# DeepCReg: Improving Cellular-based Outdoor Localization using CNN-based Regressors

**Conference Paper** · April 2020

# DeepCReg: Improving Cellular-based Outdoor Localization using CNN-based Regressors

Karim Elawaad
Alexandria University, Egypt
E-mail:kelawaad@gmail.com

Mohamed Ezzeldin
Alexandria University, Egypt
E-mail: not.muhammedezz@gmail.com

Marwan Torki
Alexandria University, Egypt
E-mail: mtorki@alexu.edu.eg

***Abstract*—In this paper, we propose DeepCReg, a convolutional neural network based regressor, that leverages the ubiquitous cellular data to estimate the location of the user in an outdoor environment. We formulate the problem of outdoor localization of a user as a regression problem. This formulation overcomes the limitations of other neural network based classification methods which estimates the position using a grid cell of pre-specified dimensions. We regress on the position directly which leads to better scalability when the testbed area is increased. Moreover, we introduce the usage of convolutional neural networks instead of fully connected neural networks to add more robustness to small changes in the environment.**

**We evaluate our system on two different datasets to emphasize on the scalability of our regression approach. The testbeds are of size 0.147 $km^2$ and 1.469 $km^2$. Our system achieves median localization error of 2.06*m* and 2.82*m* on each dataset respectively, outperforming current state-of-the-art outdoor cellular based systems by at least 877% improvement in the median localization error.**

***Keywords*—** Outdoor Localization, Neural Networks, CNN-based regressors

## I. Introduction

With the rise of outdoor location-based services such as location-based social networks [1], [2] and emergency systems [3], the need for robust and efficient localization systems have emerged. Currently GPS is the main method used to provide such services [4], however, GPS is power intensive [5], [6], requires to have direct line of sight with the satellites and is not supported by low-end phones which are still widely used in certain regions. For a localization system to be ubiquitous, it should not rely on sensors available only in high-end phones, should provide acceptable accuracy, should not consume a lot of battery power and should be relatively easy to deploy while being robust.

To overcome these issues, several approaches were proposed that leverage either embedded sensors in smartphones, WiFi signals, cellular networks or an ensemble of these methods.

Embedded sensors in smartphones (accelerometer, gyroscope, camera, etc.) have been used for outdoor localization systems [7]–[11] using dead-reckoning and have been shown to achieve good localization accuracy. This approach consumes significantly less power, however, these methods usually require continuous calibration, are more sensitive to changes in the environment and are only available in high-end phones.

Due to their ubiquity, several approaches were proposed that leverage cellular signals. One important advantage of these approaches is that they can serve low-end phones when deployed by the network operators. These approaches fall into one of two categories, *propagation models* or *fingerprint-based techniques*. Propagation model approach [12] offers low overhead deployment since it does not require calibration but it achieves low localization accuracy.

On the other hand, fingerprint-based techniques [13]–[17] can achieve very good localization accuracy, but with significant overhead, where it requires a surveyor to manually collect the received signal strength (RSS) in the area of interest, along with the location, which is usually obtained from the GPS, to build a fingerprint database. In the online phase, the collected database is used to infer the location of the user using deterministic techniques like K-Nearest Neighbor (KNN) [15], where the RSS vector in the online phase is compared to all the points in the fingerprint database and K nearest reference points according to some distance measure are averaged to find an estimate of the user's location.

The fingerprinting approach requires a surveyor to manually collect RSS samples at multiple points in the area of interest while standing for several minutes at each point. This adds a huge overhead in the data collection phase, limiting the scalability of this approach.

Another alternative proposed in [16] is superimposing a virtual grid on the area of interest, where instead of standing at an exact point for several minutes, the surveyor collects the RSS samples by freely walking in the grid cell, each grid cell is then identified by its geometric center, or more optimally by the center of mass of the points that fall in it, the problem then becomes a classification problem, where given the received signal strength from different cell towers, we wish to assign this sample a certain cell in the grid. The approach in [16] builds histograms of RSS values for each cell tower in each grid cell. A probabilistic model is then used to estimate the location of the user. A downside to this approach is that it assumes the independence of readings from different cell towers to reduce the sparsity in the data. This assumption simplifies the problem but discards the inherent correlation between readings from different cell towers. To alleviate this, [17] proposed using a Deep Neural Network (DNN) in a deep learning classification framework to capture this information. They propose using a DNN that takes as input an RSS vector and outputs a probability distribution over the grid cells.

### A. Motivation and Contributions

Previously discussed classification approaches achieve good results, however, by gridding the testbed to reduce the data collection overhead, they introduce a ground truth error in the collected data, as all the points in each grid cell are assigned to a single point, namely the center. One way to alleviate this is by decreasing the size of the grid cell, which leads to sparsity in the data and a significant increase in the size of the model, limiting their scalability to large testbeds.

To provide truly ubiquitous outdoor localization systems, the systems should not rely on sensors or chips that are only available in high-end phones and should scale well with large areas without a significant increase in the size of the model. Moreover, it is desirable that the system is robust to small changes in the environments.

This work proposes a deep convolutional neural network based regressor that leverages cellular signals while overcoming the limitations of the classification approach, with relative RSS readings in addition to absolute readings, to increase the robustness of our system to RSS variations caused by multi-path effect, shadowing and fading.

Our contributions are summarized as follows:

- We introduce the use of regression-based models to avoid introducing ground truth error while scaling well with larger testbeds.
- We use power normalized and absolute RSS values with convolutional neural networks to increase the robustness of the system and eliminate the need for preprocessing.
- We simplify the online phase by eliminating the need for post-processing the estimated grid cells as we directly regress on the position variables.
- We show a major improvement against competitive systems for outdoor localization using our *DeepCReg* system. The relative improvement in median localization error reaches up to 1400% for two different testbeds.

We implemented our system on two different devices and evaluated it in two different regions, one of area $0.147km^2$ and the other of area $1.469km^2$, achieving a median localization error of $2.06m$ and $2.82m$ respectively for each testbed.

The rest of the paper is organized as follows: Section 2 gives the background, Section 3 gives an overview of our system, Section 4 discusses the details of our approach, Section 5 presents the evaluation results, and the paper is concluded in Section 6.

## II. Approach

### A. Overview

*DeepCReg* works in two phases: *offline* phase and *online* phase. In the offline phase, cellular data is collected in the area of interest along with the GPS coordinates. In the online phase, the RSS readings on a user's cellphone are sent to a server. The data is passed to the previously trained model which directly predicts the latitude and longitude of the user.

### B. System Model

During the offline phase, a surveyor collects GPS-tagged RSS readings using their cellphone by freely walking or driving in the area of interest. The data collected is in the form $< [(CID_1, RSS_1), .., (CID_n, RSS_n)], \mathrm{L}_{GPS} >$, where CID is the cell tower ID of one of —*C*— cell towers, where *C* is the set of all cell towers heard in the collection process and *n* is the number of values received in that scan.$RSS_i$ is RSS value received from that cell tower identified by $CID_i$ and L is the coordinates of the user at the time the scan was taken. The collected data is then used to train and fine-tune our deep regressor model.

In the online phase during system operation, a user in the area of interest sends (CID, RSS) pairs to our server, which are then passed to a previously trained model and directly predicts the location of the user.

It is worth mentioning that unlike other localization systems [17], [18], our system provides an end-to-end system, directly predicting the location of the user given the RSS information, removing any need for post-processing, thus reducing the overhead of the online phase.

### C. Feature Extraction

To use the collected RSS information, we first have to change it to a suitable format for training our neural networks. Assume the number of cell towers heard in the offline collection phase is $C$, where each tower is identified by a unique cell tower ID (CID). For each collected sample $i$, we form the RSS vector $x_i$, where the value $x_{ij}$ corresponds to the RSS value from cell tower $j$. Since the testbed could be large, we may not have RSS values for some cell towers, so for those cell towers, we use the value $\lambda$ instead. For all our experiment we use $\lambda = -108dB$

For the classification approach, after overlaying the virtual grid $G$, the network takes as input the RSS vector $x_i$ and outputs a probability distribution over all the grid cells. To achieve this, the output layer of the network uses a Softmax activation function and the network is trained using cross-entropy loss. The cross-entropy loss requires that the labels are one-hot vectors, so we label each sample $x_i$ with the label $y_i$, where $y_i$ is the one-hot encoded vector with one at the grid cell that $x_i$ belongs to, and zero elsewhere.

For the regression approach the label $y_i$ is the normalized coordinates of the sample $x_i$.

### D. Channel Representation

The RSS from a given cell tower varies across time due to several reasons: multi-path effect, fading, weather effects, changes in the network topology, varying loads [19]. To provide a more robust performance, we use three representations created from the RSS vectors, namely: *Ratio Channel*, *Difference Channel* and *Tile Channel*. The intuition of relative power channels is to make use of the fact that the change in the RSS from one cell tower is often accompanied by changes in the RSSs from other cell towers, making the ratio/difference more robust to changes. Moreover, we present the raw RSS

| | | | |
|---|---|---|---|
| -108 | -71 | -91 | -53 |

(a) Original RSS vector

| | | | |
|---|---|---|---|
| 1 | 1.52 | 1.19 | 2.038 |
| 0.66 | 1 | 0.78 | 1.34 |
| 0.84 | 1.28 | 1 | 1.72 |
| 0.49 | 0.75 | 0.58 | 1 |

(b) Power ratio channel

| | | | |
|---|---|---|---|
| 0 | -37 | -17 | -55 |
| 37 | 0 | 20 | -18 |
| 17 | -20 | 0 | -38 |
| 55 | 18 | 38 | 0 |

(c) Power difference channel

| | | | |
|---|---|---|---|
| -108 | -71 | -91 | -53 |
| -108 | -71 | -91 | -53 |
| -108 | -71 | -91 | -53 |
| -108 | -71 | -91 | -53 |

(d) Tiled RSS channel

Fig. 1: Different Channels: Three channels extracted from the original RSS vector. The power ratio and power difference channels compute relative quantities. The tile channel tiles the original RSS vector to match the dimensions of the power ratio and power difference channels.

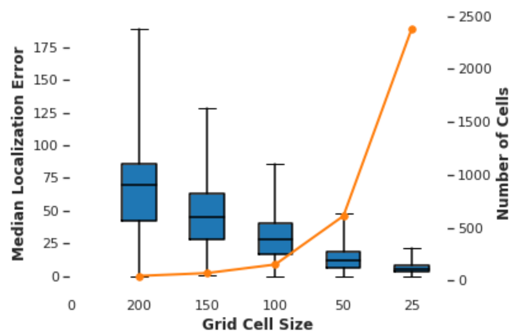


Fig. 2: Ground-truth error (meters) introduced by gridding the testbed, and number of grid cells for different cell sizes

information to the network in the *Tile Channel* as not to lose any information.

**Ratio Channel**

Similar to the approach used in [20], we use relative RSS values instead of the absolute values. For each sample $\mathbf{x} \in R^{|C|}$ we form a 2D matrix $R_{ratio} \in \mathbb{R}^{|C|\times|C|}$
, where $R_{ratio_{ij}} = x_i/x_j$.

**Difference Channel**

Another channel that leverages the concept of relative power is the difference channel which stores the difference in RSS values between different cell towers. For each training sample $\mathbf{x} \in R^{|C|}$, we form $R_{diff}$, where $R_{diff_{ij}} = x_i - x_j$.

**Tile Channel**

In order to avoid losing any information, we retain the absolute RSS values by tiling the vector $\mathbf{x}$ —C— times to form $R_{tile}$.

For each training sample $\mathbf{x}$ we form the corresponding three channels which are then used in training the CNN. An example of the three channels is shown in Figure 1.

### E. Classification Approach

The problem of outdoor localization is usually solved using classification methods. A virtual grid is overlaid on the area of interest, where all cells in the grid are of a predefined size, which is usually a hyperparameter that needs to be set to trade-off between the complexity of the employed model and the performance of the system. Each cell in the grid is represented by a point, which could be the geometric center of the cell or the average of coordinates of the points that belong to the cell. A neural network then takes as input the RSS vector and outputs a probability distribution over all the grid cells, where the predicted location is said to be the center of mass of the grid cell with the highest predicted probability.

However, because of the gridding, the points in each grid cell are represented by their respective center of mass, introducing an inherent ground truth error. This ground truth error gives a soft lower bound on the localization error that can be achieved by a classifier. To clarify, this the lowest localization error that can be achieved given a classifier that achieves a 100% classification accuracy, where this error increases proportionally as the grid cell size increases.

Figure 2 shows the computed ground truth errors for the two testbeds we used in our experiments. Since we use similar grid sizes, it is expected that we see very close ground truth errors for the two testbeds. The number of cells grows quadratically with the size of the grid cell. For a grid cell of $200m \times 200m$ area, the number of cells in **Testbed 1** is 4 cells. But for a grid cell of $25m \times 25m$ area, the number of cells grows to be 2380 cells.

One way to reduce the effect of this error is to decrease the size of the grid cell, which in turn will lead to sparsity in the data and increase the size of the output layer of the model, making the model more prone to overfitting while also requiring more computational resources for training.

Moreover, systems that employ the gridding approach offer coarse grained accuracy since their prediction can only be one of the centers of the grids.

### F. Regression Approach

As a more intuitive approach, we propose solving the problem of localization using regression. Namely, we regress directly over the latitude and longitude of the user. In particular, we use a deep learning model that takes as input the RSS information and predicts the coordinates of the user. Figure 3 shows an example of the architecture we use.

The advantages of our approach are twofold. First, we avoid introducing ground truth error caused by gridding the testbed. Second, since we predict directly the coordinates of the user, the size of the output layer is fixed regardless of the size of the testbed and the predictions are fine-grained compared to the classification approach.

In order to train the deep learning model, we first normalize the coordinates to be in the range [0,1], we then train the network using the RSS information and the normalized coordinates. In the online phase, we de-normalize the network's output to get the predicted location of the user.

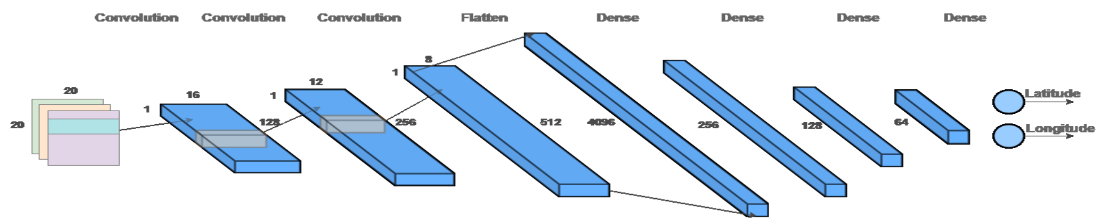


Fig. 3: *DeepCReg* Architecture: The input layer corresponds to the three features channels (Ratio, Difference and Tile): The output layer corresponds to the *longitude* and *latitude* variables.

## III. Performance Evaluation

In this section, we describe the two testbeds used in our experiments. Next, we show a comprehensive set of results using different alternatives such as classification and regression approaches with different feature sets. Then, we position our system against competitive systems. Finally, we show a discussion of the results of different settings. [1]

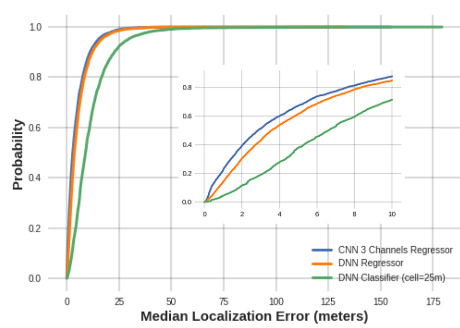


Fig. 4: Localization Error CDF: Localization error CDF of CNN regressor with three channels against DNN regressor and DNN classifier with grid cell size 25m on the Large testbed.

### A. Testbeds

We start off by describing both our testbeds. Two datasets were used to evaluate our regression-based approach against state-of-the-art existing approaches. Each dataset(fingerprint) belongs to a different testbed.

**Testbed 1** (small) is of dimensions $369m \times 399m$ spanning an area of $0.147km^2$. It consists of 12980 samples with 20 unique cell towers, so each sample $x_i \in \mathbb{R}^{|C|}$, where $|C| = 20$.

**Testbed 2** (large) We found our approach to outperform current state-of-the-art without the need for any form of pre/post-processing, so we collected a new dataset on a bigger area in a different location to test our approach for location invariance and scalability. **Testbed 2** is of dimensions $1689m \times 870m$ spanning an area of $1.469km^2$. It consists of 89328 samples with 58 unique cell towers, so each sample $x_i \in \mathbb{R}^{|C|}$, where $|C| = 58$.

[1]Code is available on GitHub. https://github.com/kelawaad/DeepCReg

### B. Results

In this section, we report our results and contrast it with the previous methods. We present the results of using our proposed CNN with the three channels and each channel separately. We also report the details of the training process and the models' architectures. All the results are summarized in Table II. It is worth noting that these results are not absolute but rather relative to the GPS since the training samples are labeled using the coordinates obtained from the GPS.

Before training, we split each dataset randomly into train, validation and test sets with ratios 0.7, 0.1, 0.2 respectively. We train our models using the training set, choose the best set of hyperparameters (number of hidden layers, size of filters, initial learning rate, etc.) using the validation set, and finally report the results on the test set.

**Classification**. For the classification approach, we quantify the effect of changing the grid cell size and the effect of power normalized channels on the localization accuracy. The DNN architecture used for all the experiments consisted of three hidden layers of size 256, 128, 256 nodes respectively. The CNN architecture consists of three convolutional layers of filter sizes —C—x3, 1x7, 1x9 respectively, followed by three fully connected layers of size 256, 128, 64 nodes. The output layer's size in both networks varies depending on the grid cell size used and the are of the testbed.

**Regression**. For the regression approach, we quantify the effect of the power normalized channel and contrast the results of the CNN with the DNN.

The DNN architecture is the same architecture used in the classification networks, except the output layer consists only of two nodes, corresponding to the latitude and longitude.

The CNN architecture consists of three convolutional layers of filter sizes —C—x5, 1x5, 1x5 respectively, followed by three fully connected layers of size 256, 128, 64 nodes and the output layer consists of two nodes as well.

All the hidden layers were followed by a ReLU activation function while the output layer was followed by a sigmoid activation. The models were trained using mean squared error between the predicted values and the ground truth as its loss function.

In all our experiments, we use *Adam* optimizer [21] with an initial learning rate of $10^{-4}$, batch size of 64 batches, and step learning rate scheduler which decreases the learning rate by a factor of 10 every fixed number of epochs.

All the training parameters for each testbed are summarized in Table I.

| | **Testbed 1** | **Testbed 2** |
|---|---|---|
| Initial learning rate (lr) | $10^{-4}$ | $10^{-4}$ |
| Epochs | 500 | 150 |
| Optimizer | Adam | Adam |
| Batch size | 64 | 64 |
| Decrease lr every | 100 epoch | 25 epoch |

TABLE I: Training parameters

*DeepCReg* CDF is shown in Figure 4. We contrast it against two other approaches. Namely, the classification approach with grid cell size = $25m \times 25m$, and a regression approach using the original RSS vector and a DNN architecture. The comparison shows that *DeepCReg* outperforms the classification approach with a large margin. Moreover, the CNN based architecture for regression (*DeepCReg*) outperforms the DNN architecture. This confirms our hypothesis about using power normalized channels against the original RSS vector.

### C. Comparative Evaluation

We position the localization accuracy of our system *DeepCReg* against DeepLoc [17] and CellSense [16] on the two testbeds. Both systems were evaluated using their best reported parameters. For DeepLoc, we did not perform any data augmentation or post-processing on the output.

**Localization Accuracy**
Figure 5 shows the localization error CDF of *DeepCReg* against other competitive systems. The figure shows that our system (*DeepCReg*) outperforms other systems with large margin.

Table III show the localization error of the three systems at different percentiles. *DeepCReg* achieves a better median localization error of 2.06m and 2.88m on **Testbed 1** and **Testbed 2** respectively, outperforming DeepLoc by at least 877% and CellSense by 981%. Since both DeepLoc and CellSense employ the gridding approach, their performance is bounded by the ground truth error. On the other hand, *DeepCReg*, regresses directly over the longitude and latitude.

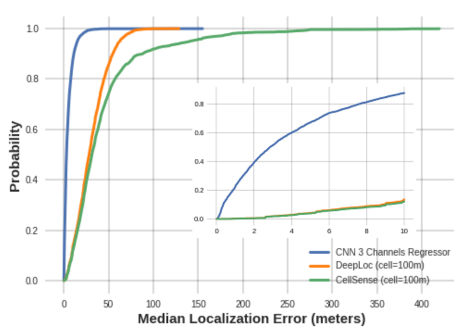


Fig. 5: Localization Error CDF of *DeepCReg* using three channels regressor, DeepLoc [17] and CellSense [16] with grid cell size 100m on the Large testbed.

### D. Discussion

The obtained results can shed light on the advantages of *DeepCReg* in various aspects.

**First**, *DeepCReg*'s motivation was to overcome the drawbacks and unscalability of previous methods that leverage classification techniques. *DeepCReg* achieves this by regressing directly over the latitude and longitude without compromising the localization accuracy as shown in Figure 4. This can be contrasted against the gridding based classification systems such as [16], [17]. Even though the classifier can learn to classify every testing sample in its correct grid cell, it just reports the center of mass of the grid cell as an estimated location. Figure 6 shows that all colored samples will be given the same label and hence the same location.

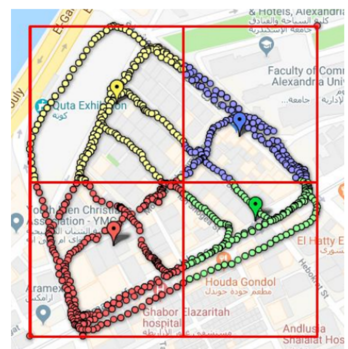


Fig. 6: Estimated labels and locations using gridding approach is bound to an error on **Testbed 1**.

**Second**, as opposed to other methods, *DeepCReg* is an end-to-end system, it performs well even with no data augmentation techniques. Also, no post-processing modules are applied to improve its performance and robustness.

**Third**, the learned models using the *DeepCReg* are more compact than those that use the classification methods. In classification methods, the size of the output layer is determined by the number of grid cells. This number of grid cells grows quadratically with the dimensions of the cell. On the other hand, the output layer of the *DeepCReg* has only two output nodes corresponding to the longitude and latitude.

## IV. Conclusion

In this paper, we proposed *DeepCReg*, an end-to-end cellular-based outdoor localization system. We showed the drawbacks of the currently employed classification techniques, namely introducing ground-truth error, less effective scalability with larger testbeds and that it offers coarse-grained accuracy.*DeepCReg* solves these issues by formulating the problem as a regression problem, where it scales well with larger testbeds and offers fine-grained accuracy, while also outperforming classification-based techniques by a large margin.

We also showed how *DeepCReg* leverages absolute and relative RSS values to increase its robustness to changes in the environment.

| Classification | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | **Testbed 1** | | | | | **Testbed 2** | | | | |
| | **CNN** | | | | **DNN** | **CNN** | | | | **DNN** |
| Size | 3 Channels | Ratio | Difference | Tile | RSS | 3 Channels | Ratio | Difference | Tile | RSS |
| 200m | 67.302 | 67.302 | 67.598 | 67.302 | 79.672 | 69.428 | 69.255 | 69.279 | 69.292 | **69.085** |
| 100m | 29.467 | **29.339** | 29.697 | 29.436 | 32.639 | 28.191 | 28.146 | 28.220 | 28.146 | **28.128** |
| 50m | 12.077 | 12.029 | 12.077 | **11.877** | 28.682 | 12.276 | 12.301 | 12.276 | 12.276 | 12.348 |
| 25m | 6.252 | 6.516 | 6.538 | **6.158** | 6.574 | **6.510** | 6.592 | 6.539 | 6.546 | 6.614 |

| Regression | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | **Testbed 1** | | | | | **Testbed 2** | | | | |
| | **CNN** | | | | **DNN** | **CNN** | | | | **DNN** |
| Size | 3 Channels | Ratio | Difference | Tile | RSS | 3 Channels | Ratio | Difference | Tile | RSS |
| - | **2.062** | 3.649 | 2.083 | 2.333 | 2.372 | 2.844 | 3.570 | **2.824** | 3.002 | 3.580 |

TABLE II: Median localization error (meters) for DNN and CNN with different channel configurations for different grid cell sizes. **Top:** Classification results. **Bottom:** Regression results.

| **Testbed 1** | | | |
|---|---|---|---|
| | $25^{th}$ | Median | $75^{th}$ |
| DeepCReg | **0.77** | **2.06** | **4.16** |
| DeepLoc | 17.7(-2183%) | 30.6(-1382%) | 44.7(-974%) |
| CellSense | 17.4(-2153%) | 32.0(-1450%) | 52.8(-1169%) |
| **Testbed 2** | | | |
| | $25^{th}$ | Median | $75^{th}$ |
| DeepCReg | **1.13** | **2.88** | **6.42** |
| DeepLoc | 16.9(-1390%) | 28.2(-877%) | 41.3(-543%) |
| CellSense | 18.2(-1512%) | 31.2(-982%) | 50.8(-691%) |

TABLE III: Comparison between *DeepCReg*, and DeepLoc [17], CellSense [16] classifiers with grid cell size 100m. Number in parenthesis corresponds to the relative improvement achieved by *DeepCReg* over competitive systems.

We implemented our system on two different testbeds of sizes 0.147 $km^2$, 1.469 $km^2$ and it achieved a median localization error of 2.06m and 2.82m for each testbed respectively, outperforming previous approaches by at least 877%.

## V. Acknowledgment

This work is supported in part by the Egyptian National Telecommunication Regulatory Authority (NTRA).

## References

[1] Y. Zheng, Y. Chen, X. Xie, and W.-Y. Ma, "Geolife2. 0: a location-based social networking service," in *2009 tenth international conference on mobile data management: systems, services and middleware*. IEEE, 2009, pp. 357–358.

[2] P. Shankar, Yun-Wu Huang, P. Castro, B. Nath, and L. Iftode, "Crowds replace experts: Building better location-based services using mobile social network interactions," in *2012 IEEE International Conference on Pervasive Computing and Communications*, March 2012, pp. 20–29.

[3] W. Bejuri, M. M. Mohamad, R. Zahilah, and R. M. Radzi, "Emergency rescue localization (erl) using gps, wireless lan and camera," *Int J Softw Eng Appl*, vol. 9, no. 9, pp. 217–232, 2015.

[4] J. Schiller and A. Voisard, *Location-based services*. Elsevier, 2004.

[5] A. Carroll, G. Heiser *et al.*, "An analysis of power consumption in a smartphone." in *USENIX annual technical conference*, vol. 14. Boston, MA, 2010, pp. 21–21.

[6] T. O. Oshin, S. Poslad, and A. Ma, "Improving the energy-efficiency of gps based location sensing smartphone applications," in *2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications*. IEEE, 2012, pp. 1698–1705.

[7] H. Aly and M. Youssef, "Dejavu: an accurate energy-efficient outdoor localization system," in *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 2013, pp. 154–163.

[8] I. Constandache, R. R. Choudhury, and I. Rhee, "Compacc: Using mobile phone compasses and accelerometers for localization," in *IEEE INFOCOM*. Citeseer, 2010, pp. 1–9.

[9] A. Irschara, C. Zach, J.-M. Frahm, and H. Bischof, "From structure-from-motion point clouds to fast location recognition," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009, pp. 2599–2606.

[10] M. Youssef, M. A. Yosef, and M. El-Derini, "Gac: energy-efficient hybrid gps-accelerometer-compass gsm localization," in *2010 IEEE Global Telecommunications Conference GLOBECOM 2010*. IEEE, 2010, pp. 1–5.

[11] M. Azizyan, I. Constandache, and R. Roy Choudhury, "Surroundsense: Mobile phone localization via ambience fingerprinting," in *Proceedings of the 15th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '09. New York, NY, USA: ACM, 2009, pp. 261–272. [Online]. Available: http://doi.acm.org/10.1145/1614320.1614350

[12] R. Elbakly and M. Youssef, "Crescendo: An infrastructure-free ubiquitous cellular network-based localization system," *arXiv preprint arXiv:1901.10214*, 2019.

[13] J. Paek, K.-H. Kim, J. P. Singh, and R. Govindan, "Energy-efficient positioning for smartphones using cell-id sequence matching," in *Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys '11, 2011.

[14] Q. D. Vo and P. De, "A survey of fingerprint-based outdoor localization," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 491–506, 2015.

[15] M. Y. Chen, T. Sohn, D. Chmelev, D. Haehnel, J. Hightower, J. Hughes, A. LaMarca, F. Potter, I. Smith, and A. Varshavsky, "Practical metropolitan-scale positioning for gsm phones," in *International Conference on Ubiquitous Computing*. Springer, 2006, pp. 225–242.

[16] M. Ibrahim and M. Youssef, "Cellsense: An accurate energy-efficient gsm positioning system," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 1, pp. 286–296, 2012.

[17] A. Shokry, M. Torki, and M. Youssef, "Deeploc: a ubiquitous accurate and low-overhead outdoor cellular localization system," in *Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 2018, pp. 339–348.

[18] K. Wu, J. Xiao, Y. Yi, D. Chen, X. Luo, and L. M. Ni, "Csi-based indoor localization," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 7, pp. 1300–1309, 2013.

[19] G. Micallef, P. Mogensen, and H.-O. Scheck, "Cell size breathing and possibilities to introduce cell sleep mode," in *2010 European Wireless Conference (EW)*. IEEE, 2010, pp. 111–115.

[20] M. Ibrahim and M. Youssef, "Enabling wide deployment of gsm localization over heterogeneous phones," in *2013 IEEE International Conference on Communications (ICC)*. IEEE, 2013, pp. 6396–6400.

[21] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.