Deep Learning-based Localization in Limited Data Regimes

Frost Mitchell
frost.mitchell@utah.edu
University of Utah
Salt Lake City, Utah

Aniqua Baset
University of Utah
Salt Lake City, Utah

Neal Patwari
Washington University in St. Louis
St. Louis, Missouri

Sneha Kasera
University of Utah
Salt Lake City, Utah

Aditya Bhaskara
University of Utah
Salt Lake City, Utah

ABSTRACT

As demand for radio spectrum increases with the widespread use of wireless devices, effective spectrum allocation requires more flexibility in terms of time, space, and frequency. In order to protect users in next-generation wireless networks from interference, spectrum managers must have the ability to efficiently and accurately locate transmitters. We present **TL;DL**, a practical deep-learning based technique for multiple transmitter localization on crowdsourced data where all sensors and transmitters may be mobile and transmit with unknown power. We map sensor readings to an image representing the sensor location, then use a convolutional neural network to learn to generate a target image of transmitter locations. We also introduce a novel data-augmentation technique to drastically improve generalization and enable accurate localization on limited data. In our evaluation, **TL;DL** outperforms previous approaches on small real-world datasets with low sensor density, in terms of both accuracy and detection.

CCS CONCEPTS

• **Networks → Location based services**; *Network manageability*; *Mobile and wireless security*.

KEYWORDS

localization, data augmentation, RF sensing, deep learning

ACM Reference Format:
Frost Mitchell, Aniqua Baset, Neal Patwari, Sneha Kasera, and Aditya Bhaskara. 2022. Deep Learning-based Localization in Limited Data Regimes. In *Proceedings of the 2022 ACM Workshop on Wireless Security and Machine Learning (WiseML '22), May 19, 2022, San Antonio, TX, USA*. ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3522783.3529529

1 INTRODUCTION

Due to high spectrum demand in both urban and rural areas, sharing spectrum is an essential component of next-generation wireless networks. Although spectrum sharing between licensed and unlicensed users is a defining characteristic of cognitive radio networks and other proposed network systems, the increasing affordability of software-defined radios (SDR) presents a convenient method for users to transmit without authorization or at power levels beyond what they have been authorized. Consider a situation where these SDR users transmit without regard for the interference caused to other users. These illegal transmissions could be accidental, or could be caused by device malware or malicious user behavior.

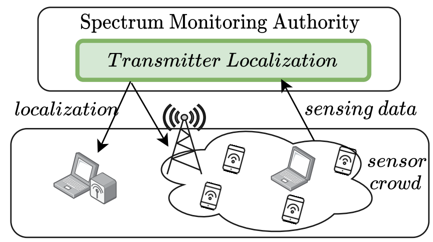

Figure 1: Spectrum monitoring and localization model

Current methods for detecting spectrum violations at the U.S. Federal Communications Commission (FCC) enforcement bureau involve complaints and extensive manual investigation, which can take months or years to identify illegal signals and resume normal service due to the mobility of violators and the human-in-the-loop. In one case from 2014, a disgruntled Florida driver regularly jammed cellular communication for over two years before cellular companies reported patterns of interference and FCC enforcement officials were able to locate the jamming device [12].

In a practical setting, we envision an automatic monitoring system where spectrum management authorities use crowdsourced data for sensing and localizing offending transmitters, shown in Figure 1, in order to prevent spectrum violations and minimize harmful interference to authorized users. A "sensor crowd" made up of various commodity wireless devices would sense the spectrum and report the sensing data to a central monitoring authority, which could use a localization algorithm to locate unauthorized transmitters. Due to the use of commodity hardware to source spectrum measurements, we propose using received signal strength indicator (RSSI) measurements for localization, as opposed to measurements such as *angle-of-arrival*, which requires specialized hardware, or *time-difference-of-arrival*, which would involve sharing recorded signals with a central user, introducing serious privacy concerns. Konings et al. [4] note that raw RSSI values are not an accurate

ACM ISBN 978-1-4503-9277-8/22/05…$15.00
https://doi.org/10.1145/3522783.3529529

measurement of true RSS values, but that with some calibration it may be possible to use raw RSSI values for localization. Our findings in this work, as well as those in [2, 9] support this idea.

RSS-based localization has been investigated for both single transmitter [5, 13] and multi-transmitter scenarios [2, 3]. These traditional methods rely solely on given RSS measurements and an assumed RF propagation model. This makes them error-prone when the assumed propagation model cannot capture the complex propagation characteristics of a real-world environment. To alleviate this issue, recent localization techniques [9, 14–16] use machine learning with environment-specific training data. Several existing studies [9, 14] demonstrate the advantage of learning-based techniques over non-learning localization methods.

Machine learning for localization has also been studied extensively in the context of fingerprint-based indoor localization or positioning [6, 11]. However, these methods require either the transmitters or the receivers to be at fixed locations. Thus these solutions cannot handle our setting of interest, where we allow unrestricted mobility for both transmitters and receivers.

Many learning-based localization methods still have high rates of *misdetection* (missing certain transmitters) and *false alarm* (detecting more transmitters than are actually present), to the degree that an automatic system for monitoring spectrum remains infeasible. The learning-based technique LLOCUS interpolates RSS measurements from mobile sensors to a set of static locations and uses the interpolated data to learn a simple machine learning model for localization [9]. Another learning-based technique, MAP*, uses a hypothesis-driven Bayesian approach to localize multiple transmitters [15]. Both of these techniques have low precision, and the computation required for both techniques becomes infeasible as the number of sensors grows, as shown in [14] and this work.

Recent promising approaches for MTL have borrowed from deep learning techniques for image processing. DeepTxFinder [16] introduces the idea of viewing sensor data as a 2D image, where the pixel value corresponding to a sensor position is set to the measured RSS value. However, this method trains multiple different networks to localize different numbers of transmitters, requiring a huge amount of data. Another deep learning technique, DeepMTL [14], uses convolutional neural networks (CNNs) to perform an image-to-image transformation, similar to this paper. However, in their simulations DeepMTL requires an extremely high sensor density as well as a huge amount of training data in order to achieve high accuracy and precision.

In a step towards a practical localization system, we present TL;DL (Transmitter Localization with Deep Learning), a supervised learning technique which uses crowdsourced RSSI measurements to accurately and efficiently localize multiple transmitters. TL;DL can accurately localize transmitters with only a small amount of training data, and maintains a short enough runtime for real-time monitoring. We allow unrestricted mobility for both transmitters and receivers, and make no assumptions about the power level, distance between, or number of transmitters.

Similar to [14, 16], we view localization as an image processing problem. In our setting, a 2D map of sensor data (RSSI values) is transformed via CNN to a 2D map of transmitter locations. Given a set of sensor locations and RSSI values, we represent each sensor as a pixel in a 2D input array, and map each transmitter location to a corresponding pixel in a 2D target array. Our CNN learns to approximate the 2D target array and predict transmitter locations. We use the residual CNN architecture from [8] which removes the need for high sensor density and allows for a deeper network with richer features.

Although deep learning approaches often require a significant amount of training data for good performance, we use novel data augmentation techniques to learn accurate localization functions for small, real-world datasets with as few as 35 training samples, in both indoor and outdoor environments. Our experimental results show that TL;DL achieves significantly higher performance than other state-of-the-art localization techniques on most datasets in terms of both localization error detection rates. We also show that deep learning-based localization is orders of magnitude faster than other state-of-the-art techniques, regardless of whether computation is run on the GPU or CPU.

The main contributions of this work are as follows:

- We develop a deep learning localization technique, TL;DL, which improves on other deep learning-based localization techniques due to an improved CNN architecture.
- Using a novel data augmentation technique, we show that TL;DL is effective even in extremely limited data regimes.
- We evaluate TL;DL in multiple environments, and compare performance with other MTL techniques.
- We discuss our vision for a practical real-time localization system. Although TL;DL is a significant step towards practical localization, several design problems remain before such a system could be implemented.

2 PROBLEM FORMULATION

We consider a geographic area in which various users operate in a shared spectrum setting. Within this area, authorized devices periodically share RSSI measurements and location data with the central manager, as depicted in Figure 1. This central manager is responsible for identifying the location of M transmitters in this area, each of which may be transmitting at a unique power level. Let this set of transmitter coordinates be Q.

At time t, the spectrum manager receives a set of RSSI measurements and coordinates, S_t, crowdsourced from a subset of active users in the environment; each measurement is assumed to contain noise in both the location and RSSI measurements. The objective of learning-based localization is to learn some function L to approximate Q, denoted $L(S_t) = \hat{Q}$.

If L does not correctly predict the number of transmitters, or if the distance between a prediction and the actual transmitter is beyond a maximum error threshold, then this is a case of either *misdetection*, where at least one transmitter is not located, or *false alarm*, where at least one prediction does not correspond to a real transmitter. Either false alarm or misdetection could be preferable in different circumstances, such as when either the cost of active enforcement or the cost of harmful interference is much higher.

3 TL;DL METHODOLOGY

The adoption of deep learning across many domains is due to one of its fundamental advantages: rather than attempting to engineer features to solve a complex learning problem, allow the intermediate layers of a neural network to learn to identify which features are helpful in the overall optimization problem. We attempt a similar approach here; rather than using interpolation between sensors, path loss estimation, power estimation, or other techniques commonly used in localization, we use a single CNN to transform sensor information into transmitter locations.

3.1 Localization via Image Transformation

Let a vector of measurements be S, and the vector of transmitters be Q. The pair (S, Q) forms a single sample of our dataset. In order to capture the spatial relationship between sensors, we use S to generate a 2D map X where each element or pixel in the array corresponds to the location of a sensor, s_i, and is set to the normalized RSSI value of the sensor. We similarly convert the location of transmitters to a 2D map Y, where the location of each transmitter is marked by a 3×3 square of pixels with a center value of 1 and exterior values of 0.5, similar to the technique used in [14]. This 3×3 target is used instead of a single pixel in order to improve the rate at which the network learns, but our experiments found this larger target is not required to achieve similar localization performance. This image-to-image formulation is shown in the left side of Figure 2, where the sensor and transmitter locations both are converted to images X and Y. Note that this conversion process will typically require some discretization of the real-valued coordinates, depending on the distance represented by each pixel. This pixel-distance relationship can have an impact on accuracy and performance, but in this work we only consider a constant scale of 1 meter per pixel.

All pixels in X and Y that do not represent a transmitter or sensor are set to the minimum RSSI value of -114 dB. Unlike other works in localization (such as [9]), we do not perform any interpolation between the sensors in each data sample since a CNN is well suited for learning interpolation functions.

3.2 Data Augmentation

Although most deep learning models are trained on thousands of unique data points, we use novel data augmentation techniques to improve localization accuracy. We use two data augmentation techniques: *power variation* and *sensor dropout*.

Since transmitters typically adjust transmission power according to the environment, we use power variation to ensure localization accuracy when transmitters adjust power levels. For power variation, we normalize all pixels in the input image to be between 0 and 1, then scale all values in each sample by a uniform random factor between 0.8 and 1, which corresponds to a ~0 to 20 dB reduction in RSSI values. Crucially, we do not modify our network architecture in any way in order to handle varying power levels, greatly simplifying the problem compared to iterative methods that find each transmitter, then attempt to subtract the influence of that transmitter from the measured values [2, 9, 14, 15].

With crowdsourced information, we do not expect a central manager to have access to a large set of data, and we also expect devices to be randomly mobile. We use sensor dropout to simulate the receiver mobility one would expect in real crowdsourced data. For sensor dropout, we repeat each data sample many times in our training data, censoring a different set of sensors each time. This process is shown in Figure 3.

3.3 Convolutional Network Architecture

For localization using deep learning, the receptive field of our output image is crucial to system performance. For a transmitter located at position i, j in the target map Y, the receptive field of Y_{ij} is the number of elements in the input X that determine Y_{ij}. If the receptive field is too small to capture any sensors in the input image, then there is no way to correctly localize the transmitter. In particular, we found that small receptive fields were one of the main failings when using earlier deep learning-based approaches for localization.

We use the U-Net architecture from Ronneberger et al. [8]. shown in Figure 4, which has a generous receptive field of 140×140, much larger than any of the images used in our evaluation. U-Net uses residual connections between layers to preserve local high-resolution information that may otherwise be lost during downsampling operations, while allowing for a deeper network.

Our U-Net model takes an input X and produces an output $\hat{Y}$, which is an approximation of the target image Y. If any pixel value in $\hat{Y}$ is greater than a threshold $\gamma = 0.2$, then this pixel location is determined to contain a transmitter. The threshold γ is set experimentally, and can be tuned to reduce the misdetection or false alarm rate. During evaluation, if multiple neighboring pixels are greater than the threshold γ, we suppress any transmitters detected in the 8 neighboring pixels of each local maximum, which reduces the number of false alarms in our output.

We use the per pixel mean-squared error (MSE) as a loss function, rather than the cross-entropy typically used for binary classification. This is due to the imbalance between the two classifications, and was experimentally verified.

Higher sub-pixel accuracy can be achieved by using an additional deep learning model for precise localization. In this work we focus on improving detection in the output image $\hat{Y}$, after which another model could be used to improve sub-pixel localization. We refer the reader to [14] for details on this topic.

One issue common in any detection problem is that the number of predictions may not necessarily match the number of actual transmitters. The predictions from $\hat{Y}$ are recorded in a vector $\hat{Q}$, which may or may not be the same size as the target vector Q. In this case, we perform a minimum distance assignment, where we find the best possible ordering of $\hat{Q}$ so that the total distance between corresponding coordinates Q_i and $\hat{Q}_i$ is minimized. This can be done efficiently, as is well-known.

In summary, TL;DL uses the following approach, shown visually in Figure 2: Convert a set of sensor observations S to a normalized image X. Using the learned CNN function L, compute $\hat{Y} = L(X)$. Use a threshold $\gamma = 0.2$ to determine if each pixel reading is detected as a transmitter, along with a 3×3 filter to suppress detection near local maxima. Given the masked and thresholded version of $\hat{Y}$, take the coordinates of each detected transmitter to make the prediction vector $\hat{Q}$ for final evaluation.

4 EVALUATION AND RESULTS

In order to evaluate the performance of our technique, we train and test TL;DL on various real-world datasets with different numbers of transmitters and different propagation characteristics. We show that TL;DL outperforms prior techniques, according to metrics

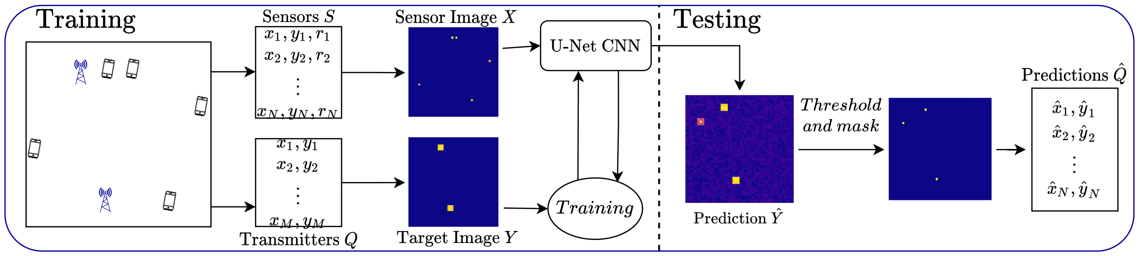

Figure 2: The CNN image training pipeline. Sensor and transmitter data is made into images which train the CNN. The network prediction $\hat{Y}$ is masked and thresholded to isolate transmitters, and the coordinates of each non-zero pixel are output.

Table 1: Datasets used for evaluation

Dataset	Location	Area	Samples	Sensors per sample	Description
1	Cluttered Office [5]	18^2 m	44 with 1-Tx	43 (10 to 12 selected)	Stationary Tx and Rx
2	Outdoor Park	70^2 m	44 with 1-Tx	> 150 (10 to 12 selected)	Stationary Tx, mobile Rx
3	Square Hallways	35^2 m	329 with 1-Tx, 103 with 2-Tx	5 to 9	Mobile Tx and Rx
4	Outdoor Campus	30^2 m	120 with 1-Tx, 308 with 2-Tx	5 to 8	Mobile Tx and Rx
5	ORBIT Testbed [7]	25^2 m	1500 samples with 1 to 5 Tx	3 to 8	Stationary Tx and Rx

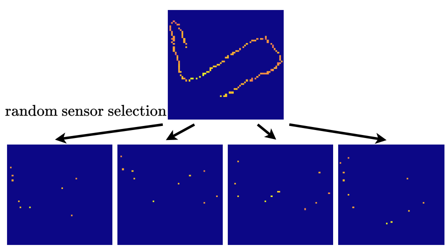

Figure 3: For augmentation via *sensor dropout*, a random subset of sensors is used to create new data samples.

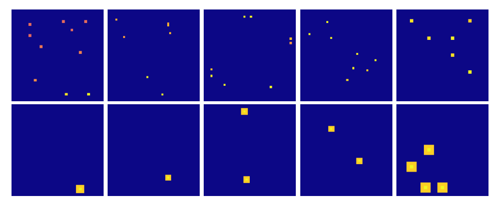

Figure 5: Example images from Datasets 1-5 (R to L). The top row is the sensor input image, with transmitter locations on bottom.

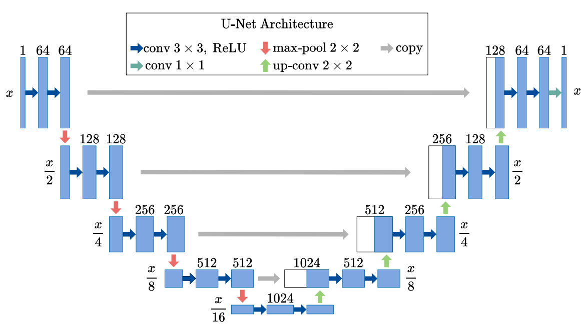

Figure 4: The U-Net architecture

of localization error ϵ_l, misdetection rate ϵ_d, false alarm rate ϵ_{fa}, and the optimal sub-patter assignment (OSPA) metric ϵ_p, which captures localization and cardinality errors in a single metric [10]. OSPA uses following formula when $M \leq \hat{M}$, where $\hat{M}$ is the number of predicted transmitters:

$$e_p = \left(\frac{1}{\hat{M}} \sum_{i=1}^{\hat{M}} d_g(\hat{Q}_i, Q_i)^2 + g^2 \left(\hat{M} - M \right) \right)^{\frac{1}{2}} \quad (1)$$

where $d_g(\hat{Q}_i, Q_i) = \min(g, d(\hat{Q}_i, Q))$. When $M > \hat{M}$, the OSPA metric is obtained by inverting Q and $\hat{Q}$ in (1). The OSPA metric is heavily reliant on a penalty parameter g, which is the penalty for false alarm or misdetection. A larger penalty increases the impact of cardinality errors but can favor predictions with large localization error ϵ_l. We use $g = 15$ as a penalty throughout this work.

We evaluate TL;DL in five different locations. Three of the datasets consist of indoor measurements, and two consist of outdoor measurements. There are between 1 and 5 transmitters per sample. See Table 1 for details on each measurement setup. For additional context, a single sample from each dataset is shown in Figure 5.

We compare TL;DL with the following localization techniques:

- *SPLOT* [2]: Discretize the area of interest into bins, use a path-loss model based on reported data to estimate the transmit power field of each bin, and declare the bin with maximum transmit power field to be the location of the transmitter.

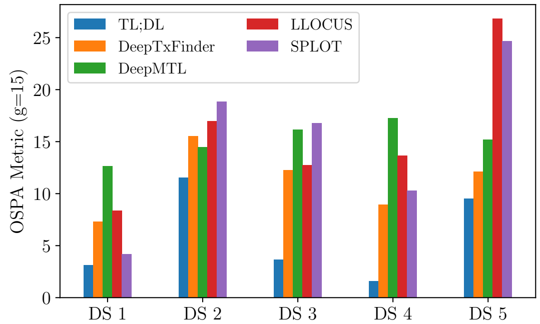

Figure 6: OSPA Metric for all datasets.

- *LLOCUS* [9]: Interpolate mobile sensor data to a set of fixed locations and train a simple ML model to predict transmitter location using interpolated training data.
- *DeepMTL* [14]: CNN-based technique using image-to-image transformation, similar to TL;DL. DeepMTL has a 17×17 receptive field, with no data additional augmentation techniques. It also considers the use of an additional CNN model for sub-pixel prediction, which is not implemented in this evaluation.
- *DeepTxFinder* [16]: Use a CNN with many branches: the first branch predicts the number of transmitters, then the branch of the network corresponding to the predicted number of transmitters outputs a set of coordinates for each transmitter.

For evaluation of each MTL technique, we randomly select 20% of the transmitter samples in each dataset for testing, with the remainder for training. We display the average results for 3 random test/train splits. In order to maintain the crowdsourced setting, we restrict the number of sensors included in each image to no more than 12 sensors per sample, as shown in Table 1.

The deep learning methods, TL;DL, DeepMTL, and DeepTxFinder, were trained independently on each dataset, with no fine-tuning on pre-trained models. Each model was trained for 100 epochs after test error reached a minimum, up to 1000 epochs. Although the training process is notoriously variable, models typically reached a minimum error after 20-200 epochs.

4.1 Experimental Results

First, we compare TL;DL against several other localization techniques for solving MTL, with OSPA results from 5 datasets shown in Figure 6 and full results shown in Table 2. TL;DL outperforms or competes with all other techniques in terms of localization error ϵ_l and the weighted OSPA metric ϵ_p. TL;DL improves on localization error in the majority of the datasets, with the most drastic improvements on Datasets 3, 4, and 5. These results show our image-to-image transformation technique is more effective than other deep learning-based models.

The importance of a large receptive field can be seen by comparing results from TL;DL and DeepMTL, both of which use a similar approach, but with different CNN architectures. In the results for Dataset 3, hallways surround office space in a 30×30 m square, and DeepMTL's small 17×17 receptive field prevents the model from learning to localize transmitters based on distant sensor values.

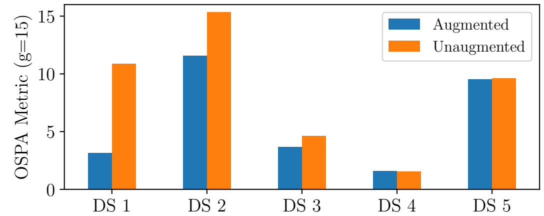

Figure 7: Impact of data augmentation on localization performance for TL;DL.

Another interesting finding shown in Figure 6 is the poor performance of SPLOT and LLOCUS on Dataset 5, which has up to 5 transmitters per test sample. Since both LLOCUS and SPLOT operate under the assumption that no transmitter can be detected without a locally maximum sensor value, they would be incapable of detecting more than a single transmitter in many configurations.

In general, all techniques perform poorly on Dataset 2, indicating that this setting presents a significantly harder localization problem. We had expected difficulty with Dataset 5, since there are multiple transmitters in close proximity, but the poor performance on Dataset 2 is somewhat surprising. We expect that this poor performance is due to large amounts of noise we observed in sensor measurements, both in GPS and RSSI values. Unlike the other datasets, each sample in Dataset 2 was captured by a single mobile sensor over several minutes, rather than multiple sensors taking one simultaneous measurement.

Benefit of augmentation: In Figure 7 we compare the performance of our sensor dropout augmentation technique. We see a notable improvement in performance on Datasets 1 and 2. As shown in Table 1, these datasets only contain 44 unique samples, so using our data augmentation technique to provide 5000 additional samples notably improves performance. The other datasets include several hundred samples, so data augmentation provides less benefit.

Timing results: Table 3 presents the runtime information for each of our various localization techniques on an AMD 3900X CPU. The runtime of various techniques is directly impacted by the type of data present in the dataset. For example, datasets 1 and 2 both contained the same number of training samples, but the matrix-inversion operation of SPLOT took ~70× longer for Dataset 2 due to over 100 sensors per sample in the training data. Similarly, the size difference of the monitored region from Dataset 1 to Dataset 2 (18×18 m to 70×70 m) caused a ~10× slowdown for TL;DL on CPU. However, we can see that deep learning-based models are faster than the competition and efficient enough to use in a real-time monitoring system, even without the use of a GPU.

5 DISCUSSION

TL;DL performs well on small datasets, but it is not yet a complete system for practical real-world localization. We note several potential limitations.

- TL;DL has not been evaluated on a large area. The POWDER [1] experimental testbed requires spectrum monitoring over a 4 km^2 area, and could use potentially use TL;DL to identify sources of interference.

Table 2: Single Transmitter Data

	DS1				DS2				DS3				DS4				DS5			
	ϵ_l	ϵ_d	ϵ_{fa}	ϵ_p	ϵ_l	ϵ_d	ϵ_{fa}	ϵ_p	ϵ_l	ϵ_d	ϵ_{fa}	ϵ_p	ϵ_l	ϵ_d	ϵ_{fa}	ϵ_p	ϵ_l	ϵ_d	ϵ_{fa}	ϵ_p
TL;DL	**2.8**	0	0.01	**3.1**	12.4	0.25	0.02	**11.6**	**2.5**	0.03	0.03	**3.7**	**0.7**	0	0.02	**1.6**	**2.3**	0.06	0.06	**9.5**
DeepTx	7.3	0	0	7.3	57.7	0.14	0.07	15.5	14.1	0.02	0.09	12.3	8.6	0	0.02	9.0	7.4	0.02	0.10	12.1
DeepMTL	6.8	0.35	0.13	12.6	19.7	0.60	0.04	14.5	22.7	0.61	0.07	16.2	12.5	0.40	0.14	17.3	3.3	0.11	0.13	15.2
LLOCUS	3.0	0.30	0.08	8.4	10.1	0.28	0.30	17.0	7.9	0.52	0.06	12.7	7.4	0.08	0.22	13.6	4.9	0.82	0	26.8
SPLOT	3.8	0.04	0	4.19	**9.2**	0	0.57	18.8	9.8	0	0.45	16.8	6.3	0	0.19	10.3	8.7	0.67	0	24.7

Table 3: Runtime of localization techniques (ms).

	DS1	DS2	DS3	DS4	DS5
TL;DL	1	14	4	10	2
DeepMTL	0.1	0.8	0.3	0.6	0.2
DeepTx	2	14	7	15	5
LLOCUS	346	380	15867	2967	43
SPLOT	1	343	33	124	9

- Dynamic environments could also affect TL;DL's performance. As noted in [4] and in our own data, there can be very large variability in RSSI values. It's possible that we could observe even larger RSSI variations over time due to changing foliage, traffic patterns, or other environmental changes. Preliminary tests with TL;DL show that models trained on one dataset do not always achieve a good accuracy in an unseen environment. How could this system become more robust to environmental changes?
- With any deep learning based solution, it is difficult to tell if the architecture used in TL;DL is the "correct" one, especially when require the model to transfer or scale to large areas.
- In each dataset used in this work, all RSSI measurements were taken with the same type of measurement device. In real crowdsourced data, we would expect RSSI values from different types of hardware to have dramatically different RSSI values. How would multiple types of hardware impact TL;DL performance?

6 CONCLUSION

In this paper we presented TL;DL, a practical deep learning-based technique for multiple transmitter localization that uses crowdsourced data to efficiently and accurately locate multiple transmitters. We addressed two primary challenges in developing this method. First, we utilized a CNN architecture with a large receptive field to gather relevant information and residual connections that maintain local features. Second, we developed sensor dropout as a data augmentation technique to make the use of TL;DL feasible in environments with very little training data. We believe that TL;DL gives an approach to design an operational system for real-time localization over large domains.

ACKNOWLEDGMENTS

This work was supported by the National Science Foundation as part of award CNS-1827940.

REFERENCES

[1] Joe Breen, Andrew Buffmire, Jonathon Duerig, Kevin Dutt, Eric Eide, Anneswa Ghosh, Mike Hibler, David Johnson, Sneha Kumar Kasera, Earl Lewis, et al. 2021. POWDER: Platform for open wireless data-driven experimental research. *Computer Networks* 197 (2021), 108281.

[2] Mojgan Khaledi, Mehrdad Khaledi, Shamik Sarkar, Sneha Kasera, Neal Patwari, Kurt Derr, and Samuel Ramirez. 2017. Simultaneous power-based localization of transmitters for crowdsourced spectrum monitoring. In *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*. 235–247.

[3] Chang-Young Kim, Dezhen Song, Yiliang Xu, Jingang Yi, and Xinyu Wu. 2014. Cooperative search of multiple unknown transient radio sources using multiple paired mobile robots. *IEEE Transactions on Robotics* 30, 5 (2014), 1161–1173.

[4] Daniel Konings, Nathaniel Faulkner, Fakhrul Alam, et al. 2017. Do RSSI values reliably map to RSS in a localization system?. In *2017 2nd Workshop on Recent Trends in Telecommunications Research (RTTR)*. IEEE, 1–5.

[5] Neal Patwari, Alfred O Hero, Matt Perkins, Neiyer S Correal, and Robert J O'dea. 2003. Relative location estimation in wireless sensor networks. *IEEE Transactions on signal processing* 51, 8 (2003), 2137–2148.

[6] Anshul Rai, Krishna Kant Chintalapudi, Venkata N Padmanabhan, and Rijurekha Sen. 2012. Zee: Zero-effort crowdsourcing for indoor localization. In *Proceedings of the 18th annual International Conference. on Mobile Computing and Networking*. ACM, 293–304.

[7] Dipankar Raychaudhuri, Ivan Seskar, Max Ott, Sachin Ganu, Kishore Ramachandran, Haris Kremo, Robert Siracusa, Hang Liu, and Manpreet Singh. 2005. Overview of the ORBIT radio grid testbed for evaluation of next-generation wireless network protocols. In *IEEE Wireless Communications and Networking Conference, 2005*, Vol. 3. IEEE, 1664–1669. https://doi.org/10.1109/WCNC.2005.1424763

[8] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*. Springer, 234–241.

[9] Shamik Sarkar, Aniqua Baset, Harsimran Singh, Phillip Smith, Neal Patwari, Sneha Kasera, Kurt Derr, and Samuel Ramirez. 2020. LLOCUS: learning-based localization using crowdsourcing. In *Proceedings of the Twenty-First International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing*. 201–210.

[10] Dominic Schuhmacher, Ba-Tuong Vo, and Ba-Ngu Vo. 2008. A consistent metric for performance evaluation of multi-object filters. *IEEE transactions on signal processing* 56, 8 (2008), 3447–3457.

[11] Sameh Sorour, Yves Lostanlen, and Shahrokh Valaee. 2012. RSS based indoor localization with limited deployment load. In *2012 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 303–308.

[12] Chris Welch. 2014. Florida man drove around as a cellphone-jamming vigilante for two years. https://www.theverge.com/2014/5/1/5672762/man-faces-48000-fine-for-driving-with-cellphone-jammer

[13] Kiran Yedavalli, Bhaskar Krishnamachari, Sharmila Ravula, and Bhaskar Srinivasan. 2005. Ecolocation: a sequence based technique for RF localization in wireless sensor networks. In *IPSN 2005. Fourth International Symposium on Information Processing in Sensor Networks, 2005*. IEEE, 285–292.

[14] Caitao Zhan, Mohammad Ghaderibaneh, Pranjal Sahu, and Himanshu Gupta. 2022. DeepMTL Pro: Deep Learning Based Multiple Transmitter Localization and Power Estimation. *Pervasive and Mobile Computing* (2022). https://doi.org/10.1016/j.pmcj.2022.101582

[15] Caitao Zhan, Himanshu Gupta, Arani Bhattacharya, and Mohammad Ghaderibaneh. 2020. Efficient Localization of Multiple Intruders in Shared Spectrum System. In *2020 19th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. IEEE, 205–216.

[16] Anatolij Zubow, Suzan Bayhan, Piotr Gawłowicz, and Falko Dressler. 2020. DeepTxFinder: Multiple Transmitter Localization by Deep Learning in Crowdsourced Spectrum Sensing. In *2020 29th International Conference on Computer Communications and Networks (ICCCN)*. IEEE, 1–8.