School of Engineering and Digital Science

Gait Recognition Using Convolutional Neural Network

Aziza Duisembay
Electrical and Computer Engineering
Nazarbayev University
Nur-Sultan

Supervisor: Grant Ellis

29-04-2020

Outline

- Introduction
- Methodology
  - Convolutional Neural Network (CNN)
  - Layer-Wise Relevance Propagation (LRP)
- Simulations & Results
- Conclusions
- Future Work
- References

Introduction [1/4]

Why gait recognition is a popular person identification approach:

- It does not require any cooperation from subjects;
- It works fine even at a large distance (usually 10m or more);
- Video resolution does not affect a network's performance dramatically;
- Walking style of a person can hardly be imitated.

Problem statement:

✓ Development of a robust neural network for camera-based gait recognition.

Project objectives:

✓ Stable & robust performance of the network insensitive to a clothing and view covariate;

✓ High accuracy rate of recognition.

Introduction [2/4]

Literature review

Research	Classification data	Classification method	Gait dataset	Accuracy achieved	Year
Gait Recognition Based on 3D Skeleton Joints Captured by Kinect, by Y. Wang et al. [1]	Horizontal & vertical distance features of skeleton model	k-Nearest Neighbors (k-NN) algorithm	20 subjects captured from a single view point	92%	2016
Multi-gait Recognition Using Hypergraph Partition, by X. Chen et al. [2]	3D tensor gait features	k-NN	120 subjects	Frontal view: 80.3% Lateral view: 89.2%	2017
Pose-based deep gait recognition, by A. Sokolova et al. [3]	Maps of optical flows	Pre-trained neural networks (VGG-19 & Wide ResNet) for feature extraction & k-NN for classification	CASIA B	92.95%	2019
Joint Intensity Transformer Network for Gait Recognition Robust Against Clothing and Carrying Status, by X. Li [4]	GEIs	Unified joint intensity transformer network (JITN)	OUTD-B	85.9%	2019

Introduction [3/4]

Literature review contd.

Research	Classification data	Classification method	Gait dataset	Accuracy achieved	Year
Multi-Task GANs for View-Specific Feature Learning in Gait Recognition, by Y. He et al. [5]	Period Energy Image (PEI)	Multi-Task Generative Adversarial Network (MGAN)	CASIA B, OU-ISIR, HumanID	74.6%, 93.2%, and 94.7% correspondingly	2019
Gender Recognition via Fused Silhouette Features Based on Visual Sensors, by S. Bei et al. [6]	Gait Energy Image (GEI) & Sub-GEI	Two-stream CNN model	CASIA B	Avg. 86% (different for each view angle in the range of 18-172°)	2019
Multi-perspective gait recognition based on classifier fusion, by X. Wang et al. [7]	Dynamic gait features & GEIs	Fusion of SVM classifier & a Hidden Markov model (HMM)	OU-ISIR	96.2%	2019
Cross-View Gait Recognition by Discriminative Feature Learning, by Y. Zhang [8]	Silhouette image	CNN & long short-term memory (LSTM) attention model	CASIA B	86.5%	2020

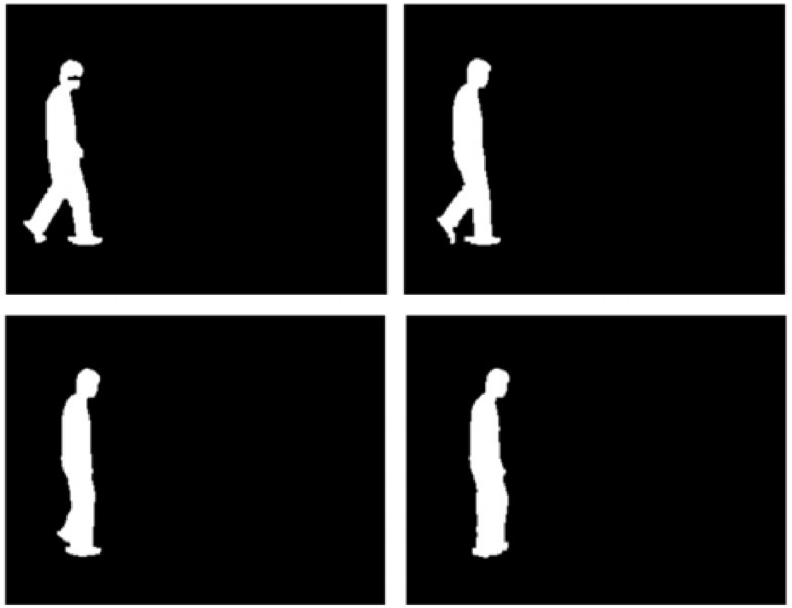
Methodology [1/10]

Gait Dataset

Comparison of popular gait datasets

Name	# of subjects	Covariates	View points
CASIA B [9]	124	Clothes & carryings	11
OU-ISIR [10]	4007	-	2
OUTD-B [10]	20	Clothes	1
TUM [11]	305	Shoes & carryings	1
USF HumanID [12]	122	Clothes & carryings	2

The dataset used in this project is CASIA B.
It has videos with extracted silhouettes on a black background.
4 corrupted classes were removed, so that 120 remain. The gait data is gathered from 85 men and 35 women.



Images from CASIA B dataset

Methodology [2/10]

Gait Energy Images (GEI)

Storing & processing video frames:

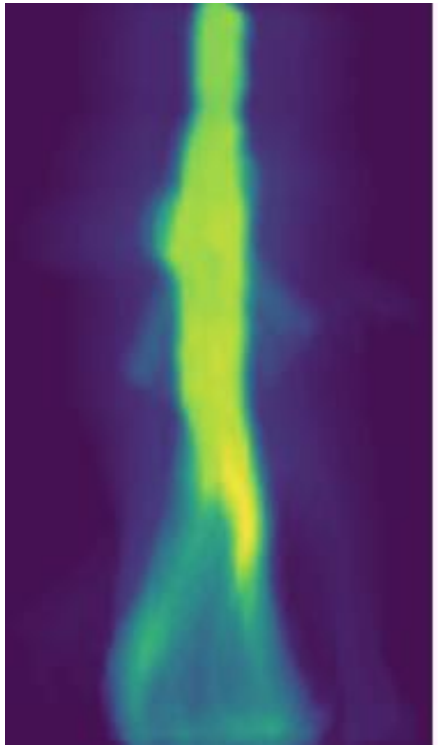
- Takes a lot of memory space;
- Is computationally expensive.

Solution:

GEI: a mean vector of all normalized frames belonging to one gait cycle:

$$G = \frac{1}{N} \sum_{n=0}^{N} I_n(x, y)$$

Where $I_n(x, y)$ is a particular pixel at position (x, y) of frame n, where n = 0,1, . . . , N, and N is a number of frames of one gait cycle.



Gait energy image

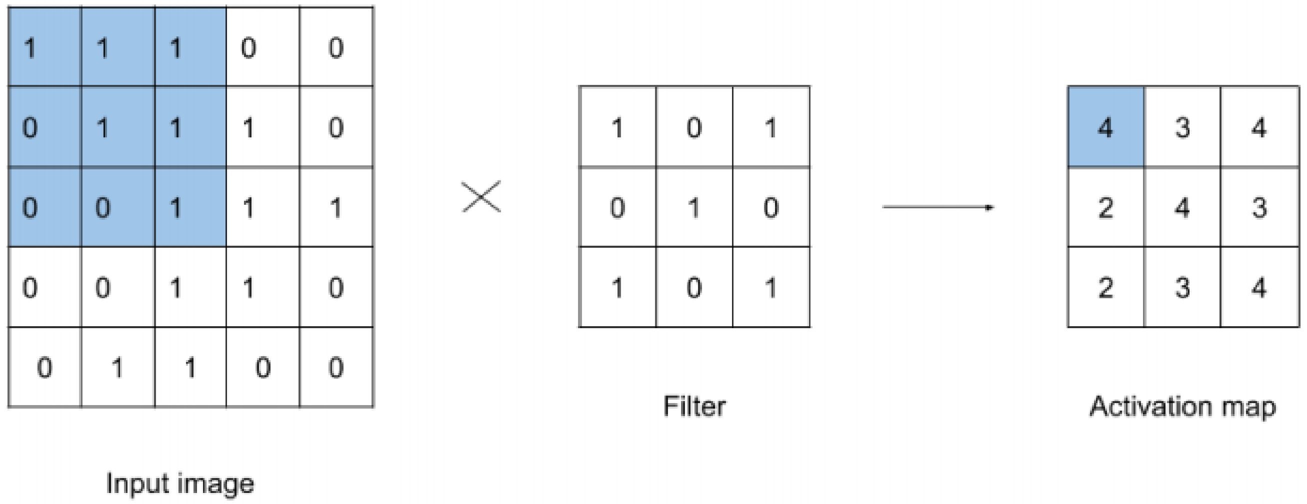
Methodology [3/10]

What is convolution?

In deep learning, the convolution's aim is to extract distinctive features of the input image.

A convolutional layer has several filters. i.e. 2D matrices, that slide/convolve around the input matrix.

As the filter slides over the input, it computes an activation map that represents the filter's reactions at every position of the input volume.



Convolution operation

Methodology [4/10]

1. Convolutional Layers

The proposed CNN has 4 convolutional layers with the following parameters:

Parameter	Description	Set Value
Num. of filters	Number of filters in each layer	16, 32, 64, 124 correspondingly
Kernel size	Filter’s window size (width and height respectively)	4x4
Stride	Step size at which a filter slides an input	2
Zero padding	Allows to control the width and height of the output volume by adding zero pixels on the borders of the input.	“same”: it adds so many zero pixels to the output volume that it will have the same size as the input
Activation function	Performs a non-linear transformation of the layer’s output. All operations in layers are linear, which does not let the network learn complex patterns of the input data.	Tanh

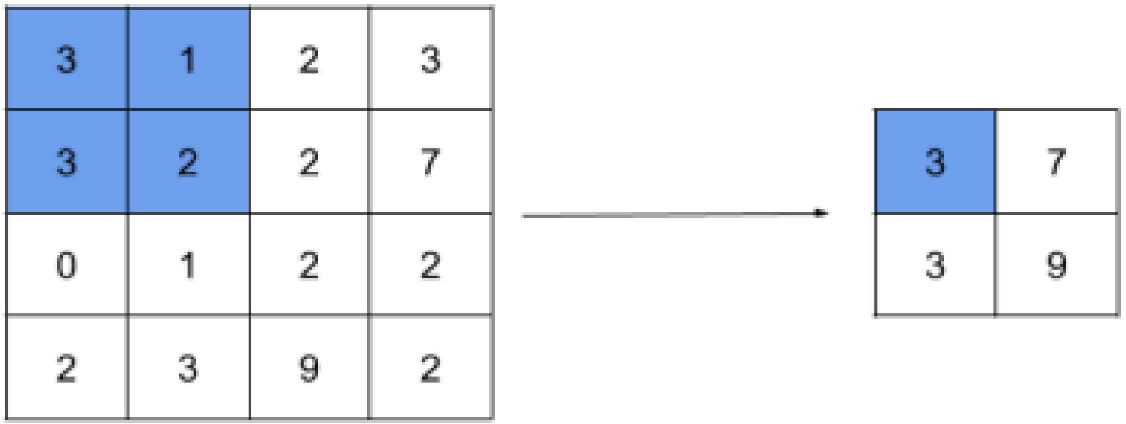
Methodology [5/10]

2. Max-Pooling Layers

In the proposed CNN, each convolutional layer is followed by a max-pooling layer. Its functions are:

- Lowering the computational cost of training & running CNN by reducing input's dimension;
- Making the input's representation invariant to minor transformations of the input.

Max-pooling process can be presented as a window that slides over the input and returns the maximum value that the window contains.



Example of max-pooling operation

Methodology [6/10]

3. Dense Layers

The proposed CNN has 2 dense layers with the following parameters:

Parameter	**Description**	**Set Value**
Num. of neurons	Number of neurons in each layer.	1024 neurons in the hidden layer, 120 neurons in the output layer
Dropout	It randomly removes certain features by setting some percentage of weights to 0	0.35
Activation function	Performs a non-linear transformation of the layer's output. All operations in layers are linear, which does not let the network learn complex patterns of the input data.	Tanh for the hidden layer, Softmax for the output layer

Methodology [7/10]

4. Loss Function & Optimizer

Loss function is a minimization function that calculates the error of CNN. The proposed CNN's loss function is a **categorical cross-entropy loss.** Mathematically, it is defined as:

$$L = -\frac{1}{M}\sum_{m=1}^{M}\sum_{j=1}^{C} y_{mj} \log s_{mj} \ ,$$

Where M is the number of training samples; C is the number of classes; s_{mj} is a predicted probability score that the input m belongs to class j; and y_{mj} is a true probability score of sample m belonging to class j.

Optimizers are algorithms that define how to update the weights based on the change of a loss function.

Adam optimizer has the following advantages:
- Straightforward implementation;
- Low memory consumption;
- The optimizer works well with sparse gradients.

Methodology [8/10]

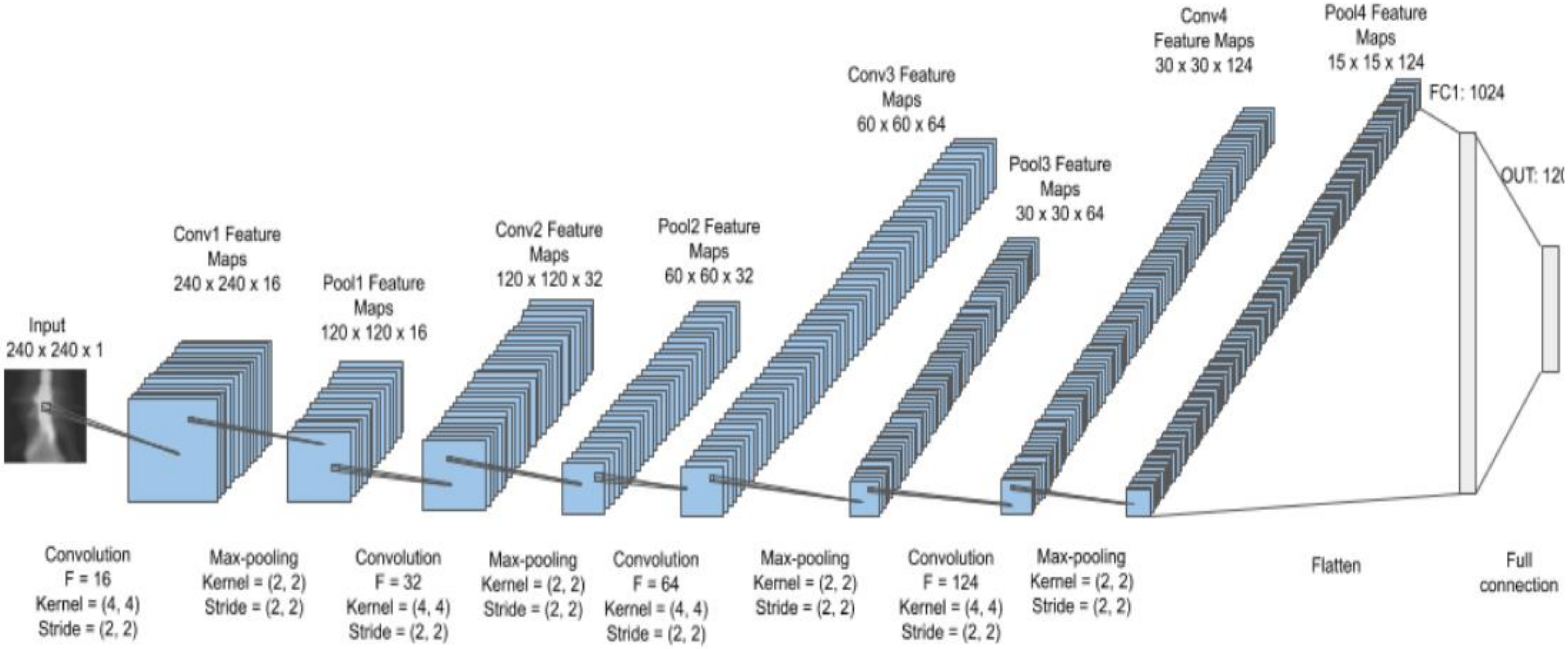
Architecture of the Proposed CNN

CNN consists of:

- 4 convolutional layers that extract distinctive features of the input;
- 4 max-pooling layers that make the network invariant to slight translations of the input matrix and decrease the number of the features;
- 2 dense layers to classify the output of the neural network.

Methodology [9/10]

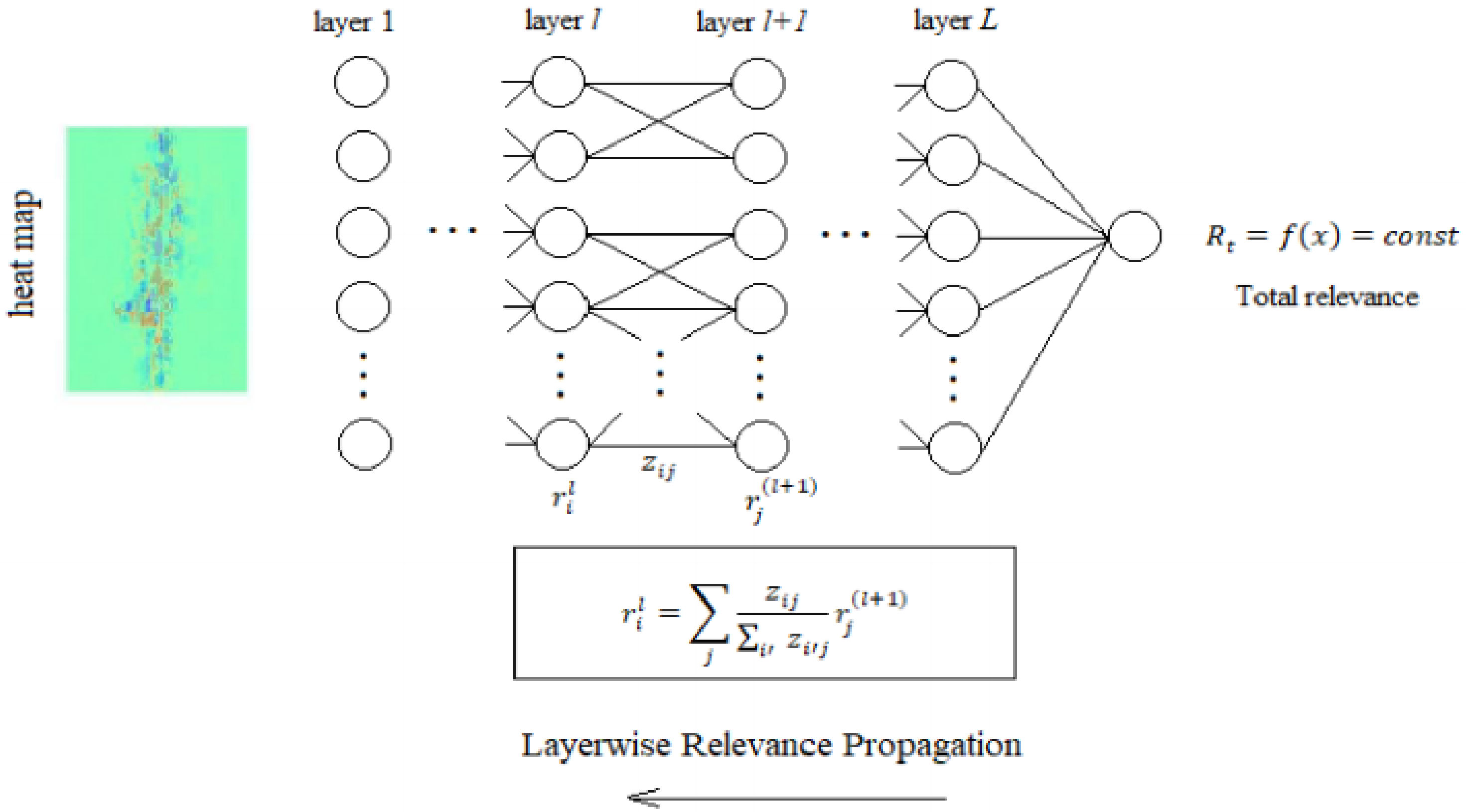
Architecture of the Proposed CNN contd.



Methodology [10/10]

Layer-wise relevance propagation (LRP):

LRP decomposes a neural network's output $f(x)$ into a heat map that indicates each input data point's relevance to the final decision of the network [13].



Here, r_i^l is a relevance score of i^{th} pixel of the l^{th} layer, and z_{ij} is a product of the i^{th} pixel value and j^{th} neuron's weight.

Simulations & Results [1/4]

Experimental Setup:

- CNN programmed in Python and using Keras/Tensorflow;
- The following optimizers were tested:

    1) Adam: the learning rate is 0.0001;

    2) Adadelta: the initial learning rate is 1.0 and the decay factor is 0.95;

    3) Nesterov-accelerated adaptive moment estimation (Nadam) with a learning rate of 0.0005.
- CNN was trained with a batch size of 12 for 40 epochs;
- The training was carried out on NVIDIA GeForce RTX 2080 Ti GPU with 11GB RAM.

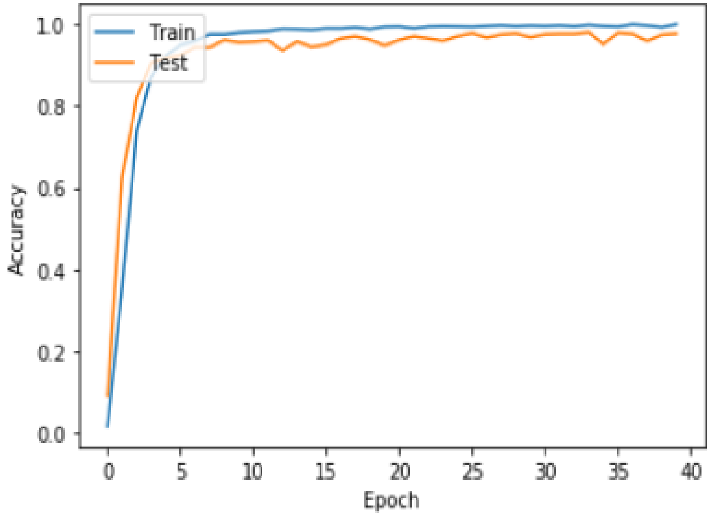
Simulations & Results [2/4]

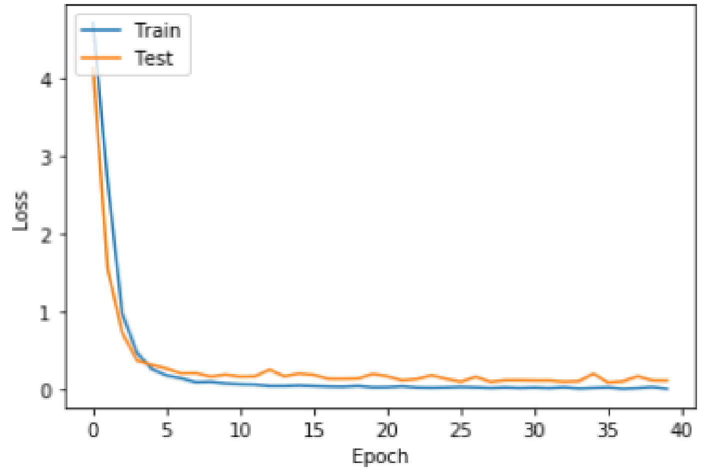
Optimizer Selection

Comparison of CNN performance with different optimizers

Optimizer	Accuracy (%)	Loss	Time for training (s)	Time for testing (s)
Adam	97.39	0.1488	563	0.058
Adadelta	97.27	0.2081	523	0.052
Nadam	96.68	0.2517	642	0.051

The Leaky ReLU activation function is applied to all the hidden layers.
Time for training and testing is measured when running a neural network on a workstation CPU 3.2 GHz with NVIDIA GeForce RTX 2080 Ti GPU with 11GB RAM.





CNN model's accuracy & loss with Adam optimizer and Leaky ReLU

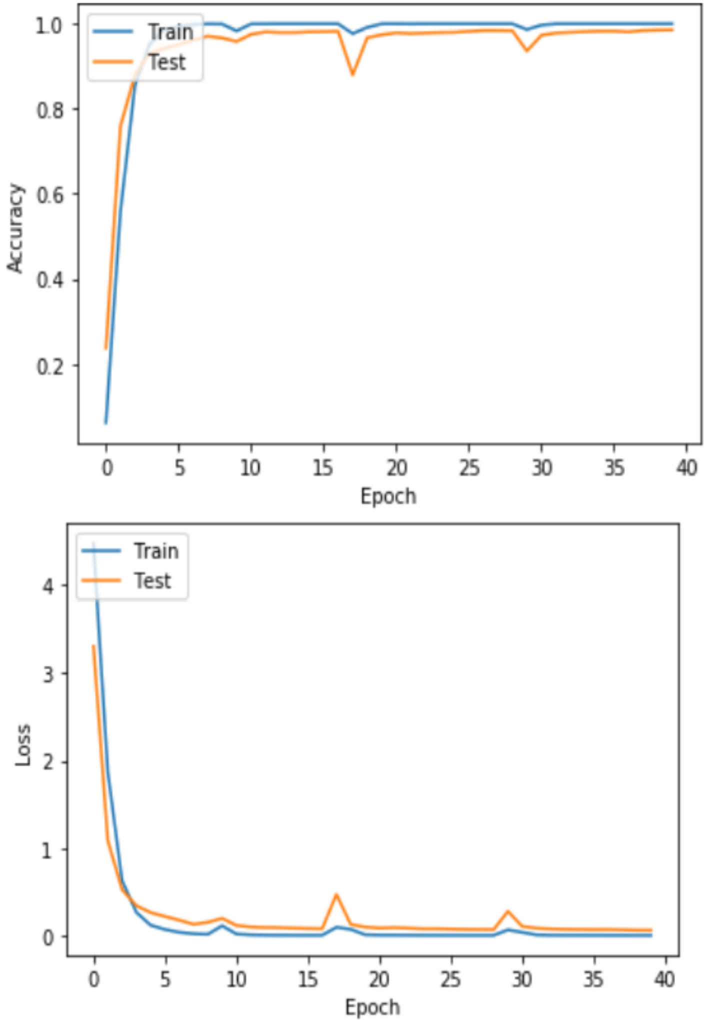
Simulations & Results [3/4]

Activation Function Selection

Comparison of CNN performance with different activation functions

Activation function	Accuracy (%)	Loss	Time for training (s)	Time for testing (s)
ReLU	96.41	0.1167	524	0.065
Leaky ReLU	97.39	0.1488	563	0.058
ELU	96.26	0.1970	524	0.057
Tanh	98.58	0.0584	527	0.058

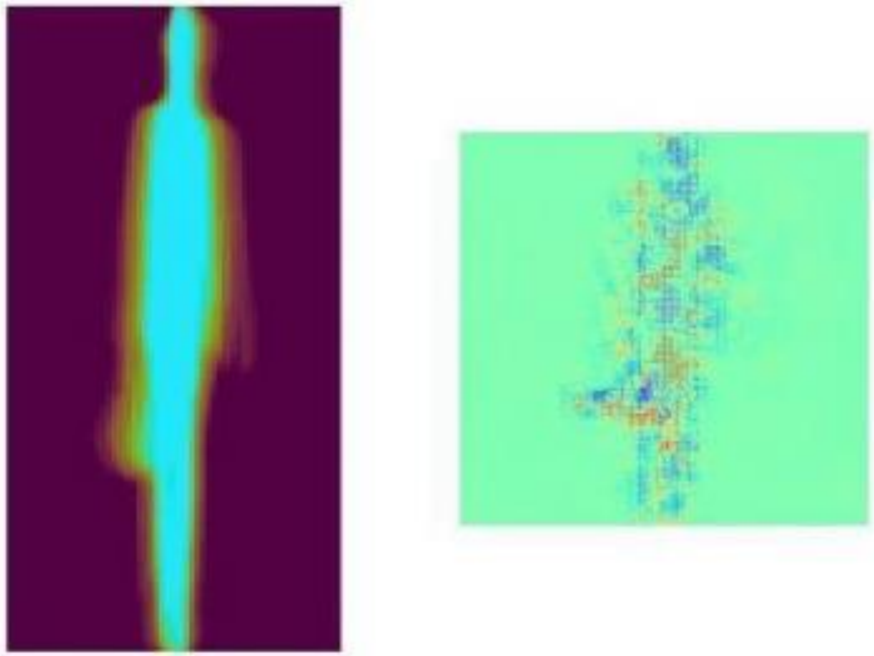
The Adam optimizer is applied to the CNN.
Time for training and testing is measured when running a neural network on a workstation CPU 3.2 GHz with NVIDIA GeForce RTX 2080 Ti GPU with 11GB RAM.



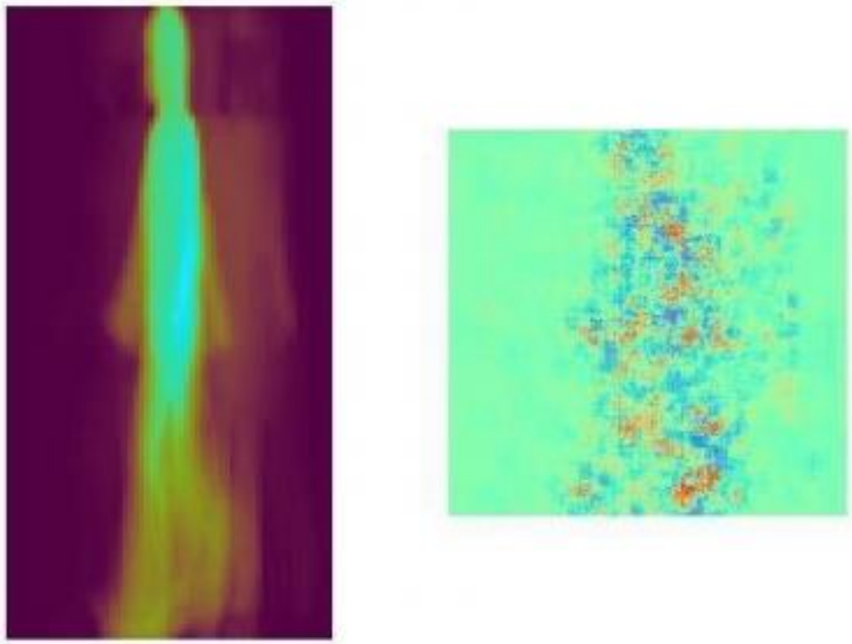
CNN model's accuracy & loss with Adam optimizer and Tanh

Simulations & Results [4/4]

Results of LRP



LRP applied to class ‘0’ at angle 180°



LRP applied to class ‘8’ at angle 90°

Conclusions

1. The proposed CNN achieved an accuracy rate of 98.58%. This is the highest accuracy rate achieved for the entire CASIA B dataset regardless of the view angle and clothing covariate: the state-of-the-art accuracy was 92.95% [3].

2. LRP has shown that the CNN focuses on the edge features like a distance between legs, height of the steps, position of hands in reference to other body parts, etc.

3. The following issues should be addressed for practical implementation of the proposed CNN:

- It is necessary to select and implement a fast background subtraction algorithm;
- The camera should be mounted at approximately the same height as it was during data collection;
- The background on the video should be relatively static for effective silhouette extraction;
- The silhouettes of several people captured on a frame should not overlap each other.

Future Work

The future work needs to seek solutions of the following problems:

- Robustness to shoes variations (heels and flats);
- Robustness to weight variation, i.e. a network should be able to recognize a person even after one's significant weight loss/gain;
- Gait recognition when several persons' silhouettes overlap each other on the video.

Also, to enlarge the CNN's recognition diapason, there are two potential solutions to be explored:

- If the number of new labels is much smaller than the initial quantity, i.e. $n \ll 120$, then it would be optimal to use the CNN's pre-trained weights, and change the output Softmax layer from 120 neurons to $120 + n$;

- If the number of new labels is significant, then the structure of the CNN itself needs to be partially changed to increase its capacity [14].

References [1/3]

[1] Y. Wang, J. Sun, J. Li, and D. Zhao, “Gait recognition based on 3D skeleton joints captured by Kinect,” in Proc. IEEE Int. Conf. Image Process. (ICIP), Sep. 2016, pp. 3151–3155.

[2] X. Chen, J. Xu, and J. Weng, “Multi-gait recognition using hypergraph partition,” Mach. Vis. Appl., vol. 28, nos. 1–2, pp. 117–127, Feb. 2017.

[3] A. Sokolova and A. Konushin, "Pose-based deep gait recognition", IET Biometrics, vol. 8, no. 2, pp. 134-143, 2019.

[4] X. Li, Y. Makihara, C. Xu, Y. Yagi and M. Ren, "Joint Intensity Transformer Network for Gait Recognition Robust Against Clothing and Carrying Status," in IEEE Transactions on Information Forensics and Security, vol. 14, no. 12, pp. 3102-3115, Dec. 2019.

[5] Y. He, J. Zhang, H. Shan and L. Wang, "Multi-Task GANs for View-Specific Feature Learning in Gait Recognition," in IEEE Transactions on Information Forensics and Security, vol. 14, no. 1, pp. 102-113, Jan. 2019.

References [2/3]

[6] [13] S. Bei, J. Deng, Z. Zhen and S. Shaojing, "Gender Recognition via Fused Silhouette Features Based on Visual Sensors," in IEEE Sensors Journal, vol. 19, no. 20, pp. 9496-9503, 15 Oct.15, 2019.

[7] X. Wang and S. Feng, "Multi-perspective gait recognition based on classifier fusion," in IET Image Processing, vol. 13, no. 11, pp. 1885-1891, 19 9 2019.

[8] Y. Zhang, Y. Huang, S. Yu and L. Wang, "Cross-View Gait Recognition by Discriminative Feature Learning," in IEEE Transactions on Image Processing, vol. 29, pp. 1001- 1015, 2020. 43

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

[10] Y. Makihara, H. Mannami, A. Tsuji, M. Hossain, K. Sugiura, A. Mori, Y. Yagi, The OU-ISIR gait database comprising the treadmill dataset, IPSJ Transactions on Computer Vision and Applications 4, 2012, pp. 53–62.

References [3/3]

[11] Martin Hofmann, Jürgen Geiger, Sebastian Bachmann, Björn Schuller, Gerhard Rigoll: "The TUM Gait from Audio, Image and Depth (GAID) Database: Multimodal Recognition of Subjects and Traits," in Journal of Visual Communication and Image Representation, Special Issue on Visual Understanding and Applications with RGB-D Cameras, vol. 25, no. 1, pp. 195-206, Elsevier, 2014

[12] S. Sarkar, P. Jonathon Phillips, Z. Liu, I. Robledo, P. Grother, K. W. Bowyer, “The Human ID Gait Challenge Problem: Data Sets, Performance, and Analysis,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, no. 2, pp. 162 – 177, Feb. 2005.

[13] G. Montavon, W. Samek and K. Müller, "Methods for interpreting and understanding deep neural networks", Digital Signal Processing, vol. 73, pp. 1-15, 2018.

[14] T.Xiao, J. Zhang, K. Yang, Y. Peng и Z. Zhang, “Error-Driven Incremental Learning in Deep Convolutional Neural Network for Large-Scale Image Classification”, in ACM Conference on Multimedia, 2014.