GaitCopy: disentangling appearance for gait recognition by signature copy

RUBEN DELGADO-ESCAÑO¹, FRANCISCO M. CASTRO¹, NICOLAS GUIL¹ AND MANUEL J. MARÍN-JIMÉNEZ.²

¹Department of Computer Architecture, University of Málaga, Spain
²Department of Computer Science and Numerical Analysis, University of Córdoba, Spain

Corresponding author: R. Delgado-Escaño (e-mail: rubende@uma.es).

This work was supported in part by the projects P18-FR-3130 (Junta de Andalucía), PID2019-105396RB-I00 (Ministry of Education) and RED2018-102511-T. We gratefully acknowledge the support of NVIDIA Corporation with the donation of a Titan X Pascal.

ABSTRACT This paper addresses the problem of gait-based people identification by copying optical flow-based signatures. The proposed model, coined as GaitCopy, receives as input a stack of gray images and returns the gait signature of the represented subject. The novel property of this network is that it is not trained to only generate discriminative signatures, but to copy signatures generated by a Master network trained on optical flow inputs. Then, GaitCopy is enforced to extract signatures based on motion and not based on appearance, despite having been trained with pixel inputs. We implement two different versions of GaitCopy, one mainly composed of 3D convolutional layers to capture local temporal information; and a second one based on GaitSet which uses 2D convolutional layers under a temporal setup. We evaluate our approach on two public gait datasets: CASIA-B and TUM-GAID. We observe that compact networks, up to $\times 4.2$ smaller for TUM-GAID, can be obtained by using our approach, while keeping a competitive recognition accuracy with respect to the state of the art, and without the need of explicit optical flow computation. Even with such network compression, the results obtained in TUM-GAID are comparable to those of the state of the art, with an average accuracy of 97% on the test set. (Code will be publicly available upon acceptance.)

INDEX TERMS Gait, master-student, CNN, biometrics.

I. INTRODUCTION

The term ‘gait’ refers to the manner in which a person walks or runs. Since each person walks in a very particular way (i.e. her ‘gait signature’), gait has become a popular biometric cue that allows people identification, even at a distance [1], [2]. In the last few years there has been significant progress in gait recognition [3], however, the task is far from solved.

Traditionally, gait recognition has been tackled using appearance-related features such as RGB images [4] or shape-based features such as silhouettes [5]. However, these types of features are sensitive to changes either in the scene (e.g. lighting or clutter background) or in the subject itself (change in clothing or other factors produced by the passage of time). These problems can be mitigated using motion-based features of the subject rather than appearance ones as gait, i.e. walking, is closely related to motion.

In computer vision, the lowest level feature used to encode motion in a sequence of images is optical flow (i.e. the direction and magnitude of each pixel displacement). There exist some works in the field of gait recognition that use optical flow [6], [7]. However, the computation of optical flow might be considered computationally expensive for some particular applications.

In this context, this work proposes GaitCopy, a deep neural network for gait recognition that receives as input a sequence of gray-scale images, containing a person walking, and generates a gait signature equivalent to the one that might be obtained by a model that received as input a sequence of optical flow maps – alleviating this way the computational cost of optical flow computation – as illustrated in Fig. 1. Note that, in that figure, for the sake of simplicity, we just represent the magnitude of the optical flow vectors with different colours, where the brighter the pixel the higher the value of the magnitude.

The main contribution of this paper is a novel learning framework for gait signature extraction able to disentangle appearance from identity thanks to motion cues. In particular, GaitCopy generates gait signature that copies the properties of the optical flow features without the need of explicit
optical flow computation at inference time. The experimental results on CASIA-B and TUM-GAID datasets indicates that the recognition accuracy achieved by using the learned signatures is in-pair or improves the one obtained by the optical flow-based ones. In addition, the amount of parameters used in the student networks on TUM-GAID is up to $\times 4.2$ smaller compared to the master ones, what is an important feature to implement the networks in low-resource systems.

The rest of the paper is organized as follows. After presenting the related work in Sec. II, our model is described in Sec. III. Then, in Sec. IV, we define and discuss the experimental results. Finally, Sec. V concludes the paper.

II. RELATED WORK

Nowadays, most gait recognition approaches use silhouettes as input data, since they encode the body shape information ignoring details such as textures or clothing. The simplest approaches use Gait Energy Images (GEI) or features derived from it [8], [9]. In these methods, the whole video sequence is summarized into a single image containing spatio-temporal information. Recent approaches [5], [10] use stacks of random silhouettes to represent the gait information, where each frame is handled independently to extract features that are combined with other frame features to build the final gait signature. However, all these approaches are sensitive to changes in the body shape produced by clothing or camera view-points. Recently, optical flow maps have been used to represent gait, since they are easy to compute and focus on the subject’s motion, independently of body shape. For example, Castro et al. [6] use different CNN models to identify subjects from optical flow maps.

A direct extension of these previous gait approaches, which focus on a single kind of input data, is the use of multi-modal models where more than one kind of data (e.g. optical flow, gray images, depth maps) are used as input. Kumar et al. [11] use data obtained from multiple inertial sensors to obtain a 3D-skeleton representation together with video images. In [7], it is proposed a CNN model that uses optical flow, depth and gray images at the same time to improve the global gait accuracy of the model. Zhang et al. [12] use RGB inputs to obtain an intermediate representation based on skeletons to disentangle appearance details. Then, global temporal descriptors are obtained with a LSTM from the intermediate skeletons.

In a non-gait context, Nagrani et al. [13] combine faces and voice to identify people, trying to find a common embedding. In [14], the authors propose a multimodal CNN for action recognition where the model learns a global multimodal embedding from multiple modalities and uses this embedding to recalibrate the input features, obtaining better global performance. Garcia et al. [15] assign teacher-student roles to each modality according to its loss value and updates the teacher branch to maximize the accuracy and the student branches to replicate the behavior of the teacher branch. A different approach is presented in [16], [17], where a student model learns from RGB information to produce features or signatures similar to optical flow ones obtained from a teacher model. Then, both are combined to boost the final results, without the need of optical flow computation at test time. In our case, to the best of our knowledge, we apply for the first time a similar idea to the task of gait recognition. Where, in contrast to [16], no feature fusion is performed, as we want to get rid of optical flow computation at inference time, directly using the data stream (i.e. pixels) provided by the camera.

Teacher-student approaches also use signatures to compress deep learning models though knowledge distillation between predictions of a big model (i.e. the teacher) and a smaller one (i.e. the student) [18], [19]. Other recent works have tried to improve the resulting compressed models. Thus, in [20] an ensemble of students learns collaboratively and teach each other during training. Distillation performance can degrade when differences in size between teacher and student models are big. To alleviate this problem a multi-step knowledge distillation can be applied [21]. In [22] a teacher-free knowledge distillation is proposed in which the student is able to learn from itself.

III. PROPOSED APPROACH

As illustrated in Fig. 1, we start by training a ConvNet for gait recognition, using as input optical flow, that will act as the master network (Sec. III-B). Then, by using the training loss function described in Sec. III-D, we train the student network (Sec. III-C), i.e. the one that has to copy the gait signatures generated by the master network, but using as input a different modality (i.e. gray). Finally, at inference time, we only use the student network to generate gait signatures.

A. INPUT DATA

1) Optical Flow

We define Optical Flow (OF) [23] as the motion pattern in a scene caused by the relative motion between an observer

![Diagram showing the flow of data and the process of GaitCopy](image-url)
and the scene between two instants of time. OF has shown excellent results in the characterization of gait [7] since it is a motion-centric representation that obviates appearance and focuses on describing a subject by a set of local and subtly varying motions.

The OF is divided into two components or channels, one representing the $y$-axis and the other representing the $x$-axis, where most of the gait motion flow is concentrated.

Let $F_t$ be an OF map at time $t$. Then, $F_t(x, y, c)$ is the value of the OF component $c$ located at coordinates $(x, y)$, containing the horizontal and vertical components of the vector.

2) Gray

In contrast to OF, gray pixels are intended to capture the subject’s appearance, keeping in mind that color is not informative for this task and that gray allows CNNs to focus more on gait information.

B. MASTER NETWORK ARCHITECTURE

We consider two possible master network architectures: a classic one [6], and a more modern one [5]. Both are described below and will be evaluated in the experimental section (Sec. IV).

1) Master-Conv2D: Classic network

This network architecture follows the one described in [6]. It is composed of the following layers: 4 Conv2D of [96, 192, 512, 4096] filters with kernel sizes $[7 \times 7, 5 \times 5, 3 \times 3, 2 \times 2]$, respectively. Then two fully-connected layers are added with 4096 and 2048 units. The final layer is a softmax with as many units as subject identities. The input is a tensor of dimensions $(50, 60, 60)$ where in the first channel alternates the $X$ and $Y$ components of an optical flow sequence of 25 frames of $60 \times 60$ pixels, forcing the person to be centered in the frame #13, as in [6]. This network is trained using a loss function that combines the Triplet loss [24] and the Cross-entropy loss with loss weights of 1.0 and 0.1, respectively.

2) Master-GaitSet: GaitSet-style network

This second type of network follows the architecture described in [5], i.e., GaitSet. This state-of-the-art network has been chosen because it is immune to permutations of the frames. So, it can use frames from different videos which have been filmed under different scenarios, viewing angles or clothing conditions. These characteristics have enabled it to be the state of the art on CASIA-B. Also, having its implementation released has allowed us to replicate its results accurately.

We have performed a small modification of it to decrease its inference time. In particular, we have changed the output of each fully-connected layer from 256 features to 128, in order to compress the model and speed up training and inference. This way, the final gait signature is reduced from 15872 to 7936 dimensions. The scheme of the architecture is shown in Figure 3.

C. STUDENT NETWORK ARCHITECTURE

Depending on the master network, we propose the use of the student architectures described below: Master-Conv2D with Student-Conv3D and Master-GaitSet with Student-GaitSet. For completeness, we summarize in Tab. 1, column ‘# of parameters’, the number of parameters (in millions) of each network used in this work.

1) Student-Conv3D: 3D convolutions

We consider the use of 3D convolutional filters (Conv3D) to define the student network in order to capture local temporal information, as the optical flow does. The sequence of the Conv3D blocks is represented in Fig. 2, which includes two versions (i.e., Student-Conv3D-A and Student-Conv3D-B) depending on the number of filters used in each layer, i.e., illustrated with different font colors inside each block. The first component of the stride (in brackets) correspond to the temporal channel. All blocks include ReLU activations, but the last one. A dropout layer (rate 0.2) is inserted before the last block, whose kernel is L2-regularized (1e-3).

2) Student-GaitSet: GaitSet-style

The corresponding student network of the Master-GaitSet is based on GaitSet as well. In this student network, due to the way gait signatures are built, the general architecture remains unchanged with respect to its master one. Only the first convolutional block is modified, adapting its convolutional kernels to deal with gray images and their single input channel. However, this change is minimal, and the architecture still matches the one shown in Figure 3. From what can be seen in Table 1, the number of parameters of the master and student networks based on GaitSet is similar.

D. LOSS FUNCTION FOR TRAINING THE STUDENT NETWORK

A student network (e.g. a gray-based one) has to learn to generate gait signatures as similar as possible to the ones generated by the master network (i.e. optical flow-based one). For teaching the student network to copy flow signatures, a regression function $L_R$ seems to be a good choice. We use as $L_R$ the Huber loss (i.e. smooth-L1) for training our

| Model                          | # of parameters | CPS  | GFLOPS |
|-------------------------------|-----------------|------|--------|
| Master-Conv2D (1024)          | 8.85M           | 172.1| 3.48   |
| Master-Conv2D (2048)          | 9.37M           | 172.3| 3.66   |
| Master-GaitSet (7936)         | 1.58M           | 642.4| 1.48   |
| Student-Conv3D-A (1024)       | 2.97M           | 272.9| 1.62   |
| Student-Conv3D-A (2048)       | 3.49M           | 273.2| 1.90   |
| Student-Conv3D-B (1024)       | 2.09M           | 272.3| 1.13   |
| Student-Conv3D-B (2048)       | 2.61M           | 272.8| 1.42   |
| Student-GaitSet (7936)        | 1.58M           | 663.7| 1.53   |

TABLE 1. Architectures comparison. Model comparison according to the number of parameters. "CPS" column shows the number of clips per second achieved by each model. 'GFLOPS' column shows the number (in billions) of floating point operations per second. All models are tested on a NVIDIA Titan Xp 12GB.
student models, as it offered better results in our preliminary experiments than the mean squared error:

$$L(y_t, y_p) = \left\{ \begin{array}{ll}
0.5 \cdot (y_t - y_p)^2, & |y_t, y_p| < \delta \\
\delta \cdot (|y_t, y_p| - 0.5 \cdot \delta), & \text{otherwise}
\end{array} \right. \quad (1)$$

where $y_t$ and $y_p$ are the true and the predicted values, respectively, and $\delta$ is a hyperparameter to be cross-validated.

In addition, as the copied signatures will probably not be exact to the ones presented by the master network, we encourage the student network to generate discriminative signatures. To achieve this second goal, we propose the use of the Triplet loss [24]:

$$L_T(A, P, N) = \max \left( \|g(A) - g(P)\|^2 - \|g(A) - g(N)\|^2 + \alpha, 0 \right), \quad (2)$$

where $g(\cdot)$ is a deep neural network, $A$ is an anchor sample, $P$ is a positive sample w.r.t. $A$, $N$ is a negative sample w.r.t. $A$, and $\alpha$ is a margin value to be cross-validated.

Then, our combined loss function $L_{MS}$ is defined as:

$$L_{MS} = L_R + \beta \cdot L_T, \quad (3)$$

where $\beta$ is a hyperparameter, to be cross-validated, that controls the contribution of $L_T$.

IV. EXPERIMENTS AND RESULTS

We start by briefly describing the datasets used in our experiments and the metrics. Then, we provide the implementation details of our models. We continue presenting the experimental setup. And the section concludes with the discussion of the results.

A. DATASETS AND METRICS

TUM-GAID [25]: It contains 305 subjects performing two walking trajectories indoors. Four situations are captured by a Microsoft Kinect: normal walk ($N$), carrying a backpack ($B$), wearing coating shoes ($S$) and, there is an elapsed time case...
where 32 subjects were recorded wearing different clothes (TN-TB-TS). We follow the train and test splits from [25]: 150 subjects for training and 155 subjects for testing. In the training set all walking conditions are used. In test, for the 155 test subjects, a gallery set is created, containing only normal walk videos. This gallery set is used to extract predictors to classify these unseen subjects. On the other hand, the probe set used to obtain the results contains all walking conditions in order to measure the robustness of the gait recognition approaches.

**CASIA-B (90°) [26]:** It consists of 124 subjects that walk indoors, captured from 11 viewpoints (0° - 180° in steps of 18°) with a resolution of 320 × 240 pixels. Three situations are considered: normal walk (nm), wearing a coat (cl), and carrying a bag (bg). Following [8], we use the first 74 subjects at training and validation, and the last 50 at test. Similarly to TUM-GAID, in the test stage two sets are created, i.e., gait and probe, with the same characteristics and purpose that have been explained above.

**Metrics.** We use Rank-1 (R1) accuracy, i.e. the percentage of correctly classified videos: $R1 = \#\text{correct}/\#\text{total}$. 

**B. IMPLEMENTATION DETAILS**

We use the Keras version of TensorFlow-2 for implementing and training our models. Regarding the master network, we have trained an OF-based master network from scratch for each dataset.

Adam is the optimizer used for training all the student networks. Each minibatch contains 150 samples with the same proportion of samples from the different scenarios (e.g. 50 N, 50 B, 50 S for TUM-GAID). We use the Batch All (BA+) triplet loss [27] to train the networks, with a batch size of $p \times k$ samples, where $p$ denotes the number of different subjects and $k$ denotes the number of training samples per subject in the batch. The learning rate starts at 0.0001 and decreases by 0.2 when the validation Mean Absolute Error (MAE) stops decreasing, with a maximum of 500 epochs. Selected by cross-validation, the parameter $\delta$ (Eq. 1) is set to 0.5 and $\alpha$ (Eq. 2) to 0.25.

**Input data.** Since each original video sequence has a different temporal length, we extract $L$-frame subsequences from the full-length sequences, with a size of 25 frames and with an overlapping of 80% between subsequences. In this way we unify the size of the input samples making sure that the temporal information of the video is not lost. In our approach, optical flow is obtained using SpyNet [28], pre-trained on MPI Sintel (more details in Sec. IV-D3).

**Data augmentation.** We randomly apply a horizontal flip (see left image in Figure 4) to the training clips to increase the amount of samples. In addition, we randomly swap the target signature of a given input (see right image in Figure 4) by choosing one of the same subject that may come from a different sequence (i.e. including different scenarios). We name this latter type of data augmentation as ‘cross-signature’, which is experimentally studied in Sec. IV-D2.

**Video sequence classification.** Note that our models receive as input a clip of 25 frames, returning the corresponding gait signature for that clip. Then, we L2-normalize the signatures and classify each individual clip by using a simple 3 Nearest Neighbor (3NN) classifier. Finally, in order to obtain a global label (i.e. subject identity) for the whole video sequence, we use a classical majority voting scheme on the individual clip labels.

**C. EXPERIMENTAL SETUP**

In order to validate our proposed approach, we train the diverse student network architectures described in Sec. III-C, and compare their performance against both their Master network and networks trained from scratch (no copy) directly on the gray modality.

In all our experiments, we use the standard data partitions (training and test) of the respective datasets, allowing a direct comparison with previous (Sec. IV-D4) and future works. Note that we follow an open-world setup for evaluating our models. That means that we always test on a set of subjects that has never been seen by the networks during training. This is more challenging that its counterpart (i.e. close-world) where a portion of the videos of the test set can be used to fine-tune the networks (e.g. to train a softmax classifier).

Also, in our experiments, we use Master-Conv2D, Student-Conv3D-A and Student-Conv3D-B for TUM-GAID, and Master-GaitSet and Student-GaitSet for CASIA-B, as master and students networks.

**D. RESULTS AND DISCUSSION**

1) Performance of the student networks

We observe in Fig. 5 that, in general terms, the student networks achieve a Rank-1 accuracy comparable to that
attained by their masters (green bars). Note that, in the CASIA-B dataset, there is a remarkable improvement in the ‘bg’ scenario and, as a consequence, in the average on the three scenarios. In our opinion, as the copies are not exact, the student networks have been able to discover identity features more robust to the ‘carrying-bag’ covariate than the master one. In fact, the same behavior with the ‘B’ scenario is observed on TUM-GAID.

Focusing on the different versions of the student networks, we notice that the smallest case ‘Student-Conv3D-B’ is even able to achieve slightly better results than their counterparts in TUM-GAID. We hypothesize that, when there is a bigger change in appearance (e.g. scenario ‘B’), a lower amount of parameters helps to reduce overfitting, i.e. to improve generalization.

In CASIA-B, we verified that our method is easy to extrapolate to other state-of-the-art models, such as GaitSet. As it can be seen in Fig. 5-right, the model learns to mimic OF-based signatures by using pixel data.

**Visual verification of the student learning.** We use the UMAP projections [29] of the signatures, to visually verify that the student networks are learning to copy the master signatures. This correct behavior can be observed in Fig. 6, that has been generated on samples from TUM-GAID. In Fig. 7 we show both the signatures obtained by a model trained from scratch on the gray modality (no copy) and the signatures obtained by a student. We notice that both types of signatures are clearly separated (i.e. ‘triangles’ do not overlap ‘circles’ of the same color). This fact reinforces our hypothesis that the student is correctly disentangling appearance from identity via motion. Finally, in Fig. 8 we represent the signatures obtained for the 16 subjects of TUM-GAID belonging to the ‘temporal’ set (where clothing was changed drastically). We observe that the student is able to generate signatures that noticeably separates the different subjects into clusters.

**Computational performance.** Tab. 1 shows the different networks together with their number of parameters (in millions), the achieved clips per second (CPS) and the number of floating point operations (in billions) per second (GFLOPS) at inference time – with a batch size of only one sample. CPS and GFLOPS have been obtained on a NVIDIA Titan Xp 12GB using the NVIDIA profiler. It can be observed that the Conv3D models are around a 58% faster than the Master-Conv2D. Also, Gaitset-based models are faster than the others models presented because they are Conv2D-based models with a lower number of parameters, which increases their CPS by approximately 300%. Regarding the GFLOPS metric, it depends on the number parameters and on the kind of layers of the model, thus it is difficult to directly compare different architectures. When
by enabling and disabling either the Triplet Loss (Eq. 3) on the student network ‘Student-Conv3D-B’, during 50 epochs, with different configurations. In particular, we have trained then, we present in Tab. 2 the performance of the model function of the number of parameters of the student network.

![FIGURE 5](image)

FIGURE 5. Signatures (training set of TUM-GAID): scratch vs student. UMAP projection of signatures learnt by two models on the same input gray modality. Same color indicates same subject identity which is also indicated by a number next to each cluster. Symbols ‘circle’ and ‘triangle’ represent a different model. Note that very different signatures are obtained by the two different models. (Best viewed in digital format)

![FIGURE 7](image)

FIGURE 7. Signatures (training set of TUM-GAID): scratch vs student. UMAP projection of signatures learnt by two models on the same input gray modality. Same color indicates same subject identity which is also indicated by a number next to each cluster. Symbols ‘circle’ and ‘triangle’ represent a different model. Note that very different signatures are obtained by the two different models. (Best viewed in digital format)

![FIGURE 8](image)

FIGURE 8. Signatures (temporal set of TUM-GAID): Student-Conv3D. UMAP projection of the signatures generated by the Student (gray) for the 16 subjects of the temporal scenario. Same color indicates same subject identity. Note that very similar signatures are obtained for the same subject, generating 16 main clusters.

comparing similar architectures that only differ on the output feature size, it is clear that, the more parameters, the more GFLOPS.

2) Ablation study

We have already studied, in the experiment summarized in Fig. 5, how the performance of the method changes as a function of the number of parameters of the student network. Then, we present in Tab. 2 the performance of the model with different configurations. In particular, we have trained the student network ‘Student-Conv3D-B’, during 50 epochs, by enabling and disabling either the Triplet Loss (Eq. 3) and/or the ‘cross-signature’ data augmentation technique (Sec. IV-B).

### 3) Optical flow study

Nowadays, there exist many optical flow approaches relying on different ways of computing this kind of low level feature. On the one hand, there are relatively old hand-crafted approaches such as [34], [35], and, on the other hand, there are new approaches based on deep learning such as [28], [36]. In this experiment, we compare two widely used optical flow methods: a hand-crafted approach (Farneback) [34] and a deep learning approach (SpyNet) [28], in order to select the

$$\beta = \{0.01, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 1, 2, 10\}; \delta = \{0.1 : 0.1 : 1\}$$
best optical flow approach for gait recognition. Fig. 9 shows the output of both approaches obtained on the same frames. Images a) and b) represent the vectors map and the magnitude of the optical flow, respectively, obtained with Farneback. On the other hand, c) and d) represent the vectors map and the magnitude of the optical flow, respectively, obtained with SpyNet. Comparing both approaches, it is clear that SpyNet produces better results in terms of sharpness and vector maps. Thus, since gait recognition must focus on subtle details of the movement, SpyNet brings an extra quality that can be decisive for gait recognition. Finally, Tab. 4 compares the execution time of both methods computed on CPU and GPU, over 25 frames of 640x480 pixels. Note that, since SpyNet is a deep learning approach, we only report execution times over 25 frames of 640x480 pixels. GPU methods are executed on a NVIDIA Titan Xp while the CPU approach is executed on a Intel Xeon CPU E5-2620 v4 @ 2.10GHz.

![Image](image.png)

**FIGURE 9. Comparison of optical flow methods.** The top row represents (a) the vectors map and (b) the magnitude of the optical flow obtained with Farneback [34]. The bottom row represents (c) the vectors map and (d) the magnitude of the optical flow obtained with SpyNet [28]. (Best viewed in digital format).

| Method               | Time per input sample (s) |
|----------------------|---------------------------|
| Farneback-CPU [34]   | 2.16                      |
| Farneback-GPU [34]   | 0.36                      |
| SpyNet [28]          | 1.94                      |

**TABLE 4. Execution time of different optical flow approaches.** Time required to compute the optical flow of an input sample with 25 frames of 640x480 pixels. GPU methods are executed on a NVIDIA Titan Xp while the CPU approach is executed on a Intel Xeon CPU E5-2620 v4 @ 2.10GHz.

4) Comparison with the state of the art

We compare here our results with previous works. On TUM-GAID (Tab. 3) we achieve state-of-the-art results with the network ‘Student-Conv3D-B’ (97% vs 96.4% [33]).

Regarding CASIA-B-90, we have only found a comparable setup in [7], where a non cross-view test setup is used. We used that setup because the goal of our approach is to produce a student model that using gray information is able to emulate the output of a master model trained with optical flow. Thus, cross-view recognition is not the goal of this work. In this experiment, the best Rank-1 (case ‘ResGray-SMP’) obtained in [7] is 96, 91, 46 and 77.7 (nm, bg, cl, Avg) while our best model achieves 98, 75, 65 and 79 (Fig. 5: Student-GaitSet). Thus, on average, our approach achieves better results and only for the bg case the results are lower than [7] but, on the other hand, our results for cl scenario clearly outperforms [7], obtaining a better average accuracy.

V. CONCLUSIONS AND FUTURE WORK

This paper has introduced a new approach named GaitCopy that is able to mimic the behavior of optical-flow-based networks for gait recognition but using gray-level pixel inputs. The student network learns to copy the gait signatures generated by the master network, but using a different input modality. We experimentally show on CASIA-B and TUM-GAID that, compared to the master networks trained on optical flow data, a similar accuracy can be obtained by the student networks trained on gray inputs, without the need of explicit optical flow computation and even using less network parameters (up to $\times 4.2$ smaller).

As future work, we plan to investigate the performance of the model on other combinations of master-student modalities, as depth maps or silhouettes, and in a multi-view setup.

REFERENCES

[1] R. Sagawa, Y. Makihara, T. Echigo, and Y. Yagi, “Matching gait image sequences in the frequency domain for tracking people at a distance,” in ACCV, 2006, pp. 141–150.
[2] “TC4 Competition and Workshop on Human Identification at a Distance,” http://hid2020.iaprc.t4.org/, 2020.
[3] C. Wan, L. Wang, and V. V. Phoha, “A survey on gait recognition,” ACM Computing Surveys (CSUR), 2019.
[4] Z. Zhang, L. Tran, X. Yin, Y. Atoum, X. Liu, J. Wan, and N. Wang, “Gait recognition via disentangled representation learning,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 4710–4719.
[5] H. Chao, K. Wang, Y. He, J. Zhang, and J. Feng, “GaitSet: Cross-view Gait Recognition through Utilizing Gait as a Deep Set,” IEEE PAMI, pp. 1–1, 2021.
[6] F. M. Castro, M. J. Marín-Jiménez, N. Guíl, S. López-Tapia, and N. P. de la Blanca, “Evaluation of CNN architectures for gait recognition based on optical flow maps,” in BIOSIG, 2017, pp. 251–258.
[7] F. M. Castro, M. J. Marín-Jiménez, N. Guíl, and N. P. de la Blanca, “Multimodal feature fusion for CNN-based gait recognition: an empirical comparison,” Neural Computing and Applications, 2020.
[8] Z. Wu, Y. Huang, L. Wang, X. Wang, and T. Tan, “A comprehensive study on cross-view gait based human identification with deep CNNs,” IEEE PAMI, vol. 39, no. 2, pp. 209–226, 2017.
[9] X. Chen, J. Weng, W. Lu, and J. Xu, “Multi-gait recognition based on attribute discovery,” IEEE PAMI, vol. 40, no. 7, pp. 1697–1710, 2018.
[10] C. Fan, Y. Peng, C. Cao, X. Liu, S. Hou, J. Chi, Y. Huang, Q. Li, and Z. He, “Gaitpart: Temporal part-based model for gait recognition,” in CVPR, 2020.

[11] P. Kumar, S. Mukherjee, R. Saini, P. Kaushik, P. P. Roy, and D. P. Dogra, “Multimodal gait recognition with inertial sensor data and video using evolutionary algorithm,” IEEE Transactions on Fuzzy Systems, vol. 27, no. 5, pp. 956–965, 2018.

[12] Z. Zhang, L. Tran, X. Yin, Y. Atoum, X. Liu, J. Wan, and N. Wang, “Gait recognition via disentangled representation learning,” in CVPR, 2019.

[13] A. Nagrani, S. Albanie, and A. Zisserman, “Learnable pins: Cross-modal embeddings for person identity,” in ECCV, 2018, pp. 71–88.

[14] H. R. V. Joze, A. Shaban, M. L. Juzzolino, and K. Koishida, “MMTM multistream transfer module for cnn fusion,” in CVPR, 2020, pp. 13289–13299.

[15] N. C. Garcia, S. A. Bargal, V. Ablavsky, P. Moreiro, V. Murino, and S. Sclaroff, “Distillation multiple choice learning for multimodal action recognition,” in Winter Conf. on Appl. of Computer Vision, 2021, pp. 2755–2764.

[16] N. Crasto, P. Weinzaepfel, K. Alahari, and C. Schmid, “MARS: Motion-Augmented RGB Stream for action recognition,” in CVPR, 2019, pp. 7882–7891.

[17] J. Stroud, D. Ross, C. Sun, J. Deng, and R. Sukthankar, “D3d: Distilled 3d networks for video action recognition,” in Winter Conf. on Appl. of Computer Vision, 2020, pp. 625–634.

[18] G. Hinton, O. Vinyals, and J. Dean, “Distilling the knowledge in a neural network,” in NIPS DLRL Workshop, 2015. [Online]. Available: http://arxiv.org/abs/1503.02531

[19] J. Ba and R. Caruana, “Do deep nets really need to be deep?” in NeurIPS, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, Eds., vol. 27, 2014.

[20] Y. Zhang, T. Xiang, T. M. Hospedales, and H. Lu, “Deep mutual learning,” in CVPR, 2018.

[21] S. H. Mirzadeh, M. Farajtabar, A. Li, and H. Ghasemzadeh, “Improved knowledge distillation via teacher assistant: Bridging the gap between student and teacher,” arXiv preprint arXiv:1902.03993, 2019.

[22] L. Yuan, F. E. Tay, G. Li, T. Wang, and J. Feng, “Revisiting knowledge distillation via label smoothing regularization,” in CVPR, 2020, pp. 3903–3911.

[23] B. K. Horn and B. G. Schunck, “Determining optical flow,” Artificial intelligence, vol. 17, no. 1-3, pp. 185–203, 1981.

[24] F. Schroff, D. Kalenichenko, and J. Philbin, “Facenet: A unified embedding for face recognition and clustering,” in CVPR, 2015.

[25] M. Hofmann, J. Geiger, S. Bachmann, B. Schuller, and G. Rigoll, “The TUM Gait from Audio, Image and Depth (GAID) database: Multimodal recognition of subjects and traits,” J. of Vis. Comm. and Image Repres., 2014.

[26] X. Lin, D. Tan, and T. Tan, “A framework for evaluating the effect of view angle, clothing and carrying condition on gait recognition,” in Proc. IJCB, 2006.

[27] A. Hermans, L. Beyer, and B. Leibe, “In defense of the triplet loss for person re-identification,” arXiv preprint arXiv:1703.07737, 2017.

[28] A. Ranjan and M. J. Black, “Optical flow estimation using a spatial pyramid network,” in CVPR, 2017, pp. 4161–4170.

[29] L. McInnes, J. Healy, and J. Melville, “UMAP: Uniform manifold approximation and projection for dimension reduction,” arXiv:1802.03426, 2018.

[30] F. M. Castro, M. J. Marin-Jimenez, N. Guil, and N. P. de la Blanca, “Automatic learning of gait signatures for people identification,” in IWANN, 2017.

[31] W. Sheng and X. Li, “Siamese denoising autoencoders for joints trajectories reconstruction and robust gait recognition,” Neurocomputing, vol. 395, pp. 86–94, 2020.

[32] M. J. Marin-Jimenez, F. M. Castro, N. Guil, F. de la Torre, and R. Medina-Camicier, “Deep multi-task learning for gait-based biometrics,” in IEEE ICIP, 2017.

[33] W. Sheng and X. Li, “Multi-task learning for gait-based identity recognition and emotion recognition using attention enhanced temporal graph convolutional network,” Pattern Recognition, vol. 114, p. 107868, 2021.

[34] G. Farnébač, “Two-frame motion estimation based on polynomial expansion,” in Scandinavian conference on Image analysis. Springer, 2003, pp. 363–370.

[35] B. D. Lucas, T. Kanade et al., “An iterative image registration technique with an application to stereo vision,” in DARPA Image Understanding Workshop. Vancouver, British Columbia, 1981.

[36] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, “Flownet 2.0: Evolution of optical flow estimation with deep networks,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 2462–2470.