多模式特征融合

Multimodal feature fusion for CNN-based gait recognition: an empirical comparison

F.M. Castro[a,], M.J. Marín-Jiménez[b], N. Guil[a], N. Pérez de la Blanca[c]

[a] *Department of Computer Architecture, University of Malaga, Spain, 29071*
[b] *Department of Computing and Numerical Analysis, University of Cordoba, Spain, 14071*
[c] *Department of Computer Science and Artificial Intelligence, University of Granada, Spain, 18071*

Abstract

People identification in video based on the way they walk (*i.e.* gait) is a relevant task in computer vision using a non-invasive approach. Standard and current approaches typically derive gait signatures from sequences of binary energy maps of subjects ~~extracted from images,~~ but this process introduces a large amount of non-stationary noise, thus, conditioning their efficacy. In contrast, in this paper we focus on the raw pixels, or simple functions derived from them, letting advanced learning techniques to extract relevant features. Therefore, we present a comparative study of different Convolutional Neural Network (CNN) architectures on three low-level features (*i.e.* gray pixels, optical flow channels and depth maps) on two widely-adopted and challenging datasets: TUM-GAID and CASIA-B. In addition, we perform a comparative study between different early and late fusion methods ~~used to combine the information obtained from each kind of low-level features~~. Our experimental results suggest that (*i*) the use of hand-crafted energy maps (e.g. GEI) is not necessary, since equal or better results can be achieved from the raw pixels; (*ii*) the combination of multiple modalities (*i.e.* gray pixels, optical flow and depth maps) from different CNNs allows to obtain state-of-the-art results on the gait task with an image resolution several times smaller than the previously reported results; and, (*iii*) the selection of the architecture is a critical point that can make the difference between state-of-the-art results or poor results.

Keywords: Gait signature, Convolutional Neural Networks, Multimodal Fusion, Optical Flow, Depth

1. Introduction

生物识别

The goal of *gait recognition* is to identify people by the way they walk. This type of biometric approach is considered non-invasive, since it is performed at a distance, and does not require the cooperation of the subject that has to be identified, in contrast to other methods as iris- or fingerprint-based approaches [1, 2]. Gait recognition has multiple applications in the context of video surveillance, ranging from control access in restricted areas to early detection of persons of interest as, for example, v.i.p. customers in a bank office.

From a computer vision point of view, gait recognition could be seen as a particular case of human action recognition [3, 4]. However, gait recognition requires more fine-grained features than action recognition, as differences between different gait styles are usually much more subtle than between common action categories (*e.g.* 'high jump' vs. 'javelin throw') included in state-of-the-art datasets [5].

In last years, great effort has been put into the problem of people identification based on gait recognition [6]. However, previous approaches have mostly used hand-crafted features, as energy maps, after preprocessing video frames by using non-linear filtering. The extracted features, apart from not being easily scalable to diverse datasets, are corrupted by no standard noise derived from the filtering transformation [7]. In addition, the noise introduced by the loss of local smoothing between adjacent frames along the time-line makes these features very noisy and variable. Recently, some

相邻的

Preprint submitted to arXiv *June 21, 2018*

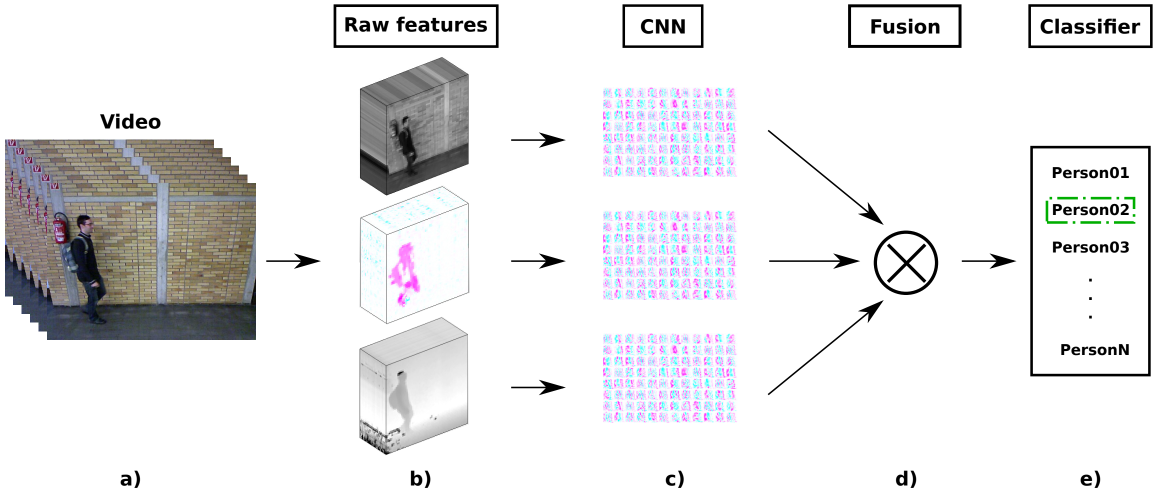

Figure 1: **Pipeline for gait recognition**. a) The input is a sequence of RGB-D video frames. b) Low-level features are extracted along the sequence and stacked building volumes. c) Volumes are passed through the CNN to obtain gait signatures. d) CNN outputs are combined. e) A final decision is taken to output an identity.

works based on Convolutional Neural Networks (CNNs) have appeared, for example, Wu *et al.* [8] presents a comparative study of CNN architectures focused on the Gait Energy Image descriptor as input.

In contrast to all the previous works, we present an approach for gait-based people identification which is independent of any strong image filtering as it uses the raw image, or simple functions derived from it, as input to find the best features (*i.e.* gait descriptor) for the identification task.

The design of our experimental study is directed towards three main objectives. The first objective is the identification of good architectures that, using as input 2D spatial information from a sequence of video frames or 3D spatio-temporal information from a finite subset of video frames, are capable of achieving high scores in the task of gait recognition. To this effect we design 2D-CNN and 3D-CNN architectures with different depth (*i.e.* layers). In addition, as previous works [9] have shown that deeper CNN models achieve better generalisation power, we have also designed a ResNet architecture based on [10]. The second objective is the use of diverse types of raw input features (*i.e.* volumes of gray-level pixels, optical flow maps and depth maps) to automatically derive gait signatures. And, the last objective is to assess if the combination of information derived from different inputs allows to obtain better models for the task of gait recognition.

To the best of our knowledge, this is the first in-depth study of the impact of CNN architectures and multimodal features on the gait recognition task using raw input data.

Therefore, the main contributions of this work are: *(i)* a comparative study of state-of-the-art CNN architectures using as input 2D or 3D information blocks representing spatial and spatio-temporal low-level information, respectively, from data; *(ii)* a thorough experimental study to validate the proposed framework on the standard TUM-GAID and CASIA-B datasets for gait identification; *(iii)* an extensive experimental study of low level feature fusion; and, *(iv)* state-of-the-art results on both datasets, being our fusion scheme the best approach.

最先进的结果

The rest of the paper is organized as follows. We start by reviewing related work in Sec. 2. An overview of the fundamentals of CNNs is presented in Sec. 3. Then, Sec. 4 explains the different CNN architectures and fusion techniques. Sec. 5 contains the experiments and results. Finally, we present the conclusions in Sec. 6.

2. Related work

2.1. Feature learning

A new realm of the feature learning field for recognition tasks started with the advent of Deep Learning (DL) architectures [11]. These architectures are suitable for discovering good features for classification tasks [12, 13]. Recently, DL approaches based on CNN have been used on image-based tasks with great success [9, 14, 15]. In the last years, deep architectures for video have appeared, specially focused on action recognition, where the inputs of the CNN are subsequences of stacked frames. The very first approximation of DL applied to stacked frames is proposed in [16], where the authors apply a convolutional version of the Independent Subspace Analysis algorithm to sequences of frames. By this way, they obtain low-level features which are used by high-level representation algorithms. A more recent approach is proposed in [17], where a complete CNN is trained with sequences of stacked frames as input. In [18], Simonyan and Zisserman proposed to use as input to a CNN a volume obtained as the concatenation of two channels: optical flow in the x-axis and y-axis. To normalize the size of the inputs, they split the original sequence in subsequences of 10 frames, considering each subsample independently.

Donahue *et al.* [19] propose a new viewpoint in DL using a novel architecture called 'Long-term Recurrent Convolutional Networks'. This new architecture combines CNN (specialized in spatial learning) with Recurrent Neural Networks (specialized in temporal learning) to obtain a new model able to deal with visual and temporal features at the same time. Recently, Wang *et al.* [20] combined dense trajectories with DL. The idea is to obtain a powerful model that combines the deep-learnt features with the temporal information of the trajectories. They train a traditional CNN and use dense trajectories to extract the deep features to build a final descriptor that combines the deep information over time. On the other hand, Perronnin *et al.* [21] proposed a more traditional approach using Fisher Vectors as input to a Deep Neural Network instead of using other classifiers like SVM. Recently, He *et al.* [10] proposed a new kind of CNN, named ResNet, which has a large number of convolutional layers and 'residual connections' to avoid the vanishing gradient problem.

Although several papers can be found for the task of human action recognition using DL techniques, few works apply DL to the problem of gait recognition. In [22], Hossain and Chetty propose the use of Restricted Boltzmann Machines to extract gait features from binary silhouettes, but a very small probe set (*i.e.* only ten different subjects) were used for validating their approach. A more recent work, [23], uses a random set of binary silhouettes of a sequence to train a CNN that accumulates the calculated features in order to achieve a global representation of the dataset. In [24], raw 2D GEI are employed to train an ensemble of CNN, where a Multilayer Perceptron (MLP) is used as classifier. Similarly, in [25] a multilayer CNN is trained with GEI data. A novel approach based on GEI is developed on [8], where the CNN is trained with pairs of gallery-probe samples and using a distance metric. Castro *et al.* [26] use optical flow obtained from raw data frames. An in-dept evaluation of different CNN architectures based on optical flow maps is presented in [27]. Finally, in [28] a multitask CNN with a combined loss function with multiple kind of labels is presented.

Despite most CNNs are trained with visual data (e.g. images or videos), there are some works that build CNNs for different kinds of data like inertial sensors or human skeletons. Holden *et al.* [29] propose a CNN that corrects wrong human skeletons obtained by other methods or devices (e.g. Microsoft Kinect). Neverova *et al.* [30] build a temporal network for active biometric authentication with data provided by smartphone sensors (e.g. accelerometers, gyroscope, etc.).

Recently, some authors have proposed the use of 3D convolutions to extract visual and temporal data from videos. Tran *et al.* [31] define a new network composed of 3D convolutions in the first layers that has been successfully applied to action recognition. Following that idea, Wolf *et al.* [32] build a CNN with 3D convolutions for gait recognition. Due to the high number of parameters that must be trained (3D convolutions implies three times more parameters per convolutional layer), Mansimov *et al.* [33] show several ways to initialize a 3D CNN from a 2D CNN.

2.2. Information fusion

Since there are different descriptors for representing the same data, an interesting idea would be to try to combine those descriptors into a single one that could benefit from the original descriptors. To perform this task, several methods have appeared [34, 35]. Also, the emergence of new cheaper devices that record multimodal spectrums (e.g. RGB, depth, infrared) has allowed to investigate how to fuse that information to build richer and more robust representations for the gait recognition problem. Traditionally, fusion methods are divided into *early fusion* methods (or feature fusion) and *late fusion* (or decision fusion). The first ones try to build descriptors by fusing features of different descriptors, frequently, using the concatenation of the descriptors into a bigger one as in [36]. On the other hand, late fusion tries to fuse the decisions obtained by each classifier of each modality, usually, by applying arithmetic operations like sums or products on the scores obtained by each classifier as in [36, 37]. Castro *et al.* [38] perform an extensive comparative between late fusion and early fusion methods including the traditional fusion schemes and others more grounded that can perform robust fusions. Fusion has been also employed with CNN to improve the recognition accuracy for different computer vision tasks. For example, two independent CNNs fed with optical flow maps and appearance information (*i.e.* RGB pixel volumes) are employed in [18] to perform action recognition. Then, class score fusion is used to combine the softmax output of both CNNs. In a similar way, Eitel *et al.* [39] have proposed a DL approach for object recognition by fusing RGB and depth input data. They concatenate the outputs of the last fully-connected layers of both networks (those processing RGB and depth data) and process them through an additional fusion layer. Wang *et al.* [40] also employ a multimodal architecture composed by two CNN networks to process RGB-D data. They propose to learn two independent transformations of the activations of the second fully-connected layer of each network, so correlation of color and depth features is maximized. In addition, these transformations are able to improve the separation between samples belonging to different classes.

In this work, we explore several fusion techniques for the problem of gait-based people identification, combining automatically-learnt gait signatures extracted from gray pixels, optical flow and depth maps.

3. CNN overview

The convolutional neural network (CNN) model is an important type of feed-forward neural network with special success on applications where the target information can be represented by a hierarchy of local features (see [11]). A CNN is defined as the composition of several convolutional layers and several fully connected layers. Each convolutional layer is, in general, the composition of a non-linear layer and a pooling or sub-sampling layer to get some spatial invariance. For images, the non-lineal layer of the CNN takes advantage, through local connections and weight sharing, of the 2D structure present in the data. These two conditions impose a very strong regularization on the total number of weights in the model, which allows a successful training of the model by using back-propagation.

空间不变性

In the last years, CNN models are achieving state-of-the-art results on many different complex applications (*e.g.* object detection, text classification, natural language processing, scene labeling, etc.) [41, 9, 42, 43]. However, to the extent of our knowledge, only very few works [26, 27, 28, 32] have applied CNN models to the problem of gait recognition using as input low-level features different to binary silhouettes (in contrast to [23]). The great success of the CNN model is in part due to its use on data where the target can be represented through a feature hierarchy of increasing semantic complexity. When a CNN is successfully trained, the output of the last hidden layer can be seen as the coordinates of the target in a high-level representation space. The fully connected layers, on top of the convolutional ones, allow us to reduce the dimensionality of such representation and, therefore, to improve the classification accuracy. In this work, we compare three kinds of CNN architectures (*i.e.* 2D-CNN, 3D-CNN and ResNet), three kinds of input features (*i.e.* gray, optical flow and depth), and diverse feature fusion techniques, applied to the problem of gait recognition.

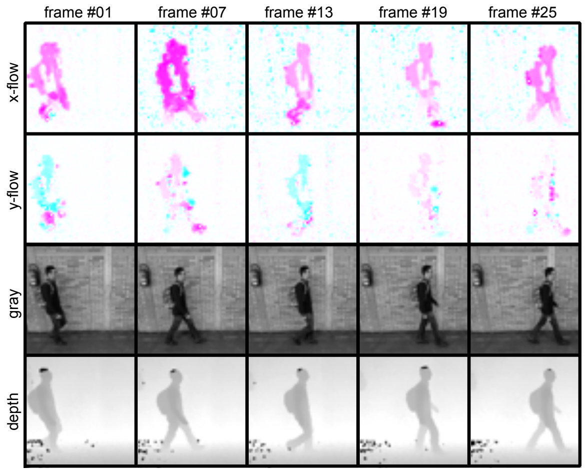

Figure 2: **CNN input data**. Sample frames extracted from a subsequence of 25 frames. **(top rows)** Optical flow in x-axis and y-axis. where positive flows are displayed in pink and negative flows in blue (best viewed in color). **(bottom rows)** Gray pixels and depth maps of the same sequence.

4. Proposed approach

In this section we describe our proposed framework to address the problem of gait recognition using CNNs. The pipeline proposed for gait recognition based on CNNs is represented in Fig. 1: *(i)* gather low-level features along the whole sequence; *(ii)* build up a data cuboid from consecutive low-level feature maps; *(iii)* feed the CNN with the low-level feature cuboid to extract the gait signature; *(iv)* fuse information from the different inputs; and, *(v)* apply a classifier to decide the subject identity.

4.1. Input data

We describe here the different types of low-level features used as input for the proposed CNN architecture. In particular, we use optical flow, gray pixels and depth maps. An example of the three types of low-level features is represented in Fig. 2. Our intuition is that this set of low-level features will cover both *motion* (*i.e.* optical flow) and *appearance* information (*i.e.* pixels and depth) of people.

4.1.1. Optical flow

The use of optical flow (OF) as input data for action representation in video with CNN has already shown excellent results [18]. Nevertheless human action is represented by a wide, and usually well defined, set of local motions. In our case, the set of motions differentiating one gait style from another is much more subtle and local.

Let F_t be an OF map computed at time t and, therefore, $F_t(x, y, c)$ be the value of the OF vector component c located at coordinates (x, y), where c can be either the horizontal or vertical component of the corresponding OF vector. The input data I_L for the CNN are cuboids built by stacking L consecutive OF maps F_t, where $I_L(x, y, 2k-1)$ and $I_L(x, y, 2k)$ corresponds to the value of the horizontal and vertical OF components located at spatial position (x, y) and time k, respectively, ranging k in the interval $[1, L]$.

Since each original video sequence will probably have a different temporal length, and CNN requires a fixed size input, we extract subsequences of L frames from the full-length sequences. In Fig. 2 we show five frames distributed every six frames along a subsequence of twenty-five frames in total (*i.e.* frames 1, 7, 13, 19, 25). The first row shows the horizontal component of the OF (x-axis displacement) and second row shows the vertical component of the OF (y-axis displacement). It can be observed that most of the motion flow is concentrated in the horizontal component, due to the displacement of the person. In order to remove noisy OF located in the background, as it can be observed in Fig. 2,

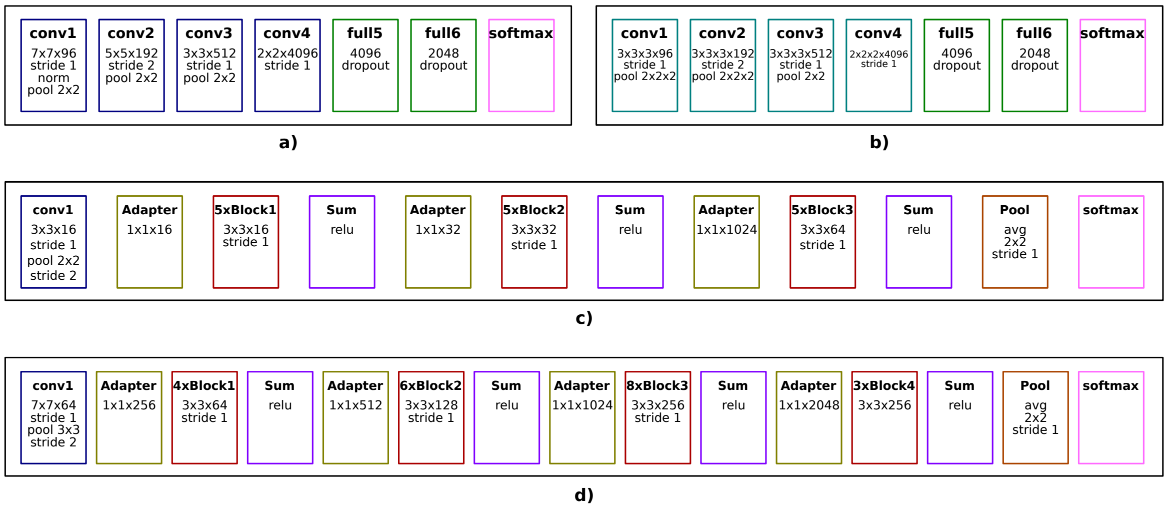

Figure 3: **Proposed CNN architectures for gait signature extraction**. **a) 2D-CNN:** linear CNN with four 2D convolutions, two fully connected layers and a softmax classifier. **b) 3D-CNN:** 3D CNN four 3D convolutions, two fully connected layers and a softmax classifier. **c) ResNet-A:** residual CNN with a 2D convolution, three residual blocks (red boxes), an average pooling layer and a final softmax classifier. **d) ResNet-B:** residual CNN with a 2D convolution, four residual blocks (red boxes), an average pooling layer and a final softmax classifier. More details in the main text.

we might think in applying a preprocessing step for filtering out those vectors whose magnitude is out of a given interval. However, since our goal in this work is to minimize the manual intervention in the process of gait signature extraction, we will use those OF maps as returned by the OF algorithm.

人工干预

4.1.2. Gray-level pixels

When using CNNs for object detection and categorization, the most popular low level features are raw pixels [9]. In contrast to [18], that uses single RGB frames for action recognition, we build cuboids of gray pixels with the aim of better capturing the important features of the subject appearance. Note that in gait recognition, color is not as informative as it is for object recognition. Therefore, using only gray intensity will eventually help CNN to focus just on the gait-relevant information. An example can be seen in the corresponding row of Fig. 2.

4.1.3. Depth maps

As far as we know, the use of depth information has not been explored much in the field of gait recognition. In [37] they basically use depth to segment people from background and compute the *Gait Energy Volume* descriptor [44]. Castro *et al.* [45] represent depth information in a gray-scale image where the intensity of a pixel is the depth value scaled to $[0, 255]$. In our opinion, depth information is rich and should be studied in depth for this problem. Therefore, given a sequence of depth maps, we extract depth volumes that will be used as input data for the corresponding CNN architecture. An example of depth maps can be seen in the bottom row of Fig. 2.

4.2. CNN architectures for gait signature extraction

We have selected the three architectures that most frequently appear in the bibliography and produce state-of-the-art results in different topics (e.g. action recognition, object detection, etc.). The three proposed architectures are: *(i)* a linear CNN with 2D convolutions (*2D-CNN*), which is the traditional and most common architecture; *(ii)* a linear CNN with 3D convolutions and pooling (*3D-CNN*), which is specially designed to capture information in videos; and, *(iii)* a 2D very deep residual CNN (*ResNet*), which produces state-of-the-art results in most challenging tasks.

The input to our CNN is a volume of gray pixels, OF channels or depth maps of size $N \times N \times L$. See Sec. 5.2 for the actual values of N and L used in the experiments.

We describe below the four models compared in the experimental section (Sec. 5). Note that, along this paper, we use the term 'softmax layer' to refer to a fully-connected layer with as many units as classes followed by a softmax exponential layer.

2D-CNN: This CNN is composed of the following sequence of layers (Fig. 3a): '*conv1*', 96 filters of size 7×7 applied with stride 1 followed by a normalization and max pooling 2×2; '*conv2*', 192 filters of size 5×5 applied with stride 2 followed by max pooling 2×2; '*conv3*', 512 filters of size 3×3 applied with stride 1 followed by max pooling 2×2; '*conv4*', 4096 filters of size 2×2 applied with stride 1; '*full5*', fully-connected layer with 4096 units and dropout; '*full6*', fully-connected layer with 2048 units and dropout; and, '*softmax*', softmax layer with as many units as subject identities. All convolutional layers use the rectification (ReLU) activation function.

3D-CNN: As optical flow has two components and the CNN uses temporal kernels, the network is split into two branches: x-flow and y-flow. Therefore, each branch contains half of the total filters described below. Then, this CNN is composed by the following sequence of layers (Fig. 3b): '*conv1*', 96 filters of size $3 \times 3 \times 3$ applied with stride 1 followed by a max pooling $2 \times 2 \times 2$; '*conv2*', 192 filters of size $3 \times 3 \times 3$ applied with stride 2 followed by max pooling $2 \times 2 \times 2$; '*conv3*', 512 filters of size $3 \times 3 \times 3$ applied with stride 1 followed by max pooling $2 \times 2 \times 2$; '*conv4*', 4096 filters of size $2 \times 2 \times 2$ applied with stride 1; '*concat*', concatenation of both branches (x-flow and y-flow); '*full5*', fully-connected layer with 4096 units and dropout; '*full6*', fully-connected layer with 2048 units and dropout; and, '*softmax*', softmax layer with as many units as subject identities. All convolutional layers use the rectification (ReLU) activation function.

ResNet-A: This CNN is composed by the following sequence of layers and residual blocks (a sequences of two convolutions of size 3×3, as defined in [10] for CIFAR Dataset). This model is specially designed for small datasets with low variability because this kind of networks tends to overfit due to its high number of layers. As our architecture follows the indications defined by the authors [10], we only describe the main blocks (Fig. 3c): '*conv1*', 16 filters of size 3×3 applied with stride 1 followed by a max pooling 2×2 and stride 2; '*block 1*', 5 residual blocks with convolutions of 16 filters of size 3×3 applied with stride 1; '*block 2*', 5 residual blocks with convolutions of 32 filters of size 3×3 applied with stride 1; '*block 3*', 5 residual blocks with convolutions of 64 filters of size 3×3 applied with stride 1; '*average pooling*', size 8×8 with stride 1; and, '*softmax*', softmax layer with as many units as subject identities. All convolutional layers use the rectification (ReLU) activation function and batch normalization.

ResNet-B: This model is an extension of the model ResNet-A. The number and size of layers of this model is increased and it is specially designed for datasets with high variability (e.g. CASIA-B). This CNN is composed by the following sequence of layers and residual blocks (a sequence of three convolutions of size 1×1, 3×3 and 1×1, as defined in [10]). As our architecture follows the indications defined by the authors, we only describe the main blocks (Fig. 3d): '*conv1*', 64 filters of size 7×7 applied with stride 1 followed by a max pooling 3×3 and stride 2; '*block 1*', 4 residual blocks with convolutions of 64 filters of size 3×3 applied with stride 1; '*block 2*', 6 residual blocks with convolutions of 128 filters of size 3×3 applied with stride 1; '*block 3*', 8 residual blocks with convolutions of 256 filters of size 3×3 applied with stride 1; '*block 4*', 3 residual blocks with convolutions of 256 filters of size 3×3 applied with stride 1; '*average pooling*', size 2×2 with stride 1; and, '*softmax*', softmax layer with as many units as subject identities. All convolutional layers use the rectification (ReLU) activation function and batch normalization.

4.2.1. Model training

For 2D and 3D models, we perform an incremental training to speed up and to facilitate the convergence. In this incremental process, initially, we train a simplified version of each model (*i.e.* less units per layer and no dropout) and, then, we use its weights for initializing the layers of a more complex version of that previous model (*i.e.* 0.1 dropout and more filters and units). By this way, we train three incremental versions using the previous weights until we obtain the final model architecture represented in Fig. 3.

During CNN training, the weights are learnt using mini-batch stochastic gradient descent algorithm with momentum equal to 0.9 in the first two incremental iterations of the 2D and 3D models, and 0.95 during the last one. Note that ResNet-A and ResNet-B are trained from scratch in just one iteration (without incremental training) so momentum for these nets is set to 0.9. We set weight decay to $5 \cdot 10^{-4}$ and dropout to 0.4 (when corresponds). The number of epochs is limited to 20 in TUM-GAID and the learning rate is initially set to 10^{-2} and it is divided by ten when the validation error gets stuck. Due to the specifics of the ResNet models, the initial learning rate is set to 0.1.

In CASIA-B we limit the training stage to 30 epochs, the learning rate is initially set to 10^{-3} and it is divided by two when the validation error gets stuck. At each epoch, a mini-batch of 150 samples is randomly selected from a balanced training set (*i.e.* almost the same proportion of samples per class). Note that for ResNet models we use a mini-batch of 64 samples. When the CNN training has converged, we perform five more epochs on the joint set of training and validation samples.

To run our experiments we use the implementation of CNN provided in MatConvNet library [46]. This library allows to develop CNN architectures in an easy and fast manner using the Matlab environment. In addition, it takes advantage of CUDA and cuDNN [47] to improve the performance of the algorithms. Using this open source library will allow other researchers to use our trained models and reproduce our experimental results.

4.3. Single modality

Once we have obtained the gait signatures, the final stage consists in classifying those signatures to derive a subject identity. Although the softmax layer of the CNN is already a classifier (*i.e.* each unit represents the probability of belonging to a class), the fully-connected layers can play the role of gait signatures that can be used as input of a Support Vector Machine (SVM) classifier. Since we are dealing with a multiclass problem, we define an ensemble of C binary SVM classifiers with linear kernel in an 'one-vs-all' fashion, where C is the number of possible subject identities. Previous works (*e.g.* [48]) indicate that this configuration of binary classifiers is suitable to obtain top-tier results in this problem. Note that we $L2$-normalize the top fully-connected layer before using it as feature vector, as early experiments shown improved results.

In Sec. 4.1, we split the whole video sequence into overlapping subsequences of a fixed length, and those subsequences are classified independently. Therefore, in order to derive a final identity for the subject walking along the whole sequence, we apply a *majority voting* strategy on the labels assigned to each subsequence.

An alternative way for obtaining a final label for a video v from the set of subsequences $\{s_i\}$ is to derive the identity from the product of softmax vectors (*i.e.* probability distributions P_i) obtained:

$$P(v = c) = \prod_{i=1}^{t} P_i(s_i = c), \quad \text{c在v中的概率=c在子序列si中的概率累乘} \tag{1}$$

where t is the number of subsequences extracted from video v, $P(v = c)$ is the probability of assigning the identity c to the person in video v and $P_i(s_i = c)$ is the probability of assigning the identity c to the person in subsequence s_i.

4.4. Multiple modalities

In the case where several low-level features have been used, we explore different approaches for combining the outputs of the CNN.

Late fusion. Focusing on the softmax scores returned by each CNN, we explore the following approaches to combine them: product and weighted sum. These approaches are considered as 'late fusion' ones, as fusion is performed on the classification scores.

A) Product of softmax vectors. Given a set of n softmax vectors $\{P_i\}$ obtained from a set of different modalities $\{m_i\}$, a new score vector S_{prod} is obtained as:

$$S_{\text{prod}}(v = c) = \prod_{i=1}^{n} P_i(m_i = c) \quad \text{c在v中的概率= 用mi方法 c 在 v 中的累乘} \tag{2}$$

where n is the number of modalities used to classify video v, $S_{\text{prod}}(v = c)$ can be viewed as the probability of assigning the identity c to the person in video v and $P_i(m_i = c)$ is the probability of assigning the identity c to the person in modality m_i.
B) Weighted sum of softmax vectors. Given a set of n softmax vectors obtained from a set of different modalities $\{m_i\}$ a new score vector S_{ws} is obtained as:

$$S_{\text{ws}}(v = c) = \sum_{i=1}^{n} \beta_i P_i(m_i = c), \quad \text{每种方法的结果具有不同的权值} \tag{3}$$

where n is the number of modalities used to classify video v, $S_{\text{ws}}(v = c)$ can be viewed as the probability of assigning the identity c to the person in video v, $P_i(m_i = c)$ is the probability of assigning the identity c to the person in modality m_i and β_i is the weight associated to modality m_i, subject to $\beta_i > 0$ and $\sum_{i=1}^{n} \beta_i = 1$.

β values are selected empirically by cross-validation. Note that the values used for each experiment are specified in its corresponding section.

Early fusion. The fusion performed at descriptor level is known as ‘early fusion’.

In our case, as we are working with CNNs, early fusion could be performed at any layer before the ‘softmax’ one. Depending on the layer, the combined descriptors are matrices (fusion before a convolutional layer) or vectors (fusion before a fully-connected layer). We have tried all the possible fusion locations for our CNNs and we have selected the best solution according to the results obtained. In our case, the best early fusion location is after layer ‘full6’ of each modality. The activations of those layers are concatenated and fed into a new set of layers to perform the actual fusion. Therefore, we extend the 2D and 3D networks shown in Fig. 3 with the set of additional layers summarized in Fig. 4:

‘*concat*:’ concatenation layer;
‘*full7*:’ fully-connected layer with 4096 units, ReLU and dropout;
‘*full8*:’ fully-connected layer with 2048 units, ReLU and dropout;
‘*full9*:’ fully-connected layer with 1024 units, ReLU and dropout; and,
‘*softmax*:’ softmax layer with as many units as subject identities.

During the training process, the weights of the whole CNN (the branch of each modality and the fusion layers) are trained altogether, automatically learning the best combination of weights for the modalities. From our point of view, this kind of fusion is considered early as it is not performed at classification-score level, as done above.

For ResNet models, due to their high number of layers, we do not stack more fully-connected layers to prevent overfitting. Therefore, the selected early fusion architecture is the same as for the rest of models but without fully-connected layers:

‘*concat*:’ concatenation layer;
‘*softmax*:’ softmax layer with as many units as subject identities and dropout.

5. Experiments and results

We present here the experiments designed to validate our approach and the results obtained on the selected datasets for gait recognition.

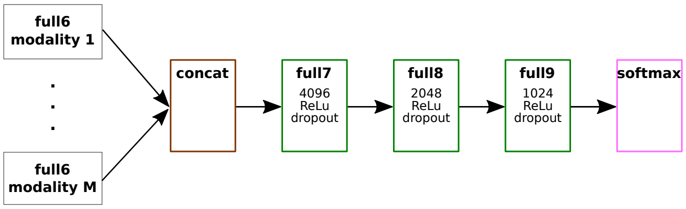

Figure 4: **Proposed set of layers for early fusion**. A concatenation layer and three fully-connected layers are followed by a softmax classifier used to directly derive an identity. More details in the main text.

5.1. Datasets

We run our experiments on two widely used and challenging datasets for gait recognition: TUM-GAID [37] and CASIA-B [49].

TUM-GAID. In 'TUM Gait from Audio, Image and Depth' (TUM-GAID) 305 subjects perform two walking trajectories in an indoor environment. The first trajectory is performed from left to right and the second one from right to left. Therefore, both sides of the subjects are recorded. Two recording sessions were performed, one in January, where subjects wore heavy jackets and mostly winter boots, and the second in April, where subjects wore different clothes. The action is captured by a Microsoft Kinect sensor which provides a video stream with a resolution of 640 × 480 pixels with a frame rate of approximately 30 fps. Some examples can be seen in the left part of Fig. 5 depicting the different conditions included in the dataset. 描述了一下TUM-GAID的数据图片状态

Hereinafter the following nomenclature is used to refer each of the four walking conditions considered: *normal* walk (*N*), carrying a *backpack* of approximately 5 kg (*B*), wearing coating *shoes* (*S*), as used in clean rooms for hygiene conditions, and *elapsed time* (*TN-TB-TS*). Each subject of the dataset is composed of: six sequences of normal walking (*N1, N2, N3, N4, N5, N6*), two sequences carrying a bag (*B1, B2*) and two sequences wearing coating shoes (*S1, S2*). In addition, 32 subjects were recorded in both sessions (*i.e.* January and April) so they have 10 additional sequences (*TN1, TN2, TN3, TN4, TN5, TN6, TB1, TB2, TS1, TS2*). Therefore, the overall amount of videos is 3400.

We follow the experimental protocol defined by the authors of the dataset [37]. Three subsets of subjects are proposed: training, validation and testing. The training set is used for obtaining a robust model against the different covariates of the dataset. This partition is composed of 100 subjects and the sequences *N1* to *N6*, *B1*, *B2*, *S1* and *S2*. The validation set is used for validation purposes and contains 50 different subjects with the sequences *N1* to *N6*, *B1*, *B2*, *S1* and *S2*. Finally, the test set contains other 155 different subjects used in the test phase. As the set of subjects is different between the test set and the training set, a new training of the identification model must be performed. For this purpose, the authors reserve the sequences *N1* to *N4*, from the subject test set, to train the model again and the rest of sequences are used for testing and to obtain the accuracy of the model. In the *elapsed time* experiment, the temporal sequences (*TN1, TN2, TN3, TN4, TN5, TN6, TB1, TB2, TS1, TS2*) are used instead of the normal ones and the subsets are: 10 subjects in the training set, 6 subjects in the validation set and 16 subjects in the test set.

In our experiments, after parameter selection, the validation sequences are added to the training set for fine-tuning the final model.

CASIA-B. In CASIA-B 124 subjects perform walking trajectories in an indoor environment (right part of Fig. 5). The action is captured from 11 viewpoints (*i.e.* from 0^o to 180^o in steps of 18^o) with a video resolution of 320 × 240 pixels. Three situations are considered: normal walk (*nm*), wearing a coat (*cl*), and carrying a bag (*bg*). The authors of the dataset indicate that sequences 1 to 4 of the '*nm*' scenario should be used for training the models. Whereas the remaining sequences should be used for testing: sequences 5 and 6 of '*nm*', 1 and 2 of '*cl*' and 1 and 2 of '*bg*'. Therefore, we follow this protocol in our experiments, unless otherwise stated. This makes a total of 496 video sequences for training, per camera viewpoint.

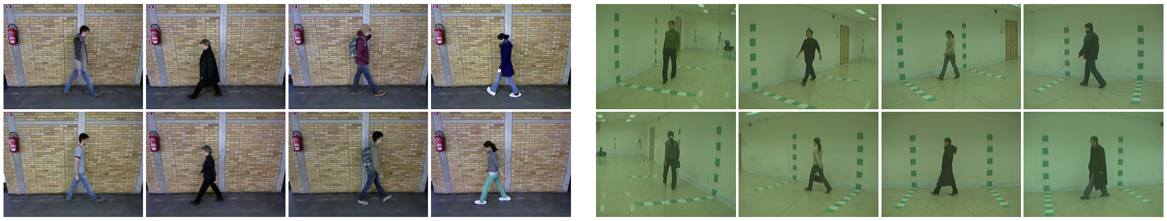

Figure 5: **Datasets for gait recognition**. **(left) TUM-GAID.** People walking indoors under four walking conditions: normal walking, wearing coats, carrying a bag and wearing coating shoes. Top and bottom rows show the same set of subjects but in different months of the same year. **(right) CASIA-B.** People walking indoors recorded from eleven camera viewpoints and under three walking conditions: normal walking, wearing coats and carrying a bag.

5.2. Implementation details

We ran our experiments on a computer with 32 cores at 2.3 GHz, 256 GB of RAM and a GPU NVidia Titan X Pascal, with MatConvNet library [46] running on Matlab 2016a for Ubuntu 16.04.

For the following experiments with CNN, we resized all the videos to a common resolution of 80×60 pixels, keeping the original aspect ratio of the video frames. Preliminary experiments support this choice [26], as this size exhibits a good trade-off between computational cost and recognition performance. Note that resolution 80×60 is 4 times lower than original CASIA-B and 8 times lower than TUM-GAID one. Given the resized video sequences, we compute dense *OF* on pairs of frames by using the method of Farneback [50] implemented in OpenCV library [51]. In parallel, people are located in a rough manner along the video sequences by background subtraction [52]. Then, we crop the video frames to remove part of the background, obtaining video frames of 60×60 pixels (full height is kept) and to align the subsequences (people are x-located in the middle of the central frame, #13) as in Fig. 2.

Finally, from the cropped *OF* maps, we build subsequences of 25 frames by stacking *OF* maps with an overlap of $\mathcal{O}\%$ frames. In our case, we chose $\mathcal{O} = 80\%$, that is, to build a new subsequence, we use 20 frames of the previous subsequence and 5 new frames. For most state-of-the-start datasets, 25 frames cover almost one complete gait cycle, as stated by other authors [53]. Therefore, each *OF* volume has size $60 \times 60 \times 50$.

The same process described above is applied to the gray pixels and depth inputs, obtaining volumes of size $60 \times 60 \times 25$. Before feeding the CNN with those data volumes, the mean of the training set for each modality is subtracted to the input data. Both gray and depth values are normalized to the range $[0, 255]$. Note that in CASIA-B, due to the high variability between viewpoints, it is necessary to normalize gray values to the range $[0, 1]$.

To increase the amount of training samples we add mirror sequences and apply spatial displacements of ±5 pixels in each axis, obtaining a total of 8 new samples from each original sample.

5.3. Performance evaluation

For each test sample, we return a sorted list of possible identities, where the top one identity corresponds to the largest scored one. Therefore, we use the following metrics to quantitative measure the performance of the proposed system: *rank-1* and *rank-5*. Metric *rank-1* measures the percentage of test samples where the top one assigned identity corresponds to the right one. Metric *rank-5* measures the percentage of test samples where the ground truth identity is included in the first five ranked identities for the corresponding test sample. Note that *rank-5* is less strict than *rank-1* and, in a real system, it would allow to verify if the target subject is any of the top five most probably ones. Final results at sequence level are obtained by applying a majority vote strategy except in the product of softmax scores which is the only case where we have probabilities between 0 and 1 and therefore, we can multiply them for obtaining a sequence probability.

Along this section, we are going to use the following notation: 'SM-Vote': softmax decision followed by majority voting to obtain the sequence level results; 'SM-Prod': softmax decision followed by the

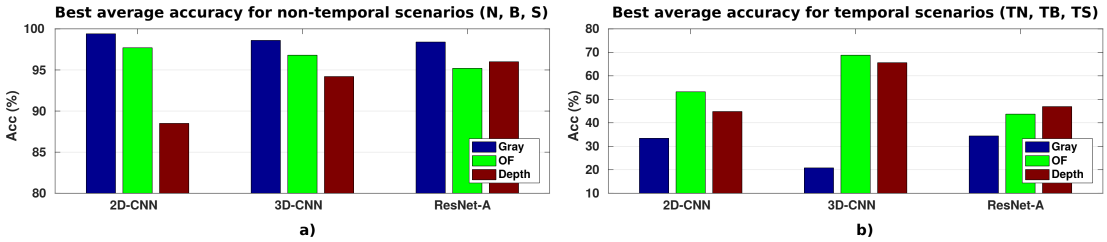

Figure 6: **Best average accuracy**. **a)** non-temporal scenarios (N, B, S) **b)** temporal scenarios (TN, TB, TS)

product of the scores to obtain the sequence level results; 'SVM-L2': SVM with L2 normalization of the features; 'SVM-SM': SVM trained with the scores of the softmax.

5.4. Experimental results on TUM-GAID

In this section, we first examine the impact of CNN architectures in automatic extraction of gait signatures from diverse low-level features, studying which one is the more convenient for the different scenarios. Afterwards, we evaluate the impact of combining gait signatures from different low-level features for people identification. Finally, we compare our results to the state-of-the-art ones.

5.4.1. Architecture and feature evaluation

In this experiment, we evaluate the individual contribution of each low-level feature (*i.e.* gray pixels, optical flow and depth maps) and each architecture (*i.e.* 2D, 3D and ResNet) for extracting discriminative gait signatures. Note that, as this dataset only contains a single viewpoint, ResNet models tend to overfit due to the lack of variability in the training data. Therefore, we use ResNet-A (see Sec. 4.2 for more details) which is shallower than traditional ResNet models. Tabs. 1, 2 and 3 summarize the identification results obtained on TUM-GAID with each modality: *Gray*, *OF* and *Depth*. Each column contains the results for rank-1 (R1) and rank-5 (R5) for each scenario. The last column '*AVG*' is the average of each case (temporal and non temporal) weighted by the number of classes.

Table 1: **Feature selection on TUM-GAID *Gray*-modality.** Percentage of correct recognition by using *rank-1* (R1) and *rank-5* (R5) metrics. Each row corresponds to a different classifier and modality. Each column corresponds to a different scenario. Best average results are marked in bold.

		N		*B*		*S*		*TN*		*TB*		*TS*		*AVG*	
		R1	R5	R1	R5	R1	R5	R1	R5	R1	R5	R1	R5	R1	R5
2D-CNN	*SM-Vote*	99.4	100	99.0	99.7	98.4	99.7	31.3	53.1	34.4	65.6	34.4	62.5	92.8	96.1
	SM-Prod	100	100	99.7	99.7	98.4	99.7	28.1	62.5	37.5	71.9	34.4	59.4	**93.2**	96.5
	SVM+L2	100	100	99.7	99.7	98.4	99.7	34.4	68.8	31.3	78.1	34.4	68.8	**93.2**	**97.2**
	SVM-SM	99.4	99.7	99.4	99.4	97.4	98.7	28.1	65.6	34.4	71.9	34.4	68.8	92.5	96.4
3D-CNN	*SM-Vote*	99.7	100	98.4	99.7	96.8	99	21.9	50	21.9	46.9	12.5	43.8	90.9	94.6
	SM-Prod	97.7	97.7	93.9	94.2	91.3	91.6	18.8	37.5	21.9	37.5	12.5	31.3	87.1	89
	SVM+L2	100	100	98.1	99.4	97.7	99	18.8	56.3	28.1	62.5	15.6	62.5	91.3	95.8
	SVM-SM	99.7	99.7	98.4	99	96.8	97.7	21.9	43.8	21.9	53.1	12.5	43.8	90.9	93.9
RESNET-A	*SM-Vote*	99.4	100	95.8	99.4	96.1	99	25	62.5	34.4	98.8	25	59.4	90.6	96.1
	SM-Prod	99	100	96.5	99.4	95.5	99	28.1	56.3	34.4	68.8	25	56.3	90.7	95.8
	SVM+L2	100	100	97.4	99	97.7	100	34.4	53.1	34.4	53.1	34.4	50	92.4	95.2
	SVM-SM	100	100	95.8	97.7	97.1	98.7	25	50	25	62.5	25	56.3	90.8	94.8

In Fig. 6 appears the best average performance for non-temporal and temporal scenarios per modality. If we focus on the non-temporal scenarios (*N*, *B* and *S*), we can see that features based on *Gray* or *Depth* are able to outperform the results obtained with *OF*. On the other hand, if we focus

Table 2: **Feature selection on TUM-GAID *OF*-modality.** Percentage of correct recognition by using *rank-1* (R1) and *rank-5* (R5) metrics. Each row corresponds to a different classifier and modality. Each column corresponds to a different scenario. Best average results are marked in bold.

		N		*B*		*S*		*TN*		*TB*		*TS*		*AVG*	
		R1	R5	R1	R5	R1	R5	R1	R5	R1	R5	R1	R5	R1	R5
2D-CNN	*SM-Vote*	99.4	100	97.4	100	96.4	99.4	53.1	96.9	43.8	87.5	56.3	93.8	93.4	**99.1**
	SM-Prod	99.4	100	97.7	100	96.1	99.4	56.3	87.5	43.8	84.4	59.4	90.6	93.6	98.7
	SVM+L2	99.4	100	96.5	99.4	96.8	99.4	50.0	90.6	56.3	84.4	43.8	90.6	93.1	98.6
	SVM-SM	99.0	100	96.8	98.4	95.5	98.7	53.1	78.1	50.0	81.3	56.3	91.3	93.0	97.6
3D-CNN	*SM-Vote*	99	99.4	95.5	99.7	94.2	98.1	65.6	90.6	65.6	93.8	59.4	87.5	93.2	98.3
	SM-Prod	98.7	99.7	97.1	99.4	94.5	98.7	71.9	87.5	68.8	87.5	65.6	84.4	**94.1**	98.1
	SVM+L2	98.7	99.4	93.9	99	92.6	98.4	65.6	87.5	65.6	81.3	56.3	90.6	92	97.8
	SVM-SM	98.7	99	95.5	99.4	94.2	97.1	65.6	90.6	65.6	81.3	59.4	84.4	93.1	97.3
RESNET-A	*SM-Vote*	94.5	99.7	81	98.4	85.1	97.7	34.4	93.8	34.4	90.6	37.5	93.8	82.1	98.1
	SM-Prod	95.2	99.4	81	98.7	86.1	97.7	34.4	96.7	40.6	93.8	43.8	93.8	83	98.2
	SVM+L2	99.4	99.4	93.9	98.1	92.2	98.1	37.5	87.5	40.6	81.3	53.1	90.6	90.4	97.4
	SVM-SM	97.4	98.7	89.7	96.5	89.6	92.6	37.5	75	43.8	84.4	46.9	75	87.6	95.4

Table 3: **Feature selection on TUM-GAID *Depth*-modality.** Percentage of correct recognition by using *rank-1* (R1) and *rank-5* (R5) metrics. Each row corresponds to a different classifier and modality. Each column corresponds to a different scenario. Best average results are marked in bold.

		N		*B*		*S*		*TN*		*TB*		*TS*		*AVG*	
		R1	R5	R1	R5	R1	R5	R1	R5	R1	R5	R1	R5	R1	R5
2D-CNN	*SM-Vote*	98.4	100	65.8	90.0	96.8	99.7	34.4	93.8	34.4	93.8	50.0	84.4	82.6	96.0
	SM-Prod	98.7	100	66.1	90.7	96.8	99.7	43.8	90.6	40.6	87.5	46.8	84.4	83.1	95.9
	SVM+L2	99.0	99.7	69.4	85.8	97.1	99.7	46.9	84.4	37.5	81.3	50.0	84.4	84.4	94.0
	SVM-SM	98.7	99.0	65.8	77.7	96.8	98.4	34.4	68.8	40.6	59.4	43.8	68.8	82.7	89.3
3D-CNN	*SM-Vote*	97.7	98.4	84.2	96.5	96.8	99.4	68.8	100	50	100	75	100	90.3	**98.3**
	SM-Prod	98.4	100	86.8	96.1	97.4	99.4	62	87.4	53.1	96.9	78.1	100	**91.4**	98.2
	SVM+L2	96.6	98.7	78.7	91.6	96.8	99.4	71.9	96.9	46.9	90.6	68.8	96.9	88.1	96.4
	SVM-SM	98.4	99	83.6	88.4	96.4	98.4	68.8	81.3	53.1	65.6	75	87.5	90.3	93.7
RESNET-A	*SM-Vote*	77.7	99.4	60	91.6	71.8	97.4	56.3	81.3	37.5	81.3	46.9	90.6	67.7	95
	SM-Prod	77.1	98.4	60	91.3	70.9	96.1	56.3	78.1	34.4	81.3	46.9	87.5	67.1	94.1
	SVM+L2	99.4	99.7	91.6	97.7	97.1	99.4	31.3	50	21.9	46.9	21.9	53.1	89.4	94.4
	SVM-SM	98.4	99.4	86.5	96.5	87.9	93.6	50	62.5	34.4	53.1	46.9	62.5	86.5	93

on the temporal scenarios (*TN*, *TB* and *TS*), the worst results are obtained with *Gray*. These results evidence the weakness of appearance models under conditions with high variability between training and test samples (like our temporal experiment). However, *OF* models have a better sturdiness against appearance changes on the inputs. On average, the best results are obtained when using optical flow (*OF*) as base for extracting the gait signature.

With regard to the type of architecture, the behaviour of all of them is very similar on the non-temporal scenarios. However, for the temporal scenarios, 3D-CNN offers its best results in combination with either *OF* or *Depth*, whereas 2D-CNN and ResNet work better with *Gray*. Considering the average accuracy over all the scenarios, 2D-CNN works better with *Gray*, and 3D-CNN with both *OF* and *Depth*.

Finally, note that all the strategies employed for obtaining the identity at video level offer similar performance. However, *SM-Prod* seems to work slightly better on average. Recall that it is defined as the product of probabilities obtained at the softmax layer (see Sec. 4.3), what does not require to train an additional classifier as SVM.

5.4.2. Feature fusion

As we can use three types of low-level features from TUM-GAID, we study here the benefits of fusing information from the different sources. We are going to use as basis the data of Tabs. 1, 2 and 3, concretely, data obtained with the *product of softmax vectors* on each modality. We have experimented with three types of fusion methods for all the combinations that include optical flow, chosen due to its sturdiness under all walking conditions.

Firstly, we analyse the results within each type of architecture. The results of Tab. 4 correspond to 2D-CNN and indicate that, in general, the best option is to combine all three modalities for all fusion methods except for *SM Prod* where it is better to use only *OF* and *Gray*. Note that for *Weighted Sum*, we have used the weights $0.4, 0.3$ and 0.3 for *OF*, *Gray* and *Depth*, respectively, when we fuse all modalities. In the case of only two modalities, we use weights 0.6 and 0.4 for *OF* and the other modality, respectively. According to the average results, all fusion approaches improves the single modality results what encourages the use of multiple modalities. Regarding the fusion strategy, the proposed *Early* fusion CNN provides on average the best results.

Table 4: **Fusion strategies in TUM-GAID with 2D-CNN.** Percentage of correct recognition for different modalities and fusion methods. Each row corresponds to a different fusion strategy. Best results are marked in bold.

Fusion	Modalities	*N*	*B*	*S*	*TN*	*TB*	*TS*	*AVG*
Single	Gray	100	99.7	98.4	28.1	37.5	34.4	93.2
	OF	99.4	97.7	96.1	56.3	43.8	59.4	93.6
	Depth	98.7	66.1	96.8	43.8	40.6	46.8	83.1
SM Prod	OF-Gray	99.7	99.7	99.0	40.6	37.5	53.1	94.3
	OF-Depth	92.9	88.1	98.7	59.4	40.6	46.9	89.1
	All	92.9	90.0	99.0	56.3	56.3	50.0	90.2
W. Sum	OF-Gray	99.4	98.4	98.7	50.0	34.4	53.1	93.9
	OF-Depth	97.7	93.9	99.0	53.1	43.8	59.4	92.7
	All	99.0	98.1	99.7	50.0	34.4	53.1	94.0
Early	OF-Gray	100	96.8	98.4	56.3	56.3	53.1	94.4
	OF-Depth	99.4	88.4	98.1	50.0	56.3	46.9	91.2
	All	99.4	98.4	98.4	50.0	62.5	59.4	**94.9**

Focusing on the results obtained with the 3D-CNN (Tab. 5), the best average accuracy is reported by the combination of all modalities by *W Sum*. However, it is only slightly better than the best result obtained by using only *OF*. Due to the low accuracy obtained with *Gray*, combining it with other features worsen the fused results.

Table 5: **Fusion strategies in TUM-GAID with 3D-CNN.** Percentage of correct recognition for different modalities and fusion methods. Each row corresponds to a different fusion strategy. Best results are marked in bold.

Fusion	Modalities	*N*	*B*	*S*	*TN*	*TB*	*TS*	*AVG*
Single	Gray	97.7	93.9	91.3	18.8	21.9	12.5	87.1
	OF	98.7	97.1	94.5	71.9	68.8	65.6	94.1
	Depth	98.4	86.8	97.4	62	53.1	78.1	91.4
SM Prod	OF-Gray	93.5	84.8	83.5	12.5	12.5	15.6	80.4
	OF-Depth	92.2	97.4	96.8	78.1	62.5	15.6	91.4
	All	78.4	84.2	83.5	12.5	21.9	12.5	75.8
W. Sum	OF-Gray	97.4	98.1	96.1	71.9	50	53.1	93.6
	OF-Depth	95.5	96.5	96.8	65.6	68.8	53.1	93.1
	All	96.8	98.4	97.1	65.6	65.6	59.4	**94.3**
Early	OF-Gray	99.4	96.8	94.5	62.5	50	56.3	93.1
	OF-Depth	84.8	97.4	97.4	71.9	68.8	71.9	91.1
	All	99.7	98.7	97.7	34.4	25	31.3	92.3

Finally, the ResNet architecture (see Tab. 6) shows unexpected low fusion results. It may indicate that the probability distribution on the classes obtained at the softmax layer does not show clearly defined maxima, and small changes in those values cause important changes in the final classes. However, *Early Fusion* improves the results on average for the combination *OF* and *Gray*, what indicates that adding more inputs to the training process can be beneficial to avoid local minima.

In summary, by using multimodal information the recognition accuracy improves 0.9% with respect to the best single modality (i.e. *OF*).

Table 6: **Fusion strategies in TUM-GAID with ResNet.** Percentage of correct recognition for different modalities and fusion methods. Each row corresponds to a different fusion strategy. Best average results are marked in bold.

Fusion	Modalities	*N*	*B*	*S*	*TN*	*TB*	*TS*	*AVG*
Single	Gray	99.0	96.5	95.5	28.1	34.4	25.0	90.7
	OF	95.2	81.0	86.1	37.5	40.6	43.8	83.1
	Depth	77.1	60.0	71.0	56.3	34.4	46.9	67.2
SM Prod	OF-Gray	84.8	77.7	79.3	46.9	40.6	50	77.3
	OF-Depth	71.2	63.6	69.6	53.1	37.5	53.1	66.2
	All	79.9	80.7	81.9	56.3	34.4	56.3	77.9
W. Sum	OF-Gray	72.8	60.7	64.4	31.3	31.3	40.6	63
	OF-Depth	68.3	53.9	62.5	37.5	46.9	56.3	60.2
	All	72.5	60	64.7	31.3	28.1	46.9	62.9
Early-RES	OF-Gray	99.4	94.8	97.7	40.6	34.4	43.8	**91.9**
	OF-Depth	95.8	93.2	96.1	40.6	37.5	43.8	89.9
	All	80.3	87.1	88.4	40.6	50	50	81.7

5.4.3. State-of-the-art on TUM-GAID

In Tab. 7, we compare our results with state-of-the-art in TUM-GAID under all modalities previously employed (*Gray*, *OF*, *Depth* and *Fusion*). First of all, we would like to remark that our approach uses a resolution of 80 × 60 while the rest of methods use 640 × 480. Therefore, our method uses 64 times less information. If we focus on the visual modality (*Gray* in our case), we can see that our method outperforms previous results in non temporal scenarios establishing a new state-of-the-art . On the other hand, in the temporal scenarios we have lower results than the other methods due to the high variability in visual information. Then, if we focus on *OF*, we can see that the best results are obtained by PFM [48] with a resolution of 640 × 480. Nevertheless, if we apply PFM with a resolution of 80 × 60, its results worsen dramatically and our CNN is able to outperform it in all scenarios. If we compare our CNN with other deep learning approaches presented in the literature, only MTaskCNN-7NN [28] is able to improve our approach. This model has been trained in a multi-task fashion so, during training, there are more information available to optimize the weights. If we focus on the other deep learning approaches, we can see that we obtain similar results (only a 0.2% lower) on average but, we obtain the state-of-the-art for temporal scenario. In *Depth* modality, we can see that our method obtains better results than other methods, which use full resolution frames, in all cases except *N*. Nevertheless, on average, we are able to obtain more than a 10% of improvement. Finally, if we fuse information from all modalities with a CNN, the average score achieved by both scenarios (temporal and non-temporal) beats all the methods shown in Tab. 7 with the exception of PFM (640x480) [48] and MTaskCNN-7NN [28], where we are 1.5% and 1.1% below, respectively. However, if we apply the same 7NN approach as in [28], and we fuse the probabilities obtained, we set a new state-of-the-art (96.5% vs 96.0%) for all scenarios with our 3D-CNN-7NN-All using Softmax Product as fusion.

5.5. Experimental results on CASIA-B

We focus here on CASIA-B dataset, which offers different covariate factors and multiple viewpoints. Note that, for the sake of comparison with other methods, we train our models with all cameras and the test is performed only with the 90° camera as done in state-of-the-art approaches [8, 48].

Table 7: **State-of-the-art on TUM GAID**. Percentage of correct recognition on TUM-GAID for diverse methods published in the literature. Bottom rows of each modality correspond to our proposal, where instead of using video frames at 640 × 480, a resolution of 80 × 60 is used. Each column corresponds to a different scenario. Best results are marked in bold. (See main text for further details).

Modality	*Input Size*	*Method*	*N*	*B*	*S*	*Avg*	*TN*	*TB*	*TS*	*Avg*	*Global Avg*
Visual Data	640 × 480	SDL [54]	-	-	-	-	96.9	-	-	-	-
		GEI [37]	99.4	27.1	52.6	59.7	44.0	6.0	9.0	19.7	56.0
		SEIM [55]	99.0	18.4	96.1	71.2	15.6	3.1	28.1	15.6	66.0
		GVI [55]	99.0	47.7	94.5	80.4	62.5	15.6	62.5	46.9	77.3
		SVIM [55]	98.4	64.2	91.6	84.7	65.6	31.3	50.0	49.0	81.4
		RSM [56]	100	79.0	97.0	92.0	58.0	38.0	57.0	51.0	88.2
	Gray 80 × 60	2D-CNN-SMP **(ours)**	100	99.7	98.4	99.4	28.1	37.5	34.4	33.3	93.2
OF	640 × 480	PFM [48]	99.7	99.0	99.0	99.2	78.1	62.0	54.9	65.0	96.0
	80 × 60	PFM [48]	75.8	70.3	32.3	59.5	50.0	40.6	25.0	38.5	57.5
		OF-CNN-NN [26]	99.7	98.1	95.8	97.9	62.5	56.3	59.4	59.4	94.3
		OF-ResNet-B [27]	99	95.5	97.4	97.3	65.6	62.5	68.8	65.6	94.3
		MTaskCNN-7NN [28]	99.7	97.4	99.7	98.9	59.4	62.5	68.8	63.6	95.6
		3D-CNN-SMP **(ours)**	98.7	97.1	94.5	96.8	71.9	68.8	65.6	**68.8**	94.1
Depth	640 × 480	DGHEI [37]	99.0	40.3	96.1	78.5	50.0	0.0	44.0	31.3	74.1
	80 × 60	3D-CNN-SMP **(ours)**	98.4	86.8	97.4	94.2	62.0	53.1	78.1	64.4	91.4
Fusion	640 × 480	DGHEI + GEI [37]	99.4	51.3	94.8	81.8	66.0	3.0	50.0	39.7	77.9
	80 × 60	2D-CNN-All **(ours)**	99.4	98.4	98.4	98.7	50.0	62.5	59.4	57.3	94.9
		3D-CNN-7NN-All **(ours)**	100	99.4	99.4	**99.6**	75	62.5	62.5	66.7	**96.5**

5.5.1. Architecture and feature evaluation

As this dataset contains eleven viewpoints, ResNet models have enough variability in the training data. Therefore, we use ResNet-B (see Sec. 4.2 for more details) which is deeper than ResNet-A. Tab. 8 summarizes the identification results obtained on CASIA-B 90^o with each modality: *Gray* and *OF*. Note that this dataset does not provide depth information. R1 and R5 columns contain the results for rank-1 (R1) and rank-5 (R5) for each scenario. The last column '*AVG*' is the average of all scenarios. The results at sequence level are obtained by multiplying the scores of the softmax layer. Note that as in CASIA-B there is no training partition to build the model, we have split the dataset into a training set composed of the first 74 subjects and a test set composed of the 50 remaining subjects, following the indications in [8]. During the training process, all viewpoints and training samples are used.

Table 8: **Feature selection on CASIA-B 90°: *Gray* and *OF* modalities.** Percentage of correct recognition by using *rank-1* (R1) and *rank-5* (R5) metrics. Each row corresponds to a different classifier and modality, grouped by architecture. Each column corresponds to a different scenario. Best average results are marked in bold.

			nm		*bg*		*cl*		*AVG*	
			R1	R5	R1	R5	R1	R5	R1	R5
Gray	2D	*SM-Vote*	91	98	82	95	37	82	70	91.7
		SM-Prod	92	100	85	98	45	90	74	96
	3D	*SM-Vote*	72	93	69	87	33	76	58	85.3
		SM-Prod	81	92	73	90	45	77	66.3	86.3
	RES	*SM-Vote*	94	100	89	98	42	83	75	93.7
		SM-Prod	96	100	91	98	46	98	**77.7**	**98.7**
OF	2D	*SM-Vote*	99	99	76	90	28	51	67.7	80
		SM-Prod	99	99	78	93	27	62	68	84.7
	3D	*SM-Vote*	98	99	86	99	37	70	73.7	89.3
		SM-Prod	98	100	88	98	36	67	74	88.3
	RES	*SM-Vote*	94	100	83	98	47	73	74.7	90.3
		SM-Prod	93	100	85	98	46	71	74.7	89.7

According to the results obtained, we can see that our model is able to identify people with a high accuracy in scenarios *nm* and *bg* while in scenario *cl* we have lower precision due to the high appearance changes. If we focus on the modality used, on average, *Gray* is the best option most of the time. In this dataset, with huge variations between points of view, the shape of the subject seems to be important and it helps to classification. In scenario *cl* our models experiment a huge decrease in accuracy mainly caused by the high variability of coats worn by the subjects. This can be seen in Fig. 5 on the right part of the last row. In these pictures, the coat occludes the legs and if we add the fact that we have different kind of coats with different number of occurrences, our CNN is not able to learn good features for this scenario due to the high variability and low number of samples.

On the other hand, *OF* seems that it is not able to find a good representation if the shape of the subject changes drastically. We think that this is because of the high variability in the appearance of the subjects seen from the different cameras used for training. Therefore, as the models receive different flow vectors, the training process cannot produce a view-independent model and the global performance decreases. For example, frontal-views produce vectors whose main movement is focused on Y-axis (there is no horizontal displacement of the subject) while lateral-views produce vectors whose movement is focused on X-axis.

If we focus on the different architectures, according to the mean results, it is clear that ResNet-B obtains the best results for each modality. That shows that ResNet is the more powerful model if data with enough variability is available. On the other hand, 3D-CNN obtains really good results for *OF* modality while 2D-CNN achieves good results for *Gray* modality.

5.5.2. Feature fusion

In this case, as we only have two modalities, fusion is performed using both. It can be observed in Tab. 9 that the best method for fusing *Gray* and *OF* features is, on average, Softmax product followed by weighted sum with weights 0.5 and 0.5 for *Gray* and *OF*, respectively. Focusing on the three architectures, again, the best option is ResNet as it obtains the best results in all cases.

In this dataset, the late fusion of both modalities improves or obtains the same result as single modality CNNs in all cases, apart from special cases where the difference between the accuracy of the fused modalities is huge (*e.g. cl* for 2D-CNN). Anyway, on average, fusion always obtains the best results with improvements of more than a 3%. In this case, early fusion is not able to improve the single modality results. In our opinion, this is due to high variability between viewpoints. In addition, we have observed that the two branches of the network have different convergence speeds, hence the final features are not fused properly producing bad representations.

Table 9: **Fusion strategies in CASIA-B 90°.** Percentage of correct recognition with different fusion methods. Each row corresponds to a different fusion method, but the two top rows that correspond to the baseline cases. Best average results are marked in bold.

	2D-CNN				*3D-CNN*				*ResNet*			
	nm	*bg*	*cl*	*AVG*	*nm*	*bg*	*cl*	*AVG*	*nm*	*bg*	*cl*	*AVG*
Gray	92	85	45	74	81	73	45	66.3	96	91	46	77.7
OF	99	78	27	68	98	88	36	74	93	85	46	74.7
SM-Prod	99	95	41	**78.3**	98	96	49	**81**	98	97	63	**86**
W. Sum	99	94	39	77.3	98	95	46	79.7	98	96	60	84.7
Early	83	61	26	56.7	76	74	46	65.3	67	63	38	56

5.5.3. State-of-the-art on CASIA-B

In Tab. 10, we compare our results with state-of-the-art in CASIA-B under all modalities used before (*Gray* and *OF*) and their fusion. First of all, we would like to remark that our approach uses a resolution of 80 × 60 while the rest of methods use 320 × 240. Therefore, our method uses 16 times less information. If we focus on the visual modality (*Gray* in our case), we can see that our ResNet-B obtains the best results compared to the state-of-the-art even using the lowest resolution. Indeed, our model sets a new state-of-the-art for visual data. If we focus on *OF*, the best results are obtained by PFM [48] with a resolution of 640 × 480. Nevertheless, if we apply it with a resolution of 80 × 60, its results worsen dramatically and our ResNet-B is able to outperform it in all scenarios. With this modality, our model sets the second best result in the state-of-the-art (apart from PFM with full resolution). Finally, our fusion (softmax product) sets the best result and it improves our ResNet-B for *Gray* modality by a 8.3%.

Focusing on [8], which is the closest approach to ours as they use also CNNs, our best average result improves a 16.3% with respect to their best average accuracy. Focusing on individual scenarios, they only improve our results in *cl* scenario if we use a single modality, probably due to the use of a gallery-probe scheme. During test time, they must compare the test sample with all the probe samples to get all distances, then, they select the class of the probe sample with the lowest distance. This approach is easier but slower than our approach where we only need to propagate the test sample through the CNN to obtain the class. However, if we use our fusion approach, we beat them in all scenarios and we miminize the changes in the shape of *cl* scenario.

Table 10: **State-of-the-art on CASIA-B, camera 90°**. Percentage of correct recognition for several methods on camera 90°. Bottom rows of each modality correspond to our proposal, where instead of using video frames at 640 × 480, a resolution of 80 × 60 is used. Acronyms: '#subjs' number of subjects used for test; '#train' number of sequences per person used for training; '#test' number of sequences per person used for test. Best results are marked in bold.

Modality	*Input Size*	*Method*	#subjs	#train	#test	*nm*	*bg*	*cl*	*Avg*
Visual Data	320 × 240	GEI [49]	124	4	2	97.6	52.0	32.7	67.8
		GEI [49]	124	4	2	97.6	52.0	32.7	67.8
		iHMM [57]	84	5	1	94.0	45.2	42.9	60.7
		CGI [58]	124	1	1	88.1	43.7	43.0	58.3
		SDCNN [25]	124	4	2	95.6	-	-	-
	126 × 126	DCNN [23]	124	4	2	81.5	-	-	-
	88 × 128	LBCNN [8]	50	4	2	91.5	63.1	54.6	69.7
	Gray 80 × 60	ResNet-B **(ours)**	50	4	2	96.0	91.0	46.0	**77.7**
OF	320 × 240	PFM [48]	124	4	2	100	100	85.5	**95.2**
	80 × 60	PFM [48]	124	4	2	88.3	66.5	44.0	66.3
		ResNet-B **(ours)**	50	4	2	93.0	85.0	46.0	74.7
Fusion	80 × 60	ResNet-B-SMP **(ours)**	50	4	2	98.0	97.0	63.0	**86.0**

5.6. Released material

In order to make reproducible the experimental results obtained in this paper, the CNN models obtained during the experiments have been publicly released for the research community at the following website:
`www.uco.es/~in1majim/research/cnngait.html`

After the review process, we also plan to release the related source code for reproducing the experiments.

6. Conclusions

We have presented a comparative study of multi-feature systems based on CNN architectures for the problem of people identification based on the way the walk (gait). The evaluated architectures are able to extract automatically gait signatures from sequences of gray pixels, optical flow and depth maps. Those gait signatures have been tested on the task of people identification, obtaining state-of-the-art results on two challenging datasets, *i.e.* TUM-GAID and CASIA-B, that cover diverse scenarios (*e.g.* people wearing long coats, carrying bags, changing shoes or camera viewpoint changes).

With regard to the type of input features, we may conclude that, under similar viewpoints (*e.g.* TUM-GAID) the weakest one is *gray pixels*, as it is highly appearance dependant. However, as it could be expected *optical flow* is the one that better encodes body motion. Depth maps work fairly well if changes in appearance are small (*i.e. Shoes* scenario). In datasets with multiple viewpoints (*e.g.* CASIA-B), *gray pixels* achieve the best results, probably due to *optical flow* produces extremely different vectors depending on the viewpoint so, during training, the optimization process is not able to build a good multiview representation of the subjects.

Regarding the type of architecture, 2D-CNN produces better results in most cases; 3D-CNN is specially useful in scenarios with appearance changes; ResNet models are designed to be very deep, therefore, they need huge datasets with high variability between samples to perform well. This has been demonstrated in our experiments where ResNet-A produces worse results than the other two architectures for TUM-GAID (dataset with low variability) but, on the other hand, ResNet-B produces the best results for CASIA-B (dataset with high variability).

Finally, the experimental results show that the fusion of multiple features allows to boost the recognition accuracy of the system in many cases or at least, it matches the best results achieved by using a single modality.

As final recommendation and, according to the results obtained, the best models are 3D-CNN and ResNet, being the latter the best option if the dataset contains enough training data. Regarding to fusion methods, the best option is late fusion approaches and, in our case, product of the softmax scores. As future work, we plan to study in depth our early fusion approach to solve the problem of different convergence rates in the branches.

Acknowledgements

This work has been funded by project TIC-1692 (Junta de Andalucía). We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan X Pascal GPU used for this research. Portions of the research in this paper use the CASIA Gait Database collected by Institute of Automation, Chinese Academy of Sciences.

References

[1] A. Agarwal, R. Keshari, M. Wadhwa, M. Vijh, C. Parmar, R. Singh, M. Vatsa, Iris sensor identification in multi-camera environment, Information Fusion 2018.

[2] D. Peralta, I. Triguero, S. García, F. Herrera, J. M. Benitez, DPD-DFF: A dual phase distributed scheme with double fingerprint fusion for fast and accurate identification in large databases, Information Fusion 32 (2016) 40 – 51.

[3] T. B. Moeslund, A. Hilton, V. Kruger, A survey of advances in vision-based human motion capture and analysis, Computer Vision and Image Understanding 104 (2006) 90–126.

[4] P. Turaga, R. Chellappa, V. S. Subrahmanian, O. Udrea, Machine recognition of human activities: A survey, Circuits and Systems for Video Technology, IEEE Transactions on 18 (11) (2008) 1473–1488.

[5] K. Soomro, A. R. Zamir, M. Shah, UCF101: A dataset of 101 human action classes from videos in the wild, in: CRCV-TR-12-01, 2012.

[6] W. Hu, T. Tan, L. Wang, S. Maybank, A survey on visual surveillance of object motion and behaviors, Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on 34 (3) (2004) 334–352.

[7] J. Han, , B. Bhanu, Individual recognition using gait energy image, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (2) (2006) 316322.

[8] Z. Wu, Y. Huang, L. Wang, X. Wang, T. Tan, A comprehensive study on cross-view gait based human identification with deep CNNs, IEEE Transactions on Pattern Analysis and Machine Intelligence 39 (2) (2017) 209–226.

[9] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in Neural Information Processing Systems, 2012, pp. 1097–1105.

[10] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778.

[11] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, 2016, `http://www.deeplearningbook.org`.

[12] M. Marín-Jiménez, N. P. de la Blanca, M. Mendoza, M. Lucena, J. Fuertes., Learning action descriptors for recognition, in: IEEE (Ed.), WIAMIS 2009, Vol. 0, London, UK, IEEE Computer Society, 2009, pp. 5–8.

[13] M. J. Marín-Jiménez, N. P. De La Blanca, M. A. Mendoza, Rbm-based silhouette encoding for human action modelling, in: Proceedings of the International Conference on Pattern Recognition, IEEE, 2010, pp. 979–982.

[14] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, CoRR abs/1409.1556.

[15] M. D. Zeiler, R. Fergus, Proceedings of the european conference on computer vision (eccv), 2014, pp. 818–833.

[16] Q. V. Le, W. Y. Zou, S. Y. Yeung, A. Y. Ng, Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2011, pp. 3361–3368.

[17] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, L. Fei-Fei, Large-scale video classification with convolutional neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2014, pp. 1725–1732.

[18] K. Simonyan, A. Zisserman, Two-stream convolutional networks for action recognition in videos, in: Advances in Neural Information Processing Systems, 2014, pp. 568–576.

[19] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, T. Darrell, Long-term recurrent convolutional networks for visual recognition and description, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015.

[20] L. Wang, Y. Qiao, X. Tang, Action recognition with trajectory-pooled deep-convolutional descriptors, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 4305–4314.

[21] F. Perronnin, D. Larlus, Fisher vectors meet neural networks: A hybrid classification architecture, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 3743–3752.

[22] E. Hossain, G. Chetty, Multimodal feature learning for gait biometric based human identity recognition, in: Neural Information Processing, 2013, pp. 721–728.

[23] Z. Wu, Y. Huang, L. Wang, Learning representative deep features for image set analysis, IEEE Trans. on Multimedia 17 (11) (2015) 1960–1968.

[24] B. Gálai, C. Benedek, Feature selection for lidar-based gait recognition, in: Computational Intelligence for Multimedia Understanding (IWCIM), 2015 International Workshop on, 2015, pp. 1–5.

[25] M. Alotaibi, A. Mahmood, Improved gait recognition based on specialized deep convolutional neural networks, in: IEEE Applied Imagery Pattern Recognition Workshop (AIPR), 2015, pp. 1–7.

[26] F. M. Castro, M. J. Marín-Jiménez, N. Guil, N. Pérez de la Blanca, Automatic learning of gait signatures for people identification, Advances in Computational Intelligence: 14th International Work-Conference on Artificial Neural Networks (IWANN) (2017) 257–270.

[27] F. M. Castro, M. J. Marín-Jiménez, N. Guil, S. López-Tapia, N. P. de la Blanca, Evaluation of cnn architectures for gait recognition based on optical flow maps, in: BIOSIG, 2017, pp. 251–258.

[28] M. J. Marín-Jiménez, F. M. Castro, N. Guil, F. de la Torre, R. Medina-Carnicer, Deep multitask learning for gait-based biometrics, in: Proceedings of the IEEE International Conference on Image Processing, 2017.

[29] D. Holden, J. Saito, T. Komura, T. Joyce, Learning motion manifolds with convolutional autoencoders, in: SIGGRAPH Asia 2015 Technical Briefs, 2015, p. 18.

[30] N. Neverova, C. Wolf, G. Lacey, L. Fridman, D. Chandra, B. Barbello, G. Taylor, Learning human identity from motion patterns, IEEE Access 4 (2016) 1810–1820.

[31] D. Tran, L. D. Bourdev, R. Fergus, L. Torresani, M. Paluri, Learning spatiotemporal features with 3d convolutional networks, in: Proceedings of the International Conference on Computer Vision (ICCV), IEEE, 2015.

[32] T. Wolf, M. Babaee, G. Rigoll, Multi-view gait recognition using 3D convolutional neural networks, in: Proceedings of the IEEE International Conference on Image Processing, 2016, pp. 4165–4169.

[33] E. Mansimov, N. Srivastava, R. Salakhutdinov, Initialization strategies of spatio-temporal convolutional neural networks, CoRR abs/1503.07274.

[34] P. K. Atrey, M. A. Hossain, A. El Saddik, M. S. Kankanhalli, Multimodal fusion for multimedia analysis: a survey, Multimedia Systems 16 (6) (2010) 345–379.

[35] S. Wu, Applying statistical principles to data fusion in information retrieval, Expert Systems with Applications 36 (2) (2009) 2997–3006.

[36] Y. Chai, J. Ren, H. Zhao, Y. Li, J. Ren, P. Murray, Hierarchical and multi-featured fusion for effective gait recognition under variable scenarios, Pattern Analysis and Applications (2015) 1–13. information fusion

[37] M. Hofmann, J. Geiger, S. Bachmann, B. Schuller, G. Rigoll, The TUM Gait from Audio, Image and Depth (GAID) database: Multimodal recognition of subjects and traits, Journal of Visual Communication and Image Representation 25 (1) (2014) 195 – 206.

[38] F. M. Castro, Marín-Jiménez, N. Guil, Empirical study of audio-visual features fusion for gait recognition, in: Proceedings of the International Conference on Computer Analysis of Images and Patterns, 2015, pp. 727–739.

[39] A. Eitel, J. T. Springenberg, L. Spinello, M. Riedmiller, W. Burgard, Multimodal deep learning for robust RGB-D object recognition, in: Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems, IEEE, 2015, pp. 681–687.

[40] A. Wang, J. Lu, J. Cai, T.-J. Cham, G. Wang, Large-margin multi-modal deep learning for RGB-D object recognition, Multimedia, IEEE Transactions on 17 (11) (2015) 1887–1898.

[41] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, P. Kuksa, Natural language processing (almost) from scratch, Journal of Machine Learning Research 12 (2011) 2493–2537.

[42] C. Farabet, C. Couprie, L. Najman, Y. LeCun, Learning hierarchical features for scene labeling, IEEE Transactions on Pattern Analysis and Machine Intelligence 35 (8) (2013) 1915–1929.

[43] X. Zhang, J. Zhao, Y. LeCun, Character-level convolutional networks for text classification, in: Advances in Neural Information Processing Systems, IEEE, 2015.

[44] S. Sivapalan, D. Chen, S. Denman, S. Sridharan, C. Fookes, Gait energy volumes and frontal gait recognition using depth images, in: Biometrics (IJCB), 2011 International Joint Conference on, IEEE, 2011, pp. 1–6.

[45] F. M. Castro, M. J. Marín-Jiménez, N. Guil, Multimodal features fusion for gait, gender and shoes recognition, Machine Vision and Applications 27 (8) (2016) 1213–1228.

[46] A. Vedaldi, K. Lenc, MatConvNet – Convolutional Neural Networks for MATLAB, in: Proceeding of the ACM Int. Conf. on Multimedia, 2015.

[47] S. Chetlur, C. Woolley, P. Vandermersch, J. Cohen, J. Tran, B. Catanzaro, E. Shelhamer, cudnn: Efficient primitives for deep learning, CoRR abs/1410.0759.

[48] F. M. Castro, M. Marín-Jiménez, N. Guil Mata, R. Muñoz Salinas, Fisher motion descriptor for multiview gait recognition, International Journal of Patt. Recogn. in Artificial Intelligence 31 (1).

[49] S. Yu, D. Tan, T. Tan, A framework for evaluating the effect of view angle, clothing and carrying condition on gait recognition, in: Proceedings of the International Conference on Pattern Recognition, Vol. 4, 2006, pp. 441–444.

[50] G. Farnebäck, Two-frame motion estimation based on polynomial expansion, in: Proc. of Scandinavian Conf. on Image Analysis, Vol. 2749, 2003, pp. 363–370.

[51] G. Bradski, OpenCV library, Dr. Dobb's Journal of Software Tools.

[52] P. KaewTraKulPong, R. Bowden, An improved adaptive background mixture model for real-time tracking with shadow detection, in: Video-Based Surveillance Systems, 2002, pp. 135–144.

[53] O. Barnich, M. V. Droogenbroeck, Frontal-view gait recognition by intra- and inter-frame rectangle size distribution, Pattern Recognition Letters 30 (10) (2009) 893 – 901.

[54] W. Zeng, C. Wang, F. Yang, Silhouette-based gait recognition via deterministic learning, Pattern Recognition 47 (11) (2014) 3568 – 3584.

[55] T. Whytock, A. Belyaev, N. Robertson, Dynamic distance-based shape features for gait recognition, Journal of Mathematical Imaging and Vision 50 (3) (2014) 314–326.

[56] Y. Guan, C.-T. Li, A robust speed-invariant gait recognition system for walker and runner identification, in: Intl. Conf. on Biometrics (ICB), 2013, pp. 1–8.

[57] M. Hu, Y. Wang, Z. Zhang, D. Zhang, J. Little, Incremental learning for video-based gait recognition with LBP flow, Cybernetics, IEEE Transactions on 43 (1) (2013) 77–89.

[58] C. Wang, J. Zhang, L. Wang, J. Pu, X. Yuan, Human identification using temporal information preserving gait template, IEEE Transactions on Pattern Analysis and Machine Intelligence 34 (11) (2012) 2164–2176.