Deep Learning for Sensing: Human Activity Recognition

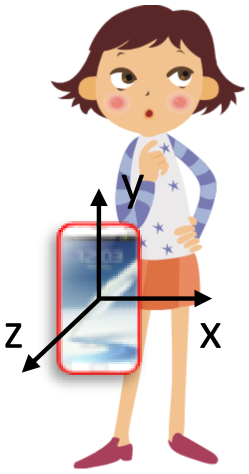
Deep Convolutional and LSTM Recurrent Neural Networks for Multimodal Wearable Activity Recognition

by Francisco Javier Ordóñez * and Daniel Roggen

Wearable Technologies, Sensor Technology Research Centre, University of Sussex, Brighton BN1 9RH, UK

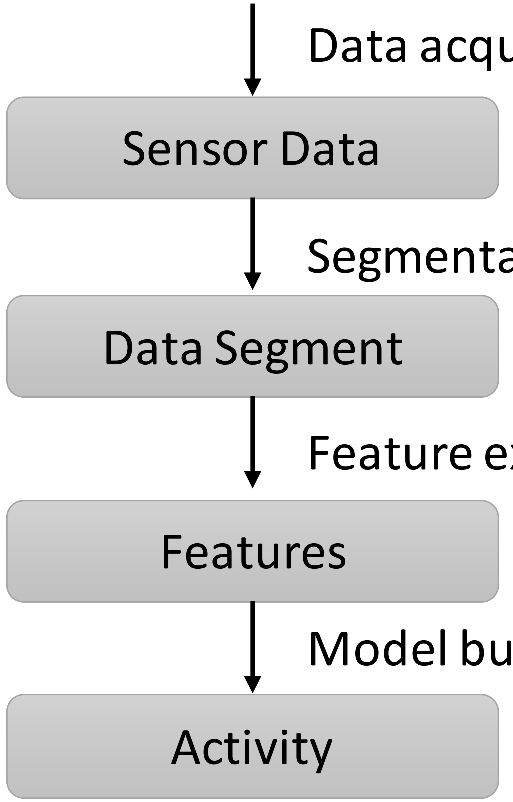
https://www.mdpi.com/1424-8220/16/1/115/htm

Activity Recognition Process



Motion sensors

accelerometer compass gyroscope



Data acquisition and pre-processing

Sensor Data

Segmentation

Data Segment

Feature extraction – mean, variance, ...

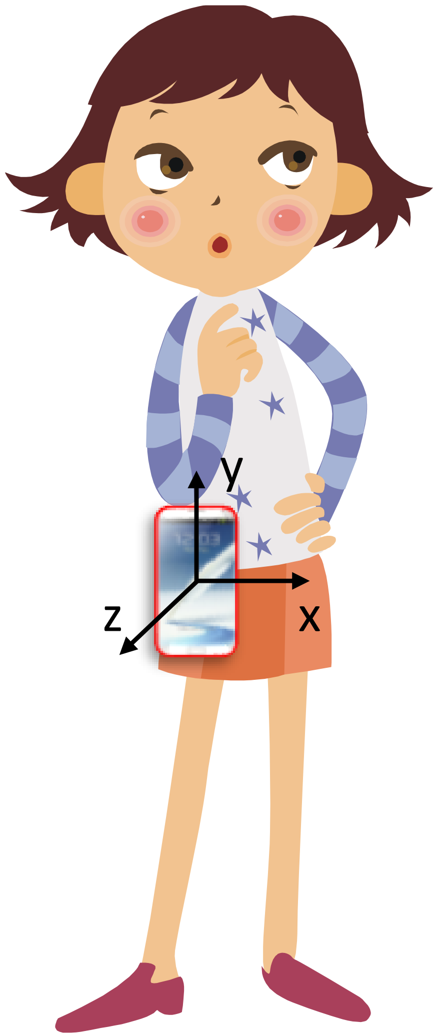
Features

Model building & Classification (Inference)

Activity

Activity Recognition Process

Phone shaken?	Phone orientation?	Current Activity
No	Upright	Standing still
Yes	Upright	Running
No	Lying down	Lying on a bed
Yes	Lying down	Nothing (=Null)



Average value of accelerometer y-axis
sensor signals for the last 2 seconds 2초이상
(if avg. Y-axis >= alpha, it is upright;
otherwise, lying)

Variance of accelerometer sensor signal
for the last 2 seconds
(if variance >= beta, it is shaken)

Activity Recognition Process

Phone shaken?	**Phone orientation**	**Activity**
No	Upright	Standing still
Yes	Upright	Running
No	Lying down	Lying on a bed
Yes	Lying down	…?

Classification

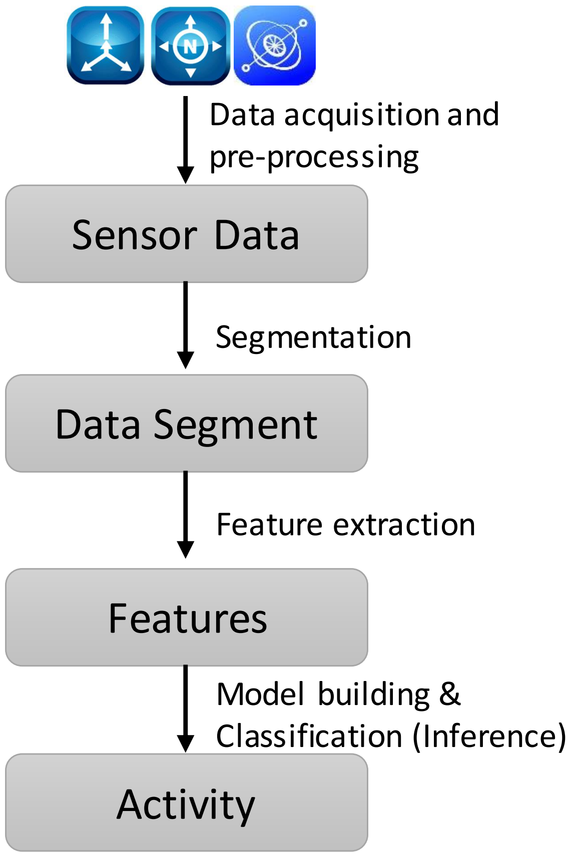
Average value of accelerometer y-axis sensor signals for the last 2 seconds

window size: 2sec.

Feature extraction

Segmenting (=windowing)

Variance of accelerometer sensor signal for the last 2 seconds



Data acquisition and pre-processing
Sensor Data
Segmentation
Data Segment
Feature extraction
Features
Model building & Classification (Inference)
Activity

Data Acquisition & Pre-processing

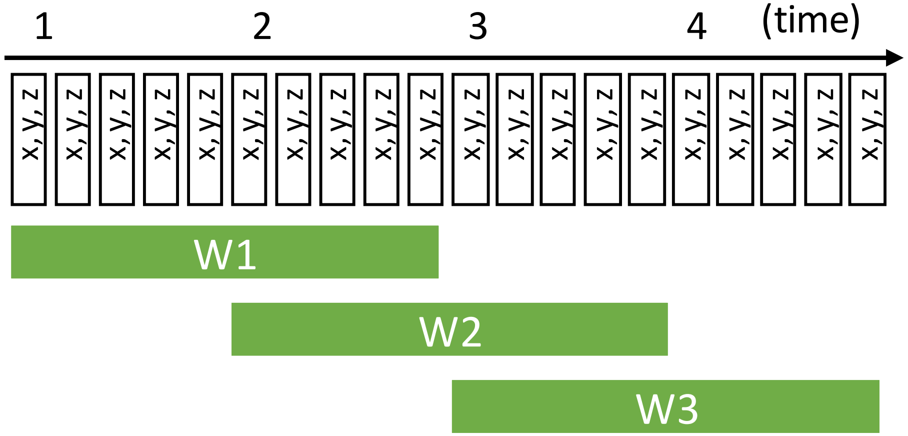
- Collecting a stream of sensor data (e.g., using Android’s sensor manager interface)
- Since most sensors provide data on some regular basis, we also need to know **sampling rate** (will learn more about this during DSP sessions) ↳ max : data ↑, 시간, 배터리 ↑
- An accelerometer, for example, may provide a stream of tuples of real numbers representing the acceleration in x, y and z-direction with 5 Hz

Data Segmentation

- For feature extraction, we need to "identify" those data segments that are likely to contain information about activities (known as "activity detection" or "spotting)
- **Sliding window**: using a window (=frame) of samples, and simply slide that window with fixed overlapping (e.g., 50%) In our example, to recognize basic physical activities we collect the data of 2 seconds from the accelerometer.
- This corresponds to 10 readings of the acceleration data (if sampling rate is fixed to 5Hz)

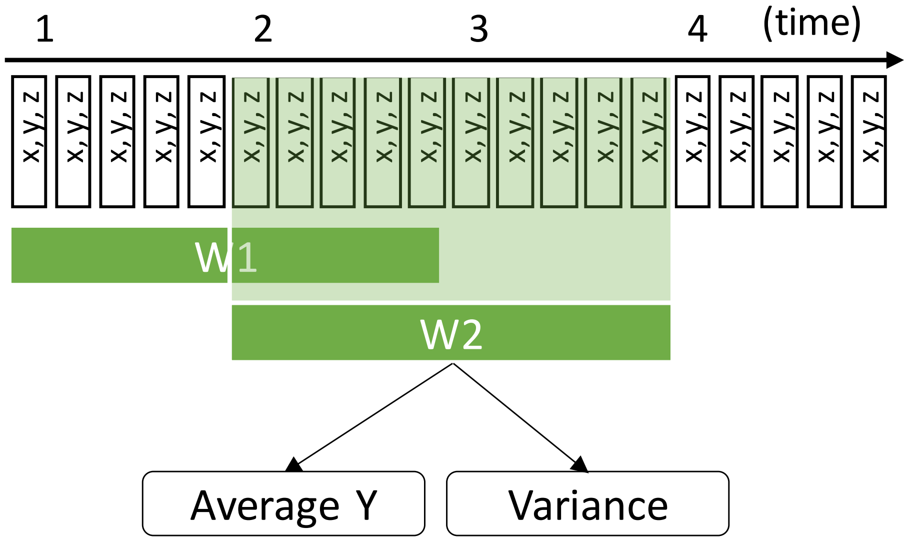
Example

Segmenting the sensor data using a window of 2 seconds with an overlap of 1 second



Feature Extraction

- Signal-based features – mean, variance, kurtosis
- Body model features – exploiting prior knowledge about human kinematics
- Event-based features – if there are any events (e.g., a sequence of eye movements – saccades, fixations, or blinks)
- Multilevel features – duration, frequency, co-occurrence, clustered data/labels



Classification

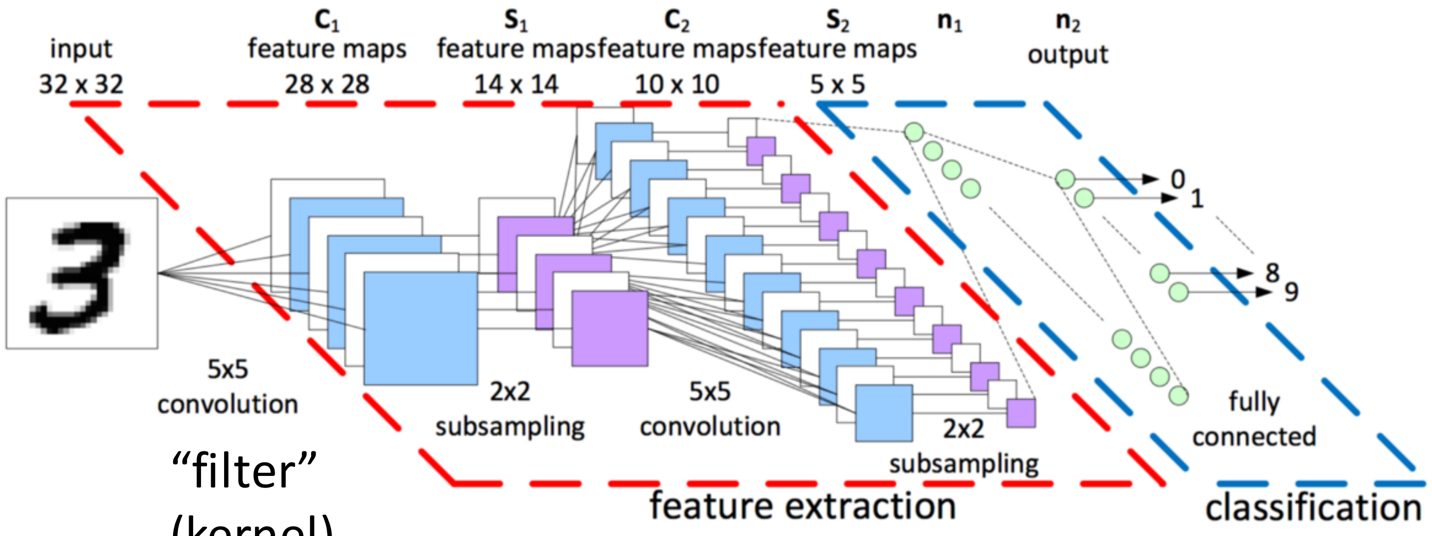
- decision / probability

- After extracting important features from the raw data, we use a classifier to determine the current activity
- Many different classification algorithms exist, and depending on the application domain, there may be one algorithm that shows the best performance
- Depending on the algorithm, the result is either a crisp decision (e.g., decision tree), or a probability distribution over activities (e.g., Naïve Bayes)
- Learning algorithms: supervised vs. unsupervised (based on whether training dataset is used or not)

Feature Engineering for HAR

- Most recognition systems select features from a pool of engineered features
- Identifying relevant features requires significant amount of time
- Difficulty to scale up activity recognition to complex high level behaviors (e.g., time, activity, device, individual diversity and variability)
- Engineered features failed to relate to “units of behavior” instead of convenient math operations
- Statistical and frequency features do not relate to semantically meaningful aspects of human motion, such as hand grasp

Feature Extraction vs. Classification

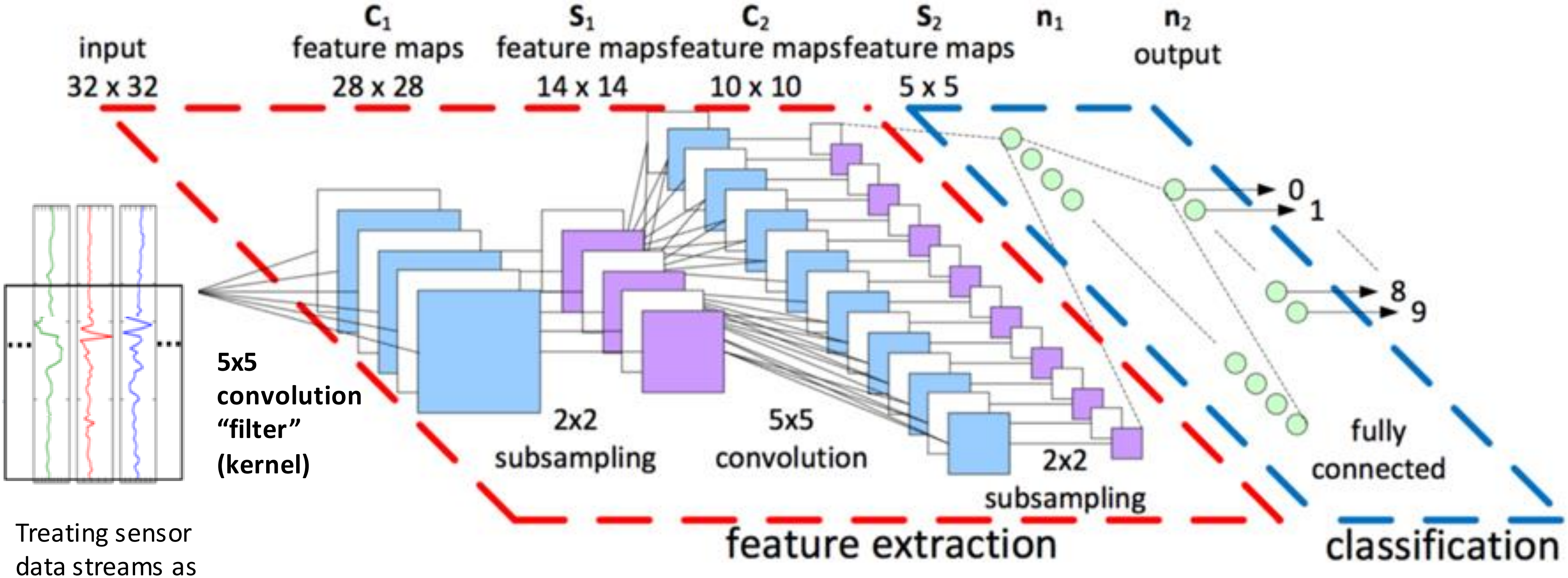


“filter”
(kernel)

- Convolutional layer extracts a feature map from the input signal through a convolution operation of the signal with a filter (or kernel)
- Convolutional operations aim to detect patterns captured by the kernels, regardless of where the pattern occurs
- The kernels are optimized as part of the supervised training process, in an attempt to maximize the activation level of kernels for subsets of classes

https://medium.com/analytics-vidhya/convolutional-neural-networks-cnn-explained-step-by-step-69137a54e5e7

Feature Extraction vs. Classification

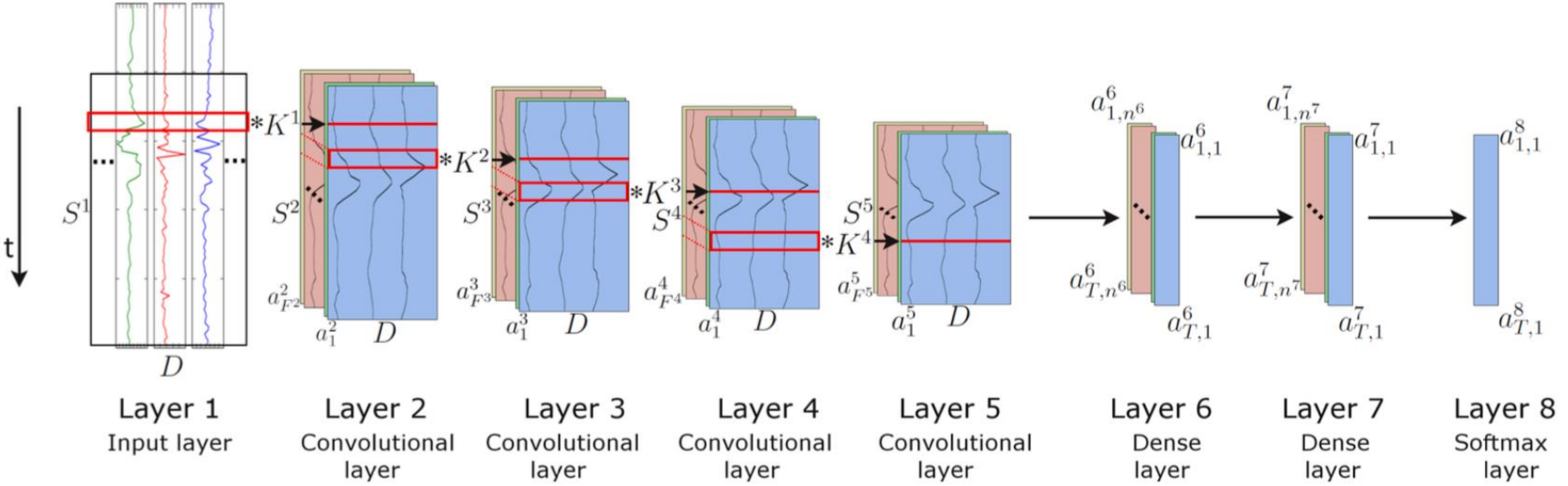


5x5 convolution “filter” (kernel)

Treating sensor data streams as 2D image data

A convolution kernel can be viewed as a filter, capable of removing outliers, filtering the data or acting as a feature detector, defined to respond maximally to specific temporal sequences within the timespan of the kernel

ConvLSTM: An Early Fusion Model



Convolutional Layer

Dense Layers

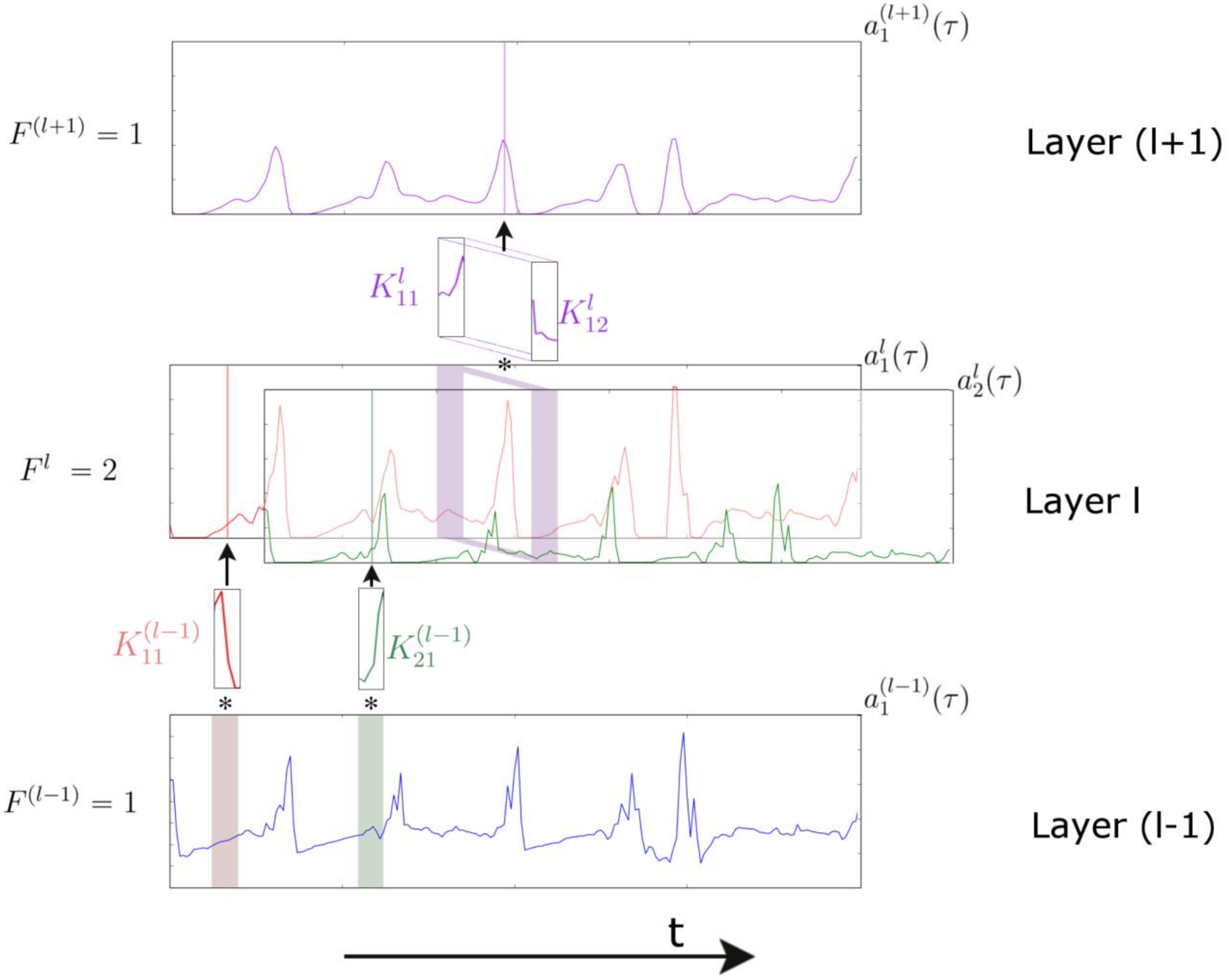
Softmax

Two options for dense layers for comparison:

Regular dense layers vs. LSTMs

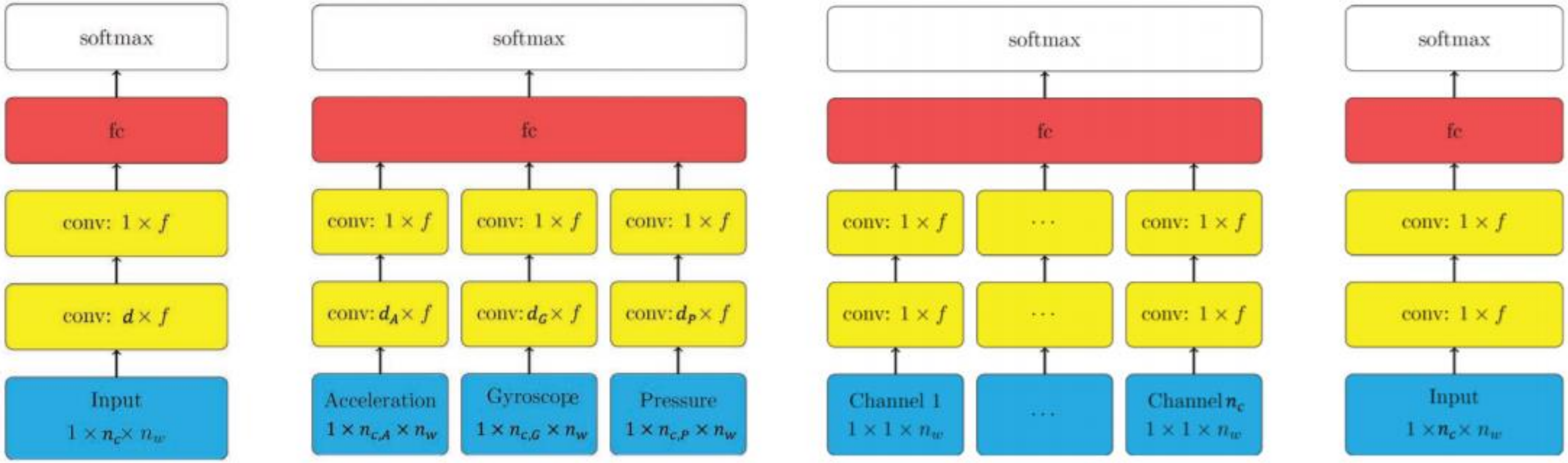
Layer	DeepConvLSTM		Baseline CNN	
	Size Per Parameter	**Size Per Layer**	**Size Per Parameter**	**Size Per Layer**
2	*K*: 64×5	384	*K*: 64×5	384
	$\mathbf{b}$: 64		$\mathbf{b}$: 64	
3–5	*K*: $64 \times 64 \times 5$	20,544	*K*: $64 \times 64 \times 5$	20,544
	$\mathbf{b}$: 64		$\mathbf{b}$: 64	
6	$W_{ai}, W_{af}, W_{ac}, W_{ao}$: 7232×128	942,592	*W*: $57,856 \times 128$	7,405,696
	$W_{hi}, W_{hf}, W_{hc}, W_{ho}$: 128×128		$\mathbf{b}$: 128	
	$\mathbf{b}_i, \mathbf{b}_f, \mathbf{b}_c, \mathbf{b}_o$: 128			
	W_{ci}, W_{cf}, W_{co}: 128			
	$\mathbf{c}$: 128			
	$\mathbf{h}$: 128			
7	$W_{ai}, W_{af}, W_{ac}, W_{ao}$: 128×128	33,280	*W*: 128×128	16,512
	$W_{hi}, W_{hf}, W_{hc}, W_{ho}$: 128×128		$\mathbf{b}$: 128	
	$\mathbf{b}_i, \mathbf{b}_f, \mathbf{b}_c, \mathbf{b}_o$: 128			
	W_{ci}, W_{cf}, W_{co}: 128			
	$\mathbf{c}$: 128			
	$\mathbf{h}$: 128			
8	*W*: $128 \times n_c$	$(128 \times n_c) + n_c$	*W*: $128 \times n_c$	$(128 \times n_c) + n_c$
	$\mathbf{b}$: n_c		$\mathbf{b}$: n_c	
Total		**996,800** $+ (128 \times n_c) + n_c$		$7,443,136 + (128 \times n_c) + n_c$

ConvLSTM: 1D Convolution



Use two filters (K_{11} and K_{21}) for convolution

Review: Sensor Fusion Strategies



(a) Early fusion

(b) Sensor-based late fusion

(c) Channel-based late fusion

(d) Shared filters hybrid fusion

Early fusion (EF)

$d = n_c$

Sensor-based late fusion (SB-LF)

$d_A = n_{c,A} = 3$
$d_G = n_{c,G} = 3$
$d_P = n_{c,P} = 1$

Channel-based late fusion (CB-LF)

Shared filters hybrid fusion (SF-HF)

*It's CB-LF, but it **uses the same filters** are used for all input channels (best performing)*

CNN-based sensor fusion techniques for multimodal human activity recognition, ISWC 2017

ConvLSTM: Model Performance

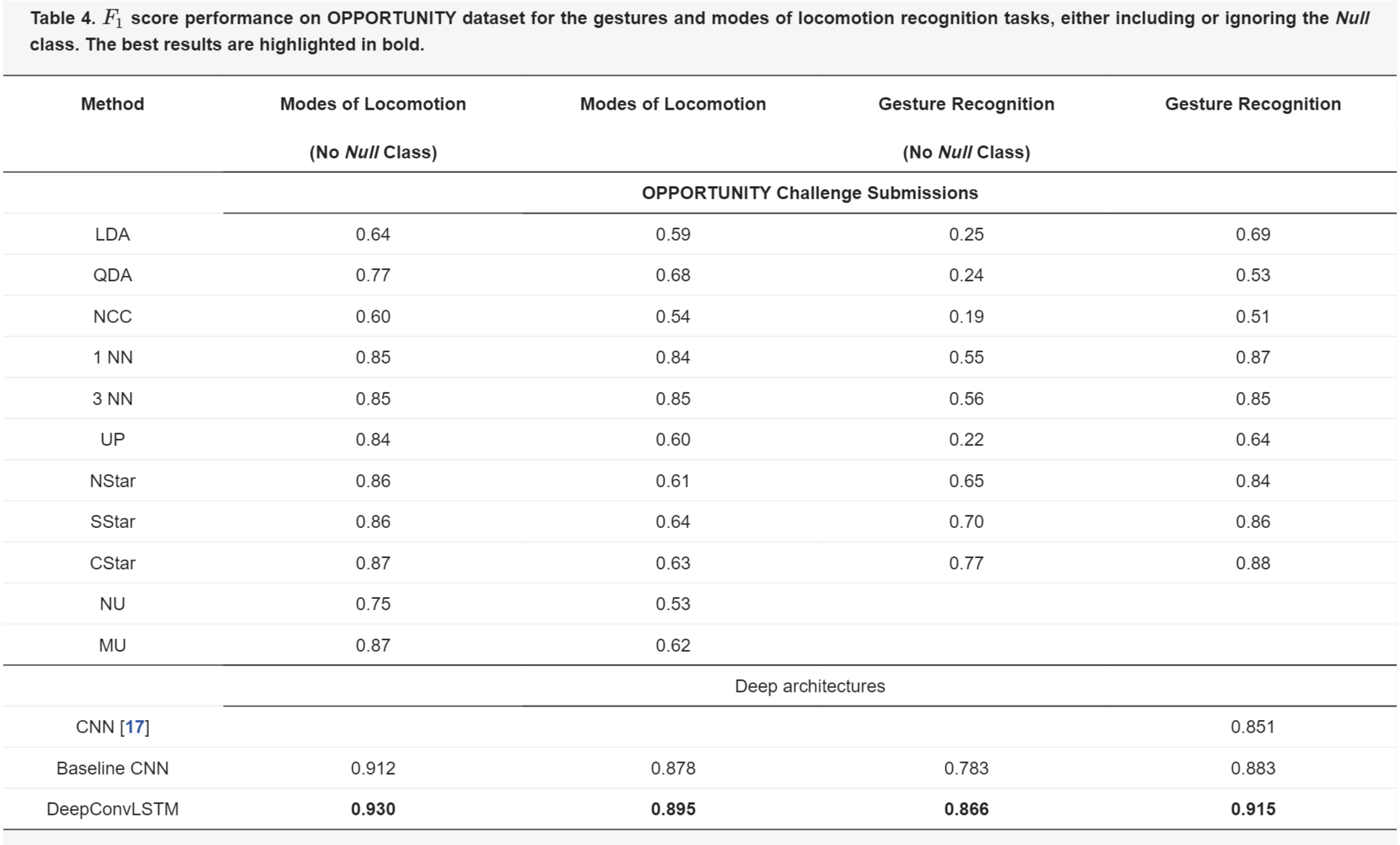
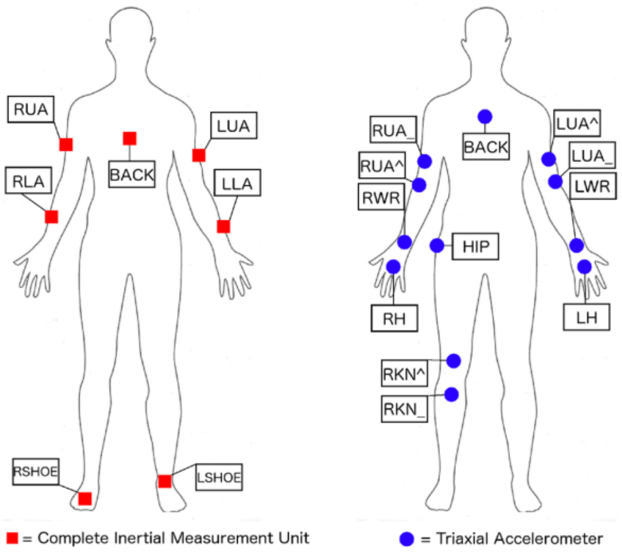


Table 4. F_1 **score performance on OPPORTUNITY dataset for the gestures and modes of locomotion recognition tasks, either including or ignoring the *Null* class. The best results are highlighted in bold.**

Method	Modes of Locomotion	Modes of Locomotion	Gesture Recognition	Gesture Recognition
	(No *Null* Class)		(No *Null* Class)	
	OPPORTUNITY Challenge Submissions			
LDA	0.64	0.59	0.25	0.69
QDA	0.77	0.68	0.24	0.53
NCC	0.60	0.54	0.19	0.51
1 NN	0.85	0.84	0.55	0.87
3 NN	0.85	0.85	0.56	0.85
UP	0.84	0.60	0.22	0.64
NStar	0.86	0.61	0.65	0.84
SStar	0.86	0.64	0.70	0.86
CStar	0.87	0.63	0.77	0.88
NU	0.75	0.53		
MU	0.87	0.62		
	Deep architectures			
CNN [17]				0.851
Baseline CNN	0.912	0.878	0.783	0.883
DeepConvLSTM	**0.930**	**0.895**	**0.866**	**0.915**

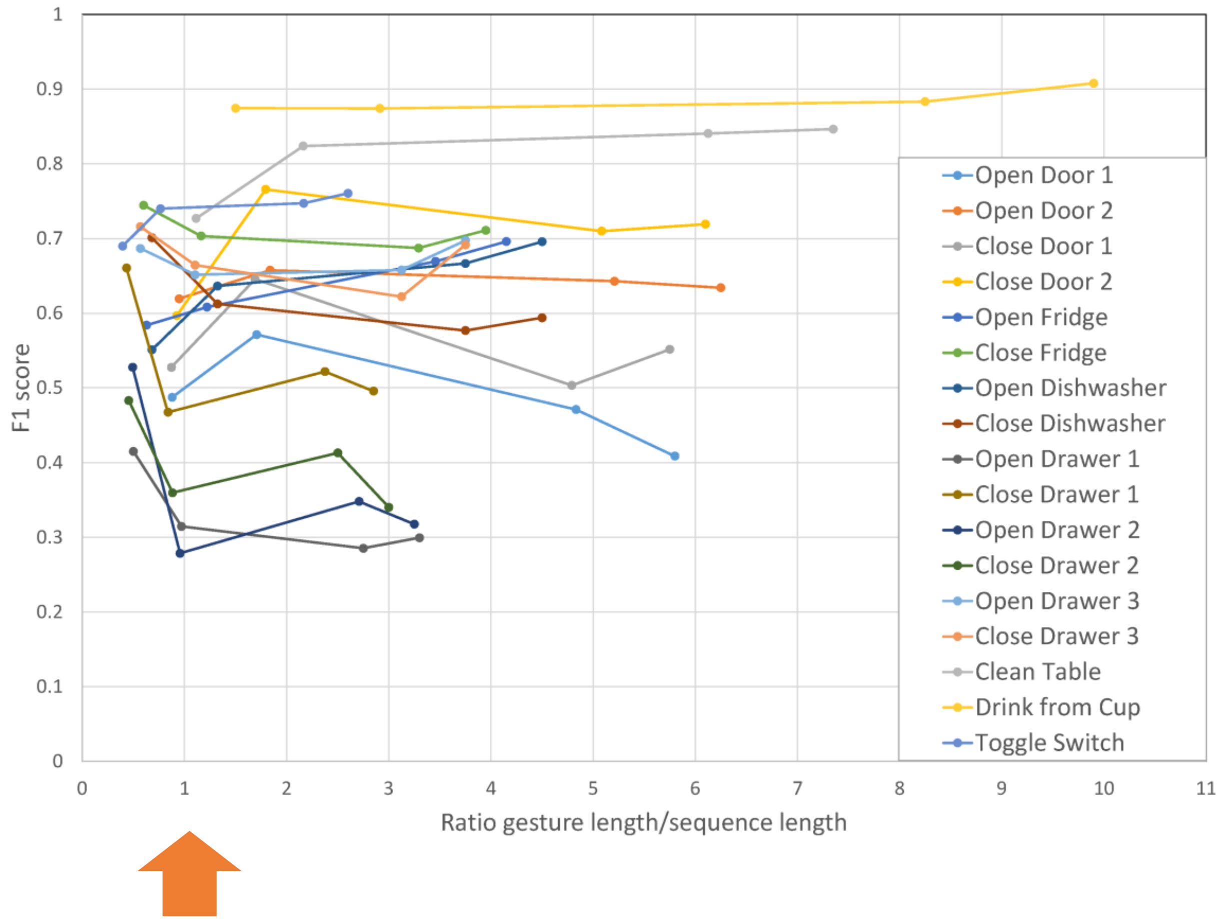
Data Source Variation



	Accel	**Gyro**	**Accel + Gyro**	**Accel+ Gyro+ Mag**	**Entire Opportunity Sensors Set**
# of sensors channels*	15	15	30	45	113
F1 score	0.689	0.611	0.745	0.839	0.864

* Each channel is a single stream of data (e.g., accel X-axis)

Optimal Sequence Length?



- Ratios under one represent performance for <u>gestures whose durations are shorter than the sequence duration</u> and, thus, that can be fully observed by the network before it provides an output prediction.
- longer gestures (as “clean table” or “drink from cup” in this dataset) may be made of several shorter characteristic patterns

F1 score performance of DeepConvLSTM on the OPPORTUNITY dataset. Classification performance is displayed individually per gesture, for different lengths of the input sensor data segments. Experiments carried out with sequences of length of 400 ms, 500 ms, 1400 ms and 2750 ms. The horizontal axis represents the ratio between the gesture length and the sequence length (ratios under one represent performance for gestures whose durations are shorter than the sequence duration).

Optimal Sequence Length?

- Currently, the input of the recurrent model is composed of a 500-ms data sequence
- Therefore, the gradient signal is unable to notice time dependencies longer than the length of this sequence
- Evaluate the influence of this parameter in the recognition performance of gestures with different durations, in particular if the gestures are significantly longer or shorter than the sequence duration

Optimal # Convolutional Layers?

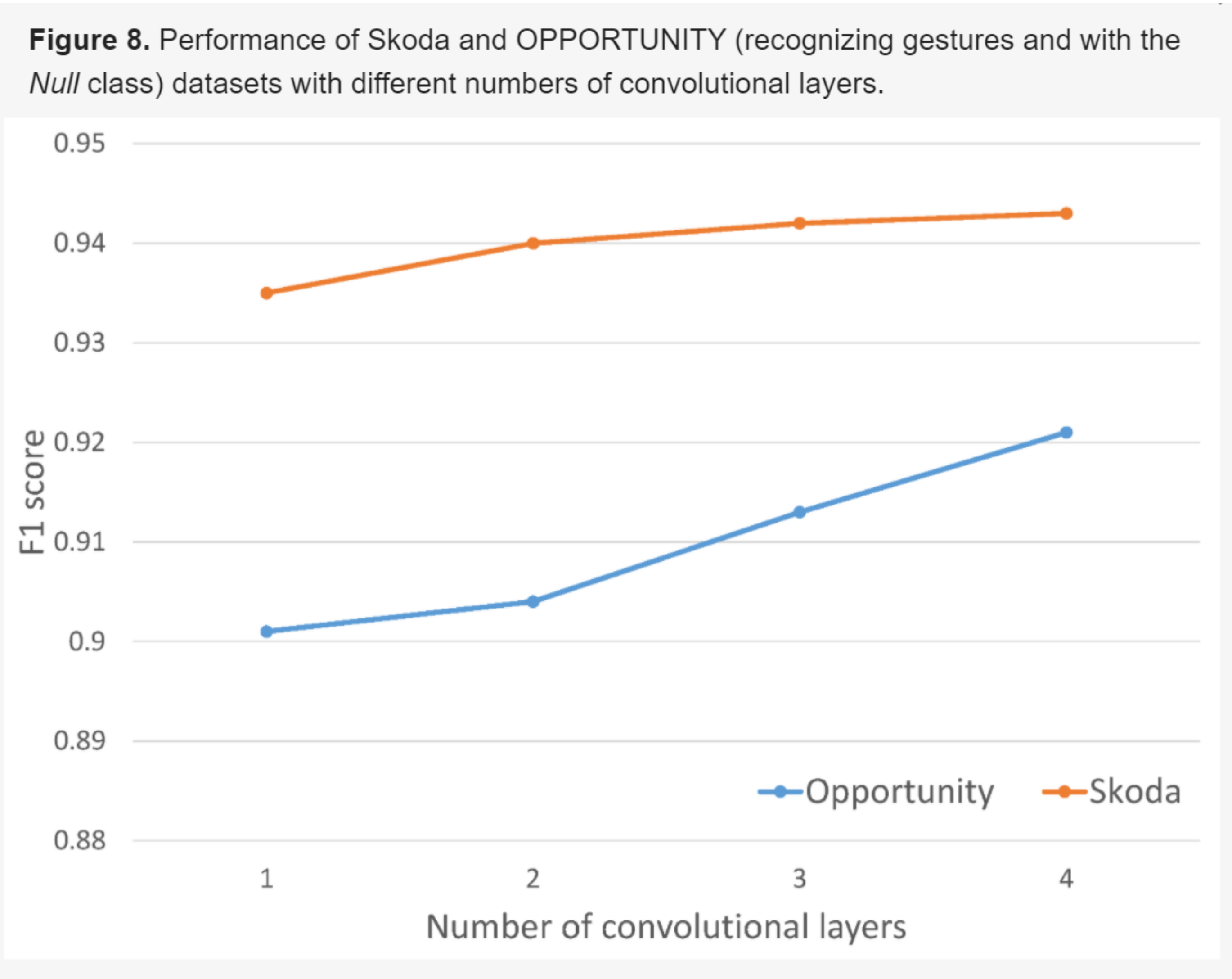
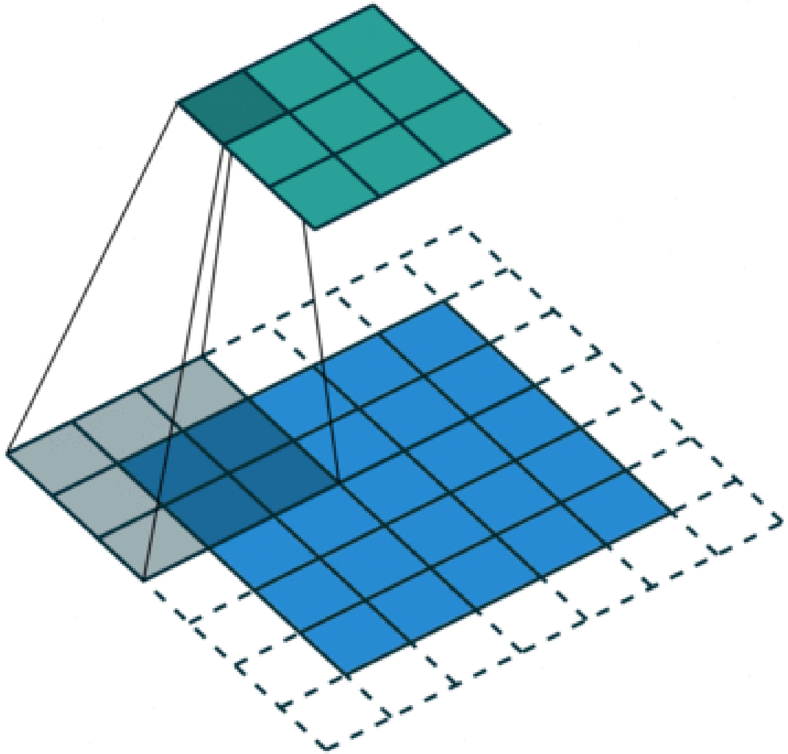
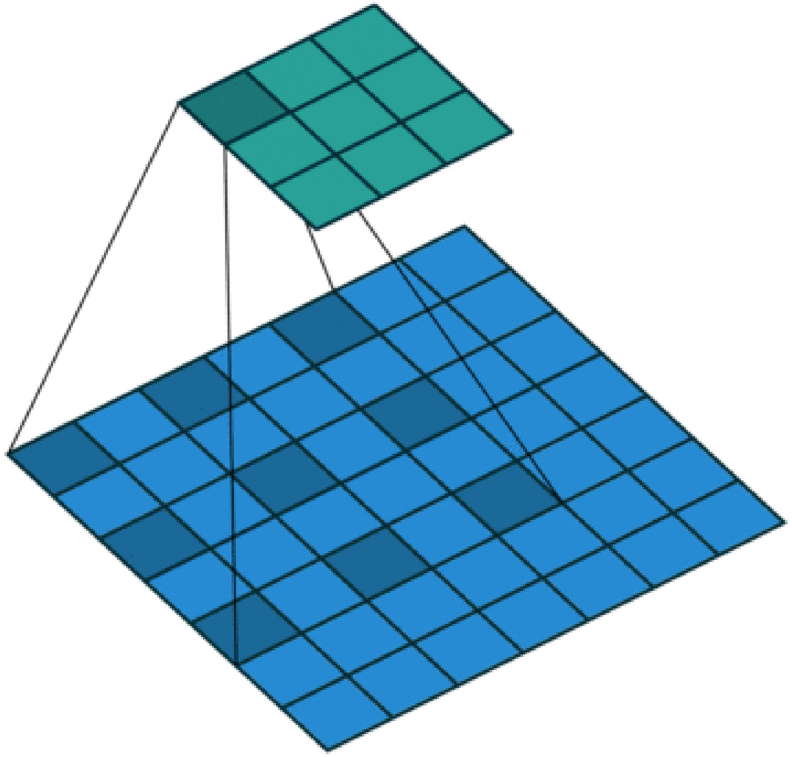


Figure 8. Performance of Skoda and OPPORTUNITY (recognizing gestures and with the *Null* class) datasets with different numbers of convolutional layers.

Dilated Convolution



Standard Convolution (l=1)



Dilated Convolution (l=2)

Multi-scale context aggregation by dilated convolutions,
Vladlen Koltun, Fisher Yo, ICLR 2016

UCI HAPT Dataset

1 : WALKING
2 : WALKING_UPSTAIRS
3 : WALKING_DOWNSTAIRS
4 : SITTING
5 : STANDING
6 : LAYING

- Data are separated by participants and each file contains raw triaxial signals (i.e., x, y, z) from the accelerometer or the gyroscope.
- There are two data (e.g., experiment #1 ~ experiment #3) for each participant.
- Sampling frequency of the accelerometer and the gyroscope is 50 Hz. In other words, samples are collected in every 20 milliseconds (1/50sec = 1000/50ms).
- This dataset was collected from a smartphone on the waist while participants were performing 6 different activities.
- The activities and the corresponding label number are as follows:

https://www.youtube.com/watch?v=XOEN9W05_4A