HUMAN ACTIVITY RECOGNITION

Yitian Shi and Jonas Vogt

Institute of Signal Processing and System Theory, University of Stuttgart, Germany

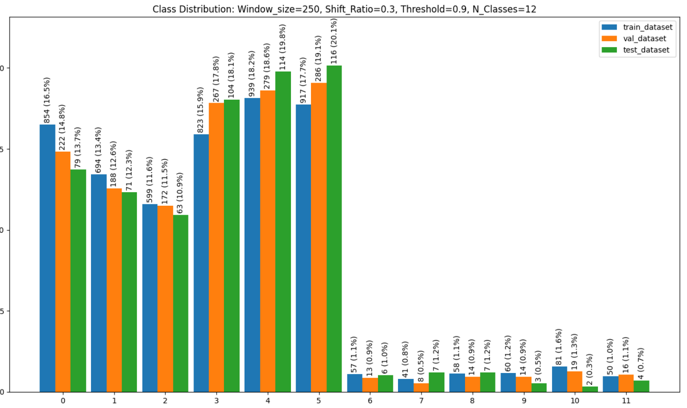
Universität Stuttgart

ISS

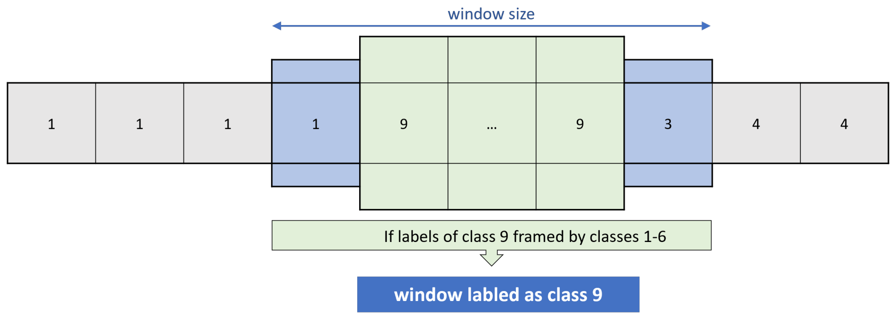
1. Datasets Overview and Preprocess

Besides the data preprocessing pipeline mentioned in the script, there are still some important data handling techniques that deal with the different characteristics and difficulties of each dataset.

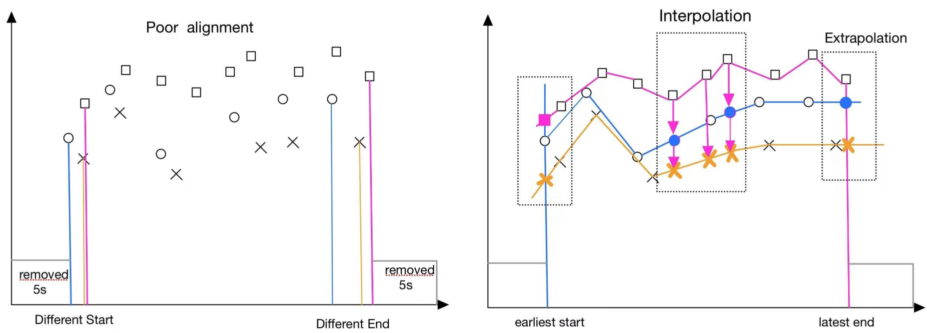
HAPT



According to the data statistics, the dataset contains highly imbalanced data between continuous and transitional classes (such as SIT_TO_STAND, STAND_TO_SIT, etc.). These activities are prioritized while using the sliding window techniques in such a way, that a windows will be labeld by the transitional class when this activity is in between (framed by) two continuous activities.



HAR realworld2016



Because of the slight differences of recording time in several milliseconds, the data is not perfectly aligned.By nearest inter/extrapolation of all the data on the same discrete-time range will solve this problem under the assumption, that the data doesn't change itself in between 20ms, which corresponds to the smallest sampling period.

2. Models & Results

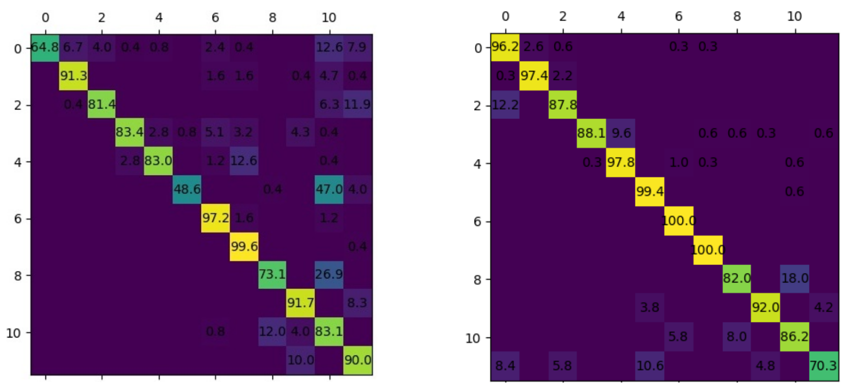
HAPT

Besides traditional RNN and LSTM architectures, we've also applied the Transformer encoder structure as the backbone with a normal classification head. We have the following configurations compared to the model structure from the base model in the paper 'Attention is all you need':

Model	layers	d_model	d_feed_forward	num_head	d_key/value
Base model	6	512	2048	8	64
Our model	1	32	128	4	8

The best results we've achieved are as follows:

Architecture	Window size	Dropout rate	Balanced test accuracy
LSTM 13	250	0.4	86.92%
LSTM 10	300	0.2	88.90%
GRU 10	300	0.2	84.44%
Stacked LSTM (11,11)	250	0.1	91.40%
Stacked GRU (11,11)	300	0.1	77.91%
Stacked LSTM (15,12)	250	0.3	88.50%
Stacked LSTM (8,11)	250	0.5	84.50%
Transformer encoder	300		82.28%
Soft-Voting	250		**91.64%**

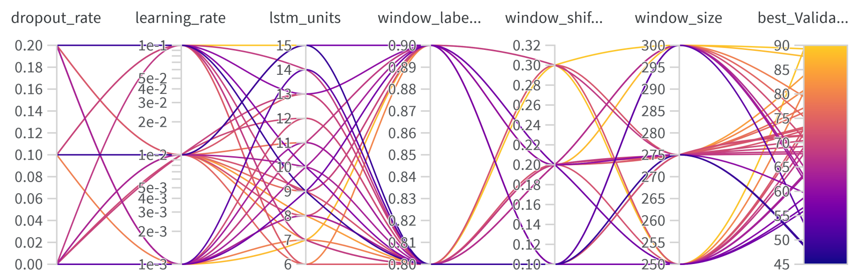


(a) Transformer encoder (b) Soft voting

Abbildung 4: *Confusion matrices examples*

Hyperparamter tuning with Weights&Biases:

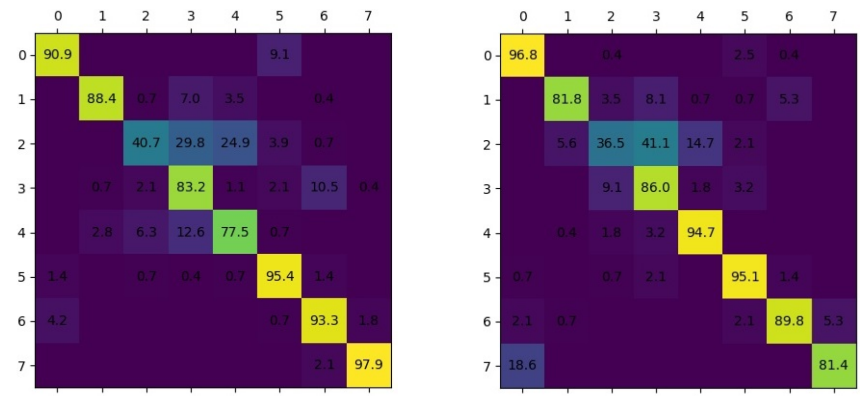
Hyperparameter	Window size	Shift-Ratio	Labeling-Threshold
value	250	0.3	0.9



HAR realworld2016

We've tried different model architectures on different sensor positions. Some positions can present good accuracies on a specific position (one against others)

Sensor position	Architecture	Balanced test accuracy
upperarm	stacked LSTM (10,10)	83.42%
chest	LSTM 10	82.76%
shin	stacked LSTM (10,10)	81.84%
waist	stacked LSTM (10,10)	75.44%
forearm	stacked LSTM (10,10)	73.03%
head	stacked LSTM (10,10)	67.32%



(a) upperarm (b) chest

Abbildung 6: *Confusion matrices examples*

Meanwhile, we've also discovered the situation, that some of the sensor data give a big difference between the best validation and test accuracy caused by different distributions of both datasets. One reason we assume is the variety of human physical characteristics which influence their activity behaviors. These results comparisons are as follows:

Position	Architecture	Validation	Test	accuracy difference
upperarm	stacked LSTM (10,10)	49.40%	83.42%	-34.02%
waist	stacked LSTM (10,10)	52.99%	75.44%	-22.45%
thigh	LSTM 11	76.69%	56.32%	20.37%

3. Android Application

Last but not least we've developed an android application to record the realtime sensor data and make predictions using Tensorflow Lite model.After each 6 seconds (300 samples with 20ms sampling period) the probabilities of each classes will be refreshed. This module is still under development because of some adaptation issues.

Deep Learning Lab 2022, February 4, Stuttgart, Germany