# Supplementary material for the paper Classification Algorithm for Person Identification and Gesture Recognition Based on Hand Gestures with Small Training Sets

(Dated: December 3rd, 2020)

Below there are pseudocodes of the algorithm proposed in the article.

**Require:** a survey $\Theta_i = (x_i^v)_{v=1}^V$, $i \in \{1, 2, \ldots, D\}$
**for** $v = 1$ **to** $2V$ **do**
  **if** $v \leq V$ **then**
    $\Theta'(1, v) \leftarrow x_1^v$
  **else**
    $\Theta'(1, v) \leftarrow 0$
  **end if**
**end for**
**for** $i \in \{2, 3, \ldots, D\}$ **do**
  **for** $v \in \{1, 2, \ldots, 2V\}$ **do**
    **if** $v \leq V$ **then**
      $\Theta'(i, v) \leftarrow x_i^v$
    **else**
      $\Theta'(1, v) \leftarrow x_i^v - x_{i-1}^v$
    **end if**
  **end for**
**end for**
**return** $\Theta'$

**Algorithm 1:** Transformation, step 1 in training and prediction phases.

**Require:** symbol table $\Pi''(i)$, $i \in \{1, 2, \ldots, D\}$
**for** $l = 1$ **to** $k_1$ **do**
  $B(l) \leftarrow 0$
**end for**
**for** $i = 1$ **to** $D$ **do**
  $B(\Pi''(i)) \leftarrow B(\Pi''(i)) + 1$
**end for**
**return** $B$

**Algorithm 2:** Frequency table calculation, step 5 in training and step 3 in prediction.

**Require:** training data $\Theta_{j,i} = (x_{j,i}^v)_{v=1}^V$, $c_j$, $j \in TR$, $i \in \{1, 2, \ldots, D\}$; clustering parameters $k_1$, $d_{clust}$; $k$NN parameters $k_2$, $d_{kNN}$

1: {Step 1.}
2: **for** $j \in TR$ **do**
3: $\quad \Theta'(J, :, :) \leftarrow \text{TRANSFORM}(\Theta_{j,:})$
4: **end for**
5: {Step 2.}
6: **for** $j \in TR$ **do**
7: $\quad$ **for** $i = 1$ **to** $D$ **do**
8: $\quad\quad$ **for** $v = 1$ **to** $2V$ **do**
9: $\quad\quad\quad \Theta''(D(j-1)+i, v) \leftarrow \Theta'(j, i, v)$
10: $\quad\quad$ **end for**
11: $\quad$ **end for**
12: **end for**
13: {Step 3.}
14: $\Omega \leftarrow \text{K\_MEANS}(\Theta'', k_1, d_{clust})$
15: {Step 4.}
16: **for** $j \in TR$ **do**
17: $\quad$ **for** $i = 1$ **to** $D$ **do**
18: $\quad\quad \Pi''(i, j) \leftarrow 1$
19: $\quad\quad mindist \leftarrow d_{clust}(\Omega(1), \Theta'(j, i, :))$
20: $\quad\quad$ **for** $c \in \{2, 3, \ldots, k_1\}$ **do**
21: $\quad\quad\quad curdist \leftarrow d_{clust}(\Omega(c), \Theta'(j, i, :))$
22: $\quad\quad\quad$ **if** $curdist < mindist$ **then**
23: $\quad\quad\quad\quad \Pi''(i, j) \leftarrow c$
24: $\quad\quad\quad\quad mindist \leftarrow curdist$
25: $\quad\quad\quad$ **end if**
26: $\quad\quad$ **end for**
27: $\quad$ **end for**
28: **end for**
29: {Step 5.}
30: **for** $j \in TR$ **do**
31: $\quad B(j, :) \leftarrow \text{FREQ\_TABLE}(\Pi''(:, j))$
32: **end for**
33: {Step 6.}
34: **for** $l = 1$ **to** $k_1$ **do**
35: $\quad$ **for** $c = 1$ **to** $C$ **do**
36: $\quad\quad \Gamma(c, l) \leftarrow 0$
37: $\quad\quad N \leftarrow 0$
38: $\quad\quad$ **for** $j \in TR$ **do**
39: $\quad\quad\quad$ **if** $c_j = c$ **then**
40: $\quad\quad\quad\quad N \leftarrow N + 1$
41: $\quad\quad\quad\quad \Gamma(c, l) \leftarrow \Gamma(c, l) + B(j, l)$
42: $\quad\quad\quad$ **end if**
43: $\quad\quad$ **end for**
44: $\quad\quad \Gamma(c, l) \leftarrow \Gamma(c, l)/N$
45: $\quad$ **end for**
46: **end for**
47: {$k$NN training with features $\Theta'$, classes $\Omega$, $k$ parameter equal to $k_2$ and distance function $d_{kNN}$.}
48: KNN_MODEL $\leftarrow$ KNN_TRAINING($\Theta'$, $\Omega$, $k_2$, $d_{kNN}$)
49: **return** $\Gamma$, KNN_MODEL

**Algorithm 3:** Training algorithm.

**Require:** new survey $\Theta_i = (x_i^v)_{v=1}^V$, $i \in \{1, 2, \ldots, D\}$; centroids $\Gamma(c, l)$, $c \in \{1, 2, \ldots, C\}$, $l \in \{1, 2, \ldots, k_1\}$; clustering parameter $k_1$; $k$NN model; distance function $d_{VSM}$

{Step 1.}
$\Theta' \leftarrow$ TRANSFORM($\Theta_i$)
{Step 2.}
**for** $i = 1$ **to** $D$ **do**
  $\Pi''_{new}(i) \leftarrow$ KNN_PREDICT(KNN_MODEL, $\Theta'_i$)
**end for**
{Step 3.}
$B_{new} \leftarrow$ FREQ_TABLE($\Pi''_{new}$)
{Step 4.}
$c_{new} \leftarrow 1$
$mindist \leftarrow d_{VSM}(\Gamma(1, :), B_{new})$
**for** $c \in \{2, 3, \ldots, C\}$ **do**
  $curdist \leftarrow d_{VSM}(\Gamma(c, :), B_{new})$
  **if** $curdist < mindist$ **then**
    $c_{new} \leftarrow c$
    $mindist \leftarrow curdist$
  **end if**
**end for**
**return** $c_{new}$

**Algorithm 4:** Prediction algorithm.