Exploring Edge Computing for Gait Recognition

Israel Raul Tiñini Alvarez*, Guillermo Sahonero-Alvarez†, Carlos Menacho‡, and Josmar Suarez§
Department of Mechatronics Engineering, Universidad Católica Boliviana "San Pablo"
La Paz, Bolivia
Email: *ir.tinini@acad.ucb.edu.bo, †guillermo.sahonero@ieee.org, ‡ch.menacho@acad.ucb.edu.bo,
§lj.suarez@acad.ucb.edu.bo

***Abstract*—Gait Recognition, as a way to identify people, is remarkably attractive for scenarios in which it is not possible to rely on subjects' collaboration. Nevertheless, from all the modalities that Gait Recognition involve, vision-based approaches are better to meet hardware and settings-limitations. Because of that, in the past years, there has been several efforts on developing robust algorithms against visual gait covariates, i.e., view, clothing and carrying variations. However, besides robustness, real-world gait recognition systems also require to be implemented considering near real-time computational demands as well as portability. In this work we propose an Edge Computing approach based on the NVIDIA Jetson Nano development board and the OpenCV OAK-D camera to perform Gait Recognition. To adapt our approach, we created two small data sets that allowed our system to particularize the system to local data. Our pipeline implies the usage of a pre-trained object detection algorithm in the OAK-D, and the execution of both the representation extraction and inference on the Jetson Nano. To test our framework, we first explore its feasibility and consistency in an offline manner. Later, we characterize the complexity and time processing when executing the procedures in an online setup. Our results show that the approach is promising as it allows online operation with an inference time of 35.8 ms.**

***Index Terms*—Gait Recognition, Biometrics, Edge Computing, Real-world conditions, OAK-D**

I. Introduction

Gait recognition is conceptualized as the usage of the walking pattern, that a given individual has, for identification purposes [1]. While simple to understand, this pattern, which is called gait, is a product of a diverse unique attributes as height or age, and it is described in physiological or computational features [2].

The application of gait recognition is natural for automatic surveillance systems. As a matter of fact, there are plenty of solutions that aimed at such goal [3]. The clear advantages of these implementations rely on the non-invasiveness property of the method, as it does not require subjects to collaborate, and the robust performance at distance when using vision-based approaches [4]. Furthermore, considering the current context in which several parts of the world mandate the use of face masks, gait is a feasible biometric trait given that systems based on face recognition may be impacted [5].

While automatic and smart surveillance systems are a key concept for smart cities [6], these are also urgently required for human trafficking prevention or general security issues. For both scenarios, gait recognition supposes an attractive topic because - in each of them - subjects of interest won't provide collaboration to build the necessary representations. On the other hand, the implementation of vision-based gait recognition systems does not require high-resolution or high-quality imaging [7] and good performance is achievable through simple but efficient workflows as in [8], [9].

Despite the remarkable results that gait recognition systems show [10], these are mostly effective in lab settings, which means that there are very well-known issues that should be overcome. Issues raised by variations in clothing, views, lightning conditions, carrying and others produce significant drawbacks in subjects identification [11]. Moreover, although state-of-the-art gait recognition achieves high accuracy rates (90% or even more) in situations where training and test data are captured under similar conditions, these works typically focus on offline processing such as [12].

Gait recognition in real-time through online processing is still a challenging task mainly because of the high computational cost that supposes high-performance human pose estimation or segmentation algorithms. To face these challenges, Edge Computing has been used and shown excellent results during the inference phase of Deep Learning algorithms by leveraging the resource-constrained available hardware for vision-based solutions, which makes it appropriate to reach real-time processing [13].

In this work, we propose an scalable edge computing approach to identify a person according to his or her gait. The system, by design, works in real-time and is relatively robust to known covariates as clothing and carrying variations. The paper is structured as follows: in section II, we describe related works; in section III, we the system's structure and the experimental considerations; in section IV, we show our results and discuss their implications; finally, we conclude the paper in section V.

II. Literature Review

Deep Learning methods involve high computational costs due to the large number of mathematical computations required [14]. This makes difficult to execute these algorithms in real-time and opens the perspective to Cloud and Edge Computing, which aim at accelerating the inference processes providing support for real-time services [15]. On one hand,

978-1-6654-0810-3/21/$31.00

Cloud Computing focuses on the implementation of a powerful centralized Deep Learning infrastructure, but it has several limitations related to network communication and latency. On the other, Edge computing offers much more flexibility by using locally available resources [16]. Considering that, current advances on Computer Vision make use of these methods, previous concerns are also valid for vision-based Gait Recognition.

The concept behind Edge Computing is deeply related to both real-time response systems and portability. According to Khan et al. [17], it is defined as the extension of Cloud Computing where the services are brought closer to the end-users, i.e., at the edge of the network. In fact, this paradigm leverages several limitations which Cloud Computing already has. For example, it alleviates the computational process during the inference stage making the user experience smoother [16].

The match between Edge Computing and Gait Recognition happens when a given system must answer online. Unfortunately, most approaches dealing with that integration do not focus on visual information, but rather on accelerometer data [13], or RFID technology [18]. Actually, to the best of our knowledge, the concept of vision-based gait recognition using Edge Computing hasn't been explored. Nevertheless, there are different works in which Computer Vision procedures have been explored through Edge Computing. For example, in [19] a tracking and counting system was developed using MobileNet, and in [20] authors designed a system to assess defects on products' images employing faster R-CNN.

III. Materials and Methods

In this section, we will briefly describe the datasets used for the experiments, the proposed framework, and the techniques employed to complete each one of the stages of our system.

A. Datasets

We used three datasets for our experiments: CASIA-B [21], which is well-known gait dataset used for benchmarking gait models; UCB-Gait53, that comprises the information of 53 subjects recorded at 90, and OAK-Gait8, a new dataset built with OAK-D devices. The last two datasets, were collected at Universidad Católica Boliviana facilities and are brand new to the community. These are accessible through https://www.imt.ucb.edu.bo/cidimec/gait-recognition/.

Similar to CASIA-B, which includes data from 124 subjects through 11 views ($0 - 180$, with separations of 18) and 10 sessions (6 normal or "nm", 2 carrying-bag or "bg", and 2 wearing-coat or "cl"), UCB-Gait53 (UCB dataset) follows the same convention but with only one view (90), 53 subjects and RGB recordings at 30FPS in 1080p. Unfortunately, we only could access the uncorrelated videos and binary masks from CASIA. Finally, OAK-Gait8 (OAK dataset) was collected with five OAK-D cameras, and it contains the information of 8 subjects recorded at 30FPS in 1080p, and walking directions of 60, 75, 90, 105, and 120. Moreover, due to the RGB-D nature of these cameras, we were able to capture three records per each OAK-D, making this dataset suitable for RGB-D processing. Similar to CASIA, OAK-Gait8 has 6 normal, 4 carrying-bag, 4 and wearing-coat walking sequences. Since our experiments mainly focus on exploring the edge of computing, all the tests have been carried out using side view. Fig. 1. shows some sample images from each dataset in all the conditions.

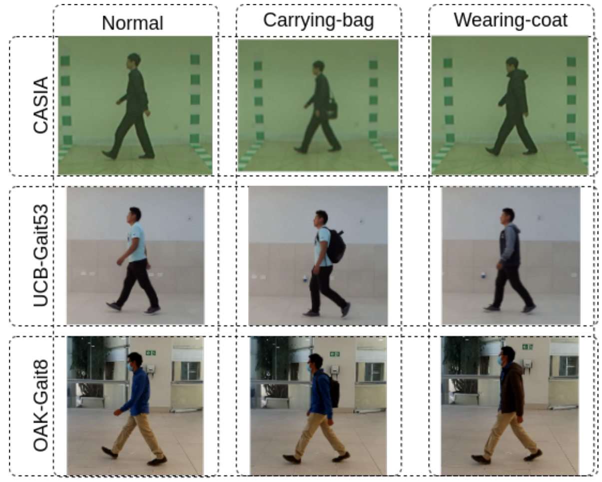

Fig. 1. Sample images from the CASIA (first row), UCB-Gait53 (second row), and OAK-Gait8 (last row) datasets

B. Framework

Our framework is based on [22], however, we included an initial step of detection and segmentation so that we can use video recordings instead of pre-segmented masks. As for the representation, feature extraction and selection, and classification, we kept the same as [22].

1) Detection: The input data in our model is composed of frames belonging to video sequences, yet live video can also be used. Hence, the first step consists on detecting the objects of interest, that in our case are people. To achieve this, we used the default model of MobileNetV2 provided by OpenCV [23].

2) Segmentation: Once the position of the subject in the scene has been determined, if any, it is necessary to obtain its silhouette. As a first step, we trained a custom segmentation model based on a U-Net and the UCB dataset given that the CASIA-B dataset will be used to compute the performance of our model. However, generating the ground truth for a segmentation task is a tedious and costly process. Thus, we computed the binary masks corresponding to the subjects silhouettes using background subtraction followed by correction steps such as filtering and morphological operations. After this step, $\sim$ 30M pairs of image-mask were collected to train the model.

We adapted U-Net from [24]. Since large images are not needed and a small model is aimed to run an edge device. We noticed that the skip connections helped us to relieve the bottleneck problem and increased the recognition accuracy.

3) Later stages: We used the modified GEI (MGEI) spatio-temporal gait-representation [22], which is produced by averaging the silhouettes extracted over a complete gait cycle without considering the chest and hips. The MGEI keeps the GEI robustness to noise and efficient computation, but it is ideal for side-view scenarios. Given that MGEI are high dimensional representations, we used PCA for feature extraction and selection to keep minimum redundancy and improve the performance of our classifier, hence, speed up the computation. Finally, to simplify computations, we used an LDA classifier since this linear discriminant combined with PCA obtains a good performance in side-view configuration [9].

4) System workflow: The whole system can be divided into two main stages: offline and online processing, as seen in Fig. 2. It may seem pretty straightforward, but the interaction between the different components differ in each stage.

In the offline stage. First, we use the UCB dataset to generate the training data for learning the segmentation model through a simple background subtraction step. Then, these pairs of image-masks generated are used to train the U-Net, which will be used later during the online processing. As next step, we perform the representation of the gait cycles of all the subjects in CASIA-B. These representations along with the target IDs are used to train an LDA classifier, which again will be used later. All this process has been executed in a CPU.

In the online processing stage, the input to our system are RGB video recordings, though it is also possible to use live video. The first stage correspond to detection, which is done by the pre-trained MobileNetV2. The localization of the subject in the scene is used to crop that region and infer the silhouette using the pretrained U-Net. These binary masks are stacked and used to compute the representation, which is finally fed to the pretrained classifier to infer the ID of the subject who is walking in the video.

IV. Results and Discussion

A. Training the segmentation model

For training the segmentation model, we trained multiple models with different architectures, taking into account the architecture recommendations from [25] and [24].

1) Training details: The U-Net can be divided into an encoder and a decoder, both use convolutional modules, instance normalization, and Leaky activation. To denote theses, Ck will represent a module composed of convolution-normalization-Leaky with k filters. Whereas CDk states for a module of convolution-normalization-dropout-Leaky activation with k layers and a dropout of 0.5. In the encoder (E), the convolutions downsampled the feature maps by a factor of 2, and the decoder (D) upsampled them by the same factor. In total, we trained three networks with the next structure:

- Model A: `E:C16-C32-C64-C128; D:CD128-C64-C32`
- Model B: `E:C16-C32-C64-C128-C128; D:CD256-CD128-C64-C32`
- Model C: `E:C32-C64-C128-C128-C128; D:CD256-CD256-C128-C64`

TABLE I
SEGMENTATION RESULTS

Model	Parameters	Training		Test accuracy	
		acc	loss	acc	loss
A	∼220K	0.9451	0.0845	0.9457	0.0826
B	∼590K	0.9525	0.0641	0.9522	0.0649
C	∼1.02M	0.9560	0.0516	0.9558	0.0523

TABLE II
IDENTIFICATION RUNNING ON EDGE AND OFFLINE COMPARISON

Model	Inference (ms)	CCR nm	CCR bg	CCR cl	Power (mW)
[26]	-	0.95	0.88	0.76	-
[27]	-	0.96	0.65	0.43	-
Ours	-	0.98	0.84	0.65	-
A	32.5	0.70	0.52	0.50	3324
B	34.7	0.50	0.40	0.43	3406
C	35.8	0.60	0.52	0.5	3449

We used Adam to train the models. We noticed that good performance was achieved after 10 epochs plus 5 more using a learning rate of 2E-2 and 2E-5 respectively, in addition to momentum parameters of $\beta1 = 0.5$, $\beta2 = 0.999$. The training process was implemented in Python using Keras and TensorFlow in Google Colab. Thus, the experiments run in a 2.3 GHz dual-core processor with 25 GB of RAM, and an NVIDIA Tesla K80 graphic card with 12 GB of memory, and 2496 CUDA cores.

2) Results: The table I shows the training and testing results of each model along with their amount of parameters. As it can be seen, the bigger the model, the higher the segmentation accuracy and the lower the loss.

B. Identification

Table II shows the recognition accuracy on the CASIA-B dataset. The first two rows correspond to the reported values from other works which follow the same experimental setup as us. The third row corresponds to our model during the offline train-testing using the precomputed binary silhouettes. The last rows show the results obtained when processing online the video recordings using the pretrained models for segmentation. We also estimated the energy consumption by the Jetson Nano, so we can also understand how models behave. The results point to a very similar behavior between them, which suggests that the corresponding computational load is alike. This is an expected result, given that the models are not being executed in the Jetson Nano.

As shown in Table II, appearance variations such as clothing or charging change drastically affects the recognition performance. Although the CCR in normal conditions it considerably high in [26], [27] and in our offline model, in real-world scenarios, clothing changes are impossible to avoid.

V. Conclusions

Gait Recognition is an attractive topic for developing smart surveillance systems. However, its deployment on real-world condition depends on the development of edge computing

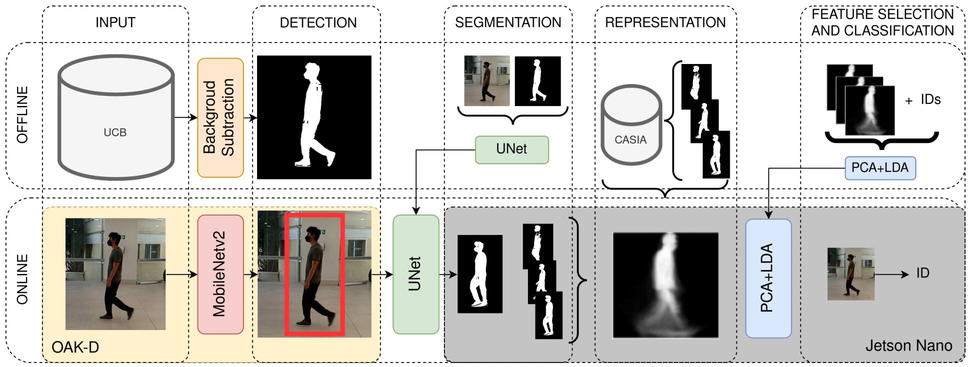

Fig. 2. System workflow. The image allows to differentiate between the offline and online stages that were followed.

solutions for the task. In this work, we proposed an edge computing approach for gait recognition using as a framework the combination of the Jetson Nano and the OpenCV's OAK-D. Our results show that the deployed models on the OAK-D, and the implemented algorithms on the Jetson Nano make the framework suitable for Edge Computing based Gait Recognition. In addition, we contributed with two new Gait Recognition data sets which will be available to the community. Finally, this work also provides the background to start developing more intelligent approaches for real-time processing, e.g., better spatio-temporal sampling methods.

ACKNOWLEDGMENT

This research was supported by a grant provided by the OpenCV AI Competition 2021 and a small research projects grant from the Universidad Católica Boliviana "San Pablo".

REFERENCES

[1] T. K. Lee, M. Belkhatir, and S. Sanei, "A comprehensive review of past and present vision-based techniques for gait recognition," *Multimedia tools and applications*, vol. 72, no. 3, pp. 2833–2869, 2014.

[2] Z. Zhang, M. Hu, and Y. Wang, "A survey of advances in biometric gait recognition," in *Chinese conference on biometric recognition*. Springer, 2011, pp. 150–158.

[3] I. Bouchrika, "A survey of using biometrics for smart visual surveillance: Gait recognition," in *Surveillance in Action*. Springer, 2018, pp. 3–23.

[4] J. P. Singh, S. Jain, S. Arora, and U. P. Singh, "A survey of behavioral biometric gait recognition: current success and future perspectives," *Archives of Computational Methods in Engineering*, pp. 1–42, 2019.

[5] M. Gomez-Barrero, P. Drozdowski, C. Rathgeb, J. Patino, M. Todisco, A. Nautsch, N. Damer, J. Priesnitz, N. Evans, and C. Busch, "Biometrics in the era of covid-19: Challenges and opportunities," 2021.

[6] M. Kashef, A. Visvizi, and O. Troisi, "Smart city as a smart service system: Human-computer interaction and smart city surveillance systems," *Computers in Human Behavior*, p. 106923, 2021.

[7] M. Kumar, N. Singh, R. Kumar, S. Goel, and K. Kumar, "Gait Recognition Based on Vision Systems: A Systematic Survey," *Journal of Visual Communication and Image Representation*, 2021.

[8] V. C. Lima, V. H. Melo, and W. R. Schwartz, "Simple and efficient pose-based gait recognition method for challenging environments," *Pattern Analysis and Applications*, vol. 24, no. 2, pp. 497–507, 2021.

[9] I. R. Tiñini Alvarez and G. Sahonero-Alvarez, "Cross-view gait recognition based on u-net," in *2020 International Joint Conference on Neural Networks (IJCNN)*, 2020, pp. 1–7.

[10] X. Li, Y. Makihara, C. Xu, Y. Yagi, S. Yu, and M. Ren, "End-to-end model-based gait recognition," in *Proceedings of the Asian conference on computer vision*, 2020.

[11] M. Sivarathinabala, S. Abirami, and R. Baskaran, "A study on security and surveillance system using gait recognition," in *Intelligent techniques in signal processing for multimedia security*. Springer, 2017.

[12] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 14 225–14 233.

[13] X. Zeng, X. Zhang, S. Yang, Z. Shi, and C. Chi, "Gait-based implicit authentication using edge computing and deep learning for mobile devices," *Sensors*, vol. 21, no. 13, p. 4592, 2021.

[14] J. Chen and X. Ran, "Deep learning with edge computing: A review." *Proc. IEEE*, vol. 107, no. 8, pp. 1655–1674, 2019.

[15] X. Wang, Y. Han, V. C. Leung, D. Niyato, X. Yan, and X. Chen, "Convergence of edge computing and deep learning: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, 2020.

[16] M. Véstias, "Processing systems for deep learning inference on edge devices," in *Convergence of Artificial Intelligence and the Internet of Things*. Springer, 2020, pp. 213–240.

[17] W. Z. Khan, E. Ahmed, S. Hakak, I. Yaqoob, and A. Ahmed, "Edge computing: A survey," *Future Generation Computer Systems*, vol. 97, pp. 219–235, 2019.

[18] C. Luo, J. Wu, J. Li, J. Wang, W. Xu, Z. Ming, B. Wei, W. Li, and A. Y. Zomaya, "Gait recognition as a service for unobtrusive user identification in smart spaces," *ACM Transactions on Internet of Things*, vol. 1, no. 1, pp. 1–21, 2020.

[19] A. C. Cob-Parro, C. Losada-Gutiérrez, M. Marrón-Romera, A. Gardel-Vicente, and I. Bravo-Muñoz, "Smart video surveillance system based on edge computing," *Sensors*, vol. 21, no. 9, p. 2958, 2021.

[20] Y. Wang, M. Liu, P. Zheng, H. Yang, and J. Zou, "A smart surface inspection system using faster r-cnn in cloud-edge computing environment," *Advanced Engineering Informatics*, vol. 43, p. 101037, 2020.

[21] S. Yu, D. Tan, and T. Tan, "A framework for evaluating the effect of view angle, clothing and carrying condition on gait recognition," *Proceedings - International Conference on Pattern Recognition*, vol. 4, 2006.

[22] I. R. Tiñini Alvarez and G. Sahonero-Alvarez, "Gait recognition based on modified gait energy image," in *2018 IEEE Sciences and Humanities International Research Conference (SHIRCON)*, 2018, pp. 1–4.

[23] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 4510–4520, 2018.

[24] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," *IEEE Access*, 2015. [Online]. Available: http://arxiv.org/abs/1505.04597

[25] P. Isola, J. Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-January, pp. 5967–5976, 2017.

[26] M. Alotaibi and A. Mahmood, "Improved gait recognition based on specialized deep convolutional neural network," *Computer Vision and Image Understanding*, vol. 164, pp. 103–110, 2017.

[27] S. Yu, H. Chen, Q. Wang, L. Shen, and Y. Huang, "Invariant feature extraction for gait recognition using only one uniform model," *Neurocomputing*, vol. 239, pp. 81–93, 2017.