Class-Balanced Siamese Neural Networks

Samuel Berlemont, Grégoire Lefebvre, Stefan Duffner, Christophe Garcia

Class-Balanced Siamese Neural Networks

Samuel Berlemont, Grégoire Lefebvre[1]

Orange Labs, R&D, 28 chemin du vieux chêne, 38240 Meylan, France

Stefan Duffner and Christophe Garcia

Université de Lyon, INSA, LIRIS, UMR 5205, 17, avenue J.Capelle, 69621 Villeurbanne, France

Abstract

This paper focuses on metric learning with Siamese Neural Networks (SNN). Without any prior, SNNs learn to compute a non-linear metric using only similarity and dissimilarity relationships between input data. Our SNN model proposes three contributions: a tuple-based architecture, an objective function with a norm regularisation and a polar sine-based angular reformulation for cosine dissimilarity learning. Applying our SNN model for Human Action Recognition (HAR) gives very competitive results using only one accelerometer or one motion capture point on the Multimodal Human Action Dataset (MHAD). Performances and properties of our proposals in terms of accuracy, convergence and complexity are assessed, with very favourable results. Additional experiments on the "Challenge for Multimodal Mid-Air Gesture Recognition for Close Human Computer Interaction" Dataset (ChAirGest) confirm the competitive comparison of our proposals with state-of-the-arts models.

Keywords: Siamese Neural Networks, Metric Learning, Human Action Recognition.

1. Introduction

As consumer devices become more and more ubiquitous, new interaction solutions coping with the large variations and noise in unconstrained environments are required . In this study, we explore the use of Neural-Network-based approaches for similarity metric learning in the field of Human Activity Recognition (HAR). While video-based studies propose state-of-the-art performance in HAR, with strategies tackling multi-view learning [1] or low-resolution [2]; and neural networks

[1]Corresponding author: gregoire.lefebvre@orange.com

we focus here on inertial sensor data, with activities holding a semantic value such as *climbing*, *running* or *jumping*. In this context of motion-data-based activity or gesture recognition, three main approaches exist. The earliest methods suggest to model the temporal structure of a gesture class, for example with Hidden Markov Models (HMM) [3]; while another approach consists in matching gestures with reference instances using a non-linear temporal alignment technique generally based on Dynamic Time Warping (DTW) [4]. Finally, higher-level features, designed to be invariant to noise and deformations to some extent, can be extracted from gesture signals in order to train specific classifiers, such as Support Vector Machines (SVM) [5]. For the real-time applications that we aim at here, speed and delay constraints are critical, leading us to the choice of neural-based models. While Extreme Learning Machines (ELM) [6], Bi-Directional Long Short-Term Memory (BLSTM) [7] and Convolutional Neural Networks (CNN) [8] have already been investigated, the main issue is to tackle an open-world problem, which does not only require a good classification performance but also an excellent generalization capability and the possibility to reject unknown classes and add new classes without retraining the existing model. Our work focuses thus on similarity metric learning between gesture sample signatures using the "Siamese" architecture, which aims at modelling semantic relationships between classes to extract discriminating features. Applied to the Single/Multiple Feed Forward Neural Network (FNN), this approach is called Siamese Neural Network (SNN). It differs from other neural-network based similarity metric models where neural networks are essentially used for dimensionality reduction [9], as the SNN is directly trained on semantic similarity information. In this study, after a pre-processing step where the data is filtered and normalized spatially and temporally, the SNN computes a higher-level representation of the gesture, where features are collinear for similar gestures, and orthogonal for dissimilar ones. We propose three contributions to the SNN:

1) We opt for a training strategy that does not require additional parameter fine-tuning. In contrast to classical input set selection strategies like pairs of examples: {similar, dissimilar} or triplets: {reference, similar, dissimilar}, not directly suited to multi-class problems, we propose to include at the same time a similar sample as well as samples from every available dissimilar class, resulting in a better structuring of the output space.

2) While the original SNN model gives good results for classification, multiple mathematical problems are identified, proving that its discrimination potential is not developed to its best capacity. Consequently, during training, we apply a regularization on the outputs to better condition

the objective function.

3) We introduce the notion of *polar sine* to redefine the angular problem as the maximization of a normalized volume induced by the outputs of the reference and dissimilar samples, which results in a non-linear discriminant analysis. Indeed, this approach is the only one to take into account the information about every dissimilarity relationship within the training sets, and not only the relationships involving the reference sample.

With our experimental study on real world datasets, *i.e.* the Multimodal Human Activity Database (MHAD) [10] and the Challenge for Multimodal Mid-Air Gesture Recognition for Close Human Computer Interaction Dataset (ChAirGest), we show very competitive results compared to the state-of-the-art for our proposed SNN configurations.

This paper is organized as follows: Section 2 presents related work on SNNs. In Section 3, we propose our contributions. Then, Sections 4 and 5 describe our experimental protocols and results. A computational analysis of our proposals are discussed in Section 6. Finally, we draw our conclusions and present some perspectives in Section 7.

2. Siamese Neural Networks

A SNN learns a non-linear similarity metric, and essentially differentiates itself from classical neural networks by its specific training strategy involving sets of samples labelled as similar or dissimilar. The capabilities of different SNN-based methods depend on four main points: the network architecture, the training set selection strategy, the objective function, and the training algorithm [11]. In the following, we will explain the three first points in more detail as they are most related to our contributions.

2.1. Architecture

The name "Siamese" historically comes from the necessity to collect the state of a single network for two different activation samples during training. It can be seen as using two identical, parallel neural networks NN sharing the same set of weights W(see Fig. 1). These sub-networks each receive an input sample $\mathbf{X}$, and produce output feature vectors $\mathbf{O_X}$ that are supposed to be *close* for samples from the same class and *far apart* for samples from different classes, according to some distance measure, such as the cosine similarity metric. During the training step, the identical networks are joined at their ends by an output layer whose objective function E_W is defined by a

distance between features from each of the input samples. Siamese networks mainly involve two types of networks: CNN and MLP (Multi-Layer Perceptron).

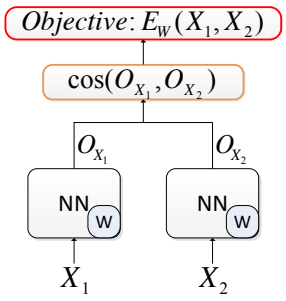


Figure 1: Original SNN training architecture.

Bromley *et al.* [11] introduced the Siamese architecture in 1994, using a Siamese CNN with two sub-networks for a signature verification system handling time-series of hand-crafted low-level features. In 2005, Chopra, Hadsell and LeCun [12] formalized the Siamese architecture applying a CNN on raw images for face verification, before adapting it to a dimensionality reduction technique [13]. More recently, Siamese CNNs have been used successfully for various tasks, such as person re-identification [14], speaker verification [15], and face identification [16].

CNN-based architectures are more specific to image inputs, and several research works propose to use feed-forward perceptrons to handle more general vector inputs. For example, Yih *et al.* [17] apply SFNNs to learn similarities on text data, Bordes *et al.* [18] on entities in Knowledge Bases, and Masci *et al.* [19] on multi-modal data. Other works on face verification use MLP Siamese architectures [20, 21] or Restricted Boltzmann Machines (RBM) [22] to learn a *non-linear* similarity metric.

2.2. Training Set Selection

The selection strategy for training examples depends mostly on the application and the kind of knowledge about similarities that one wants to incorporate in the model. For many applications, such as face or signature verification, the similarity between samples depend on their "real-world" origin, *i.e.* faces/signatures from the same person, and the network allows to determine the genuineness of a test sample with a binary classification. Most approaches use pairs of training samples $(\mathbf{X}_1, \mathbf{X}_2)$ and a binary similarity relation which takes different values for similar and dissimilar

Table 1: A summary of notations used in this paper.

Notation	Description
Y	Similarity label for a pair of samples (similar,dissimilar)
$t(Y)$	Target value for the similarity metric for a pair of samples with a similarity label Y
$C = \{C_1, .., C_K\}$	Set of K classes
$\mathcal{N}_k = [1, .., K] \setminus \{k\}$	Set of class indexes different from class k
$\mathbf{X}_j, \mathbf{O}_{X_j}$	sample $\mathbf{X}$ from class j and its SNN output NB: $\mathbf{X}_1, \mathbf{X}_2$ may be used as an example pair of random samples
$T_k = \{\mathbf{R}_k, \mathbf{P}_k, \{\mathbf{N}_l, l \in \mathcal{N}_k\}\}$	Tuple training set for an SNN update with a reference sample $\mathbf{R}$ from class k, a similar sample $\mathbf{P}$, and a set of dissimilar samples
W	Set of weights for the SNN
E_W	Error function applied for the SNN training

pairs. Lefebvre *et al.* [20] expand the information about expected neighbourhoods, and suggest a more symmetric representation: by considering a reference sample $\mathbf{X}_R$ for each known relation, it is possible to define triplets $(\mathbf{X}_R, \mathbf{X}_+, \mathbf{X}_-)$, with $\mathbf{X}_+$ forming a genuine pair with the reference $\mathbf{X}_R$, while $\mathbf{X}_-$ is a sample from another class.

2.3. Objective Functions

The objective function computes a similarity metric between the higher-level features extracted from multiple input patterns. Minimizing this function iteratively during training ensures that the distance between similar patterns gets smaller, and the one between dissimilar gets larger. In this regard, different metrics have been used in the literature:

Square Error. Given a network with weights W and two samples $\mathbf{X}_1$ and $\mathbf{X}_2$ with their labels Y, a target $t(Y)$ is defined for the cosine value between the two respective output vectors $\mathbf{O}_{X_1}$ and $\mathbf{O}_{X_2}$ as "1" for similar pairs and "-1" (or "0") for dissimilar pairs [11]:

$$E_W(\mathbf{X}_1, \mathbf{X}_2, Y) = \left(t(Y) - \cos(\mathbf{O}_{X_1}, \mathbf{O}_{X_2})\right)^2 . \quad (1)$$

Triangular Similarity Metric. Zheng *et al.* [23] imply the same targets and impose additional constraints on the norms of the output vectors $\mathbf{O}_{X_1}$ and $\mathbf{O}_{X_2}$. Integrating a geometrical interpretation using the triangle inequality, the resulting objective function becomes:

$$E_W(\mathbf{X}_1, \mathbf{X}_2, Y) = \frac{1}{2} \|\mathbf{O}_{X_1}\|^2 + \frac{1}{2} \|\mathbf{O}_{X_2}\|^2 - \|\mathbf{O}_{X_1} + t(Y)\, \mathbf{O}_{X_2}\| + 1 \; . \tag{2}$$

Triplet Similarity. Lefebvre *et al.* [20] use simultaneously targets for genuine and impostor pairs by presenting triplets of a reference $\mathbf{X}_R$, a positive $\mathbf{X}_+$ and a negative $\mathbf{X}_-$ sample. The output of the positive pair $(\mathbf{O}_R, \mathbf{O}_+)$ is trained to be collinear, whereas the output of the negative pair $(\mathbf{O}_R, \mathbf{O}_-)$ is trained to be orthogonal. Thus:

$$E_W(\mathbf{X}_R, \mathbf{X}_+, \mathbf{X}_-) = (1 - \cos(\mathbf{O}_R, \mathbf{O}_+))^2 + (0 - \cos(\mathbf{O}_R, \mathbf{O}_-))^2 \tag{3}$$

Deviance. Yi *et al.* [14] use the binomial deviance to define their objective function:

$$E_W(\mathbf{X}_1, \mathbf{X}_2, Y) = \ln\left(\exp^{-2t(Y)\, \cos(\mathbf{O}_{X_1}, \mathbf{O}_{X_2})} + 1\right) \tag{4}$$

Two Pairs. Yih *et al.* [17] consider two pairs of vectors, $(\mathbf{X}_{p1}, \mathbf{X}_{q1})$ and $(\mathbf{X}_{p2}, \mathbf{X}_{q2})$, the first being known to have a higher similarity than the second. The main objective is then to maximize

$$\Delta = \cos(\mathbf{O}_{X_{p1}}, \mathbf{O}_{X_{q1}}) - \cos(\mathbf{O}_{X_{p2}}, \mathbf{O}_{X_{q2}}) \tag{5}$$

in a logistic loss function

$$E_W(\Delta) = \log(1 + \exp(-\gamma\Delta)) \; , \tag{6}$$

with γ being a scaling factor.

Probability-driven. Nair *et al.* [22] add a final neuron to their architecture whose activation function computes the probability P of two samples $\mathbf{X}_1$, $\mathbf{X}_2$ being from the same class:

$$P = \frac{1}{1 + \exp(-(w.\cos(\mathbf{O}_{X_1}, \mathbf{O}_{X_2}) + b))} \; , \tag{7}$$

with w and b being scalar parameters.

Norm-Based. Several works [12, 13, 16, 19] propose to use the norm, *e.g.* ℓ_2-norm, between the output vectors as a similarity measure:

$$d_W(\mathbf{X}_1, \mathbf{X}_2) = \|\mathbf{O}_{X_1} - \mathbf{O}_{X_2}\|_2 \; . \tag{8}$$

For example, Chopra *et al.* [12] define an objective composed of an "impostor" I (t(Y)=1) and a "genuine" G term (t(Y)=0):

$$E_W(\mathbf{X}_1, \mathbf{X}_2, Y) = (1 - t(Y))E_W^G(\mathbf{X}_1, \mathbf{X}_2) + t(Y).E_W^I(\mathbf{X}_1, \mathbf{X}_2) \tag{9}$$

$$\text{with } E_W^G(\mathbf{X}_1, \mathbf{X}_2) = \frac{2}{Q}(d_W)^2, E_W^I(\mathbf{X}_1, \mathbf{X}_2) = 2Qe^{(-\frac{2.77}{Q}d_W)} \text{ , where } Q \text{ is the upper bound of } d_W. \tag{10}$$

Statistical. Chen *et al.* [15] compute the first and second-order statistics, $\mu^{(i)}$ and $\Sigma^{(i)}$, over sliding windows on the SNN outputs of a speech sample i, and define the objective function as:

$$E_W(\mathbf{X}_1, \mathbf{X}_2, Y) = (1 - t(Y))(D_m + D_S) + t(Y).(\exp(\frac{-D_m}{\lambda_m}) + \exp(\frac{-D_S}{\lambda_S})) \, , \tag{11}$$

where

$$D_m = \left\| \mu^{(i)} - \mu^{(j)} \right\|_2^2 \quad , \quad D_S = \left\| \Sigma^{(i)} - \Sigma^{(j)} \right\|_F^2 \tag{12}$$

are incompatibility measures of these statistics between two samples i and j, λ_m and λ_s are tolerance bounds on these measures, and $\|.\|_F$ is the Frobenius norm.

Thus, SNNs differ from classical networks essentially by their training strategy involving multiple samples. However, since they produce "only" a similarity metric in their projection space, it is necessary to use another model on the projected data to obtain the final decision for classification (in the simplest case: a KNN classifier).

3. Proposed SNN

In the following, we propose multiple contributions aiming at increasing the discriminating potential of the network and improve its convergence for multi-class classification or verification. Firstly, we suggest a more general way of defining similarities in multi-class problems, with training sets representing every class at once, since the pair or triplet strategies are not sufficient to define the complex relationships between the different classes. Indeed, they lead to biased and unclear training set selection strategies. Moreover, our mathematical analysis proves that cosine-based objectives can lead to numerical instabilities. Therefore, we introduce a regularization of the original metric, giving a simpler and computationally more efficient function. Finally, we propose to modify the objective function further by including a term based on the *polar sine*, allowing for a very elegant and effective solution to manage every similarity and dissimilarity relationship in the training process.

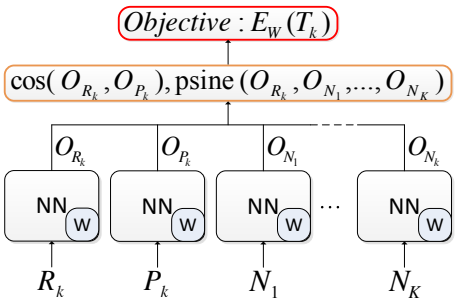


Figure 2: Proposed training architecture of the SNN.

3.1. Architecture

In our study, we use an MLP-based architecture with hyperbolic tangent activation functions for every layer. We choose a squared error cosine-based objective function as in [11] and [20] (see Eq. 1). We set the target value for dissimilar pairs to 0 instead of -1 to avoid instabilities in the training, as this target cannot be met for more than two classes.

However, we do not use the common two-sub-network training architecture, since the latter is greatly modified by a our specific training set selection strategy presented below.

3.2. Training Set Selection Strategy

Existing training set selection strategies for SNNs consists in defining a subjective number of similar and dissimilar pairs deemed representative of the global relationships within the data. Generally, this induces a bias, since it is difficult to ensure a good coverage for every relationship.

For this reason, we propose a unified approach for multi-class problems (see Fig. 2). Let $C = \{C_1, .., C_K\}$ be the set of classes, $\mathbf{O}_{R_k}$ the output vector of the reference sample $\mathbf{R}_k$ from the class C_k presented to the model for update, $\mathbf{P}_k$ the output of a different sample $\mathbf{P}_k$ from the same class (i.e. the *positive* comparison), and $\mathbf{O}_{N_l}$ the output of a sample $\mathbf{N}_l$ from another class C_l (i.e. a *negative* comparison).

In order to keep symmetric roles for every class and to optimize the update efficiency, we propose to minimize an error criterion for tuples. Given a class C_k, and the set $\mathcal{N}_k = [1, .., K] \setminus \{k\}$ of the indexes of every other class, a tuple is defined as $T_k = \{\mathbf{R}_k, \mathbf{P}_k, \{\mathbf{N}_l, l \in \mathcal{N}_k\}\}$ involving one reference sample from the class C_k, one positive sample and one negative sample from every other

class. The objective function becomes:

$$E_{W_1}(T_k) = (1 - \cos(\mathbf{O}_{R_k}, \mathbf{O}_{P_k}))^2 + \sum_{l \in \mathcal{N}_k} (0 - \cos(\mathbf{O}_{R_k}, \mathbf{O}_{N_l}))^2. \tag{13}$$

Thus, our network architecture involves as many sub-networks as classes. Every sample is taken once as a reference, while the others are drawn at random. This facilitates the selection of representative samples and gives a global, non-parametric approach. Moreover, this strategy is similar to mini-batch learning, *i.e.* it limits the number of updates required before convergence.

3.3. Objective Function

3.3.1. Norm Regularization

In order to improve convergence, we also study the behaviour of a weight update over the projected samples. In the following, we will analyse the cosine metric as a function of two vectors of dimension n,

$$cos_{X_1,X_2} : \mathbb{R}^{2n} \to \mathbb{R} \,|\, (\mathbf{X}_1, \mathbf{X}_2) \to \frac{1}{2}(1 - \cos(\mathbf{X}_1, \mathbf{X}_2))^2. \tag{14}$$

Let $(\mathbf{O}_1, \mathbf{O}_2)$ be a pair of outputs used for update. Given the functions

$$\begin{aligned} cos_{O_1} &: \mathbb{R}^n \to \mathbb{R} \,|\, \mathbf{X} \to \frac{1}{2}(1 - \cos(\mathbf{O}_1, \mathbf{X}))^2 \\ cos_{O_2} &: \mathbb{R}^n \to \mathbb{R} \,|\, \mathbf{X} \to \frac{1}{2}(1 - \cos(\mathbf{X}, \mathbf{O}_2))^2 \end{aligned} \tag{15}$$

respectively evaluated at the points $\mathbf{O}_2$ and $\mathbf{O}_1$, the cos_{X_1,X_2} directional derivative at $(\mathbf{O}_1, \mathbf{O}_2)$ can be expressed as the concatenation of the two directional derivatives $\nabla_{cos_{O_1}}(\mathbf{O}_2)$ and $\nabla_{cos_{O_2}}(\mathbf{O}_1)$.

We will show here that every stochastic gradient descent will increase the norms for both samples. Considering the function cos_{O_1}, the update of $\mathbf{O}_2$ is

$$\mathbf{O}_2^{t+1} = \mathbf{O}_2^t - \lambda . \nabla_{cos_{O_1}}(\mathbf{O}_2), \lambda \in \mathbb{R}. \tag{16}$$

Figure 3 gives a graphical illustration in three dimensions. The line directed by the vector $\frac{\mathbf{O}_2}{\|\mathbf{O}_2\|}$ belongs to the equipotential for the $\cos_{O_1}$ function. By definition, we can conclude that the directional derivative $\nabla_{cos_{O_1}}(\mathbf{O}_2)$ is orthogonal to $\mathbf{O}_2$. According to Pythagoras' theorem, we can conclude:

$$\left\|\mathbf{O}_2^{t+1}\right\|^2 = \left\|\mathbf{O}_2^t\right\|^2 + \lambda^2 \left\|\nabla_{cos_{O_1}}(\mathbf{O}_2)\right\|^2 \Rightarrow \left\|\mathbf{O}_2^{t+1}\right\| > \left\|\mathbf{O}_2^t\right\|. \tag{17}$$

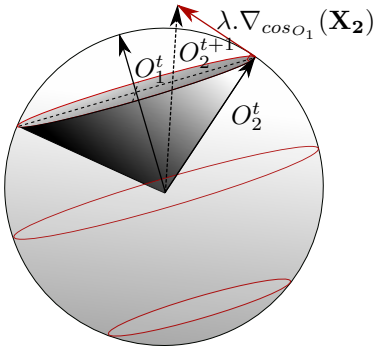


Figure 3: An update step on the projection norm for a pair $(\mathbf{O}_1^t, \mathbf{O}_2^t)$. The sphere centre corresponds to the origin. The grey cone represents the equipotential surface for the function cos_{O_1}.

Increasing the norms of the output vectors may incur progressive divergence. Moreover, with hyperbolic tangent activation functions, the output space is a hyper-cube of dimension n, which restricts the norms to a maximum of $\sqrt{n}$. Therefore, we propose to add constraints on the norms of every output by forcing them to 1 and thus avoid any undesired saturation effects.

We modify our objective function E_{W_1} (see Eq.13) for a training subset T_k:

$$E_{W_2}(T_k) = E_{W_1}(T_k) + \sum_{\mathbf{X}_p \in T_k} (1 - \left\|\mathbf{O}_{X_p}\right\|)^2. \tag{18}$$

Given $\forall(\mathbf{O}_1, \mathbf{O}_2) \in (\mathbb{R}^n, \mathbb{R}^n)$, $\cos(\mathbf{O}_1, \mathbf{O}_2) = \frac{\mathbf{O}_1 \cdot \mathbf{O}_2}{\|\mathbf{O}_1\| . \|\mathbf{O}_2\|}$, we also propose to replace the cosine distance for each pair by the scalar product of the pair outputs, since the norms of the two outputs are set to one during training.

Thus, the final objective function for one training subset T_k is defined as:

$$E_W(T_k) = (1 - \mathbf{O}_{R_k} \cdot \mathbf{O}_{P_k})^2 + \sum_{l \in \mathcal{N}_k} (0 - \mathbf{O}_{R_k} \cdot \mathbf{O}_{N_l})^2 + \sum_{\mathbf{X}_p \in T_k} (1 - \left\|\mathbf{O}_{X_p}\right\|)^2. \tag{19}$$

While this formulation is more suited to handle angular updates, it turns out impractical for an increasing number of classes. Moreover, in the derivative of the mean squared error objective, the cosine error is weighted by the difference between the target and the cosine value which tends to zero and slows down the convergence. Thus, we propose a new error function that preserves the targets, while addressing both of these problems.

3.3.2. Angle Problem Reformulation

In the following, we propose a reformulation of the objective function based on a higher-dimensional dissimilarity measure, the polar sine metric. We will then show that this new analysis leads to a non-linear discriminant analysis.

Inspired by the 2D sine function, Lerman *et al.* [24] define the polar sine for a set $V_m = \{\mathbf{v}_1, \ldots, \mathbf{v}_n\}$ of m-dimensional linearly independent vectors ($m > n$) as a normalized hyper-volume. Given $\mathbf{A} = \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_m \end{bmatrix}$ and its transpose $\mathbf{A}^\top$:

$$PolarSine(\mathbf{v}_1, \ldots, \mathbf{v}_n) = \frac{\sqrt{\det\left(\mathbf{A}^\top \mathbf{A}\right)}}{\prod_{i=1}^{n} \|\mathbf{v}_i\|}. \tag{20}$$

In the special case where $m = n$, the matrix product in the determinant is replaced by the square matrix $\mathbf{A}$

Given the matrix $\mathbf{S}$ so that $\forall (i,j) \in [1,..,n]^2, \mathbf{S}_{i,j} = \cos(\mathbf{v}_i, \mathbf{v}_j)$, this measure can be rewritten as $PolarSine(\mathbf{v}_1, \ldots, \mathbf{v}_n) = \sqrt{\det\left(\mathbf{S}\right)}$. For numerical stability reasons during the derivation process and to make this value independent from the number of classes, we introduce the polar sine metric:

$$psine(\mathbf{v}_1, \ldots, \mathbf{v}_n) = \sqrt[n]{\det\left(\mathbf{S}\right)}. \tag{21}$$

3.3.3. Polar sine for learning dissimilarities

The polar sine metric only depends on the angles between every vector of the set. It reaches its maximum value when all the vectors are orthogonal, and thus can be used as another metric for dissimilarity.

With two comparable similarity estimators whose values are between 0 and 1, one for similar and one for dissimilar samples, it is now possible to redefine the objective function for our training sets T_k:

$$\begin{aligned} E_{W_3}(T_k) &= Esim_W(T_k) + E\overline{sim}_W(T_k) \ , \\ Esim_W(T_k) &= (1 - \cos(\mathbf{O}_{R_k}, \mathbf{O}_{P_k}))^2, \\ E\overline{sim}_W(T_k) &= (1 - psine(\mathbf{O}_{R_k}, \mathbf{O}_{N_1}, \ldots, \mathbf{O}_{N_K}))^2 \ . \end{aligned} \tag{22}$$

Optimizing the polar sine corresponds to assigning a target of 0 to the cosine value of every pair of outputs from different vectors drawn in $T_k \setminus \{\mathbf{R}_k\}$, i.e. we assign a target for every pair of dissimilar samples. This actually holds more information than our original cosine or scalar objective functions, which would only define a target for pairs including the reference sample. As a

consequence, the *psine* function allows for a truly complete representation, in every training set, of every available relationship present in the dataset. Given a set number of sources, the initial inputs are transformed into maximally independent, multi-dimensional components. Thus, our Siamese network, combined with this new objective function presents all the properties of a supervised, stochastic non-linear Independent Component Analysis, with one more advantage, as the number of components is adjustable by modifying the network output layer structure.

4. Experiments on the Multimodal Human Action Dataset

We state four hypotheses H_1 to H_4 about our contributions that we will experimentally validate on a small dataset, the Multimodal Human Action Dataset (MHAD):

- H_1: an *orthogonality* objective for the cosine value between negative samples pairs leads to a better, more stable convergence.
- H_2: a *tuple-based* training set selection strategy allows for a better representation of the relationships between classes, leading to a better structuring of the output space.
- H_3: the proposed *regularization* scheme leads to a better convergence and a more stable norm evolution.
- H_4: the *polar sine metric*-based objective, through a non-linear discriminant analysis, is more efficient at separating classes than other objective functions.

4.1. Dataset presentation and protocols

The Multimodal Human Action Dataset [10] comprises the recordings from 12 participants performing 11 different actions (with 5 repetitions) which were designed to cover diverse dynamics of different body extremities and which were not specifically explained to the subjects beforehand in order to collect a representative range of different styles for each action. Figure 4 illustrates the different action types.

Although the MHAD comprises data collected from multiple types of sensors, we focus our study on two main sensors, consisting in the right wrist inertial sensor (A_1) and motion capture (M_{20}), as it gave the best classification results.

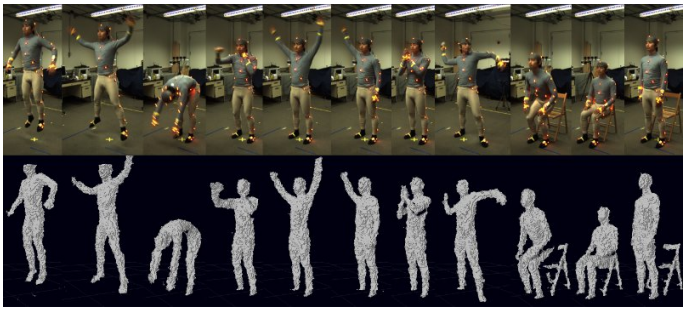


Figure 4: MHAD actions: *jumping*, *jumping jacks*, *bending*, *punching*, *waving two hands*, *waving one hand*, *clapping*, *throwing*, *sit down/stand up*, *sit down*, *stand up*.

A general amplitude scaling is performed, where each component of every sample forming a action record is divided by the maximum norm over all the samples of this action. Then, a low-pass filter is applied to the inertial signals to filter out the involuntary small shakes and electronic noise. Finally, the data are temporally normalized, with specific strategies suited to each method to compare. For the methods requiring a fixed-size input, the three-dimensional 30Hz inertial signal for each action record is re-sampled to a 135-dimensional (45×3) sample according to the estimated curvilinear length of the signal.

The same network architecture is selected for every SNN variant: a 3-layer SFNN, with an input layer size adapted to the data dimensionality (135 neurons for MHAD), a 45-neuron hidden layer, and a 90-neuron output layer. The hyperbolic tangent is chosen as the activation function for every neuron, and the learning rate is set to 0.001. During training, the network is independently and successively activated with every sample of a training set. Each activation state is stored, and reused to compute the weight updates. The computation details are available in [25].

The following aspects are explored:

4.1.1. Negative Target Selection (Hypothesis H_1)

The target cosine value for dissimilar pairs can be set to -1 or to 0. We propose an analysis of both choices. Indeed,the latter allows for a better convergence as the minimum for the error function (0) can actually be attained, and it limits the space occupied by the samples. However, it also requires the output space dimension to be greater or equal to the number of classes represented in the training set, at the risk of overstraining the system and causing instability.

4.1.2. Training Set Selection Strategy - pairs, triplets or tuples (Hypothesis H_2)

We compare three different SNN selection strategies, and assess their relative complexities and computation costs. Let N_c be the number of classes, and N_s the number of samples in the dataset.

The first strategy is the most common [11] and consists in selecting a specific number of pairs of samples, labelled as "similar" or "dissimilar". Here, we propose to form one similar pair with every sample of the dataset, where the second element of the pair is drawn at random. Dissimilar pairs are selected so as to represent every dissimilarity relationship available for every sample of the dataset, for a total of $N_c - 1$ dissimilar pairs per sample. The total number of updates for one epoch is then equal to $N_c \times N_s$, with $2(N_c \times N_s)$ activations.

The second strategy [20] favours a symmetry between similarities and dissimilarities, forming triplets, with one reference, one similar and one dissimilar sample per set. As for the first strategy, every dissimilarity relationship is represented. As a consequence, a triplet is formed for each dissimilarity, for a total of $N_c - 1$ positive and negative pairs per sample. As such, the total number of updates is equal to $(N_c - 1) \times N_s$, with three activations per update.

The third strategy consists in our contribution, which generalizes the triplets approach with tuples. Each sample is used as a reference once, in a training set representing every similarity and dissimilarity relationships. This limits the number of updates per epoch, with only N_s updates, for $[(N_c + 1) \times N_s]$ activations.

4.1.3. Cost Function: (Hypotheses H_3, H_4)

Our study focuses on the comparison between the initial cosine-based error function, referred as *cos*, and our two contributions: the norm regularization, resulting in the scalar product-based objective function, referred as *scal*, and the polar sine metric-based function, referred as *psine*.

These three points lead to a total of 15 different SNN configurations (where the *psine* function is incompatible with -1 targets for negative examples). Each SNN is complemented by a KNN classifier, with $K = 1$, and the classification accuracy is reported. A leave-one-out strategy is adopted, with every sample from 11 users selected for training, and the samples from the last user for the testing phase.

4.1.4. Comparison with state-of-the-art models

We propose a comparison between the *psine*-based SNN and multiple gesture recognition approaches: geometric-based methods with Dynamic Time Warping (DTW) [4], classifier-based meth-

Table 2: Recognition rates for our MHAD protocols with different training set selection strategies (pairs, triplets, tuples) and different negative target values (1 for positives, -1 for negatives: "t-11", and 0 for negatives: "t01").

A1	pairs		triplets		tuples	
targets	t-11	t01	t-11	t01	t-11	t01
cos	0.901 ± 0.11	**0.916 ± 0.10**	0.881 ± 0.10	**0.916 ± 0.09**	0.880 ± 0.09	0.915 ± 0.10
scal	0.883 ± 0.09	0.869 ± 0.09	0.871 ± 0.11	0.869 ± 0.11	0.880 ± 0.08	0.886 ± 0.08
psine	-	0.910 ± 0.10	-	0.910 ± 0.08	-	**0.918 ± 0.09**

ods with the classical Support Vector Machine (SVM) [5], and neural-based methods with the MLP.

We focus on two sensors: the accelerometer A_1 and the motion capture sensor M_{20}, with the same leave-one-out strategy as described above.

4.2. Results

We subsequently discuss the results of each protocol:

4.2.1. Negative Target Selection

Table 2 shows the classification accuracy for the different configurations. For the cosine-based objective function, an orthogonality between negative vectors (target 0) produces better overall results for every training set selection strategy than the repulsion scheme (target -1). Only the scalar-based objective function with norm regularization gives a slight advantage to the -1 target for the negative pairs and triplets-based strategies. This phenomenon can be explained by the decomposition of the target angle in three sub-targets. The norm objectives have to be met perfectly for the scalar product objective to propose the right angle correction. These norms dynamically change with each update, and norm corrections may be favoured over angle corrections. Thus, a target equal to -1 amplifies the scalar product error, and gives an extra edge to the amplitude of the corresponding objective error over the norm objectives error, speeding up the convergence.

Thus, hypothesis H_1 is validated for the cosine and polar sine metric-based objective functions.

4.2.2. Training Set Selection Strategy

The training set selection strategy does not impact significantly the classification scores for the cosine-based objective function, with similar scores around 91.5% for the three strategies combined with a zero negative target. It is however interesting to note a reduced standard deviation for the

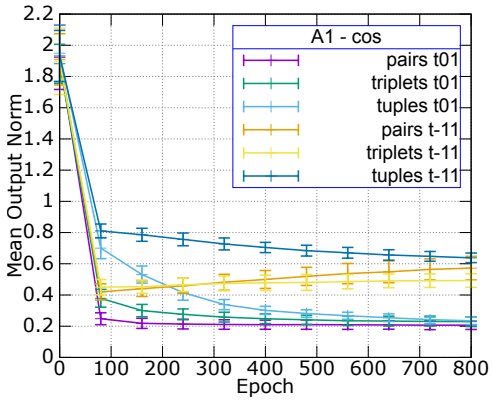
triplet set strategy with a cosine-based objective, due to the increased representation of the similarity relationships which reduces the intra-class distances. Our proposed tuple selection strategy gives the best results on average.

Finally, the polar sine metric-based SNN shows a good performance in every case, comparable to the cosine-based SNN. While the pair and triplet strategies give the advantage to the cosine-based objective function, the potential of the polar sine metric approach is revealed with tuples, which cover every relationship available during training. This method shows then the best accuracy, equal to 91.8%, with a reduced standard deviation of 9.1% over the cosine-based one, whose accuracy is equal to 91.5% with a standard deviation of 10.2%. Thus, we can conclude that the polar sine metric-based objective gives competitive results, as it corrects the cosine values more reliably than the scalar product-based one. This analysis validates hypothesis H_2, with a tuple-based strategy producing better or similar results as other strategies for the three objective functions. Moreover, H_4 is validated as well, as the polar sine metric-based objective combined with a tuple-based strategy produces the best classification score. However, the resulting projections differ for every objective functions, as seen below.

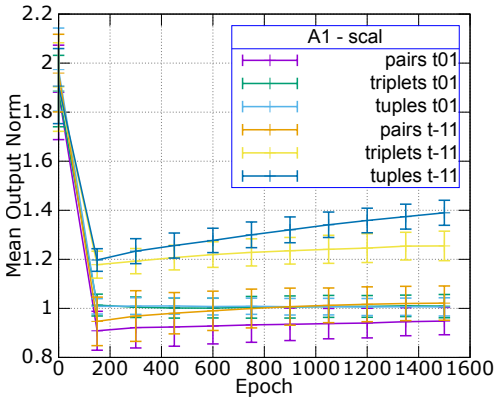
4.2.3. Projection Space Analysis

As explained in the previous section, a gradient descent update with the cosine function leads to instabilities in the norms of the outputs. Indeed, for a model where the gradient descent is simply performed on the components of the output vectors, we proved that these norms increase with each update. Figures 5a, 5b and 5c show the evolution of the mean norms of the output and their corresponding standard deviations for the different negative targets and training set strategies, respectively combined with a cosine-based, scalar product-based and polar sine metric-based objective function.

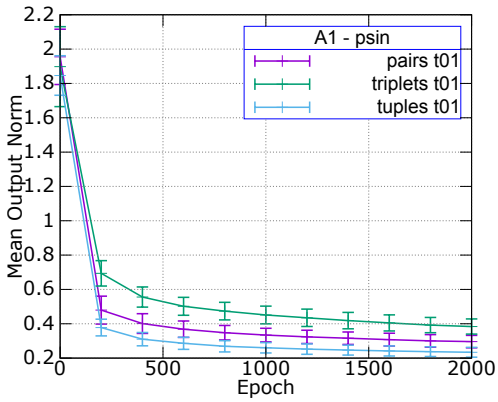
First, one can observe that "-1" negative targets introduce an instability, which leads to the expected increasing norms for the pairs and triplets strategies, for both cosine and scalar product-based objective functions. Unexpectedly, with "0" negative targets, the norms tend to decrease quickly at first, then slightly but steadily with each update, for both cosine (see Fig. 5a) and polar sine metric-based (see Fig. 5c) objective functions. The scalar product-based strategy proves more reliable in that case (see Fig. 5b), although it implies a longer training with 1600 epochs. Indeed, it is devised to counteract this norm variation phenomenon, and proves to be efficient with



(a) cos



(b) scal

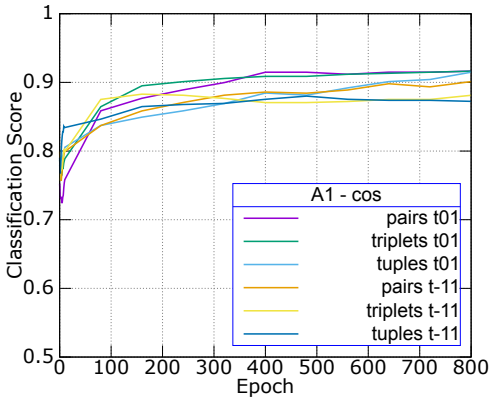


(c) psine

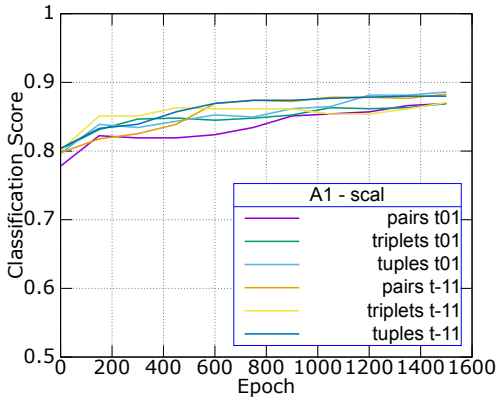
Figure 5: Mean output norm evolution for each objective function.

mean output norms close to one and small standard deviations for the triplets and tuples strategies. While the unit mean norm is not reached for the pairs strategy, it still stabilizes the norm evolution, validating hypothesis H_3, except for tuple and triplet objectives combined with a -1 negative target whose mathematical instabilities lead to increasing norms.

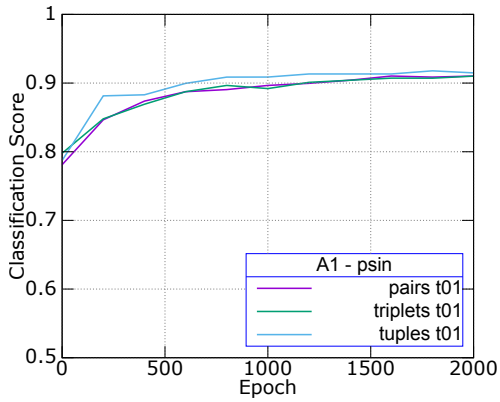
As a conclusion, angle updates with SNN networks imply a compromise between efficient angle corrections and output norms stabilization. And the observed adverse effects of some the studied SNN variants are most likely accentuated when applied on larger scale problems.



(a) cos



(b) scal



(c) psine

Figure 6: Classification rate evolution for each objective function.

Table 3: Comparison with the state of the art on MHAD.

	A_1	M_{20}
DTW [26]	0.790 ± 0.107	0.888 ± 0.076
MLP [26]	0.818 ± 0.099	0.913 ± 0.067
SVM [27]	0.867	-
cos	0.915 ± 0.102	0.906 ± 0.080
scal	0.886 ± 0.079	0.910 ± 0.065
psine	$\mathbf{0.918 \pm 0.091}$	$\mathbf{0.924 \pm 0.068}$

4.3. Comparison with state-of-the-art results

Finally, we compare the performances of each SNN objective function variant with a tuple-based training selection strategy, denominated *cos*, *scal* and *psine*. The results are shown in Table 3. For our analysis, we focus on the classification rates obtained from isolated sensor data with the accelerometer A_1 and the motion capture sensor M_{20}, and we report the results for the DTW and MLP methods proposed in [26] and the SVM-based approach [27] on these same sensors.

The SNN-based approaches show globally superior results on the inertial sensor A_1, with a lowest score of 88.6% for the SNN-scal, compared to a best score in the literature of 86.7% for the SVM approach. This shows that a SNN-based approach is very competitive. This conclusion is verified for the M_{20} sensor. Our SNN-psine approach, implementing the polar sine metric and tuple-based set selection strategies contributions, show the best result of 92.4%.

5. Experiments on the ChAirGest dataset

We make a final hypothesis H_5, which we seek to confirm the larger scale, "Challenge for Multi-modal Mid-Air Gesture Recognition for Close Human Computer Interaction" Dataset (ChAirGest) [28]:

- H_5: the performance and competitive results of our psine-based SNN contribution can be generalized to other datasets.

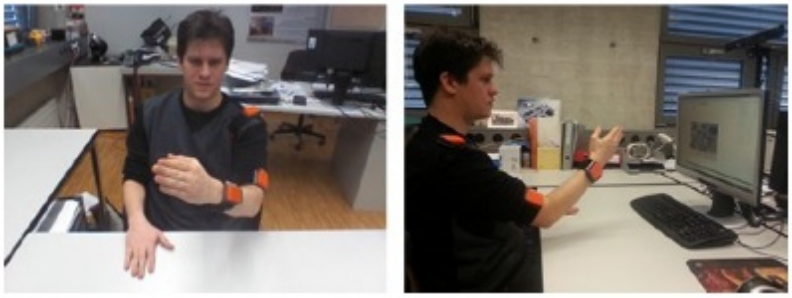


Figure 7: Visuals of the ChairGest recording setup [28]. On the left, an image captured from the Kinect RGB stream. On the right, an external view of a subject performing a gesture in pseudo-recording conditions. Note that the accelerometers were hidden under the subject's clothes to avoid visual clues.

5.1. Dataset presentation and protocol

This dataset contains 6 hours of continuous multimodal recordings from 10 subjects mimicking gestures seen on a computer screen. The data have been acquired from a Kinect camera and 4 Inertial Motion Units (IMUs) attached to the right arm of the subject (see Fig.7). The dataset contains the 10 following gestures: *Swipe left*, *Swipe right*, *Push to screen*, *Take from screen*, *Palm-up rotation*, *Palm-down rotation*, *Draw a circle I*, *Draw a circle II*, *Wave hello* and *Shake hand*.

In the following experiment, we use a concatenation of 3 Kinect points, K_2, K_6 and K_{10}, respectively associated with the head, the right hand, and the left hand. A 10-cross-validation testing is performed, with 400 samples drawn at random as the training set, 50 samples for the validation set, and 50 samples for the test set.

The same network architecture is used for our SNN in this experiment: a 3-layer SFNN, with an input layer size adapted to the data dimensionality (900 neurons for ChAirGest), a 45-neuron hidden layer, and a 90-neuron output layer.

We propose to compare our psine-based SNN to the three HAR state-of-the-art models previously identified on MHAD: DTW, MLP and SVM.

5.2. Results

This trend identified in our MHAD experiments is indeed confirmed on the ChAirGest dataset, see Table 4. While the MLP lacks the generalization potential to take into account the inter-subject variability with a classification rate of 87.8%, other methods such as the SVM or the KNN-based ones perform better, with a 89.2% rate for the DTW and 90.0% rate for the SVM. Our SNN-psine

Table 4: Comparison with the state of the art on ChAirGest.

	$K_2K_6K_{10}$
DTW	0.892 ± 0.018
MLP	0.878 ± 0.038
SVM	0.900 ± 0.046
SNN-psine	$\mathbf{0.910 \pm 0.034}$

Table 5: Number of relationships per update given a reference, with N_c the number of classes.

	pairs	triplets	tuples
cos	1	2	N_c
scal	1	2	N_c
psine	1	2	$\frac{N_c(N_c-1)}{2} + 1$

achieves the best result, with a classification rate of 91.0%. This score is competitive with the best scores available to this day on this dataset, in particular Cao *et al.* [29] who obtain a classification rate of 0.918% with their HMM decision fusion strategy. It is however important to note that these results are not directly comparable as they use the whole set of Kinect points, contrary to the 3 points we chose, with a different testing protocol, based on a leave-one-out cross validation.

6. Computational Complexity

Finally, we assess the computational cost of the three proposed training set selection strategies.

Table 5 shows the difference in the number of relationships represented in each update. The tuple-based objective function has a much more balanced representation, thus requiring more memory but at the same time reducing the number of necessary updates.

Table 6 illustrates this trade-off between the memory requirements, the number of back-propagations and updates necessary to present every available relationship to the model without bias, given a reference sample. Furthermore, there is a second trade-off between the required number of updates and the computational complexity of each update (as illustrated in Fig. 6a-6c). The polar sine metric-based SNN requires around twice the number of epochs of the cosine-based SNN to reach its optimum configuration.

Table 6: Number of updates for each strategy given a reference, with N_c the number of classes.

	pairs	triplets	tuples
networks/update	2	3	$N_c + 1$
backprop./update	$2N_c$	$3N_c$	$N_c + 1$
updates	N_c	$N_c - 1$	1

Table 7: Average Time for one update (in ms) on MHAD.

A1	pairs	triplets	tuples
cos	0.301	0.35	1.61
scal	0.317	0.39	1.72
psine	0.274	0.33	2.05

The cosine and scalar-based objective functions with the tuples strategy imply an error computation complexity which is linear in terms of the output dimension N_O and the number of classes N_c, *i.e.* a complexity in $O(N_O\, N_c)$. While the polar sine metric function involves a matrix inversion which can be done very efficiently for low-dimensional matrices.

Thus, the global complexity of the error resides in the matrix multiplication between the inverse cosine matrix and the normalized output matrix, for a final complexity in $O(N_O\, N_c^2)$. As such, the polar sine metric-based method can stay competitive for the most common small to medium-scale problems. One approach for larger-scale problems would be a decomposition into smaller sets of relationships in order to limit the amplitude of N_c.

Table 7 reports the average training times for one update on MHAD (Computing times on an Intel Core i7-4800MQ CPU@2.70GHz), for each training set selection strategy and objective function. We can see that the order of magnitude identified above are respected, with a triplet strategy taking about 20% more time than the pair strategy, while the tuple strategy increases the update time by more than 400%, which is acceptable as the total number of updates per reference is divided by the number of classes.

7. Conclusion

We proposed an adaptation of the Siamese strategy for neural networks to a multi-class classification context with stochastic training. Beyond the typical pairs and triplets of samples labelled as similar or dissimilar, a generalized training set selection strategy is suggested, which integrates one sample from every class, effectively simplifying the constitution of these training sets and balancing the representation of every relationship. Further, following the choice of the cosine similarity metric for the output space, mathematical flaws are identified, in particular concerning the uncontrolled norms of the outputs during training. Simplifying the original cosine function, angle updates are decomposed into three independent targets for a pair of vectors, namely a target on the scalar product between these vectors, and a target on their respective norms. Finally, we proposed a unified similarity function, the Polar Sine Metric, which holds and represents all the available information about dissimilarities in the training set, and leads to a supervised, stochastic non-linear Independent Component Analysis learning. We performed a thorough analysis of the different proposed strategies and SNN configurations and experimentally showed competitive performance on the task of human action recognition.

Future research may focus on the analysis of the proposed approaches for different tasks and types of data, as well as larger scale problems. Also, the possibility to include a temporal aspect in the model would be of great interest.

References

[1] Z. Gao, G. Zhang, H. Zhang, Y. Xue, G. Xu, 3d human action recognition model based on image set and regularized multi-task leaning, Neurocomputing 252 (C) (2017) 67–76. doi:10.1016/j.neucom.2016.01.126.
URL https://doi.org/10.1016/j.neucom.2016.01.126

[2] Y. Zhao, H. Di, J. Zhang, Y. Lu, F. Lv, Y. Li, Region-based mixture models for human action recognition in low-resolution videos, Neurocomputing 247 (2017) 1 – 15. doi:https://doi.org/10.1016/j.neucom.2017.03.033.
URL http://www.sciencedirect.com/science/article/pii/S0925231217305416

[3] X. Zhang, X. Chen, Y. Li, V. Lantz, K. Wang, J. Yang, A Framework for Hand Gesture Recognition Based on Accelerometer and EMG Sensors, IEEE Transactions on Systems, Man

and Cybernetics, Part A: Systems and Humans 41 (6) (2011) 1064–1076. doi:10.1109/TSMCA.2011.2116004.

[4] K. Barczewska, A. Drozd, Comparison of methods for hand gesture recognition based on Dynamic Time Warping algorithm, in: Federated Conference on Computer Science and Information Systems (FedCSIS), 2013, pp. 207–210.
URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6644000

[5] J. Wu, G. Pan, D. Zhang, G. Qi, S. Li, Gesture recognition with a 3-d accelerometer, Ubiquitous Intelligence and Computing 5585 (2009) 25–38.
URL http://www.springerlink.com/index/K1Q27M3918711318.pdf

[6] T. V. Nguyen, B. Mirza, Dual-layer kernel extreme learning machine for action recognition, Neurocomputingdoi:https://doi.org/10.1016/j.neucom.2017.04.007.
URL http://www.sciencedirect.com/science/article/pii/S092523121730677X

[7] G. Lefebvre, S. Berlemont, F. Mamalet, C. Garcia, Inertial Gesture Recognition with BLSTM-RNN, in: P. Koprinkova-Hristova, V. Mladenov, N. K. Kasabov (Eds.), Artificial Neural Networks, Vol. 4, Springer International Publishing, Cham, 2015, pp. 393–410.
URL http://link.springer.com/10.1007/978-3-319-09903-3_19

[8] S. Duffner, S. Berlemont, G. Lefebvre, C. Garcia, 3D gesture classification with Convolutional Neural Networks, in: International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2014, pp. 5432–5436.
URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6854641

[9] J. Yu, X. Yang, F. Gao, D. Tao, Deep multimodal distance metric learning using click constraints for image ranking, IEEE Transactions on Cybernetics PP (99) (2017) 1–11. doi:10.1109/TCYB.2016.2591583.

[10] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal, R. Bajcsy, Berkeley MHAD: A comprehensive multimodal human action database, in: IEEE Workshop on Applications of Computer Vision (WACV), 2013, pp. 53–60.
URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6474999

[11] J. Bromley, I. Guyon, Y. Lecun, E. Säckinger, R. Shah, Signature Verification using a "Siamese" Time Delay Neural Network, in: Proceedings of NIPS, 1994.
URL http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.28.4792&rep=rep1&type=pdf

[12] S. Chopra, R. Hadsell, Y. LeCun, Learning a similarity metric discriminatively, with application to face verification, in: Conference on Computer Vision and Pattern Recognition CVPR, Vol. 1, IEEE, 2005, pp. 539–546.
URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1467314

[13] R. Hadsell, S. Chopra, Y. LeCun, Dimensionality reduction by learning an invariant mapping, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Vol. 2, 2006, pp. 1735–1742.
URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1640964

[14] D. Yi, Z. Lei, S. Liao, S. Z. Li, Deep metric learning for person re-identification, in: International Conference on Pattern Recognition (ICPR), 2014, pp. 34–39.
URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6976727

[15] K. Chen, A. Salman, Extracting Speaker-Specific Information with a Regularized Siamese Deep Network, in: J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, K. Q. Weinberger (Eds.), Advances in Neural Information Processing Systems 24, Curran Associates, Inc., 2011, pp. 298–306.
URL http://books.nips.cc/papers/files/nips24/NIPS2011_0216.pdf

[16] Y. Sun, X. Wang, X. Tang, Deep learning face representation by joint identification-verification, CoRR abs/1406.4773.
URL http://arxiv.org/abs/1406.4773

[17] W.-t. Yih, K. Toutanova, J. C. Platt, C. Meek, Learning discriminative projections for text similarity measures, in: Proceedings of the Fifteenth Conference on Computational Natural Language Learning, 2011, pp. 247–256.
URL http://dl.acm.org/citation.cfm?id=2018965

[18] A. Bordes, J. Weston, R. Collobert, Y. Bengio, Learning structured embeddings of knowledge

bases, in: Conference on Artificial Intelligence, 2011.
URL http://infoscience.epfl.ch/record/192344/files/Bordes_AAAI_2011.pdf

[19] J. Masci, M. M. Bronstein, A. M. Bronstein, J. Schmidhuber, Multimodal Similarity-Preserving Hashing, IEEE Transactions on Pattern Analysis and Machine Intelligence 36 (4) (2014) 824–830. doi:10.1109/TPAMI.2013.225.
URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6654144

[20] G. Lefebvre, C. Garcia, Learning a bag of features based nonlinear metric for facial similarity, in: IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), 2013, pp. 238–243.
URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6636646

[21] J. Hu, J. Lu, Y.-P. Tan, Discriminative Deep Metric Learning for Face Verification in the Wild, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 1875–1882. doi:10.1109/CVPR.2014.242.
URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6909638

[22] V. Nair, G. E. Hinton, Rectified Linear Units Improve Restricted Boltzmann Machines, in: Proceedings of the 27th International Conference on Machine Learning (ICML), 2010, pp. 807–814.
URL http://www.icml2010.org/papers/432.pdf

[23] L. Zheng, K. Idrissi, C. Garcia, S. Duffner, A. Baskurt, Triangular similarity metric learning for face verification, in: IEEE International Conference on Automatic Face and Gesture Recognition (FG), 2015.
URL http://www.researchgate.net/profile/Lilei_Zheng/publication/273958279_Triangular_Similarity_Metric_Learning_for_Face_Verification/links/551168770cf24e9311ce42bb.pdf

[24] G. Lerman, J. T. Whitehouse, On d-dimensional d-semimetrics and simplex-type inequalities for high-dimensional sine functions, Journal of Approximation Theory 156 (1) (2009) 52–81. doi:10.1016/j.jat.2008.03.005.
URL http://dx.doi.org/10.1016/j.jat.2008.03.005

[25] S. Berlemont, Automatic non linear metric learning : Application to gesture recognition, Ph.D. thesis, 2016LYSEI014 (2016).
URL http://www.theses.fr/2016LYSEI014

[26] J. Cumin, G. Lefebvre, A priori Data and A posteriori Decision Fusions for Human Action Recognition, in: 11th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, 2016.
URL http://www.researchgate.net/publication/284879044_A_priori_Data_and_A_posteriori_Decision_Fusions_for_Human_Action_Recognition

[27] C. Chen, R. Jafari, N. Kehtarnavaz, Improving Human Action Recognition Using Fusion of Depth Camera and Inertial Sensors, IEEE Transactions on Human-Machine Systems 45 (1) (2015) 51–61. doi:10.1109/THMS.2014.2362520.

[28] S. Ruffieux, D. Lalanne, E. Mugellini, Chairgest: A challenge for multimodal mid-air gesture recognition for close hci, in: Proceedings of the 15th ACM on International Conference on Multimodal Interaction, ICMI '13, ACM, New York, NY, USA, 2013, pp. 483–488. doi:10.1145/2522848.2532590.
URL http://doi.acm.org/10.1145/2522848.2532590

[29] C. Cao, Y. Zhang, H. Lu, Multi-modal learning for gesture recognition, in: 2015 IEEE International Conference on Multimedia and Expo (ICME), 2015, pp. 1–6. doi:10.1109/ICME.2015.7177460.