American Sign Language recognition using LeapMotion sensor

Gustavo L. Mourão

February 12, 2018

1 Proposition

The proposed system has aimed to implement a strategy able to identify a set of American Sign Language characters. Initially will be adopt a set of features based on recognition of static symbols. After that will be expose a real time application based on the models previously obtained.

Initially will be characterized the models used in this application.

2 Applied Models and Real-Time Application

The application was development in Python environment, version 2.7. To evaluate different models were used the scientific toolbox scikit-learn.

2.1 SVM

loading...

2.2 KNN

loading...

2.3 ANN

loading...

2.4 Nearest Centroid Classifier

loading...

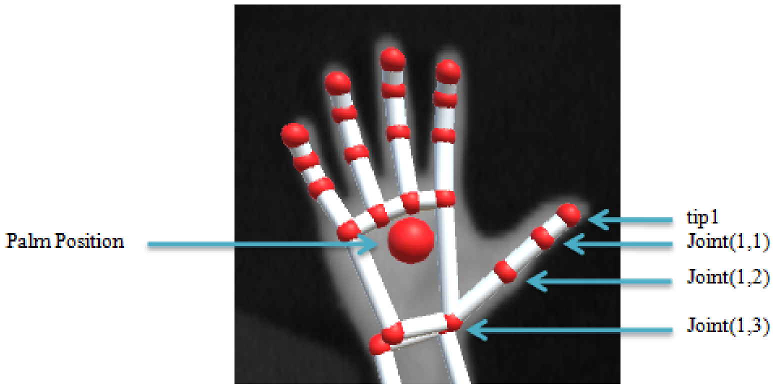

Figure 1: Positions used as base of features

3 Features Extraction

In this section will be defined a strategy for calculate a set of features from static symbols. To evaluate the performance of each model, the results will be present in Confusion Matrix format.

The features extracted were collected based on the position of each finger (in terms of Cartesian Coordinates), by implementation built on Python language using SDK (Software Development Kit) library.

The first feature calculated was the tip to palm ratio, which represents the relation of the distance between a finger tip (α) to the palm to the summation of distances between finger and the palm (as represented in Figure 1), given by Equation 1 [2].

$$RTP(\alpha) = \frac{D_{(palm,tip_\alpha)}}{\sum_{i=1}^{5} D_{(palm,tip_i)}} \tag{1}$$

Secondly was obtained the tip to tip ratio. Basically is the ratio of the distance between finger tips tip α and tip β to the total distance among finger tips, which is given by Equation 2.

$$RTT(\alpha,\beta) = \frac{D_{(tip_\alpha,tip_\beta)}}{\sum_{i=1}^{4}\sum_{j=i+1}^{5} D_{(tip_i,tip_j)}} \tag{2}$$

Finally, the ration between $D_{(tip_\alpha,joint_{\alpha,3})}$ and the total length of the finger α (the tip to joint ratio), were calculated as follows by Equation 3.

Figure 2: Example of collected symbol

$$RTJ(\alpha) = \frac{D_{(tip_\alpha, joint_{\alpha,3})}}{D_{(tip_\alpha, joint_{\alpha,1})} + \sum_{i=1}^{2} D_{(tip_{\alpha,i}, joint_{\alpha,i+1})}} \tag{3}$$

4 Experimental Results and Analysis

4.1 Off-line Training and Getting Models

The samples used in experiment consist of 24 representation of static letters from the American Sign Language standard. For each word were extract five (5) RTP features, ten (10) RTT features and five (5) RTJ feature.

Also, in order to get uncorrelated data, the set of features data were extract from two (2) people. The distance from the sensor were adopt as 30cm. For enable the storage of the collection of the data, were adopted a threshold based on the palm norm vector. Once the absolute value of the normal vector were higher than 0.9, then the data were enable to storage the set of data. Figure 2 represents an example of collected symbol.

After calculate the set of features, 60% were used to train the each model and of 40% of prediction.

The precision related to each model is represented on Table 1.

Table 1: Precision for each model

Model	Precision%
SVM	67
KNN	97
ANN	94
Nearest Centroid	64

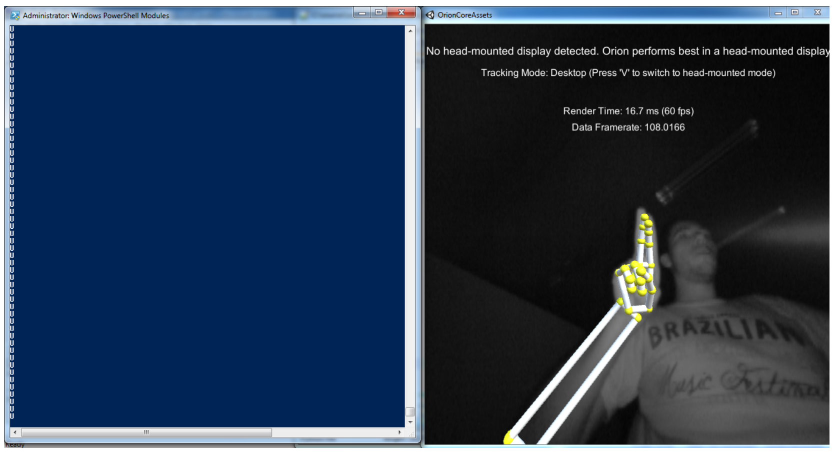

Figure 3: Interpretation of U symbol from ANN model

In the Table 2 is represented the confusion matrix refer to SVM model, considering a RBF kernel. Tables 3, 4 and 5 represents KNN, ANN and Nearest Centroid model, respectively.

4.2 On-line Application

Next, was implemented a basic real time application, based on the acquisition of the features calculated and to the models obtained

Figures 3 and 4 represents the results refer to U and V symbols, respectively. There represents the response from Real-time implementation from ANN model.
In terms of time processing related to each model obtained, Table 6 represents the aspects to each one.

It can be noted comparing Tables 1 and 6 that the algorithm KNN return high precision with low processing time. However, an important aspect observed from the real-time application was recognition symbols realized in transition from each other. For example, between the transition of symbols F and I could be possible a realization of symbol A.

To avoid the false recognition, in terms of transition symbols, could be fea-

Table 2: SVM

	A	B	C	D	E	F	G	H	I	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
A	227	0	70	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0
B	0	398	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C	0	0	398	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D	0	0	0	398	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E	0	0	0	0	398	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F	0	0	0	0	0	398	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
G	0	0	0	0	0	0	398	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
H	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	398	0	0	0	0
I	0	0	0	0	0	0	0	0	398	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
K	0	0	0	0	0	0	0	0	0	199	0	0	0	0	0	0	0	0	0	199	0	0	0	0
L	0	0	0	0	0	0	0	0	0	0	398	0	0	0	0	0	0	0	0	0	0	0	0	0
M	0	0	0	0	199	0	0	0	0	0	0	0	0	0	0	0	0	199	0	0	0	0	0	0
N	0	0	0	0	0	0	0	0	0	0	0	0	203	0	0	0	0	166	29	0	0	0	0	0
O	0	0	0	0	5	0	0	0	0	0	0	0	0	393	0	0	0	0	0	0	0	0	0	0
P	0	0	0	289	0	0	107	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0
Q	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	398	0	0	0	0	0	0	0	0
R	0	0	0	0	0	0	199	0	0	0	0	0	0	0	0	0	0	0	0	199	0	0	0	0
S	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	398	0	0	0	0	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	111	287	0	0	0	0	0	0
U	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	398	0	0	0	0
V	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	398	0	0	0
W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	398	0	0
X	0	0	0	199	0	0	199	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Y	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	398

Table 3: KNN

A	B	C	D	E	F	G	H	I	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
223	0	64	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	1	0	0	0
0	398	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	398	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	379	0	0	0	0	0	0	0	0	0	0	19	0	0	0	0	0	0	0	0	0
0	0	0	0	398	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	398	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	398	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	398	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0
0	0	0	0	0	0	0	0	398	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	398	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	398	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	06	0	0	0	0	0	0	0	398	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	354	0	0	0	0	44	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	398	0	0	0	0	0	0	0	0	0	0
0	0	0	24	0	0	0	0	0	1	0	0	0	0	288	0	2	0	0	1	0	0	82	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	398	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	398	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	81	0	0	0	0	0	317	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	398	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	398	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	398	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	398	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	398	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	398

Table 4: ANN

A	B	C	D	E	F	G	H	I	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
225	0	72	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	1	0	0	0
0	398	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	398	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	291	0	0	0	0	0	0	0	0	0	0	107	0	0	0	0	0	0	0	0	0
0	0	0	0	275	0	0	0	0	0	123	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	398	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	398	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	398	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0
0	0	0	0	0	0	0	0	398	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	398	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	398	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	07	0	0	0	0	0	0	0	398	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	352	0	0	0	0	0	46	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	398	0	0	0	0	0	0	0	0	0	0
0	0	0	17	0	0	1	0	0	1	0	0	0	0	350	0	0	0	0	1	0	0	28	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	398	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	398	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	138	60	0	0	0	0	200	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	398	0	0	0	0	0
0	0	0	0	0	0	0	8	0	0	0	0	0	0	0	0	0	0	0	390	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	398	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	398	0	0
0	0	0	0	0	0	0	44	0	0	0	0	0	0	0	155	0	0	0	0	0	0	199	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	398

Table 5: Nearest Centroid

	A	B	C	D	E	F	G	H	I	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
A	228	0	69	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0
B	0	199	0	0	0	0	0	0	0	0	0	0	199	0	0	0	0	0	0	0	0	0	0	0
C	0	0	398	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D	0	0	0	398	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E	0	0	0	0	398	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F	0	0	0	0	0	398	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
G	0	0	0	0	0	0	398	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
H	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	398	0	0	0	0
I	0	0	0	0	0	0	0	0	398	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
K	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	199	0	0	199	0	0	0	0
L	0	0	0	0	0	0	0	0	0	0	398	0	0	0	0	0	0	0	0	0	0	0	0	0
M	0	0	0 ∞	0	199	0	0	0	0	0	0	0	0	0	0	0	0	199	0	0	0	0	0	0
N	0	0	0	0	0	0	0	0	0	0	0	0	199	0	0	0	0	165	34	0	0	0	0	0
O	0	0	0	0	0	0	0	0	0	0	0	0	0	398	0	0	0	0	0	0	0	0	0	0
P	0	0	0	153	0	0	43	0	0	0	0	0	0	0	151	0	0	0	0	1	0	0	28	0
Q	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	398	0	0	0	0	0	0	0	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	199	0	0	191	0
S	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	398	0	0	0	0	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	112	0	160	126	0	0	0	0	0
U	0	0	0	0	0	0	0	8	0	0	0	0	0	0	0	0	0	0	0	398	0	0	0	0
V	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	398	0	0	0
W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	398	0	0
X	0	0	0	199	0	0	199	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Y	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	398

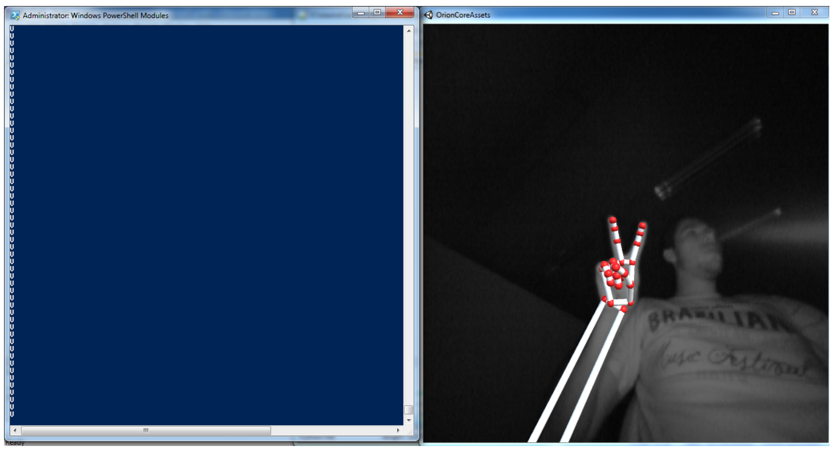

Figure 4: Interpretation of V symbol from ANN model

Table 6: Processing Time related to each model

Model	Processing Time (s)
SVM	0.0023
KNN	0.0042
ANN	0.0048
Nearest Centroid	0.0012

sible apply an median filter. This approach is largely disseminate, which would not bring innovation. Other approach is apply an adaptive temporal trajectory filtering using Wiener algorithm [1].

5 Possible Future Approach

As first approach to the next step the objective is apply a Wiener filtering with aim to reduce false unexpected recognition which derive from transition effect.

In terms of next steps, also could be implement a strategy to get available features from dynamic symbols (letters J and Z), and apply to the proposed models.

Other possible approach is apply some face detection algorithm (Viola Jones, for example), to each frame from NIR LeapMotion camera. This could be combined with hand features since there are signal gestures which has the face as reference.

Finally, could be possible to combine the face recognition system from Kinect sensor with LeapMotion recognition. With this approach could be possible to interpret dynamic gestures which uses face as reference.

References

[1] M. Esche, A. Glantz, A. Krutz, and T. Sikora. Adaptive temporal trajectory filtering for video compression. *IEEE Transactions on Circuits and Systems for Video Technology*, 22(5):659–670, 2012.

[2] K.-Y. Fok, N. Ganganath, C.-T. Cheng, and K. T. Chi. A real-time asl recognition system using leap motion sensors. In *Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), 2015 International Conference on*, pages 411–414. IEEE, 2015.