AI for Sports

- Prakash K Naikade (7000433),
prna00001@stud.uni-saarland.de

Camera Calibration

- To find Intrinsic and Extrinsic parameters of the camera

- Intrinsic Parameters:
  - Camera Matrix and Distortion Coefficients
- Extrinsic Parameters:
  - Rotation Matrix and Translation Vector

- *Distortion coefficients* = $(k_1, k_2, p_1, p_2, k_3)$ *where,* $k_n = n^{th}$ *radial distortion coefficient ,*
  $p_n = n^{th}$ *tangential distortion coefficient*

- *Camera matrix* = $\begin{bmatrix} f_x & 0 & o_x \\ 0 & f_y & o_y \\ 0 & 0 & 1 \end{bmatrix}$ *where,* (f_x, f_y) *= Focal length,* (o_x, o_y) *= Camera Center*

- *Rotation matrix* = $\begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix}$ *where, '**R and t**' together describes how to transform points in world coordinates to camera coordinates,*

  *matrix **R** represent the directions of the world-axes in camera coordinates,*

- *Translation vector* = $\begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}$ *vector **t** can be interpreted as the position of the world origin in camera coordinates,*

Camera Calibration

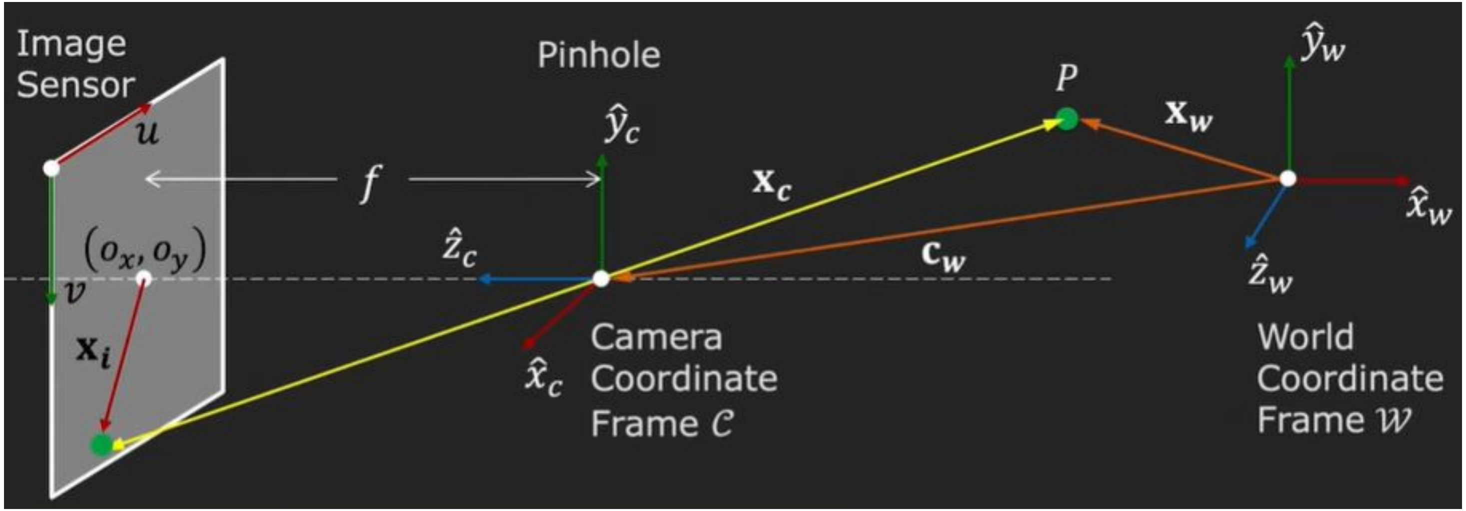


Fig. 1 - World to Camera to Image (Credit: Prof. Shree Nayar, Columbia University)

Position c_w and Orientation *R* of the camera in the world coordinate frame *w* are the camera's Extrinsic Parameters.

$$R = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \begin{array}{l} \Longrightarrow \textit{Direction of } x_c \textit{ in world coordinate frame} \\ \Longrightarrow \textit{Direction of } y_c \textit{ in world coordinate frame} \\ \Longrightarrow \textit{Direction of } z_c \textit{ in world coordinate frame} \end{array}$$

Camera Calibration

Projection Matrix *P*

Camera to Pixel

$$\begin{bmatrix} \tilde{u} \\ \tilde{v} \\ \tilde{w} \end{bmatrix} = \begin{bmatrix} f_x & 0 & o_x & 0 \\ 0 & f_y & o_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix}$$

$$\tilde{u} = M_{int}\, \tilde{x}_c$$

World to Camera

$$\begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}$$

$$\tilde{x}_c = M_{ext} \tilde{x}_w$$

Combining the above two equations, we get the full projection matrix P:

$$\begin{bmatrix} \tilde{u} \\ \tilde{v} \\ \tilde{w} \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}$$

(Credit: Prof. Shree Nayar, Columbia University)

Camera Calibration

Step 1: Capture images of an object with known geometry

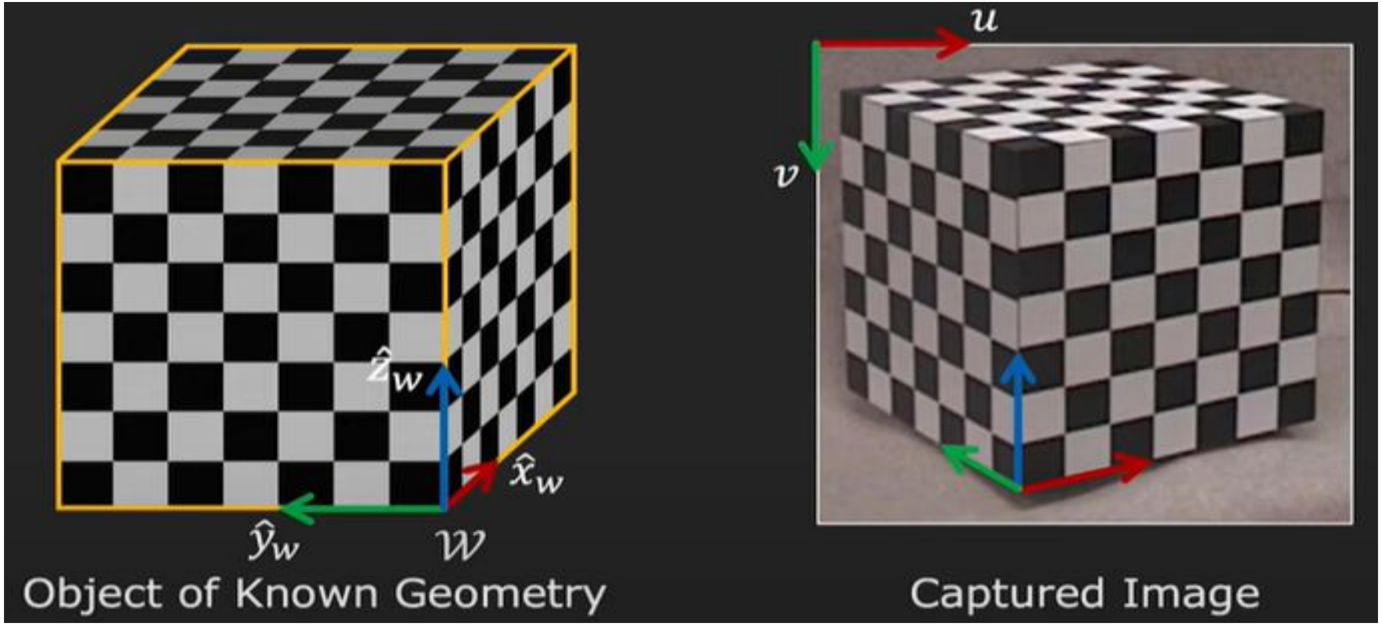


Fig. 2 - Camera Calibration (Credit: Prof. Shree Nayar, Columbia University)

Camera Calibration

Step 2: Identify correspondences between 3D scene points and 2D image points

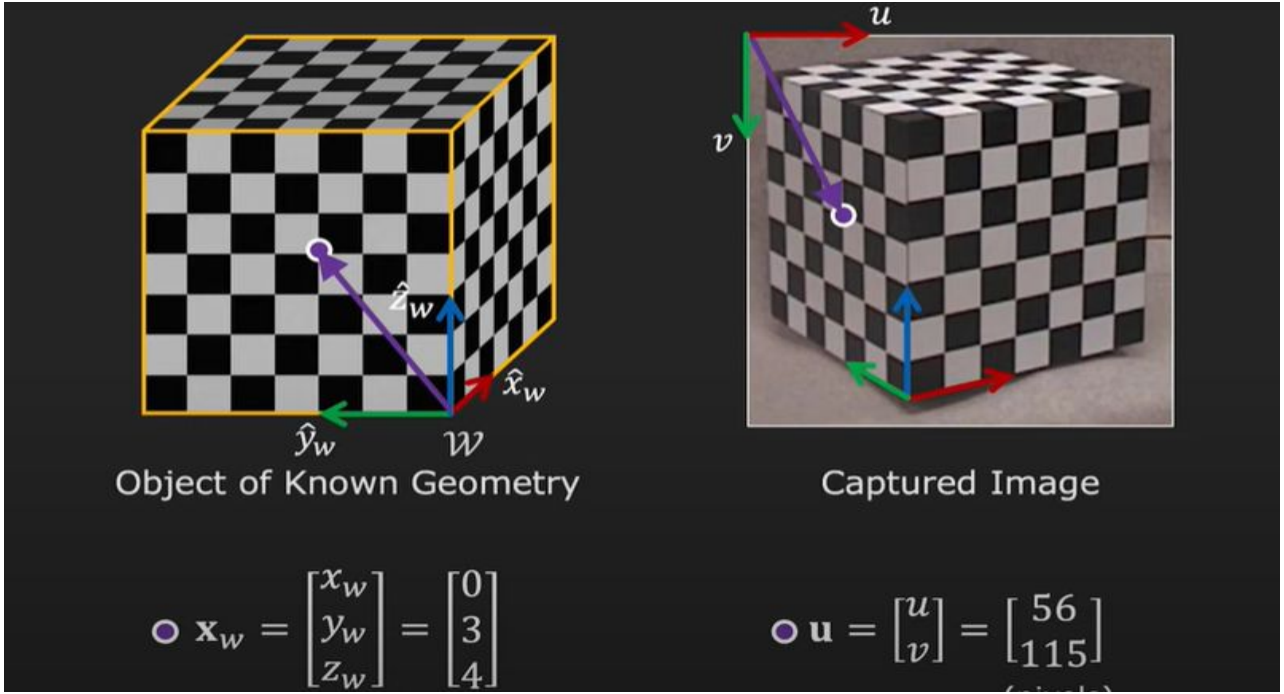


Fig. 3 - Camera Calibration (Credit: Prof. Shree Nayar, Columbia University)

Step 3: For each corresponding point '*i*' in scene and image

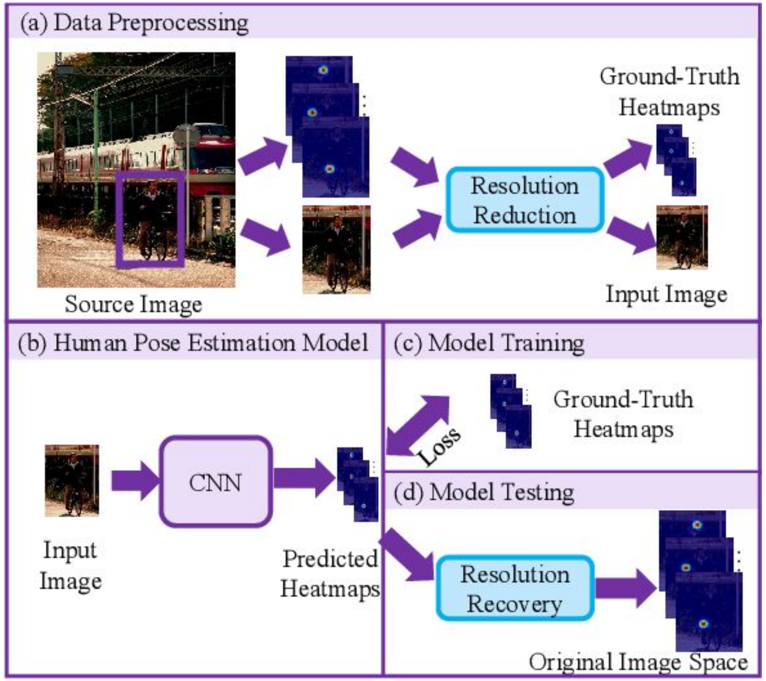
$$\underbrace{\begin{bmatrix} u^{(i)} \\ v^{(i)} \\ 1 \end{bmatrix}}_{\text{known}} = \underbrace{\begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix}}_{\text{unknown}} \underbrace{\begin{bmatrix} x_w^{(i)} \\ y_w^{(i)} \\ z_w^{(i)} \\ 1 \end{bmatrix}}_{\text{known}}$$

$\Longrightarrow$ *solve this problem by constrained least square method by reducing problem to eigenvalue problem*

$\Longrightarrow$ *Direct Linear Transformation (DLT)*

(Credit: Prof. Shree Nayar, Columbia University)

2D-Keypoints Detection

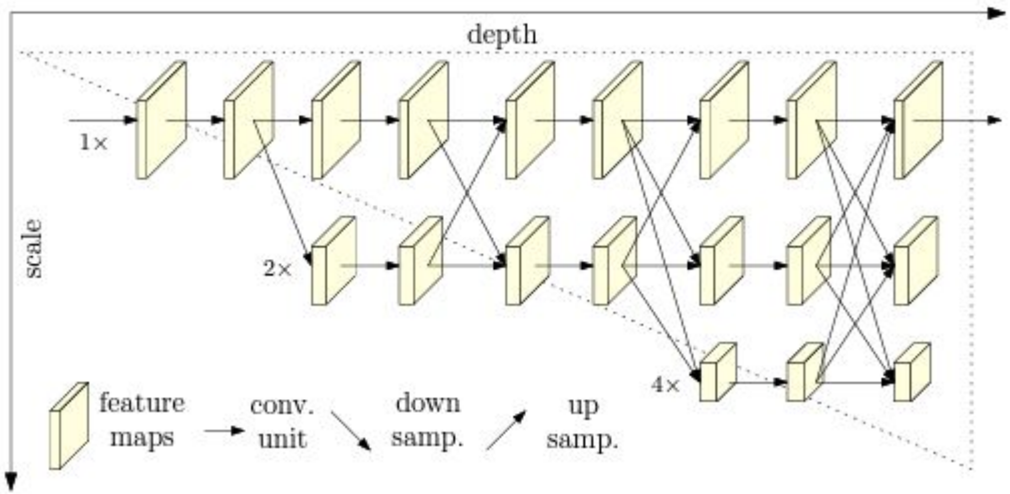


- *Objective is to predict the joint coordinates in a given input image*
- *Requirement of label representation for encoding the body joint coordinate labels*
- *To calculate the supervised learning loss and joint coordinates*
- *Encoding the ground truth joint coordinates into heatmaps*
- *Encoded heatmap will be learning target*
- *Decoding the predicted heatmap into joint coordinates*

Fig. 5 - HPE Pipeline (Credit: CVPR 2020 - Distribution-Aware Coordinate Representation for Human Pose Estimation)

2D-Keypoints Detection

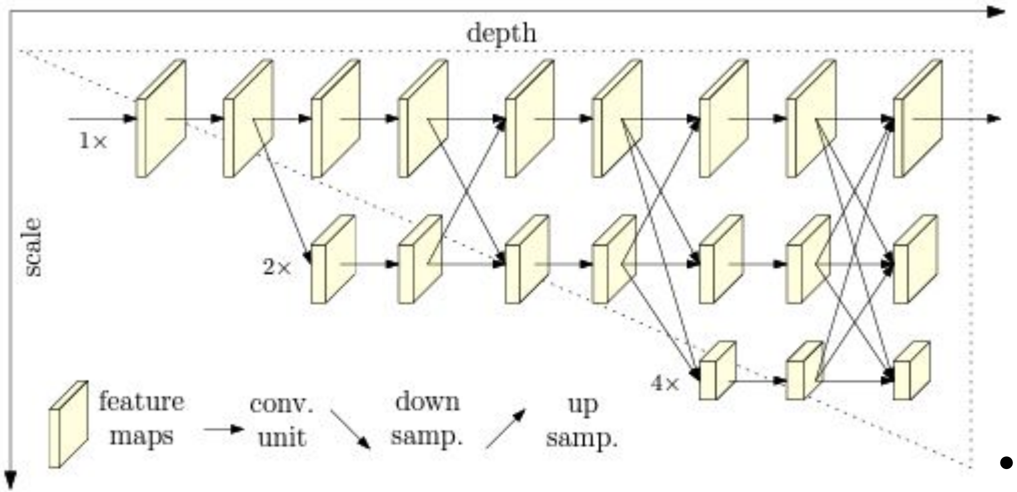
- Using deep-learning
- Transfer- learning
- Fine tuning of pre-trained networks



- *Parallel high-to-low resolution subnetworks with repeated information exchange across multi-resolution subnetworks (multi-scale fusion)*
- *Horizontal direction correspond to the depth of the network*
- *Vertical direction correspond to the scale of the feature maps*

Fig. 6 - Network Architecture for HPE
(Credit: CVPR 2019 - Deep High-Resolution Representation Learning for Human Pose Estimation)

2D-Keypoints Detection



- *Repeated multi-scale fusions such that each of the high-to-low resolution representations receives information from other parallel representations over and over, leading to rich high resolution representations*

- *Predicted keypoint heatmap is potentially more accurate and spatially more precise*

Fig. 6 - Network Architecture For HPE
(Credit: CVPR 2019 - Deep High-Resolution Representation Learning for Human Pose Estimation)

3D-Keypoints Detection

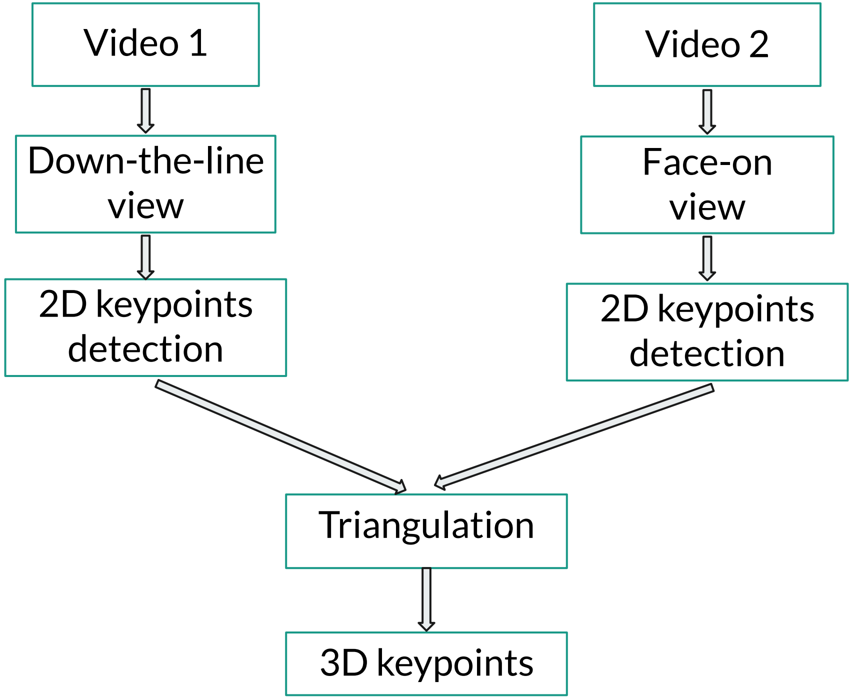


Fig. 7 - Algorithm Pipeline

Triangulation

- Process of determining a point in 3D space given its projections onto two, or more, images
- Needs Projection Matrix

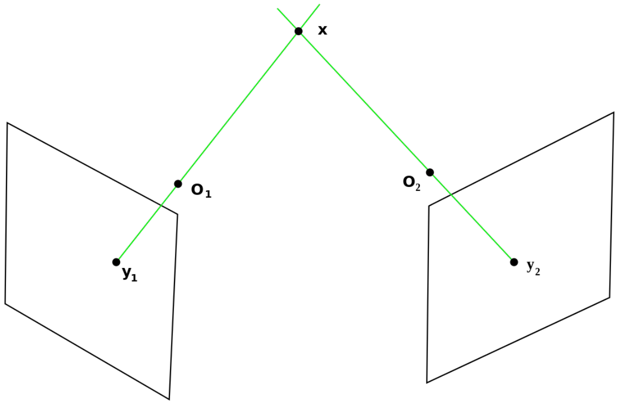


Fig 8a: Ideal Case - Epipolar Geometry

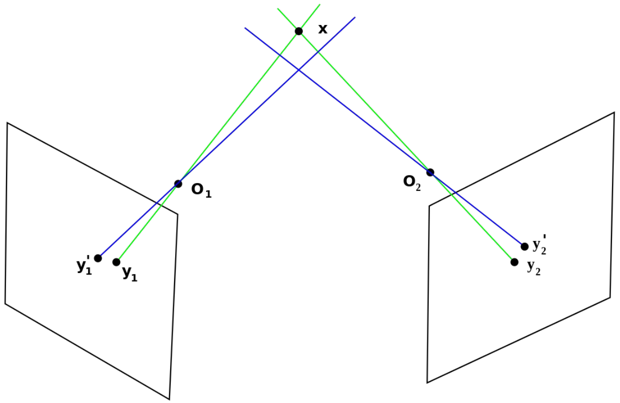


Fig 8b: Actual Case - Epipolar Geometry

Fig. 8 - Triangulation
(Credit: Wikipedia)

Golf Sequence Detection

- Hybrid of deep convolutional and recurrent network
- Maps a sequence of RGB images **I** to a corresponding sequence of event probabilities ***e***
- Sequence of feature vectors **f** generated by *MobileNetV2*
- **f** are input to a bidirectional LSTM
- Softmax is applied to obtain the event probabilities

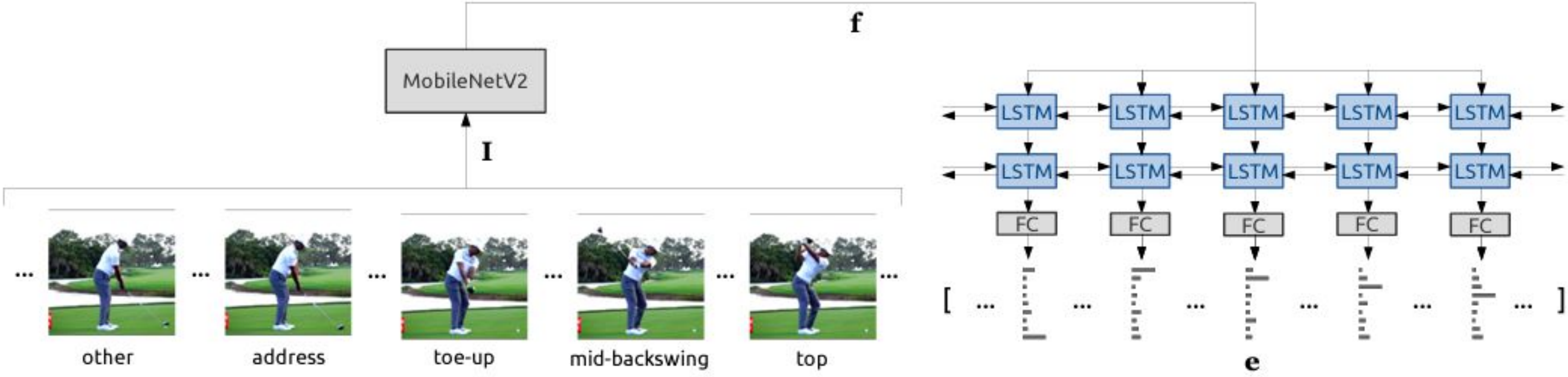
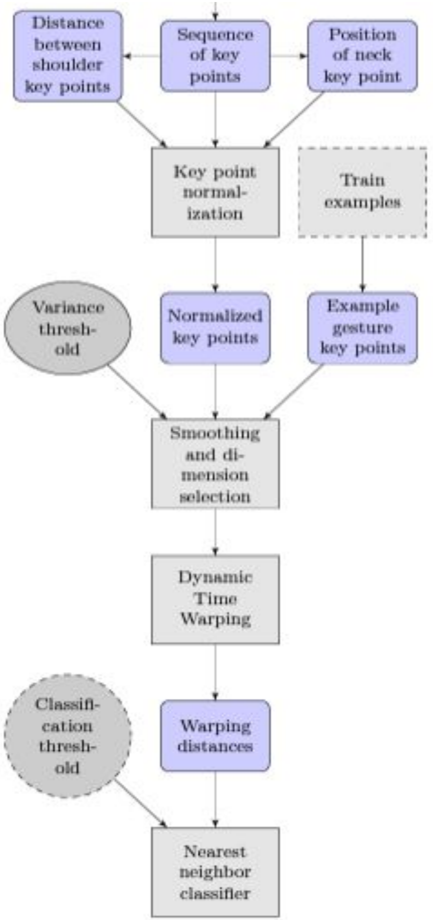


Fig. 13 - Swing Net for Golf Sequence Detection
(Credit: GolfDB: A Video Database for Golf Swing Sequencing)

Two Pose Sequence Comparison



- Input - Two 3D pose Sequences
- Normalize the keypoints by translation and scaling
- Removing the keypoints that don't vary a lot during the pose sequences
- Calculating the similarity score by using dynamic time warping
- Output - Decimal number, the lower the better.

Fig. 15 - Pipeline for Pose Comparison
(Credit: Gesture Recognition in RGB Videos Using Human Body Keypoints and Dynamic Time Warping)

Thank You !

Questions Please !