PDRNet: A Deep-Learning Pedestrian Dead Reckoning Framework

Omri Asraf, Firas Shama, and Itzik Klein, *Senior Member, IEEE*

***Abstract*— Pedestrian dead reckoning is a well-known approach for indoor navigation. There, the smartphone's inertial sensors readings are used to determine the user position by utilizing empirical or bio-mechanical approaches and by direct integration. In this paper, we propose PDRNet, a deep-learning pedestrian dead reckoning framework, for user positioning. It includes a smartphone location recognition classification network followed by a change of heading and distance regression network. Experimental results using a publicly available dataset show that the proposed approach outperforms traditional approaches and other deep learning based ones.**

***Index Terms*— Pedestrian Dead Reckoning, Inertial Sensors, Indoor Navigation, Deep-Learning, Smartphone Location Recognition, Residual Networks.**

I. Introduction

OUTDOORS positioning of a pedestrian using a smartphone is usually done by using the global navigation satellite system (GNSS) receiver, installed within the smartphone. However, GNSS signals are not available indoors, thus positioning is based on other approaches, such as vision based [1], radio-frequencies signals [2] or inertial sensors [3]. Employing the latter, two main approaches exists: 1) pedestrian dead reckoning (PDR) [4] and 2) shoe mounted inertial navigation system (INS) [5].

The latter, requires installation of an INS on the shoe and means to communicate with the smartphone to provide the user position. In order to mitigate the navigation solution drift, zero velocity updates are used each time stationary conditions are identified (shoe on the ground) [6]. Although, this approach leverages from zero velocity updates, it requires installation of an additional INS on the shoe.

In PDR, user steps are detected using the accelerometers measurements, followed by a step length estimation process, based on empirical or bio-mechanical approaches [7]–[9]. By integrating the gyroscopes measurements directly [11], [12], or by a using gravity based approaches [10], the change in heading during the user step is also estimated. Thus, given the user initial conditions, step length and heading angle, the current user two-dimensional position can be calculated. The main advantage of using a PDR approach, based on the smartphone sensors, is that it doesn't require an additional set of sensors to be mounted on the shoe. Yet, the major drawback of the PDR approach is the requirement for gain calculation. In order to estimate the step-length in empirical or bio-mechanical approaches, a gain should be calculated prior PDR application. This gain remains constant throughout PDR operation. However, the gain was shown to be very sensitive to user dynamics [13], [14] and thus, an activity recognition (AR) model was suggested to identify the user dynamics and choose a proper gain. Later on, the smartphone location on the user was also shown to effect the PDR performance [15], [25]. To that end, in [25] a smartphone location recognition (SLR) feature based model was proposed. Recently, a deep-learning (DL) based robust SLR approach, which was evaluated on 107 people using four smartphone locations - talk, text, swing and pocket, was suggested [26].

To further improve PDR performance, more attention is given to machine learning (ML) approaches in PDR algorithms. Those works can be separated into three categories:

1) Direct calculation of the user velocity followed by an integration to obtain user position.
2) Step-length or heading estimation, based on machine learning approaches and/or combined with traditional PDR.
3) Both user position and heading estimation, based on machine learning approaches instead of traditional PDR.

One of the first works to apply ML algorithms was RIDI [18] - robust IMU double integration. There, the velocity vectors were regressed, using a vector regression model, in the smartphone coordinate frame, while relying on traditional sensor fusion methods to estimate device orientations. The regressed velocity, was used to correct the accelerometer measurements prior to double integration to obtain the user position. In [16] a DL approach was suggested to estimate the pedestrian velocity, while the smartphone is in the user pocket, instead of integrating the acceleration. Learning individual user patters while outdoors using GPS data as ground truth to train a deep learning model was suggested in [22]. Indoors, the trained model is used to regress the user velocity and traveled distance. Later, [32] continued their work in [16] using a recurrent neural network approach to estimate the velocity. Speed estimation using DL was

Date - January 11, 2021
O. Asraf, is with the University of Haifa, Haifa, Israel.
F. Shama, is with the University of Haifa, Haifa, Israel.
I. Klein, is with the Department of Marine Technologies, University of Haifa, Haifa, Israel (e-mail: kitzik@univ.haifa.ac.il).

suggested also by [17] in applications for strapdown inertial navigation.

In the second category, the heading determination and the step length estimation are separated, where one is estimated using traditional PDR algorithms and the other based on ML approaches. A long short-term memory (LSTM) model with denoising autoencoders architecture [21] was suggested to regress the user change in distance in each predefined time-window. In addition to the accelerometers and gyroscopes raw data, PDR features were added to the input of the network. Their proposed approach showed high accuracy, yet requires the pedestrians to hold their phone horizontally with their hand in front of their chest. The same authors follow up paper [23], focused on estimating the stride length while employing a SLR model before the regression process. Recently, [27], StepNet - a family of deep-learning based approaches to regress the step-length or change in distance was suggested. For heading determination, a network architecture based on spatial transformer networks and LSTM was suggested [24].

In the third category, the initial deep-learning based PDR approach, IONet [19], uses normalized accelerometers and gyroscopes measurements to regress the change in distance and heading in a predefined time window using an LSTM network. However, using normalized sensor reading may degrade the accuracy. Later, [20] employed, in addition to the accelerometers and gyroscopes measurements, also the device orientation. They examined several deep learning architectures to regress the user velocity in 2D. The velocity is then integrated to obtain the user position. In their approach, no SLR module was applied and the main difficulty was to estimate the user change in heading. Currently, this approach yields the best performance in the field.

In this paper, we leverage from [19], [26] and [20] and propose PDRNet - a deep learning PDR framework. As in [19], we use only accelerometers and gyroscopes measurements but instead of normalized values, raw sensor data is taken. From [26], we employ a smartphone location recognition network to distinguish between different smartphone locations. Additionally, following the successful use of ResNet [20] for deep-learning regression from IMU measurements in global coordinate frame to velocities, the ResNet network is also employed here, but with different inputs and outputs. In this work we used ResNet to map from raw IMU measurements to the change in distance and heading. The main contribution of the proposed approach is the definition of a modular structure framework to perform PDR using deep learning. Specifically, the basic building blocks, that should be applied in any similar approach, were defined. Those include the smartphone location recognition network, ResNet blocks in the regression model and definition of the input/output of the network. In addition, we show a strong connection between the change in distance and heading estimates. This leads to a conclusion that future research in the field should focus on the combined estimation instead of splitting the two. Also, by employing the RIDI dataset [18], we show that PDRNet obtains much better performance than classical PDR and the current state of the art [20] approaches.

The rest of the paper is organized as follows: Section II presents the traditional PDR approach. Section III gives the proposed PDRNet framework. Section IV presents method analysis and results while Section V gives the conclusions of the paper.

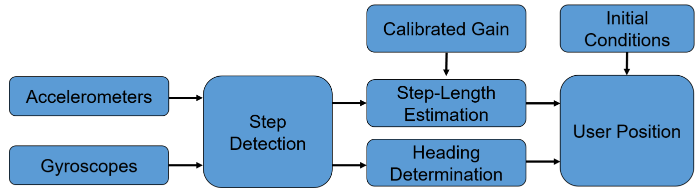


Fig. 1: Block diagram showing traditional pedestrian dead reckoning approach.

II. Traditional PDR

A block diagram showing a top-view block diagram of traditional PDR is presented in Figure 1. The input to the algorithm is the accelerometers and gyroscopes readings and the output is the pedestrian two-dimensional position. There, using the accelerometers readings, user steps are detected and, by using an empirical approach the user step-length can be estimated. The user change in heading is obtained using the gyroscopes measurements. Finally, the current pedestrian position is calculated given the heading change, step-length, and initial conditions.

A. Step-Length Estimation

Several approaches exists to estimate the step length, such as regression-based, bio-mechanical models or empirical relationships. One of the commonly used approaches is the Kim's approach [7]:

$$s = G\left(\frac{\sum_{j=1}^{N} f_j}{N}\right)^{1/3} \tag{1}$$

where G is a predefined gain, f_j is the specific force magnitude calculated based on the specific force vector at point j, and N is the number of samples in the step. The Kim-based step-length estimation is denoted by s.

Prior to PDR application, the gain G should be determined. The gain calibration procedure should to be made for each user according to the activity (walking slow/fast) and to the smartphone location (talking, pocket and etc.), since the gain is very sensitive to those quantities [25]. This gain sensitivity is one of the main disadvantage of the traditional PDR approaches.

B. Heading Determination

The pedestrian trajectory estimation requires an estimation of the user's direction/heading. The most common method for smartphone heading estimation is based on fusion of magnetometer and gyroscope sensors in an attitude and heading reference system framework [30], [31]. According to this approach, the gyro outputs are integrated through the attitude kinematic

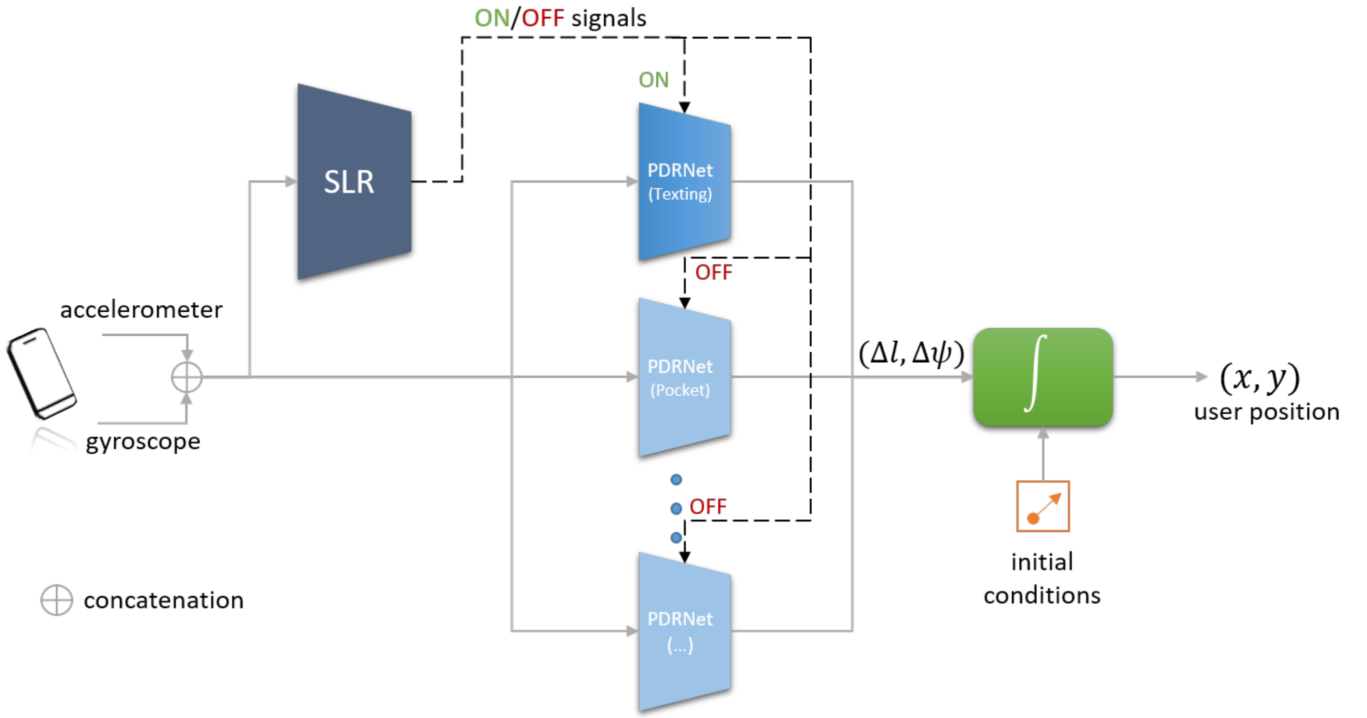


Fig. 2: PDRNet framework: the PDR problem is divided into two tasks: smartphone location recognition followed by user position determination. The SLR network predicts the smartphone location and activates only one of the dedicated PDRNet networks (one for each smartphone location), accordingly. The PDRNet infers changes in distance and heading which are then integrated to estimate the user's position.

equations and aided with magnetometer measurements.
In this paper only the gyroscope measurements are used to determine the user heading. Let the unit quaternion $\mathbf{q}_{ns} = q_w + q_x i + q_y j + q_z k$ denote the orientation of the sensor frame s relative to the navigation frame n. A three dimensional vector in frame n, $\mathbf{v}_n$ can be rotated by the quaternion $\mathbf{q}_{ns}$ as

$$\begin{bmatrix}\mathbf{v}_n \\ 1\end{bmatrix} = \mathbf{q}_{ns}^* \otimes \begin{bmatrix}\mathbf{v}_b \\ 1\end{bmatrix} \otimes \mathbf{q}_{ns} \tag{2}$$

where q^* is the conjugate of the quaternion q and the quaternion product, denoted by $\otimes$. Given the gyro measurements $\omega = [\omega_x, \omega_y, \omega_z]^T$ the following kinematic equation represents the rotation from sensor frame to navigation frame [31]:

$$\dot{\mathbf{q}} = \frac{1}{2}\begin{bmatrix} -q_x & -q_y & -q_z \\ q_w & -q_z & q_y \\ q_z & q_w & -q_x \\ -q_y & q_x & q_w \end{bmatrix}\begin{bmatrix}\omega_x \\ \omega_y \\ \omega_z\end{bmatrix}. \tag{3}$$

Next, by solving (3), the smartphone heading angle ψ^S is calculated

$$\psi^S = \operatorname{atan2}(2q_x q_y - 2q_w q_z, 2q_w^2 + 2q_x^2 - 1) \tag{4}$$

The user heading, ψ_U, is related to the smartphone heading by

$$\psi^U = \psi^S + \psi^{SU} \tag{5}$$

where ψ^{SU} is the relative heading between the smartphone and the user. At the beginning of the trajectory, the initial user heading is determined by the magnetometers readings, thus by substituting (4) into (5) the initial ψ^{SU} is determined. It is assumed that the smartphone is pointing to the user direction and held rigidly during the walking, thus ψ^{SU} remains constant during the trajectory. This, assumption is made to simplify the process and focus only on the proposed PDRNet approach. In practice, the difference between the user direction and the smartphone's heading can be calculated using principle component analysis (PCA) on the acceleration measurements [28], [29].

C. Position update

Given the pedestrian previous position $[x_{k-1}, y_{k-1}]^T$ at step $k-1$, the current step length s_k and heading ψ_k^U , the current user position is calculated by

$$\begin{bmatrix}x_k \\ y_k\end{bmatrix} = \begin{bmatrix}x_{k-1} + s_k \cos\psi_k^U \\ y_{k-1} + s_k \sin\psi_k^U\end{bmatrix} \tag{6}$$

III. PDRNet Framework

Inspired by latest state-of-the-art methods [19], [20], [27], we employ deep learning approaches to tackle the PDR problem. Motivated by the fact that every human action has its own motion characteristics, in the proposed approach the general PDR task is divided into two tasks: smartphone location recognition followed by user position determination. More specifically, a dedicated network is trained to recognize the current smartphone location based only on the inertial sensors readings. In addition, to each smartphone location a unique PDR deep networks is also trained to regress the user position. In contrast to traditional PDR, where the change in distance and heading is calculated according to the step times, the dedicated PDR networks predict the change in distance Δl and heading $\Delta\psi$ in a predefined constant time interval. Finally, the updated user position is given by:

$$\begin{bmatrix} x_k \\ y_k \end{bmatrix} = \begin{bmatrix} x_{k-1} + \Delta l \cos(\psi_{k-1} + \Delta\psi) \\ y_{k-1} + \Delta l \sin(\psi_{k-1} + \Delta\psi) \end{bmatrix} \tag{7}$$

We name this proposed solution, illustrated in Fig. 2, PDRNet.

A. SLR Network Architecture

Motivated by [26] a one-dimensional CNN architecture is employed for the SLR classification task. The input to the network is the raw accelerometers and gyroscopes measurements with the required window size, W, resulting in a $6 \times W$ matrix, as input. The output of the SLR network is the smartphone location: texting, pocket or body, where we refer to the smartphone location in texting activity as ”texting”, and to the smartphone location when attached to the chest bas ”body”. The SLR architecture consists of two consecutive CNN layers each with 64 units with a ReLU activation function. After dropout of 0.2 two consecutive CNN layers each with 32 units with an ReLU activation function are applied. Then, another dropout of 0.4 followed by a dense layer with 32 units and Softmax activation function. The loss function was categorical cross entropy (CCE) for a single label categorization and the optimization was performed with the RMS propagation algorithm (RMSProp).

B. PDRNet Architecture

Similar to [20] network, our dedicated PDR networks are one dimensional deep convolutional network, based on the popular Resnet architectures [33], that proved to be highly efficient and are easy to train due to the residual learning blocks that are stacked in the networks.

There are several variants of Resnet architectures. For example, Resnet-18 architecture stacks eight of residual blocks including convolution layers with kernel size of three. The proposed approach uses a Resnet-50 architecture which stacks 16 residual blocks. In both types of networks, the residual blocks are preceded with convolution layer of kernel size of seven, the stack of residual blocks are then followed by four fully connected layers which eventually produce the final prediction in the required output dimension.

Similar to SLR, PDR network gets an input of raw IMU measurements without any manipulation (such as noise reduction or normalization), and predicts the change in heading and distance. More precisely, our input $I \in \mathbb{R}^{6\times \mathrm{W}}$ is comprised of raw gyroscope and accelerometer readings, taken from a non-overlapping windows, each with a length of W. That is, the network provides an estimation every W samples. The network provides an output $O \in \mathbb{R}^2$ result of the change in distance in meters and heading in radians

$$O = \begin{bmatrix} \Delta l \\ \Delta\psi \end{bmatrix} = f_\theta(I) \tag{8}$$

The following objective function is used in the training process

$$L = \mu(\lambda) \cdot \Big(L_2(\Delta l, \Delta l^{GT}) + \lambda \cdot L_2(\Delta\psi, \Delta\psi^{GT}) \Big) \tag{9}$$

where $\Delta l, \Delta\psi$ indicate the distance and heading components of the output. respectively. The GT superscript indicates their ground truth counterparts. A parameter λ is essential to balance between the heading loss and distance loss, while the function μ is a normalization term given λ such that

$$\mu(\lambda) = \frac{1}{1+\lambda}. \tag{10}$$

We use L_2 losses as we experimentally found they provide better results than L_1. The optimization is done using Adam optimizer with learning rate of 10^{-4} and a scheduler with automatic learning rate reduction by 10. Depending on the dataset size, we train the networks for 200 to 350 epochs. Our baseline network is based on Resnet-18 architecture, trained with window size of $W = 200$,and batch size of $B = 128$, where batch size is the amount of randomly selected windows, used for single optimization step.

IV. Analysis and Results

In section IV-A, the dataset used to validate the proposed approach is presented, followed by the SLR network performance. Then, several experiments to optimize the training parameters, and propose further improvements on the training scheme (which lead eventually to state-of-the-art results) are described. Among them, hyper-parameter tuning for batch size B, window size W and evaluated on both Resnet-18 and Resnet-50 network architectures. When the dataset of a particular activity is not sufficiently big, we first pretrained the networks on the dataset of all activities. Furthermore, we balanced the weight λ of the objective and show that a multitasking network that predicts both change in distance and heading may perform better than a two separate networks.

As an evaluation metric, the average of the position error with respect to the ground truth is used

$$e = \frac{\sum_{i=1}^{N} \sqrt{(x_i^{GT} - x_i)^2 + (y_i^{GT} - y_i)^2}}{N}, \tag{11}$$

where N is the number of sampling windows in the trajectory.

A. Dataset

In order to evaluate the performance of the proposed approach the RIDI [18] dataset is employed. It includes IMU measurements and ground-truth of many trajectories divided into four categories of smartphone locations: pocket, texting, bag and body. The dataset was recorded with sampling rate of 200Hz with an overall of 150 minutes of data. We did not use the Bag category as we suspect consists some erroneous recordings.

The data of each category was divided to non-overlapping groups of train and test, such that the networks training is done only on the train group, while test group is used for evaluation only and it includes measurements of people not participating in the train dataset. The pocket mode includes recordings from

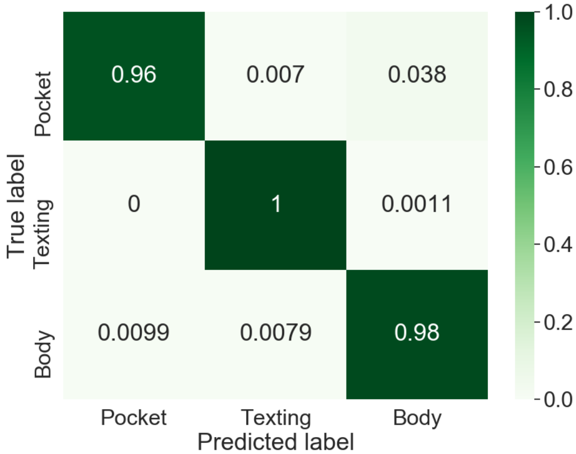


Fig. 3: Confusion matrix of smartphone location mode network on the test dataset.

five different people with a total of 10 different recordings. The texting mode includes recordings from 11 different people with a total of 22 different recordings while the body mode includes recordings from 8 different people with a total of 17 different recordings. Three different recordings in each mode were used as the test dataset.

B. SLR

The SLR classification network defined in III was trained and then implemented on the test dataset consisting of nine different recordings, three for each smartphone location. The overall training accurcy was 98.4% and the test accuracy was 97.6%. The resulting confusion matrix of the test dataset is presented in Fig. 3. There, all texting samples were accurately classified while pocket mode obtained 96% and body mode 98% of accuracy.

Next, we present several experiments in order to examine and improve the initial baseline network that's based on Resnet-18 network architecture and uses input window size $W = 200$, batch size $B = 128$ and objective balance value of $\lambda = 100$.

C. Hyper-parameter tuning

Results of the tuned Resnet-18 network after performing a grid-search over the hyper-parameters of batch size B and window size W are presented. In Fig. 4, the performance of the network on multiple values of batch size is compared. It can been seen, that training using a batch size of 32 achieves lower train and validation loss compared to the other values examined.

In order to tune the window size hyper-parameter, we choose to evaluate the overall performance of the track predictions, since comparing errors across different windows sizes may be misleading. Table I shows the position average error, calculated using (11), on the test set under different window sizes. A window size of 200 samples achieves the lowest error compared to other sizes. Given that the input sampling frequency is 200 Hz, each input window covers one second of sensors data. Thus, it may capture the complete dynamics of a pedestrian step, while smaller window sizes may capture only parts of the step dynamics, leading to a non structured input and a harder learning task. For this experiment the baseline network with a batch size of 32 was employed. Note that the results presented in Fig. 4 and Table I are from experiments done only with on texting mode. Although not presented, other modes show the same behaviour.

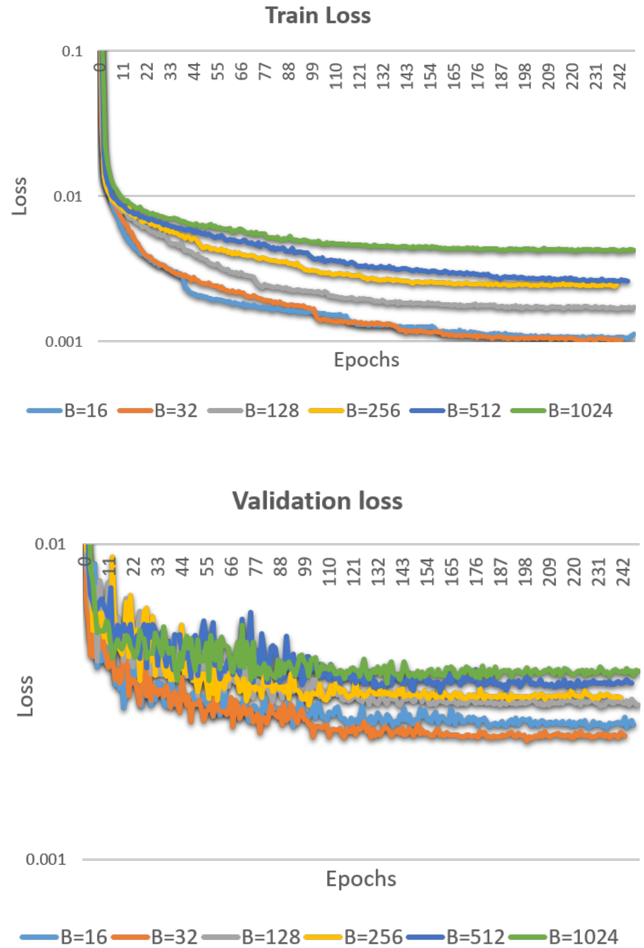


Fig. 4: Loss optimization curves for different batch sizes - above train and below validation.

TABLE I: Prediction error using several window sizes

Window size	50	100	200	300
Avg. Position Error [m]	9.29	2.57	**2.11**	2.2

D. Network Architecture

The use of a deeper network with higher complexity and number of parameters is examined. To that end, a network of ResNet-50 architecture, with same training process and hyper-parameters as presented earlier on ResNet-18, was trained. Indeed, ResNet-50 has a better learning curve, with faster convergence and final loss value. Interestingly, validation

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2021.3066840, IEEE Sensors Journal

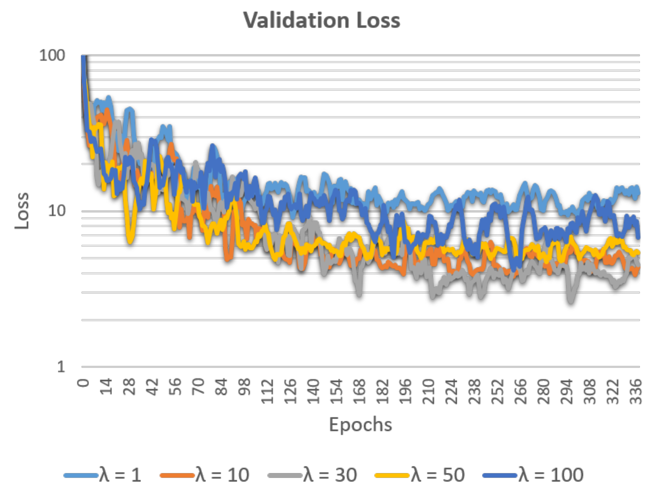


Fig. 5: Validation loss of the change in heading prediction during training with several λ values

loss is also improved, as a sign of non over-fitting model. Compared to our previous best result i.e. position average error of 2.11m, ResNet-50 achieved an average error of **1.76**m. Despite the longer run-time, higher memory, and being more prone to over-fitting, we focus on ResNet-50 as our network to regress the user position for further analysis.

E. Objective Balance and Multi-Tasking

According to the objective function (9), the network is optimized on two different tasks. Here, we perform several experiments by tuning the loss balance parameter λ in order to achieve lowest average position error. It is also shown that the predication of the change in heading, benefits from the task of distance change estimation. That is, in terms of performance, change in heading prediction achieves higher accuracy when trained in parallel with change in distance prediction. As observed in Fig. 5, networks trained with $\lambda = 30$ achieves lower validation loss and better generalization abilities for heading change prediction. Networks trained with lower λ values have less focus on heading change prediction, while networks trained with higher λ values lack the regularization that distance change loss may provide. Not only the best change in heading was achieved with $\lambda = 30$, but also overall performance improves. To show that the following error measures are employed

$$e_{\Delta l} = \frac{\sum_{i=1}^{N} \| \sum_{j=1}^{i} \Delta l_j - \sum_{j=1}^{i} \Delta l_j^{GT} \|}{N}, \qquad (12)$$

$$e_{\Delta \psi} = \frac{\sum_{i=1}^{N} \| \sum_{j=1}^{i} \Delta \psi_j - \sum_{j=1}^{i} \Delta \psi_j^{GT} \|}{N}. \qquad (13)$$

Those equations presents the average cumulative error (ACE) for the change in distance and heading. Table II summarizes the results obtained on the test dataset.

TABLE II: Networks tasks performance vs lambda

λ	ACE of dist[m]	ACE of angle[degree]	Avg. Pos. Error[m]
10^{-4}	**0.78**	99.19	42.42
1	1.05	9.5	3.0
10	1.33	4.15	2.09
30	1.16	**1.29**	**1.64**
50	5.43	3.0	1.97

F. Transfer learning

In this experiment we evaluate the use of pre-training, or the so-called transfer learning, when the data in hand seems insufficient. To demonstrate that, the pocket dataset is used as a final target data as it has the least amount of data. But. now, before training on it, first pre-train the network on the data from all categories. Especially for a small dataset, pre-training on bigger dataset sets a better starting point for training, with network parameters trained already on common input features and characteristics. In our case, a network trained only on pocket mode achieves average position error of $3.04m$ on the test set, while a network that was pretrained first on all the data and then fine-tuned on the pocket dataset, achieves lower error of $\mathbf{2.1m}$ on test set. That is, an improvement of 31% was obtained.

G. Comparison to Existing Approaches

The resulting tuned PDRNet network is compared to traditional PDR, and a state of the art deep learning based PDR method, RONIN to evaluate its performance. For PDRNet, three networks, one for each smartphone location mode was separately trained. According to the hyper-parameters optimization pretested in the previous subsections, the PDRNet parameters that chosen for three networks are: window size 200, batch size 32, L_2 loss, and loss weighed ratio of $\lambda = 30$. In addition, pocket mode network was pretrained on all the dataset, then on the target pocket dataset. The optimal traditional PDR is achieved by step length gain calibration based on the training set for each activity separately, while heading estimation is achieved by gyro integration approach. RONIN network's implementation is available from the published code [20], the network is trained on same RIDI trian-set but without separation of smartphone activities as suggested in [20]. Additionally the RONIN approach used Android's game rotation vector as device orientations (similarity to gyro integration) for rotate the IMU sensors to normalized coordinate frame before it feed-froward to the RONIN network. That is the device orientations are used as inputs in the RONIN approach and not used in our PDRNet approach.

All three approaches (PDRnet, RONIN and traditional PDR) are tested on three scenarios from each of the three smartphone location modes: texting, pocket and body. Fig. 6 shows three examples of trajectories, where the ground truth trajectory is colored in black, the traditional PDR estimation trajectory in green, the RONIN estimation trajectory in blue, and the PDRNet trajectory in red.

Table III summarizes the results for all test scenarios and shows the average of position error and normalized position

TABLE III: Position average error [m] (left side) and normalized error to length path [%] (right side) for the test dataset.

Scenarios	Length Path [m]	TPDR		RONIN		PDRNet	
Texting 1	63.2	6.16	9.75	1.7	2.69	1.9	3.01
Texting 2	66.6	3.92	5.89	1.64	2.46	0.97	1.46
Texting 3	138.8	4.96	3.57	6.52	4.7	1.11	0.8
Pocket 1	65.9	3.05	4.63	1.86	2.82	1.17	1.78
Pocket 2	147.8	5.05	3.42	1.07	0.72	2.37	1.6
Pocket 3	70	5.18	7.4	2.46	3.51	2.26	3.23
Body 1	49.5	2.15	4.34	1.78	3.6	0.49	0.99
Body 2	144.7	5.45	3.77	1.2	0.83	1.75	1.21
Body 3	66.4	4.52	6.81	0.95	1.43	1.24	1.87
AVG	**90.3**	**4.5**	**5.5**	**2.13**	**2.53**	**1.47**	**1.77**
STD	**38.2**	**1.18**	**2**	**1.61**	**1.26**	**0.59**	**0.79**

error of the three methods. There, PDRNet obtained the best performance compared to other methods. More specifically, in five out of nine test scenarios the proposed PDRNet method was preferable and in the total average of all cases PDRNet outperforms RONIN by 30%. In addition, it can be seen that PDRNet method achieved the lowest standard deviation (STD), which can give an indication about the uncertainty of our approach compared to the others. The analysis of the average result of each mode individually, showed that in the texting and body scenarios, PDRNet methods achieves the best performance, when in pocket scenarios RONIN method was preferable. Therefore, there is an advantage to the proposed modular structure of PDRNet which could be combined with other approaches in smartphone modes where they have a better performance and also when adding other smartphone locations.

V. Conclusion

A novel approach for deep learning based pedestrian dead reckoning, PDRNet, was proposed and evaluated. PDRNet includes a classification network for smartphone location recognition, and a regression network to obtain the change in distance and heading, to enabling the determination of the user position. The proposed approach was examined on scenarios with three different smartphone location modes - texting, pocket, and body. After hyper-parameter tuning and objective balance, the ResNet-50 with a window size of 200 samples, batch size 32, L_2 loss, and loss weighed ratio of $\lambda = 30$, obtained the best performance.

The PDRNet framework was then compared to traditional PDR approaches, and to RoNIN approach. Results on the testing dataset demonstrated that in average of all cases PDRNet achieved state-of-the-art performance compared to the other approaches with more than 67% improvement relative to traditional PDR, and 30% from RoNIN.

Besides the improvement in performance, the main contribution of the proposed approach is the modular structure of PDRNet. it allows to fit the best NN for each smartphone location, and offers the simplicity of adding other smartphone location modes. Also, compared to traditional PDR, this approach doesn't require any calibration process to each smartphone location. Compared to RoNIN, the device orientation is not used as input in PDRNet and also it is not needed to rotate the inputs to global frame as needed in RONIN.

In addition, a strong connection between the change in distance and heading estimates is demonstrated. That is, regression of both of them together is preferred then to regress them separately.

References

[1] W. Elloumi, A. Latoui, R. Canals, A. Chetouani and S. Treuillet, "Indoor Pedestrian Localization With a Smartphone: A Comparison of Inertial and Vision-Based Methods," in IEEE Sensors Journal, vol. 16, no. 13, pp. 5376-5388, July1, 2016, doi: 10.1109/JSEN.2016.2565899.

[2] Y. Zhuang, Y. Li, L. Qi, H. Lan, J. Yang and N. El-Sheimy, "A Two-Filter Integration of MEMS Sensors and WiFi Fingerprinting for Indoor Positioning," in IEEE Sensors Journal, vol. 16, no. 13, pp. 5125-5126, July1, 2016, doi: 10.1109/JSEN.2016.2567224.

[3] J. Nilsson, I. Skog, P. Händel and K. V. S. Hari, "Foot-mounted INS for everybody - an open-source embedded implementation," Proceedings of the 2012 IEEE/ION Position, Location and Navigation Symposium, Myrtle Beach, SC, 2012, pp. 140-145, doi: 10.1109/PLANS.2012.6236875.

[4] H. Zhang, W. Yuan, Q. Shen, T. Li and H. Chang, "A Handheld Inertial Pedestrian Navigation System With Accurate Step Modes and Device Poses Recognition," in IEEE Sensors Journal, vol. 15, no. 3, pp. 1421-1429, March 2015, doi: 10.1109/JSEN.2014.2363157.

[5] G. V. Prateek, R. Girisha, K. V. S. Hari and P. Händel, "Data Fusion of Dual Foot-Mounted INS to Reduce the Systematic Heading Drift," 2013 4th International Conference on Intelligent Systems, Modelling and Simulation, Bangkok, 2013, pp. 208-213, doi: 10.1109/ISMS.2013.46.

[6] S. Y. Cho and C. G. Park, "Threshold-less Zero-Velocity Detection Algorithm for Pedestrian Dead Reckoning," 2019 European Navigation Conference (ENC), Warsaw, Poland, 2019, pp. 1-5, doi: 10.1109/EURONAV.2019.8714170.

[7] J. W. Kim, H. J. Jang, D.-H. Hwang, and C. Park, "A step, stride and heading determination for the pedestrian navigation system," J. Global Positioning Syst., vol. 1, nos. 1–2, pp. 273–279, 2004.

[8] H. Weinberg, "Using the ADXL202 in pedometer and personal navigation applications," Analog Devices, Norwood, MA, USA, Appl. Note 602, 2002.

[9] J. Scarlet, "Enhancing the Performance of Pedometers Using a Single Accelerometer," Analog Devices AN-900 Application Note, 2005.

[10] Manos A, Klein I, Hazan T. Gravity-based methods for heading computation in pedestrian dead reckoning. Sensors. 2019 Jan;19(5):1170.

[11] P. D. Groves, Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems, 2nd ed. Norwood, MA, USA: Artech House, 2013.

[12] D. Titterton and J. Weston, emphStrapdown Inertial Navigation Technology (Radar, Sonar and Navigation) 2nd Edition, The Institution of Engineering and Technology, 2005.

[13] M. Elhoushi, J. Georgy, A. Noureldin, and M. J. Korenberg, *A survey on approaches of motion mode recognition using sensors*, IEEE Trans. Intell. Transp. Syst., vol. 18, no. 7, pp. 1662–1686, Jul. 2017.

[14] L. Qian, J. Ma, G. R. Yin, P. Liu, and P. Pei, *An improved indoor localization method using smartphone inertial sensors*, in Proc. Int. Conf. Indoor Positioning Indoor Navigat. (IPin), Oct. 2013, pp. 1–7.

[15] M. Zeng, L. T. Nguyen, B. Yu, O. J. Mengshoel, J. Zhu, P. Wu and J. Zhang *Convolutional Neural Networks for human activity recognition using mobile sensors*," 6th International Conference on Mobile Computing, Applications and Services, Austin, TX, 2014, pp. 197-205.

[16] T.Feigl, , S. Kram, , P.Woller, R.H. Siddiqui, M. Philippsen, and C. Mutschler, *A Bidirectional LSTM for Estimating Dynamic Human Velocities from a Single IMU*, In 2019 International Conference on Indoor Positioning and Indoor Navigation (IPIN) (pp. 1-8). IEEE, 2019.

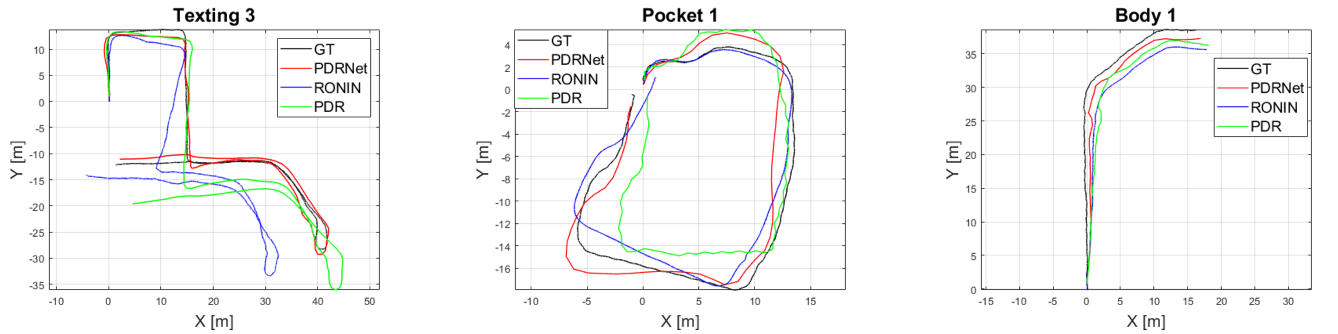
[17] S. Cortés, A. Solin and J. Kannala, *Deep learning based speed estimation for constraining strapdown inertial navigation on smartphones*, 2018 IEEE 28th International Workshop on Machine Learning for Signal Processing (MLSP), Aalborg, 2018, pp. 1-6.

[18] H. Yan, Q. Shan, and Y. Furukawa, *RIDI: Robust imu double integration*, In Proceedings of the European Conference on Computer Vision (ECCV), pp. 621–636, 2018.

[19] C. Chen, X. Lu, A. Markham, and N. Trigoni, *IONET: Learning to cure the curse of drift in inertial odometry*, In Thirty-Second AAAI Conference on Artificial Intelligence, 2018.

[20] Yan H, Herath S, Furukawa Y. RoNIN: Robust Neural Inertial Navigation in the Wild: Benchmark, Evaluations, and New Methods. arXiv preprint arXiv:1905.12853. 2019 May 30.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2021.3066840, IEEE Sensors Journal



(a) (b) (c)

Fig. 6: Three examples of position estimation comparison between the three methods: PDRNet (red line), RoNIN (blue line), and traditional PDR (green line). (a) Texting scenario. (b) Pocket scenario. (c) Body scenario.

[21] Q. Wang, H. Luo, L. Ye, A. Men, F. Zhao, Y. Huang, and C. Ou, *Pedestrian Heading Estimation Based on Spatial Transformer Networks and Hierarchical LSTM*, in IEEE Access, vol. 7, pp. 162309-162322, 2019.

[22] J. Kang, J. Lee and D-S. Eom, *Smartphone-Based Traveled Distance Estimation Using Individual Walking Patterns for Indoor Localization*, Sensors 18, 3149; doi:10.3390/s18093149, 2018.

[23] Q. Wang, L. Ye, H. Luo, A. Men, F. Zhao, and Y. Huang, *Pedestrian Stride-Length Estimation Based on LSTM and Denoising Autoencoders*. Sensors, 19, 840, 2019.

[24] Q. Wang, L. Ye, H. Luo, A. Men, F. Zhao, and C. Ou, *Pedestrian Walking Distance Estimation Based on Smartphone Mode Recognition*. Remote Sensing, 11, 1140; doi:10.3390/rs11091140, 2019.

[25] I. Klein, Y. Solaz, and G. Ohayon, *Pedestrian dead reckoning with smartphone mode recognition*, IEEE Sensors Journal, 18, pp. 7577–7584, 2018.

[26] I. Klein, *Smartphone Location Recognition: A Deep Learning-Based Approach*, Sensors, 20, 214; doi:10.3390/s20010214, 2020.

[27] I. Klein and O. Asraf, *StepNet—Deep Learning Approaches for Step Length Estimation*, in IEEE Access, vol. 8, pp. 85706-85713, 2020, doi: 10.1109/ACCESS.2020.2993534.

[28] Deng, Z.A., Wang, G., Hu, Y. and Wu, D., 2015. Heading estimation for indoor pedestrian navigation using a smartphone in the pocket. Sensors, 15(9), pp.21518-21536.

[29] Steinhoff, U.; Schiele, B. Dead reckoning from the pocket—An experimental study. In Proceedings of the IEEE International Conference on Pervasive Computing and Communications (PerCom), Mannheim, Germany, 29 March–2 April 2010; pp. 162–170.

[30] R. Mahony, T. Hamel and J. Pflimlin, "Nonlinear Complementary Filters on the Special Orthogonal Group," in IEEE Transactions on Automatic Control, vol. 53, no. 5, pp. 1203-1218, June 2008, doi: 10.1109/TAC.2008.923738.

[31] J. Farrell, Aided navigation: GPS with high rate sensors, McGraw-Hill, Inc., 2008.

[32] T.Feigl, , S. Kram, , P.Woller, R.H. Siddiqui, M. Philippsen, and C. Mutschler, *RNN-aided human velocity estimation from a single IMU* Sensors, 20(13), 3656, 2020.

[33] K. He, X. Zhang, S. Ren and J. Sun, "Deep residual learning for image recognition." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770-778. 2016.

FIRAS SHAMA received his B.Sc. and M.Sc degrees in electrical engineering from the Technion-Israel Institute of Technology, in 2014 and 2020 respectively. He is former worker of Intel Corporation and currently he is working at Huawei Tel-Aviv Research Center. His fields of interest include computer-vision, deep learning, image manipulation, mobile-mapping and localization.

ITZIK KLEIN (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees in aerospace engineering and the Ph.D. degree in civil engineering from the Technion-Israel Institute of Technology, in 2004, 2007, and 2011, respectively. He is currently an Assistant Professor with the University of Haifa, heading the Department of Marine Technologies' Autonomous Navigation and Sensor Fusion Lab. His research interests include navigation, novel inertial navigation architectures, autonomous underwater vehicles, sensor fusion, and estimation theory.

OMRI ASRAF received the B.Sc. in Mechanical Engineering from Ben-Gurion University and the M.Sc. in the department of Aerospace Engineering at the Technion- Israel Institute of Technology, in 2017 and 2020 respectively. Omri is working as a Navigation Algorithmic Researcher at Huawei Tel-Aviv Research Center. His research fields include navigation, SLAM, planning and deep learning.