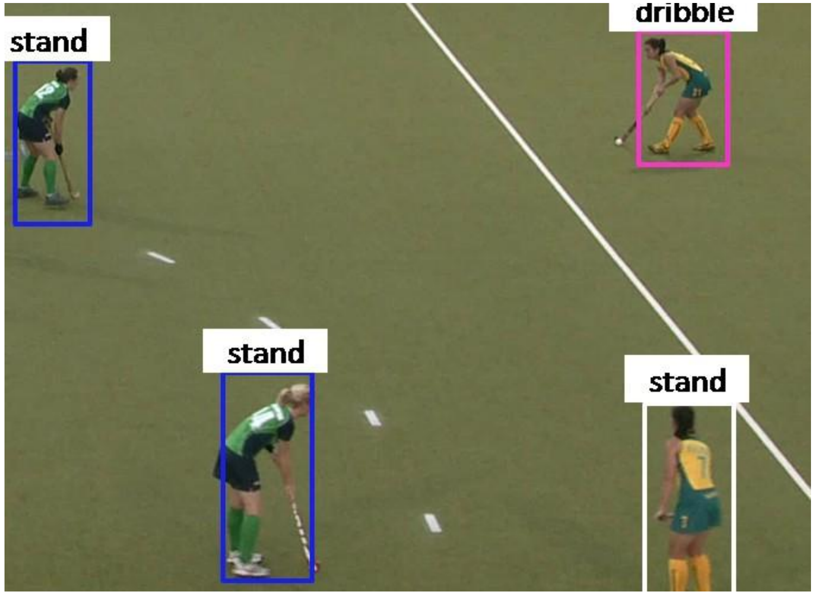
A Human Activity Recognition System Using Skeleton Data from RGBD Sensors

Enea Cippitelli, Samuele Gasparrini, Ennio Gambi and Susanna Spinsante

-Presented By: Dhanesh Pradhan

Motivation

- Activity Recognition is an important technology in pervasive computing because it can be applied to many real-life, human-centric problems such as eldercare and healthcare.



- Many applications, including video surveillance systems, human-computer interaction, and robotics for human behavior characterization, require a multiple activity recognition system.

Motivation

Contents

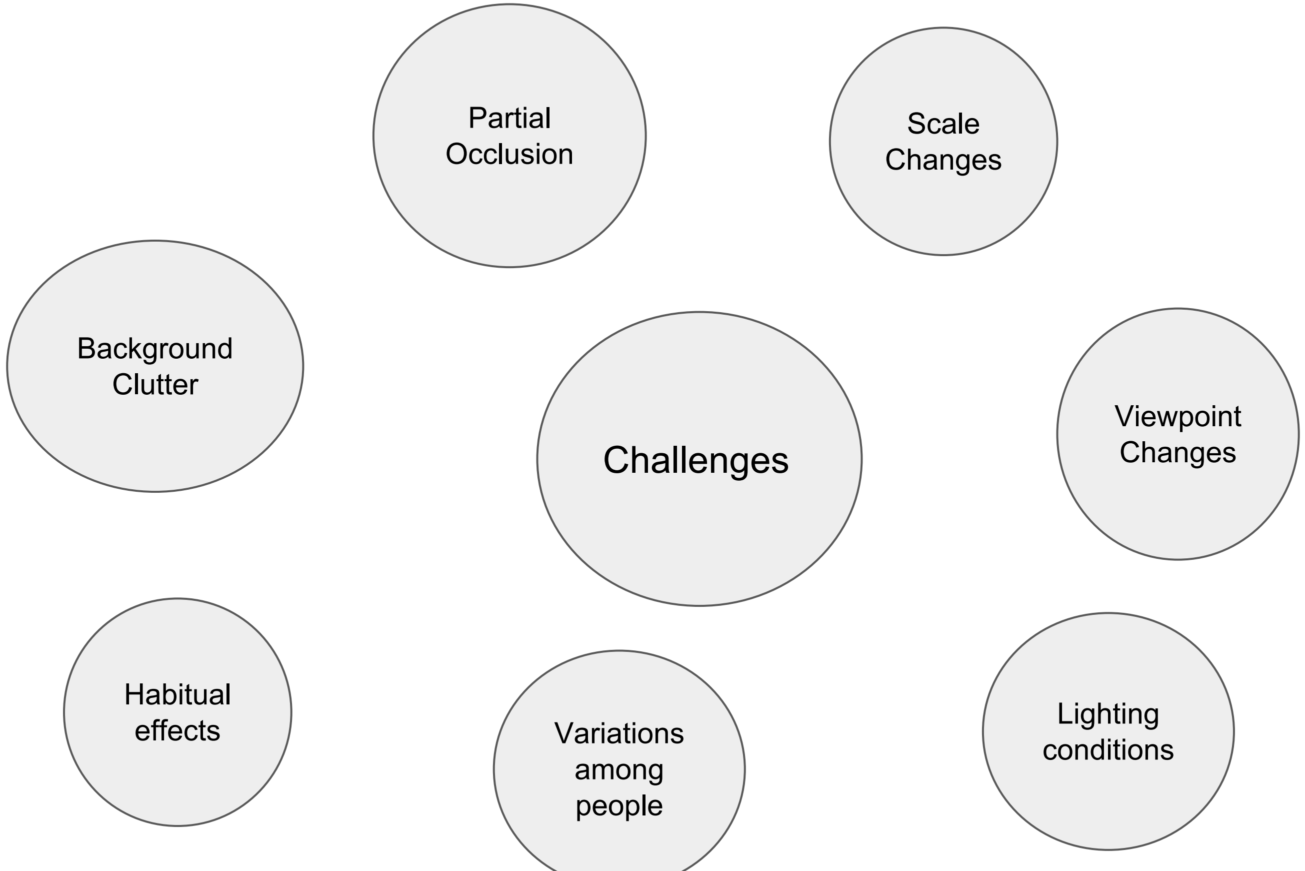
Introduction: Human Activity Recognition

- Human activity recognition plays a significant role in human-to-human interaction and interpersonal relations.

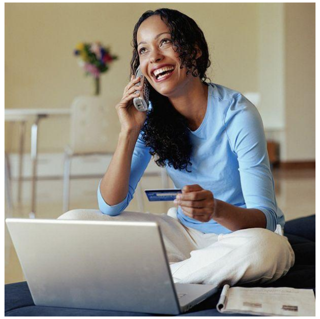
- Among various classification techniques two main questions arise: “What action?” (i.e., the recognition problem) and “Where in the video?” (i.e., the localization problem).

- Human activities, such as “walking” and “running,” arise very naturally in daily life and are relatively easy to recognize. On the other hand, more complex activities, such as “peeling an apple,” are more difficult to identify.

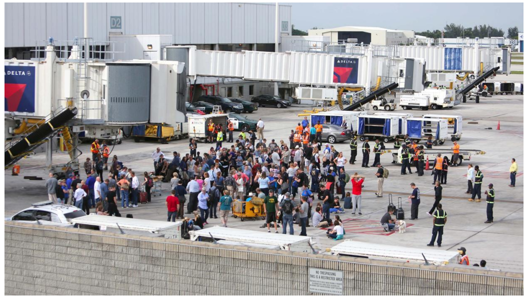
Challenges in Human Activity Recognition



Challenges in Human Activity Recognition (Contd.)



Concurrent Activities



Multiple Person

Further Challenges

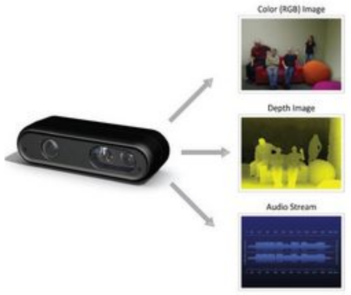
Interleaved Activities

Ambiguity of interpretation

Overview of Depth Camera Systems

Mesa Imaging SwissRanger 4000
Range - 5 to 8 m, Resolution - 176 x 144
Field of view - 43.6 x 34.6 degrees, 54 fps
Cost - $ 9000

PMD Technologies CamCube 2.0
Range - 7 m, Resolution - 204 x 204
Field of view - 40.0 x 40.0 degrees, 25 fps
Cost - $12000



PrimeSense
Resolution - 640 x 480, 60 fps

Kinect:RGBD Sensor

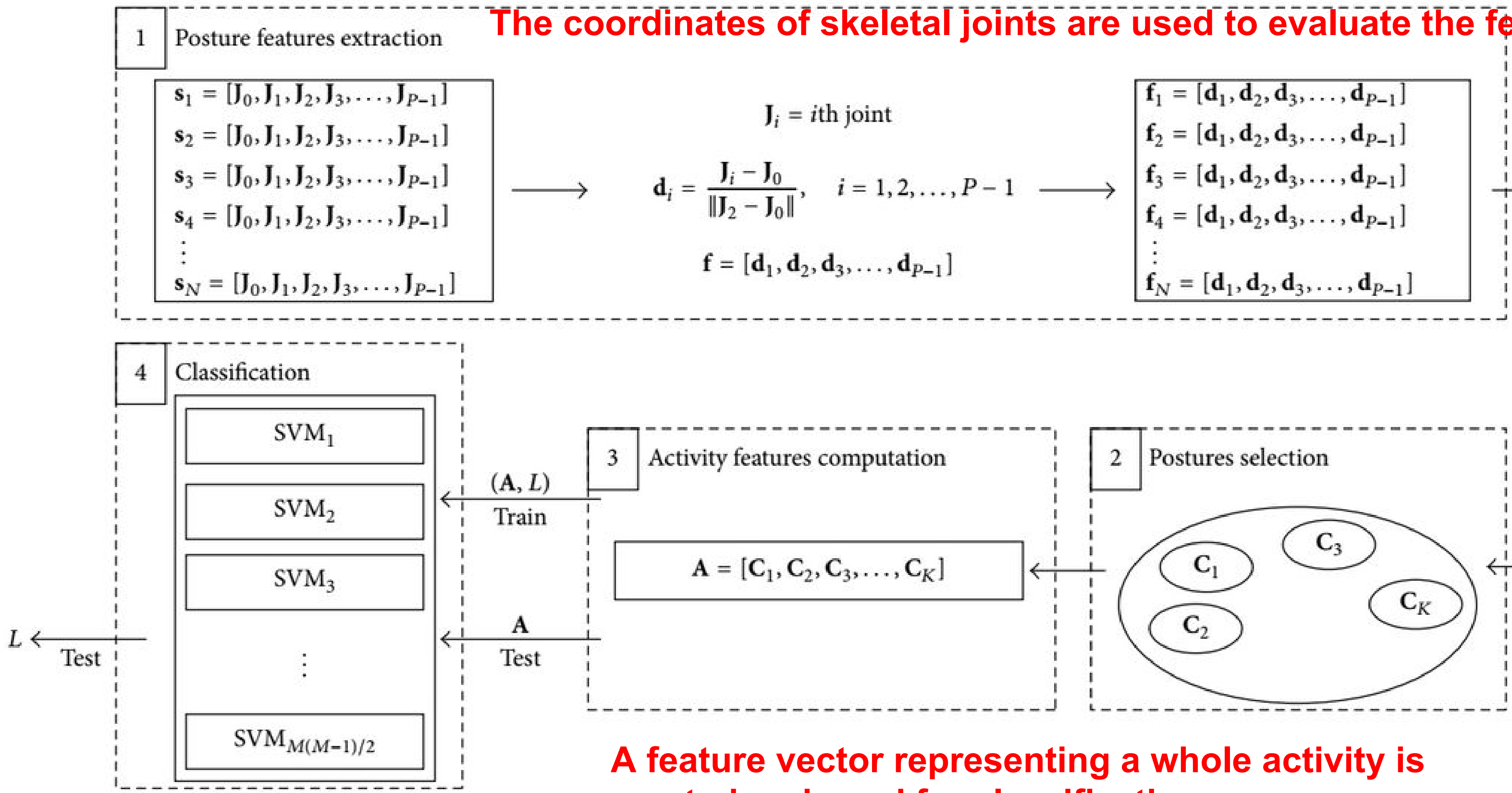
- **Kinect** (codenamed **Project Natal** during development) is a line of motion sensing input devices by Microsoft

- Kinect has following in-built sensors:

1. **Color VGA video camera**
2. **Depth sensor**
3. **Multi-array microphone**

Video and depth sensor cameras have a 640 x 480-pixel resolution and run at 30 fps, Range - 4 m
Field of view - 43 x 57 degrees horizontal and 27 degrees vertical tilt angle

Activity Recognition Architecture



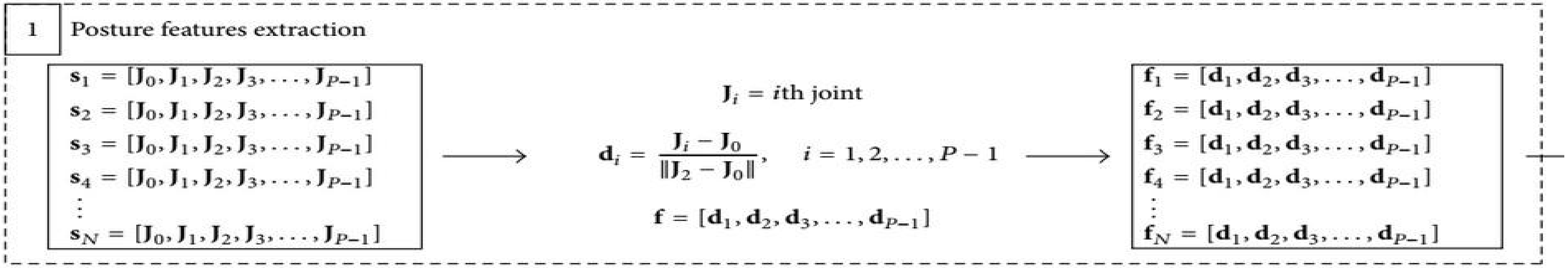
1
Posture features extraction
$s_1 = [J_0, J_1, J_2, J_3, \dots, J_{P-1}]$
$s_2 = [J_0, J_1, J_2, J_3, \dots, J_{P-1}]$
$s_3 = [J_0, J_1, J_2, J_3, \dots, J_{P-1}]$
$s_4 = [J_0, J_1, J_2, J_3, \dots, J_{P-1}]$
$s_N = [J_0, J_1, J_2, J_3, \dots, J_{P-1}]$
J_i = ith joint
$d_i = \frac{J_i - J_0}{\|J_2 - J_0\|}, \quad i = 1, 2, \dots, P-1$
$f = [d_1, d_2, d_3, \dots, d_{P-1}]$
$f_1 = [d_1, d_2, d_3, \dots, d_{P-1}]$
$f_2 = [d_1, d_2, d_3, \dots, d_{P-1}]$
$f_3 = [d_1, d_2, d_3, \dots, d_{P-1}]$
$f_4 = [d_1, d_2, d_3, \dots, d_{P-1}]$
$f_N = [d_1, d_2, d_3, \dots, d_{P-1}]$
4
Classification
SVM_1
SVM_2
SVM_3
$SVM_{M(M-1)/2}$
(A, L)
Train
A
Test
L
Test
3
Activity features computation
$A = [C_1, C_2, C_3, \dots, C_K]$
2
Postures selection
C_1
C_2
C_3
C_K

The coordinates of skeletal joints are used to evaluate the feature vectors

The most important postures for each activity are selected

A feature vector representing a whole activity is created and used for classification

Posture Features Extraction



$$\begin{aligned} s_1 &= [J_0, J_1, J_2, J_3, \ldots, J_{P-1}] \\ s_2 &= [J_0, J_1, J_2, J_3, \ldots, J_{P-1}] \\ s_3 &= [J_0, J_1, J_2, J_3, \ldots, J_{P-1}] \\ s_4 &= [J_0, J_1, J_2, J_3, \ldots, J_{P-1}] \\ &\vdots \\ s_N &= [J_0, J_1, J_2, J_3, \ldots, J_{P-1}] \end{aligned} \longrightarrow$$

$$J_i = i\text{th joint}$$

$$\mathbf{d}_i = \frac{\mathbf{J}_i - \mathbf{J}_0}{\|\mathbf{J}_2 - \mathbf{J}_0\|}, \quad i = 1, 2, \ldots, P-1 \longrightarrow$$

$$\mathbf{f} = [\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3, \ldots, \mathbf{d}_{P-1}]$$

$$\begin{aligned} f_1 &= [d_1, d_2, d_3, \ldots, d_{P-1}] \\ f_2 &= [d_1, d_2, d_3, \ldots, d_{P-1}] \\ f_3 &= [d_1, d_2, d_3, \ldots, d_{P-1}] \\ f_4 &= [d_1, d_2, d_3, \ldots, d_{P-1}] \\ &\vdots \\ f_N &= [d_1, d_2, d_3, \ldots, d_{P-1}] \end{aligned}$$

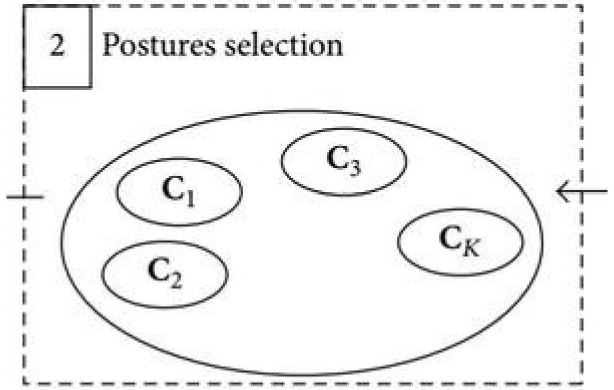
- Each joint J_i is represented by 3D vector in coordinate space of kinect.
- Consider skeleton composed of P joints, J_0 - torso joint, J_2 - neck joint
- A posture feature vector f is created for each skeleton frame:

$$f = [d_1, d_2, d_3, \ldots]$$

- A set of N feature vectors is computed, having an activity in N frames

$$\mathbf{d}_i = \frac{\mathbf{J}_i - \mathbf{J}_0}{\|\mathbf{J}_2 - \mathbf{J}_0\|}, \quad i = 1, 2, \ldots, P-1.$$

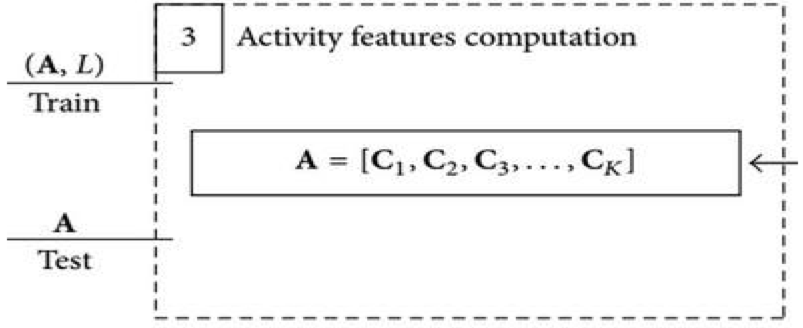
Posture Selection



- Using k-means clustering approach, N feature vectors are grouped into K clusters.
- The clustering algorithm gives as outputs N clusters ID and K vectors
  $[C_1, C_2, C_3, \ldots, C_K]$
- The feature vectors are partitioned into clusters $S_1, S_2, \ldots, S_K$ so as to satisfy

$$\arg\min_{\mathbf{S}} \sum_{j=1}^{K} \sum_{\mathbf{f}_i \in S_j} \left\| \mathbf{f}_i - \mathbf{C}_j \right\|^2 .$$

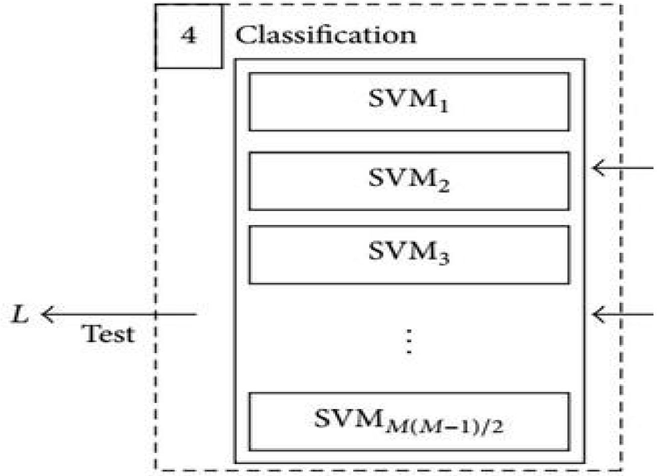
Activity Feature Computation



- The third phase is related to computation of feature vector which models whole activity.
- Consider an activity featured by N = 10 and K = 4, after running k-means clustering, one possible output can be following sequence of cluster IDs,

  [2,2,2,3,3,1,1,4,4,4]
- So now activity feature vector A = $[C_2, C_3, C_1, C_4]$

Classification



- SVM is used for classification.
- Consider l training vectors, $x_i \in R^n$ and a vector of labels $y \in R^l$ where $y_i \in \{-1,1\}$, a binary SVM can be formulate

$$\min_{\mathbf{w},b,\xi} \quad \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{l}\xi_i$$

$$\text{subject to} \quad y_i\left(\mathbf{w}^T\phi(\mathbf{x}_i) + b\right) \geq 1 - \xi_i,$$

$$\xi_i \geq 0, \quad i = 1, \ldots, l,$$

where

$$\mathbf{w}^T\phi(\mathbf{x}) + b = 0$$

- Radial Basis Function (RBF) kernel has been used where $K\left(\mathbf{x}_i, \mathbf{x}_j\right) = e^{-\gamma\|\mathbf{x}_i - \mathbf{x}_j\|^2}, \quad \gamma > 0.$

Experimental Results

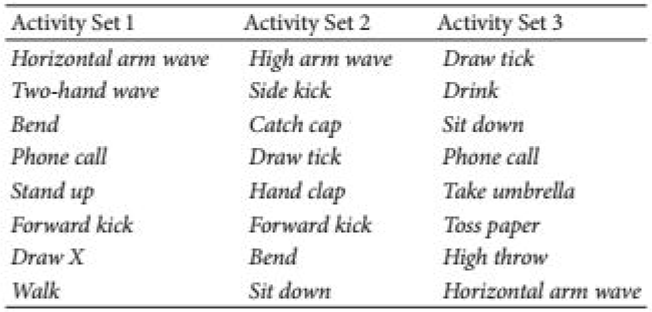
- The algorithm is evaluated on 5 publicly available datasets:

1. KARD Dataset
2. Cornell Activity Dataset (CAD-60)
3. UTKinect Dataset
4. Florence3D Dataset
5. MSR Action3D Dataset

- The algorithm overcomes state-of-the-art results for KARD and CAD-60 datasets.

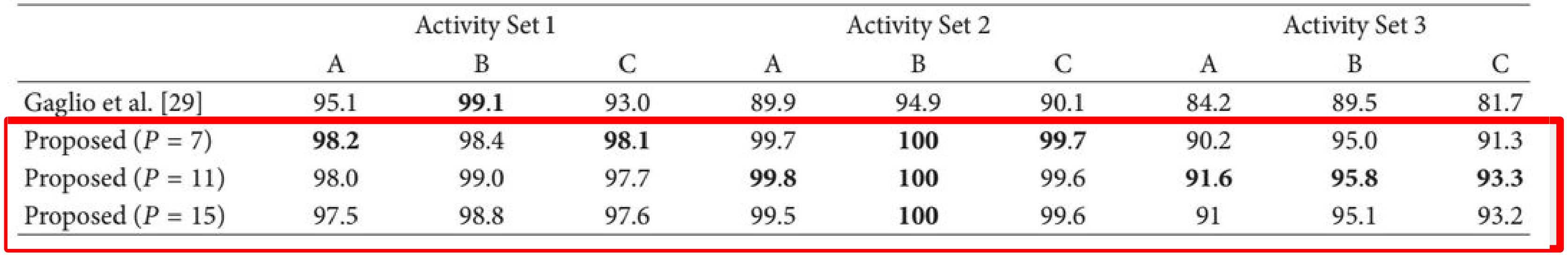
KARD Dataset Results

- KARD Dataset is composed of 18 activities that can be divided into 10 gestures and 8 actions.



Activity Set 1	Activity Set 2	Activity Set 3
Horizontal arm wave	*High arm wave*	*Draw tick*
Two-hand wave	*Side kick*	*Drink*
Bend	*Catch cap*	*Sit down*
Phone call	*Draw tick*	*Phone call*
Stand up	*Hand clap*	*Take umbrella*
Forward kick	*Forward kick*	*Toss paper*
Draw X	*Bend*	*High throw*
Walk	*Sit down*	*Horizontal arm wave*

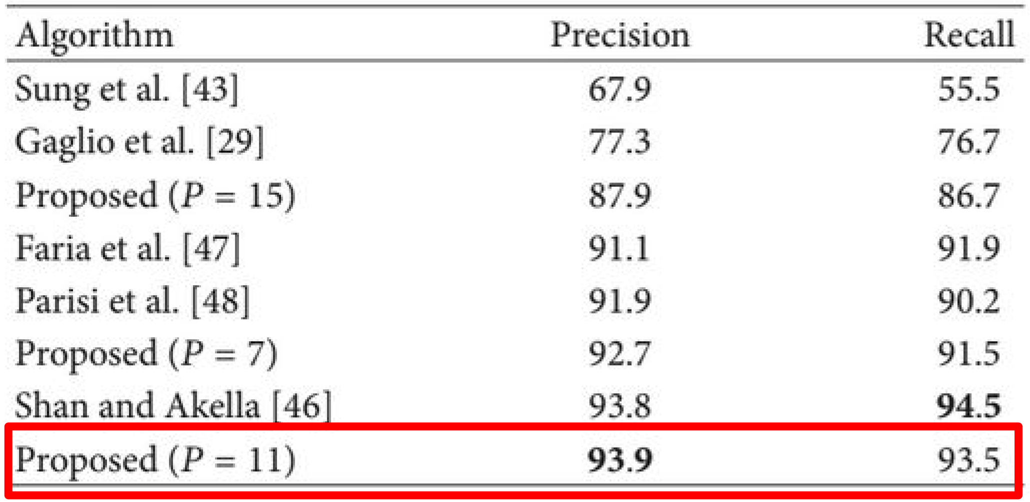
Table 2: Accuracy (%) of the proposed algorithm compared to the other using KARD dataset with different Activity Sets and for different experiments.



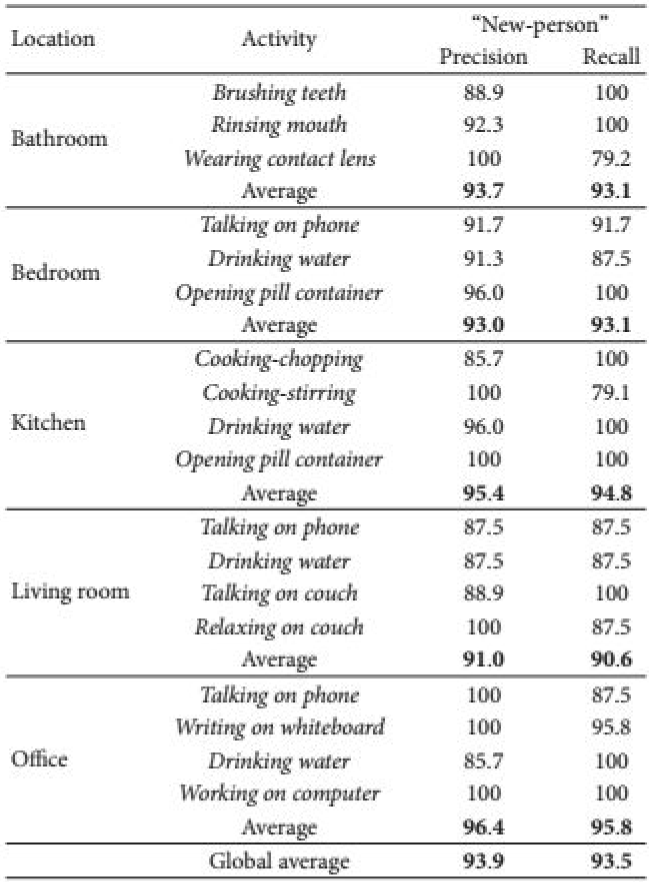
	Activity Set 1			Activity Set 2			Activity Set 3		
	A	B	C	A	B	C	A	B	C
Gaglio et al. [29]	95.1	**99.1**	93.0	89.9	94.9	90.1	84.2	89.5	81.7
Proposed ($P = 7$)	**98.2**	98.4	**98.1**	99.7	**100**	**99.7**	90.2	95.0	91.3
Proposed ($P = 11$)	98.0	99.0	97.7	**99.8**	**100**	99.6	**91.6**	**95.8**	**93.3**
Proposed ($P = 15$)	97.5	98.8	97.6	99.5	**100**	99.6	91	95.1	93.2

CAD-60 Dataset

- This dataset consists of 12 activities performed by 4 people in 5 different environments.



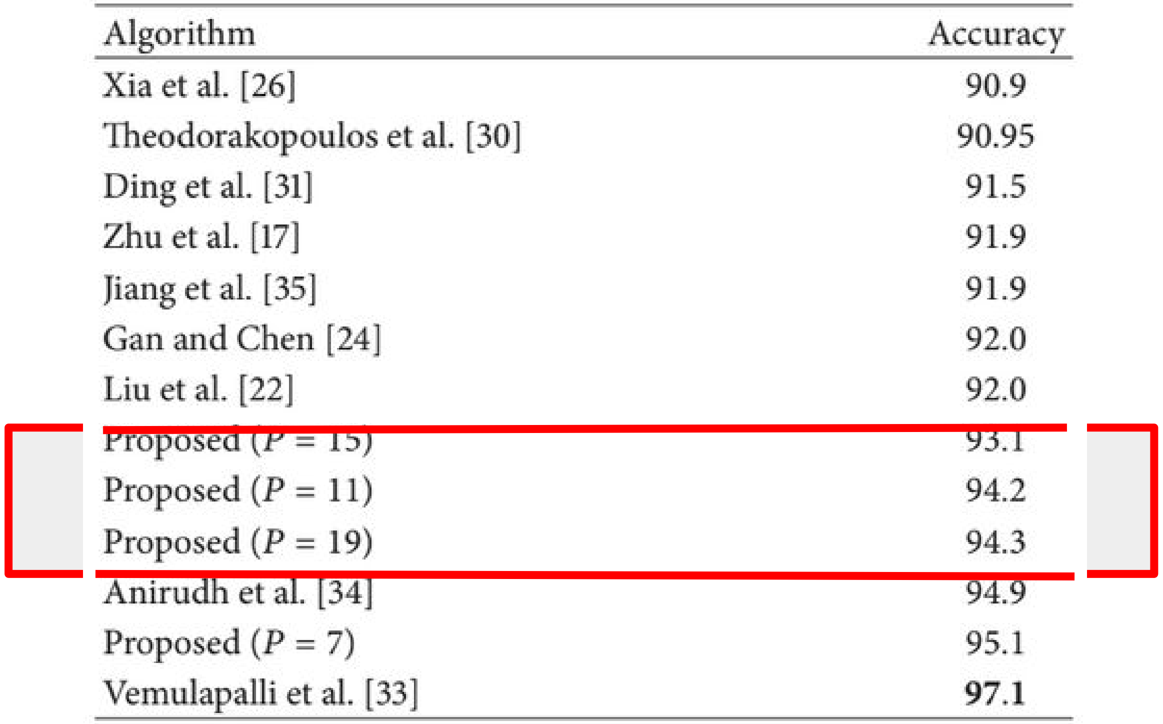
Algorithm	Precision	Recall
Sung et al. [43]	67.9	55.5
Gaglio et al. [29]	77.3	76.7
Proposed ($P = 15$)	87.9	86.7
Faria et al. [47]	91.1	91.9
Parisi et al. [48]	91.9	90.2
Proposed ($P = 7$)	92.7	91.5
Shan and Akella [46]	93.8	**94.5**
Proposed ($P = 11$)	**93.9**	93.5



Location	Activity	"New-person"	
		Precision	Recall
Bathroom	*Brushing teeth*	88.9	100
	Rinsing mouth	92.3	100
	Wearing contact lens	100	79.2
	Average	**93.7**	**93.1**
Bedroom	*Talking on phone*	91.7	91.7
	Drinking water	91.3	87.5
	Opening pill container	96.0	100
	Average	**93.0**	**93.1**
Kitchen	*Cooking-chopping*	85.7	100
	Cooking-stirring	100	79.1
	Drinking water	96.0	100
	Opening pill container	100	100
	Average	**95.4**	**94.8**
Living room	*Talking on phone*	87.5	87.5
	Drinking water	87.5	87.5
	Talking on couch	88.9	100
	Relaxing on couch	100	87.5
	Average	**91.0**	**90.6**
Office	*Talking on phone*	100	87.5
	Writing on whiteboard	100	95.8
	Drinking water	85.7	100
	Working on computer	100	100
	Average	**96.4**	**95.8**
	Global average	**93.9**	**93.5**

UTKinect Dataset

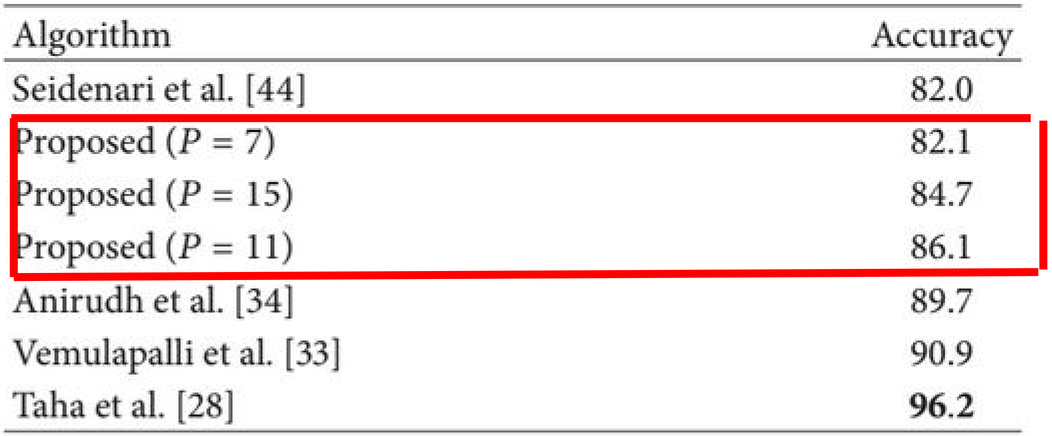
- This dataset is composed of 10 activities performed twice by 10 subjects.



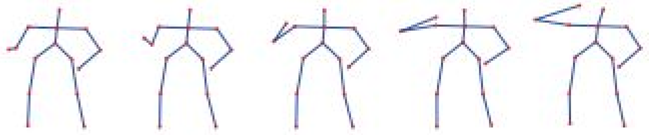
Algorithm	Accuracy
Xia et al. [26]	90.9
Theodorakopoulos et al. [30]	90.95
Ding et al. [31]	91.5
Zhu et al. [17]	91.9
Jiang et al. [35]	91.9
Gan and Chen [24]	92.0
Liu et al. [22]	92.0
Proposed ($P = 15$)	93.1
Proposed ($P = 11$)	94.2
Proposed ($P = 19$)	94.3
Anirudh et al. [34]	94.9
Proposed ($P = 7$)	95.1
Vemulapalli et al. [33]	**97.1**

Florence3D dataset

- This dataset is composed of 9 activities and 10 people performing them multiple times resulting in 215 activities.



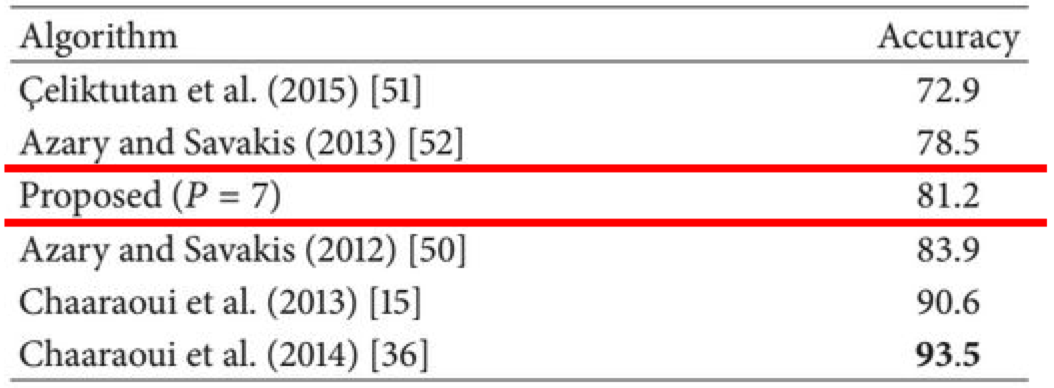
Algorithm	Accuracy
Seidenari et al. [44]	82.0
Proposed ($P = 7$)	82.1
Proposed ($P = 15$)	84.7
Proposed ($P = 11$)	86.1
Anirudh et al. [34]	89.7
Vemulapalli et al. [33]	90.9
Taha et al. [28]	**96.2**



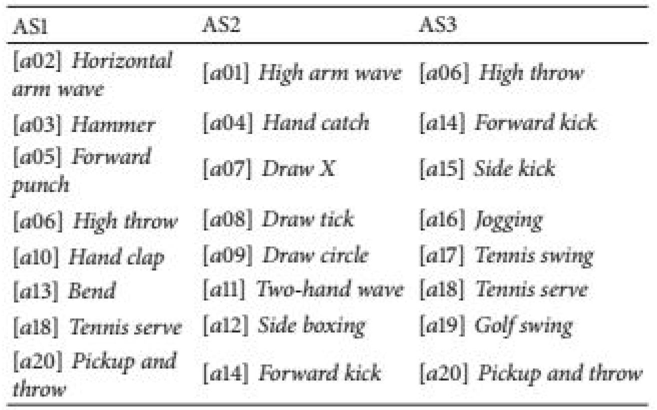
Example Activity: Drink from a bottle

MSR Action3D dataset

- This dataset is composed of 20 activities which are mainly gestures.



Algorithm	Accuracy
Çeliktutan et al. (2015) [51]	72.9
Azary and Savakis (2013) [52]	78.5
Proposed (*P* = 7)	81.2
Azary and Savakis (2012) [50]	83.9
Chaaraoui et al. (2013) [15]	90.6
Chaaraoui et al. (2014) [36]	**93.5**



AS1	AS2	AS3
[a02] *Horizontal arm wave*	[a01] *High arm wave*	[a06] *High throw*
[a03] *Hammer*	[a04] *Hand catch*	[a14] *Forward kick*
[a05] *Forward punch*	[a07] *Draw X*	[a15] *Side kick*
[a06] *High throw*	[a08] *Draw tick*	[a16] *Jogging*
[a10] *Hand clap*	[a09] *Draw circle*	[a17] *Tennis swing*
[a13] *Bend*	[a11] *Two-hand wave*	[a18] *Tennis serve*
[a18] *Tennis serve*	[a12] *Side boxing*	[a19] *Golf swing*
[a20] *Pickup and throw*	[a14] *Forward kick*	[a20] *Pickup and throw*

Conclusion

- The paper presents a simple yet effective activity recognition algorithm.
- It is able to overcome state-of-the-art results in two publicly available datasets - KARD and CAD-60, outperforming more complex algorithms in many conditions.

Hyperdimensional Computing: An Introduction to Computing in Distributed Representation with High-Dimensional Random Vectors

Pentti Kanerva (Redwood Center for Theoretical Neuroscience, UC Berkeley)

- **Presented By: Dhanesh Pradhan**

Contents

Hyperdimensional Computing

- Brain inspired hyperdimensional computing explores the emulation of cognition by computing with hypervectors as an alternative to computing with numbers.
- Hypervectors are high-dimensional, holographic, and (pseudo) random with independent and identically distributed (i.i.d) components.

Properties of Hyperdimensional Computing

- ❑ **Hyperdimensionality:**
  - Brain’s circuits are massive in terms of number of neurons and synapses, suggesting large circuits are fundamentally required.
  - When dimensionality is in thousands then we call it hyperdimensional or hyperspace. Vectors in such space are hypervectors.
- ❑ **Robustness:**
  - Brain’s architecture is amazingly tolerant of component failure. The robustness comes from redundant representation, in which many patterns are considered equivalent.
  - The number of places at which equivalent patterns may differ becomes quite large in hyperdimensional representation. Proportion of allowable errors increases with dimensionality.

Properties of Hyperdimensional Computing (contd.)

- ❑ **Independence from Position: Holistic Representation**
  - For maximum robustness the information encoded into a representation should be distributed “equally” over all the components. This representation is referred to as holographic or holistic.
- ❑ **Randomness**
  - Mind cannot be “downloaded” from one brain to another. The system builds its model of the world from random patterns, by starting with vectors drawn randomly from the hyperspace.

Backup Slides

References

1. Enea Cippitelli, Samuele Gasparrini, Ennio Gambi, and Susanna Spinsante “A Human Activity Recognition System Using Skeleton Data from RGBD Sensors”
2. M Vrigkas, C Nikou, I Kakadiaris “A Review of Human Activity Recognition Methods”
3. Eunju Kim,Sumi HelalandDiane Cook “Human Activity Recognition and Pattern Discovery”

Introduction: The Brain as a Computer

- No two brains are identical yet they can produce the same behavior- they can be **functionally equivalent**
- Brains with different “hardware” and internal code accomplish the same computing. Details of code are established over time through interaction with the world.
- **Disparity in architecture** of brains and computers is matched by **disparity in performance**. Computers excel in routine tasks such as calculation, whereas they are yet to be programmed for universal human traits.

System tradeoffs based on Representation

- Choice of representation is often a compromise. Example- Any number can be represented as product of its prime factors and it makes multiplication easy but all other arithmetic difficult. So base-2 system is an excellent compromise for overall efficiency in arithmetic.
- Similarly, brain’s representation of numbers and magnitude are subject to all sorts of context effects. So brain is not optimized for fast and accurate arithmetic but for more vital functions.
- Many machine learning algorithms try for dimensionality reduction. But high dimensionality can be a blessing instead of curse. Example- Numbers (scalars) are 1-dimensional but represented as 32 bit (32 dimensional) binary vector in computers.

Language Identification: Application of HDC

- Language identification based on letter trigrams.
- Key results are:

1. HD classifier is 96.7% accurate which is 1.2% lower than a conventional machine learning. But operates with only half of the energy (47% energy of baseline classifier).

2. Able to tolerate 8.8 fold probability of failure of memory cells while maintaining 94% accuracy.

Language Identification System Architecture

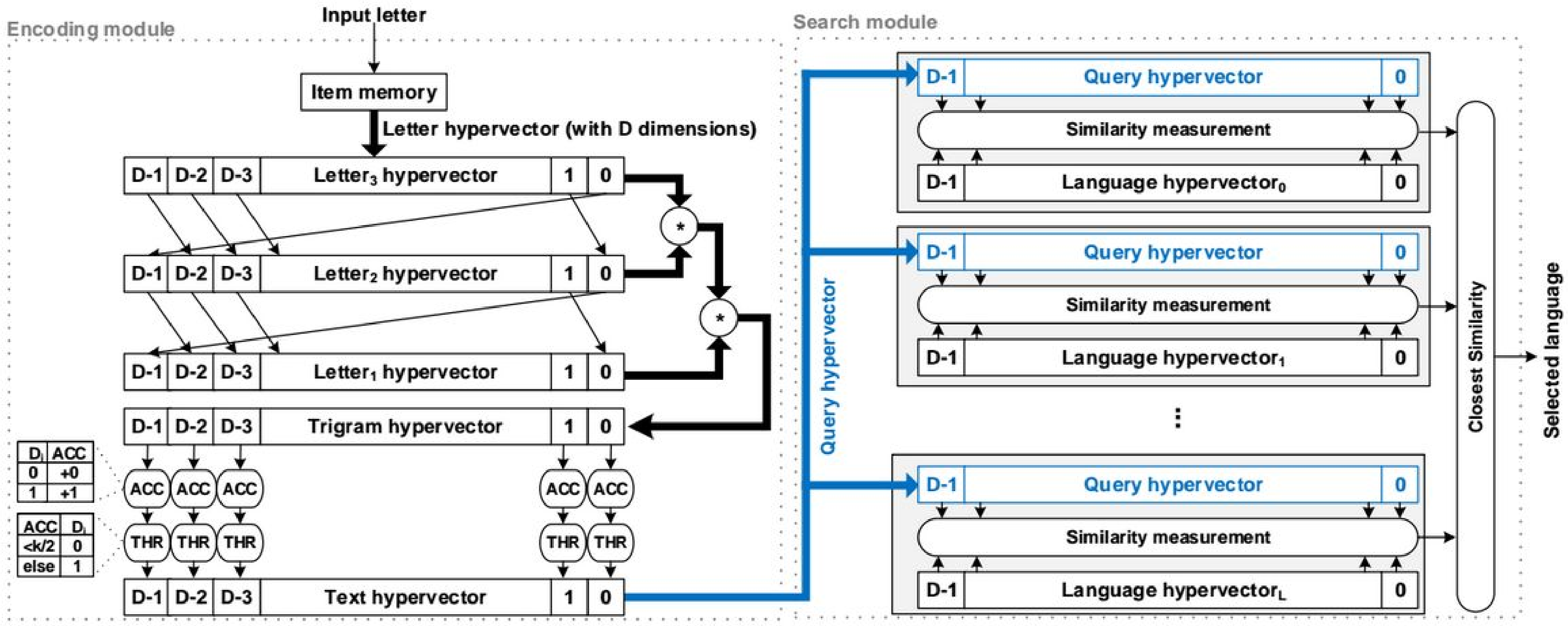

Encoding module
Input letter
Item memory
Letter hypervector (with D dimensions)
D-1 D-2 D-3 Letter₃ hypervector 1 0
D-1 D-2 D-3 Letter₂ hypervector 1 0
D-1 D-2 D-3 Letter₁ hypervector 1 0
D-1 D-2 D-3 Trigram hypervector 1 0
D-1 D-2 D-3 Text hypervector 1 0
ACC
THR
Dᵢ ACC
0 +0
1 +1
ACC Dᵢ
<k/2 0
else 1
Search module
Query hypervector
D-1 Query hypervector 0
Similarity measurement
D-1 Language hypervector₀ 0
D-1 Language hypervector₁ 0
D-1 Language hypervector_L 0
Closest Similarity
Selected language

Figure 1: Memory-centric architecture for hyperdimensional computing: encoding module and search module.

Language Identification System Architecture (Contd.)

- Same architecture for training and testing.
- Encoding Module:
  1. Item memory assigns a unique but random hypervector that is called letter hypervector to an input letter.
  2. Compute a hypervector for a block of N consecutive letters. For example, a window of three letters or trigrams.
  3. Consider a trigram of A-B-C, A hypervector is rotated twice ($\rho\rho A$), B hypervector is rotated once (ρB), no rotation for C hypervector. Pointwise multiplication are then applied between these new hypervectors to compute trigram hypervector, i.e,. $\rho\rho A*\rho B*C$. Multiplication is implemented using XOR gates.
  4. All trigram hypervectors are added to form text hypervector. Some threshold is applied to convert it to binary.

Language Identification System Architecture (Contd.)

- Encoding module is used for both training and testing. During training we refer text hypervector as language hypervector. While testing we call text hypervector as query hypervector.
- Similarity Search Module:
    - It stores a set of language hypervectors that are precomputed by encoding module.
    - We compare query hypervector with language hypervectors over a distributed fashion using an associative memory. We use cosine similarity as a metric.
    - $\text{distance}_{\text{cos}} = (LV_i . QV_i)/ (|LV_i| |QV_i|)$
    - If distance is close to 1 then we identified the language.

References

1. A. Rahimi, P. Kanerva, and J. M. Rabaey, “A robust and energy efficient classifierusing brain-inspired hyperdimensional computing,” in Low Power Electronics andDesign (ISLPED), 2016 IEEE/ACM International Symposium on, August 2016.
2. P. Kanerva, “Hyperdimensional computing: An introduction to computing indistributed representation with high-dimensional random vectors,” Cognitive Com-putation, vol. 1, no. 2, pp. 139–159, 2009.
3. https://github.com/abbas-rahimi/HDC-EMG