Evaluation of CNN architectures for gait recognition based on optical flow maps

F.M. Castro [1] M.J. Marín-Jiménez[2] N. Guil [1]S. López-Tapia [3] N. Pérez de la Blanca [3]

Abstract: This work targets people identification in video based on the way they walk (*i.e.*gait) by using deep learning architectures. We explore the use of convolutional neural networks (CNN) for learning high-level descriptors from low-level motion features (*i.e.*optical flow components). The low number of training samples for each subject and the use of a test set containing subjects different from the training ones makes the search of a good CNN architecture a challenging task. We carry out a thorough experimental evaluation deploying and analyzing four distinct CNN models with different depth but similar complexity. We show that even the simplest CNN models greatly improve the results using shallow classifiers. All our experiments have been carried out on the challenging TUM-GAID dataset, which contains people in different covariate scenarios (*i.e.*clothing, shoes, bags).

Keywords: Deep Neural Networks, Gait Recognition, Optical Flow, ResNet, 3D-CNN.

1 Introduction

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. Gait recognition has application 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.

In last years, great effort has been put into the problem of people identification based on gait patterns [Hu04]. However, previous approaches have mostly used hand-crafted features for representing the human gait [BD09, HB06, Ca17], which do not easily adapt to diverse datasets, due to the specificity of the hand-crafted descriptors obtained for each dataset. Therefore, we propose an end-to-end approach based on convolutional neural networks that given low-level optical flow maps, directly extracted from video frames (see Fig. 1), is able to learn and extract higher-level features suitable for representing human gait: *gait signature*. In addition, we also present a fair comparative between four models based on three of the most popular kinds of CNN architectures used in computer vision tasks: LeNet [LB95], VGG [SZ14] and ResNet [He16]. The contribution of this paper is twofold: (*i*) a set of CNN models for gait recognition using optical flow; and, (*ii*) a thorough experimental study to validate the proposed models on the standard TUM-GAID dataset for gait identification, obtaining state-of-the-art results.

The rest of the paper is organized as follows. We continue by reviewing the related work. Then, Sec. 2 explains our four different models for learning gait signatures and identifying

[1] University of Málaga, Department of Computer Architecture, Spain
[2] University of Córdoba, Department of Computing and Numerical Analysis, Spain
[3] University of Granada, Department of Computer Science and Artificial Intelligence, Spain

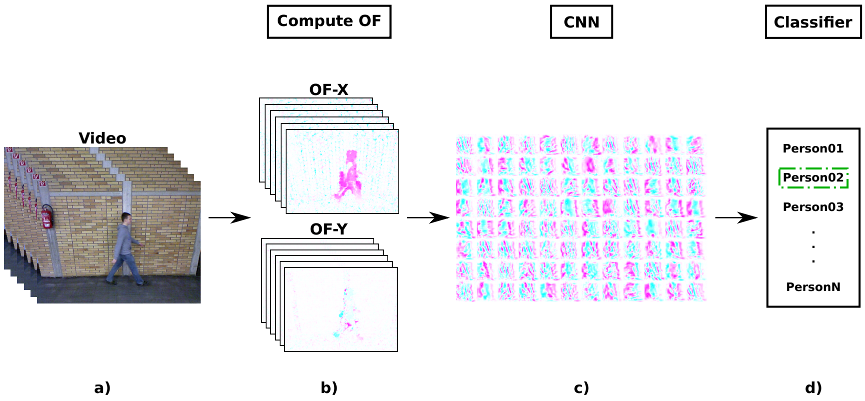

Fig. 1: **Pipeline for gait recognition**. a) The input is a sequence of RGB video frames. b) Optical flow is computed along the sequence. c) Optical flow subsequences are passed through the CNN to obtain gait signatures. e) Classification of the extracted gait signatures. Note: positive flows are in pink and negative flows in blue. (Best viewed in colour).

people. Sec. 3 contains the experiments and results. Finally, we present the conclusions and future work in Sec. 4.

1.1 Related work

Traditionally, deep learning approaches based on Convolutional Neural Networks (CNN) have been used in image-based tasks with great success [KSH12]. 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. In [SZ14], Simonyan and Zisserman proposed to use as input to a CNN a volume obtained as the concatenation of frames with two channels that contain the optical flow in the *x*-axis and *y*-axis respectively. To normalize the size of the inputs, they split the original sequences into subsequences of 10 frames, considering each subsample independently. A natural modification is presented by Ji *et al.* [Ji13], where a 3D convolutional network is developed to capture temporal information from multiple frames. Then, Tran *et al.* [Tr15] propose a new 3D network which uses raw videos as input, instead of preprocessed inputs. Recently, a new approach has been developed by He *et al.* [He16]. They propose a new kind of CNN which has a large number of layers and residual connections to avoid the vanishing gradient problem. Although several papers can be found for the task of human action recognition using deep learning techniques, it is hard to find such type of approaches applied to the problem of gait recognition. In [HC13], 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) was used for validating their approach. A more recent work, [WHW15], uses a random set of binary silhouettes from a sequence to train a CNN that accumulates the calculated features to achieve a global representation of the dataset. In [AM15], raw 2D GEI are employed to train a simple CNN for gait recognition. A more

This work has been funded under projects TIC-1692 (Junta de Andalucía), TIN2016-75279-P (Spanish Ministry of Science and Tech.) and University of Málaga (Campus de Excelencia Internacional Andalucía Tech). The GPU Titan X Pascal used for this research was donated by NVIDIA.

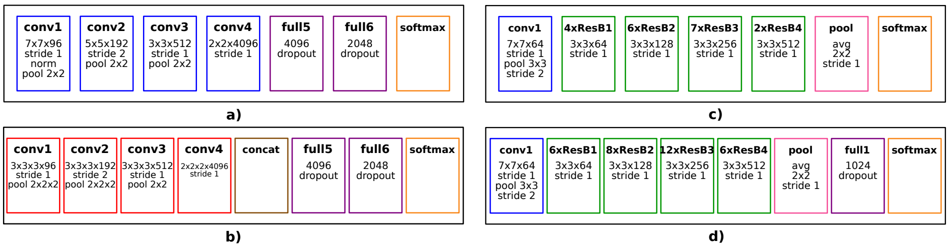

Fig. 2: **Proposed CNN models for gait signature extraction**. **a) 2D-CNN:** linear CNN with four 2D convolutions, two fully connected layers and a softmax classifier. **b) 3D-CNN:** four 3D convolutions, two fully connected layers and a softmax classifier. **c) ResNet-A:** residual CNN with a 2D convolution, four residual blocks, an average pooling layer and a final softmax classifier. **d) ResNet-B:** extended version of ResNet-A. Note that before the first block of each kind (ResB 1, 2, 3, 4), there is an adapter convolution to resize the input image to the size of the next block.

complex work is presented in [GB15] where GEI are used to train an ensemble of CNN and a Multilayer Perceptron is employed as classifier. In [Wu17], given two GEI descriptors, they learn a metric to decide whether both descriptors belong to the same subject or not. All those previous CNN-based approaches propose precomputed GEI descriptors as input features. In contrast, our approach builds a spatio-temporal volume of optical flow [SZ14] as input to a CNN specially designed for gait recognition, what will allow the CNN to learn characteristic gait patterns directly from the source, *i.e.*the motion.

2 Proposed approach

In this section we describe our proposed framework to address the problem of gait recognition using CNN. The proposed pipeline is represented in Fig. 1: *(i)* compute optical flow (OF) along the whole sequence; *(ii)* build up a data cuboid from consecutive OF maps; *(iii)* feed the different CNNs with an OF cuboid to extract the gait signature; and, *(iv)* using the gait signature, decide the subject identity.

2.1 Input data

The use of optical flow (OF) as input data for action representation in video with CNN has already shown excellent results [SZ14]. 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 original video sequences have different temporal length, and CNN requires a fixed size input, we extract subsequences of L frames from the full-length sequences.

2.2 CNN architectures for gait signature extraction

We have selected three of the 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 proposed architectures are: *(i)* the LeNet architecture [LB95], adapted to a model named (*2D-CNN*), which is the most common architecture; *(ii)* the VGG architecture [SZ14], adapted to use 3D convolutions on optical flow inputs and named (*3D-CNN*), which is specially designed to capture information in video sequences; and, *(iii)* two CNN models with residual units (named *ResNet* [He16]), used to experiment with deeper models on this task, as the network depth has been recently pointed out as one the most relevant factors to achieve the state of the art in many tasks [KSH12].

To carry out a fair comparison, three of the four models have been designed to have a similar number of parameters, where the 2D-CNN model has been taken as a reference (*i.e.*$\sim 18.5M$). This choice allows us to carry out a comparative study which is independent of the network capacity. Due to the particular design of the fourth one, it has a different number of parameters.

We describe below the four models compared in the experimental section (Sec. 3):

2D-CNN (16 layers): This CNN is composed of the sequence of layers shown in Fig. 2.a). All convolutional layers use a ReLU function and all *conv* blocks contain a max-pooling operation.

3D-CNN (16 layers): 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. Then, this CNN is composed of the sequence of layers shown in Fig. 2.b). Note that 'concat' layer concatenates both branches (*x*-flow and *y*-flow) into a single one. All convolutional layers use a ReLU function and all *conv* blocks contain max-pooling.

ResNet-A (167 layers): This CNN is composed of the sequence of layers and residual blocks (a sequence of two convolutions of size 3×3 and a sum layer, as defined in [He16]) shown in Fig. 2.c). As our model follows the indications defined in [He16], we only describe the main blocks. Note that all convolutional layers use the rectification (ReLU) activation function and batch normalization.

ResNet-B (268 layers): This CNN is an extended version of ResNet-A, composed of the sequence of layers and residual blocks shown in Fig. 2.d). Note that all convolutional layers use the parametric rectification (PReLU) [He15] activation function, local response normalization (LRN) and batch normalization. The use of PReLU is specially useful in our case as optical flow has negative components which contain important information about motion. Therefore, the network uses more information and the gradients are more powerful, avoiding the vanishing gradient problem.

2.3 Training details

For models 2D-CNN, 3D-CNN and ResNet-A, during training, the weights are learnt using mini-batch stochastic descent algorithm with momentum equal to 0.9. We set weight decay to $5 \cdot 10^{-4}$ and dropout to 0.4 (2D-CNN and 3D-CNN). The learning rate is initially set to 10^{-2} and divided by 10 when the validation error gets stuck. At each epoch, a mini-batch of 150 samples is constructed by random selection over a balanced training set (*i.e.*almost same proportion of samples per class).

As ResNet-B has some peculiarities, training parameters must be adapted. In this case, mini batches of size 64 are used. The learning rate policy follows a triangular scheme that consists of varying the learning rate between a minimum and a maximum value following a triangular pattern with the training iterations. The triangular learning rate parameters range from 0.003 to 0.015 during 4 epochs. The model was trained with a total of 24 epochs. Finally, dropout is used before each fully connected layer with a value of 0.1. Also weight decay regularization with value 0.0005 was imposed. Note that all hyperparameters have been cross-validated and only the best ones are presented in this paper.

3 Experiments and results

3.1 Dataset

TUM-GAID [Ho14] contains 305 subjects walking on four different conditions: normal walking (*N*), carrying a backpack (*B*), wearing coating shoes (*S*) and elapsed time (*TN*, *TB*, *TS*). We follow the standard experimental protocol defined by the authors of the dataset [Ho14]. Therefore, we use 100 subjects as training set, 50 different subjects as validation set and 155 different subjects as test set – note that it is distinguished between ‘subject partitions’ and ‘sequence partitions’, *i.e.*, for each subject, training, validation and test sequences are available. As we have different subjects between training and testing, it is needed to fine-tune the model with four training sequences of normal walking of the test subject partition. Note that the sequences used for fine-tuning are not used during testing. For testing, we use six sequences that have never been seen before by our model according to the partitions defined in [Ho14].

3.2 Implementation details

All videos are resized to a common resolution of 80×60 pixels, keeping the original aspect ratio of the video frames. Given the resized video sequences, we compute dense OF on pairs of frames by using the method of Farneback [Fa03] implemented in OpenCV library. In parallel, people are located in a rough manner along the video sequences by background subtraction [KB02]. 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). Finally, from the cropped OF maps, we build subsequences of 25 frames by stacking OF maps with an overlap of $\Theta\%$ frames. As this dataset is relatively small, we need to choose an intermediate overlapping rate value that allows to obtain training samples with enough variability between them. In our case, we empirically choose $\Theta = 80\%$, 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 [BD09]. Therefore, each OF volume has size $60 \times 60 \times 50$.

To increase the amount of training samples we add mirror sequences and apply spatial displacements of ± 5 pixels per axis, obtaining a total of 8 new samples from each original one. Then, mirror sequences are computed, obtaining about $270k$ training samples. Note that in Sec. 2.1, 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 whole sequence, we multiply the probabilities returned by the Softmax layer for all subsequences of the same sequence. Before feeding each sample into the CNN, the mean value of the whole training dataset is subtracted.

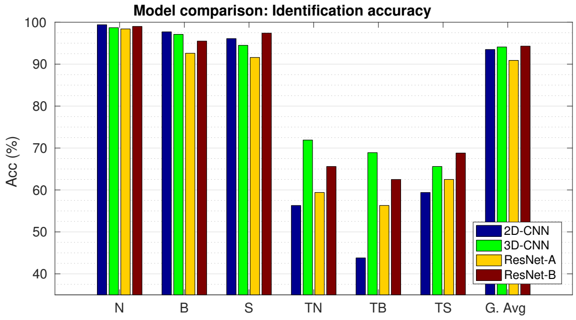

Fig. 3: **Model comparison in terms of identification accuracy**. Results grouped per scenario: normal 'N', backpacks 'B', shoes 'S' and temporal cases 'T*x*'. Group 'G.Avg' corresponds to global average on the six scenarios.

We ran our experiments on a PC with 32 cores at 2.2 GHz, 256 GB of RAM and a GPU NVIDIA Titan X Pascal, with MatConvNet library [VL15] running on Matlab 2016a for Ubuntu 14.04 and Caffe [Ji14] library for ResNet-B.

3.3 Experimental results

After splitting the training sequences (of the training subjects) into subsequences, we got a training set composed of 269352 samples used for learning the filters; and a second training set composed of 108522 samples for training the softmax layer from the subset of test subjects. Test sequences are never used for training or validation of the model.

Fig. 3 offers a visual comparison of the results obtained with each of the four tested architectures grouped per scenario type. In terms of scenario type, note that the temporal ones (T*x*) are the most challenging, as there exists a large change in subject appearance with regard to the non-temporal cases where the filters of the networks were trained.

To put our results in context, Tab. 1 contains the state-of-the-art and the comparison between the four different models (rows '2D-CNN', '3D-CNN', 'ResNet-A' and 'ResNet-B'). We have applied the PFM descriptor [Ca17] on resized videos of 80×60 to obtain a fair comparison. Comparing the CNN results with the state-of-the-art , 2D-CNN achieves on average the best results for the *non-temporal* scenarios. For the *temporal* cases, 3D-CNN obtains the best results. On global average (column '*G.Avg*'), ResNet-B sets a new state-of-the-art with an accuracy 0.2% better than the rest of CNNs and 6.1% better than the best handcrafted method. Note that CNNs use an input 16 times lower than the rest of the compared methods.

4 Discussion and Conclusions

The relevance of the complexity in CNN architectures, when applied to the gait recognition task, has been analysed through a comparative study of four models (from three deep architectures) and its comparison to results from methods based on handcrafted features. The first conclusion is that in this task, as in many others, the deep CNN architectures overcome shallow and handcrafted methods. This fact points out the importance of the architecture depth to extract relevant features. The second conclusion is that the four deep

	Method	*N*	*B*	*S*	*Avg*	*TN*	*TB*	*TS*	*Avg*	*G. Avg*
640 × 480	GEI [Ho14]	99.4	27.1	52.6	59.7	44.0	6.0	9.0	19.7	56.0
	SEIM [WBR14]	99.0	18.4	96.1	71.2	15.6	3.1	28.1	15.6	66.6
	GVI [WBR14]	99.0	47.7	94.5	80.4	62.5	15.6	62.5	46.9	77.3
	SVIM [WBR14]	98.4	64.2	91.6	84.7	65.6	31.3	50.0	49.0	81.4
	RSM [GL13]	100	79.0	97.0	92.0	58.0	38.0	57.0	51.3	88.2
80 × 60	PFM [Ca17]	75.8	70.3	32.3	59.5	50.0	40.6	25.0	38.5	57.5
	2D-CNN	99.4	97.7	96.1	**97.7**	56.3	43.8	59.4	53.2	93.5
	3D-CNN	98.7	97.1	94.5	96.7	71.9	68.9	65.6	**68.8**	94.1
	ResNet-A	98.4	92.6	91.6	94.2	59.4	56.3	62.5	59.4	90.9
	ResNet-B	99.0	95.5	97.4	97.3	65.6	62.5	68.8	65.6	**94.3**

Tab. 1: **State-of-the-art on TUM GAID**. Percentage of correct recognition on TUM-GAID for diverse methods published in the literature. Bottom rows correspond to our proposal, where instead of using video frames at 640 × 480, a resolution of 80 × 60 is used. Each column corresponds to either a different scenario or average on scenarios (*i.e.Avg*, *G.Avg*). Best results are marked in bold.

models achieve similar results in the *non-temporal* scenario, but in the *temporal* one the differences are more significant. The filters used by the 3D-CNN model make the difference in the *temporal* scenario. The standard convolutional architectures obtain the best results on the *non-temporal* and *temporal* scenarios as its design is focused on the main variations of the signal, spatial in 2D-CNN and temporal in 3D-CNN. Regarding the two ResNet models there are many differences between them in terms of design (see Fig.2) and training parameters. The ResNet-B model is a much more deeper architecture needing of PReLU activations and adaptive learning rate to obtain a good optimum. A final fully connected layer with dropout was added as well. Nevertheless and despite all these improvements, an increment of only 3.4 points in score is obtained w.r.t. ResNet-A. This result shows that the addition of residual layers although allows to fit deeper models, needs of a good learning rate policy to obtain a good optimum. The ResNet architecture achieves the overall best results when it is properly fitted. Our results reinforce, for the gait recognition task, the empirical finding of other works that indicates that architectures with enough depth are needed in order to obtain high classification accuracy. In addition, the use of appropriate activation functions has also shown to be a very relevant choice on this task. Focusing on the training speed, independently of the number of parameters, 3D-CNN needs more training time, followed by 2D-CNN and ResNet which is the fastest one.

As future work, we plan to extend our study to identify the kind of architectures more suitable to combine motion with appearance (*i.e.*RGB data), applying them to more gait datasets in which optical flow can be computed – this would allow us to perform transfer learning between networks trained on different data.

References

[AM15] Alotaibi, M.; Mahmood, A.: Improved Gait recognition based on specialized deep convolutional neural networks. In: AIPR Workshop. pp. 1–7, 2015.

[BD09] Barnich, Olivier; Droogenbroeck, Marc Van: Frontal-view gait recognition by intra- and inter-frame rectangle size distribution. Patt. Recogn. Letters, 30(10):893 – 901, 2009.

[Ca17] Castro, Francisco M.; Marín-Jiménez, M.J.; Guil Mata, N.; Muñoz Salinas, R.: Fisher Motion Descriptor for Multiview Gait Recognition. Intl. J. of Patt. Recogn. in Artificial Intelligence, 31(1), 2017.

[Fa03] Farnebäck, Gunnar: Two-Frame Motion Estimation Based on Polynomial Expansion. In: Proc. of Scandinavian Conf. on Image Analysis. volume 2749, pp. 363–370, 2003.

[GB15] Gálai, Bence; Benedek, Csaba: Feature selection for Lidar-based gait recognition. In: Computational Intelligence for Multimedia Understanding (IWCIM). pp. 1–5, 2015.

[GL13] Guan, Yu; Li, Chang-Tsun: A robust speed-invariant gait recognition system for walker and runner identification. In: Intl. Conf. on Biometrics (ICB). pp. 1–8, 2013.

[HB06] Han, Ju; Bhanu, Bir: Individual recognition using gait energy image. IEEE PAMI, 28(2):316–322, 2006.

[HC13] Hossain, Emdad; Chetty, Girija: Multimodal Feature Learning for Gait Biometric Based Human Identity Recognition. In: NIPS. pp. 721–728, 2013.

[He15] He, Kaiming; Zhang, Xiangyu; Ren, Shaoqing; Sun, Jian: Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. In: ICCV. pp. 1026–1034, 2015.

[He16] He, K.; Zhang, X.; Ren, S.; Sun, J.: Deep Residual Learning for Image Recognition. In: CVPR. pp. 770–778, June 2016.

[Ho14] Hofmann, Martin; Geiger, Jrgen; Bachmann, Sebastian; Schuller, Bjrn; Rigoll, Gerhard: The TUM Gait from Audio, Image and Depth (GAID) database: Multimodal recognition of subjects and traits. J. of Visual Com. and Image Repres., 25(1):195 – 206, 2014.

[Hu04] Hu, Weiming; Tan, Tieniu; Wang, Liang; Maybank, Steve: A survey on visual surveillance of object motion and behaviors. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, 34(3):334–352, 2004.

[Ji13] Ji, S.; Xu, W.; Yang, M.; Yu, K.: 3D Convolutional Neural Networks for Human Action Recognition. IEEE PAMI, 35(1):221–231, Jan 2013.

[Ji14] Jia, Yangqing; Shelhamer, Evan; Donahue, Jeff; Karayev, Sergey; Long, Jonathan; Girshick, Ross; Guadarrama, Sergio; Darrell, Trevor: Caffe: Convolutional Architecture for Fast Feature Embedding. arXiv preprint arXiv:1408.5093, 2014.

[KB02] KaewTraKulPong, P.; Bowden, R.: An Improved Adaptive Background Mixture Model for Real-time Tracking with Shadow Detection. In: Video-Based Surveillance Systems, pp. 135–144. 2002.

[KSH12] Krizhevsky, Alex; Sutskever, Ilya; Hinton, Geoffrey E.: ImageNet Classification with Deep Convolutional Neural Networks. In: NIPS. pp. 1097–1105, 2012.

[LB95] LeCun, Yann; Bengio, Yoshua: Convolutional networks for images, speech, and time series. The handbook of brain theory and neural networks, 3361(10), 1995.

[SZ14] Simonyan, Karen; Zisserman, Andrew: Two-stream convolutional networks for action recognition in videos. In: NIPS. pp. 568–576, 2014.

[Tr15] Tran, Du; Bourdev, Lubomir D.; Fergus, Rob; Torresani, Lorenzo; Paluri, Manohar: Learning Spatiotemporal Features with 3D Convolutional Networks. In: ICCV. IEEE, 2015.

[VL15] Vedaldi, A.; Lenc, K.: MatConvNet – Convolutional Neural Networks for MATLAB. In: Proceeding of the ACM Int. Conf. on Multimedia. 2015.

[WBR14] Whytock, Tenika; Belyaev, Alexander; Robertson, NeilM.: Dynamic Distance-Based Shape Features for Gait Recognition. Journal of Mathematical Imaging and Vision, 50(3):314–326, 2014.

[WHW15] Wu, Zifeng; Huang, Yongzhen; Wang, Liang: Learning Representative Deep Features for Image Set Analysis. IEEE Trans. on Multimedia, 17(11):1960–1968, Nov 2015.

[Wu17] Wu, Z.; Huang, Y.; Wang, L.; Wang, X.; Tan, T.: A Comprehensive Study on Cross-View Gait Based Human Identification with Deep CNNs. IEEE PAMI, 39(2):209–226, 2017.