RUTGERS
WINLAB | Wireless Information Network Laboratory

Pose Estimation Overlay

Suhani Sengupta
Advisors: Ivan Seskar and Jennifer Shane

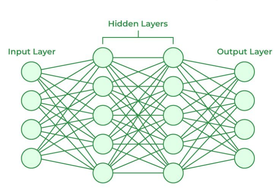
Scan for demo & website!

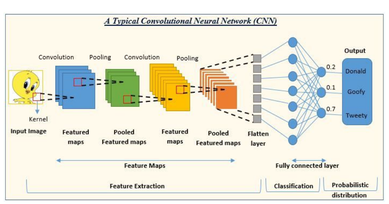
OVERVIEW

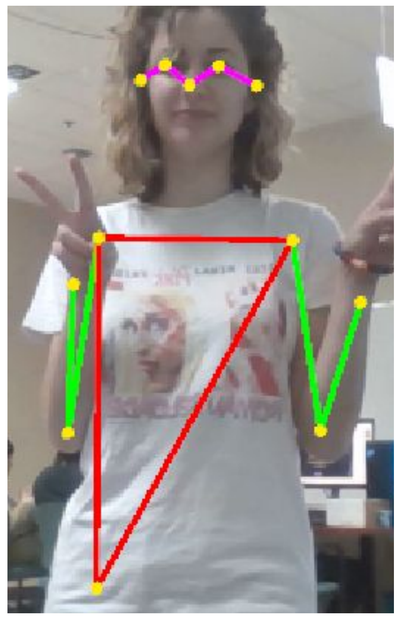
- This progam is an innovative augmented reality application designed to assist doctors by overlaying a detailed skeleton or muscular system onto a patient's body.
- Utilizing advanced pose estimation technology, the system will highlight problematic areas and provides immediate visual feedback.
- It aims to integrate patient charts and medical data, offering a comprehensive view for medical professionals to enhance diagnosis and treatment efficiency.

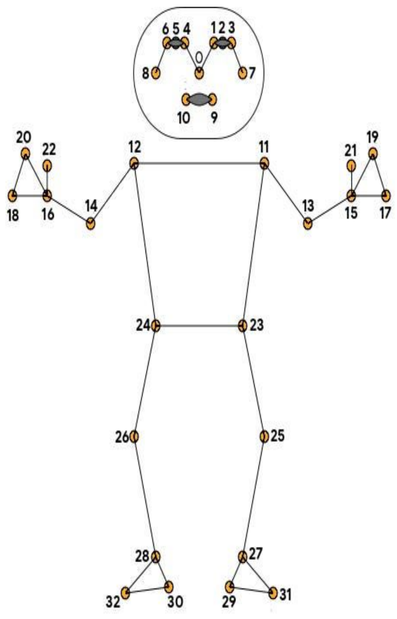
FRAMEWORK

- **Data Collection**
  - A camera captures video footage of the patient, providing a continuous visual stream of the body in motion.
- **Pose Detection**
  - The program processes this video feed using a deep learning model specifically designed for body pose estimation.
  - The model identifies and tracks 33 critical landmarks on the body, including joints and significant body parts. It analyzes spatial relationships and patterns within the video to detect these key points.
- **Landmark Prediction**
  - The program employs convolutional neural networks (CNNs) to predict the precise positions of each landmark. These networks are trained to recognize and follow the landmarks by examining the visual data for distinguishing features and movements.
- **Skeleton Mapping**
  - With the landmarks identified, the program maps these points onto a virtual skeleton or muscular system.
  - This mapping is continuously updated in real-time as the patient moves, ensuring the virtual overlay accurately reflects the patient's current pose.
- **Visualization**
  - The pose data is rendered in Unity, creating a dynamic and interactive overlay of the anatomical structures on the patient's body.
  - This allows for a clear and precise visualization of body movement and posture.

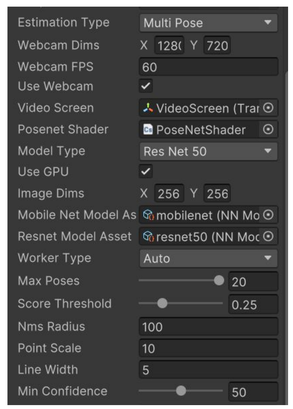








On top is an image of the pose estimation script in action. Below are the options that the user is able to select and change within the program (left) and the logic used behind determining the relative location of the body parts (right)



```
// Get the model output
Tensor heatmaps = engine.PeekOutput(predictionLayer);
Tensor offsets = engine.PeekOutput(offsetsLayer);
Tensor displacementFWD = engine.PeekOutput(displacementFWDLayer);
Tensor displacementBWD = engine.PeekOutput(displacementBWDLayer);

// Calculate the stride used to scale down the inputImage
int stride = (imageDims.y - 1) / (heatmaps.shape.height - 1);
stride -= (stride % 8);

if (estimationType == EstimationType.SinglePose)
{
    poses = new Utils.Keypoint[1][];

    // Determine the key point locations
    poses[0] = Utils.DecodeSinglePose(heatmaps, offsets, stride);
}
else
{
    // Determine the key point locations
    poses = Utils.DecodeMultiplePoses(
        heatmaps, offsets,
        displacementFWD, displacementBWD,
        stride: stride, maxPoseDetections: maxPoses,
        scoreThreshold: scoreThreshold,
        nmsRadius: nmsRadius);
}

heatmaps.Dispose();
offsets.Dispose();
```

FEATURES

- **Multi-Pose vs Single Pose Toggle**
  - Allows users to switch between tracking one pose or multiple poses simultaneously, ideal for both individual and group assessments.
- **Video Input Option**
  - Enables users to input pre-recorded videos for pose estimation, allowing for analysis of previously captured footage.
- **Confidence Level Slider**
  - Adjusts the threshold for pose detection accuracy, ensuring only highly accurate pose estimations are considered.
- **Max Poses Slider**
  - Sets the maximum number of poses to detect and analyze, providing control over performance and tracking multiple individuals simultaneously.

CHALLENGES

Developing the program presented several significant challenges. One major hurdle was the inability to connect the program to the HoloLens, which required rethinking the implementation strategy. Additionally, the overall development process was complex and demanding, involving intricate coding, integration of multiple technologies, and ensuring real-time accuracy and responsiveness. These challenges required persistent problem-solving and adaptation to create a functional and reliable system.

FUTURE WORK

- **HoloLens Integration**
  - The next step is to successfully connect the program to the HoloLens, enabling augmented reality visualization directly on the headset. This will enhance the program's usability and application in clinical settings.
- **Skeleton Attachment**
  - Attaching a detailed skeleton model to the pose estimation framework will provide a more comprehensive visualization of the patient's anatomy. This feature will allow doctors to better understand the alignment and movement of the skeletal system in real-time.
- **System Refinement**
  - Continuous refinement of the program is planned to improve its accuracy, responsiveness, and user interface. Enhancements will focus on optimizing performance, reducing latency, and ensuring a seamless user experience. This includes fine-tuning the pose detection algorithms and improving the integration of patient charts and medical data.

ACKNOWLEDGEMENTS

I would like to thank Dr. Ivan Seskar and Jennifer Shane, for continuously guiding me and helping me throughout this project

I would like to thank my teammates on AR Mural, for always helping me in resolving the numerous errors I would encounter while developing this program

I would also like to thank everyone who volunteered as a test subject for the estimation script for allowing me to test my program on them

WINLAB