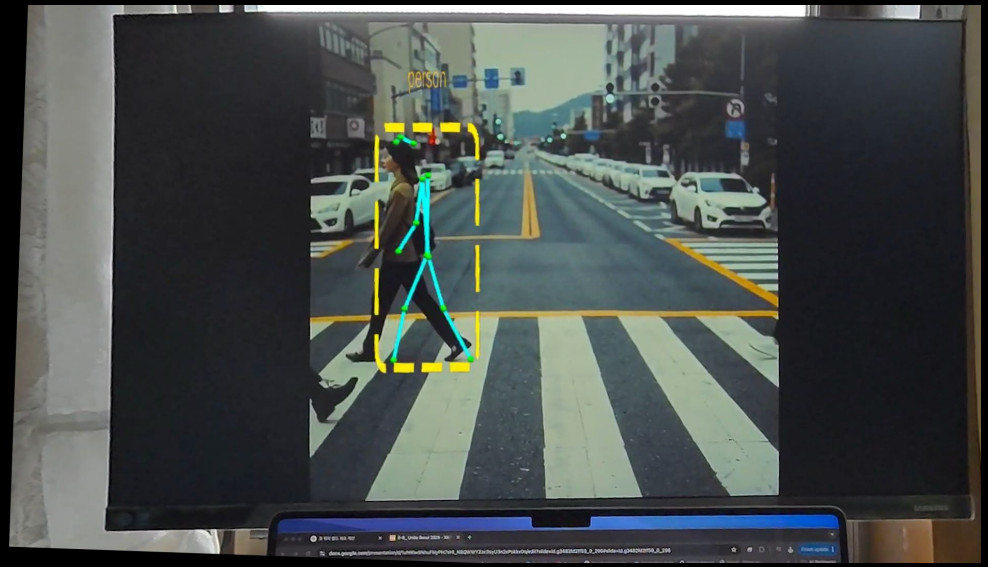
(Demo) Passthrough Human-pose Tracking Demo

→ Meta Quest 3 + Passthrough Camera API

→ Yolov8n-pose: Multi-person tracking model (Sentis)

→ Bounding Box + Pose Tracking



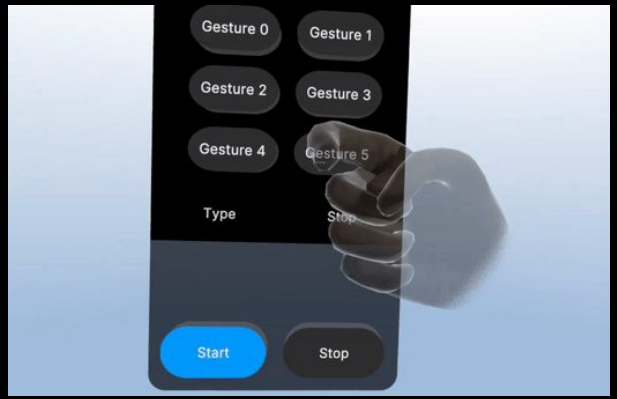


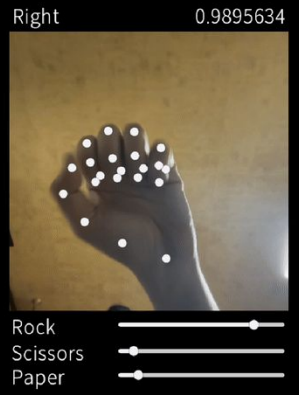
BREAKOUT SESSION

Creating a Gesture Recognition Model with Your Own Data

UNITE/

Gesture Classification Overview





Sign Language Recognition on Meta Quest

Create Custom Sign Language Recognition using XR Hands Keypoints with MLP

https://github.com/skykim/XR-Gesture-Detection

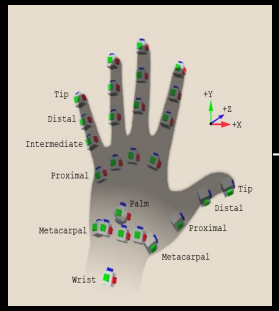
Hand Gesture Recognition on Mobile Devices

Recognize Rock/Paper/Scissors using Blaze Hand Keypoints with MLP

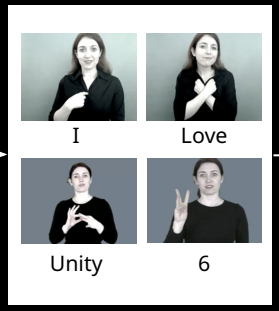
https://github.com/skykim/202407_TechTalk_SentisDemo

Sign Language Recognition on Meta Quest

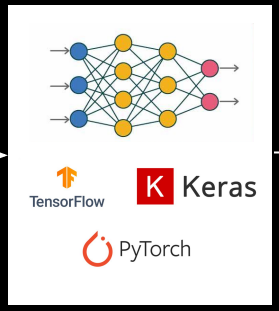
→ Custom Sign Language Recognition Using XR Hands Keypoints (MLP)
→ Tutorial: https://github.com/skykim/XR-Gesture-Detection



Define the Input Data Format
(XR Hands Keypoints)



Generate the Dataset
(Use existing datasets or capture your own)

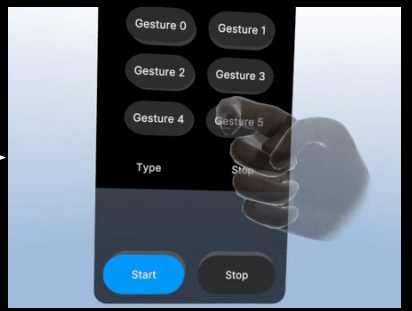


Train the MLP Model
(Tensorflow, Keras, Pytorch)

tf2onnx

torch.onnx

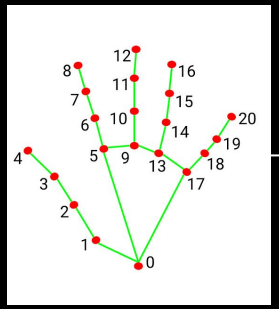
Convert to ONNX



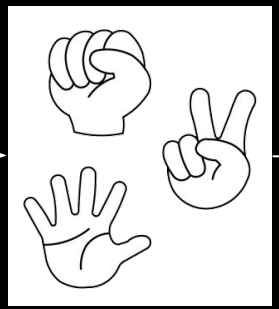
Run the Model with Sentis

Hand Gesture Recognition on Mobile Devices

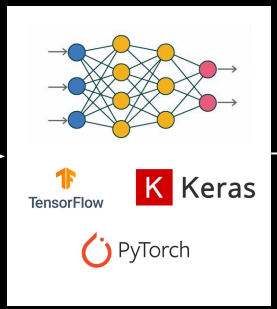
→ Rock/Paper/Scissors Recognition Using Blaze Hand Keypoints (MLP)
→ Tutorial: https://github.com/skykim/202407_TechTalk_SentisDemo



Define the Input Data Format
(BlazeHands Keypoints)



Generate the Dataset
(Use existing datasets or capture your own)

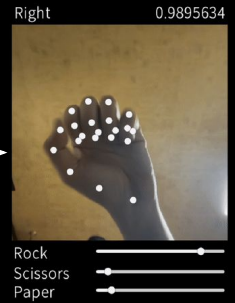


Train the MLP Model
(Tensorflow, Keras, Pytorch)

tf2onnx

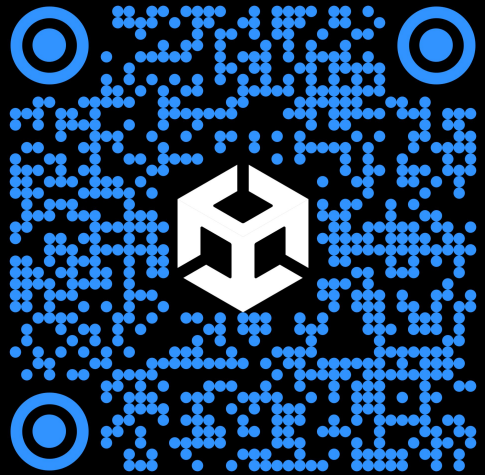
torch.onnx

Convert to ONNX



Run the Model with Sentis

Sample Code



https://github.com/skykim/uniteseoul2025-motiontracking

Thank you