Person recognition based on footstep sounds using a deep convolutional neural network

Lépéshang alapú személyfelismerés mély konvolúciós neurális hálóval

Kristóf Rácz
PhD student
Géza Pattantyús-Ábrahám Doctoral School of Mechanical Engineering Sciences
Budapest University of technology and Economics, Faculty of Mechanical Engineering,
Department of Mechatronics, Optics and Engineering Informatics
racz.kristof@mogi.bme.hu

Abstract

Biometrics are methods to identify persons based on their biological features. From these, gait is one of the best features to use: it can be measured without any extra action from the person, and it is difficult to fake and mask. There have been attempts in the past use a small subset of information available from gait as data for identification: the sound of footsteps. Many methods have been proposed but none of them uses deep learning to solve the problem. Here, a solution is presented for the problem using a deep convolutional neural network (InceptionV3), which achieves a 98% accuracy in identifying the persons based on their footstep sounds.

Absztrakt

A biometrika azon módszerek gyűjtőfogalma, amelyek az egyes személyeket biológiai tulajdonságaik alapján igyekszik azonosítani. Ezek közül a tulajdonságok közül a járáskép mondható az egyik legmegbízhatóbb módszernek: az azonosítandó személy külön beavatkozása nélkül mérhető, és e mellett kifejezetten nehéz megváltoztatni vagy imitálni. A múltban már volt példa arra, hogy a járáskép egy kis alhalmazát, a járás hangját alkalmazzák emberek felismerésére. Több módszer is megjelent, azonban ezek közül egyik sem alkalmazta a deep learning technikáját. Jelen dokumentumban a feladat egy mély konvolúciós hálót (InceptionV3) alkalmazó megoldás kerül bemutatásra, mely 98%-os pontosságot ért el a személyek játáshang alapú azonosításának feladatában.

1 Introduction

Biometrics can be used to identify people: from our fingerprints and iris to our DNA, numerous biological parameters can be used to identify a person (Chinegaram, 2017). Security systems use this to control who has access to important places or information (e.g. unlocking our phone or pc via a fingerprint sensor), and Surveillance systems have been known to track people by recognizing their face. However, these systems are not infallible: contact lenses for iris scanners, printed silicon fingerprints for fingerprint scanners, masks for face recognition or voice recording for voice detection. However, it's also been shown numerous times, that a person's gait (the way he/she walks) is unique for all individuals. Using gait as a biometric identifier has some key advantages:

- It can be detected visually (using cameras), or by the vibrations it generates (microphones for the generated sound, or sensors recording mechanical vibrations), without interrupting the activities of the person
- It is difficult to replicate an other person's patterns
- It is also difficult to mask one's gait patterns. This could provide more reliable tracking in surveillance systems, as hiding or changing one's face is much easier than hiding one's walking patterns.

Identifying someone based on the vibrations their steps generate could prove useful in today's world of smart devices. For example a corridor instrumented with vibration sensors could be used to unlock doors for the right person without interrupting their flow at all.

Although the research presented here deals with footstep sound recorded with a microphone, the principles and processes would be easily adaptable to seismic data: the only difference is the media through which the vibrations reach the sensor, and the sensors themselves.

2 Literature

A number of studies have explored the topic of identifying people based on footstep sounds. In 1999 TANAKA and INOUE recorded footsteps on a tile floor using a single microphone. They were able to achieve an 83% recognition rate by extracting the pitch frequency and first peak frequency of footsteps and performing what is essentially K-means clustering. It's not exactly clear as to what this recognition rate is in reference to though, whether it is for recognizing people, or recognizing footwear.

SHOJI ET AL. preformed a similar experiment in 2005, where they used one microphone to record the footsteps of five persons wearing slippers. They used mel-cepstrum analysis to extract a feature vector from the recordings, and applied the k-means method to these. No clear overall accuracy is given for recognizing the persons.

In 2014 GEIGER ET AL. achieved 65.5% recognition rate on a large database of 305 subjects using hidden markov models. In 2017 GUO AND WANG implemented a Bayesian decision classifier for samples in noisy environments, resulting in a more robust classifier than previous works, which is able to achieve over 80% accuracy in ideal conditions.

MUKHOPADHYAY ET AL. used a different approach on recording in 2018: they used geophones (vibration sensors) to collect footstep data. Using a Gaussian support vector machine, they were able to achieve a 92.29% accuracy.

Also in 2018, RIWUROHI ET AL. used a multi-microphone setup for data collection, calculated Mel Frequency Cepstrum Coefficients from the recordings, and used a Feed forward neural network with 5 inputs (minimum, maximum, mean, standard deviation and median values of the cepstrum) and 2 hidden layers. They achieved 98.7% accuracy.

3 Methods

3.1 Data

3.1.1 Data collection

Data was collected in the Motion Laboratory of the Department of Mechatronics, Optic and Engineering Informatics.

The recordings of footstep sound were collected from 6 individuals, with a total of 32 individual shoe - person combinations (including socks). For each combination a total of 20 takes were recorded (except for one case with only 19 takes). One take consisted of the person walking from the edge of the room towards the microphones in a straight line. The recording was stopped when the person reached the microphones. This took them between 3 and 7 steps.

The recording equipment was lent by the Kármán Studio, including: a dynamic (Shure SM58) and a condenser (AudioTechnica AT2020) microphone, a small mixer and a usb sound card, as well as the required cables and microphone stands. The signal from the two microphones were recorded simultaneously on the stereo tracks of the recordings (one microphone recorded only to the right, the other only to the left channel). The data was recorded using Audacity[1] on a 64 bit Windows 10 machine at a rate of 44.1 kHz, and 32 bit resolution. The recording setup can be seen on *figure 1*. A sample of the equipment and environment noise was also recorded by recording a few seconds in 'silence'.

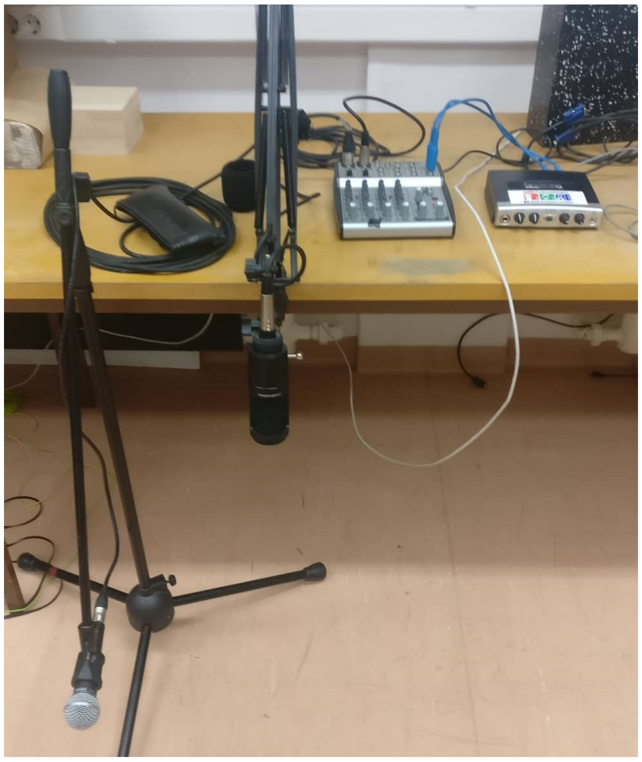

Figure 1: Audio recording setup with two microphones, a mixer and an USB sound card

3.1.2 Data preprocessing

Each take was exported to an individual `*.wav` file with the naming scheme `ID_shoeType_Take#.wav`. A python scrip was used to process these files into mel spectrograms. Only the recordings of the AudiTechnica microphone was used. The script performed the following steps to process the audio files (all Butterworth filters were second order):

1. load the recording of environment and equipment noise data
2. separate the mono track of the noise profile of the AudioTechnica microphone from the recording
3. filter the noise profile with a Butterworth highpass filter ($f_{cutoff} = 1kHz$) filter
4. iterate through the recordings in the input directory:
   (a) load the next recording and separate the mono channel of the AudioTechnica microphone
   (b) apply a noise reduction based on the noise profile
   (c) apply a Butterworth highpass filter ($f_{cutoff} = 100Hz$) filter
   (d) apply a Butterworth notch filter ($f_{cutoff} = 45Hz, width = 2Hz$) filter
   (e) calculate the mel spectrogram using the `librosa` package
   (f) plot the spectrogram using `matplotlib`

[1]https://www.audacityteam.org

(g) save the spectrogram to a file in the appropriate train/validate/test folder based on the take number of the original file (takes 1-16 go to train, 17-18 to validate and 19-20 to test data)

An example of the output can be seen in *figure 2*.

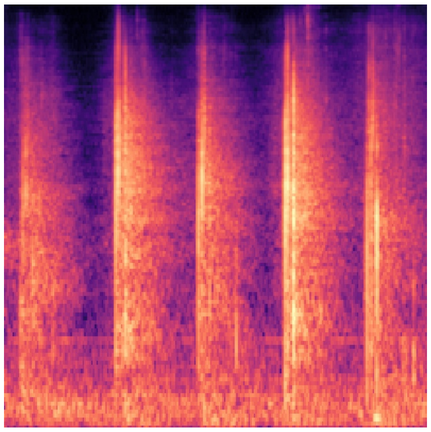

Figure 2: Example mel spectrogram output of data preprocessing

3.2 Model

I choose the `InceptionV3` (SZEGEDY ET AL. 2015) convolutional neural network as the basis for my model. The top part of the model is replaced with a dense feed forward network to classify the data into 6 classes (as per the 6 persons whose footsteps were recorded). The structure of this part is determined based on hyperparameter optimization, with the following general structure:

1. input layer from InceptionV3, with a shape of (6, 6, 2048)
2. a pooling layer
   - `GlobalMaxPooling()`
   - `GlobalAveragePooling()`
   - `None`
3. a flatten layer
4. 1 to 3 hidden dense layers, with each having
   - 8, 16, 32, 64, 128, 256 or 512 neurons (independent of the other dense layers)
   - an activation function: `ReLU`, `LeakyReLU` or `Swish` (RAMACHANDRAN ET AL. 2017)
   - a normalization layer: either `BatchNormalization` before activation, or `Dropout` after activation
5. an output dense layer with 6 neurons and `Softmax` activation

In addition, the following parameters related to training the model:

- an optimizer
  - `Adam` or `SGD`
  - A learning rate of 0.01, 0.001 or 0.0001
- a `batch_size` of 4, 8, 16, 32, 64, 128 or 256

3.3 Training

I tried to follow the general steps of transfer learning on a CNN, but training the classification model on the bottleneck features of InceptionV3 with the ImageNet wights proved to be of little use: it was clear that the feature extraction needed for the spectrograms is different than what is used for normal photographs. Based on this, my strategy for training the model was the following:

1. pretrain `InceptionV3` for feature extraction on spectrograms:
   (a) put an arbitrary classifier on top (`Flatten, Dense(126, activation='relu'), Dense(32, activation='relu'), Dense(6, activation='softmax')`)
   (b) Train the model on the train dataset, validate with validation dataset
   (c) Save trained weights of `InceptionV3`
2. generate bottleNeck features from the data with the pretrained `InceptionV3`
3. perform hyperparameter optimization with `keras-tuner` on the dense classification network using the generated bottleneck features
4. train the selected classification architecture on the bottleneck features
5. join `InceptionV3` with the pretrained wights to the trained classification network
6. train the complete model on the data to fine-tune everything

4 Results

4.1 Hyperparameter optimization

The newly released `keras-tuner` framework was used to perfrom a hyperparameter otimization using the HyperBand algorithm (LI AND JAMIESON, 2018). The hyperparameters of the first 20 best performing models can be seen in *table 1*. I did not find an easy way to also get the metrics related to the performance metrics of the models, such as their validation accuracy and loss, the time it took to complete one epoch of training, and the total number of epoch before stopping.

Based on the first 20 models, the parameters that really matter are the:

- pooling, from which no pooling layer is the best
- the regularization layer, where Dropout is a clear winner over using BatchNorm
- the optimizer, from which SGD with a learning rate of 0.0010 is the most common

For the rest of the hyperparameters the following values were selected with the goal of minimizing the number of parameters to train:

- Swish activation function
- 1 hidden layer with 64 neurons
- A batch_size of 16

Due to an error in defining the model which resulted in only having 0.1 tested as the dropout rate, I've run an additional round of hyperparameter optimization, this time with hyperparameters selected above, and a dropout rate between 0.1 and 0.5, with 0.005 steps. There was no really significant difference, but the results showed that 0.2 is the best value for this parameter.

The complete model has total of 26 521 830 parameters, of which 21 802 784 belong to InceptionV3, and 4 719 046 belong to the selected fully connected classifier.

4.2 Training

As described earlier, the classification model was trained on the bottleneck features first, and then the whole model was trained on the original data. The evaluation of the final model resulted on the metrics reported in *table 2*. The confusion-matrix fo the predictions can be seen on *figure 3*. In summary, the network only misclassified a single sample from the test dataset, which can be considered very good. With these results it is among, if not the best performing solution for identifying people based on their footstep sounds compared with the literature.

5 Conclusion

The trained model performed the task excellently, maybe even too well. It could be an indication that it's much more complex than it needs to be, or the data is too homogeneous. I am planning on collecting more data from new persons, and seeing how many classes can the model handle.

Table 1: Hyperparameters of the first 20 models

Place	Pooling	Activation	Regularization	learning_rate	Optimizer	batch_size	nLayers	nNeurons_0	nNeurons_1	nNeurons_2
0	None	Swish	Dropout	0.0100	SGD	16	2	128	16	16.0
1	None	LeakyReLU	Dropout	0.0010	SGD	8	2	256	64	128.0
2	None	ReLU	Dropout	0.0010	SGD	64	2	64	64	256.0
3	None	LeakyReLU	Dropout	0.0010	SGD	8	2	256	64	128.0
4	None	LeakyReLU	Dropout	0.0010	Adam	128	2	256	16	128.0
5	None	LeakyReLU	Dropout	0.0010	SGD	32	3	64	256	128.0
6	None	ReLU	Dropout	0.0010	SGD	64	2	64	64	256.0
7	None	LeakyReLU	Dropout	0.0010	SGD	128	2	128	8	64.0
8	None	ReLU	Dropout	0.0010	SGD	256	3	256	512	128.0
9	None	LeakyReLU	Dropout	0.0010	SGD	32	3	64	256	128.0
10	None	Swish	Dropout	0.0001	SGD	16	2	256	16	64.0
11	None	Swish	Dropout	0.0010	Adam	16	1	64	16	32.0
12	None	LeakyReLU	Dropout	0.0010	SGD	8	2	256	64	128.0
13	None	Swish	Dropout	0.0010	Adam	32	1	256	8	128.0
14	None	Swish	Dropout	0.0001	SGD	16	2	256	16	64.0
15	None	Swish	Dropout	0.0010	Adam	16	1	64	16	32.0
16	None	Swish	Dropout	0.0001	SGD	32	2	512	16	64.0
17	None	Swish	Dropout	0.0100	SGD	16	2	128	16	16.0
18	None	ReLU	Dropout	0.0010	SGD	256	3	256	512	128.0
19	None	Swish	Dropout	0.0001	Adam	64	1	32	128	256.0

Table 2: Evaluation metrics of the model

Metric	Value
Test accuracy:	0.98
Test loss:	0.0462
Test precision:	0.976
Test recall:	0.983
Test f1_score:	0.978
Number of misclassified samples:	1

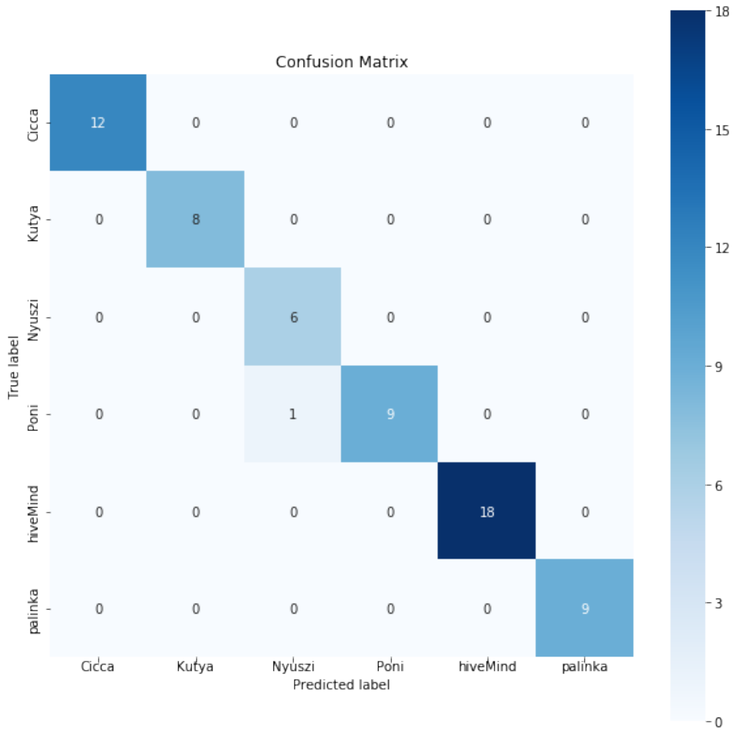

Figure 3: Confusion matrix for the test samples

Another option is to streamline the architecture os InceptionV3. The input originally consist of colored images, spectrograms however, by nature, can be represented as grayscale images. The first layers of the network could be modified accordingly. It'd also be interesting to explore if the depth of the convolutional model can be reduced without loosing performance.

Another interesting challenge would be to split the data in a different manner, based on shoe types. Assign one shoe type as validation, an other shoe type as test (preferably the socks option, as intuitively that is the hardest type to predict, as the amount of information is minimal, the footsteps are very quiet). Unfortunately, as some persons only recorded with three shoes, this would make the data even more sparse than it already is.

References

Chinegaram, K. (2017) Various Biometric Authentication Techniques: A Review. *Journal of Biometrics & Biostatistics* **8** (5):371.

Geiger, J. T. & Kneißl, M. & Schuller, B. & Rigoll, G. (2014) Acoustic gait-based person identification using hidden markov models. *MAPTRAITS 2014 - Proceedings of the 1st ACM Audio/Video Mapping Personality Traits Challenge and Workshop, Co-located with ICMI 2014* pp. 25-30

Guo, F. & Wang, X. (2017) Robust footstep identification system based on acoustic local features. *IET Biometrics* **6** (6):387-392

Li, L. &, Jamieson, K. (2018) Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization. *Journal of Machine Learning Research* **18** (185):1-52.

Mukhopadhyay, B. & Anchal, S. & Kar, S. (2018) Person Identification using Seismic Signals generated from Footfalls. *arXiv:1809.08783*

Ramachandran, P. & Zoph, B. & Le, Q. (2017) Swish: a Self-Gated Activation Function. *arXiv:1710.05941.*

Riwurohi, J. E. & Istiyanto, J. E. & Mustofa, K. & Putra, A. E. (2018) People Recognition through Footstep Sound Using MFCC Extraction Method of Artificial Neural Network Back Propagation. *International Journal of Computer Science and Network Security* **18** (4):28-35

Shoji, Y. & Itai, A. & Yasukawa, H. (2005) Personal identification using footstep detection in in-door environment. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences* **E88-A** (8):2072-2076

Szegedy, C. & Vanhoucke, V. $ Ioffe, S. & Shlens, J. & Wojna, Z. (2015) Rethinking the Inception Architecture for Computer Vision *arXiv:1512.00567.*

Tanaka, M. & Inoue, H. (1999) A Study on walk-recognitions by frequency analysis of footsteps. *EEJ Transactions on Electronics, Information and Systems* **119** (6):762-763