SENSE MAKING OF SENSOR DATA

Dr TIAN Jing

tianjing@nus.edu.sg

Module objective

Module: Making sense of sensor data

Knowledge and understanding

- Understand the fundamentals of sense making pipeline of both single type of sensor data and multiple types of sensor data

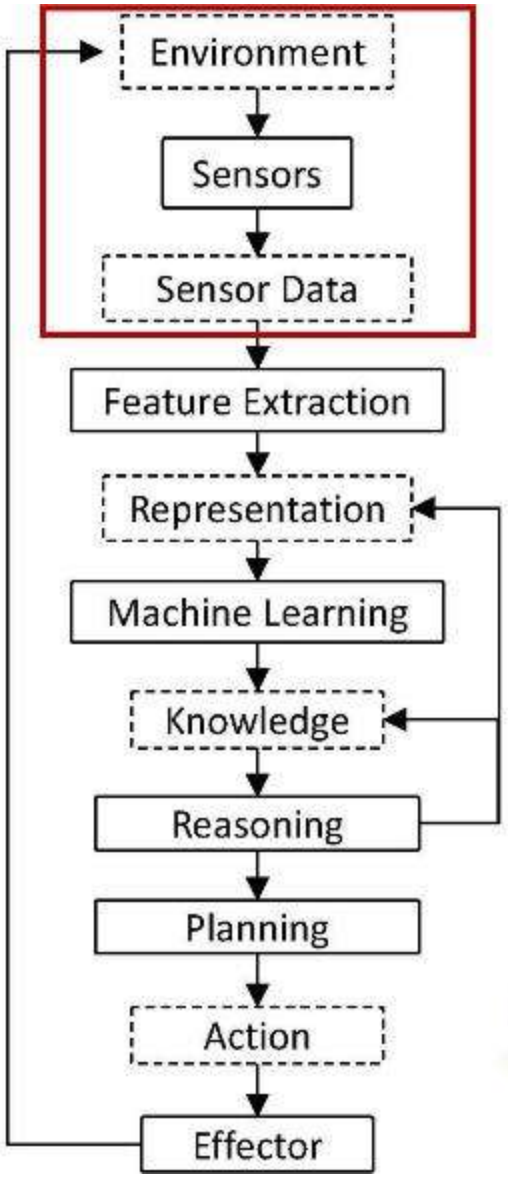
Key skills

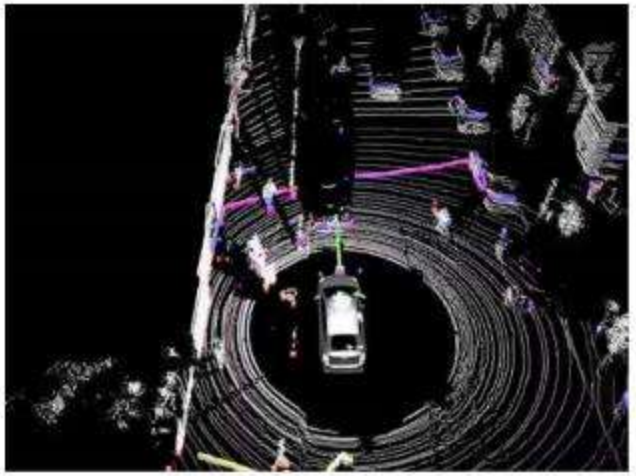
- Design, build, implement intelligent sensing system for real-world applications

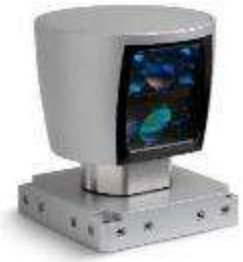
Major reference

- [Introduction] O. Tickoo and R. Iyer, ***Making Sense of Sensors: End-to-End Algorithms and Infrastructure Design from Wearable-Devices to Data Centers***, Apress, 2016.
- [Intermediate] L. A. Klein, ***Sensor and Data Fusion: A Tool for Information Assessment and Decision Making***, SPIE Press, 2012.
- [Advanced] T. Giannakopoulos, ***Multimodal Information Processing & Analysis***, https://github.com/tyiannak/multimodalAnalysis
- [Advanced] ***Advanced Multimodal Machine Learning***, CMU, https://piazza.com/cmu/spring2017/11777/home

Overview of intelligent sensing system (1)







Lidar

Camera
(Visible, Infrared)

Radar

GPS

Stereo Camera

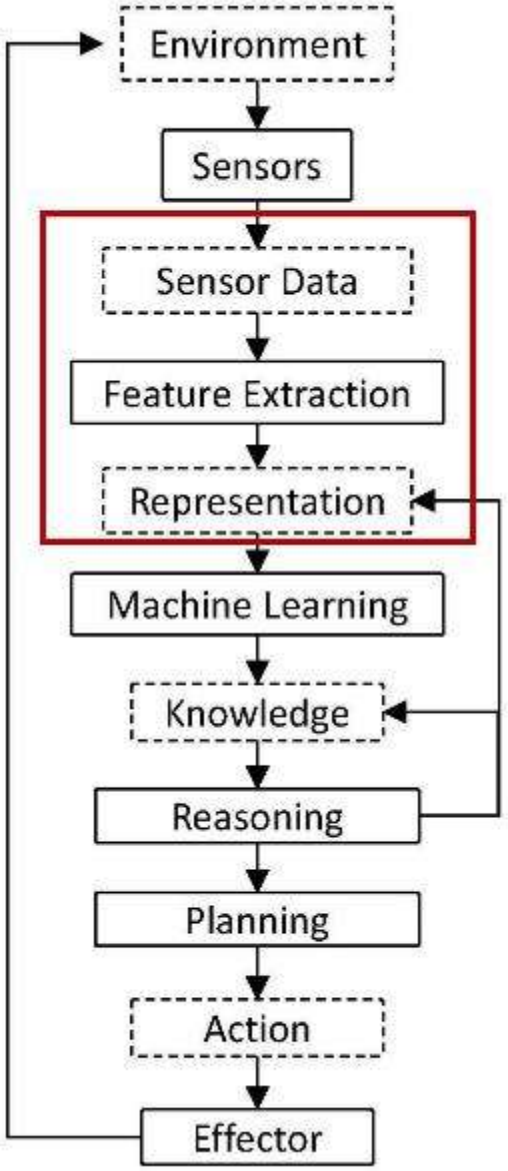
Microphone

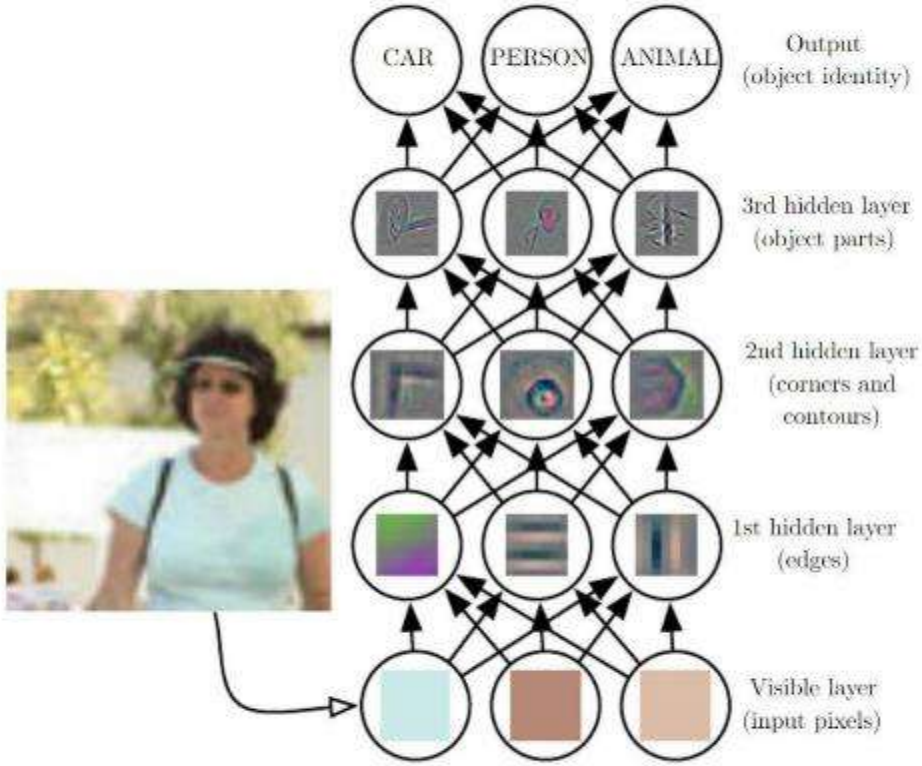
Networking
(Wired, Wireless)

IMU

Source: MIT 6.S099: Artificial General Intelligence, https://agi.mit.edu/

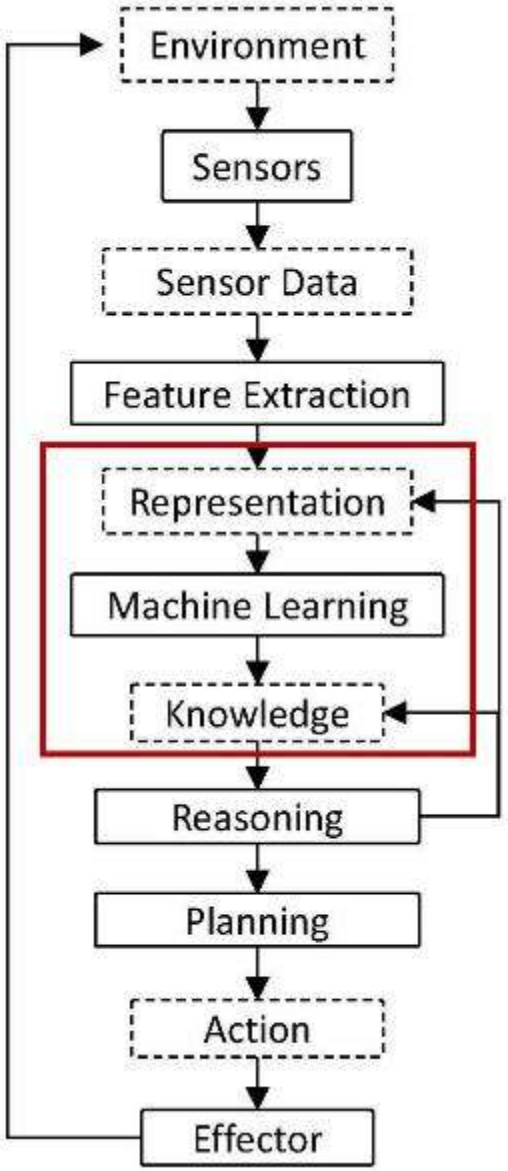
Overview of intelligent sensing system (2)

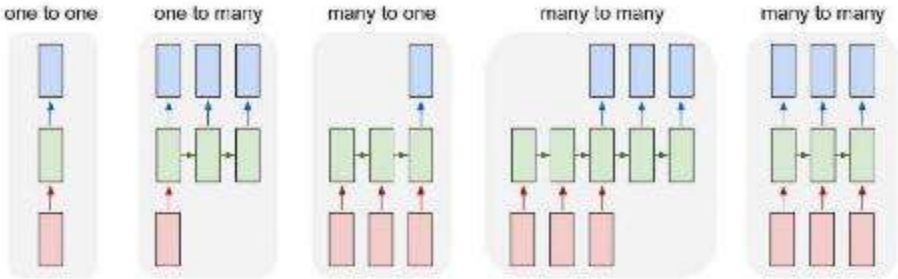




Source: MIT 6.S099: Artificial General Intelligence, https://agi.mit.edu/

Overview of intelligent sensing system (3)





Source: MIT 6.S099: Artificial General Intelligence, https://agi.mit.edu/

Overview of intelligent sensing system (4)

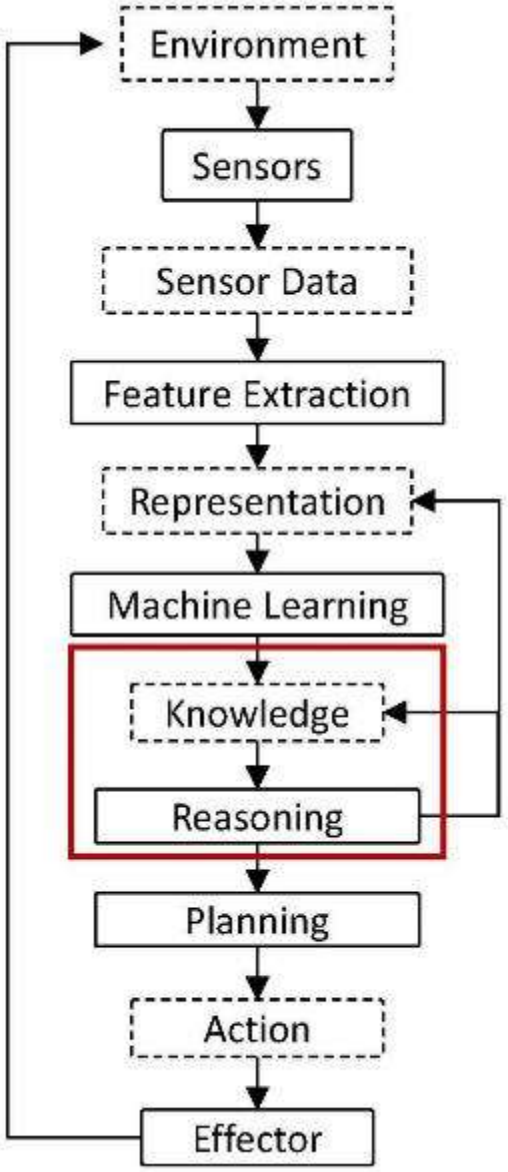


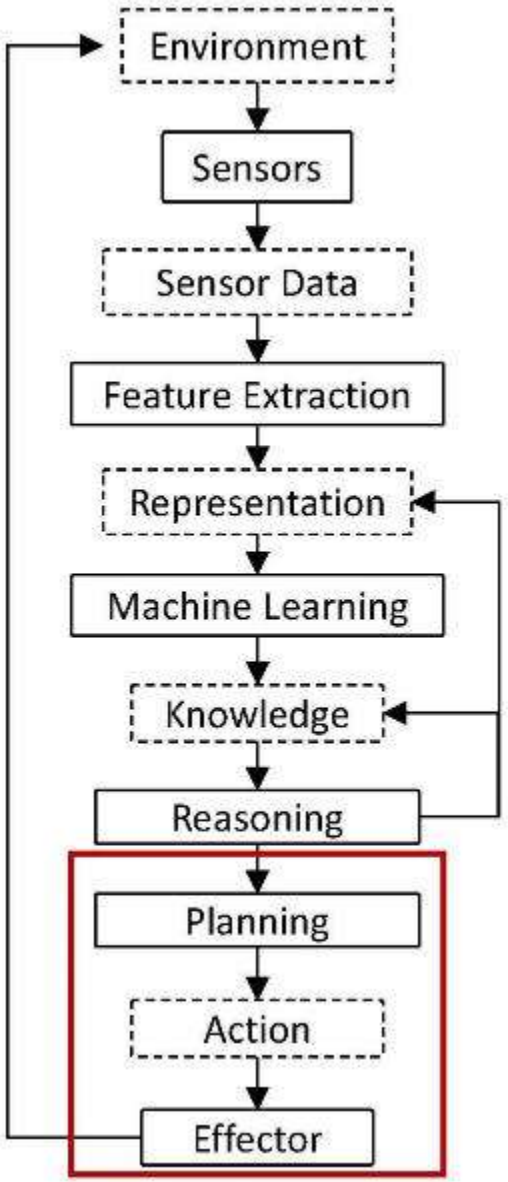
Image Recognition:
If it looks like a duck

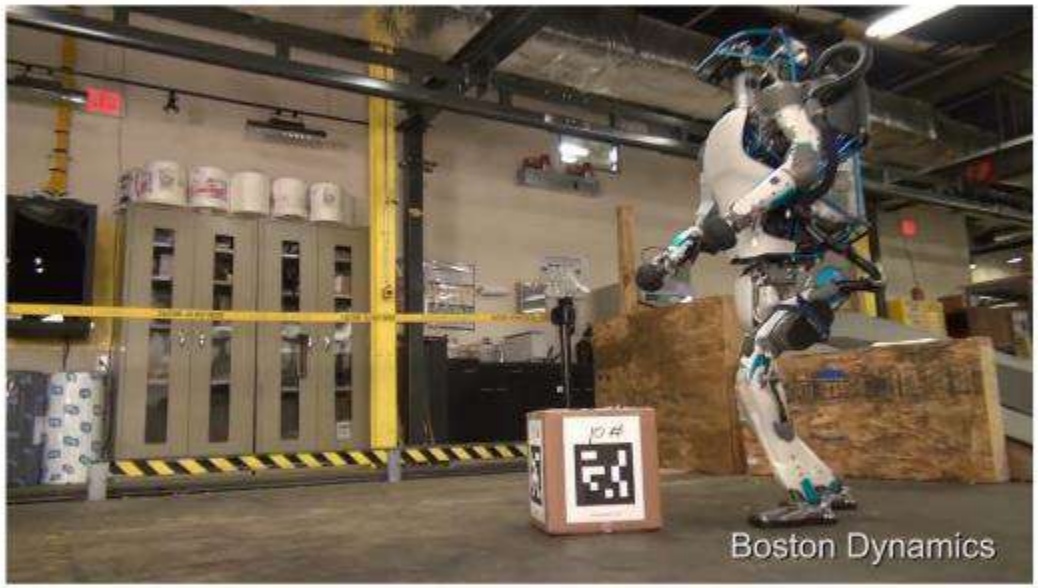
Audio Recognition:
Quacks like a duck

Activity Recognition:
Swims like a duck

Source: MIT 6.S099: Artificial General Intelligence, https://agi.mit.edu/

Overview of intelligent sensing system (5)



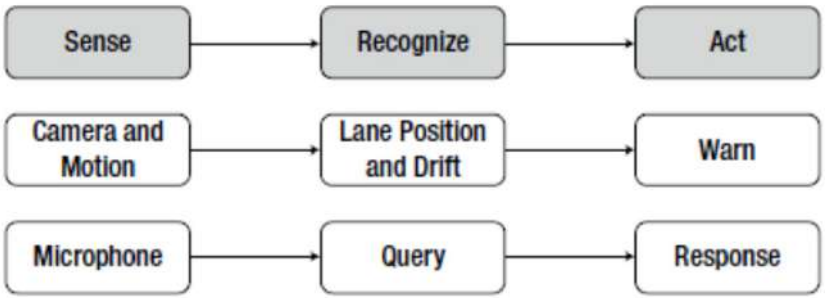


Source: MIT 6.S099: Artificial General Intelligence, https://agi.mit.edu/

An example

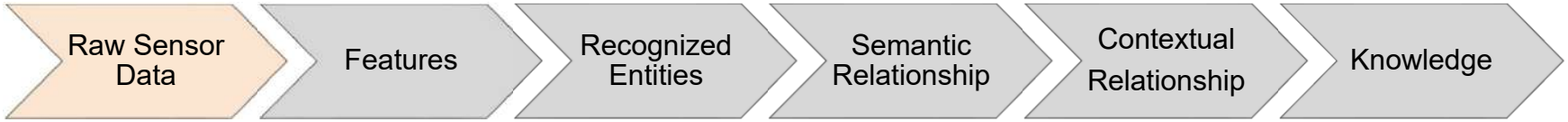
Example: Automatic Driver Assistance System (ADAS)

- It uses a set of sensors like cameras, motion sensors, etc. to provide a driver with assistance for safety and comfort, such as tracking the position of a car in the traffic lanes and provide audible warnings in case the car drifts from its lane (Lane Departure Warning).
- It works on the fundamental processing principles of *Internet of Things* (IoT) comprising of sensors collecting data (cameras, motion sensors), algorithms recognizing the data (relative position of the car in the lane, car drift), and applications acting on the recognized data (audible warning on/off).



Source: O. Tickoo and R. Iyer, Making Sense of Sensors: End-to-End Algorithms and Infrastructure Design from Wearable-Devices to Data Centers, Apress, 2016.

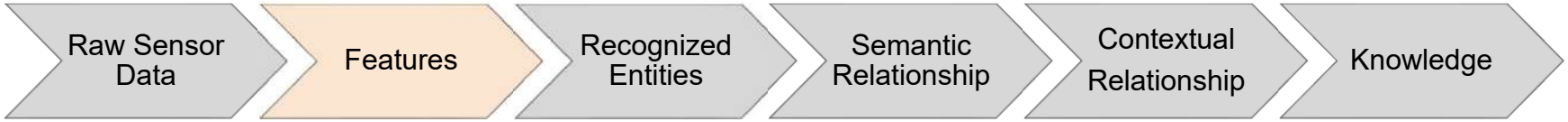
Making sense of sensor data pipeline (1)



Raw Sensor Data

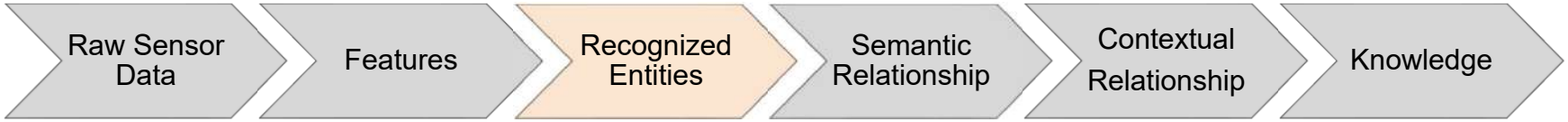
- Raw (or unprocessed) sensor data is captured by sensors at the front end of the pipeline
- **Vision**: Camera sensors take still or video shots of a scene
- ***Audio/Speech****:* The microphone-like sensors typically capture the audio and analog-to-digital conversion is applied
- **Motion:** Inertial sensors typically measure the acceleration and motion
- **BMI sensors**: The Brain Machine Interface sensors, like EEG, report the brain activity measurements as activity graphs
- **Other IoT sensors**: temperature, humidity, etc.

Making sense of sensor data pipeline (2)



Category	Representative features
Geometrical	Edges, lines, line widths, line relationships (e.g., parallel, perpendicular), circles, shapes, size of enclosed area
Structural	Surface area; relative orientation; orientation in vertical and horizontal ground plane;
Statistical	Number of surfaces, area and perimeter, moments, Fourier descriptors, mean, variance, kurtosis, skewness, entropy
Spectral	Color coefficients, spectral peaks and lines
Time domain	Pulse characteristics (rise and fall times, amplitude), pulse width, pulse repetition interval, moments
Frequency domain	Fourier coefficients, time-frequency domain feature such as Wavelet

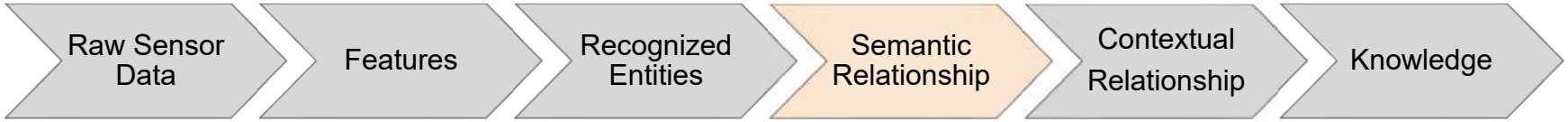
Making sense of sensor data pipeline (3)



Recognized Entities

- Complex task involving spatial and temporal analysis of the extracted features to map the aggregate to pre-known entities.
- For vision, the recognition task may involve classifying the extracted features to recognize shapes like objects and faces
- For audio, statistical analysis and classification of features allows for aggregated features to be recognized as words

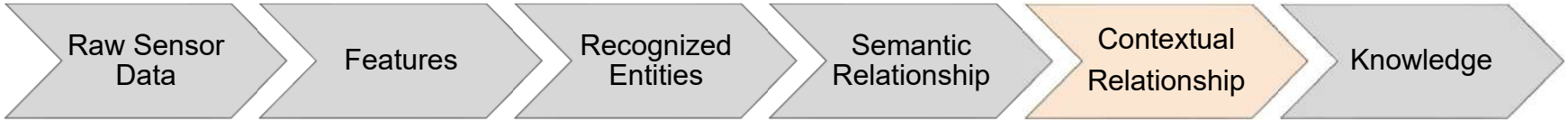
Making sense of sensor data pipeline (4)



Semantic Relationship

- The connection between various recognized entities, for example the semantic relationship between a key and a lock refers to the operation of opening or closing the lock with the key
- For audio, the sentence "*I like coffee*" connects "I" (the subject) with "coffee" (object) using the predicate "like."
- For vision: semantic context of the video can be analyzed based on identified entities (multiple objects, locations, people, and activities) and scoring these entities with respect to their co-occurrence as well as relation to the type of classification scenario of importance.

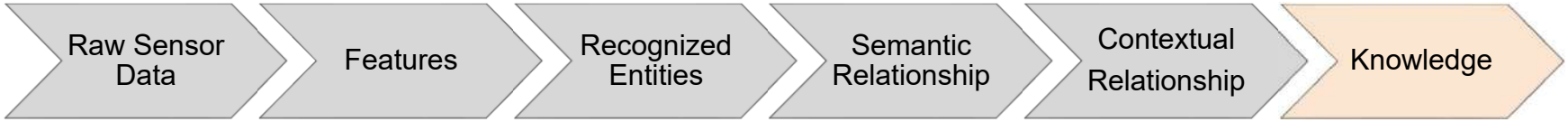
Making sense of sensor data pipeline (5)



Contextual Relationship

- Understand the recognized data in context. The context can be obtained in multiple ways. These include using other sensors as well as using the history of recognized data for context recognition.
- For vision: A face recognition surveillance system can identify a potential invasion or normal operation depending on when the recognition takes place (day vs. night) or depending on the person that is recognized.

Making sense of sensor data pipeline (6)



Knowledge

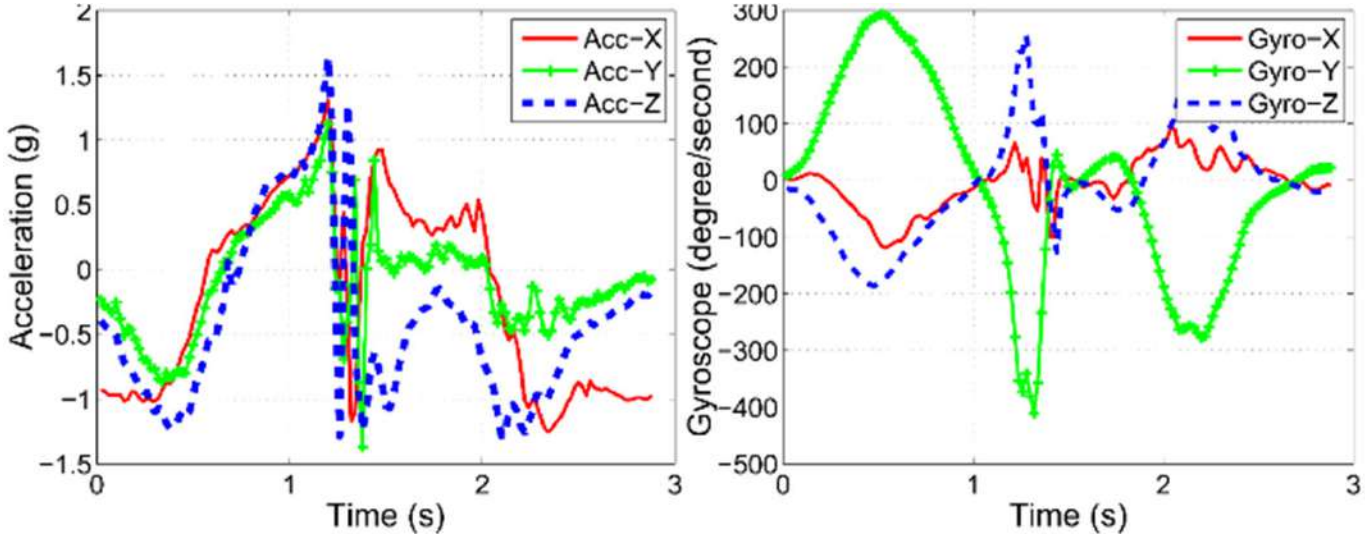
- Knowledge representation: Knowledge needs to be stored and represented in a manner such that the semantic information and relationships between various concepts is retained and is modifiable.
- Knowledge operations: These include tools, APIs, and programming languages to insert and extract data from the knowledge database. This involves understanding the incoming semantic data and providing a mechanism to find and manipulate the correct relationships. The extraction process involves responding to different queries targeting relationships between different entities and concepts.

Single type of sensor data: Inertial

- An accelerometer essentially measures the force (proper acceleration) along x, y and z-axes.

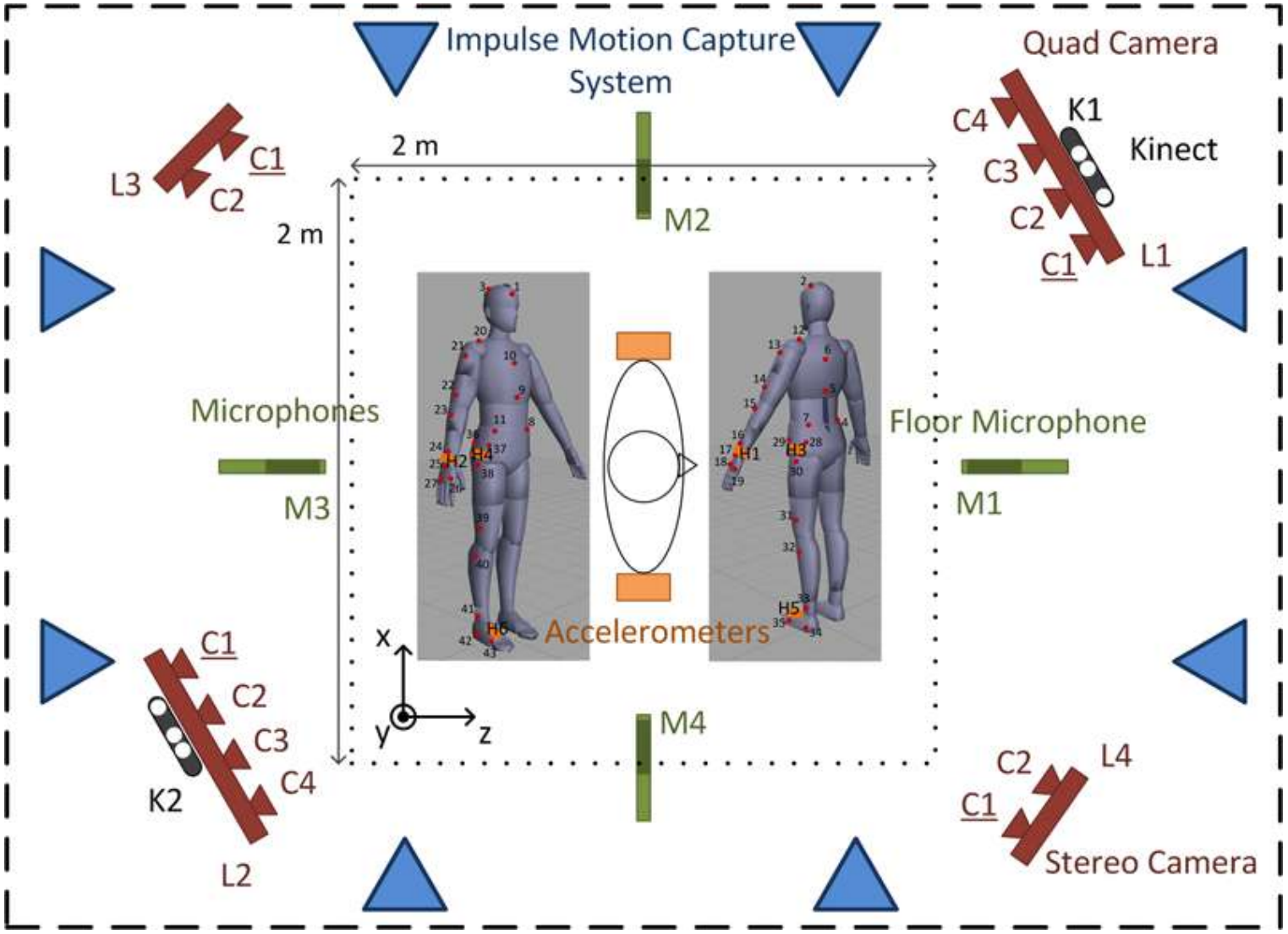
Use cases

- Understanding of position/orientation helps mobile phones re-orient the screen in portrait or landscape mode and reverse direction as required.
- Gesture recognition based on buffering continuous data and looking at the change in force and orientation.



Source: http://www.utdallas.edu/~kehtar/UTD-MHAD.html

Example: Berkeley Multimodal Human Action Database (MHAD)



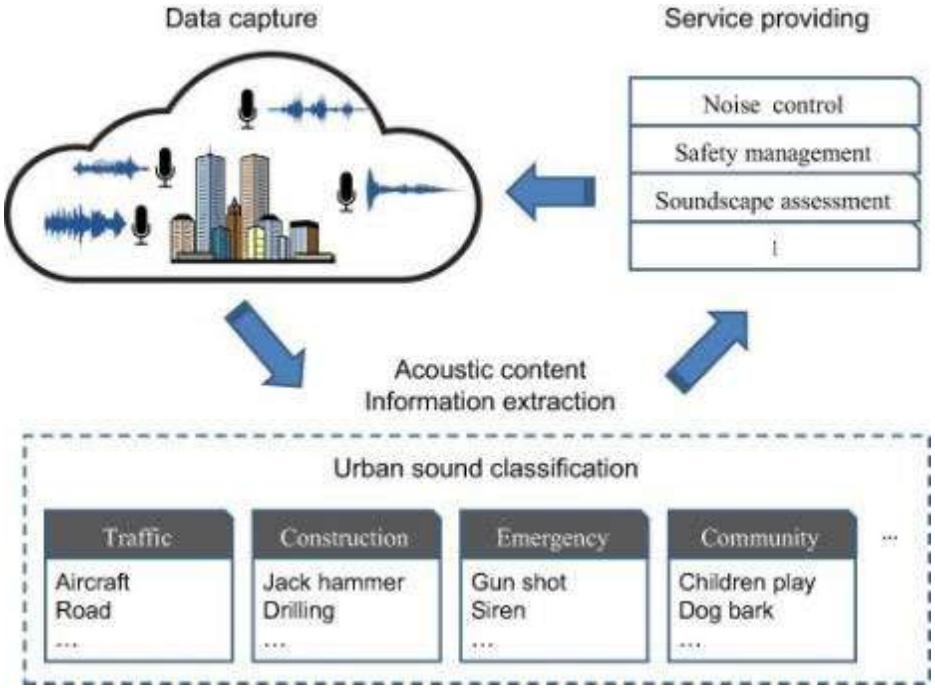
Source: http://tele-immersion.citris-uc.org/berkeley_mhad

Single type of sensor data: Audio

- Audio classification: A common IoT use case for microphones is to classify the environment that the sound was captured in.
- Voice activity detection: Determine whether there is voice in the captured audio.
- Speaker recognition: Speaker recognition, voice recognition, attempts to determine who is speaking.
- Keyword recognition: Recognize whether a particular word was uttered. Keyword recognition can also be generalized to key-phrase recognition and both are typically used as triggers for additional activity such as starting a session of commands or bringing up an application.
- Command and control: Refers to using a small set of phrases in speech recognition. For illustration, this could include a set of commands to control a toy car such as “move forward,” “move backward,” “go faster,” “go slower,” “turn right/left,” etc.

Example: Urban sound classification

- Dataset: Urban sound dataset, https://urbansounddataset.weebly.com/urbansound.html
- It contains 1302 labeled sound recordings. Each recording is labeled with the start and end times of sound events from 10 classes: air_conditioner, car_horn, children_playing, dog_bark, drilling, enginge_idling, gun_shot, jackhammer, siren, and street_music.



Source: https://www.sciencedirect.com/science/article/abs/pii/S0003682X16302274

Single type of sensor data: Vision

- Object recognition: Identify objects in an image and potentially matching them to a pre-existing database of objects that have been captured before.
- Face recognition: Detect a face in an image as well as matching that face against a database to label the face accordingly.
- Gesture recognition: Recognize static poses or moving gestures either specific to the hand/arm or the human body.
- Scene recognition: Identify multiple objects, faces, and people in an image and using that information to determine the likely activity or context.
- Anomaly detection: Identify if any anomaly occurred which should trigger additional analysis.
- Video summarization: Summarize the salient aspects of a long video stream. This includes scene changes, key scenarios and objects/characters that are the focus of the video.

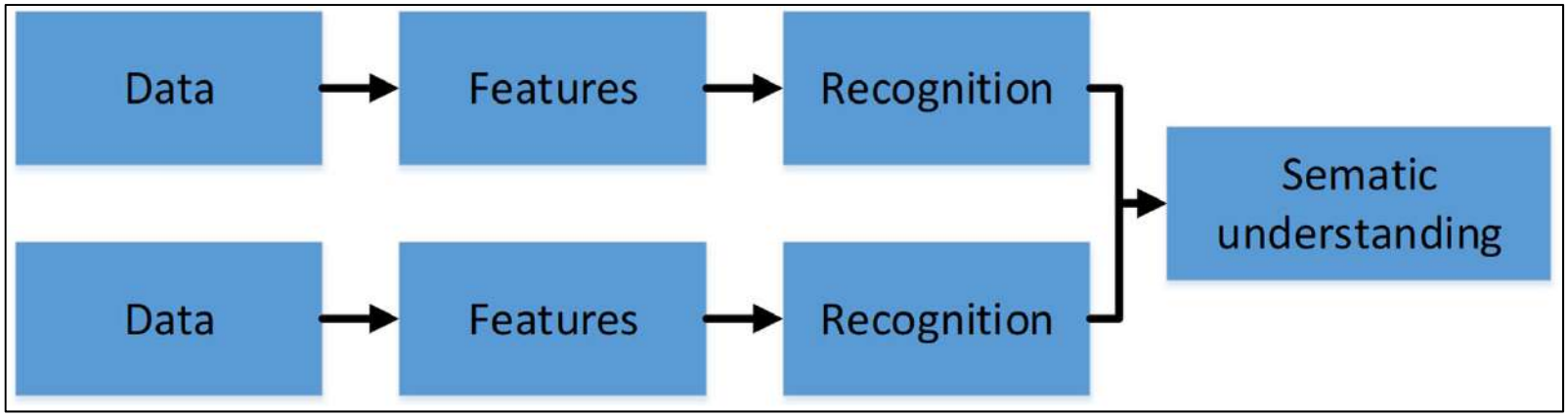
Example: Video understanding

- Dataset: YouTube-8M, a video understanding challenge, https://ai.googleblog.com/2017/02/an-updated-youtube-8m-video.html

Source: https://ai.googleblog.com/2017/02/an-updated-youtube-8m-video.html

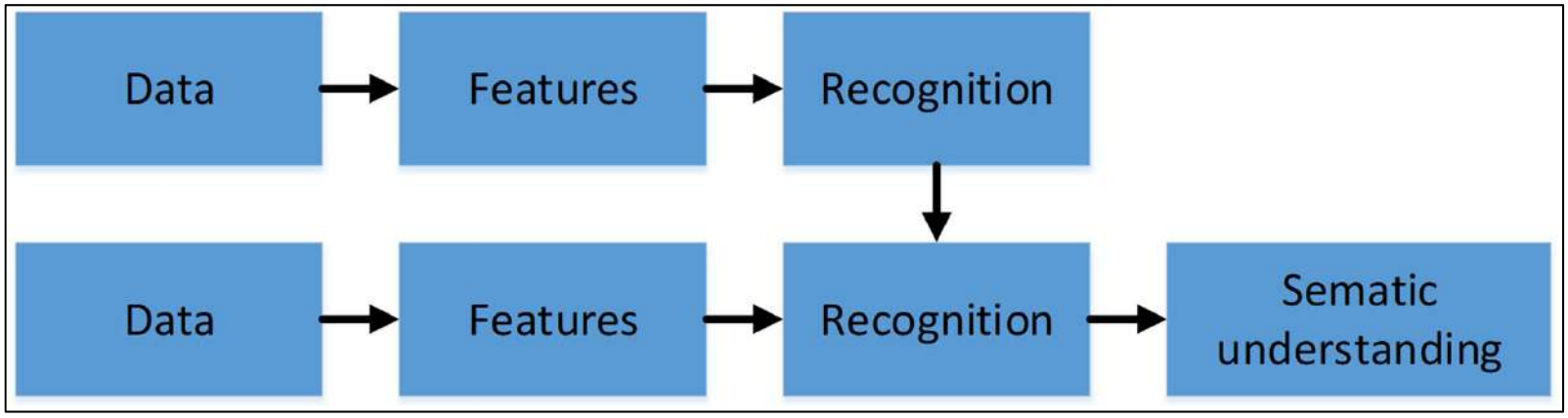
Multimodal sensor data recognition (1)

- Uncoupled sensor data fusion/Semantic fusion: The data fusion occurs at the last possible stage in the respective pipelines. The advantage of this method is that the sensor fusion is simple and can be accomplished using existing technology pipelines for recognition. Domain experts like this approach because cross-domain technical knowledge requirements are minimal.



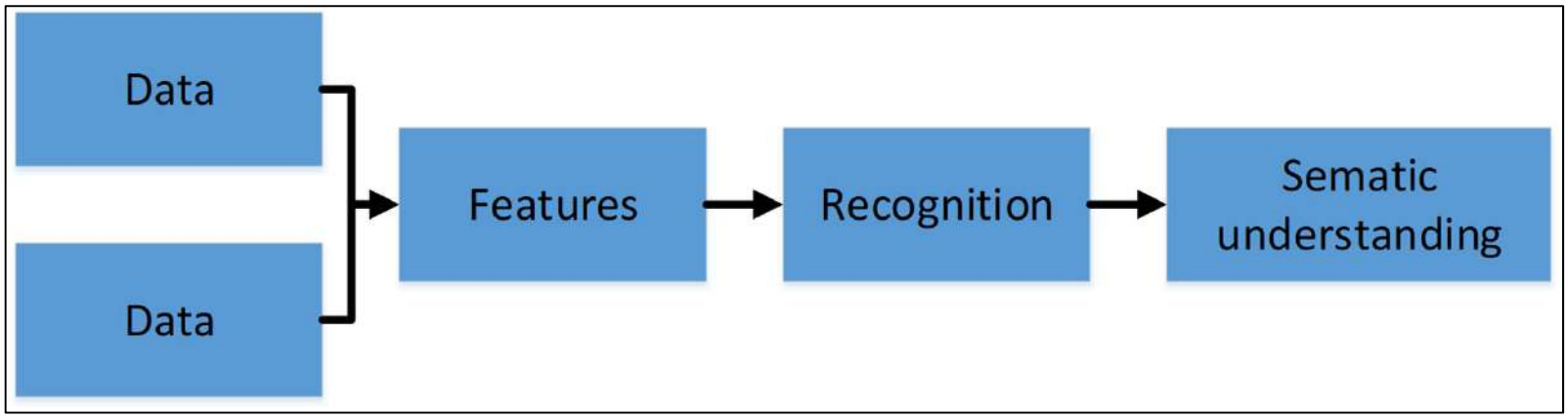
Multimodal sensor data recognition (2)

- Loosely coupled sensor data fusion/Restricted recognition: Restricts the recognition search space of mode modality based on the results from the other. In this approach, one recognition pipe helps to set the context for the other. Once that is accomplished, the second recognition pipe need only perform recognition within the established context boundaries, leading to potential savings in compute and higher performance.



Multimodal sensor data recognition (3)

- Tightly coupled sensor data fusion/data-feature level fusion: It allows for combination of multiple sensors that appear as a single complex sensor for most of the applications. Data representation at the lower levels of stack is very sensor-specific, so experts are needed to make sensible decisions for merging data from multiple sources. Each sensor model has a different data representation format at the lowest level, making it very hard to come up with a homogenous representation that preserves the information content.



Sensor data fusion (1)

- Representation: The information of the fusion process has an abstract level higher than each input data set.
- Certainty: If V is the sensor data before fusion and $p(V)$ is the *a priori* probability of the data before fusion, then the gain in certainty is the growth in $p(V)$ after fusion.
- Accuracy: The standard deviation on the data after the fusion process is smaller than the standard deviation provided directly by the sources.
- Completeness: Bringing new information to the current knowledge on an environment allows a more complete the view on this environment.

Sensor data fusion (2)

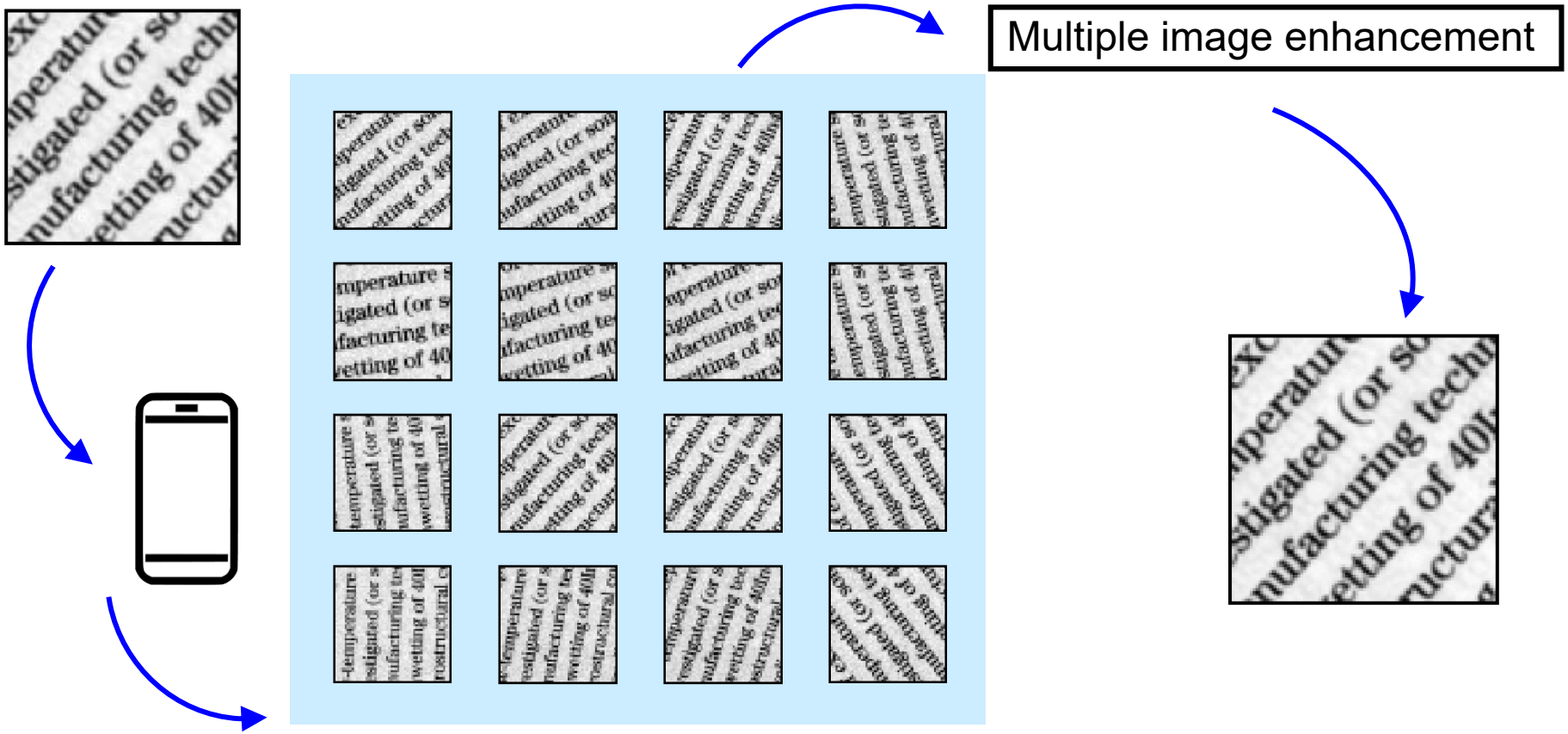
- Fusion across sensors (same type): A number of sensors nominally measure the same property as a number of temperature sensors measuring the temperature of an object.
- Fusion across domains (same type, different definition/range): A number of sensors measure the same attribute over a number of different ranges or domains. This arises, for example, in the definition of a temperature scale.
- Fusion across attributes (different types): A number of sensors measure different quantities associated with the same experimental situation, such as measuring air temperature, pressure and humidity to determine air refractive index.
- Fusion across time: Current measurements are fused with historical information, for example, from an earlier calibration.

Sensor data registration

- Spatial alignment. Transformation of the local spatial positions to a common coordinate system. The process involves geo-referencing the location and field-of-view of each sensor.
- Temporal alignment. Transformation of the local times to a common time axis. In many applications, is performed using a dynamic time warping algorithm.

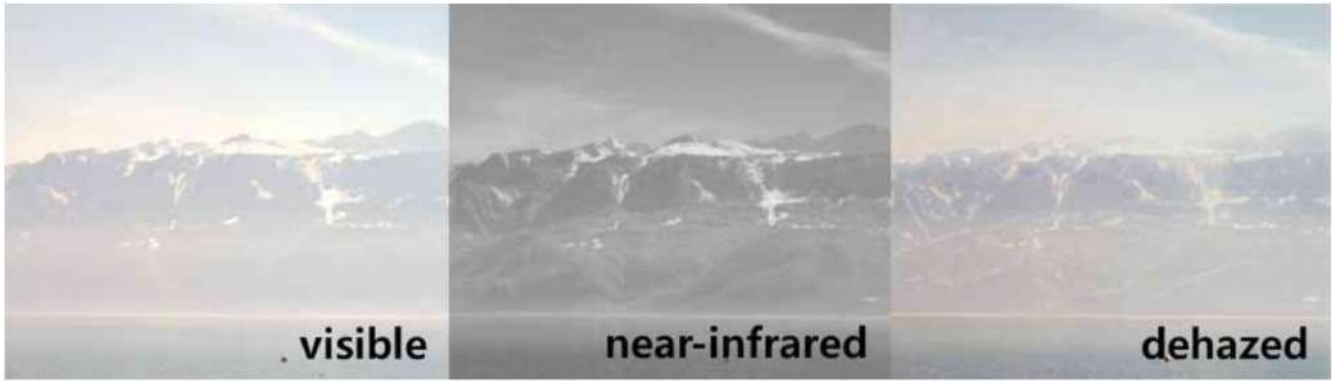
Sensor data registration (1)

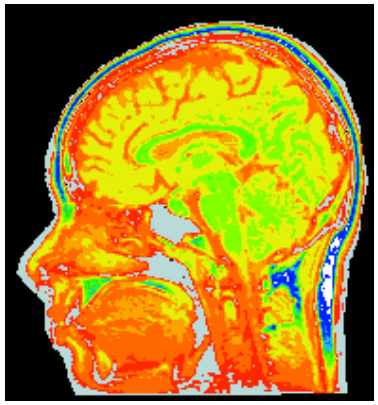
- At different times (multi-temporal fusion)

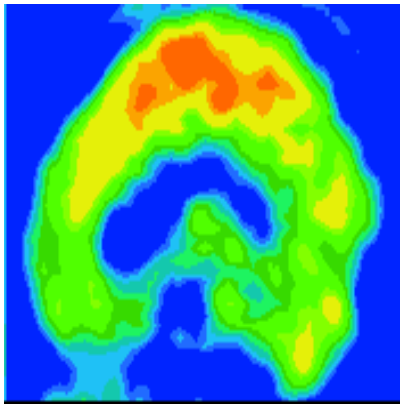


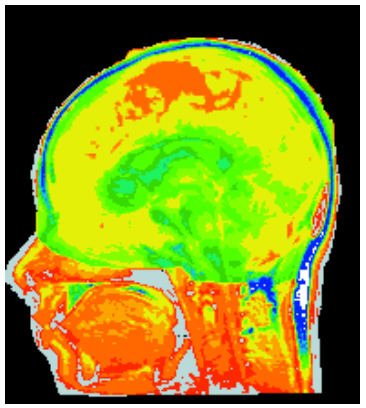
Sensor data registration (2)

- With different sensors (multi-modal fusion)









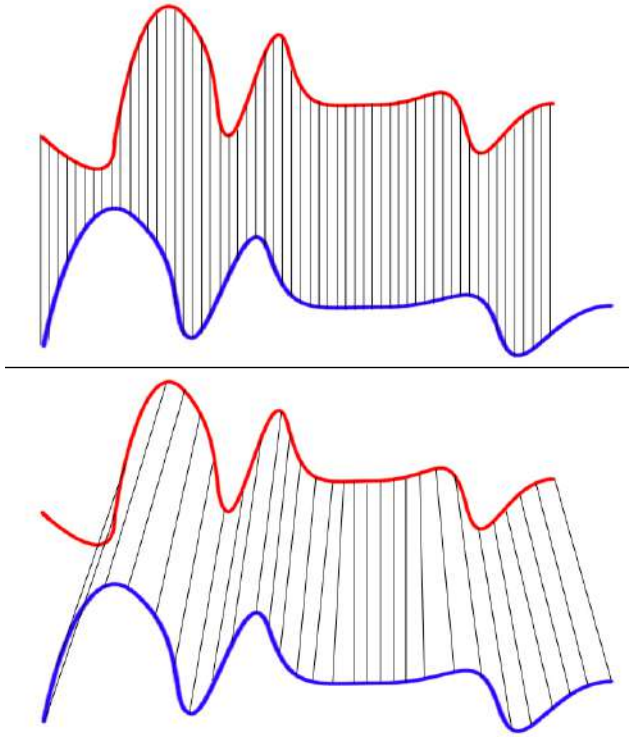
Sensor data registration (3)

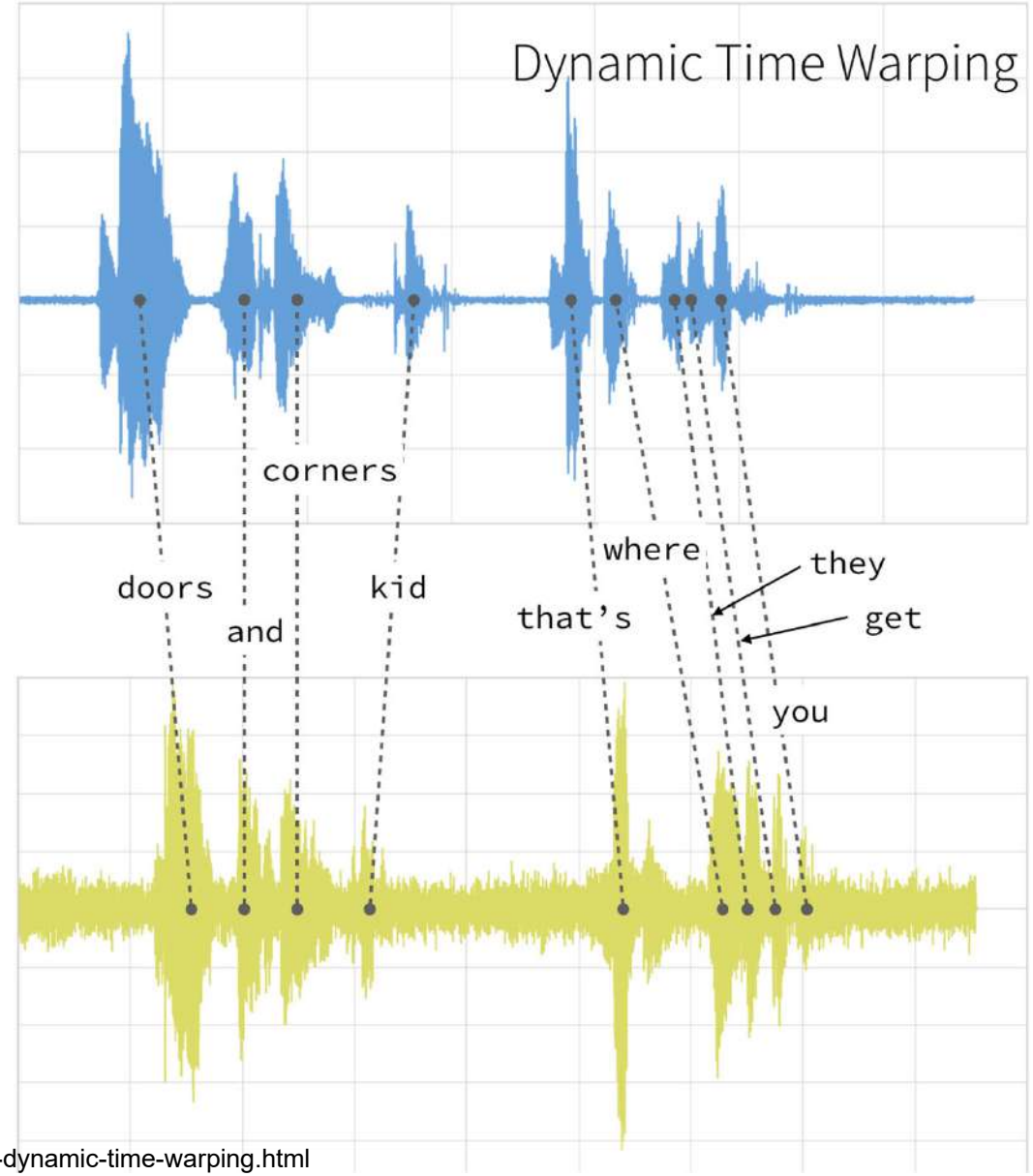
- From different viewpoints (multi-view fusion)

Sensor data registration (4): Audio

Euclidean registration (top figure): Align the i-th point on one time series with the same i-th point.

Dynamic time warping registration (bottom figure): Allows similar shapes to match even if they are different in the time axis.



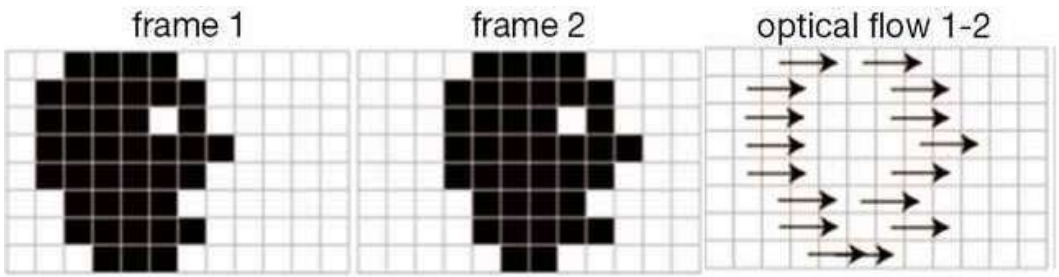


Reference: https://databricks.com/blog/2019/04/30/understanding-dynamic-time-warping.html

Sensor data registration (5): Video

Objective: Find point to point correspondence between two frames in the video sequence.

The pixel intensities of an object do not change between consecutive frames. Neighbouring pixels have similar motion.



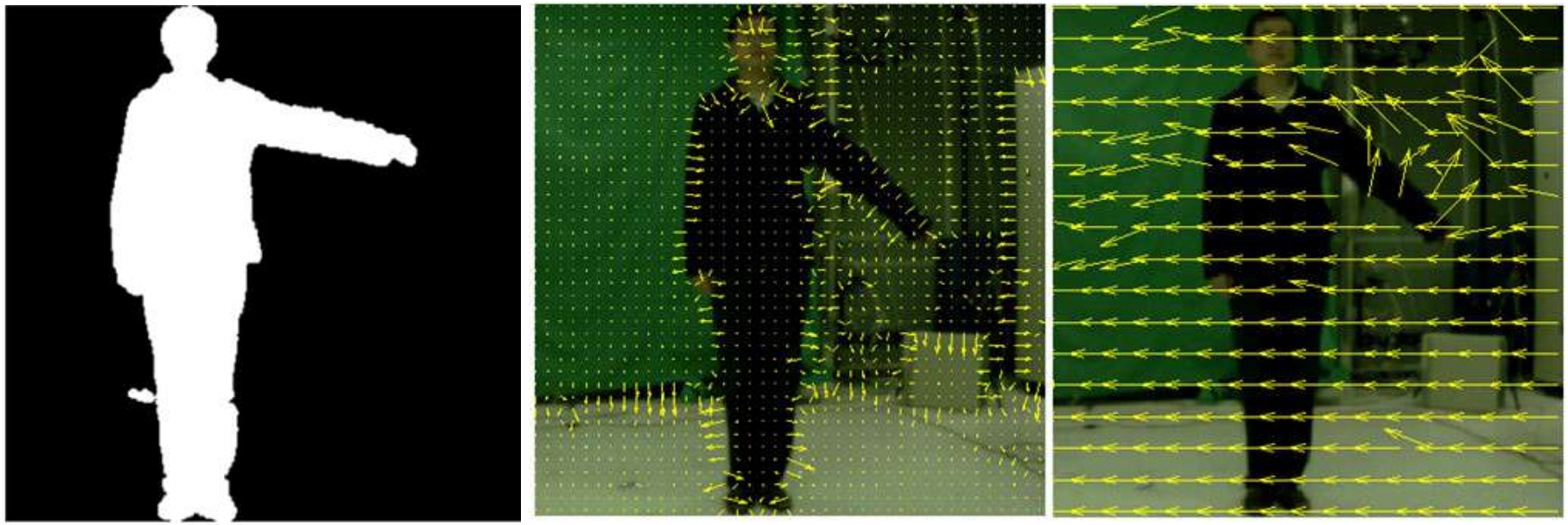
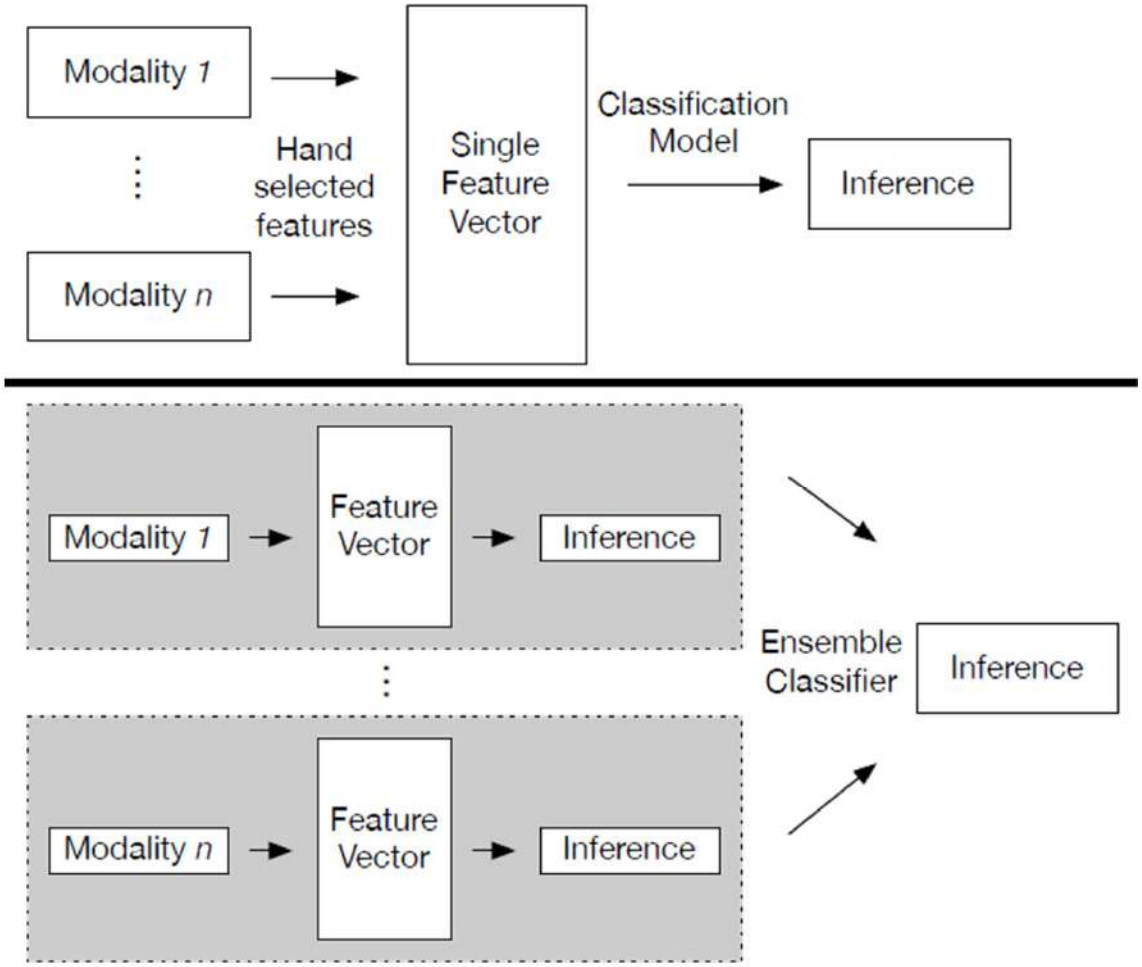


Photo: http://users.umiacs.umd.edu/~zhuolin/Keckgesturedataset.html

Sensor data fusion

- Feature concatenation with shallow classifiers: Hand-selected features from each modality are combined into a single feature vector presented to a classifier for detection across all features.
- Ensemble of shallow classifiers: Separate classifiers operating on each sensor (modality) provide their estimation. These estimations provided by each sensing modality classifier are fused to yield an overall class estimation.



Reference: V. Radu, *et al.*, Multimodal Deep Learning for Activity and Context Recognition, 2018, http://www.fahim-kawsar.net/papers/Radu.UbiComp2018-Camera.pdf

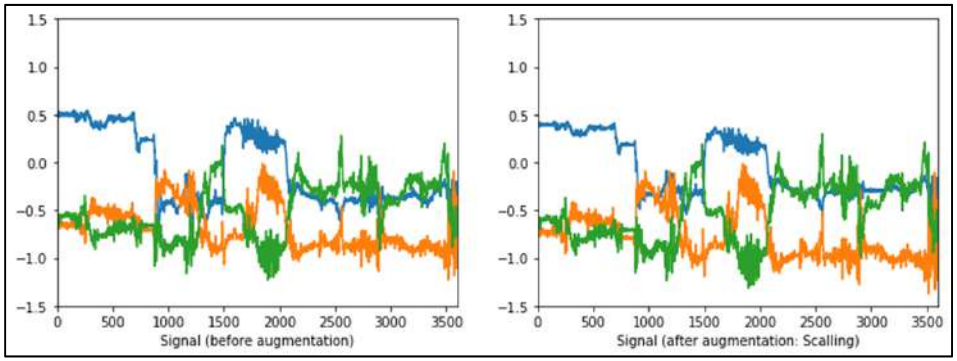
Data augmentation

- Data augmentation can be viewed as an injection of prior knowledge about the invariant properties of the data against certain transformations. Augmented data can cover unexplored input space, prevent overfitting, and improve the generalization ability of a machine learning model.
- Scaling changes the magnitude of the data in a window by multiplying by a random scalar.
- Magnitude filtering changes the magnitude of each sample by convolving the data window with a smooth curve varying around one.
- Jittering is simulating additive sensor noise. These data augmentation methods may increase robustness against multiplicative and additive noise and improve performance.
- Time-warping perturbs the temporal location. By smoothly distorting the time intervals between samples, the temporal locations of the samples can be changed using time-warping.

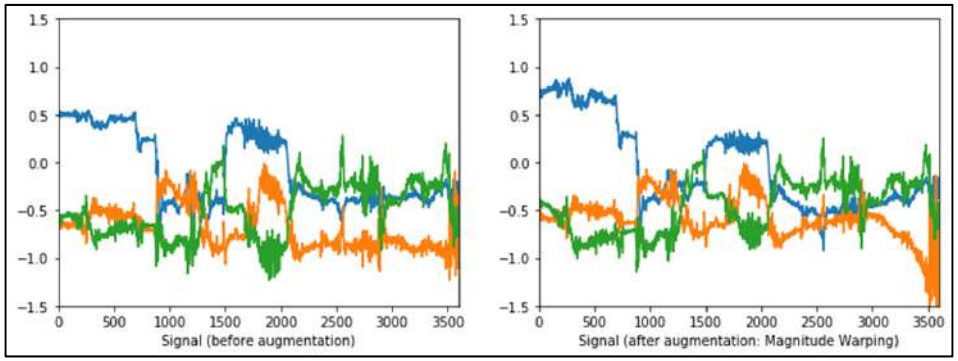
Reference: T. T. Um, *et al.*, "Data augmentation of wearable sensor data for parkinson's disease monitoring using convolutional neural networks," ACM Int. Conf. on Multimodal Interaction, New York, NY, USA, 2017, pp. 216–220.

Data augmentation

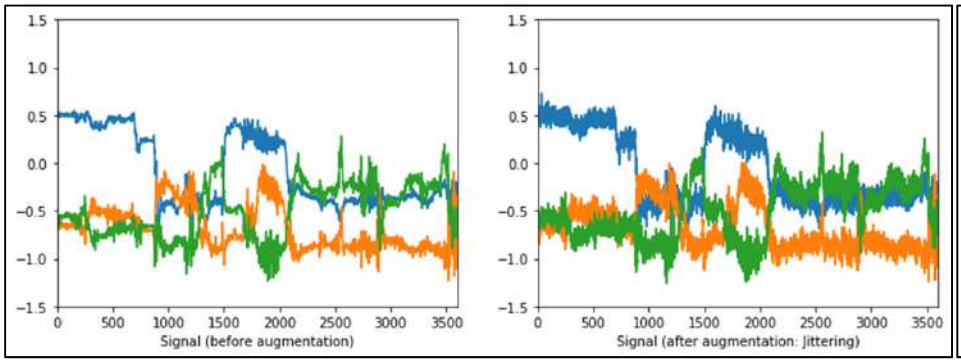
Scaling



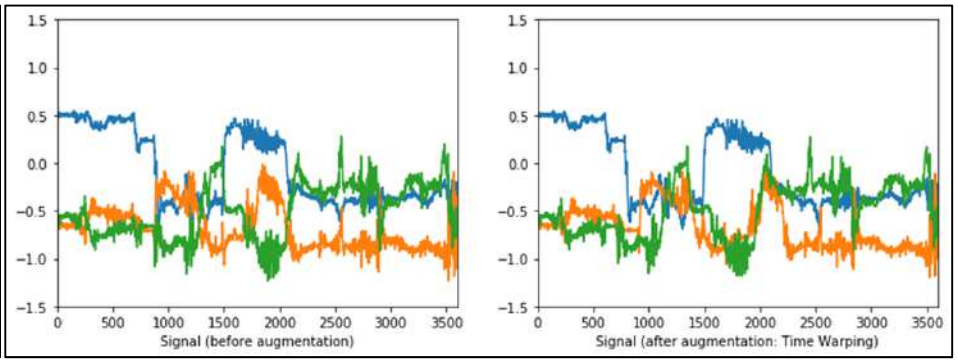
Magnitude filtering



Jittering



Time-warping



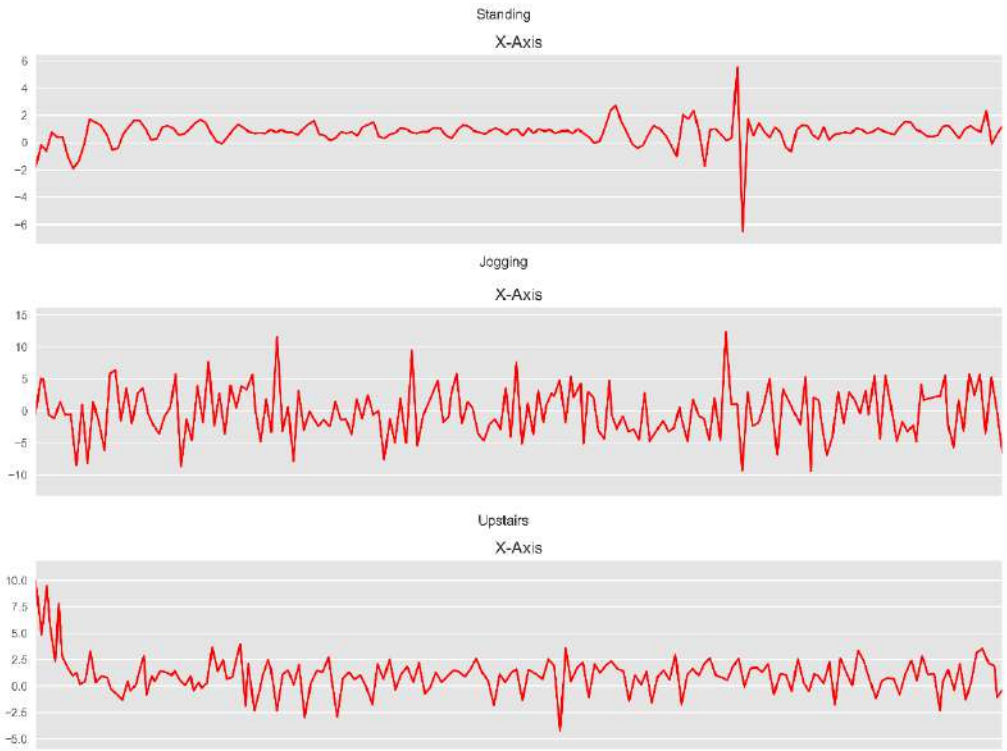
Example: Human activity classification

Example: Train a 1D convolutional neural network (1-D CNN) to recognize the type of human movement (e.g., Walking, Running, Jogging, etc.) based on a given set of accelerometer data from a mobile device carried around a person's waist.

Dataset: WISDM data set (Activity Prediction), http://www.cis.fordham.edu/wisdm/dataset.php.

Number of examples: 1,098,207
Class Distribution

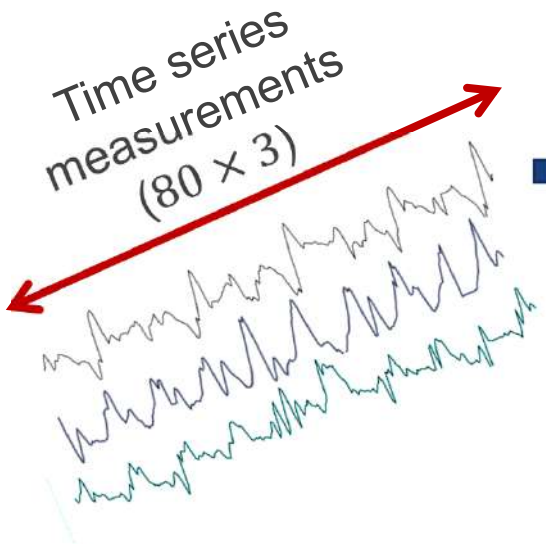
- Walking: 424,400 (38.6%)
- Jogging: 342,177 (31.2%)
- Upstairs: 122,869 (11.2%)
- Downstairs: 100,427 (9.1%)
- Sitting: 59,939 (5.5%)
- Standing: 48,395 (4.4%)



Example: Time-series data classification

Problem statement: We have chopped accelerometer data into (user-determined) 4 second chunks, can we recognize human activity from their gait in 4 seconds?

Model input	Model	Model output
Human accelerometer data (say, 4 seconds data)	To build a neural network model	Human activity (6 categories for this dataset)



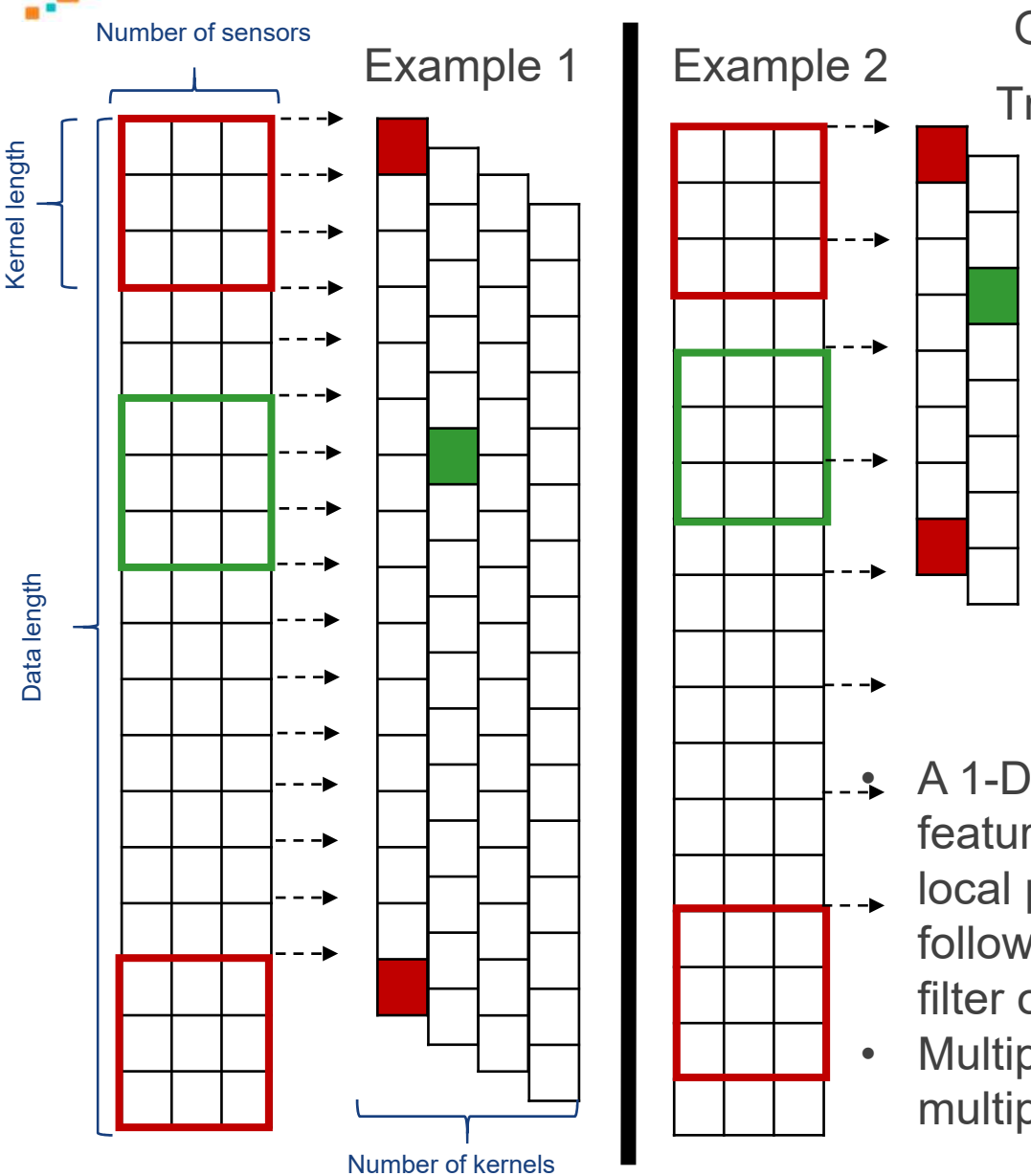
Neural network model

Human activity classification scores over (6) categories

# of measurements (4 seconds, 20 Hz)	80
# of sensors (accelerator X, Y, Z)	3
# of activities (e.g., walking)	6

Reference: H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, P.-A. Muller, "Deep learning for time series classification: a review," Data Mining and Knowledge Discovery, 2019, https://arxiv.org/abs/1809.04356

1-D Convolution



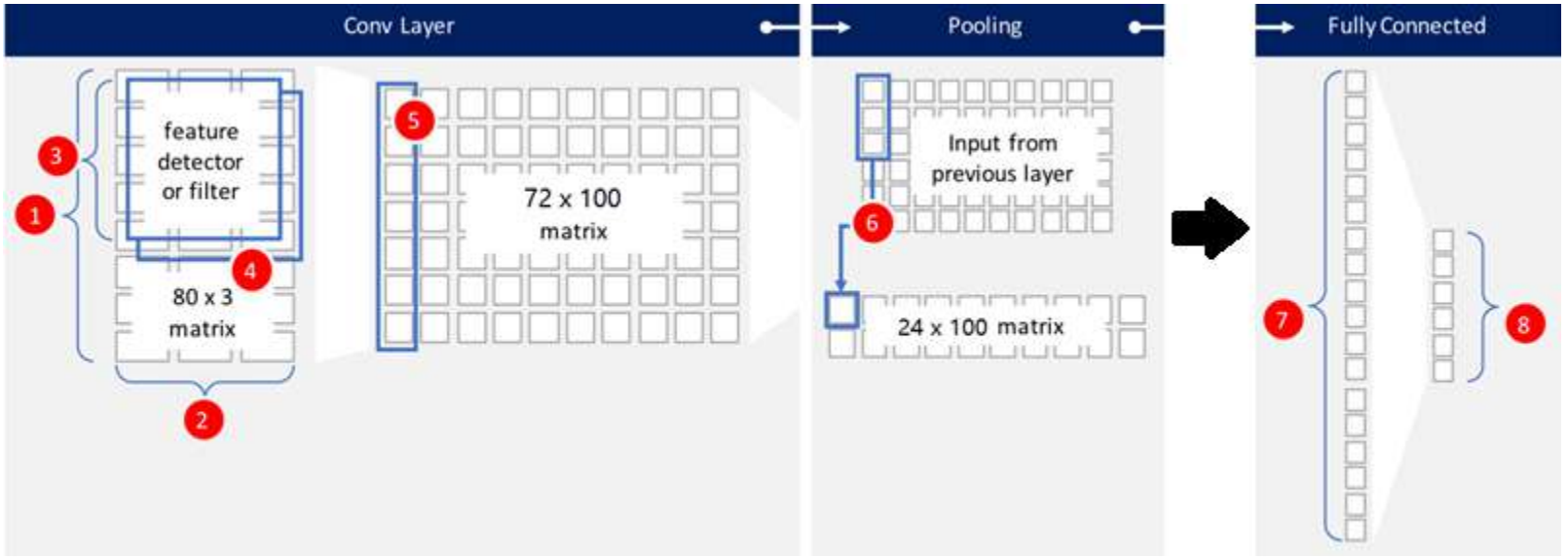
Output layer size: $\frac{L-K+2P}{S}+1@F$

Trainable parameters: $(K \times N+1) \times F$

Comparison between two examples			
L	Data length	18	18
K	Kernel length	3	3
P	Padding for input	0	0
S	Stride (sliding kernel)	1	2
F	Number of kernels	4	2
N	Number of sensors	3	3

- A 1-D $conv$ filter is used to create a feature map, by applying the filter on a local patch to get weighted summation, followed by activation, and sliding the filter over the whole data with a stride.
- Multiple $conv$ filters are used to create multiple feature maps.

Example: Human activity classification



Detailed descriptions are provided in the next slide.

Photo: Modified from https://blog.goodaudience.com/introduction-to-1d-convolutional-neural-networks-in-keras-for-time-sequences-3a7ff801a2cf

Example: Human activity classification

1 Height

The length of one data set which is fed into the network. In our case, the length is 80. Is also referred to as the input height.

2 Width

The width of the data set which is fed into the network. In our case, the width is 3 (x, y and z accelerometer reading). Is also referred to as the depth.

3 Kernel Size

This is the size / height of the sliding window that convolves across the data. Also referred to as kernel size or filter length. 10 entries in one window in our case.

4 Filters

Defines how many sliding windows will run through the data. In our case 100, therefore 100 different features can be detected in this layer. Also called feature detectors.

5 Output

The kernel size and the height of the original data determines the amount of neurons in the output.

6 Max Pooling

This step will prevent overfitting of the learned features by taking the max value of multiple features. Again, a sliding window approach is used.

7 Dense

The last layer is a fully connected layer (Dense in Keras). It uses a Softmax activation function to produce a probability distribution over the 6 output classes.

8 Final Output

The final output layer consists of 6 neurons (one for each label "Walking", „Jogging", etc.) including its probability.

Photo: Modified from https://blog.goodaudience.com/introduction-to-1d-convolutional-neural-networks-in-keras-for-time-sequences-3a7ff801a2cf

Example: Human activity classification

```
# 1D CNN neural network
model_m = Sequential()
model_m.add(Reshape((TIME_PERIODS, num_sensors), input_shape=(input_shape,)))
model_m.add(Conv1D(100, 9, activation='relu', input_shape=(TIME_PERIODS, num_sensors)))
model_m.add(MaxPooling1D(3))
model_m.add(Conv1D(160, 10, activation='relu'))
model_m.add(GlobalAveragePooling1D())
model_m.add(Dense(num_classes, activation='softmax'))
print(model_m.summary())
```

Convolution kernel size: 9

Manually crop input data to be segments with size 80×3

Layer (type)	Output Shape	Param #	
reshape_1 (Reshape)	(None, 80, 3)	0	
conv1d_1 (Conv1D)	(None, 72, 100)	2800	(9*3+1)*100 =2800
max_pooling1d_1 (MaxPooling1	(None, 24, 100)	0	
conv1d_2 (Conv1D)	(None, 15, 160)	160160	(10*100+1)*160 =160160
global_average_pooling1d_1 (	(None, 160)	0	
dense_1 (Dense)	(None, 6)	966	(160+1)*6=966

80-9+1=72

72/3=24

24-10+1=15

Summary

- Making sense of sensor data pipeline
- Single type of sensor data analytics
- Multiple types of sensor data analytics

Thank you!

Dr TIAN Jing
Email: tianjing@nus.edu.sg