



# Pose Estimation

# Jetson Nano

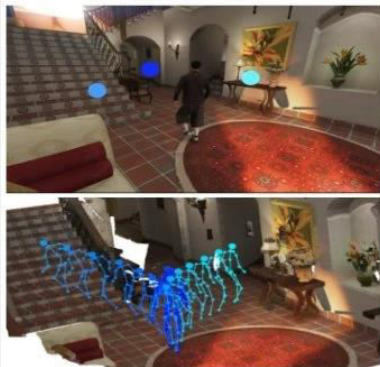
## INTRODUCTION



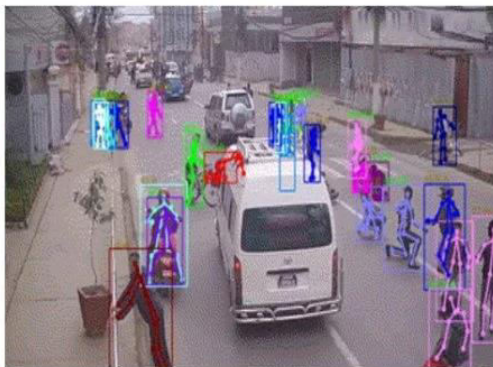
# What is **Pose Estimation**?

= "recovers the pose of an articulated body, which consists of joints and rigid parts using image-based observations."

# Applications



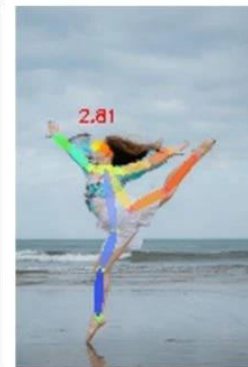
**Action prediction**



**Surveillance**



**Cloth Parsing**



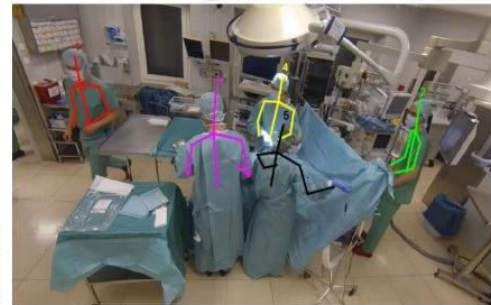
**Online Coaching**



**Movie and Game**



**AR and VR**



**Healthcare**

# Challenges?

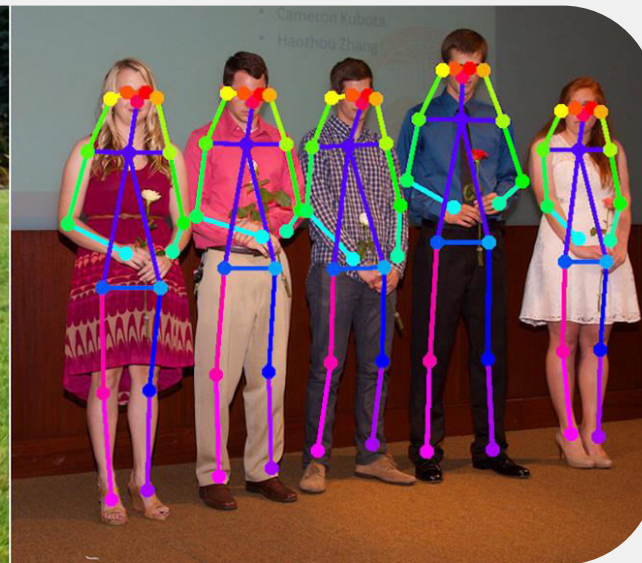


# Methods

- **Holistic View**
- **Local View**
- **Combining Local and Global**
- **Others**

=

# POSENET



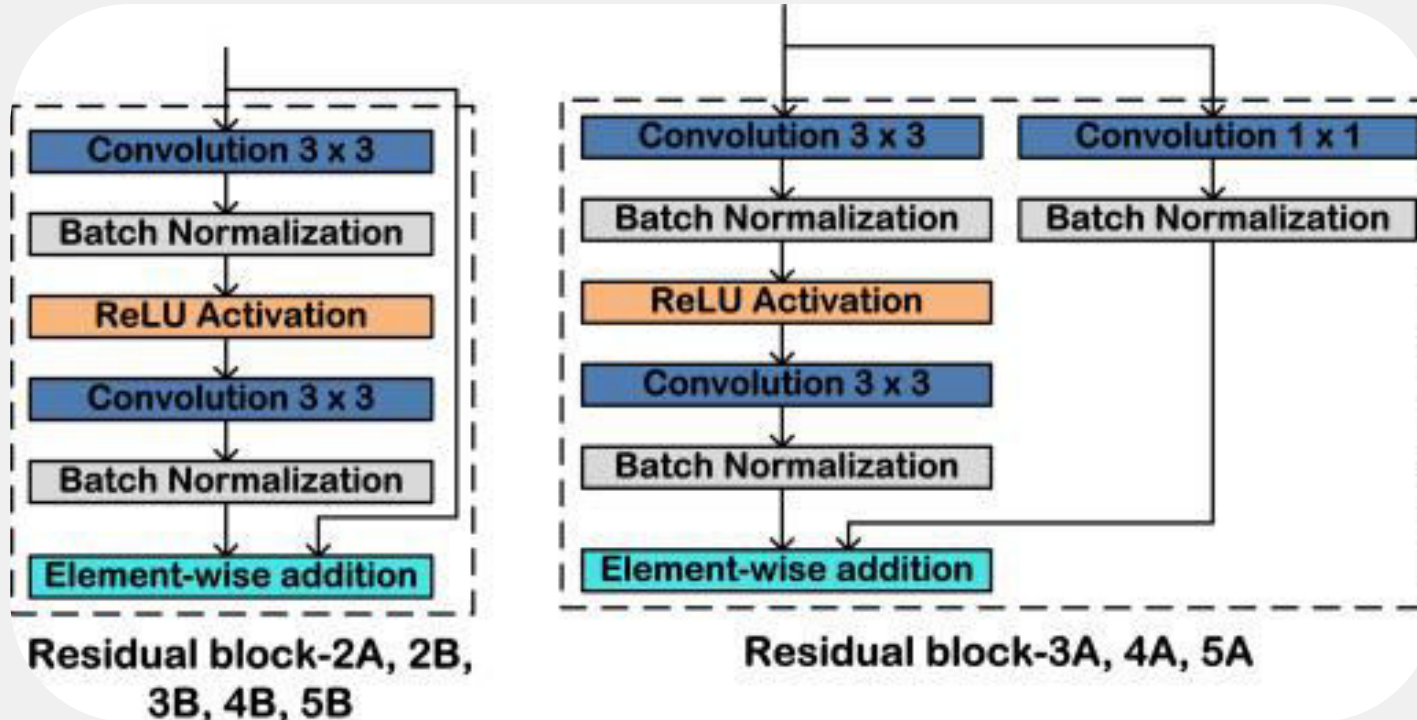
# Models

| Model                 | CLI argument     | NetworkType enum | Keypoints |
|-----------------------|------------------|------------------|-----------|
| Pose-ResNet18-Body    | resnet18-body    | RESNET18_BODY    | 18        |
| Pose-ResNet18-Hand    | resnet18-hand    | RESNET18_HAND    | 21        |
| Pose-DenseNet121-Body | densenet121-body | DENSENET121_BODY | 18        |





# Architecture





# Setting up Environment

=

- **Jetson Nano**
- **Jetson Inference**
- **Good Internet**

# Real Time Inference



AUGMENTED STARTUPS  
Computer Vision | AI | Robotics

- webcam
- Jetson nano
- Nvidia Jetpack

# Implementation

```
from jetson_inference import
```

```
poseNet
```

```
from jetson_utils import
```

```
videoSource, videoOutput, logUsage
```

```
# load the pose estimation model
```

```
net = poseNet(opt.network, sys.argv,
```

```
opt.threshold)
```

# Result

