ARC-Net: Activity Recognition Through Capsules

Hamed Damirchi, Rooholla Khorrambakht and Hamid D. Taghirad, Senior Member, IEEE
Faculty of Electrical and Computer Engineering
K. N. Toosi University of Technology
P.O. Box 16315-1355, Tehran, lran
Email: hdamirchi, r.khorrambakht@email.kntu.ac.ir, taghirad@kntu.ac.ir

***Abstract*—Human Activity Recognition (HAR) is a crucial factor in assisted living systems and elderly care solutions where the activity of the subject can be used to ensure the safety of the elderly and provide more efficient services. HAR is a challenging problem that needs advanced solutions than using handcrafted features to achieve a desirable performance. Deep learning has been proposed as a solution to obtain more accurate HAR systems being robust against noise. In this paper, we introduce ARC-Net and propose the utilization of capsules to fuse the information from multiple inertial measurement units (IMUs) to predict the activity performed by the subject. We hypothesize that this network will be able to tune out the unnecessary information and will be able to make more accurate decisions through the iterative mechanism embedded in capsule networks. We provide heatmaps of the priors, learned by the network, to visualize the utilization of each of the data sources by the trained network. Then, gradient based interpretations are provided and further discussed. By using the proposed network, we were able to increase the accuracy of the state-of-the-art approaches by 2%. Furthermore, we investigate the directionality of the confusion matrices of our results and discuss the specificity of the activities based on the provided data.**

***Index Terms*—Activity recognition, capsule network, interpretable artificial intelligence, multimodal fusion.**

I. Introduction

In the human activity recognition field, the goal is to predict the activity of a human subject based on a window of measurements provided by the available sensors. These activities may range from lying to rope jumping. A wide variety of sensors such as accelerometer, gyroscope, magnetometer, force and light sensor may be used to classify the activity performed in that window of collected data. Even everyday devices such as smartphones and smartwatches may be used in order to obtain the required data for activity recognition. Due to availability and the low cost of the mentioned sensors, HAR is utilized in many areas such as design of exoskeletons [1] and more commonly, elderly care where HAR may be utilized for pervasive healthcare [2] and assisted living [3]. Furthermore, the collected data from sensors may be collected and utilized to provide more personalized services through the monitored activities. For example, the data collected from body-worn sensors and the behavior extracted form this data can help increase security [4] and allow the creation of better user interfaces [5].

Deep learning has proven to be an adequate tool for recognizing patterns and extracting rich features that may be utilized to classify data [6]. Human activities may be classified based on the patterns that are seen in the input data. Therefore, rather than looking only at the measurements from individual data sources, we require our algorithm to view the input data on different levels and fuse the extracted features in such a way that the perceived patterns would tell us about the activity being performed, e.g., to differentiate between walking and running, it would be misleading to only look at the data from a motion sensor located on the ankle of the subject and we would reach better performances by also using the data from subject's forearm. This calls for a technique that is able to adequately fuse the information from every data source and use the said information to make a prediction about the activity.

In this paper, we propose using a single Convolutional Neural Network (CNN) as an encoder to extract the features from each of the IMUs and pass the said features to CapsNet [7] in order to fuse the extracted information and make a prediction about the activity of the subject. We will use two datasets to evaluate the performance and generalization of our approach. Moreover, a modified version of the network proposed in [8] will be used as our encoder which uses stage-based fusion to extract the necessary information. Our contributions are as follows:

- CapsNet is used to fuse the information obtained from each IMU
- Provide, gradient-free, intuitive interpretations regarding the utilization of each of the IMUs based on the true label
- Provide gradient based interpretations describing the sensitivity of each class with respect to the given measurements
- Provide comparisons to empirically examine the capability of capsules in rejecting corrupted modalities
- Our method outperforms the state-of-the-art deep learning based approaches

This paper is organized as follows: In Section 2, we go through various available classical and deep learning-based approaches to HAR alongside the current state-of-the-art(SOTA) approaches. In Section 3, we explain the proposed architecture and provide an introduction to capsules. In the last section, we present quantitative and qualitative evaluations of the proposed method and its variations against SOTA approaches and provide visualizations alongside corresponding discussions.

II. Related work

From an algorithmic point of view, approaches to HAR may be divided into three groups, namely, classical approaches

978-1-7281-8470-8/20/$31.00 ©2020 IEEE
DOI 10.1109/ICMLA51294.2020.00215

that utilize preprocessing methods such as pose estimation, machine learning-based approaches that rely on hand-crafted features devised by an expert and deep learning-based methods that rely on gradient backpropagation in order to both extract features and perform classification. Deep learning based methods can be separated into multiple groups of architectures themselves. Each group utilizes a specific characteristic of an architecture to improve the accuracy of their predictions. This category of approaches may be divided further based on the memory of past inputs [13]–[15], unsupervised methods [11], [12], non-recurrent methods [8], [16].

In [9], video sequences captured by a monocular camera are used to classify the activity of subjects. Optical flow alongside background extraction methods is used as features which are then passed to a support vector machine (SVM) to generate predictions. [10] uses decision trees to classify the activity of the subject based on features extracted from an accelerometer of a smartphone. Ten features such as phone position on the human body, user location, age and sensor readings are passed through a decision tree to make predictions. [11] uses activity sets rather than single label ground truth values to predict the potential activities in the corresponding window of measurements from IMUs. This work also uses unsupervised learning methods prior to supervised learning in their training process to achieve a more effective feature representation. In [12], unsupervised learning methods are deployed where the number of activities is unknown. This is done by using clustering methods that operate on the frequency components of the measured acceleration and angular velocity values.

[13] proposes DeepConvLSTM which is an LSTM based network that utilizes a CNN to extract the features from inputs. Due to the usage of recurrent layers, this model falls into the category of memory-based networks. DeepConvLSTM aims to increase the performance of HAR systems by modeling temporal dependencies while using raw sensor data. In [14] various architecture choices are compared to reach a conclusion about the necessity of recurrence in HAR models. [15] uses variants of Recurrent Neural Networks (RNNs) to observe the cognitive decline by monitoring the daily activities of the subject, while [13]–[15] all use IMU measurements as network input.

[16] stacks IMU measurements from a smartphone and creates a window of measurements. These measurements are then passed through a CNN to predict the activity of the subject. This work aims to model the temporal connections between raw sensor measurements using the convolution operations that reside in CNNs. [8] also stacks the raw measurements from the available IMUs and uses convolutional layers to extract the features from said inputs. This work uses specific kernel sizes that allow the network to perform data fusion in multiple levels. Reference [8] also proposes that late sensor fusion is more effective due to the separate processing of each axis of sensor module in the initial layers. We base our encoder on this architecture with modifications to prevent loss of information during pooling layers. Moreover, we view HAR as a multimodal fusion problem. We use a single encoder to extract features from IMUs with no constraints on the position of the data source on subject's body. Thereafter, we use mechanisms that fuse the collected information and predict the performed activity.

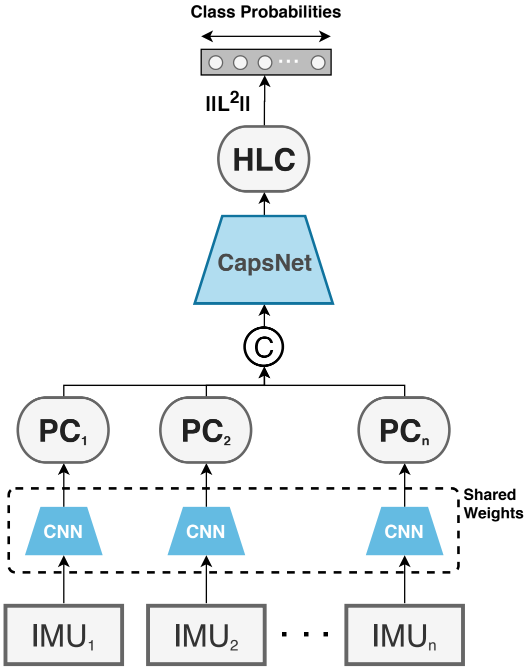


Fig. 1: A general overview of the proposed architecture

III. The Proposed Approach

We propose using CapsNets to fuse the high dimensional features passed from an encoder. The general overview of our architecture is shown in Fig. 1. We stack a pre-specified number of measurements from each IMU and create a two-dimensional array where columns represent each measurement. These arrays are passed to a single CNN separately and the features corresponding to each of the IMUs are extracted. Then, primary capsules are formed by reshaping the output of the CNN and concatenating the extracted features from each of the encoders. The concatenated features are then passed through CapsNet to obtain a probability vector of the activities. These steps are elaborated in detail in Section 3.A and 3.B.

A. Encoder

In order to extract features from each of the IMUs, we use a CNN with varying kernel sizes at each layer similar to the architecture proposed in [8]. A modified version of this architecture is presented in Fig. 2. As it is seen in this figure, at the first layer of the encoder, we perform a 1-dimensional convolution operation that does not perform any fusion on the given inputs and mainly acts as a filter. In the next layer, a 2-dimensional convolution is applied to the features of the former layer. Due to the size of the kernel and the strides of this layer, each module of the IMU is processed separately and the features of the accelerometer and gyroscope are not

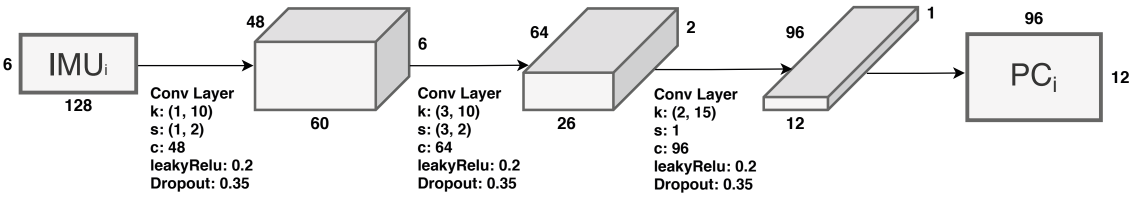


Fig. 2: The architecture of the encoder. k, s and c correspond to kernel size, stride size and number of output channels, respectively

fused in this layer. In the final layer, a kernel size of (2×15) convolves over the features of the second layer and fuses the information extracted from each of the modules. To be able to pass the extracted features to CapsNet, we will need to reshape these features as shown in Fig. 2.

B. CapsNet and Dynamic Routing

CapsNet [7] uses vectors of neurons (capsules) to represent the entities that may be present in the given input. Each capsule looks at a small window of the input and gives the probability of the existence of a specific pattern in the data. The magnitude of a capsule gives the probability of the existence of an entity and the orientation of the capsule in its high dimensional space defines the characteristics of that entity. The capsules in the first layer of the network, which are called primary capsules, are extracted through the encoder as shown in Fig. 2. CapsNet uses an iterative mechanism to dynamically route each of the lower layer capsules to the higher layer ones. Through this mechanism, the network is able to use the information about the existence of low level entities to decide about the presence of higher level ones. In our approach we only use one layer on top of the primary capsules in order to obtain predictions about the activity performed by the subject. Therefore, the higher level capsules will represent the activity that is potentially present in the given window of measurements. In other words, the routing mechanism will allow us to link the existence of specific patterns in each of the IMU measurements to the potential activity that is being performed by the subject.

By concatenating the primary capsules from each of the IMUs, we will have a feature map of size $12n \times 96$ where n is the number of IMUs. As seen in Fig. 2, each primary capsule has 96 dimensions and 12 capsules are extracted from each of the IMUs. Assuming U_i represents the i^{th} lower layer capsule and V_j is the j^{th} higher layer capsule, Algorithm 1 describes the routing mechanism used in our approach for r number of iterations. The squash non-linearity is formulated as follows

$$\text{squash}(\hat{\mathbf{V}}_j) = \frac{\left\|\hat{\mathbf{V}}_j\right\|^2}{1 + \left\|\hat{\mathbf{V}}_j\right\|^2} \cdot \frac{\hat{\mathbf{V}}_j}{\left\|\hat{\mathbf{V}}_j\right\|} \tag{1}$$

While, the softmax function is formulated as follolws:

$$\text{softmax}(b) = \frac{\exp(b_{ij})}{\sum_k \exp(b_{ik})} \tag{2}$$

Algorithm 1: Dynamic routing algorithm

```
initialize the log prior matrix b and set $b_{ij} \leftarrow 0$
$\hat{U}_{j|i} = U_i W_{ij}$
$c_{ij} \leftarrow \text{softmax}(b)$
for $r$ iterations do
    $\hat{c}_{ij} \leftarrow \text{softmax}(b)$
    $c_{ij} \leftarrow \eta \hat{c}_{ij} + c_{ij}$
    $\hat{V}_j \leftarrow \sum_i c_{ij} \hat{U}_{j|i}$
    $V_j \leftarrow \text{squash}(\hat{V}_j)$
    $b_{ij} \leftarrow b_{ij} + \hat{V}_j \cdot \hat{U}_{j|i}$
end
```

Following [17], we added a soft updating rule with the coefficient η in Algorithm 1 to prevent overrouting. Furthermore, we used the original loss function from [7] which is a margin loss that calculates a separate loss value for each of the predicted classes.

IV. Experiments

The experiments were conducted using an NVIDIA Tesla P100 with 16 gigabytes of RAM and 3584 CUDA cores. We used the PyTorch [18] framework to implement the proposed architecture. The code is available in our GitHub repository[1].

A. Datasets

The proposed method was tested on the PAMAP2 [19] and RealWorld [20] HAR datasets. The validation set of the PAMAP2 dataset was used to tune the model hyperparameters. The quantitative results from each of the datasets were compared against the state of the art methods with the same preprocessing characteristics. We report our results against PerceptionNet [8], DeepConvLSTM [13] and CNN-EF [16] on the PAMAP2 dataset and to test the generalizability of our method, we also provide quantitative results for PerceptionNet [8] and DeepConvLSTM [13] alongside our method on the RealWorld dataset. To train the network and to infer the activity of each of the subjects from either dataset, we only use the IMU measurements, namely accelerometer and gyroscope.

1) RealWorld: This dataset contains measurements from 15 subjects (8 males and 7 females) with recorded data from the chest, forearm, head, shin, thigh, upper arm, and

[1] https://github.com/hamed-d/ARC-Net.

waist of each of the subjects. For each subject, this dataset provides measurements from GPS, IMU, gyroscope, light, sound level data and magnetic field sensors. The annotated activities from this dataset are climbing up and down the stairs, jumping, lying, standing, sitting and running. The data is not synchronized and is recorded at a frequency of 50Hz. After synchronizing the IMUs, we segment the data into windows of 128 measurements (2.56 seconds) with 60% overlap. We use the leave-one-subject-out approach to evaluate our method. Specifically, we use subjects 10 and 11 as validation and test subjects and the rest are used to train the network.

2) PAMAP2: The physical activity monitoring dataset consists of 18 different physical activities from 9 subjects (8 males and 1 female). Three Colibri wireless inertial measurement units were used to provide measurements alongside a heart rate monitor. This dataset contains IMU measurements from the chest, dominant wrist and dominant ankle at a frequency of 100Hz. We downsampled this dataset to 50Hz in order to match the frequency of the RealWorld dataset and stacked 128 measurements with the same overlapping that was used for the preprocessing stage of the RealWorld dataset. Similar to PerceptionNet, we chose a leave-one-subject-out approach to validate and test our model. Subjects 1 and 5 were chosen as test and validation sets, respectively.

B. Performance evaluation metrics

Due to an imbalance in the number of labels for both of the datasets, we chose the weighted $F1$ score to report the quantitative results of our network. This score is formulated as below

$$wF1 = \sum_{c} \frac{N_c}{N} \frac{2 \cdot \text{Precision}_c \cdot \text{Recall}_c}{\text{Precision}_c + \text{Recall}_c} \tag{3}$$

Where c represents each class, while N is the total number of data and N_c is the number of data with label c. Moreover, we report the precision, recall and accuracy values separately and compare our results against the state of the art methods.

We also provide the confusion matrix as a qualitative measure for both of the datasets. This matrix allows us to interpret how the model wrongly classifies each of the categories. In the provided matrices, the rows correspond to the actual labels while columns represent the predicted classes. We will also take a look at the directionality of this matrix and discuss how easy it is for a model to confuse one class with the other but not the other way around. This approach will allow us to get a look at the specificity of the classes.

V. Results and Discussion

We used the Optuna [21] library to optimize the hyperparameters of the network. Due to a limited amount of computational power, we only used this library to search for iteration number(r), soft-updating value(η) and the initial learning rate. Twenty trials were conducted while each trial lasted 200 epochs. The best set of hyperparameters were chosen based on the average validation loss. Moreover, an exponential learning rate scheduler with a multiplicative factor equal to 0.98 was used to train the network on both datasets.

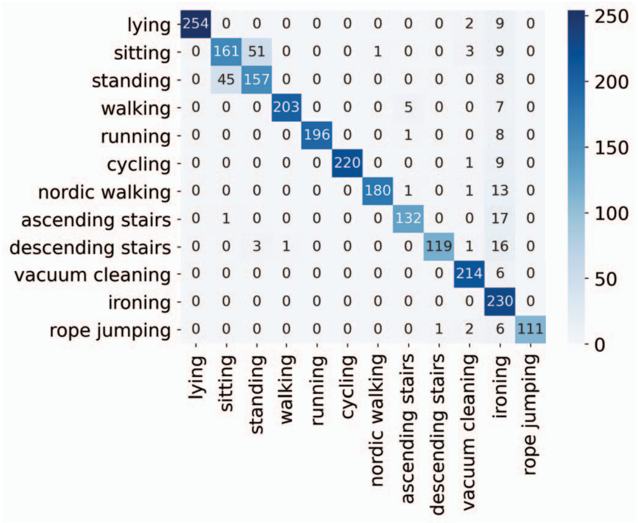


Fig. 3: Confusion map of the test set on the PAMAP2 dataset

A. Results on PAMAP2

Loss margins were set to 0.95 and 0.05 for this dataset. Moreover, the iteration number and the soft updating coefficient were set to 3 and 0.1, respectively, while the training proceeded with a batch size of 64. The test accuracy of our model ranges from 89.18% to 90.51% across multiple training sessions of the same model and the best single epoch based on the validation loss achieves an accuracy score of 90.51% on the test set. Due to the observed variance in the performance of each model during training, we also formed a horizontal voting ensemble [22], based on epochs of a single model using the top validation scores but the metrics did not improve when using this ensemble. Table I provides a comparison between the obtained results of our model and the reported metrics from the state of the art models. As it can be seen in Table I, the results from our model surpass the state of the art results on all metrics. The largest gap between ARC-Net and PerceptionNet can be seen in the precision achieved on the test results which is about 2.01%. Furthermore, A consistent improvement is seen on other metrics.

TABLE I: Results on PAMAP2

	Precision	Recall	wF1 Score	Accuracy
CNN-EF	85.51%	84.53%	84.57%	84.53%
DeepConvLSTM	87.75%	86.78%	86.83%	86.78%
PerceptionNet	89.76%	88.57%	88.74%	88.56%
ARC-Net	**91.77%**	**90.52%**	**90.76%**	**90.51%**

The confusion matrix of the test results of the network on the PAMAP2 dataset is provided in Fig. 3. The directionality of the confusion of the network in classifying sitting and standing activities suggests that the two activities are similar to each other based on the provided data. This is expected since this dataset does not provide any data from an IMU

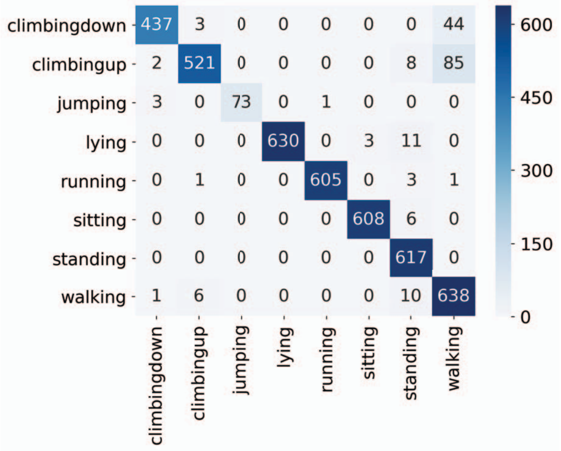


Fig. 4: Confusion map of the test set on the RealWorld dataset

that is positioned in such a way that would allow an easier differentiation between the two classes. If an IMU that is placed inside a pocket has been used, it would allow the network to be able to differentiate between the two more easily. Furthermore, lying, cycling, rope jumping and running have a precision of 100%. The precision of the ironing activity is low while having a high recall, specifically ironing has a recall of 100% while its precision is only 68.04%.

B. Results on RealWorld

The accuracy of our method on the test set ranges from 95.47% to 95.64% across multiple training sessions. We report our best epoch in Table II alongside a full comparison of our network against the state of the art methods. Due to the observed variance in the performance of the network during training, we also formed a horizontal voting ensemble based on the validation loss of each of the epochs of a single model. This way, we were able to increase the accuracy of our model to 95.92%. Same as the results achieved on PAMAP2, a consistent improvement of all the metrics is seen in Table II. Because of the relatively large number of sensors in this dataset, the number of iterations was increased to 7 while the soft updating coefficient was dropped to 0.01. By this means, the larger amount of iterations would allow the network to focus more on the routing of each primary capsule from each IMU to the high level capsules. Moreover, the smaller soft updating coefficient would prevent overrouting of the network. The confusion matrix corresponding to the test results of this dataset is provided in Fig. 4. This visualization shows that some samples from each activity are consistently being misclassified as the standing activity. This is due to the data collection protocol where the subjects are asked to stand up before performing any of the activities available in the dataset. Moreover, climbing up and down is also being largely misclassified as walking. Based on the footage of the activities provided by [20], the misclassifications have to do with the sections of the path where no climbing up or down is performed and the subject is in transition between two staircases. Therefore, a large number of these samples are because of the wrong annotations in the dataset rather than being wrong predictions of the network.

TABLE II: Results on RealWorld

	Precision	Recall	wF1 Score	Accuracy
DeepConvLSTM	92.83%	92.65%	92.63%	92.65%
PerceptionNet	94.78%	94.20%	94.27%	94.20%
ARC-Net	**96.08%**	**95.64%**	**95.67%**	**95.64%**

C. Number of Parameters

To investigate the effect of the number of parameters on the obtained metrics from each of the datasets, we reduced the encoder's number of parameters by 80% from 300k to 60k. The same training and testing splits were used to train the smaller network. The results from this experiment are presented in Table III. In the case of the RealWorld dataset, almost no changes are seen in the accuracy and wF1-score of the ARC-Net with smaller encoder size. This is while an almost 1% reduction in the same metrics are seen on the PAMAP2 dataset, leaving an improvement of about 1% over PerceptionNet. These results show that the main factor in the improvement over SOTA is the utilization of capsule networks rather than the increase in the number of parameters.

TABLE III: Effect of reducing the number of parameters

	RealWorld		PAMAP2	
	wF1	Accuracy	wF1	Accuracy
PerceptionNet	94.78%	94.20%	88.74%	88.56%
ARC-Net	**95.67%**	**95.64%**	**90.76%**	**90.51%**
ARC-Net Small	**95.67%**	**95.65%**	89.61%	89.35%

D. Prior Matrix Visualization

The prior matrix denoted by b in the routing mechanism of CapsNet is a learnable parameter and is also responsible for setting the values that describe the model's prior belief regarding the routing between two layers of capsules. Therefore, it is

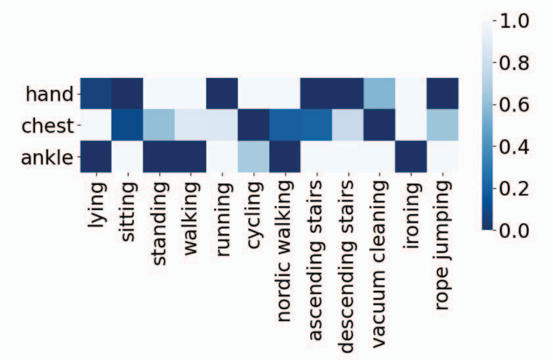


Fig. 5: Normalized prior matrix heatmap of the network trained on PAMAP2

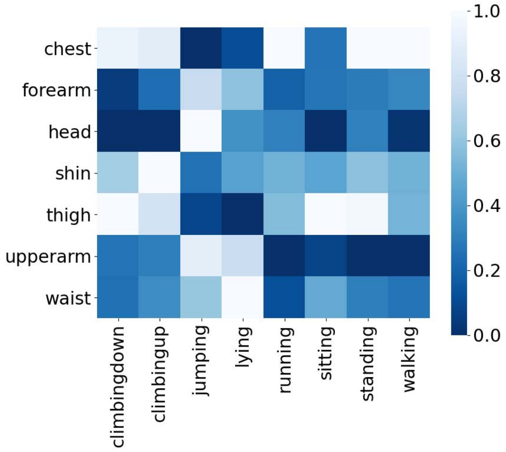


Fig. 6: Normalized prior matrix heatmap of the network trained on RealWorld

TABLE IV: Changes in Accuracy and F1-score with Modality Corruption

	PAMAP2		RealWorld	
	ΔwF1	ΔAcc	ΔwF1	ΔAcc
PerceptionNet	20.47%	18.63%	8.21%	8.1%
ARC-Net	**10.60%**	**10.93%**	**3.67%**	**3.66%**

possible to extract the values of this matrix after training and visualize the routing between each IMU and the performed activities. Fig. 5 and Fig. 6 visualize the prior matrix of our trained network on PAMAP2 and RealWorld, respectively.

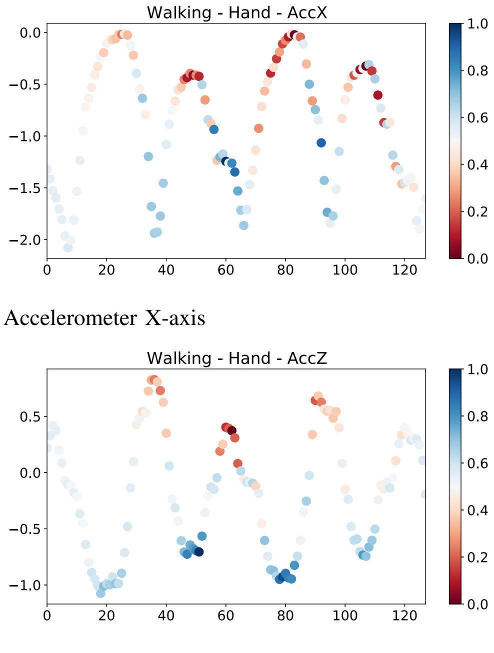
Based on Fig. 5, the routing that the network has come up with seems intuitive in activities such as the ones related to walking or climbing up and down the stairs where the network relies on the movement of the ankle or hand more than other modalities to make predictions. Moreover, correlations can be seen between the heatmaps of PAMAP2 and RealWorld for the same activities, e.g. the activity of lying relies on the measurements from the waist/chest when trained on either datasets. Based on Fig. 6, the addition of a thigh modality has allowed the standing and sitting activities to be more distinguishable with respect to PAMAP2. This has resulted in a substantial improvement in the directionality between the two classes in Fig. 4 with respect to Fig. 3.

E. Modality Corruption Test

One of the modalities was randomly corrupted by replacing its measurements with a zero matrix to simulate the potential case of modality failure during inference. The drops in the accuracy and weighted F1 scores of our method and PerceptionNet on the test sets of each dataset are reported in Table IV. It can be seen that on both datasets, our approach is significantly more robust against modality corruptions compared to a CNN only approach.

F. Gradient Based Interpretation

To visualize the patterns that are recognized by the network for a specific activity and visualize sensitivity of each class with respect to measurements from each IMU, we use guided backpropagation [23] through the Captum [24] library. Guided backpropagation calculates the gradient of a specific output with respect to the given inputs. To be able to get a better intuition of the detected patterns in the data, we provide visualizations corresponding to activities that, based on Fig. 5, rely, mostly, on one of the available IMUs from the PAMAP2 dataset. Based on Fig. 7, the network is able to detect the swing patterns of the hand during walking. The network ignores sections of the swing where the z-axis of the hand accelerometer nears the maximum of the periodic patterns seen in Fig. 7b. The same behaviour is seen in Fig. 7a. Moreover, the measurements that are towards the center of the frame, are more effective for the network in determining what the performed activity of that frame is. Based on Fig. 8a and Fig. 8b, we can see how the network is ignoring the z-axis of the ankle IMU while ascending stairs and focuses on the measurements provided by the x-axis of the same accelerometer.



(a) Accelerometer X-axis

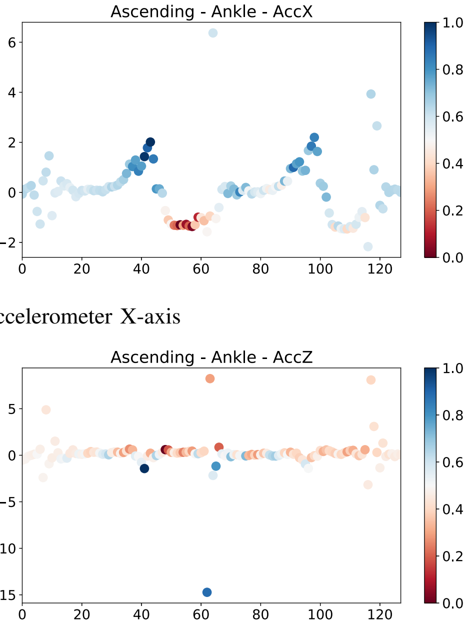
(b) Accelerometer Z-axis

Fig. 7: Guided backprop of hand IMU for ascending walking.

VI. Conclusions and Future Work

In this paper, we developed a method for human activity recognition that relies on CapsNet to fuse the information from multiple IMUs. We tested our approach on two datasets with varying sensor positions and compared our results against the SOTA. Our results surpassed that of SOTA by about

2% in accuracy and weighted F1 score. We also reduced the number of parameters in the encoder of the proposed network and showed that no drops were seen in the case of the RealWorld datset and only 1% decrease was observed on PAMAP2, leaving a 1% increase over the SOTA. This showed that the improvement was mainly due to the utilization of CapsNet rather than the increase in the number of parameters. Moreover, the confusion matrices of the test set of each dataset were presented and specificity of the classes was investigated. Our approach allowed for the visualization of routing between modalities and activities. Through these visualizations, we were able to interpret the importance of each modality for correct classification of each activity. On the other hand, gradient based interpretations allowed for visualization of the pattern recognition capability of the network. Based on both interpretation methods, it was evident that the network is able to detect the patterns present in the measurements from each modality which allows the network to selectively use or disregard each modality based on the data from all modalities and the prior belief achieved through training. Finally, modality corruption was simulated by passing an array of zeros instead of one random modality during inference and the capability of our method in noise rejection was shown. Additionally, we plan to investigate the effects of preintegration methods [25] to achieve even better efficiency.



(a) Accelerometer X-axis

(b) Accelerometer Z-axis

Fig. 8: Guided backprop of ankle IMU for ascending stairs.

References

[1] T. Poliero, L. Mancini, D. Caldwell, J. Ortiz, *Enhancing Back-Support Exoskeleton Versatility based on Human Activity Recognition*, Wearable Robotics Association Conference (WearRAcon), Scottsdale, AZ, USA, 2019, pp. 86-91.

[2] V. Osmani, S. Balasubramaniam, *Human activity recognition in pervasive health-care: Supporting efficient remote collaboration*, Journal of Network and Computer Applications, 31, 2008, 628-655.

[3] S. Chernbumroong, S. Cang, A. Atkins, H. Yu, *Elderly activities recognition and classification for applications in assisted living*, Expert Systems with Applications, 40, 2013, 1662-1674.

[4] R. Tripathi, A. Jalal, and S. Agrawal, *Suspicious human activity recognition: a review*, Artificial Intelligence Review, 2017, 50.

[5] Y., Du, Y., Lim and Y. Tan, *A Novel Human Activity Recognition and Prediction in Smart Home Based on Interaction*, Sensors 2019, 19, 4474.

[6] Md. Z. Alom, et al. *A State-of-the-Art Survey on Deep Learning Theory and Architectures*, Electronics. 8. 292., 2019.

[7] S. Sabour, N. Frosst, and G. Hinton, *Dynamic routing between capsules*, In Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17). Curran Associates Inc., Red Hook, NY, USA, 3859–3869.

[8] P. Kasnesis, C. Patrikakis and I. Venieris, *PerceptionNet: A Deep Convolutional Neural Network for Late Sensor Fusion*, Intelligent Systems and Applications. IntelliSys 2018. Advances in Intelligent Systems and Computing, vol 868.

[9] K. G. Manosha Chathuramali and R. Rodrigo, *Faster human activity recognition with SVM*, International Conference on Advances in ICT for Emerging Regions (ICTer2012), Colombo, 2012, pp. 197-203.

[10] L. Fan, Z. Wang and H. Wang, *Human Activity Recognition Model Based on Decision Tree*, 2013 International Conference on Advanced Cloud and Big Data, Nanjing, 2013, pp. 64-68.

[11] A. Varamin, E. Abbasnejad, Q. Shi, D. Ranasinghe, H. Rezatofighi, *Deep Auto-Set: A Deep Auto-Encoder-Set Network for Activity Recognition Using Wearables*, Proceedings of the 15th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services, 2018.

[12] Y. Li, D. Shi, B. Ding and D. Liu, *Unsupervised Feature Learning for Human Activity Recognition Using Smartphone Sensors*, Mining Intelligence and Knowledge Exploration. Lecture Notes in Computer Science, vol 8891.

[13] L. Fan, Z. Wang and H. Wang, *Human Activity Recognition Model Based on Decision Tree*, 2013 International Conference on Advanced Cloud and Big Data, Nanjing, 2013, pp. 64-68.

[14] N. Hammerla, S. Halloranm, T. Ploetz. *Deep, Convolutional, and Recurrent Models for Human Activity Recognition Using Wearables*, In Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI-16), 2016.

[15] D. Arifoglu and H. Bouchachia, *Activity Recognition and Abnormal Behaviour Detection with Recurrent Neural Networks*, Procedia Computer Science, 2017 110. 86-93.

[16] R. Charissa and C. Sung-Bae, *Human activity recognition with smartphone sensors using deep learning neural networks. Expert Systems with Applications*, 2016.

[17] A. Lin, J. Li, Z. Ma, *On Learning and Learned Representation with Dynamic Routing in Capsule Networks*, ArXiv abs/1810.04041, 2018

[18] A. Paszke, et al., *PyTorch: An Imperative Style, High-Performance Deep Learning Library*, Advances in Neural Information Processing Systems 32, 8024-8035, 2019.

[19] A. Reiss and D. Stricker, *Introducing a New Benchmarked Dataset for Activity Monitoring*, 2012 16th International Symposium on Wearable Computers, Newcastle, 2012, pp. 108-109.

[20] T. Sztyler and H. Stuckenschmidt, *On-body localization of wearable devices: An investigation of position-aware activity recognition*, 2016 IEEE International Conference on Pervasive Computing and Communications (PerCom), Sydney, NSW, 2016, pp. 1-9.

[21] T. Akiba, et al. *Optuna: A Next-generation Hyperparameter Optimization Framework*, In Proceedings of the 25th International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 2623–2631.

[22] J. Xie, B. Xu, Z. Chuang, *Horizontal and Vertical Ensemble with Deep Representation for Classification*, ArXiv abs/1306.2759, 2013.

[23] J.T. Springenberg, A. Dosovitskiy, T. Brox, M. Riedmiller, *Striving for Simplicity: The All Convolutional Net,* 2015 International Conference on Learning Representations.

[24] N. Kokhlikyan, et al., *PyTorch Captum,* 2019, [Online]. Available https://github.com/pytorch/captum.

[25] R. Khorrambakht, H. Damirchi, H. D. Taghirad, *Preintegrated IMU Features For Efficient Deep Inertial Odometry*, ArXiv abs/2007.02929, 2020.