Recognizing Human Gait Signatures with GOTCHA

Brenner Marvin
Audiovisual Media (AM3)
Stuttgart Media University
Stuttgart, Germany
Marvin.Brenner@web.de

***Abstract*—This paper explores the possibility of identifying intruders based on their individual way of moving through inside or outside environments. Their gait, mathematically represented by the spatio-temporal relation of 18 skeleton joints like neck, elbows and ankles, is estimated by the 2D Pose Estimator by Cao, Simon, Wei, S. & Sheikh. Then it is broke down into an array of essential features, enabling a three-dimensional Convolutional Neural Network to classify each person by their movement. This reidentification combined with the ability to differentiate between different observed locations might be a step to further advance security analytics for home surveillance camera setups and public surveillance purposes.**

***Index Terms*—gait, identification, cnn, security-analytics**

I. Introduction

Since the recent massive leaps forward of Object Identification by trained Neural Networks, it has also become possible to surpass human level performance of re-identifying a person just by their looks[1]. However in security analytics, for example when surveilling an airport or the garden of a private home, we must take it one step further and also recognize people that may try to hide their identity. Criminals often wear masks and try to change up their appearance as much as possible making it increasingly diffcult to catch them afterwards. One way to recognize a human still is to recognize the individual movement or "gait" of a person. The gait of a person is determined by an individuals weight, body size, footwear, limb length and posture as well as their characteristic structure of muscles and bones. Therefore it can be used as a biometric to recognize humans even when they completely cover or change their appearance, as long as they don't consciously change their way of movement as well. However even changing your movement is limited by factors like your physical characteristics and could be taken into account later in more advanced models. Until now, gait analysis had to be performed mainly by observing experts in areas like sports or medicine, for example for rehabilitation from injury. With the current advancement in technology, computers can help making gait analysis more reliable to identify nuances that even well-trained eyes of professionals might miss and support them in decision-making [2]. Although many researches have expressed their concerns when it comes to machine learning supported surveillance, the computer vision field has progressed in a remarkable fashion. Especially government's intelligent agencies and for example airports are highly interested in biometrics being used to identify criminal individuals in order to prevent crime [3]. In comparison to wearable devices for gait analysis like in clinical applications [4], their use cases prioritize algorithms that are able to analyse as much as possible from simple camera recordings. Recently even smartphones cameras have been proven to provide enough image information for such analysis[5].

II. Related Work

As seen in [6] , one approach to quantify the gait of a person is to take the following 5 steps: 1) Calculating an average image of a gait cycle and dividing the human body into multiple areas 2) Extracting affine movement invariants at each area as gait features 3) Calculating an average image of a subject and dividing the human body into multiple areas as well 4) a matching weight at each area is estimated according to similarity between features of each subject and the database 5) The Subject is classified by weighted integration of similiarity over all areas. This approach can be transferred easily to the concept of Training and Testing Data in Machine Learning. With the help of the 2D Pose Estimation Algorithm of [7], the coordinates of 18 skeleton joints in each image can be extracted directly from images. The Data then can be reshaped to set each sample from a frame into temporal relation to the other frames in order to quantify gait. Additionally some more feature vectors can be defined, depending on use case, and added to the matrix. For the architecture of the neural network, models like LSTMs and three-dimensional CNNs might offer a reliably classification of persons by going through XY-Positions in a time series. While both network types provide reasonable accuracy, in [8] a combination of multiple CNNs as well as an SVM have outperformed most other concepts when it comes to learning spatiotemporal features. Providing depth information with RGB to extract skeletons as in case of the kinect sensor[9] has also proven to further improve classification accuracy of gait recognition.

III. Approach

The approach of this paper was to first extract consecutive frames from a surveillance video, then perform a Pose Estimation on each frame to get the coordinates of the skeleton joints of each subject over time. The resulting dataframe should therefore represent the temporal-spatial relation of each person's gait. Additionally it can contain other features like size of their steps to enable a neural network to reliably identify them by just their movement. Each of the 62 subjects

in the Gotcha DB should be identified by their gait and additionally the network should classify if they are walking inside or outside of buildings and whether or not they are lit by a flashlight. The accuracy of recognition should have priority over optimization for achieving real-time calculation speed for now.

A. The Gotcha DB

"GOTCHA database is used for scientific research of biometrics and soft biometric recognition in-the-wild. It is collected by Biometric Image Processing Lab (BIPLab) of University of Salerno in collaboration with Instituto Universitario de Sistemas Inteligentes y Aplicaciones Numéricas en Ingeniería (SIANI) of University of Las Palmas De Gran Canaria" [10]

The Dataset consists of 12 Folders, the first 6 folders consist of videos of 62 different subjects that have been recorded walking inside of the university, outside on campus and inside the university while it's dark and they are being lit by a flashlight. The folders also differentiate between subjects looking towards the camera (cooperative) and a version where each person tries to avoid looking into the camera(non-cooperative). The 7th folder contains 180° Shots of the faces of all 62 subjects while the folders 8 to 11 have more walking footage of the subjects outside on stairs holding different objects in their hands (for example a bag) and another one showing them walking on a curved outdoor path. Additionally folder 12 will soon features some 3D Scans of all 62 subjects as well as containing some body and face landmark of each person. In near future when the DB is complete, it might provide a good amount of data to train neural networks on different aspects of security analytics such as facial recognition , prediction of behaviour or recognition of tools associated with criminal activity.

B. Preprocessing

First of all, the 372 videos (62 subjects x 6 Variations) of the first 6 folders of the Gotcha DB need to be loaded into the notebook. Since all videos are around 6 seconds long and the users start walking after roughly 2 seconds, the core movement of every users gait is found roughly after one third in the video. Therefore at this point in the video, extraction of consecutive frames starts which is enough for each person to take roughly one step depending on their walking speed.

In the next step all the extracted frames get resized from 1920x1080 to 192x108 while interpolation is set to bicubic. That way the dimensions for the 2D Pose Estimation are greatly reduced but you can still see all necessary details in the images in order to identify body parts. Because a lot of the videos were shot in vertical mode, they also get rotated into the correct position. While testing, a great drop in performance of the 2d pose estimation algorithm was noticed, when the frames are rotated in a wrong direction, p.e upside down. Usually the Pose Estimator missed around 40% of coordinates of almost all joints, when the image wasn't in correct rotation.

C. 2D Pose Estimation

In order to perform the 2D Pose Estimation Algorithm by Cao, Simon, Wei and Sheikh from 2017, the image demo of their papers's github[12] is used inside a function to calculate all the X and Y coordinates of 18 skeleton joints for every extracted frame. The coordinates are saved in an 18 dimensional array with elements which describe the joints of the following order: Nose, neck, right shoulder, right elbow, right wrist, left shoulder, left elbow, left wrist, right hip, right knee, right ankle, left hip, left knee, left ankle, left eye, right eye, left ear, right ear. For each frame the coordinate array and an image of the frame with a skeleton drawn on top get saved. This can help identifying what caused joints to be missed, for example frequently the subject has already closed in to the camera to a degree where their ankles aren't visible any longer. After recollection of all joint data files, the first dataframe is constructed. For each sample it should contain a label, the filename it was extracted from (including the number of the frame to indicate temporal relation to other frames) and 18 tuples of joint coordinates.

D. Structure of Dataframe

Analysis of the Data illustrated that apart from the joints for the ankles and the ears, the pose estimator always extracted more than 78% of coordinates per joint in each frame. The 2D Pose Estimator struggled to detect the ears of subjects with long hair, however these can be neglected for gait analysis.The missing joints of the ankles were due to the fact that all videos were shot in american framing, where each person was only visible from knee-height upwards, it is assumed that the missing ankles will reduce overall accuracy of gait recognition. The data columns of the ears and ankles therefore got removed in order to provide a more even distribution of balanced training data. The coordinates were split in 2 columns for X and Y of each joint to observe their changes individually. At this point the data is an accurate representation of joint positions for each frame but to set them in temporal relation, they had to be grouped for every video again. Identification of the 62 subjects per frame just based on the positions of their in the Gotcha DB was achieved with 97% accuracy on testing data, however this result was only achieved on a small amount of training data (11.744 frames of 372 videos) and not robust to different camera angles than from the training data. The difference in coordinates from one timestep to another needed to be calculated because the network shouldn't recognize subjects based on their body size from only one camera angle by their individual movement.

Some videos were later observed to feature subjects with black tshirts in front of black backgrounds and therefore lead to a high loss in joints for the 2d pose estimation algorithm. Therefore some specific samples causing these problems were excluded from training.

E. Architecture of the Neural Network

The Data-Array of 10000 samples with 30 features each was split with scikit-learn, whereas 90% of data was used for

training and 10% for testing. The Labels were declared to be the person's ID resulting in an target array of length 62 , each class being one of the 62 subjects. The Training Array X for the input was scaled with a standard scaler transformation into a value range of 0 to 1 because the coordinates of the joints could vary strongly depending on camera angles and height of each person. After testing, the data was finally fed into a test neural network with the following architecture: The first Dense layer consisted of an LSTM with 512 units, activated with a tanh function. The Output was then passed onto a identical second LSTM with only 256 units. Followed by 2 hidden layers with 128 and then 64 Nodes plus tanh activation. The final layer had 62 nodes, one for each class and used the softmax activation. Between the layers, each output was normalized with a BatchNormalization Layer. The loss of the network was because of the multiclass classification problem calculated with categorical crossentropy while being optimized with stochastic gradient descent with a learning rate of 0.0001, a lr-decay of 0.000001 and a momentum of 0.9. The Network trained for 400 epochs with a batchsize of 16 samples. In a second step the C3D Network consisting of multiple threedimensional convolutional networks to set the coordinates in temporal relation was tested. However the architecture was too high in complexity for just a coordinate classification problem.

IV. Results

TABLE I
GAIT RECOGNITION - TRAINING RESULTS

Neural Network/ Layers of the Model	Experiments		
	Epochs	*Val acc*	*Parameters*
L64 x D32 x D62[1]	100	40%	lr = 0.001
L256-L128-D32-L62-D62	100	52,4%	lr = 0.0001
L512-L256-D128-D64-D62	250	68,15%	lr = 0.0001

[1]L = LSTM, D = Dense Layer, the number refers to the number of units.

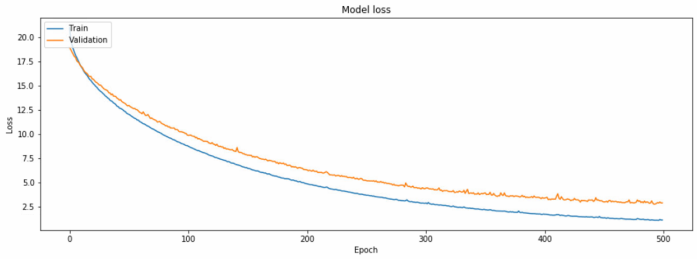

Fig. 1. Gait Recognition LSTM Model loss of the final configuration

A. Performance

The final iteration managed to achieve 68.15% accuracy on recognition of the 62 subjects in 6 different environments. After around 400 epochs the network wasn't able to further increase neither training nor validation accuracy.

The small network who learned to distinguish the 3 different environment conditions (outdoor/indoor/indoor with flashlight) , by looking at brightness histograms of frames, already managed to classify 97% of the environments after only 6 epochs of training.

B. Limitations

Apart from the limited timeframe of 2 months, some limitations come from the dataset itself, the performance of the 2d pose estimation algorithm and the structure of the dataframe. The Dataset is relatively small for a 62 person classification problem, containing only 6 videos for each subject. The missing joints of ankles and knees in many videos make it additionally difficult to define important features for gait recognition for example the step length and velocity of each subject. In big number of cases the 2d pose estimation algorithm still fails to locate the joints correctly. This is often the case when videos of the dataset features subjects in tshirt with the same color like the background or wear wide jackets covering their body shape. Taking just a small number of frames of each video however might result in insufficient data to distinguish 62 classes. Additional features apart from the body joint coordinates like for example step length might be necessary to reliably differentiate the gait of the 62 subjects.

V. Conclusion

Person Identification by Gait is a potentially very useful method of distinguishing subjects that have changed or hidden their appearance in the surveillance sector. As our results on the Gotch Dataset have shown, the method does however require a large amount of training data in order for the algorithm to learn from the very subtle differences in movement that persons of similar body posture, height and weight have. The data would have to be collected from every subject beforehand and it is important that the subjects are clearly visibile without noise in the background in order to capture their movement reliably when using 2d pose estimation. When trained with the first 6 folders of the Gotcha Dataset, the model had trouble distinguishing some subjects more than others, especially when their bodies provided little contrast to the background of their environment.This has massively decreased the overall accuracy of the model. However for some subjects with better visibility, the model never failed to recognize their gait. All in all, with additional finetuning of the model and samples of training data with less noise, persons can be recognized just by their gait and LSTMs have proven capability to recognize the sequences of their joint coordinates. Future research could aim to further improve accuracy with more training data and also develop more robust models for different camera angles, wide clothing and recognizing multiple subjects at the same time.

Acknowledgment

I'd like to thank Professor Michele Nappi, Carmen Bisognin, Paola Barra and all students of the C.A.S.A course.

REFERENCES

[1] G. Eason, B. Noble, and I. N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955.

[2] Zhang, X., Luo, H., Fan, X., Xiang, W., Sun, Y., Xiao, Q., ... & Sun, J. (2017). Aligned Re-ID: Surpassing human-level performance in person re-identification. arXiv preprint arXiv:1711.08184.

[3] Seidenberg, P. H., & Beutler, A. I. (2008). The sports medicine resource manual. Elsevier.

[4] Bouchrika, I. (2018). A survey of using biometrics for smart visual surveillance: Gait recognition. In Surveillance in Action (pp. 3-23). Springer, Cham.

[5] Muro-De-La-Herran, A., Garcia-Zapirain, B., & Mendez-Zorrilla, A. (2014). Gait analysis methods: An overview of wearable and non-wearable systems, highlighting clinical applications. Sensors, 14(2), 3362-3394.

[6] Zou, Q., Wang, Y., Zhao, Y., Wang, Q., Shen, C., & Li, Q. (2018). Deep Learning Based Gait Recognition Using Smartphones in the Wild. arXiv preprint arXiv:1811.00338.

[7] Iwashita, Y., Uchino, K., & Kurazume, R. (2013). Gait-based person identification robust to changes in appearance. Sensors, 13(6), 7884-7901.

[8] Cao, Z., Simon, T., Wei, S. E., & Sheikh, Y. (2017). Realtime multi-person 2d pose estimation using part affinity fields. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 7291-7299).

[9] Tran, D., Bourdev, L., Fergus, R., Torresani, L., & Paluri, M. (2015). Learning spatiotemporal features with 3d convolutional networks. In Proceedings of the IEEE international conference on computer vision (pp. 4489-4497).

[10] Prathap, C., & Sakkara, S. (2015, August). Gait Recognition using skeleton data. In 2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI) (pp. 2302-2306). IEEE.

[11] Github Repository, https://gotchaproject.github.io/ (last checked on 12/01/2020)

[12] Github Repository, https://github.com/michalfaber/keras_Realtime_Multi-Person_Pose_Estimation_ (last checked on 12/01/2020)