Gesture Recognition

P. Papageorgiou, N. Papanastasiou, D. Makrygiannis,
Prof. Ioannis Pitas
Aristotle University of Thessaloniki
pitas@csd.auth.gr
www.aiia.csd.auth.gr

Version 5.1

Artificial Intelligence &
Information Analysis Lab

Gesture Recognition

- **Introduction**
- Gesture types
- Gesture Acquisition Devices
- Human-Machine Interaction
- Gesture Recognition Datasets
- Gesture Recognition Algorithms
- Deep Gesture Recognition
- Skeleton-Based Gesture Recognition
- Multimodal Gesture Recognition
- Egocentric Gesture Recognition
- Applications

Introduction

- ***Gesture***: expressive meaningful body motion involving physical movement of head, body, hands etc.
- Intention:
  - Convey meaningful information
  - Interact with environment.
- Gestures can be:
  - ***Static***: certain body posture or configuration.
  - ***Dynamic***: prestrike, stroke and poststroke phases.

Introduction

- Gestures are cultural specific.
- Gestures can be categorized based on the body part as:
  - ***Hand gestures***:
    - hand poses, sign language etc.
  - ***Head and face gestures***:
    - Shaking head
    - Speaking by opening and closing the mouth
    - Raising the eyebrows
    - Emotions: surprise, anger, happiness, sadness
  - ***Body gestures***: full body motion.

Introduction

Gesture taxonomy

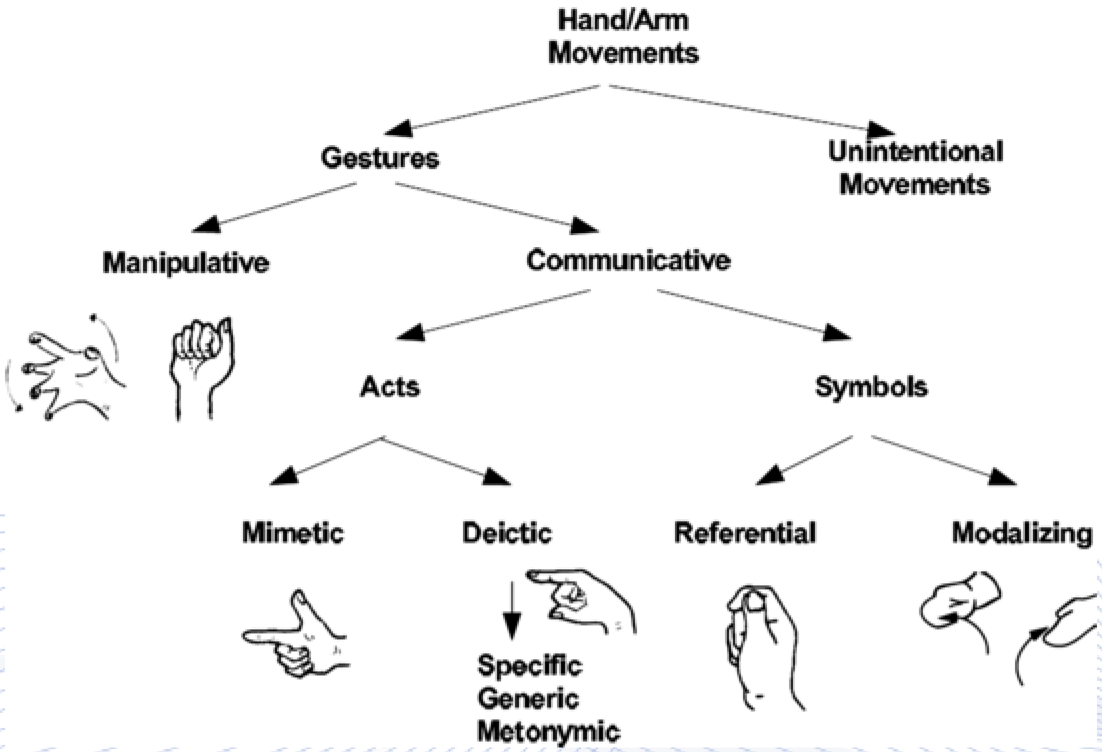
- ***Emblems***: “OK” sign with thumb and index finger connected in a circle with the other three fingers sticking up
  - other culture-specific “rude” gestures.
- ***Gesticulations*** are spontaneous movement of hands and arms, accompanying speech.
- ***Pantomimes*** are gestures depicting objects or actions.

Introduction

Hand and arm movements

  - Unintentional or intentional.
- ***Manipulative Gestures***:
  - They act on objects in an environment (object movement, rotation etc.)
  - Communicative.
- ***Communicative gestures*** can be:
  - Acts
    - Mimetic or Deictic
  - Symbols: gestures with linguistic role
    - Referential or Modelizing.

Introduction



Hand and arm movements.

Introduction

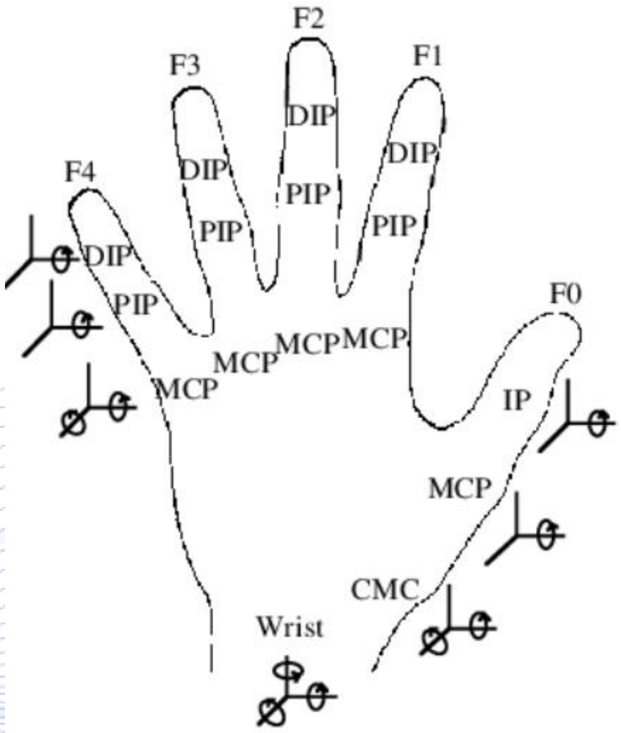
Gesture Recognition

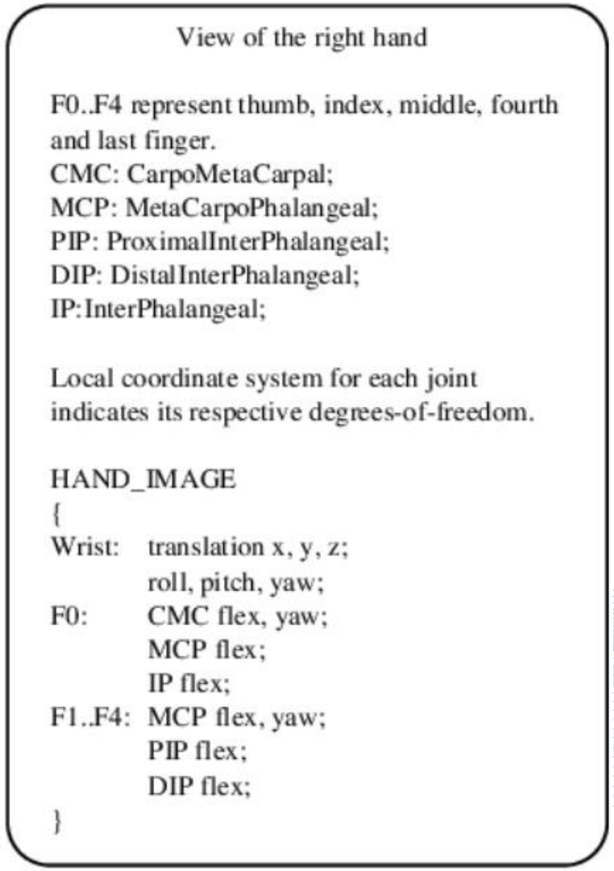
- a process where users perform gestures
- receiver recognizes them
- Goal: interpretation of human gestures through mathematical structures and algorithms.
- A way for computers to understand human body language.

Introduction

- Human gestures from visual data are analyzed by:
  - Computer Vision
  - Machine Learning
- Data sources:
  - Visual: RGB, Depth, Thermal images
  - Wearable: Magnetic field trackers, body suits, instrumented gloves (active or passive)
  - Audio

Hand morphology



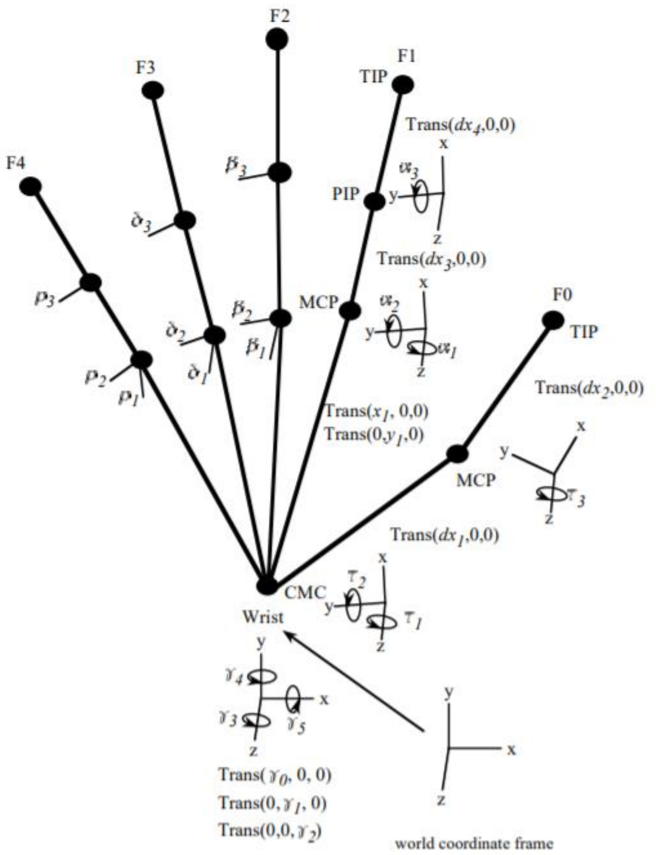

View of the right hand

F0..F4 represent thumb, index, middle, fourth and last finger.
CMC: CarpoMetaCarpal;
MCP: MetaCarpoPhalangeal;
PIP: ProximalInterPhalangeal;
DIP: DistalInterPhalangeal;
IP:InterPhalangeal;

Local coordinate system for each joint indicates its respective degrees-of-freedom.

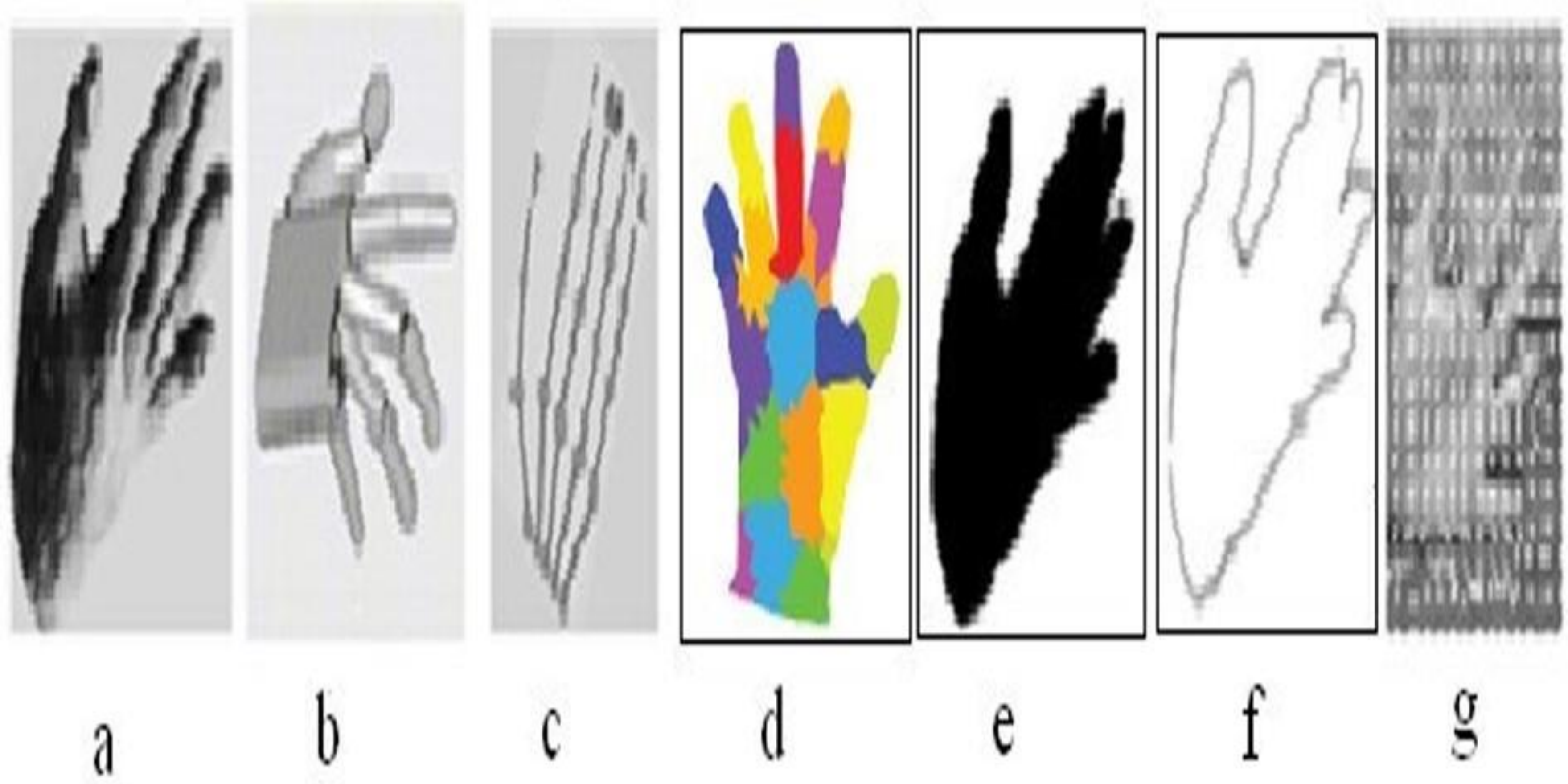
```
HAND_IMAGE
{
Wrist:     translation x, y, z;
           roll, pitch, yaw;
F0:        CMC flex, yaw;
           MCP flex;
           IP flex;
F1..F4:    MCP flex, yaw;
           PIP flex;
           DIP flex;
}
```

Hand morphology



Hand skeleton model.

Hand morphology



a) 3D volumetric hand model; b) 3D geometric model; c) 3D Skeleton model, d) colored marker based model; e) Non-geometric shape model (binary silhouette); f) 2D deformable template model; g) Motion based model [RAF2012].

Gesture Recognition

- Introduction
- **Gesture types**
- Gesture Acquisition Devices
- Human-Machine Interaction
- Gesture Recognition Datasets
- Gesture Recognition Algorithms
- Deep Gesture Recognition
- Skeleton-Based Gesture Recognition
- Multimodal Gesture Recognition
- Egocentric Gesture Recognition
- Applications

Gesture types

Gesture types based on variousbody parts.

- ***Hand Gestures***: waving goodbye, showing points…
- ***Head Gestures***: nodding, winking…
- ***Body Gestures***: kicking, raise knee or elbow…

Gesture types

Gesture types based on duration.

- ***Static Gestures***: static body (part) posture in an instant time.
  - An image of the position of the body part (showing the thumb…).
- ***Dynamic Gestures***: changeable pose over a small period of time (waving palm…).

Gesture types

Gesture types based on timing information.

- ***Online gestures***:
  - Instant interpretation of gestures. They are used to manipulate an object (scaling, rotation).
- ***Offline gestures***:
  - They are processed after the procedure of the user interaction with the object.
  - Example: the gesture to activate a menu.

Gesture Recognition

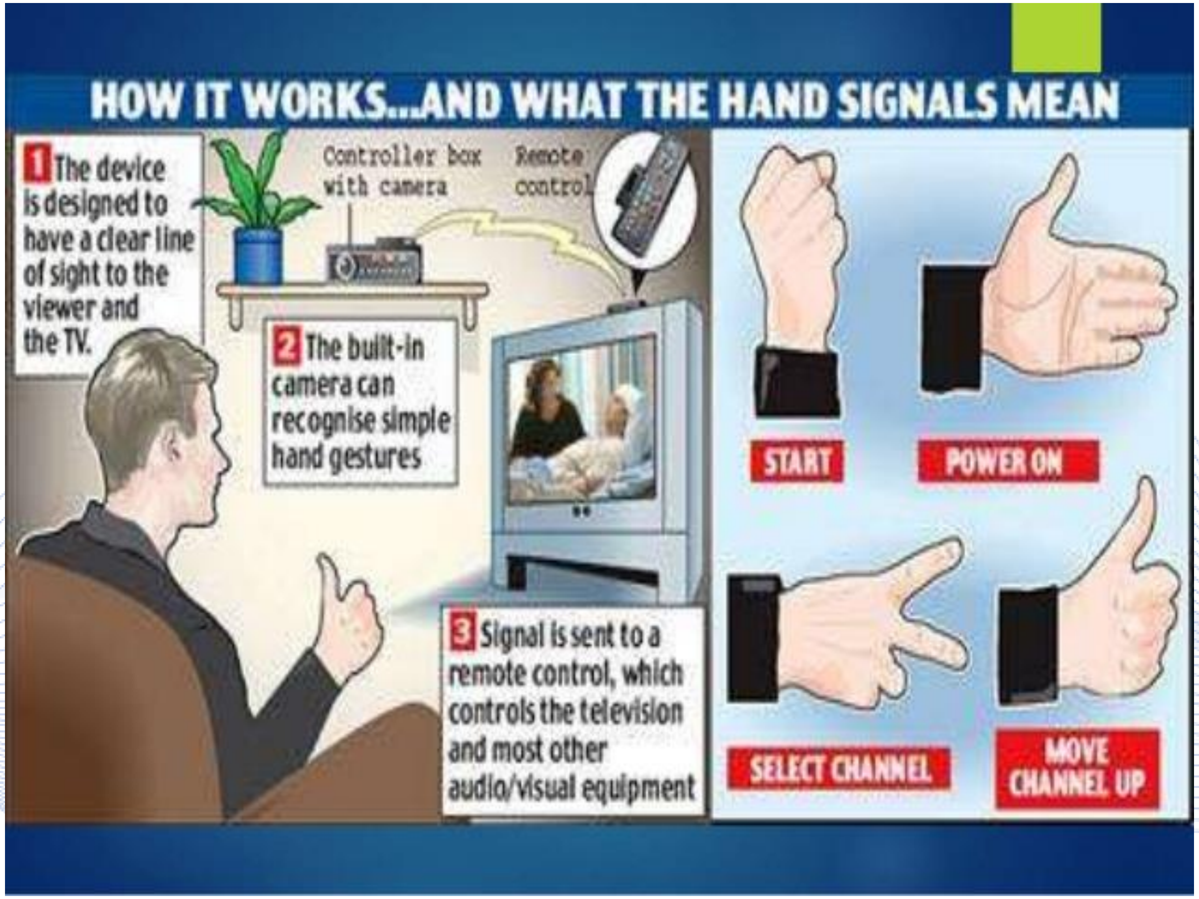
- Introduction
- Gesture types
- **Gesture Acquisition Devices**
- Human-Machine Interaction
- Gesture Recognition Datasets
- Gesture Recognition Algorithms
- Deep Gesture Recognition
- Skeleton-Based Gesture Recognition
- Multimodal Gesture Recognition
- Egocentric Gesture Recognition
- Applications

RGB Cameras

Digital cameras produce red, green and blue color channels.

- High image resolution.
- Cheap and ubiquitous sensors.
- RGB videos can be used to recover depth information.

Sensor Technologies


HOW IT WORKS...AND WHAT THE HAND SIGNALS MEAN
1 The device is designed to have a clear line of sight to the viewer and the TV.
Controller box with camera
Remote control
2 The built-in camera can recognise simple hand gestures
3 Signal is sent to a remote control, which controls the television and most other audio/visual equipment
START
POWER ON
SELECT CHANNEL
MOVE CHANNEL UP

Sensor Technologies

- Stereo Camera
  - Two lens with the same distance apart.
  - It simulates human vision.
  - It indirectly conveys 3D information.

- Depth cameras
  - They produce depth image maps
  - They may produce registered RGB and depth images.

Sensor Technologies

- Thermal Cameras
  - It images the infrared radiation emanating from a target object.

- Proximity sensors:
  - Used in situation when other forms of image-based recognition is inconvenient.

Depth Data

Depth images: a map of per-pixel data containing depth-related information.

  - Disparity or depth map.
  - Conversion methods, focus information, and camera calibration.

- A depth map pixel describes the distance to an image object from the camera plane.

- Horizontal ***disparity***:

$$d = x_r - x_l \leq 0.$$

- d is inversely proportional to scene depth Z_w (by triangle similarity):

  - $d = -f\frac{T}{Z_w}$.

Depth Data

- Depth sensing devices produce depth data.
- They can provide automatic body part segmentation via skeleton tracking.

- Advantage:
  - track and segment hands automatically, providing depth data related specifically to the hands

Depth Data

Microsoft Kinect.

Depth Data

Microsoft Kinect

- Kinect creates a skeletal representation of the body real-time for RGB video and depth images.
- The ***body skeleton*** is represented as a graph with vertices the joints of the body that shows the main parts of the body such as the head, the hands or the legs.
- For each joint we extract its 3D coordinates that identify every single joint.

Depth Data

Microsoft Kinect

- It consists of an infrared laser projector combined with a monochrome CMOS sensor.
- Captures video data in 3D under any ambient light condition.
- Achieves a rate of 30 frames per second of depth sensing.
- Infrared projector sends out modulated infrared light.
- Light reflecting off closer objects will have a shorter time of flight than those more distant.
- Infrared sensor captures the amount deformation of the modulation pattern from the time of flight, pixel-by-pixel.

Depth Data

Microsoft Kinect

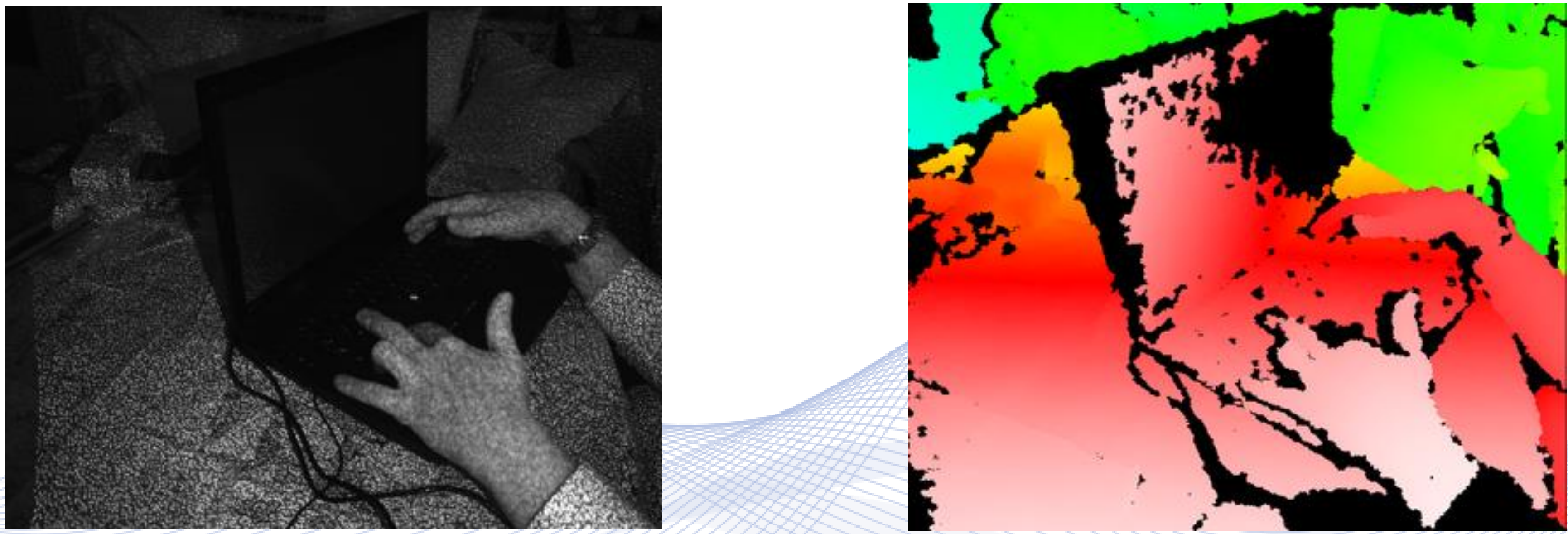
- Uses a type of edge detection to delineate closer objects from the background of the shot.
- Track moving objects.
- Assumption: only people will be moving around in the image.
- Isolate human shapes from the image.
- Shape segmentation can be performed to identify specific body parts like the head, arms, and hands, and track those segments individually.
- Construct a ***20-point skeleton*** of the human body.

Depth Data

Microsoft Kinect

- Includes a QVGA (320×240) depth camera and a VGA (640×480) video camera, which produce image streams at 30 fps.
- Microsoft developed the Kinect for full-body tracking to be used for interacting with games, videos and menus on the Xbox 360 game console.
- The proprietary body-tracking methods, as well as access to the depth and video streams, are avavailabe through a closed-sourse Kinect SDK or through the open-source OpenNI4 framework.

Depth Data



a) Infrared image taken by the Kinect infrared (left); b) pseudocolored depth image.

Depth Data

- A common method to isolate the hands is with depth thresholding.
- This determines the hands as the points between some near and far distance thresholds around the depth of the expected predetermined 3D hand centroid.
- An effective method of reducing noise sensitivity is also to place limits on the area of the detected hand.

Depth Data

Depth data Preprocessing

- The raw data we get from the sensors need to be processed in order to get the desired results from our model.
    - E.g., the images may come from different sources or they have many missing or irrelevant information.
- Data registration.
- Data segmentation.
    - Segment the raw depth data into skeleton and joint representation

Data Augmentation.

- Augmenting the existing dataset by slightly modified versions of the existing images, like scaling, rotations etc.

Depth Cameras

RGBD cameras provide a map for the depth in each pixel. In this way, the depth map describes the distance of an object.

- The advantage of this data is that body segmentation can be performed and body skeleton can be detected and tracked.

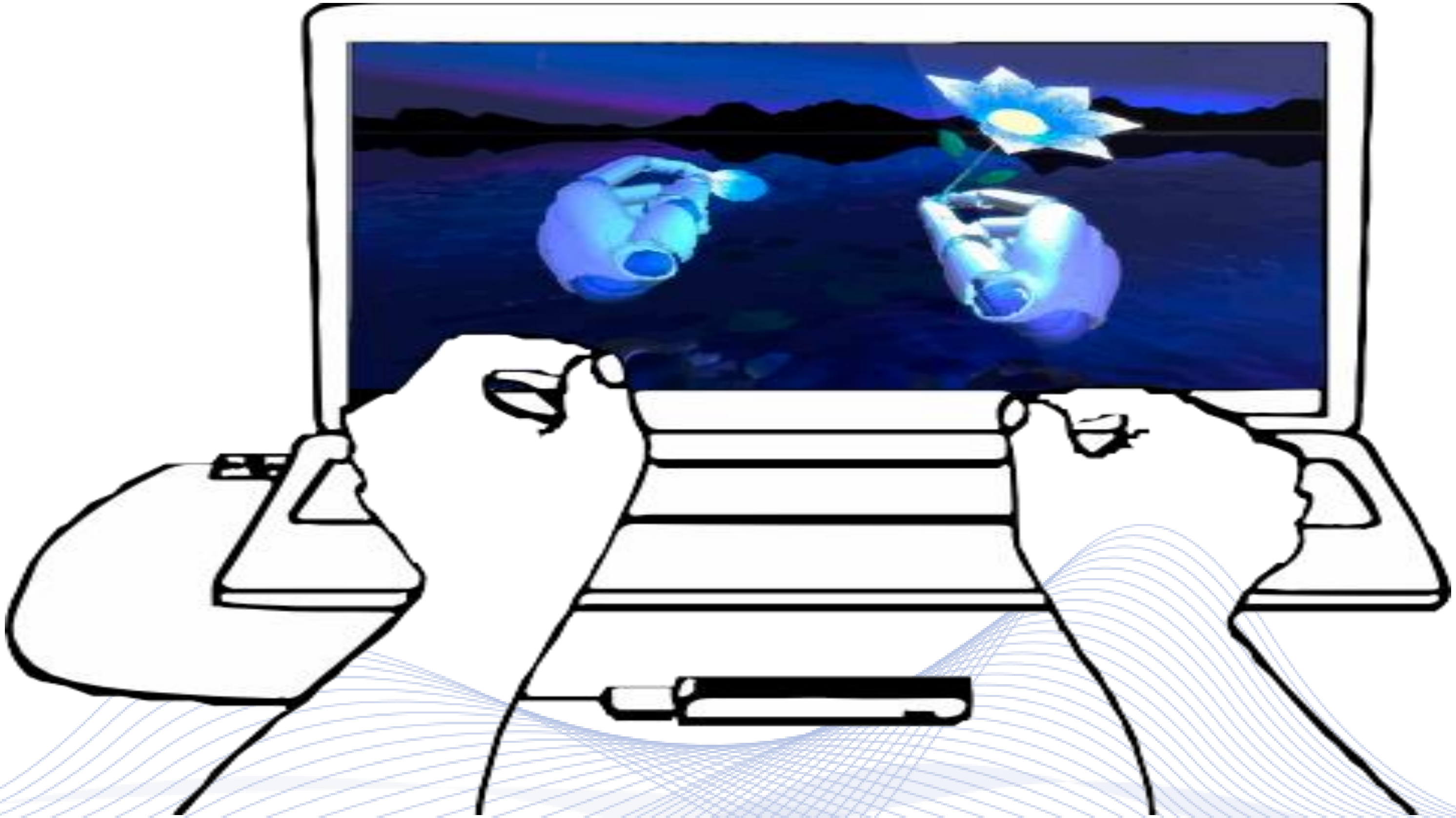
- In this way, hands can be tracked and the segmented.

- Image distortions may be present, due to lens imperfactions.

Leap Motion

Leap Motion device is compact and economical and it is used for gesture recognition.

- It can track the 3D forearms, the hands and the fingers in real time.
- It consists of two infrared cameras, situated in an 120° angle and three LEDs on infrared radiation.
- Up to 200 fps (frames per second) can be recorded.



Leap Motion operating mode [JES2020].

Sensor Technologies

Wired Gloves

- Provide input to the computer about position and rotation
- Magnetic or inertial devices
- Data Glove
  - A glove-type device
  - Detects hand position, movement and finger bending
- Fiber gloves use fiber optic cables
  - Light pulses are created when fingers bent
  - Light leaks thought small cracks giving the approximation of hand pose

Gesture Acquisition Devices

Data Gloves

  - Gloves can be used to calculate the hand position and the motion.

- ***Active gloves*** have a sensor or accelerometer.
- ***Passive gloves*** have colour markers for image identification.

Passive gloves to help differentiate finger position [JES2020].

Electromyography

Electromyography (***EMG***).

- Instead of gloves, wearable bracelets with electromyography sensors can measure the electrical signals from the muscles.
- EMG has been used for medical diagnosis, control of prosthetics and for the rehabilitation after severe musculoskeletal injuries.

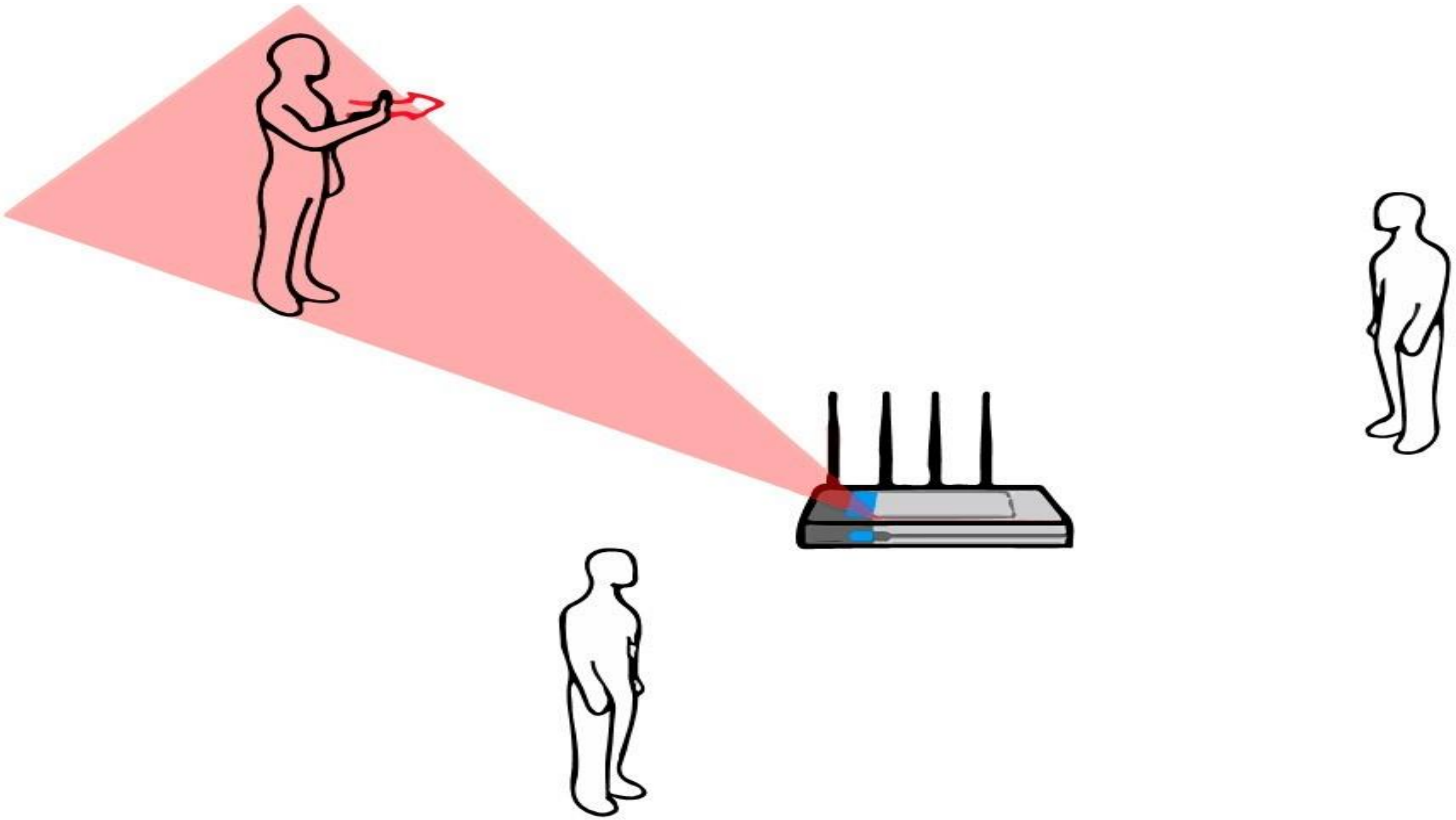
Ultrasound

Sonomyography uses ultrasound images and provide the observation of the muscles in the human body on a real-time.

- A technique recording based on the ***Doppler Effect*** uses ultrasonic frequency signals, where a device emits ultrasonic continuous tones.

WiFi

- WiFi has the ability to perform ***Non-Line Of Sight*** (NLOS) gesture recognition.
- There are already researches for the strength of a signal indicator-RSSI, also on the signal of the indicator of flight time –ToF, or where there is an observation of the channel status information-CSI.



WiFi recognition of different hand positions [JES2020].

Radio Frequency Identification

Radio Frequency Identification (***RFID***) systems embody ultra-high frequency-UHF readers from the commercial market, which they can detect labels.

- Information such as ***phase change*** from received UHF signals and they can be used for gesture recognition with very good results.

Gesture Recognition

- Introduction
- Gesture types
- Gesture Acquisition Devices
- Human-Machine Interaction
- **Gesture Recognition Datasets**
- Gesture Recognition Algorithms
- Deep Gesture Recognition
- Skeleton-Based Gesture Recognition
- Multimodal Gesture Recognition
- Egocentric Gesture Recognition
- Applications

Gesture Recognition Datasets

- Recently, deep learning models are used for gesture recognition.
- Data inputs of various modalities(the skeleton joints, the shape of the body of human, RGB, the optical flow, and the depth frames) are combined for the training of these models.
- There is a significant number of datasets which had been created for gesture recognition.

DVS128 Gesture Dataset

- It consists of ***11 hand gestures*** which are performed by 29 participants in 3 different illuminations [DVS].
- Each trial consists of 2 files: the data file which contains the events of DVS128, and the annotation file which describes the time stamps of the beginning and of the end for each gesture.

3D Skeletal Dataset for Hand Gesture Recognition

- There are ***14 hand gestures*** in this dataset executed in 2 ways: with a specific finger and with the whole hand [SKD].
- There are 2800 sequences, every gesture is performed in a range of 1 to 10 times by 28 subjects in both ways.
- In every sequence has gesture, number of the used fingers, user and trial information.
- In the 2D depth image space and in the 3D world space, there are 22 joints, which form a full hand skeleton. Their coordinates are contained in each frame of sequences.
- The videos were recorded at 30 frames per second.

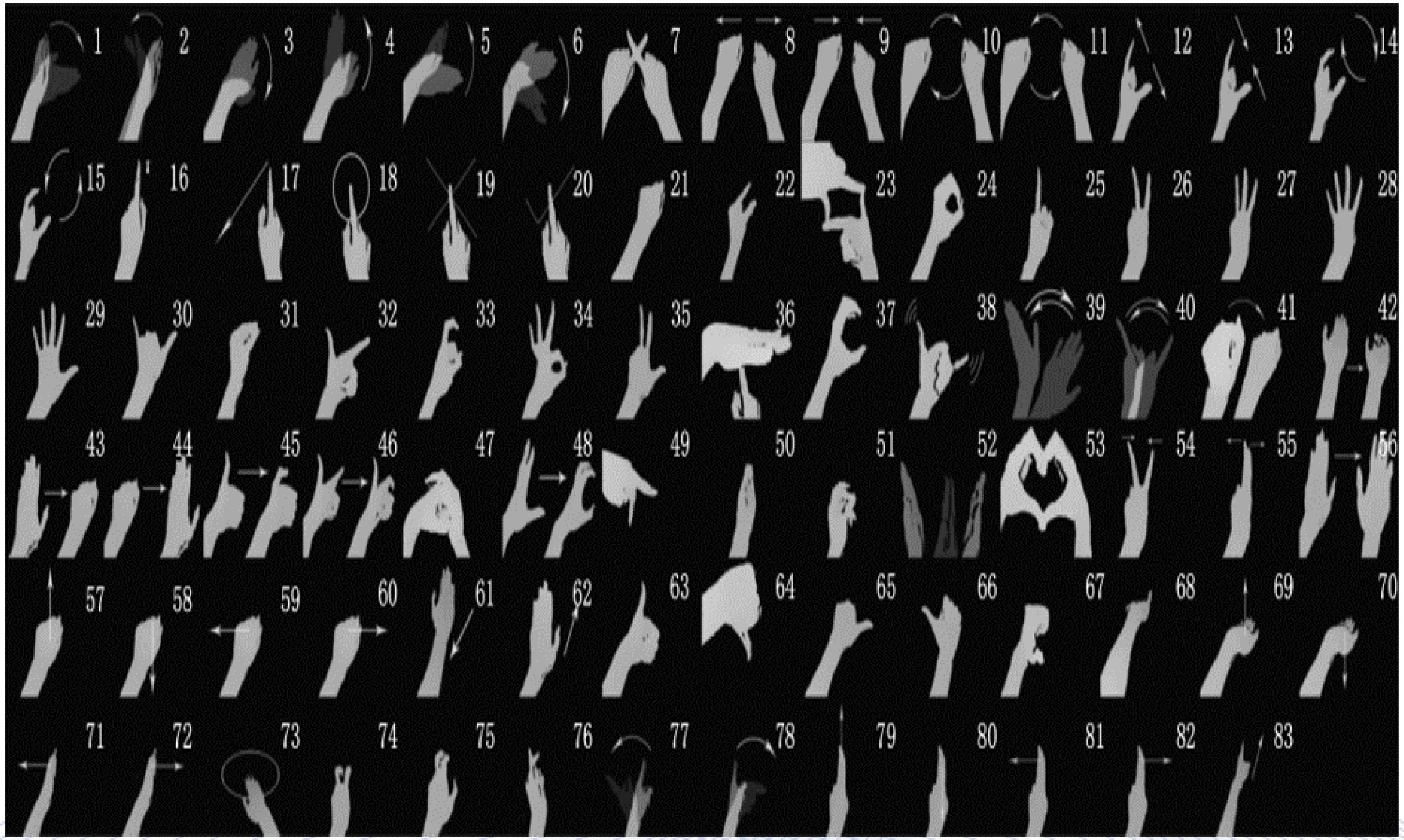
HGM-4 dataset

- The HGM-4 dataset is used for hand gesture recognition [GM4].
- It consists of 4,160 color images (1280×700 pixels) of ***26 hand gestures*** which are recorded by 4 cameras, each one at a different position.
- The images were taken indoor at different positions and the background was removed semi-automatically.
- This dataset can be used for Multiview hand gesture recognition.
- Every image from 4 cameras were combined to be in the set for training or in the testing set with all possible combinations.

EgoGesture Dataset

EgoGesture dataset consists of 2081 RGB-D videos, 24161 samples of gestures and 2,953,224 frames from 50 participants [EGO2018].

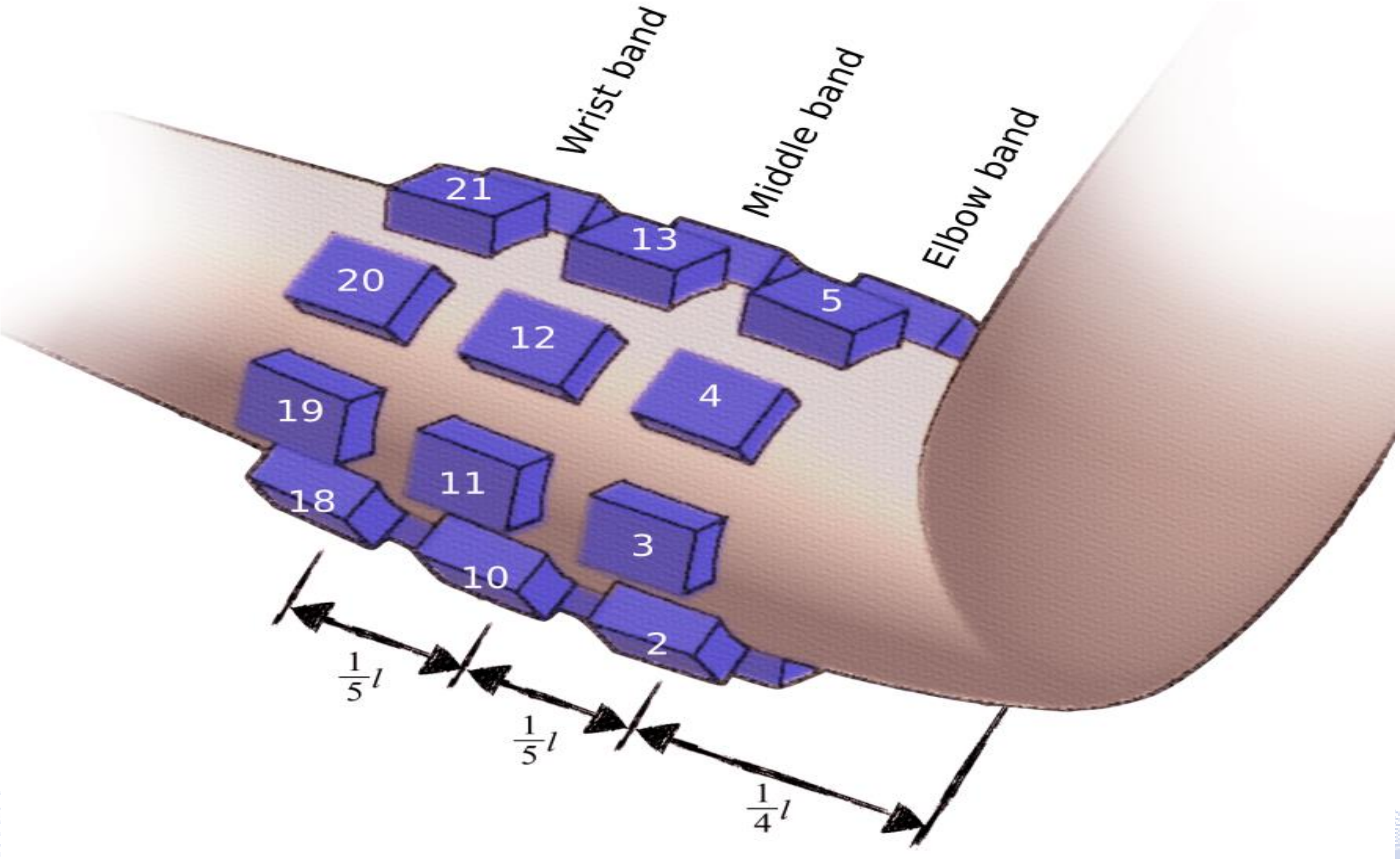
- There are static or dynamic gestures classified in ***83 gesture classes***. They are performed with wearable devices.
- The videos were made in 6 different scenes: 4 indoor and 2 outdoor ones. There are videos where the gestures were performed by people, while they were walking.
- The RGB and depth videos are captured with a resolution of $640\text{x}480$ pixels at 30 fps.
- The gestures were performed in random order. Thus the videos can be applied for the evaluation of the gesture detection in sequence.
- The volume of the data is about 46 Gbyte of RGB-D videos and about 32Gbyte of 320×240 pixel images.



The 83 gesture classes of the EgoGesture dataset [EGO2018].

PUTEMG and PUTEMG-FORCE

- They are datasets of ***electromyographic activity***, which was captured on the forearm surface [PEM].
- Signals were captured from 24 electrodes, which were fixed around participant right forearm with the use of 3 elastic bands, with the creation of a 3×8 matrix, Data are sampled at the rate of 200 Hz.
- The dataset contains **7 active gestures** (hand flexion, extension…) + idle and a set of trials with isometric contractions.
- They are used for gesture recognition and for grasp force recognition.
- While a gesture was executed, a HD Camera giving an RGB feed and a depth camera with a close view of hand of the participant were utilized. Gesture depth images and videos are attached with EMG data.



Electromyography signal acquisition [PEM].

HMD Gesture Dataset

- This dataset contains about 360,000 image pairs for gesture recognition. They were recorded using 31 participants and 30 different environments [HMG].
- The images are recorded by a stereo monochrome fisheye pair mounted in front of an HMD system.
- The dataset contains ***8 gesture classes***. For each image pair, the following information is provided: a gesture class label and a bounding box where hands are located.
- Images also may contain cluttered background and exacting lighting conditions.

WLASL

WLASL dataset is a large-scale signer-independent ***American Sign Language (ASL)*** dataset containing 34404 videos [DON2020].

- Ddata annotations:
  - Temporal sign boundary.
  - Body Bounding-box.
  - Signer Diversity: there are inter-participant variations (example: participant appearance and signing pace).
  - Dialect Variation Annotation: the dialect variations of signs containing different sign primitives, like hand-shapes and movements, have been annotated for each gloss (written approximation of ASL).

20BN-jester Dataset V1

- This is a hand gesture dataset. The subjects performed hand gestures in front of a webcam or laptop camera [JES].
- The video data is split into parts of 1 GB. The number of the videos is 148092 and their total size is 22.8 GB.
- The number of videos for the training, testing and validation sets are 118562, 14743 and 14787, respectively.

UAV-Gesture

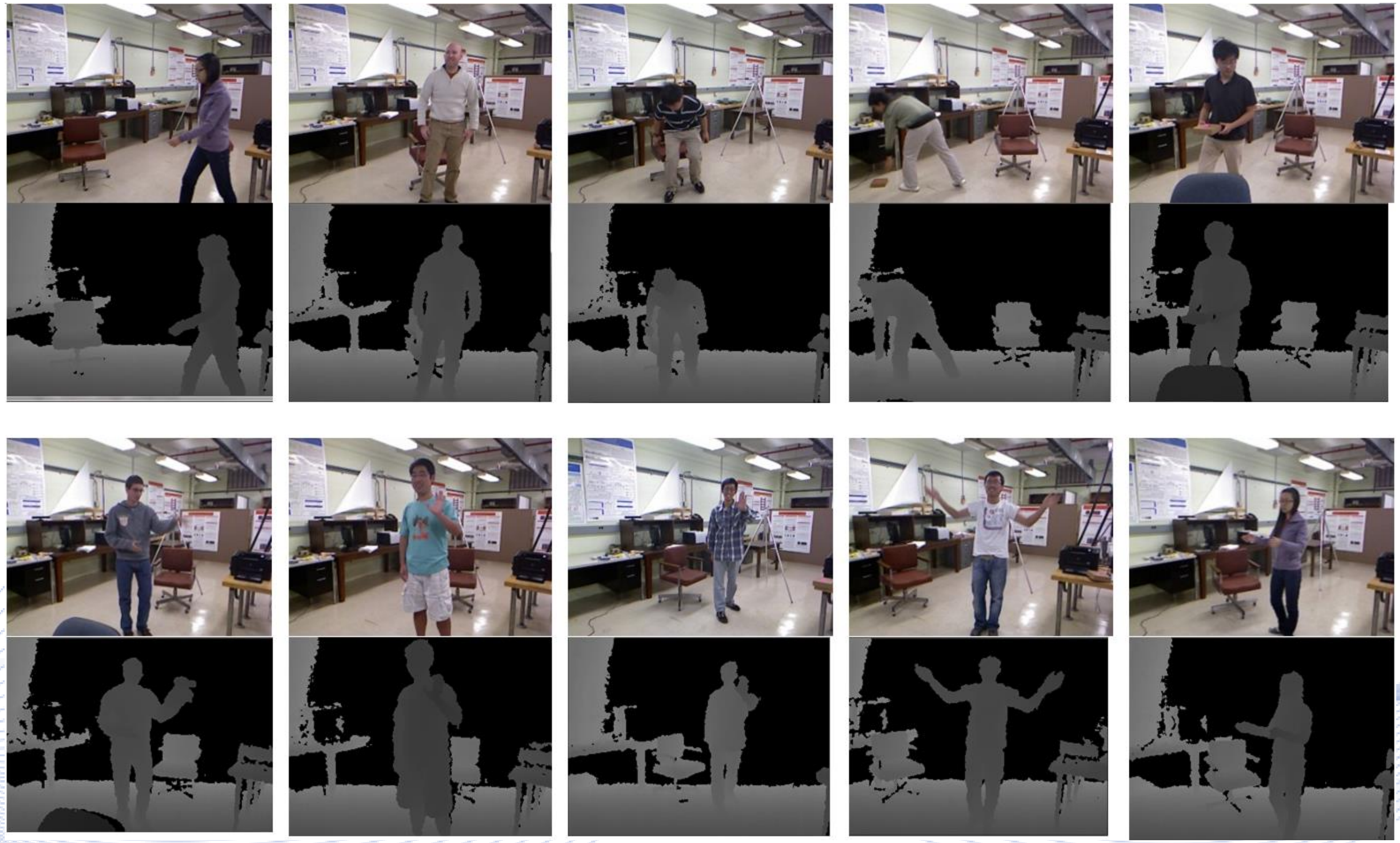
- The data was recorded in a wheat field from a rotorcraft UAV (3DR Solo) in low-altitude and slow flight [ASA2018].
- The videos have a HD resolution (1080×1920 pixels) at 25 fps.
- In the videos, the participant is located in the center of the frame and executes ***13 gestures***.
- Each of gesture is performed between five to ten times.
- There is an annotation of 13 body joints in 37151 frames, e.g., for ankles, knees, hip-joint, wrists, elbows, shoulders and head.
- Each annotation has also the gesture class, the participant identity and the bounding box.

Datasets

UT-Kinect

- Contains ***10 different gestures*** such as push, pull, pick up etc.
- These are carried out by 10 subjects and captured using Kinect.
- The participants move around creating gestures that have different starting, illumination, orientation etc.
- The data contains three synchronized channels for RBG, depth and skeleton data.

Datasets



UT-Kinect Dataset.

Datasets

DHG 14/28

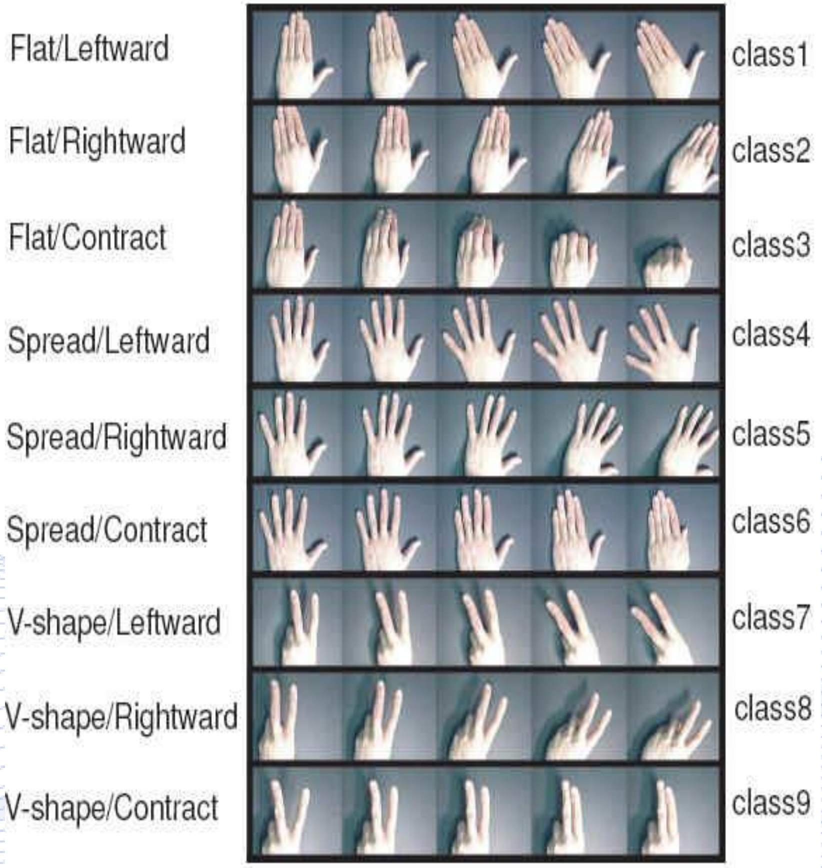
- Dataset involved 20 participants making ***14 different gestures*** using either one finger or the whole hand for 5 times.
- Every frame contains the depth map and the coordinates of 22 joints that create the skeleton data.
- The depth images were collected using the intel RealSense camera with resolution of 640×480 and 30 fps.

Datasets

Cambridge Hand Gesture

- Has 9 classes of 100 images per class for right and left hands with 5 different illuminations and arbitrary motions.
- The samples were recorded using a fixed camera.
- Gestures are composed of flat, spread and V-shape hand shapes and leftward, rightward and contract motions.

Datasets



Cambridge Hand Gesture Dataset.

Artificial Intelligence &
Information Analysis Lab

Datasets

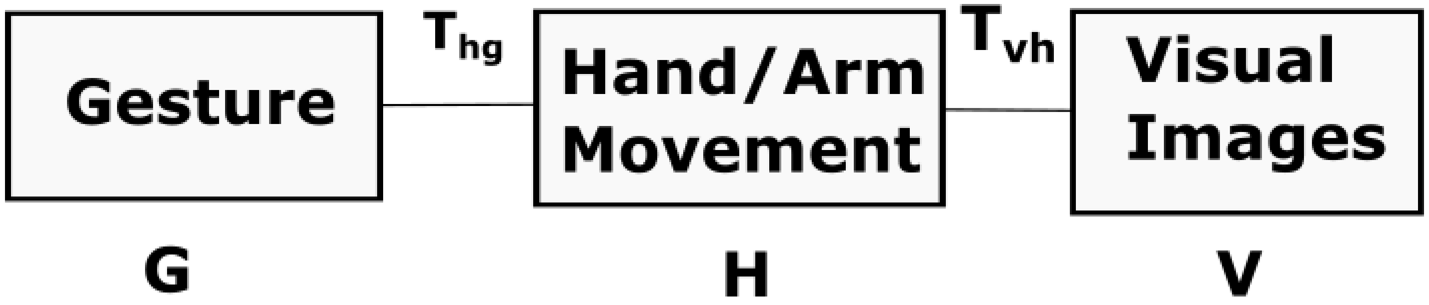
Leap Motion Dynamic Hand Gesture (LMDHG)

- Contains 608 motion and 526 rest gesture samples, corresponding to a total of 1134 gestures using a Leap Motion sensor.
- These gesture instances fall into ***14 gesture classes*** with unequal size:
  - Point to
  - Catch
  - Zoom
  - Rest
  - Scroll etc.

Gesture Recognition

- Introduction
- Gesture types
- Gesture Acquisition Devices
- Human-Machine Interaction
- Gesture Recognition Datasets
- **Gesture Recognition Algorithms**
- Deep Gesture Recognition
- Skeleton-Based Gesture Recognition
- Multimodal Gesture Recognition
- Egocentric Gesture Recognition
- Applications

Gesture Recognition
Problem Statement



Production and perception of gestures.

Gesture Recognition Problem Statement

- Gestures are a way of communication.
- Models used in language processing can be applied.
- The production and perception modeling:

$$H = T_{hg}G, \qquad V = T_{vh}H,$$

$$V = T_{vh}\left(T_{hg}G\right) = T_{vg}G.$$

- T: transformation of different models
- G: gestures, V: visual images, H: hand and arm motion.

Gesture Recognition Problem Statement

- Gestures are made in a dynamic process.
- Important to consider temporal characteristics of gestures.
- Three phases in gesture making:
  - ***Preparation***: preparatory movement
  - ***Nucleus***: definite form
  - ***Retraction***: returns to the resting position or repositions for the new gesture phase
- Consider a set of classes $\mathcal{D} = \{\mathcal{C}_i\}_{i=1}^{m}$ where m is the number of different gestures and $\mathbf{x}_i$ is a single gesture.

Gesture Recognition Problem Statement

Spatial Gesture Model

- 3D Hand/Arm Model:
  - ***Volumetric models***: describe the 3D visual appearance
  - ***Skeletal models***
  - Joints connecting the bones naturally exhibit different degrees of freedom (DoF)
    - The human hand skeleton consists of 27 bones, divided in three groups:
      - carpals (wrist bones—eight)
      - metacarpals (palm bones—five)
      - phalanges (finger bones—14).

Gesture Recognition Problem Statement

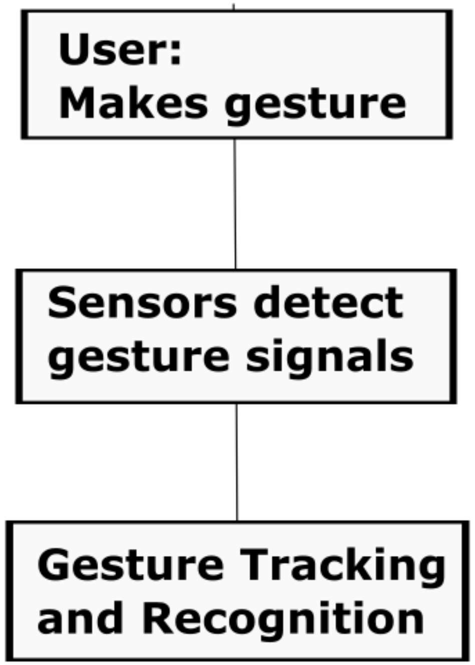
Spatial Gesture Model

- ***Appearance-Based Model***:
  - Based on the appearance in the visual image.
  - Deformable 2D templates
    - The sets of points on the outline of an object, used as interpolation nodes for the object outline approximation.
    - The simplest interpolation function used is a piecewise linear function.
    - The templates consist of the average point sets, point variability parameters, and so-called external deformations.
    - Point variability parameters describe the allowed shape deformation (variation) within that same group of shapes.

Gesture Recognition
Problem Statement

- Dynamic gestures present difficulties in recognition:
  - Time variability,
  - Space Complexity.
  - Starting and ending point is not clear,
  - Repetitiveness.
- Gesture can be considered as states.

Gesture Recognition Problem Statement



User:
Makes gesture

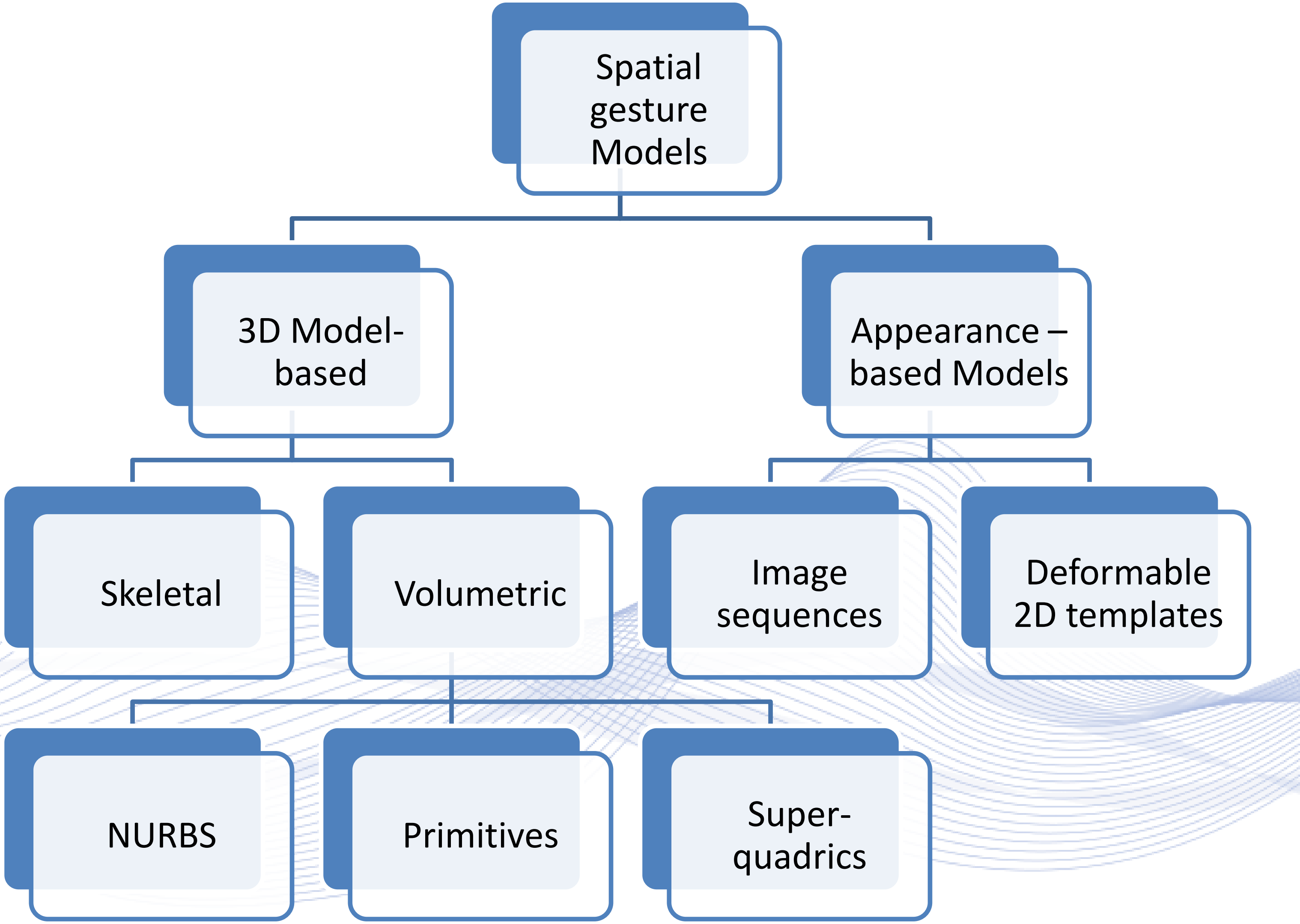
Sensors detect gesture signals

Gesture Tracking and Recognition

Algorithms of Gesture Recognition

- ***3D model-based algorithms***: the interpretation of an object as a list of vertices and lines in the 3D mesh version.
- ***Skeletal-based algorithms***: models the object but with less parameters than the version of volumetric.
- ***Appearance-based models***: acquire the parameters from the images or from the videos directly, with the use of a template database.
- ***Electromyography-based models***: classify the body movement with data of electrical signals generated by the muscles.

Tree of algorithms



Problems of Gesture Recognition

- ***Illumination condition***: the changes of the light in the image make the extraction of the skin region more difficult.
- ***Rotation***: there is problem when the subject that performs the gesture is rotated in any direction.
- ***Scaling***: the pose of the body or the hand have different sizes in the image of the gesture.
- ***Interpretation***: the changes of the position of the subject in different images, may give false representation of the features.
- ***Background***: when there is complex background with different objects that confuse the detection and lead to misclassification.

Gesture Recognition

- Introduction
- Gesture types
- Gesture Acquisition Devices
- Human-Machine Interaction
- Gesture Recognition Datasets
- Gesture Recognition Algorithms
- **Deep Gesture Recognition**
- Skeleton-Based Gesture Recognition
- Multimodal Gesture Recognition
- Egocentric Gesture Recognition
- Applications

Deep Gesture Recognition

DeepGRU

- Deep learning methods applied.
- An end-to-end network-based gesture recognition utility.
- Recurrent architecture neural network for action recognintion.
- Proposed by NVIDIA researchers M. Maghoumi and J. LaViola.
- Uses raw skeleton, pose or vector data.

Deep Gesture Recognition

DeepGRU

- Input: raw samples from the device represented as a temporal sequence of gesture images.
- Dimension of the feature vector is *N*, depends on the device.
- $\mathbf{x}_t \in \mathbb{R}^n$: the feature column vector at time *t*.
- $\mathbf{x} \in \mathbb{R}^{n \times L}$: the entire temporal sequence of a single sample, where L is the length of the sequence.
- Consider 3D position of 21 joints human skeleton in L time steps then

$n = 3 \times 21 = 63$ dimensional and if we double the number of time steps $\mathbf{x} \in \mathbb{R}^{63 \times L}$.

Deep Gesture Recognition

DeepGRU

- Encoder network serves as feature extractor.
- Input is the all the training samples of gestures collected.
- Consists of five stacked unidirectional GRUs.
- GRUs are simpler and faster to train for small number of parameters.
- Consider input $\mathbf{x}_t$ and hidden state vector of previous time step $\mathbf{h}_{t-1}$.

$$\mathbf{r}_t = \sigma((\mathbf{W}_x^r x_t + \mathbf{b}_x^r) + (\mathbf{W}_h^r \mathbf{h}_{t-1} + \mathbf{b}_h^r))$$

$$\mathbf{u}_t = \sigma((\mathbf{W}_x^u x_t + \mathbf{b}_x^u) + (\mathbf{W}_h^u \mathbf{h}_{t-1} + \mathbf{b}_h^u))$$

$$\mathbf{c}_t = tanh((\mathbf{W}_x^c x_t + \mathbf{b}_x^c) + r_t(\mathbf{W}_h^c h_{t-1} + \mathbf{b}_h^c))$$

$$\mathbf{h}_t = \mathbf{u}_t \circ \mathbf{h}_{t-1} + (1 - \mathbf{u}_t) \circ \mathbf{c}_t$$

Deep Gesture Recognition

DeepGRU

- $\mathbf{r}_t$: reset
- $\mathbf{u}_t$: update
- $\mathbf{c}_t$: candidate gates
- $\mathbf{W}_p^q$: trainable weights
- $\mathbf{b}_p^q$: biases
- $\mathbf{h}_o$: initially set to zero for all GRUs
- σ: the sigmoid function

Deep Gesture Recognition

DeepGRU

- The output of the encoder network is a set of features for performing classification.
- Given all hidden states $\mathbf{h}$ attention module computes the attentional context vector $\mathbf{c} \in \mathbb{R}^{128}$:

$$\mathbf{c} = \left(\frac{\exp(\boldsymbol{h}_{L-1}^{T} \boldsymbol{W}_c \boldsymbol{h})}{\sum_{t=0}^{L-1} \exp(\boldsymbol{h}_{L-1}^{T} \boldsymbol{W}_c \boldsymbol{h}_t)} \right).$$

Deep Gesture Recognition

DeepGRU

- Contextual feature vector is formed through concatenation $[\boldsymbol{c}; \boldsymbol{h}_{L-1}]$.
- Auxiliary context $\mathbf{c} = \Gamma_{\text{attn}}(\mathbf{c}, \mathbf{h}_{L-1})$, Γ_{attn} is the attentional GRU.
- Attention module output $o_{attn} = [\mathbf{c}; \mathbf{c}']$
- Final layers consist of two FC layers, ReLU activation and the probability distribution is calculated through:

$$\hat{\mathbf{y}} = softmax\left(F_2\left(ReLU\left(F_1(o_{attn})\right)\right)\right).$$

- The input of F_1 and F_2 is processed with batch normalization and dropout. In the training process, cross-entropy is applied to reduce false predictions.

Dynamic Gesture Recognition with Recurrent 3D CNN

The method proposed by [MOL2015] for camera-based gesture recognition uses a Convolutional Neural Network (CNN) to classify dynamic hand gestures.

- VIVA challenge dataset used, which contains 19 different hand gestures and 855 intensity and depth video sequences acquired by Microsoft Kinect with a resolution of 115×250 pixels.
- Preprocessing with resampling to 32 frames using nearest neighbor interpolation and downsampling the intensity and depth of the images by by 2.

Dynamic Gesture Recognition with Recurrent 3D CNN

- The architecture of the convolutional neural network includes a low-resolution and a high-resolution network with parameters $\mathcal{W}_L$ and $\mathcal{W}_H$.
- Consider a gesture input $\mathbf{x}$ and a class $\mathcal{C}$ then the probability of the class-membership is given by:

$$P(\mathcal{C}|\mathbf{x}) = P(\mathcal{C}|\mathbf{x}, \mathcal{W}_L)P(\mathcal{C}|\mathbf{x}, \mathcal{W}_H).$$

- The class label is predicted:

$$c^* = argmaxP(\mathcal{C}|\mathbf{x}).$$

Dynamic Gesture Recognition with Recurrent 3D CNN

- The ***high resolution layer*** consists of four 3D convolutional layers, which are all followed by a max-pooling.
- After those layers, there are two fully-connected layers and finally a softmax layer.
- The final output of the high-resolution layer gives the probability of class-membership$P(\mathcal{C}|\mathbf{x}, \mathcal{W}_H)$.
- Similarly, the ***low-resolution network*** has the same architecture with different sizes of the convolutional kernels and outputs the probability of class-membership $P(\mathcal{C}|\mathbf{x}, \mathcal{W}_H)$.

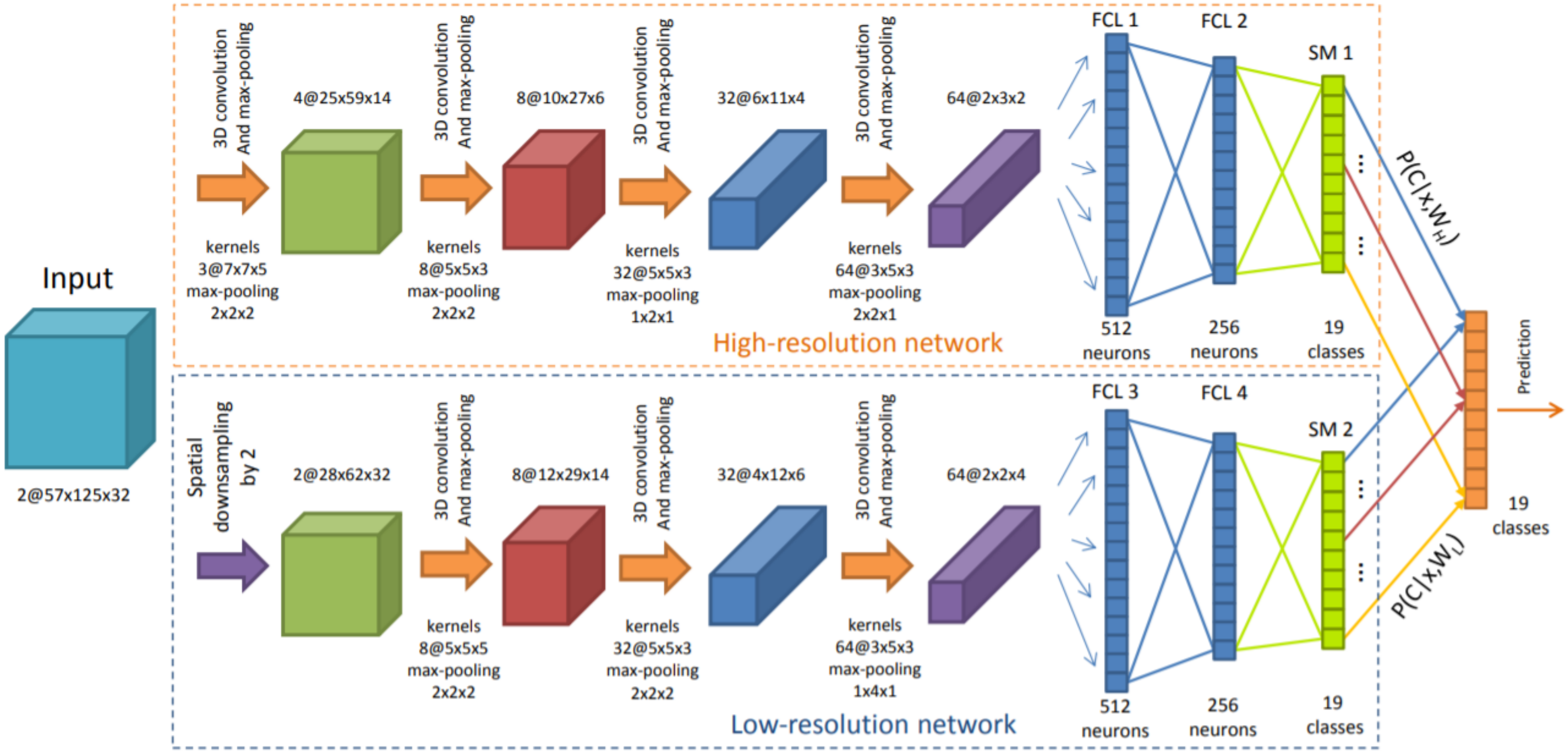
Dynamic Gesture Recognition with Recurrent 3D CNN

- All the layers except the softmax used the ReLU activation function.
- The output of the softmax layers is given by:

$$P(\mathcal{C}|\mathbf{x},\mathcal{W}) = \frac{\exp(z_{\mathcal{C}})}{\sum_q \exp(z_q)}$$

for z_q the output of the neuron.

Dynamic Gesture Recognition with Recurrent 3D CNN


Input
2@57x125x32
3D convolution And max-pooling
4@25x59x14
kernels 3@7x7x5 max-pooling 2x2x2
3D convolution And max-pooling
8@10x27x6
kernels 8@5x5x3 max-pooling 2x2x2
3D convolution And max-pooling
32@6x11x4
kernels 32@5x5x3 max-pooling 1x2x1
3D convolution And max-pooling
64@2x3x2
kernels 64@3x5x3 max-pooling 2x2x1
High-resolution network
FCL 1
512 neurons
FCL 2
256 neurons
SM 1
19 classes
$P(C|x,W_H)$
Spatial downsampling by 2
2@28x62x32
3D convolution And max-pooling
8@12x29x14
kernels 8@5x5x5 max-pooling 2x2x2
3D convolution And max-pooling
32@4x12x6
kernels 32@5x5x3 max-pooling 2x2x2
3D convolution And max-pooling
64@2x2x4
kernels 64@3x5x3 max-pooling 1x4x1
Low-resolution network
FCL 3
512 neurons
FCL 4
256 neurons
SM 2
19 classes
$P(C|x,W_L)$
Prediction
19 classes

High and low resolution 3D CNN network architecture.

Dynamic Gesture Recognition with Recurrent 3D CNN

- The training of the network involves the minimization of the cost function:

$$\mathcal{L}(\mathcal{W},\mathcal{D}) = -\frac{1}{|\mathcal{D}|}\sum_{i=0}^{|\mathcal{D}|}\log\left(P\left(\mathcal{C}^{(i)}\middle|\mathbf{x}^{(i)},\mathcal{W}\right)\right).$$

- The optimization algorithm used for this task is stochastic gradient descent with mini-batches of 40 training samples for the LRN and 20 for the HRN.

Dynamic Gesture Recognition with Recurrent 3D CNN

- The parameters $w \in \mathcal{W}$ are updated at every iteration i using ***Nestorov Acelerated Gradient***:

$$\nabla w_i = < \frac{\delta \mathcal{L}}{\delta\,(w_{i-1})} >_{batch},$$

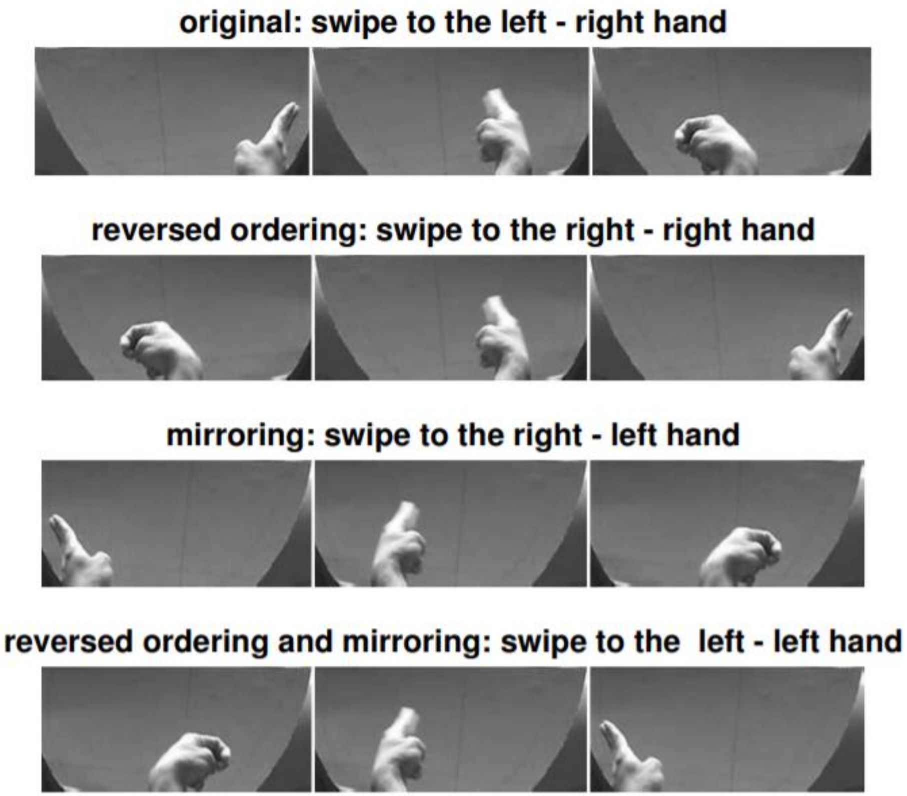
$$v_i = \mu\, v_{i-1} - \lambda\, \nabla w_i,$$

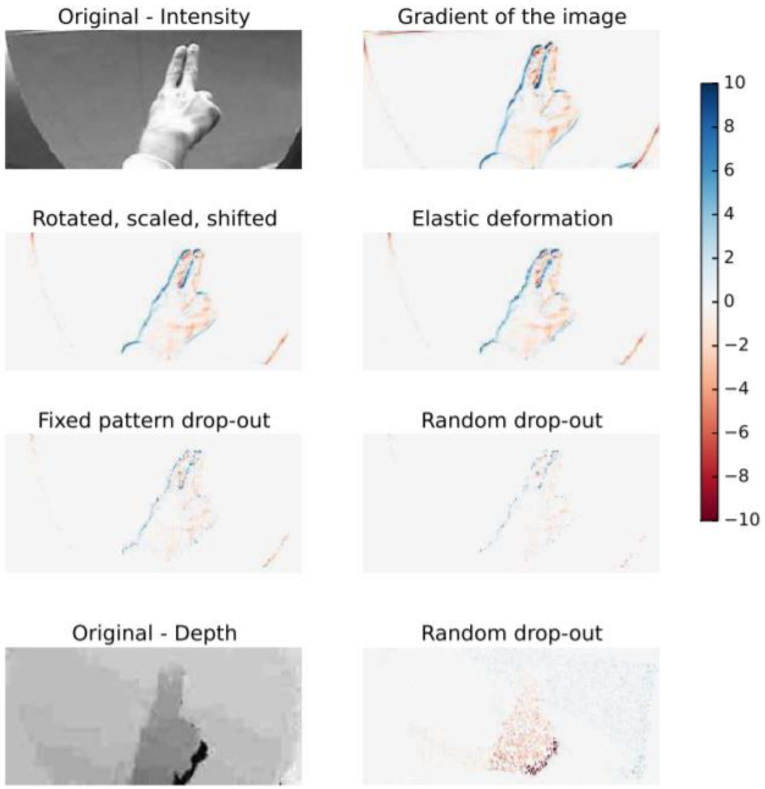
$$w_i = w_{i-1} + \mu\, v_i - \lambda\, w_i.$$

where λ is the learning rate, μ the momentum coefficient. The weights of the 3D convolution layers are initializes with values from the uniform distribution. The two LRN and HRN networks are trained separately.

Artificial Intelligence & Information Analysis Lab

Dynamic Gesture Recognition with Recurrent 3D CNN


original: swipe to the left - right hand
reversed ordering: swipe to the right - right hand
mirroring: swipe to the right - left hand
reversed ordering and mirroring: swipe to the left - left hand



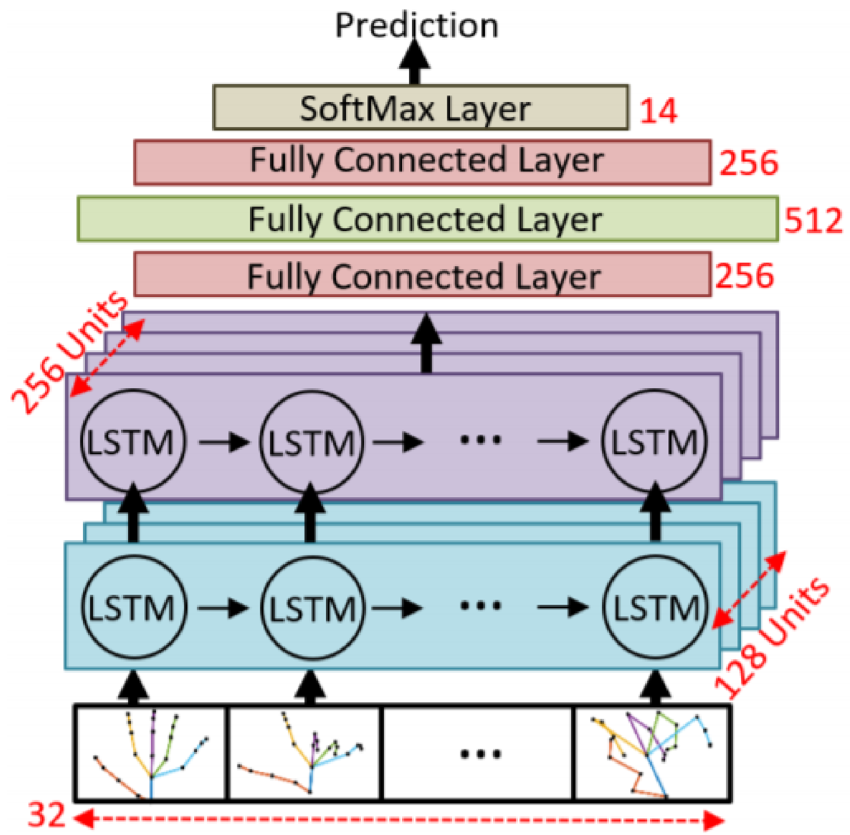
CNN RNN Depth and Skeleton based Gesture Recognition

- The paper [LAI2020] introduced a method that combines Convolutional Neural Networks with Recurrent Neural Networks (RNN) for hand gesture recognition using as input depth depth and skeleton data.
- This proposed method includes CNN and an RNN that uses features from depth sensors. There are two main components in the architecture:
  - the depth-based CNN and RNN and,
  - the skeleton-based RNN.

CNN RNN Depth and Skeleton based Gesture Recognition

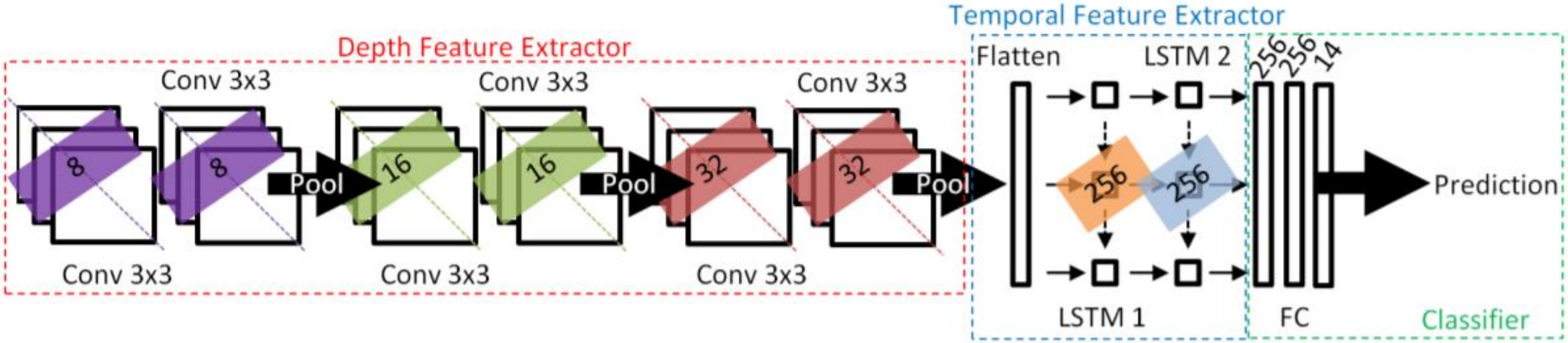
- For CNN+RNN module, the RNN processes the time series of the dynamic gesture images using two Long Short Memory network (LSTM) of 256 units.
- CNN consists of six 3 × 3 convolutional layers, each followed by a max pooling 2 × 2 layer and is used to extract features that are needed for the classification step using Multilayer Perceptron.
- Finally, for the decision-making step there are three fully-connected layers of 256, 265 and 14 units and a softmax layer.

CNN RNN Depth and Skeleton based Gesture Recognition



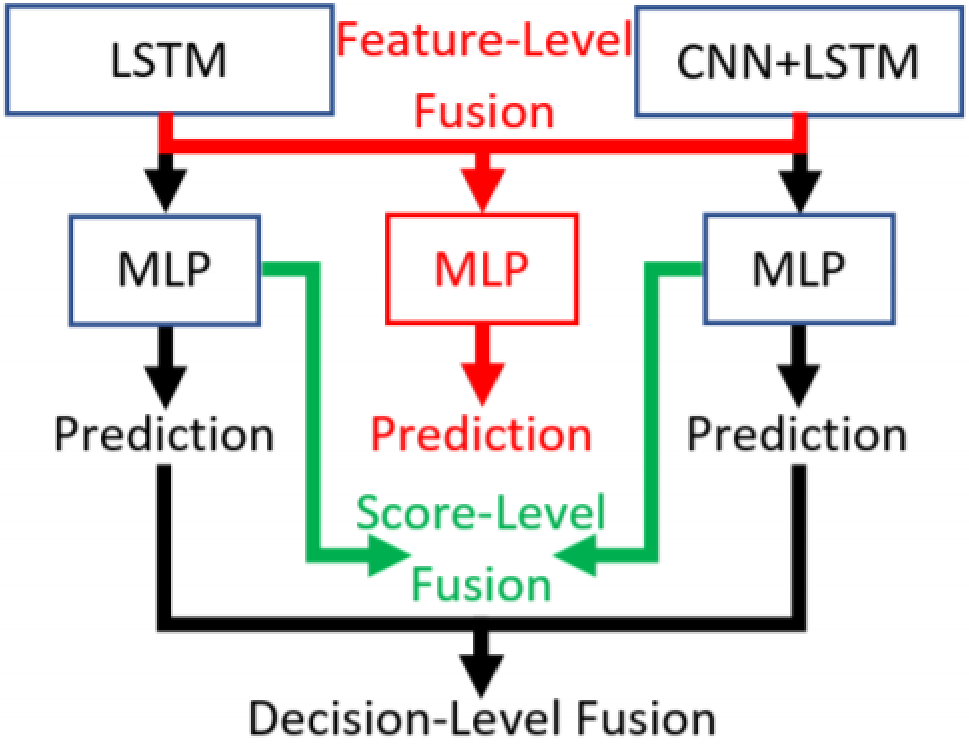
Skeleton based LSTM gesture recognition.

CNN RNN Depth and Skeleton based Gesture Recognition



Depth based CNN+LSTM gesture recognition.

CNN RNN Depth and Skeleton based Gesture Recognition



CNN+LSTM and LSTM gesture recognition.

CNN RNN Depth and Skeleton based Gesture Recognition

- Similarly, the RNN skeleton-based module extracts features from the skeleton data.
- The architecture is the same as the previous with the difference that there are LSTM units at each layer and one more fully connected layer.
- The researchers used feature-level, which is performed before the MLP, and score-level fusion, which is performed between the fully connected and softmax layer.

CNN RNN Depth and Skeleton based Gesture Recognition

- For the experimental results, the DHG-14/28 dynamic hand gesture dataset was used which has data collected from a depth sensor and contains , also, the skeleton data for many of them.
- The depth images were normalized, the images were cropped and kept only the region of the gesture.
- The 2D skeleton points in a sequence are normalized by subtracting every point by the palm location from the initial frame.

Dynamic Gesture Recognition via Hybrid Model

- The [LI2019] shows in the paper a hybrid deep learning model for dynamic gestures. The model consists of three different parts: a CNN, a MVRB and a NN.
- The CNN takes as input a video, which is a sequence of images with spatial information.
- For every frame, it produces a feature vector. When it obtains all features for a single video, the vectors are put all together and create a matrix.

Dynamic Gesture Recognition via Hybrid Model

- The CNN is pretrained using labeled frames.
- The model includes 5 convolutional layers with kernel size 15, 3 max-pooling layers, 2 fully-connected layers with 64 nodes and a softmax layer.
- The Matrix Variate Restricted Boltzmann Machine or MVRBM helps us get robust representation of the 3D hand gesture that is transformed into matrix form.

Dynamic Gesture Recognition via Hybrid Model

- The Neural Network that predicts the class label of a given video depicting a gesture.
- The model is trained using the input from MVBR.
- The weights are initialized and modified properly in order to discriminate with minimum number of errors the gestures.
- The next step involves a testing phase to prove how well the model predicts labels for the video samples.

Dynamic Gesture Recognition via Hybrid Model

- The data are preprocessed by removal of illumination variation and data augmentation.
- The experimental results show the performance of Neural Network without pretraining, using pretraining by the CNN-RBM-NN and finally using CNN-MVRBM-NN.
- The MVRBM and NN submodels are implemented by Matlab 2014a while the CNN by caffe VS2015.T
- he models are run on an Intel Core i7-4470 3.50 GHz CPU machine with 12G RAM.

3D CNN+LSTM ConvLSTM

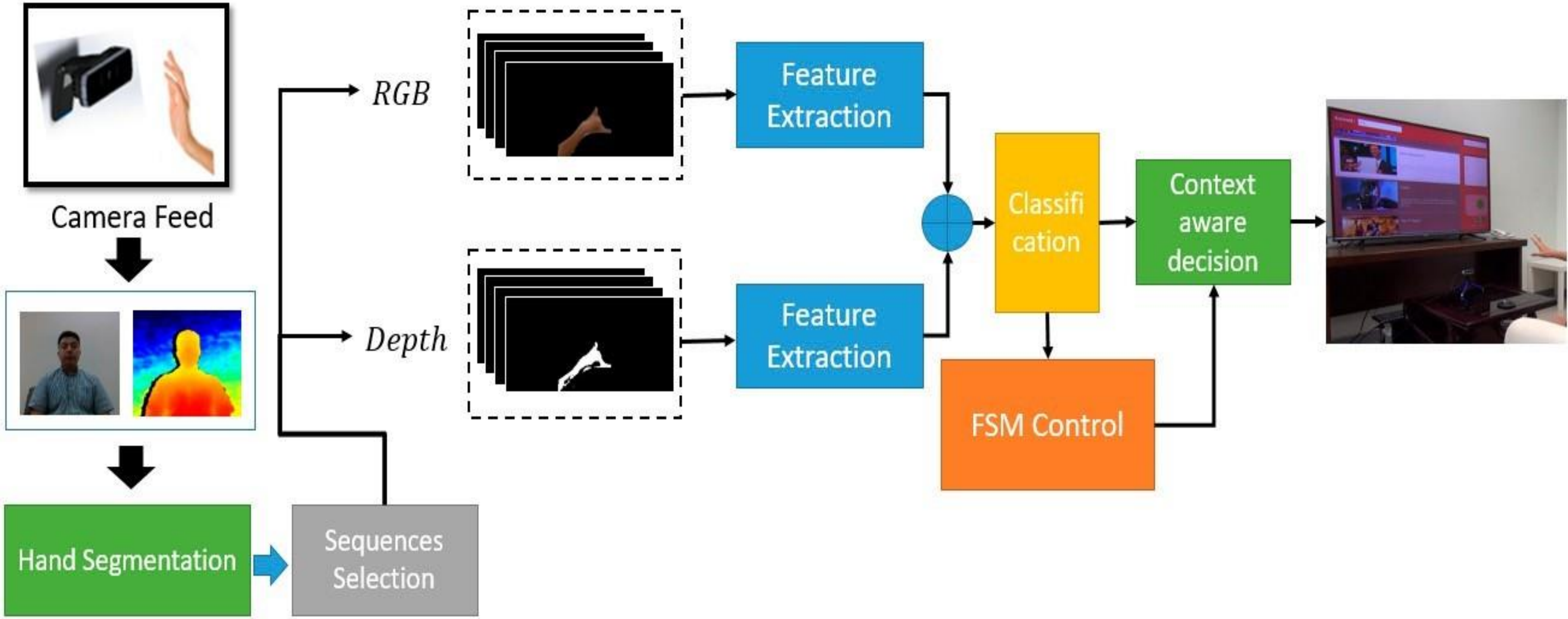
- The neural model combines a 3D convolution neural network connected with a long short-term memory (LSTM) [NOO2019].
- As input data there can be Depth data, RGB and multimodal data.
- The control with the Finite State Machine-FSM help to reduce some flows of the gesture and to border the classes of the recognition.
- The removing of the background and all the pixels that are unnecessary is made by the attention on the hand.
- The global side of the hand gesture recognition is analyzed. The input data includes the whole handshape instead of the use of the finger feature for classification.

3D CNN+LSTM ConvLSTM

- The datasets contains 2162 videos of participants while they perform 24 gestures, 13 static and 11 dynamic.
- Each gesture sequence contains a dynamic gesture of 3 seconds which consists of 120 frames.
- The sensor for the data was the depth camera **Real Sense SR300** thus the dataset includes the RGB and Depth data.
- For the extraction of the hand, given the whole RGB image I_r , and depth image I_d , fixed distance dt to cut off the background and minimum distance $\min of\ Id$ as the range filter was defined. Let I_{rb} be the RGB image and I_{db} be Depth image, after the cut off of the background. The calculation of the average distance of a point $d_{av}\ in\ Idb$ in the range $[\min, dt]$:

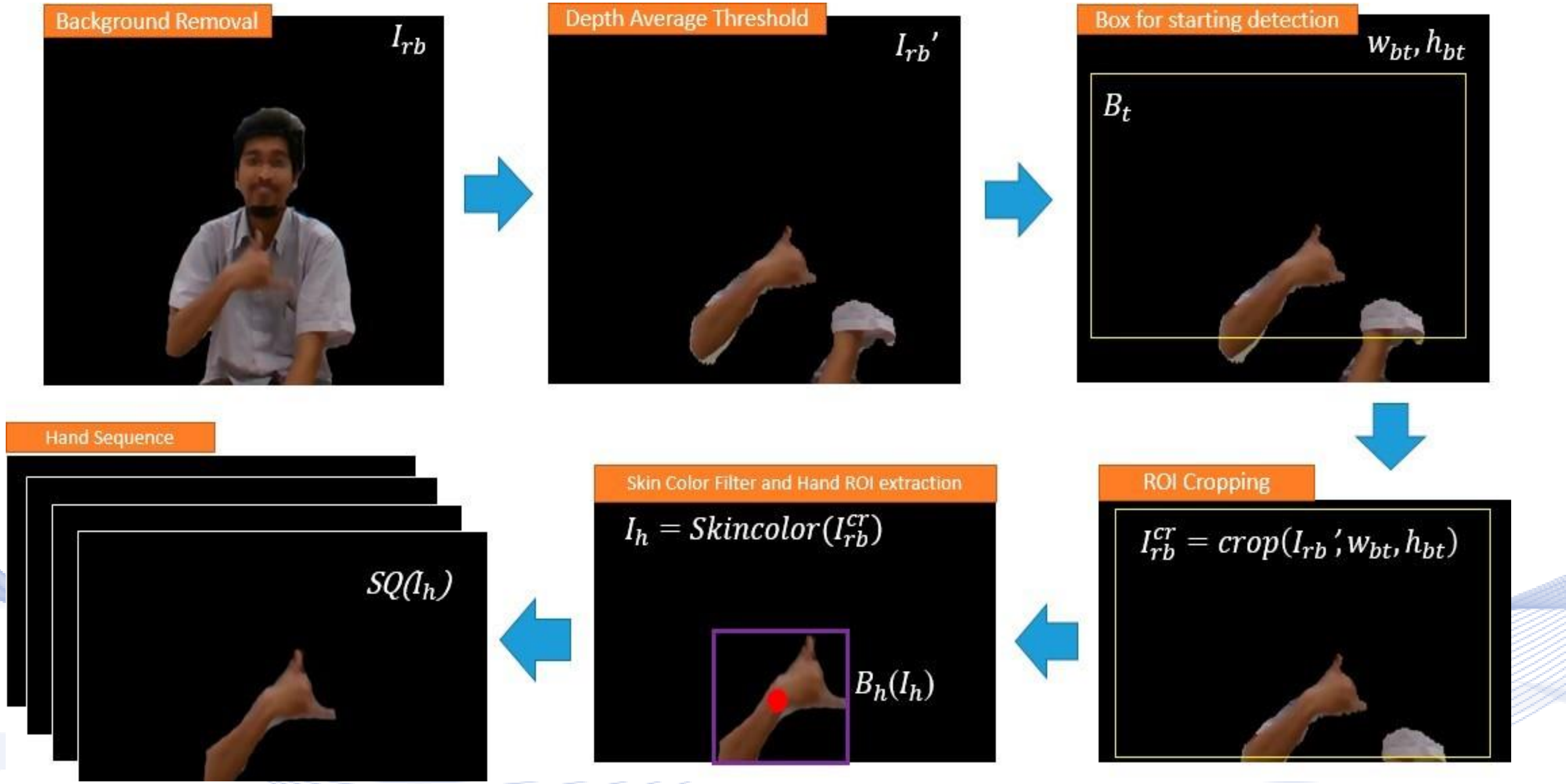
$$d_{av} = \frac{\sum_i^n I_{db}^i}{n}\ where\ I_{db}^i > \min and\ I_{db}^i < d_t$$

3D CNN+LSTM ConvLSTM



The architecture of the model [NOO2019].

3D CNN+LSTM ConvLSTM

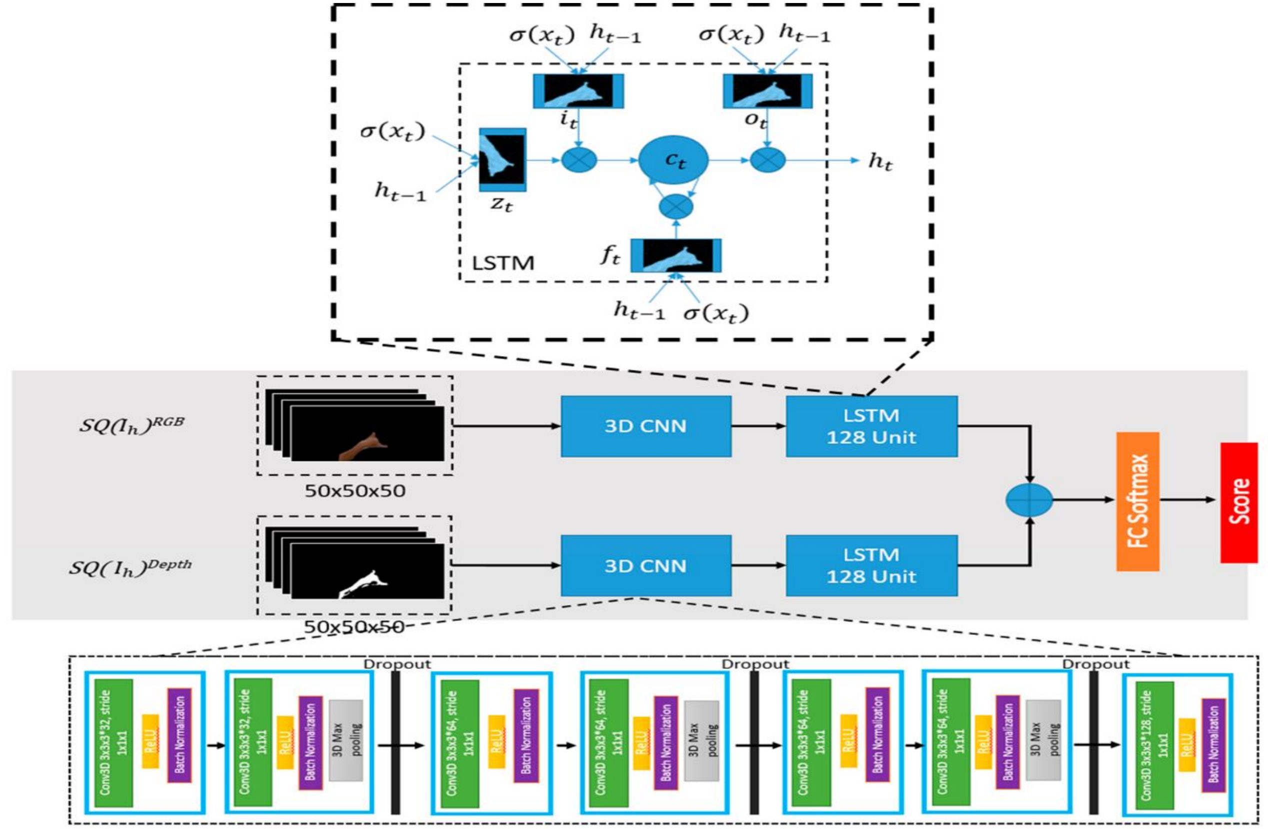

Background Removal
I_{rb}
Depth Average Threshold
I_{rb}'
Box for starting detection
w_{bt}, h_{bt}
B_t
ROI Cropping
$I_{rb}^{cr} = crop(I_{rb}; w_{bt}, h_{bt})$
Skin Color Filter and Hand ROI extraction
$I_h = Skincolor(I_{rb}^{cr})$
$B_h(I_h)$
Hand Sequence
$SQ(I_h)$

The preprocess of the extraction of the based on the threshold of the average depth [NOO2019].

3D CNN+LSTM ConvLSTM

- The architecture of the model consists of 3D-CNN layers, followed by one stack LSTM layer and, then there is a fully connected layer and in the end the softmax layer.
- The use of the Batch normalization make possible for the model to utilize learning rates much higher.
- The size of the kernel of each Conv3D layer is $3 \times 3 \times 3$, the stride and padding are sizes of $1 \times 1 \times 1$.
- After the Conv3D layer, there is a batch normalization layer, followed by a ReLU layer and a 3D Max Pooling layer with a pool size of $3 \times 3 \times 3$.
- There is an extraction of the features from the 3D-CNN, then there is an one stack of LSTM with 256 sizes of the unit.
- There is a value of 0.5 addition in every section in several dropout layers.

3D CNN+LSTM ConvLSTM

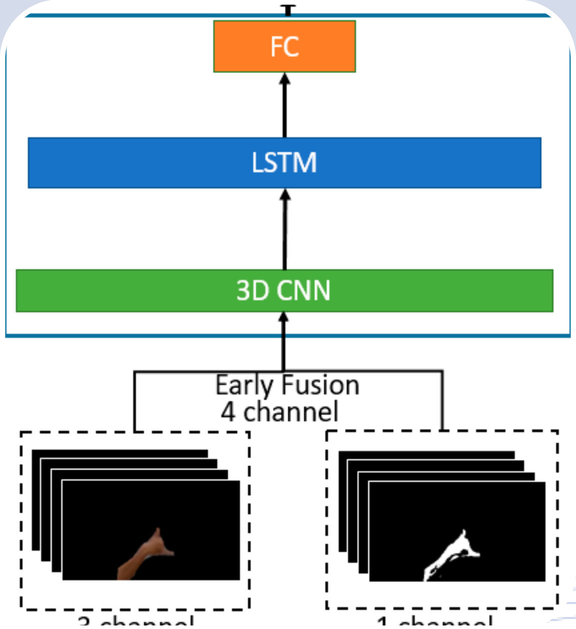


The proposed 3DCNN + LSTM architecture [NOO2019].

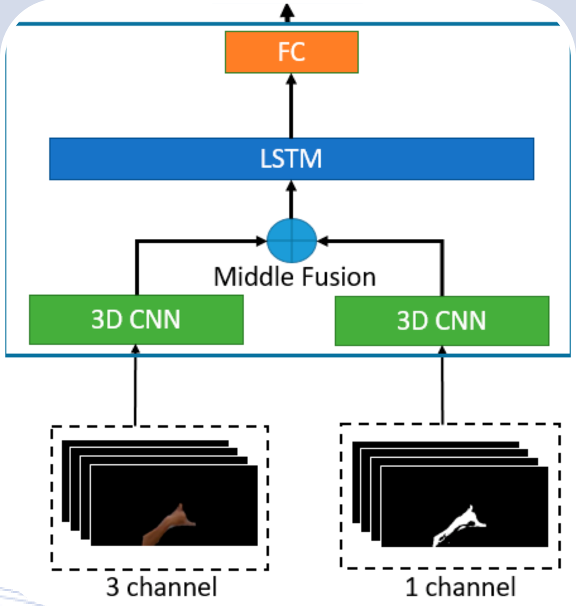
3D CNN+LSTM
ConvLSTM

- The depth and RGB fused to form the input data, and that may have a better result rather than the use of input data from one stream.
- The three kinds of multimodal types based on the level of their fusion:
  - Early fusion: 4 channels, 3 channels RGB+1 channel depth, fused before the input to the 3D CNN layers.
  - Middle fusion: the output of the separated 3D CNN layers, one from the RGB and the other from the Depth, are fused for the input to LSTM layer.
  - Late fusion: in the same way, the fusion takes place after the LSTM layer and before Fully Connected layer.

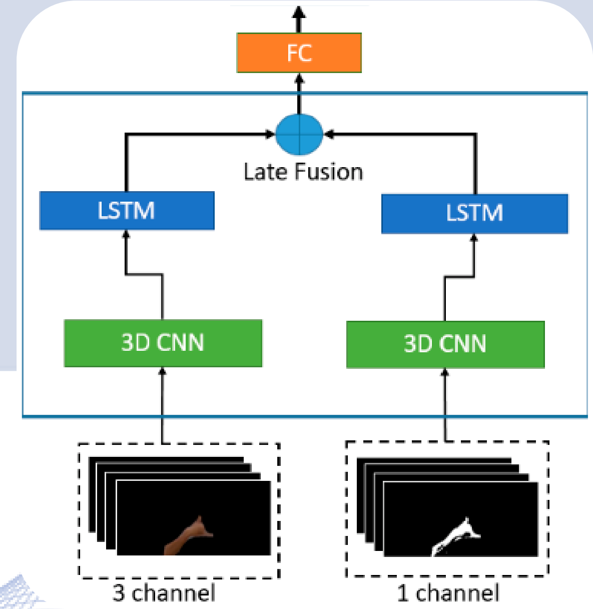
3D CNN+LSTM ConvLSTM



Early fusion+Depth+RGB hand only, accuracy rate 95,8%



Middle fusion+Depth+RGB hand only, accuracy rate 95,1%



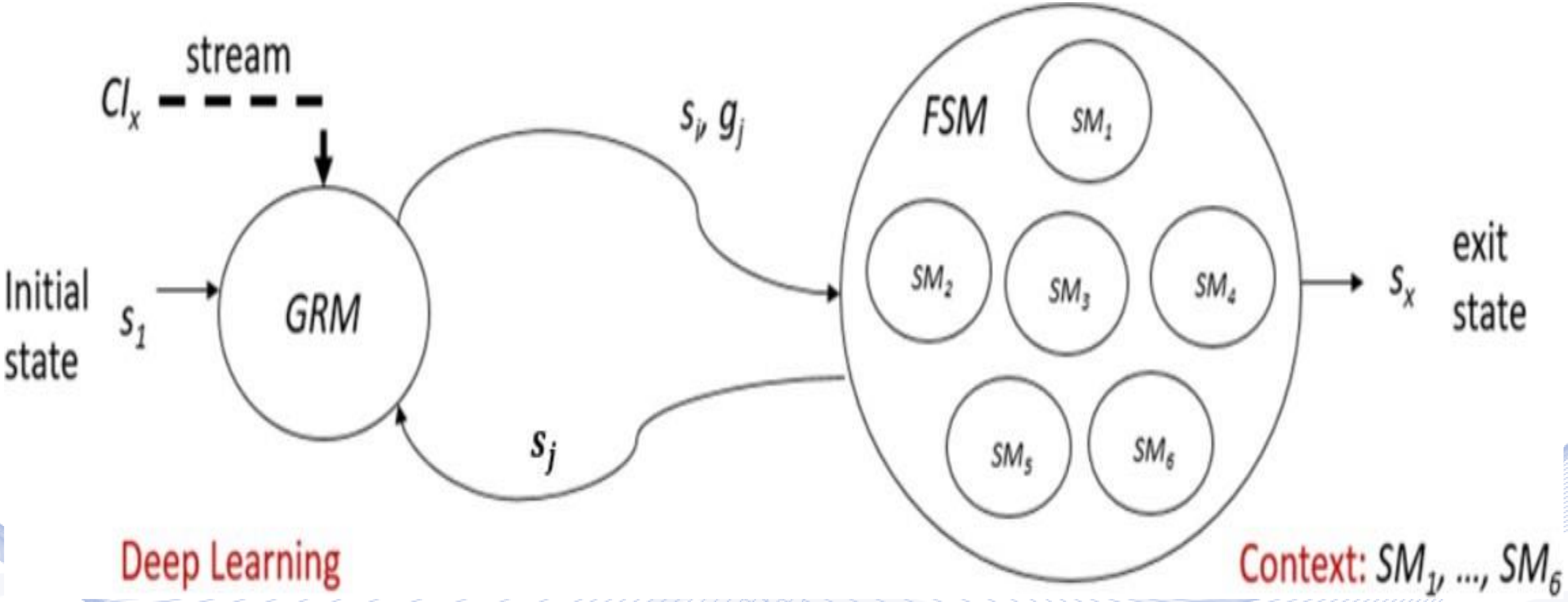
Late fusion+Depth+RGB hand only, accuracy rate 97,6%

[NOO2019]

Finite State Machine

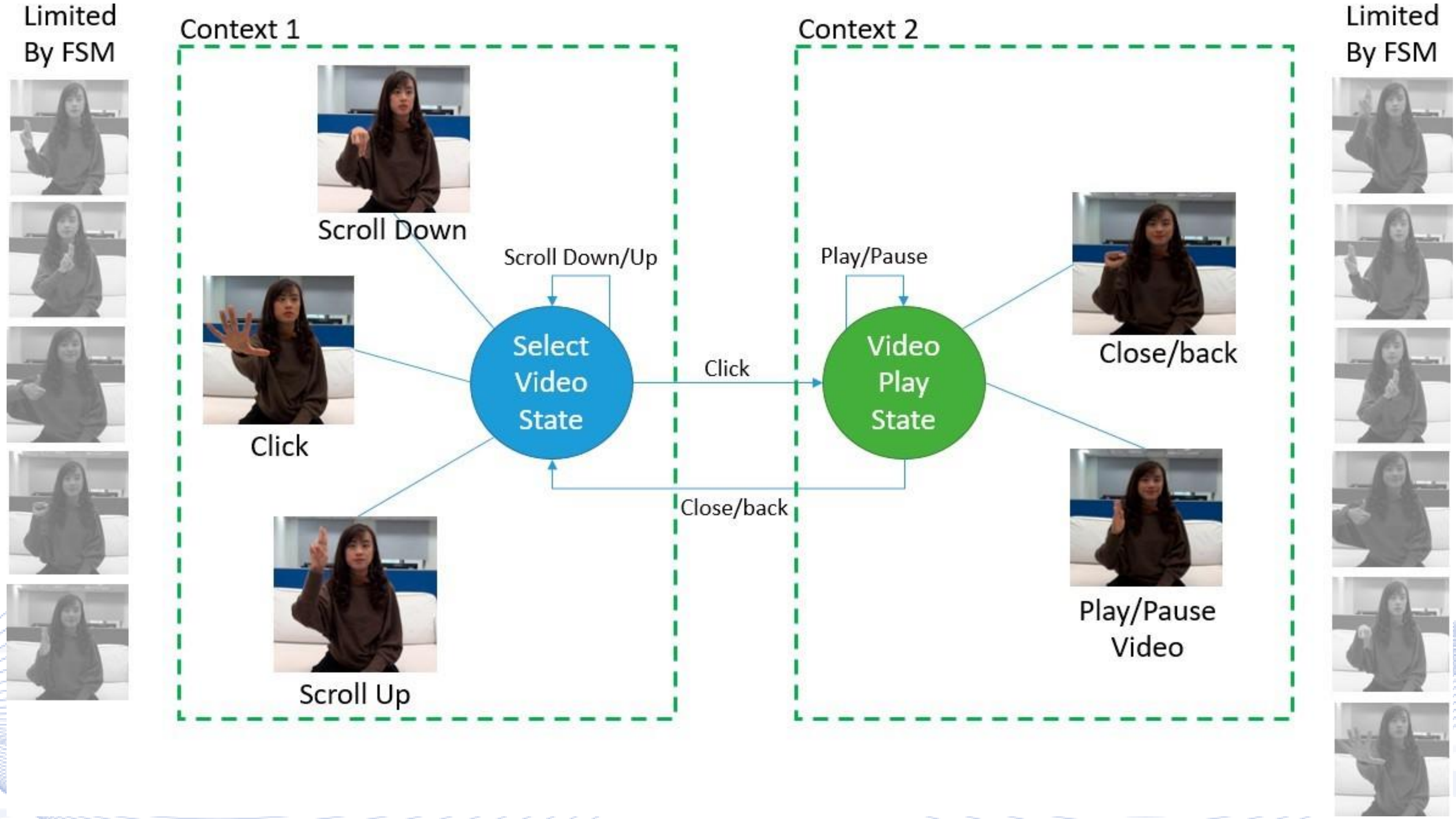
- For the increase of the rate of the recognition in the system there must be a recognition of smaller gesture class.
- In a Context-Aware recognition control system, the class for the recognition was bounded in every context.
- The Finite State Machine-FSM is attached with the Deep learning model and restrict the softmax decision probability with the manipulation of the weight in the last layer.
- The system in a current context or state interacts with the Finite State Machine to make a decision about which gesture not to be taken in account.
- The weights which were pre-defined to the last layer's node that join to the FSM ignored class are applied, thus just the correct gestures are accepted.

Finite State Machine



FSM model controller with GRM-Gesture Recognition Machine [NOO2019].

Finite State Machine



A two contexts example in FSM model
[NOO2019]

Deep learning for spatiotemporal systems

- According to [EGO2018] there are 4 ways of modelling for spatiotemporal systems:
  - The 2D ConvNets extract features of one frame. The classifiers are trained for the prediction of the labels of videos based on the frame features.
  - The 3D ConvNets can derive features of video clips. Afterwards, they accumulate the clip features into video descriptors.
  - The usage of recurrent neural networks-RNN to handle the temporal frames sequences is based on features of convolution.
  - Formatting a video in one or in multiple compact frames and classify it with a neural network.

Deep learning for spatiotemporal systems

- In [EGO2018], deep learning models are being compared using the dataset EgoGesture.
- The data splits into training set (60%) 1.239 videos, in validation set (20%) 411 videos and in testing set (20%) sets 431 videos.
- To classify, there is a segmentation of the video sequences into isolated samples of gestures based on the first and the last frames, which have annotations in advance.
- The aim of the learning is to anticipate the labels of the class for each sample of gesture.

Deep learning for spatiotemporal systems

- VGG16 is a 2D CNN with 13 convolutional and 3 FC layers.
- C3D is a 3D CNN with eight 3D convolutional layers, one 2D pooling layers, four 3D pooling layers and three fully-connected layers.
- C3D+hand mask: C3D method is for the segmentation for the hand, since the depth camera get rid of most of the background information and consider the depth frame as a hand mask.
- C3D+LSTM+RSTTM: a C3D augmented model with a recurrent spatiotemporal transform module (RSTTM).
- VGG16+LSTM a single-layer LSTM with 256 hidden units after the first fully-connected layer of VGG16 .
- IDMM+CaffeNet handles spatial and temporal data of a video into an image called improved depth motion map (IDMM)and in this way there can be classified by 2D ConvNets.

Gesture Classification Accuracy of the models with EGOGESTURE Dataset
[EGO2018]

METHOD	RGB	DEPTH	RGB-D
IDMM+CaffeNet	-	0,664	-
VGG16 softmax	0,572	0,579	0,612
VGG16 fc6	0,625	0,623	0,665
VGG16+LSTM softmax	0,673	0,690	0,725
VGG16+LSTM lstm7	0,747	0,777	0,814
C3D fc6, 8 frames	0,817	0,844	0,865
C3D softmax, 16 frames	0,851	0,868	0,887
C3D fc6, 16 frames	0,864	0,881	0,897
C3D+HandMask	-	-	0,872
C3D+LSTM+RSTTM	0,893	0,906	0,922

CS: cross subject setting, when the set of training and of testing are from different participants

Classification accuracy with or without CS

[EGO2018]

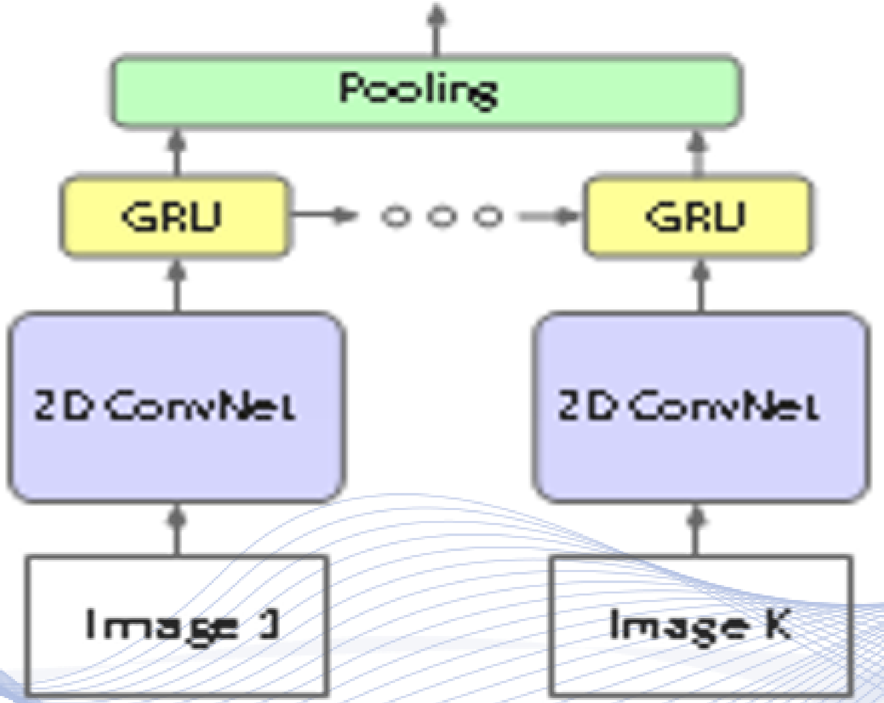
Method	Modality	Accuracy without CS	Accuracy with CS	Variance
VGG16 fc6	RGB	0,667	0,625	0,042
VGG16+LSTM lstm7	RGB	0,764	0,689	0,075
C3D fc6, 16 frames	RGB	0,892	0,864	0,028
VGG16 fc6	depth	0,647	0,623	0,024
VGG16+LSTM lstm7	depth	0,801	0,732	0,069
C3D fc6, 16 frames	depth	0,907	0,881	0,026
VGG16 fc6	RGB-D	0,697	0,665	0,32
VGG16+LSTM lstm7	RGB-D	0,826	0,753	0,73
C3D fc6, 16 frames	RGB-D	0,922	0,897	0,025

DNN Architecture Comparison

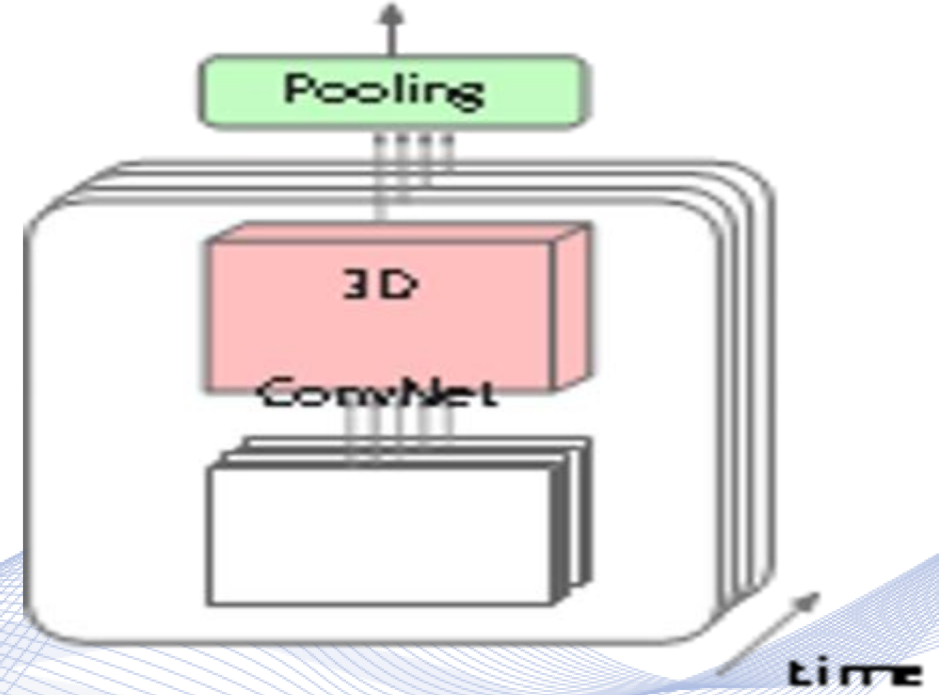
- The [DON2020] shows a study of comparing models which were trained with WLASL dataset
  - 2D CNN+RNN: RNN are used for the temporal relations and 2D CNN for the spatial features of the frames.
  - 3D CNN can be applied for both the spatial and temporal relationship between the frames.
  - Pose based models: use RNNs to interpret the sequences of the poses to analyze the movements.
  - Temporal Graph Convolution Networks-TGCN models the spatiotemporal dependencies of the pose sequence.

DNN Architecture Comparison

2D Conv. RNN



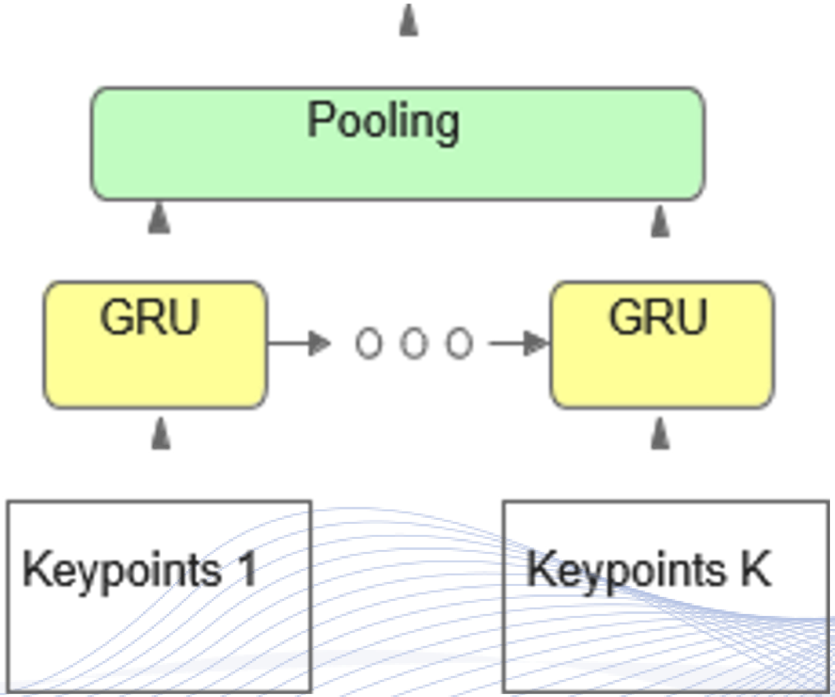
3D CNN



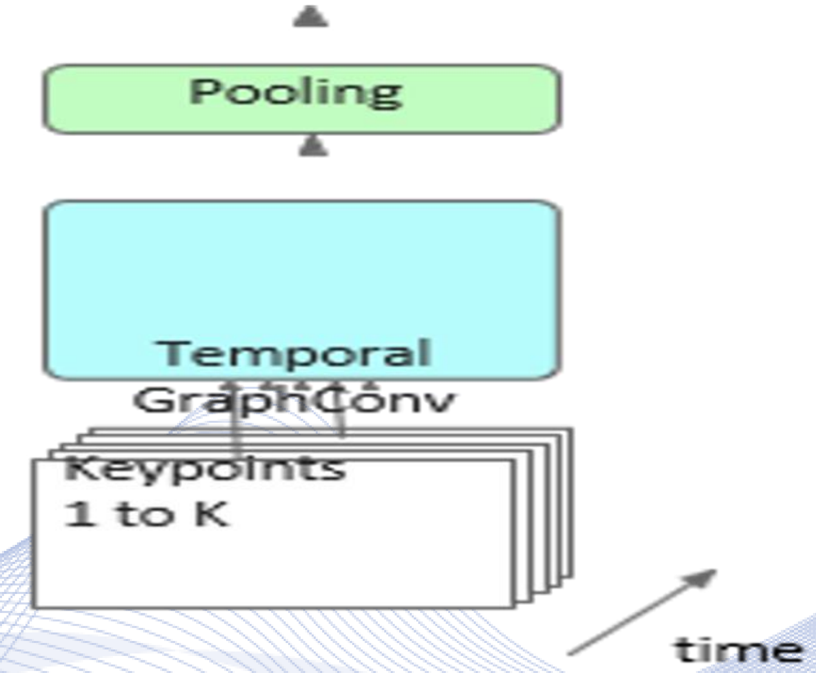
[DON2020]

DNN Architecture Comparison

Pose RNN



Pose TGCN



the keypoints are the joints of human bodies

[DON2020]

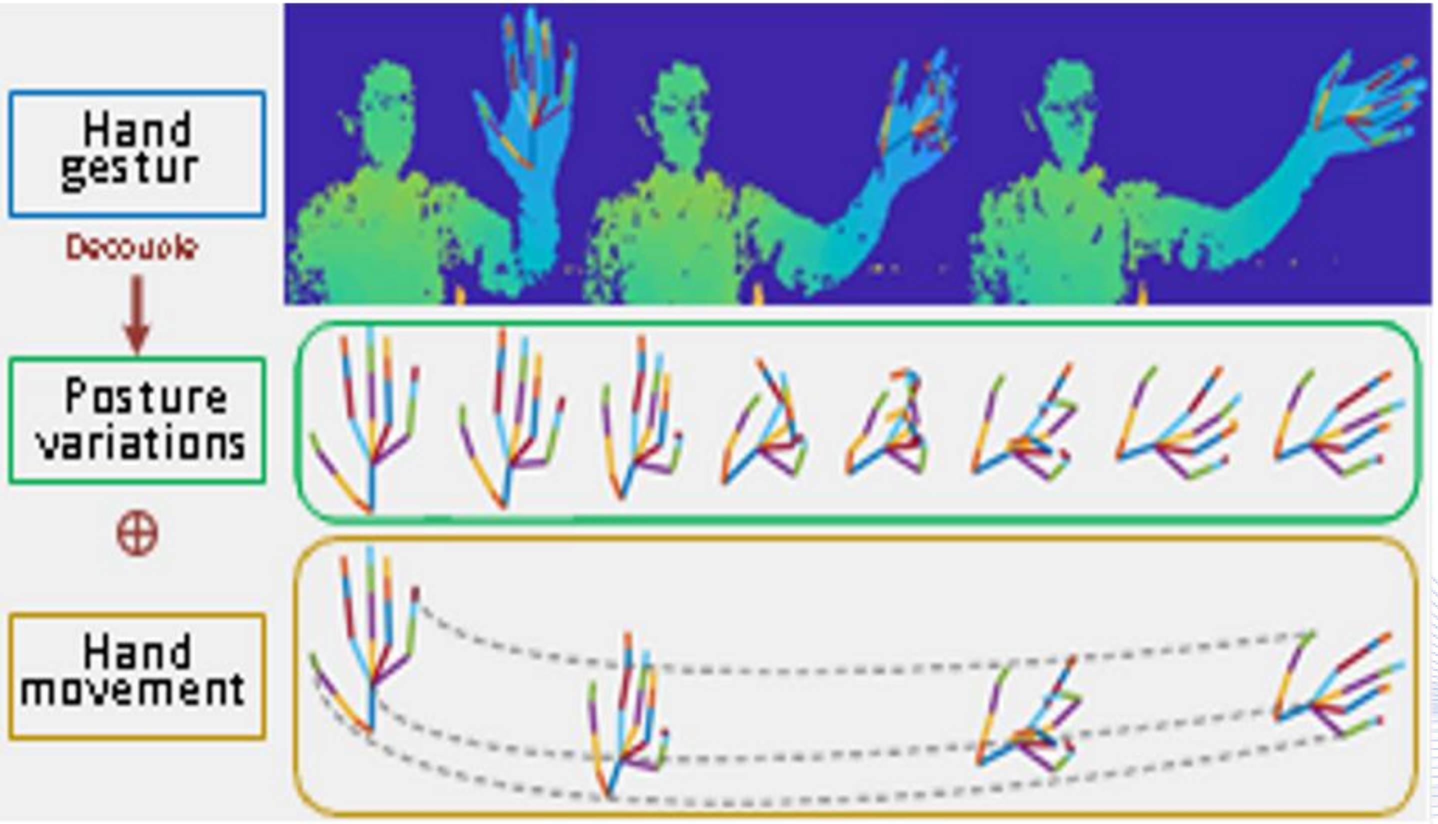
Gesture Recognition

- Introduction
- Gesture types
- Gesture Acquisition Devices
- Human-Machine Interaction
- Gesture Recognition Datasets
- Gesture Recognition Algorithms
- Deep Gesture Recognition
- **Skeleton-Based Gesture Recognition**
- Multimodal Gesture Recognition
- Egocentric Gesture Recognition
- Applications

Skeleton-based Gesture Recognition

- The [LIU2020] proposed architecture is built for a skeleton-based Gesture recognition and its approach is that the gesture is a sequence of complexly composite movements.
- The innovation of this architecture is that it is combined of two model: one applied on the hand posture variations and the other on the hand movements.
- HPEV (3D hand posture evolution volume) is the model applied on the posture variations and HMM-2D hand movement map model captures holistic movements.
- The HPEV integrates spatiotemporal information of hand postures with a 3D CNN, and on the other hand, the HMM uses a 2D CNN model to manipulate features of hand motions.

Skeleton-based Gesture Recognition



Hand gesture separated into variations of hand posture and hand movements **[LIU2020].**

Skeleton-based Gesture Recognition

- FRPV describes the movement of the finger and the positions of the 4 fingers and the thumb of each frame construct it.
- For frame t, the 4 relative positions are concatenated as a vector u_t :

$$u_t = \left(P_{I,t}, P_{M,t}, P_{R,t}, P_{L,t}\right) - (P_{0,t}, P_{0,t}, P_{0,t}, P_{0,t})$$

- where $p(0,t)$ is the coordinate of thumb of the $t-th$ frame, and $p(I,t), p(M,t), p(R,t)\ and\ p(L,t)$ are the coordinates of index fingertip, middle fingertip, ring fingertip and little fingertip at frame t respectively.
- The FRPV is the concatenation of all the u_t of each frame $[N = number\ of\ frames]$: $U_{FRPV} = (u_1, u_2, \dots, u_t, \dots, u_N)$

Artificial Intelligence & Information Analysis Lab

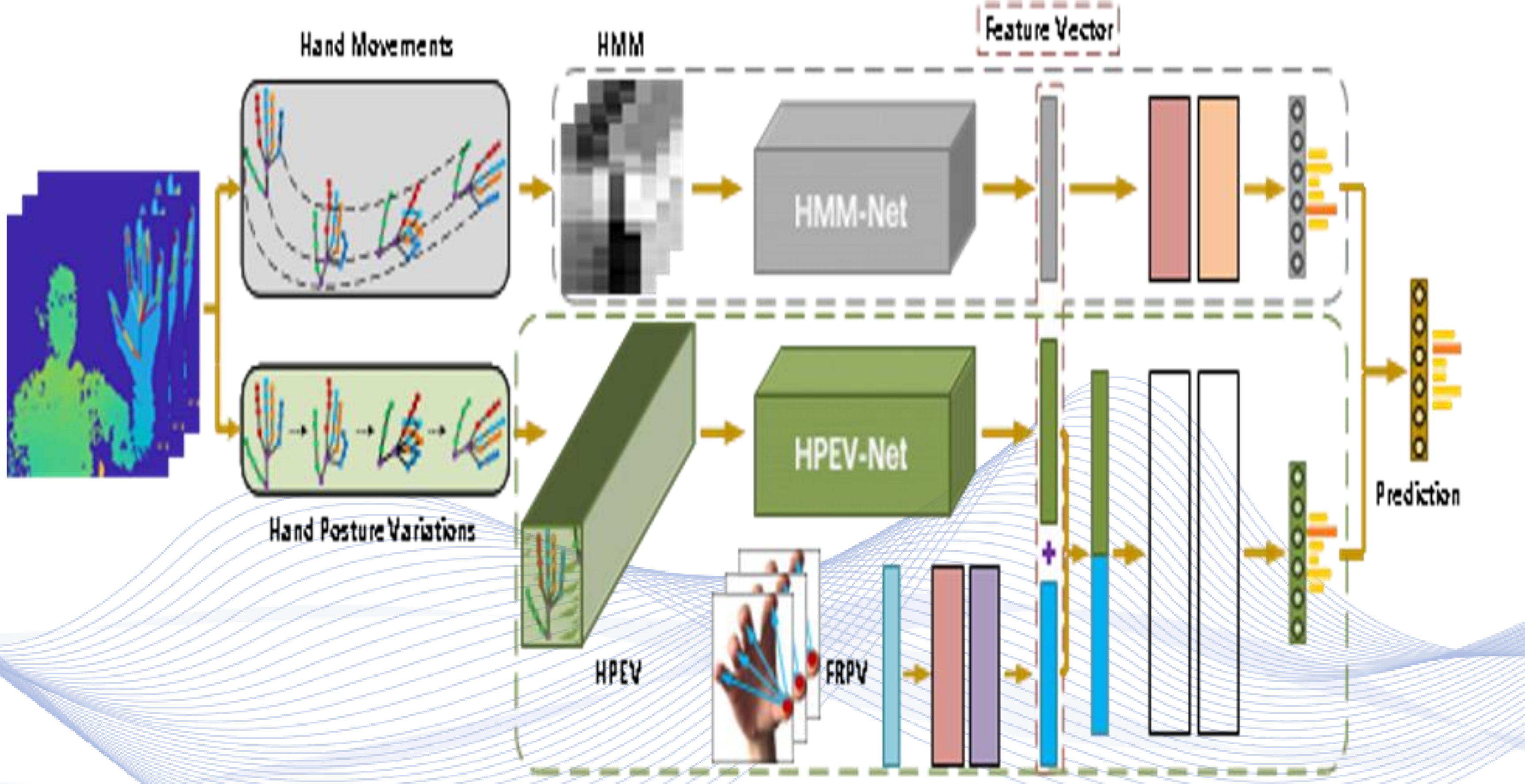
Skeleton-based Gesture Recognition

- The system consists two main streams and each of them extracts a vector of the features that has built for: HPEV-Net and HMM-Net.
- HPEV-Net is applied for the hand movement map and uses a 3D CNN for the low-level features with the size of the kernel 7x3x3 and afterwards there is a stack of four bottleneck modules [KAI2016] for the high level features.
- In every CNN layer the activation function is RELU, and the batch normalization is used. Also, the max pooling layers are 4x2x2.
- The Fingertip Relative Position Vector-FRPV, is applied to describe movements of the fingers, because the restrictions on the resolution make very difficult the encoding of these movements.

Skeleton-based Gesture Recognition

- The output of FRPV goes to the next FC layer with Batch Normalization and ReLU activation.
- In the same stream, the outputs of HPEV and FRPV, are concatenated and there is a classification of the hand gesture sequences with a softmax algorithm.
- On the other stream, Hand Movements Map (HMM), uses a 2D CNN for the for the motion of the hand.
- HMM-Net is based on the Hierarchical Co-occurrence Network (HCN) the [CHA2018] proposed network, to extract features.
- In the same way like the HPEV-Net, there is a stack of four bottleneck modules.
- The output channels of the four bottleneck modules are 128, 128, 256 and 512.
- The output of the last bottleneck after global average pooling ends into a feature vector.

Skeleton-based Gesture Recognition

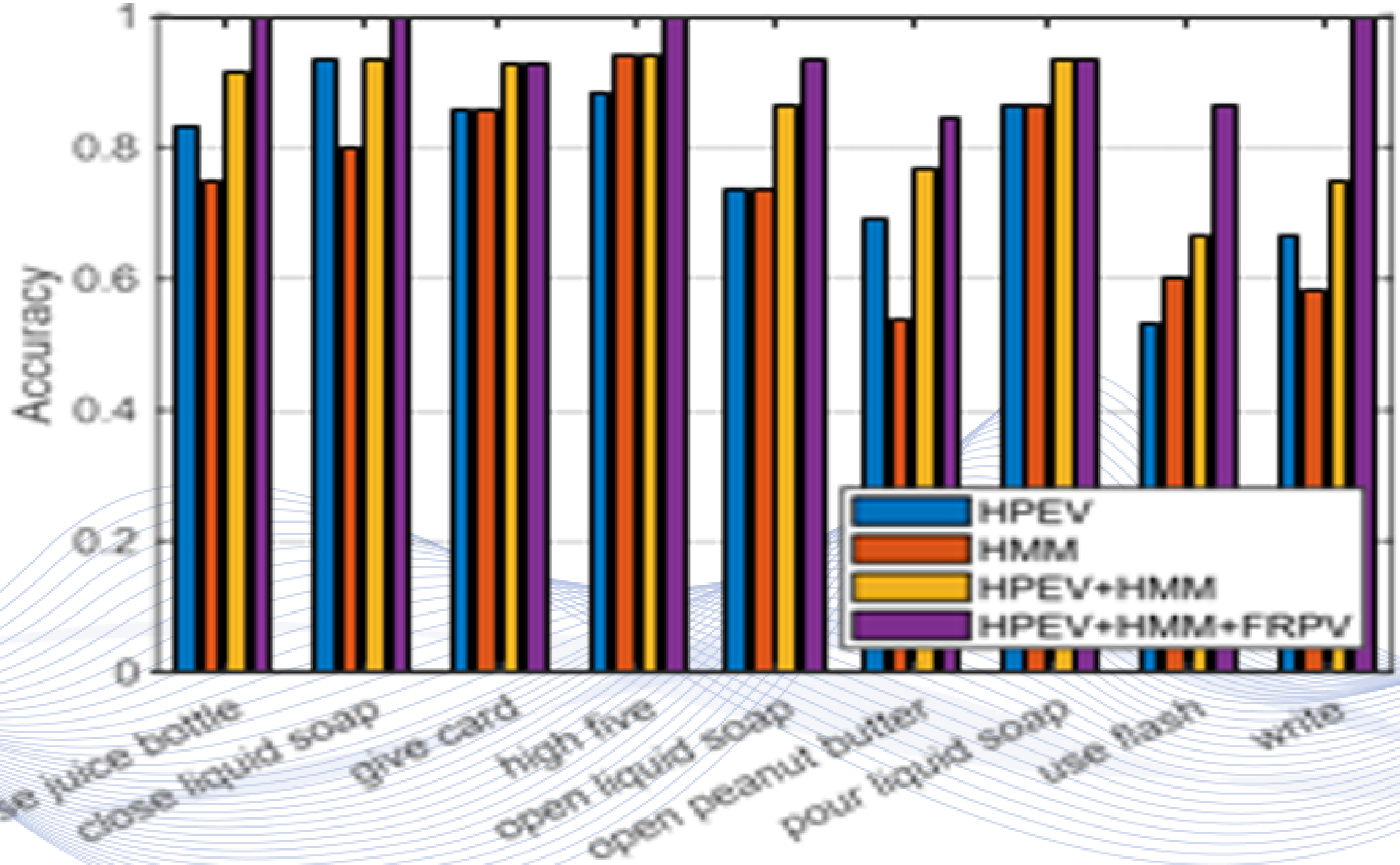


The two Neural Networks HPEV-Net+HMM-Net **[LIU2020].**

Skeleton-based Gesture Recognition

- SHREC'17 Track: This dataset [QUE2017] contains 14 gestures. They are performed twice: with one finger and with the whole hand. It includes 2800 sequences, 1960 for the training set and 840 for testing set.
- DHG-14/28: The dataset [HAZ2016] comprises 14 gestures with 2800 sequences. The DHG-14/28 and the SHREC'17 Track datasets have the same hand joints and the same method of data collection.
- FPHA: The last dataset [GUI2018] provides dynamic hand sequences. It includes 1175 action had sequences, with 45 categories handling 26 different objects in 3 scenarios. It has one less hand joint from the SHREC'17 Track dataset. The training set (600 sequences) and testing set (575 sequences) have almost the same percentage of data.

Skeleton-based Gesture Recognition



Gesture recognition accuracy rates with the use of different combinations of input on FPHA dataset **[LIU2020].**

Skeleton-based Gesture Recognition

Method	SHREK		FPHA
	14G	28G	
HPEV	0,734	0,714	0,770
HMM	0,927	0,866	0,677
FRPV	0,628	0,588	0,664
HPEV+HMM	0,944	0,902	0,829
HPEV+HMM+FRPV	0,948	0,922	0,909

Gesture Recognition accuracy rates for different input combinations on SHREC'17 Track and FPHA dataset.
14G:14 gestures, 28G:28 gestures
[LIU2020]

Skeleton-based Gesture Recognition

METHOD	ACCURACY 14G	ACCURACY 28G
HON4D [OMA2013]	0,785	0,740
SoCJ+Direction+Rotation [SME2017]	0,869	0,842
SoCJ+HoHD+HoWR [HAZ2016]	0,882	0,819
Two-stream 3D CNN [JUA2018]	0,834	0,774
Res-TCN [JIN2018]	0,911	0,873
STA-Res-TCN [JIN2018]	0,936	0,907
ST-GCN [YAN2018]	0,927	0,877
ST-TS-HGR-NET [XUA2019]	0,942	0,894
DG-STA [YUX2019]	0,944	0,907
HPEV+HMM+FRPV	0,948	0,922

Gesture recognition comparison of some of the latest proposed models with SHREC’17 dataset
14G:14 gestures, 28G:28 gestures **[LIU2020].**

Temporal Graph Convolution Networks

- In the Temporal Graph Convolution Networks (TGCN), the input sequence of poses $X_{1:N} = [x_1, x_2, \ldots, x_N]$ where $N = frames$ and $X_i \in \mathbb{R}^K$ are the 2D keypoints in K dimensions.
- encodes the body movements as a holistic representation of the trajectories of body keypoints.
- In this way the dependencies among the joints of the human body are represented in a graph network.
- A residual graph convolutional block stacks two graph convolutional layers.

Temporal Graph Convolution Networks

- a human body is represented as graph that is a fully-connected with K vertices and the edges in the graph as a weighted adjacency matrix:

$$A \in \mathbb{R}^{K*K}$$

- In a deep graph convolutional network, the $n-th$ graph layer is a function G_n that take as input features a matrix: $H_n \in \mathbb{R}^{K*F}$

  F is the feature dimension output by the previous layer.

- The set of trainable weights: $W_n \in \mathbb{R}^{F*F'}$
- The computation of a graph convolutional layer: $H_{n+1} = G_n(H_n) = \sigma(A_n H_n W_n)$

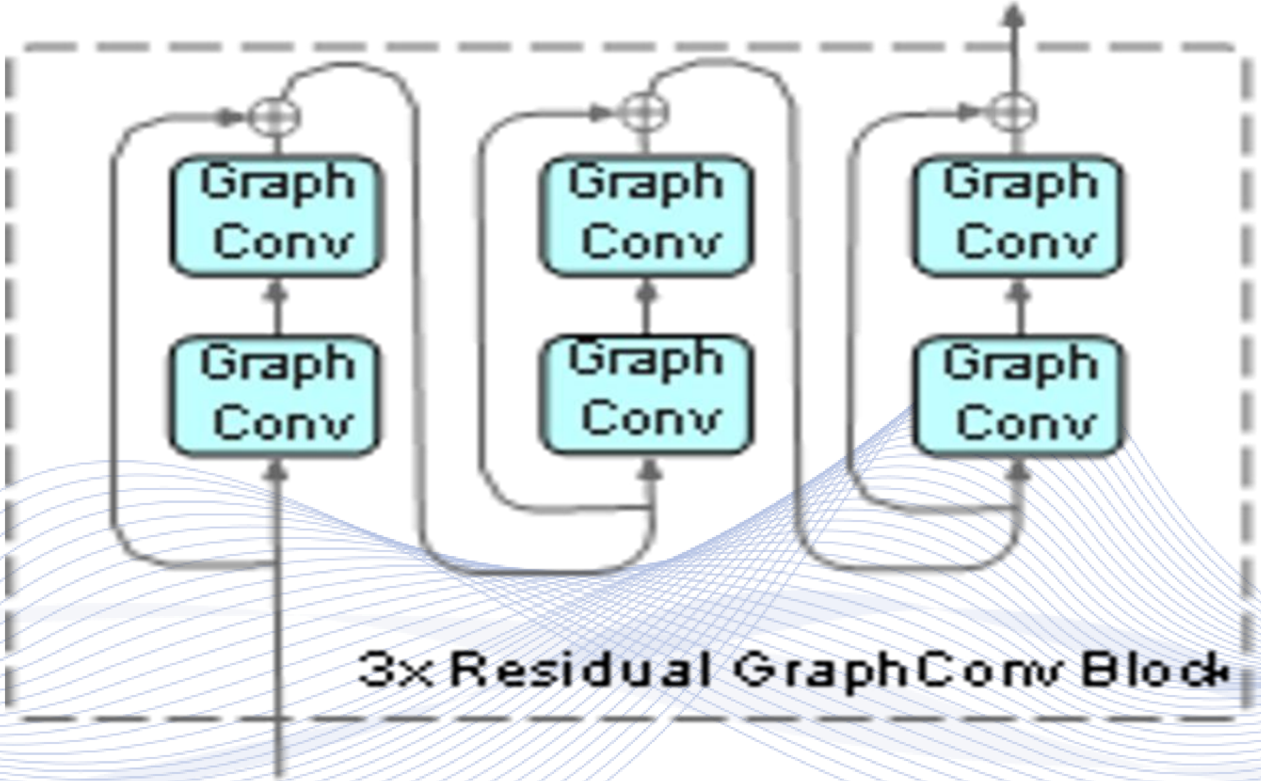
  where A_n is a trainable adjacency matrix for $n-th$ layer and $\sigma(\cdot)$ implies the activation function $\tanh(\cdot)$

Temporal Graph Convolution Networks

Method	WLASL 100	WLASL 300	WLASL 1000	WLASL 2000
Pose GRU	0,856	0,760	0,701	0,613
Pose TGCN	0,876	0,796	0,719	0,622
VGG+GRU	0,639	0,610	0,493	0,325
3D CNN	0,899	0,869	0,843	0,663

top-10 accuracy rates by each model on WLASL subsets with different number of glosses
[DON2020]

Temporal Graph Convolution Networks



Residual Graph Convolution Block **[DON2020].**

Temporal Graph Convolution Networks

TRAINING SET/TESTING SET	WLASL100		WLASL300		WLASL1000		WLASL2000	
	3D CNN	TGCN	3D CNN	TGCN	3D CNN	TGCN	3D CNN	TGCN
WLASL100	0,899	0,876	-	-	-	-	-	-
WLASL300	0,883	0,814	0,869	0,796	-	-	-	-
WLASL1000	0,852	0,775	0,862	0,742	0,843	0,719	-	-
WLASL2000	0,720	0,678	0,711	0,654	0,673	0,645	0,663	0,622

Top-10 accuracy rates of 3D CNN and Pose-TGCN with different training set (rows) and testing set (columns) on WLASL subsets **[DON2020]**

Gesture Recognition

- Introduction
- Gesture types
- Gesture Acquisition Devices
- Human-Machine Interaction
- Gesture Recognition Datasets
- Gesture Recognition Algorithms
- Deep Gesture Recognition
- Skeleton-Based Gesture Recognition
- **Multimodal Gesture Recognition**
- Egocentric Gesture Recognition
- Applications

Deep Multimodal Multi-stream Activity Recognition

[SON2016] Propose of a multimodal multi-stream DL framework for egocentric activity recognition, using video & sensor data.

1) Experiment & Extend a multi-stream CNN to learn spatial and temporal features from egocentric videos.
2) Proposal of a multi-stream LSTM architecture to learn the features from multiple sensor streams (accelerometer, gyroscope, etc.).
3) Propose of using a two-level fusion technique and experiment different pooling techniques to compute final prediction results.

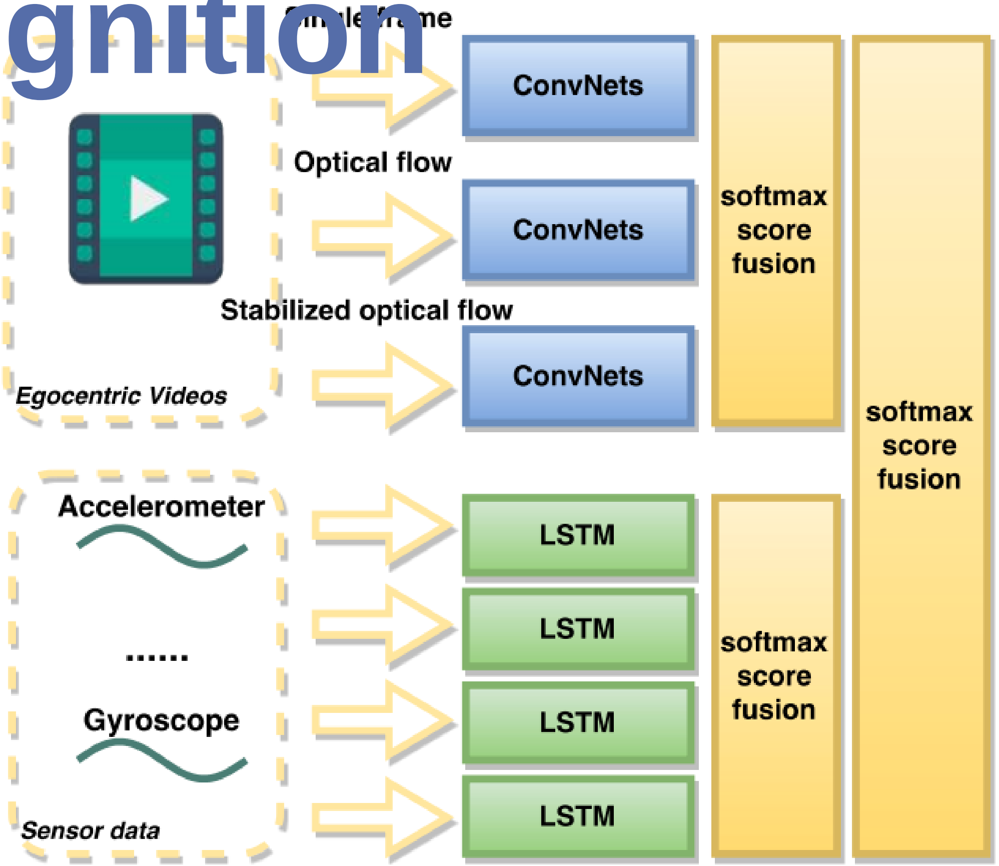
Deep Multimodal Multi-stream Activity Recognition

For video data → extension of 2-stream ConvNets to a 3-stream ConvNets for spatial, optical flow & stabilized optical flow data.

For sensor data → multi-stream LSTM framework to analyze multiple-axis sensor measurements: accelerometer, gyroscope, magnetic field & rotation.

To fuse results (spatial, optical flow and stabilized optical flow for video data, various sensor measurements)→ average pooling & maximum pooling & a two-level fusion approach

Deep Multimodal Multi-stream Activity Recognition



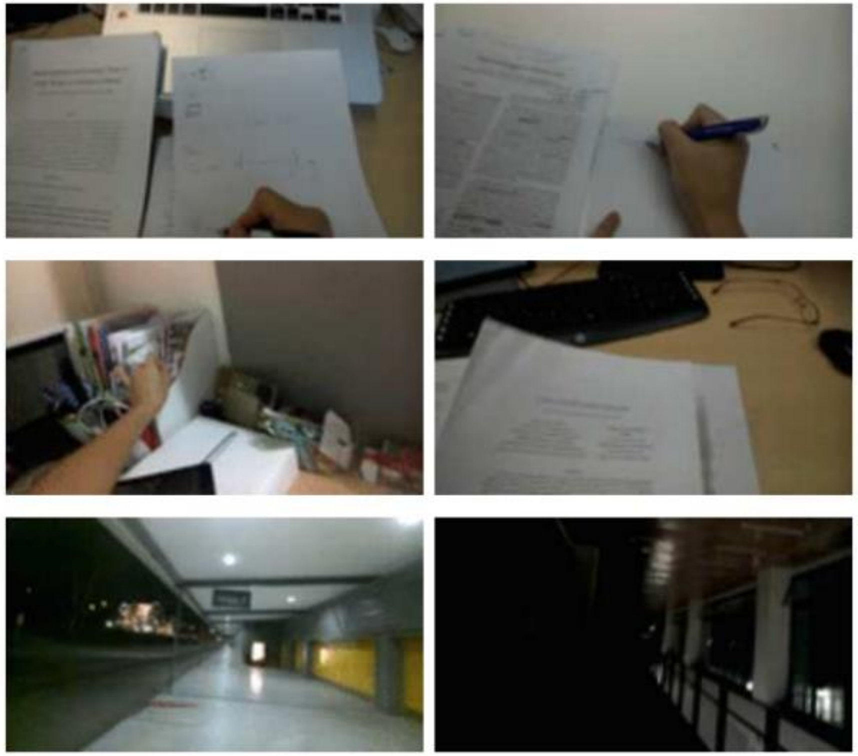
[SON2016] To fuse the three streams (spatial, optical flow & stabilized optical flow) → average pooling & maximum pooling to predict labels of activities.

Deep Multimodal Multi-stream Activity Recognition

<u>Types of sensor data:</u>

accelerometer, gravity, gyroscope, linear acceleration, magnetic field & rotation vector.

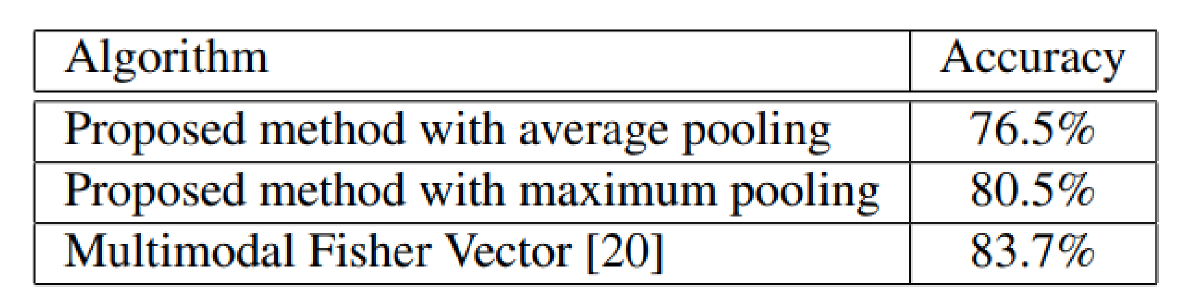
15 seconds duration & sampling rate of 10



[SON2016] Sample frames of video data from Multimodal Egocentric Activity Dataset

Deep Multimodal Multi-stream Activity Recognition

Basic RNN takes in sequential input and for each data in the sequence, it calculates hidden states which take part in predicting the next data in the sequence → performs prediction or classification for a certain data point by finding the temporal relationship from the previous data point in the sequence.



Algorithm	Accuracy
Proposed method with average pooling	76.5%
Proposed method with maximum pooling	80.5%
Multimodal Fisher Vector [20]	83.7%

Accuracy results

Gesture Recognition

- Introduction
- Gesture types
- Gesture Acquisition Devices
- Human-Machine Interaction
- Gesture Recognition Datasets
- Gesture Recognition Algorithms
- Deep Gesture Recognition
- Skeleton-Based Gesture Recognition
- Multimodal Gesture Recognition
- **Egocentric Gesture Recognition**
- Applications

Deep Ecocentric Activity Recognition

[CAO2017] Challenge: arises from the global camera motion caused by the spontaneous head movement of the device wearer

→ address problem by a recurrent 3D convolutional neural network for end-to-end learning

Deep Ecocentric Activity Recognition

Design end-to-end learnable egocentric gesture recognition model without detecting hand and estimating head motion explicitly & independently

1) **Egocentric motion:** since camera is worn on user's head, camera motion can be significant due to the head movement
2) **Hands in close range**: due to short distance from camera to hands and the narrow field-of-view of the egocentric camera, hands are prominent in the frame but meanwhile could be partly or even totally out of the field-of-view.

Reference

Deep Ecocentric Activity Recognition

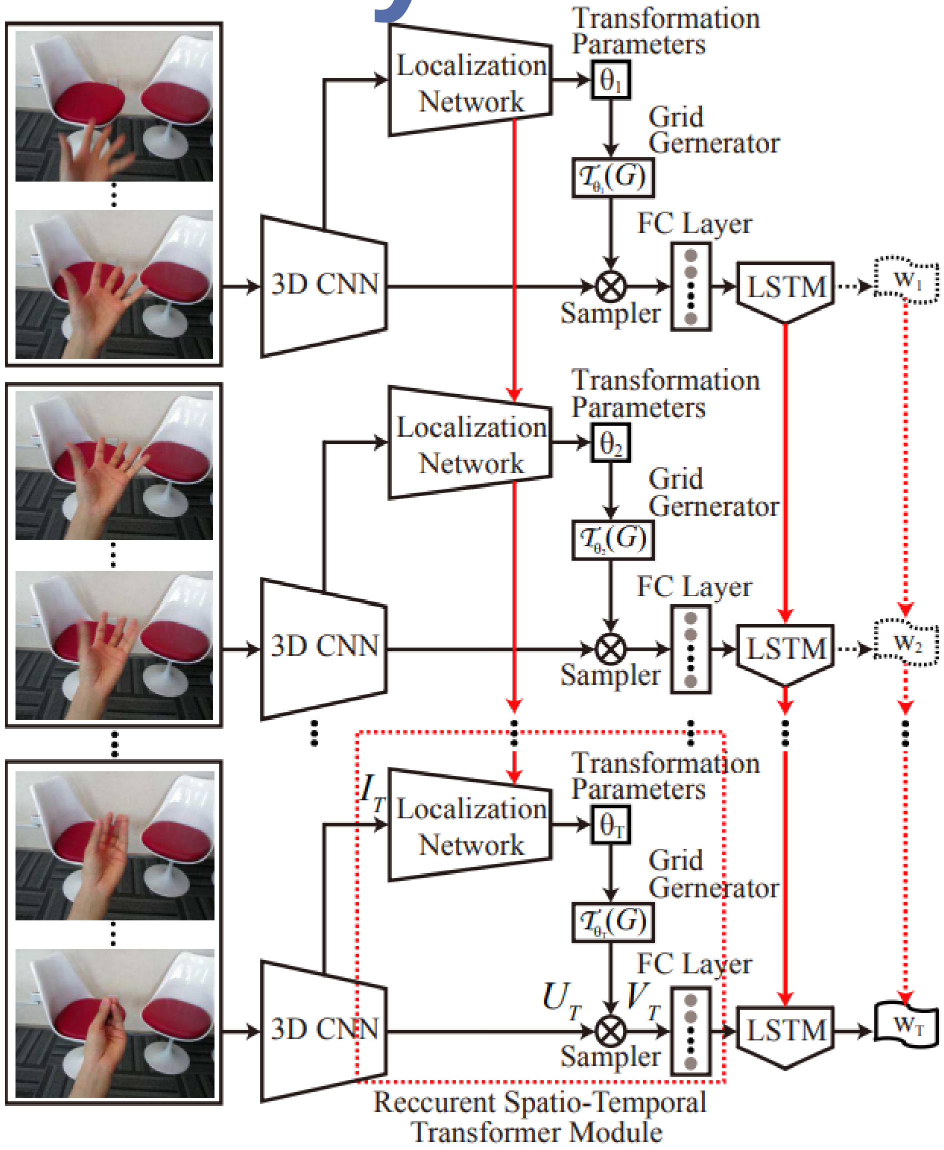
→ 3D CNN + RNN to process video sequences

→ Proposal of a spatiotemporal transformer module (STTM) to transform 3D feature maps to a canonical view in both spatial and temporal dimensions.

- Use of homography transformations to deal with head motion
- Estimate the transformation parameters at current time based on the previous ones on video sequences by introducing recurrent connections

Reference

Deep Ecocentric Activity Recognition



Reference

[CAO2017]

Deep Ecocentric Activity Recognition

Propose a framework of recurrent 3D CNN in an end-to-end learning paradigm, which can not only capture short-term spatiotemporal features, but also model long-term dependencies

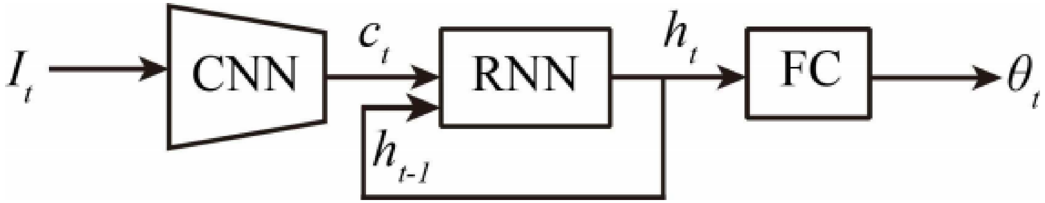
There are 3 parts in a recurrent spatiotemporal transformer module: a localization network, a grid generator and a sampler.

Reference

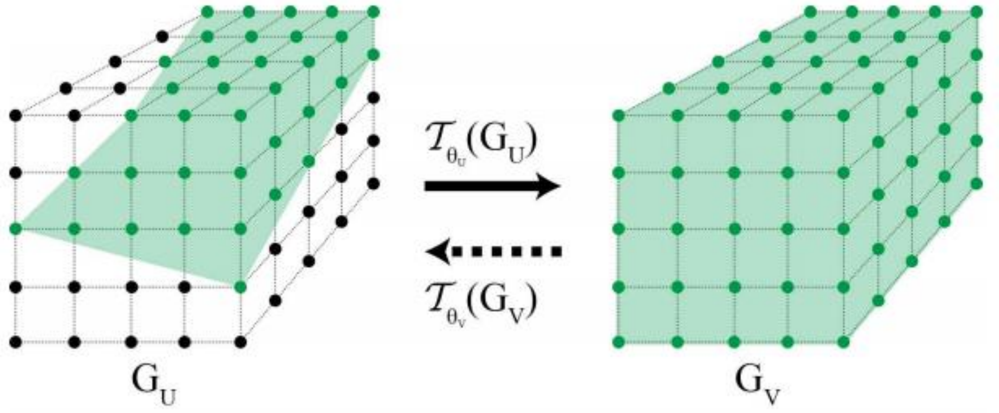
[CAO2017]

Example of the 3D grids before and after a transformation of homography

Deep Ecocentric Activity Recognition



Localization network



Example of the 3D grids before and after a transformation of homography

[CAO2017]

Reference

Gesture Recognition

- Introduction
- Gesture types
- Gesture Acquisition Devices
- Human-Machine Interaction
- Gesture Recognition Datasets
- Gesture Recognition Algorithms
- Deep Gesture Recognition
- Skeleton-Based Gesture Recognition
- Multimodal Gesture Recognition
- Egocentric Gesture Recognition
- **Applications**

Applications

- Sign Language.
- Navigation or/and the manipulation in VR environment.
- Distance learning.
- Understanding of the human behavior in the interaction of a human with a computer.
- The distance controlling of devices and of machines.

Applications

- ***Sign Language***
  - Developing aid for the deaf
- Navigating or/and manipulating in virtual environment
- Distance learning
- Understand human behavior in human-computer interaction
- Monitoring machines from a distance
  - automobile drivers' alertness levels
  - Doctors monitor patient states

Applications

Sign Language Recognition

- Manual gesture features either combined or individually posed from hands, face or other body part.
- Each region defines each own lexicon and linguistics.
- Three main components:
  - ***Finger spelling***: spelling words letter by letter
  - ***Manual features***: gestures made with hands that include motion and express meaningful things
  - ***Non-manual features***: facial expressions, hand and arm moves or body posture

Applications

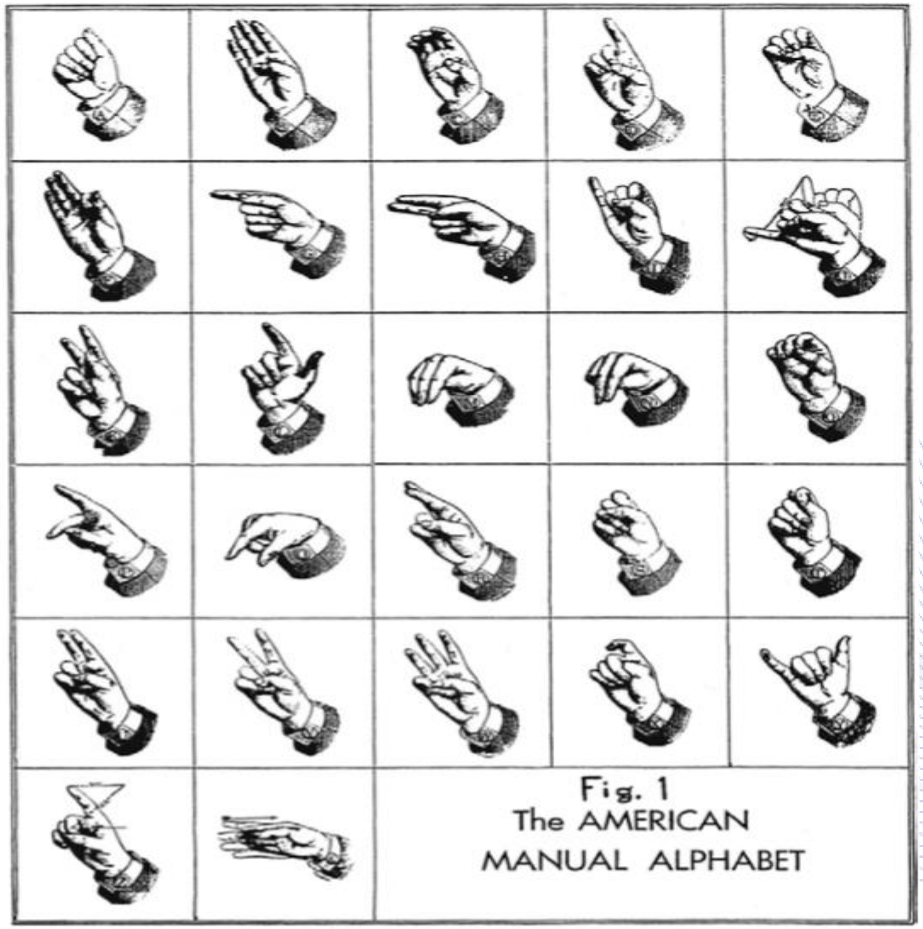
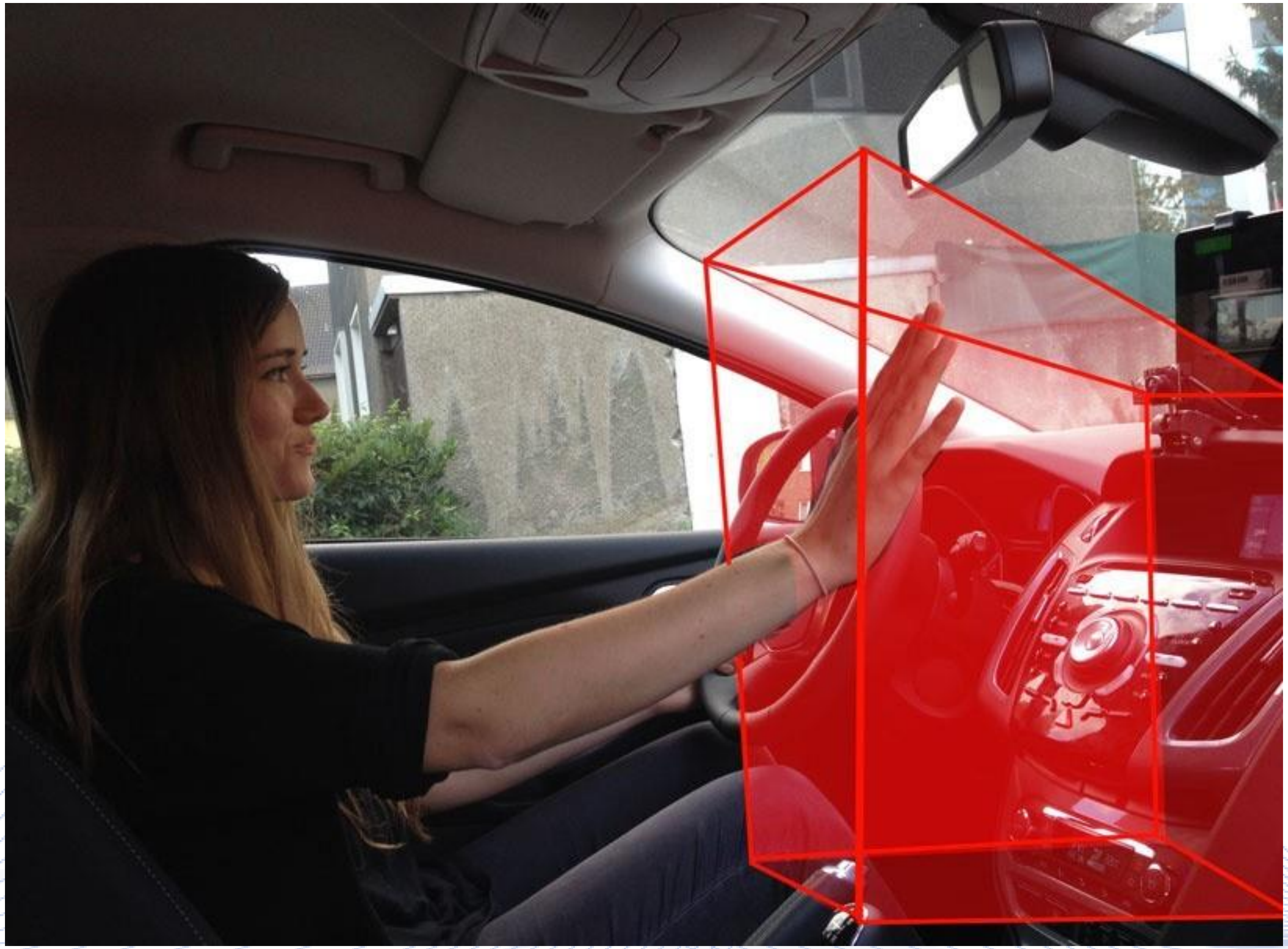


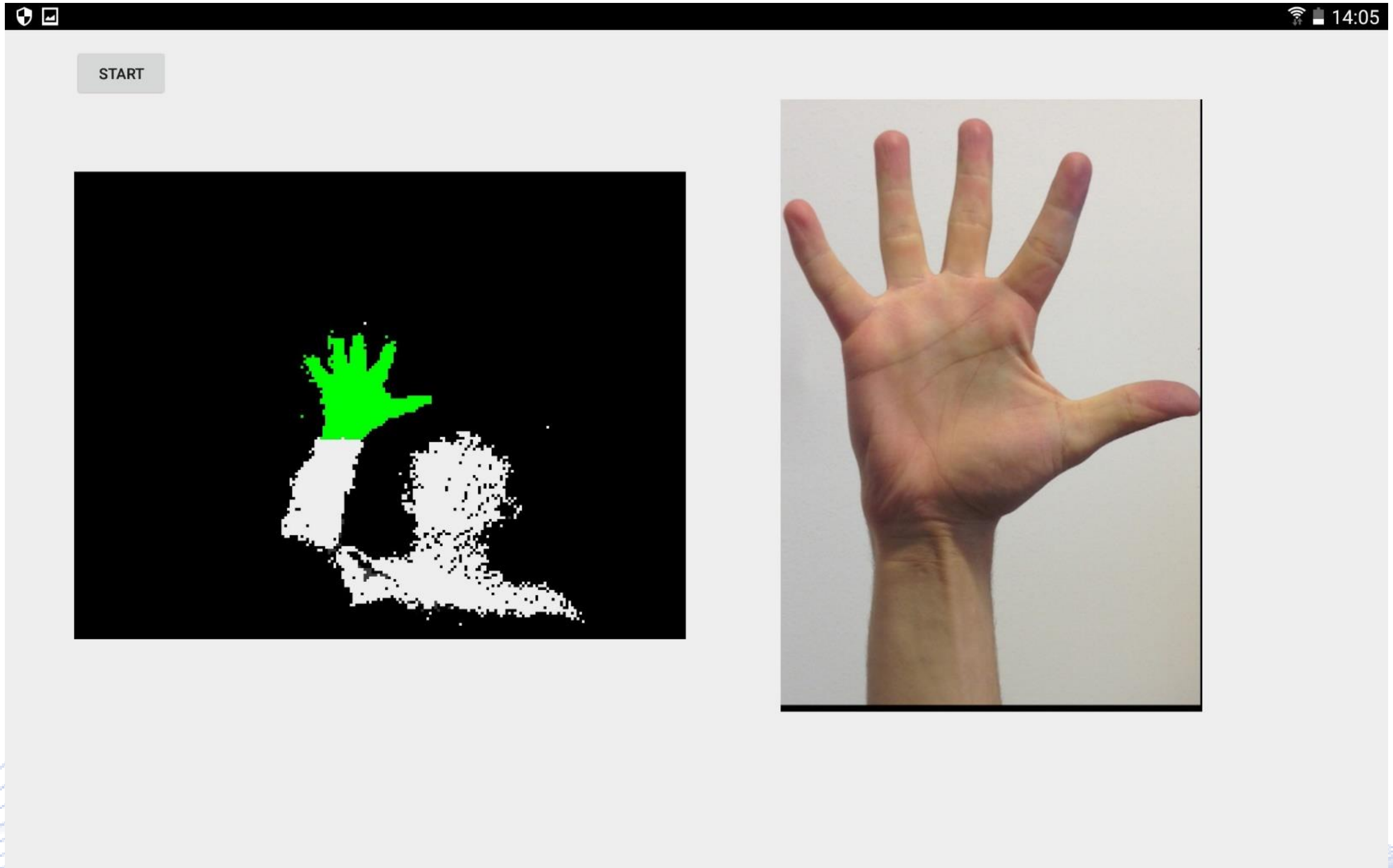
Fig. 1
The AMERICAN
MANUAL ALPHABET

Human-Machine Interaction

- The interaction between a human and a device can be performed with gesture recognition.
- A driver–vehicle interaction is presented in [ZEN2018]. Camboard Nano ***Time-of-Flight*** (***ToF***) sensor is used for the getting depth data.
- ToF sensors are able to record as depth data the hand gestures in real time.
- This sensor provides depth images. Their resolution is 165×120 pixels at a frame rate of 90 fps.
- The control system is applied on a computer-tablet which is situated on the center of the console of the vehicle.



Performing hand gesture detected in the range of the sensor of time-of-flight-ToF (area of detection in red) [ZEN2018].



Preprocessing data with PCA

the green area (left) remain to classify the class of the correct hand posture (right)

[ZEN2018]

Lane change with gesture control [ZEN2018].

Human-Robot Interaction

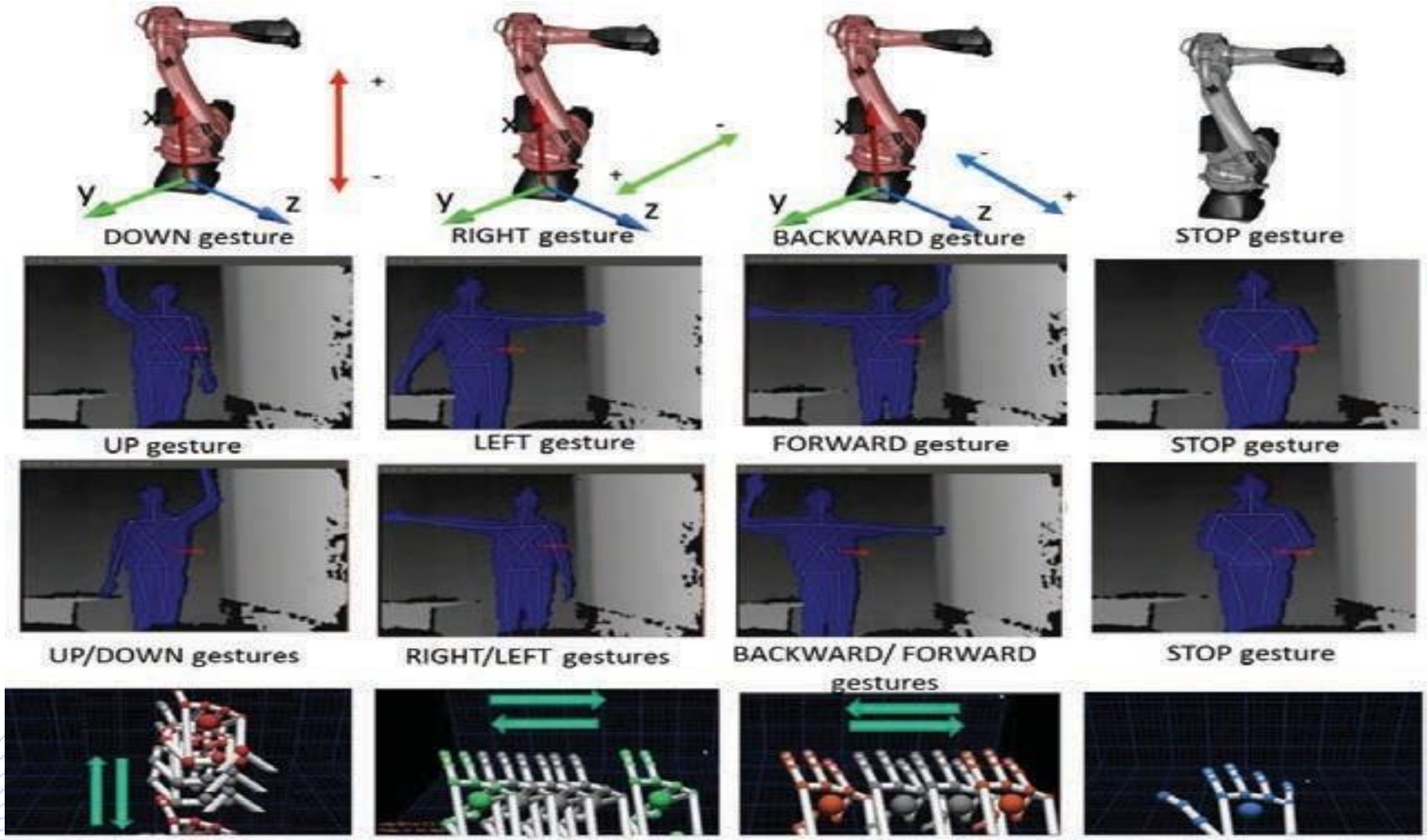
Gesture-controlled robots are used in various fields:

- Industry (car factories).
- Medicine (remote robotic surgery).
- In military (armed robotic vehicles).
- In space (articulated hands).

Robot programming using gestures

Robot programming using body and hand gestures [TSA2016].

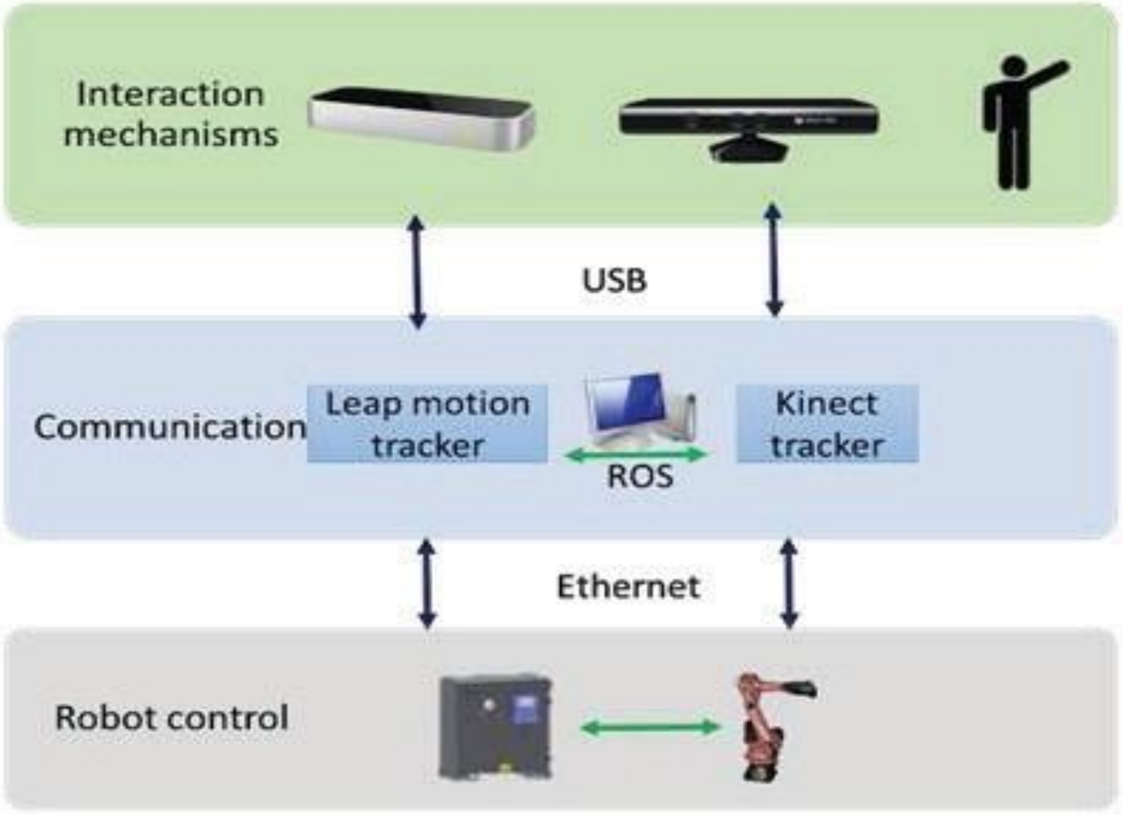
- The data come from two devices:
  - An RGB-D device for the capture of the body gestures,
  - Leap motion for the hand gestures.
- The body gestures control robot arm motion in six directions: $+x, -x, +y, -y, +z, -z$.
- Dynamic hand gestures involve finger movements. They create identical movements to the ones of the body gestures.
- It is applied to on line human and industrial robot interaction.



Performing body gestures as commands for robot [TSA2016].

Robot programming using gestures

- The two sensors (Kinect and Leap motion) are connected to a computer through an ethernet cable.
- A vocabulary includes body and hand gestures as high level robot commands.
- The body gestures are acquired through Kinect. 18 human skeleton nodes are detected.
- Analysis of the nodes ends up to vocabulary-based command classification.
- The data from the Leap motion sensor is used to recognize of the hand gestures in the same way.



HW architecture of Robot programming using gestures [TSA2016].

Human-Drone Interaction

A remote drone control by gesture recognition has 5 modules [HUA2019]:

- A scene-understanding module.
- A pilot detection module.
- An action detection and recognition module.
- A gesture recognition module.
- A joint reasoning and control module.

Human-Drone Interaction

Gesture Controlled Drones

- Issues by using hand controlled devices:
  - limited control by the range of electromagnetic radiation
  - susceptibility to interference noise
- Researchers investigate the use of computer vision methods because of the ability of drones camera to capture surrounding
- Two types:
  - Fixed wing
  - Multirotor

Human-Drone Interaction

Gesture Controlled Drones

- Camera sensors are used because they are low cost, low power.
- Image outputted by the camera may be used for a range of purposes and is send directly to the controller.
- Controller may command the drone a new instruction via a remote control device, depending on the current environment image.
- Current controllers use wi-fi or Bluetooth as a channel to communicate and mobile devices such as phones, tablets or wired gloves.
- Computer vision techniques enable users to move their hands and fingers and perform gestures, which are then converted in digital commands, recognized by sensors.

Human-Drone Interaction

Gesture Controlled Drones

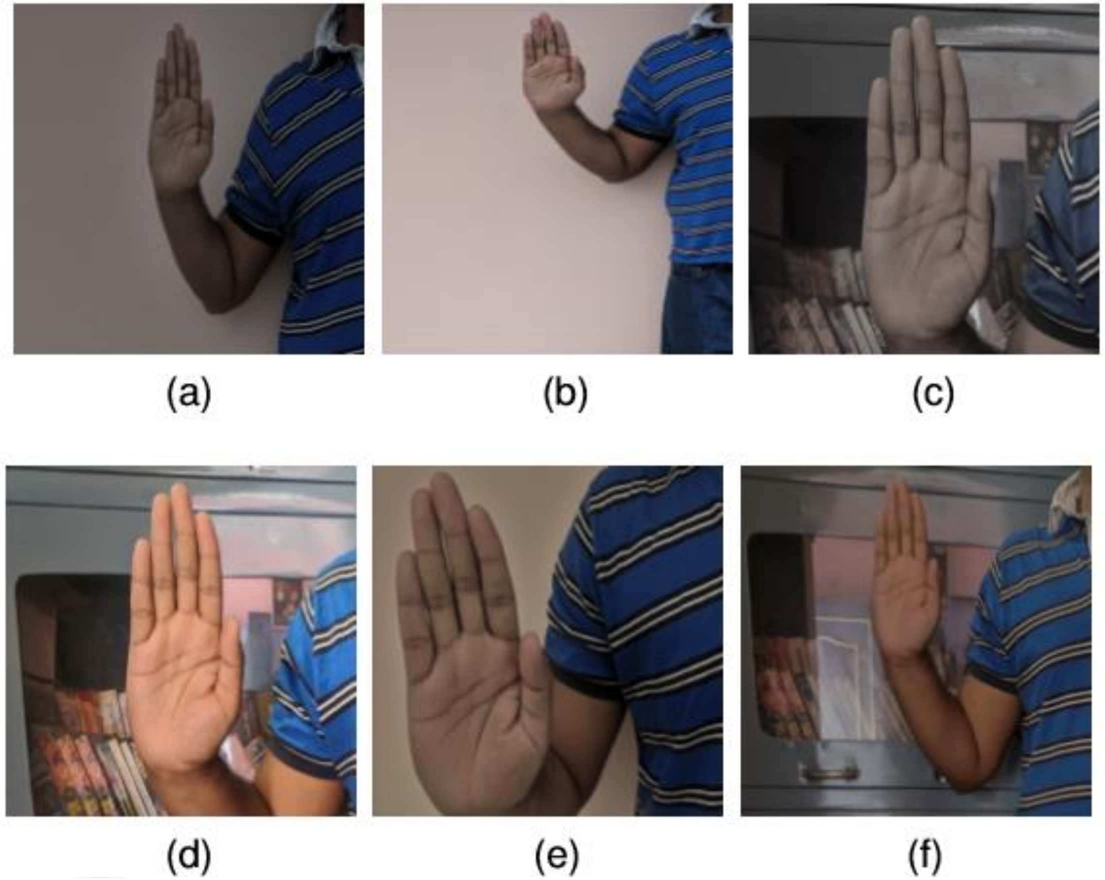
- Video stream is recorded through the camera and segmented into sequences of images.
- Each image is then recognized by a classification process.
- Typical commands:
  - Take off
  - Land
  - Move right or left
- Finally the action planner on the drone

Human-Drone Interaction

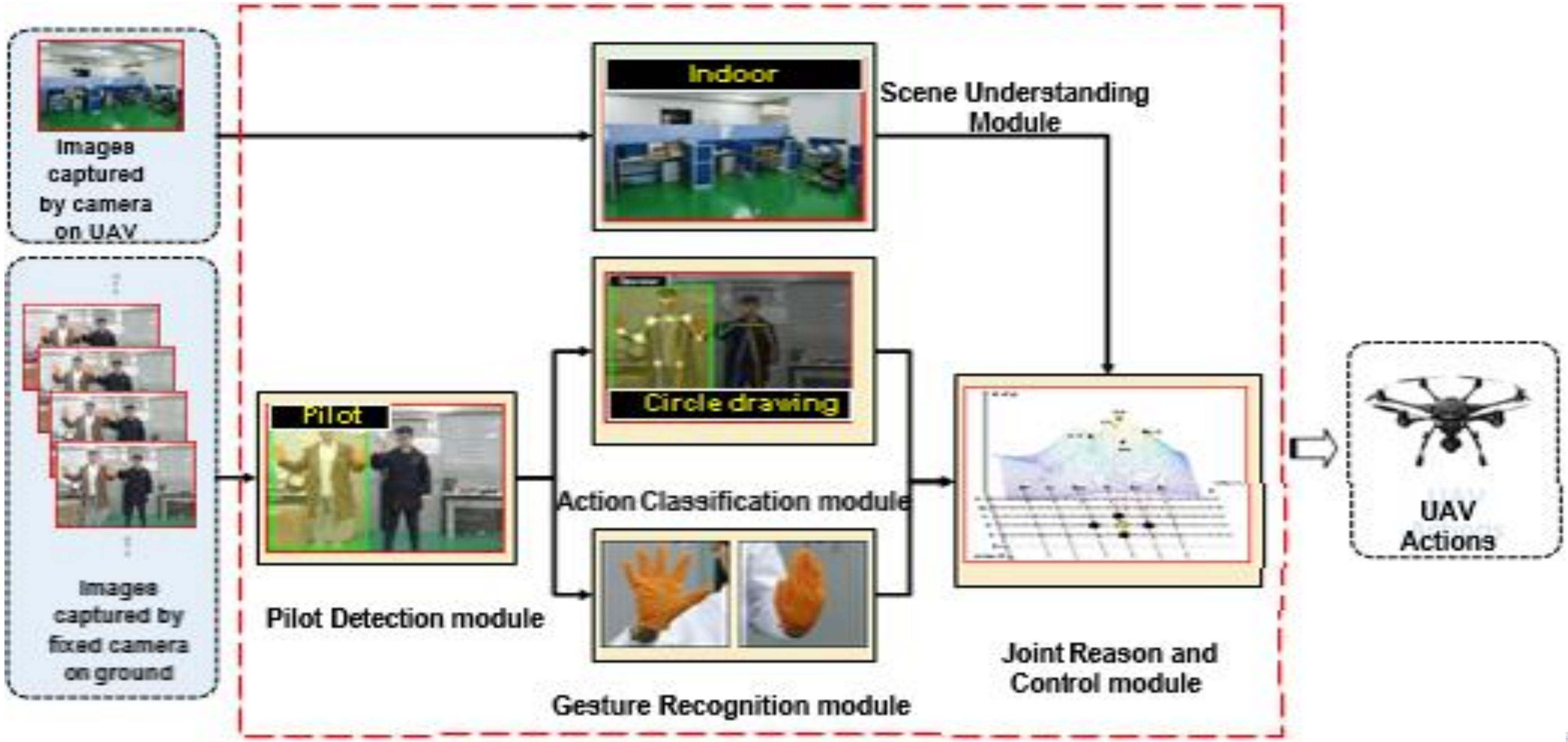
Gesture Controlled Drones

- Safety issues:
  - Misinterpreted gesture
  - Execution of the most appropriate action according to the environment
  - Collision avoidance due to wind, air flows etc

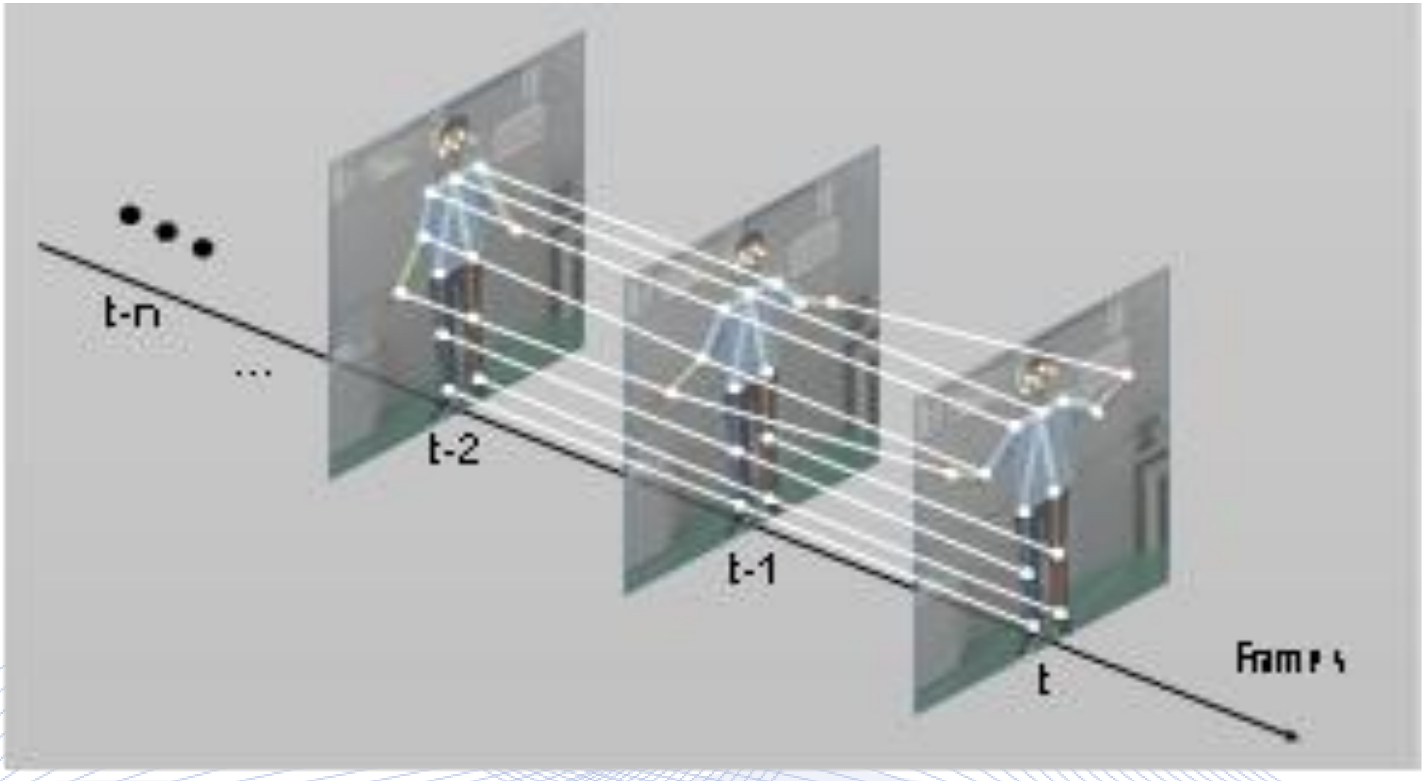
Human-Drone Interaction



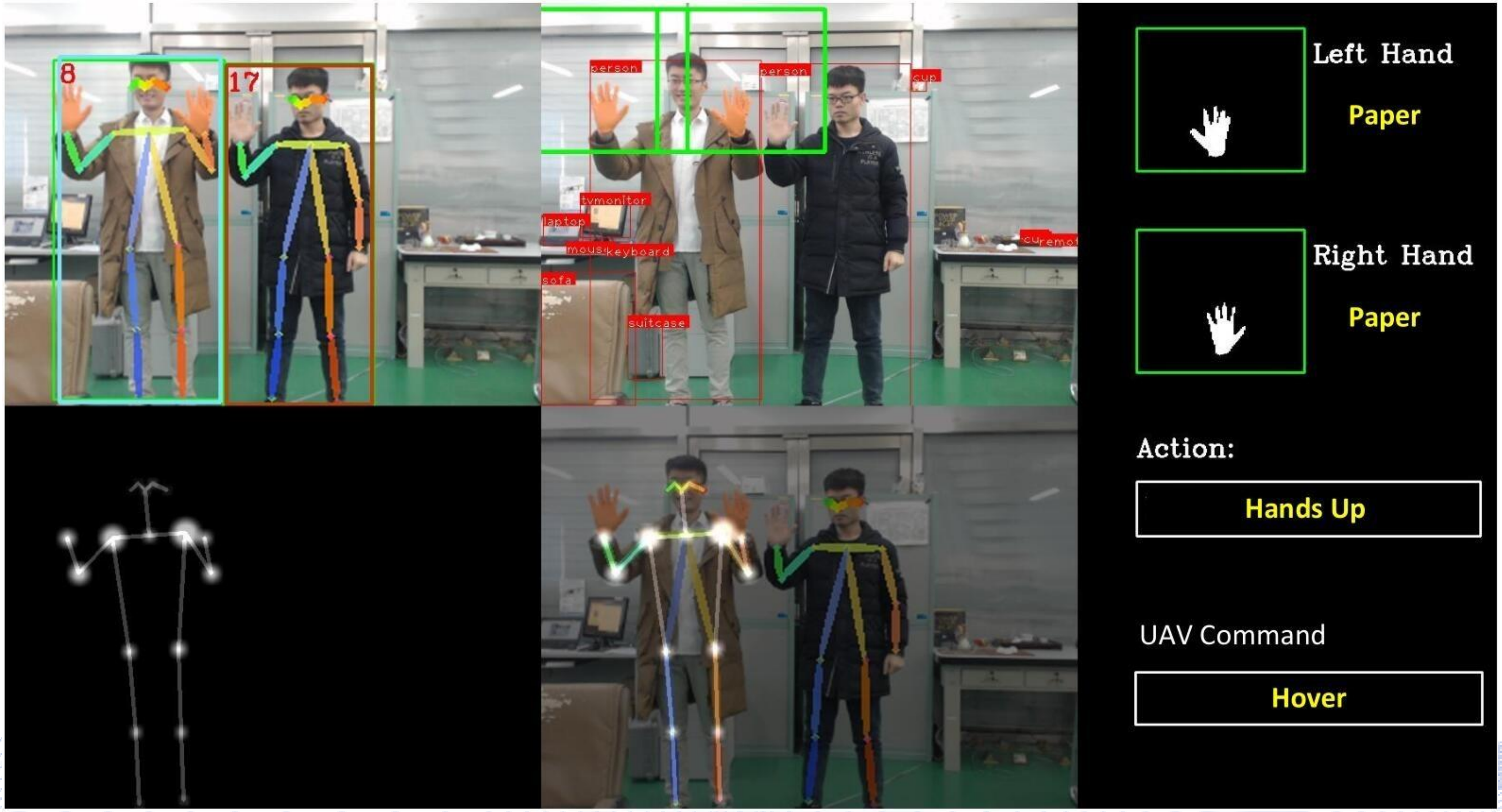
Gestures for drone control [NAT2018].


Images captured by camera on UAV
Images captured by fixed camera on ground
Indoor
Scene Understanding Module
Pilot
Pilot Detection module
Circle drawing
Action Classification module
Gesture Recognition module
Joint Reason and Control module
UAV Actions

Human-Drone Interaction model [HUA2019].



A skeleton sequence represented with the ST-GCN spatial temporal graph, which is used for human action classification [HUA2019].



System performance when two users perform identical gestures [HUA2019].

Bibliography

[WAL2000] Stefan Waldherr, Roseli Romero & Sebastian Thrun " A Gesture Based Interface for Human-Robot Interaction”

[LU2016] Wei Lu, Member, IEEE, Zheng Tong, and Jinghui Chu “Dynamic Hand Gesture Recognition With Leap Motion Controller” IEEE SIGNAL PROCESSING LETTERS, VOL. 23, NO. 9, SEPTEMBER 2016

[VIB1999] M. K. VIBLIS and K. J. KYRIAKOPOULOS “Gesture Recognition: The Gesture Segmentation Problem” Journal of Intelligent and Robotic Systems 28: 151–158, 2000

[YAS2019] Mais Yasen and Shaidah Jusoh “A systematic review on hand gesture recognition techniques, challenges and applications”

[NAT2018] Kathiravan Natarajan, Truong-Huy D. Nguyen, Mutlu Mete “Hand Gesture Controlled Drones: An Open Source Library”

Bibliography

[ZHA2017] Liang Zhang, Guangming Zhu, Peiyi Shen, Juan Song, Syed Afaq Shah, Mohammed Bennamoun “Learning Spatiotemporal Features using 3DCNN and Convolutional LSTM for Gesture Recognition” 2017 IEEE International Conference on Computer Vision Workshops (ICCVW)

[SUN2018] Ying Sun, Cuiqiao Li, Gongfa Li, Guozhang Jiang, Du Jiang, Honghai Liu, Zhigao Zheng,Wanneng Shu “Gesture Recognition Based on Kinect and sEMG Signal Fusion” Mobile Networks and Applications volume 23, pages 797–805(2018)

[KÖP2019] Okan Köpüklü, Ahmet Gunduz, Neslihan Kose, Gerhard Rigoll “Real-time Hand Gesture Detection and Classification Using Convolutional Neural Networks” IEEE International Conference on Automatic Face and Gesture Recognition (FG 2019)

[ESC2017] Sergio Escalera, Isabelle Guyon, Vassilis Athitsos “Gesture Recognition” Book

[ANA] https://www.analyticsvidhya.com/blog/2018/10/computer-vision-approach-hand-gesture-recognition/

[OYE2016] Oyebade K. Oyedotun, Adnan Khashman ” Deep learning in vision-based static hand gesture recognition” Neural Computing and Applications volume 28, pages3941–3951(2017)

[PAV1997] Vladimir I. Pavlovic, Student Member, IEEE, Rajeev Sharma, Member, IEEE, and Thomas S. Huang, Fellow, IEEE “Visual Interpretation of Hand Gestures for Human-Computer Interaction: A Review”

Bibliography

[KIN] https://en.wikipedia.org/wiki/Kinect

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6197001/

[RGB] https://en.wikipedia.org/wiki/RGB_color_model

[WIK]https://en.wikipedia.org/wiki/Artificial_neural_network

[MAG2019] Mehran Maghoumi Joseph J. LaViola Jr. “DeepGRU: Deep Gesture Recognition Utility” published in ISVC 2019

[TRA2019]Dinh-Son Tran, Ngoc-Huynh Ho, Hyung-Jeong Yang * , Eu-Tteum Baek , Soo-Hyung Kim andGueesang Lee "Real-Time Hand Gesture Spotting and RecognitionUsing RGB-D Camera and 3D ConvolutionalNeural Network«

[MOL2016] Pavlo Molchanov Xiaodong Yang Shalini Gupta Kihwan Kim Stephen Tyree Jan KautzNVIDIA "Online Detection and Classification of Dynamic Hand Gestureswith Recurrent 3D Convolutional Neural Networks«

[LUP2020] Katia Lupinetti, Andrea Ranieri, Franca Giannini, and Marina Monti "3D dynamic hand gestures recognition using theLeap Motion sensor and convolutional neuralnetworks"

Bibliography

- **[NOO2019]** Noorkholis Luthfil Hakim, Timothy K. Shih, Sandeli Priyanwada Kasthuri Arachchi, Wisnu Aditya, Yi-Cheng Chen and Chih-Yang Lin “**Dynamic Hand Gesture Recognition Using 3DCNN and LSTM with FSM Context-Aware Model**”.
- **[RAF2012]** Rafiqul Zaman Khan and Noor Adnan Ibraheem “**COMPARATIVE STUDY OF HAND GESTURE RECOGNITION SYSTEM**”.
- **[JES2020]** Jesús Galván-Ruiz, Carlos M. Travieso-González , Acaymo Tejera-Fettmilch, Alejandro Pinan-Roescher, Luis Esteban-Hernández and Luis Domínguez-Quintana “**Perspective and Evolution of Gesture Recognition for Sign Language: A Review**”.
- **[DVS]** https://research.ibm.com/dvsgesture/
- **[SKD]** https://www.kaggle.com/c/odhgdata/data?select=infos_test.csv
- **[JES]** https://20bn.com/datasets/jester/v1
- **[PEM]** https://biolab.put.poznan.pl/putemg-dataset/
- **[HMG]** https://sites.google.com/view/hmd-gesture-dataset/

Bibliography

- **[ASA2018]** Asanka G Perera, Yee Wei Law, and Javaan Chahl "**UAV-GESTURE: A Dataset for UAV Control and Gesture Recognition**".
- **[EGO]** http://www.nlpr.ia.ac.cn/iva/yfzhang/datasets/egogesture.html
- **[GM4]** https://data.mendeley.com/datasets/jzy8zngkbg/4
- **[RAO2020]** Chengping Rao, Yang Liu "**Three dimensional convolutional neural network (3D-CNN) for heterogeneous material homogenization**".
- **[EGO2018]** Yifan Zhang, Congqi Cao, Jian Cheng, and Hanqing Lu **"EgoGesture: A New Dataset and Benchmark for Egocentric Hand Gesture Recognition".**
- **[DON2020]** Dongxu Li, Cristian Rodriguez Opazo, Xin Yu, Hongdong Li **"Word-level Deep Sign Language Recognition from Video: A New Large-scale Dataset and Methods Comparison**".
- **[LIU2020]** Jianbo Liu, Yongcheng Liu, Ying Wang, Veronique Prine, Shiming Xiang, Chunhong Pan "**Decoupled Representation Learning for Skeleton-Based Gesture Recognition**".

Bibliography

- **[CHA2018]** Chao Li, Qiaoyong Zhong, Di Xie, Shiliang Pu. "**Co- occurrence feature learning from skeleton data for action recognition and detection with hierarchical aggregation**".
- **[QUE2017]** Quentin De Smedt, Hazem Wannous, Jean-Philippe Vande- borre, Joris Guerry, Bertrand Le Saux, and David Filliat. "**Shrec'17 track: 3d hand gesture recognition using a depth and skeletal dataset**".
- **[HAZ2016]** Quentin De Smedt, Hazem Wannous, and Jean-Philippe Van- deborre "**Skeleton-based dynamic hand gesture recognition**".
- **[GUI2018]** Guillermo Garcia-Hernando, Shanxin Yuan, Seungryul Baek, and Tae-Kyun Kim "**First-person hand action bench- mark with rgb-d videos and 3d hand pose annotations**".
- **[OMA2013]** Omar Oreifej and Zicheng Liu. Hon4d "**Histogram of oriented 4d normals for activity recognition from depth sequences**".
- **[SME2017]** Quentin De Smedt "**Dynamic hand gesture recognition-From traditional handcrafted to recent deep learning approaches**".

Bibliography

- **[JUA2018]** Juanhui Tu, Mengyuan Liu, and Hanying Liu “**Skeleton- based human action recognition using spatial temporal 3d convolutional neural networks**”.
- **[JIN2018]** Jingxuan Hou, Guijin Wang, Xinghao Chen, Jing-Hao Xue, Rui Zhu, Huazhong Yang “**Spatial-temporal attention res-tcn for skeleton-based dynamic hand gesture recognition**”.
- **[YAN2018]** Sijie Yan, Yuanjun Xiong, Dahua Lin “**Spatial temporal graph convolutional networks for skeleton-based action recognition**”.
- **[XUA2019]** Xuan Son Nguyen, Luc Brun, Olivier Le ́zoray, and Se ́bastien Bougleux “**A neural network based on spd manifold learning for skeleton-based hand gesture recognition**”.
- **[YUX2019]** Yuxiao Chen, Long Zhao, Xi Peng, Jianbo Yuan, and Dimitris N Metaxas “**Construct dynamic graphs for hand gesture recognition via spatial-temporal attention**”.
- **[KAI2016]** Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun “**Deep residual learning for image recognition”.**

Bibliography

- **[ZEN2018]** Nico Zengeler , Thomas Kopinski and Uwe Handmann **“Hand Gesture Recognition in Automotive Human–Machine Interaction Using Depth Cameras”**
- **[TSA2016]** Panagiota Tsarouchi, Athanasios Athanasatos, Sotiris Makris, Xenofon Chatzigeorgiou, George Chryssolouris **“High level robot programming using body and hand gestures”**
- **[HUA2019]** Bo Chen, Chunsheng Hua, Decai Li, Yuqing He and Jianda Han **“Intelligent Human–UAV Interaction System with Joint Cross-Validation over Action–Gesture Recognition and Scene Understanding”**
- **[AZO]** https://www.azosensors.com/article.aspx?ArticleID=1149
- **[SON2016]** S. Song *et al*., "Multimodal Multi-Stream Deep Learning for Egocentric Activity Recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Las Vegas, NV, 2016, pp. 378-385, doi: 10.1109/CVPRW.2016.54.

Bibliography

- **[CAO2017]** C. Cao, Y. Zhang, Y. Wu, H. Lu and J. Cheng**, "Egocentric Gesture Recognition Using Recurrent 3D Convolutional Neural Networks with Spatiotemporal Transformer Modules,"** *2017 IEEE International Conference on Computer Vision (ICCV)*, Venice, 2017, pp. 3783-3791, doi: 10.1109/ICCV.2017.406.

Bibliography

[PIT2021] I. Pitas, “Computer vision”, Createspace/Amazon, in press.

[PIT2017] I. Pitas, “Digital video processing and analysis” , China Machine Press, 2017 (in Chinese).

[PIT2013] I. Pitas, “Digital Video and Television” , Createspace/Amazon, 2013.

[NIK2000] N. Nikolaidis and I. Pitas, “3D Image Processing Algorithms”, J. Wiley, 2000.

[PIT2000] I. Pitas, “Digital Image Processing Algorithms and Applications”, J. Wiley, 2000.

Q & A

Thank you very much for your attention!

More material in
http://icarus.csd.auth.gr/cvml-web-lecture-series/

Contact: Prof. I. Pitas
pitas@csd.auth.gr