Rotation-Invariant Gait Identification with Quaternion Convolutional Neural Networks (Student Abstract)

Bowen Jing,[1] Vinay Prabhu,[2] Angela Gu,[1] John Whaley[2]
[1] Stanford University, Stanford, CA
[2] UnifyID, Inc., Redwood City, CA
bjing@stanford.edu, vinayup@gmail.com, angelajgu@alumni.stanford.edu

Abstract

Accelerometric gait identification systems should ideally be robust to changes in device orientation from the enrollment phase to the deployment phase. However, traditional Convolutional Neural Networks (CNNs) used in these systems compensate poorly for such distributional shifts. In this paper, we target this problem by introducing an $SO(3)$-equivariant quaternion convolutional kernel inside the CNN. Our architecture (Quaternion CNN) significantly outperforms a traditional CNN in a multi-user gait classification setting. Additionally, the kernels learned by QCNN can be visualized as basis-independent trajectory fragments in Euclidean space, a novel mode of feature visualization and extraction.

Introduction

Accelerometric gait identification systems have increasingly embraced convolutional neural networks (CNNs) in lieu of hand-crafted features and shallow ML approaches (Gafurov 2007). These systems entail an *enrollment* phase in which the CNNs are trained to distinguish among a set of enrolled users by their gait (Gadaleta and Rossi 2018), and a deployment phase where newly presented gait cycles are classified as belonging to one of the enrolled users. As long as the user maintains the same device orientation, these CNNs perform with high accuracy in the deployment phase. However, they experience a catastrophic drop in accuracy if the user changes the orientation. This device-flip and the resulting distributional shift are typically tackled by either using only the acceleration *magnitude*, or by explicitly performing rotation invariant transforms (Gadaleta and Rossi 2018). The first option lowers the accuracy by discarding the rich 3D spatial information in the input accelerometric tensors, and the second option is both computationally expensive and adds to the software complexity of the system, especially for on-device implementations.

We instead incorporate rotation invariance *inside* the CNN architecture by introducing a novel convolutional kernel which leverages quaternion representations of spatial rotations to learn $SO(3)$-equivariant maps between trajectory fragments in $\mathbb{R}^3$. Such kernels can then be stacked to form a network which is agnostic to device orientation and instead learns to recognize basis-independent gait signatures. Our work differs from a number of existing $SO(3)$-equivariant architectures (Thomas et al. 2018) in that it operates on arrays of vectors, not featurized point clouds or volumetric data—that is, it is *not* permutation invariant, since input gait cycles are properly viewed as a vector time series, not a point cloud or shape. It also differs from previous Quaternion CNNs (Zhu et al. 2018) which leverage quaternion algebra for parameter efficiency, but do not operate on spatial data.

Method

A QCNN kernel takes in a linear array of quaternions, performs a convolution-like operation on sliding windows of the input array, and outputs another array of quaternions. A gait cycle is a linear array of accelerometric vectors in $\mathbb{R}^3$; we represent such vectors as pure quaternions and consider a general quaternion to be pair of a real number and a vector: $\hat{r} = r + \mathbf{r}$. We also make use of the representation of spatial rotation as *quaternion conjugation*: if $\hat{r} = r + \mathbf{r}$, then $\hat{r}\hat{v}\hat{r}^{-1}$ yields a quaternion whose vector part is $\mathbf{v}$ rotated by an angle $\theta = 2 \arccos r/|\hat{r}|$ about the axis defined by $\mathbf{r}$ and whose real part is unchanged.

Let a window of quaternions $\hat{q}_0, \hat{q}_1, \dots, \hat{q}_{2l}$ be the input to a single-channel convolutional filter of length $2l + 1$. Then the output of the filter for that window is

$$f(\hat{q}_0, \hat{q}_1, \dots, \hat{q}_{2l}) = \sum_{i=0}^{2l} a_i(\hat{q}_i + b_i)(\hat{q}_l + c_i)\hat{q}_i(\hat{q}_l + c_i)^{-1}$$

where $a_i, b_i, c_i \in \mathbb{R}$ are the learnable parameters of the filter. This output is itself a quaternion, and is equivariant under 3D rotations of the *vector part* of the input. That is, for any $\hat{r} \in \mathbb{H}$, $\hat{r}f(\hat{q}_0, \hat{q}_1, \dots, \hat{q}_{2l})\hat{r}^{-1} = f(\hat{r}\hat{q}_0\hat{r}^{-1}, \hat{r}\hat{q}_1\hat{r}^{-1}, \dots, \hat{r}\hat{q}_{2l}\hat{r}^{-1})$.

Quaternion convolutional kernels can be stacked with no nonlinearity required between layers, as the composition of multiple convolutions is not reducible to a single convolution. The last quaternion layer can be joined to real-valued layers by just taking the magnitude $\hat{q} \to |\hat{q}|$ of each quaternion to construct a deep, rotation-*invariant* classifier.

Dataset	QCNN	Standard CNN
Training ($n = 529$)	79.0%	**83.3%**
Validation ($n = 106$)	57.7%	**74.5%**
Testing ($n = 635$)	**46.5%**	4.4%

Table 1: Mean classification accuracies across 10 training runs on the three datasets in the cotemporal experiments.

Train set/Test set	QCNN	Standard CNN
Original/Original	33.28%	**36.82%**
Original/Rotated	**33.28%**	17.40%
Rotated/Rotated	**33.41%**	29.32%

Table 2: Top-5 test classification accuracies on the three datasets in the multi-user experiments.

Figure 1: Six kernels from the first layer of the trained QCNN from the multi-user experiments. Each kernel is visualized by a trajectory fragment which maximizes the magnitude of the output quaternion, shown as an arrow.

Experiments

We compare against standard CNNs on two tasks. In the cotemporal experiments, gait cycles are recorded from users simultaneously carrying two devices with opposite orientations — one for training/enrollment and one for testing/deployment. This setup mimics the case of users enrolling in one orientation, but presenting another orientation to authenticate. In multi-user experiments, we examine a larger cohort for whom cotemporal data is not available, and randomly rotate the gait cycles to generate orientation-agnostic training and test sets. By training the CNN on rotated data, we are also able to evaluate the efficacy of data augmentation.

Cotemporal experiments We enroll a cohort of eight users who collectively generated 635 gait cycles in each orientation. The cycles in one orientation were split 529:106 train-val, and all 635 cycles of the other orientation were used for testing. We use a 4-layer network for both the CNN and QCNN; the CNN has 213k parameters while the QCNN has 59k parameters.

The results, averaged across 10 trials, are shown in Table 1. When there is only one fixed orientation, the standard CNN performs better than the QCNN. However, upon encoutering the test gaits in a different orientation, the CNN suffers a sharp drop in accuracy. The QCNN, on the other hand, is much more robust and has a 10x fold improvement over the CNN on the test set.

Multi-user experiments We use data from 100 users consisting of 1000 train/val and 100 test cycles per user. We then compare QCNN to a standard CNN when trained and tested on the original gaits, when trained on the original gaits and tested on the rotated gaits (similar to the cotemporal setup), and when both trained and tested on the rotated gaits (corresponding to training with data augmentation), When the training and test sets are both in their original orientations, the standard CNN has higher accuracy, but its performance drops precipitously when the test set is freely rotated (Table 2). Importantly, data augmentation — that is, training on rotated data — is unable to rescue the standard CNN to the performance level of the QCNN.

Kernel visualization We can visualize the features learned by the first quaternion layer as *trajectory fragments* in Euclidean space defined with respect to the *origin* and a *chirality* but independent of the axes (Figure 1). This is different from the features detected by shape and point-cloud networks, which are independent of the origin and oftentimes also reflection-equivariant. While kernels may appear to correspond to similar input features, they can map them to different outputs. Similarly, any single kernel may be activated by multiple inputs. QCNNs can therefore learn rich representations over the input space of gait cycles.

Conclusions

We have presented a neural network specifically tailored for gait-invariant accelerometric gait classification. This network outperforms standard convolutional networks, is parameter-efficient, and learns features which are easily visualized in 3D. Future work may focus on theoretical and empirical analyses of training and parameter initialization.

Ethics statement We have described a technique that seeks to improve a privacy-enhancing passive biometric system. Only users who have been clearly educated about this technology and who have provided an active consent of data usage were enrolled as part of the classification cohort.

Acknowledgments

BJ and AG acknowledge support from the UnifyID AI Fellowship.

References

Gadaleta, M.; and Rossi, M. 2018. Idnet: Smartphone-based gait recognition with convolutional neural networks. *Pattern Recognition* 74: 25–37.

Gafurov, D. 2007. A survey of biometric gait recognition: Approaches, security and challenges. In *Annual Norwegian computer science conference*, 19–21. Annual Norwegian Computer Science Conference Norway.

Thomas, N.; Smidt, T.; Kearnes, S.; Yang, L.; Li, L.; Kohlhoff, K.; and Riley, P. 2018. Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds. *arXiv preprint arXiv:1802.08219* .

Zhu, X.; Xu, Y.; Xu, H.; and Chen, C. 2018. Quaternion convolutional neural networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 631–647.