INTL JOURNAL OF ELECTRONICS AND TELECOMMUNICATIONS, 2026, VOL. 72, NO. 2, PP. 1-7
Manuscript received November 03, 2025; revised April 2026. doi: 10.24425/ijet.2026.157917

Neural networks for efficient touch detection on capacitive panels

Oleksandr Karpin[a,b], Zinovii Liubun, Vasyl Mandziy, and Andriy Luchechko

***Abstract*—The advancement of capacitive-based touch panel technologies has opened new opportunities for their incorporation into embedded devices. However, this progress also underscores the need for improved software algorithms to achieve high precision in calculating touch coordinates. Traditional position calculation methods often exhibit diminished accuracy when applied to smaller panels, and modifying and tuning these methods can be time-consuming and labor-intensive. To address these limitations, this study investigates the performance of two neural network architectures, specifically a two-layer fully connected neural network and a radial basis function network, in enhancing the accuracy of touch coordinate calculation. A key advantage of these models is their ability to learn efficiently from limited datasets while minimizing the risk of overfitting. The high touch position accuracy achieved by the proposed neural network solutions makes them suitable for deployment in devices with limited computing resources, such as microcontrollers. Furthermore, the simplicity of the proposed models enables their implementation in embedded systems with low power consumption, offering a practical and scalable solution for a wide range of applications. Overall, the integration of these neural network models in touch coordinate processing provides notable benefits in terms of accuracy, efficiency, and adaptability.**

***Keywords*—capacitive sensor; multilayer neural network; radial-basis function; embedded devices; microcontrollers**

I. INTRODUCTION

THE proliferation of capacitive touch panels has gained significant momentum in recent years. According to market projections, the global touchscreen controller market is anticipated to reach a substantial value of $22.2 billion within the next decade [1]. This remarkable growth can be attributed to a multifaceted array of factors, including the escalating demand for touch-enabled consumer electronics, such as smartphones, tablets, and laptops. Furthermore, the expanding integration of touchscreens into various domains, including automotive displays, home appliances, and industrial controls, has also contributed to this trend. Additionally, advancements in touch technology, particularly the development of flexible and foldable displays, have played a crucial role in driving this growth. The burgeoning trend of smart homes and Internet of Things (IoT) devices has also created a surge in demand for compact, energy-efficient, and highly responsive capacitive touch panels. Moreover, continuous innovation in semiconductor technology, and microcontroller architecture has enabled the development of high-performance touch controllers, which have further accelerated the adoption of capacitive touch panels.

Concomitantly, the effective utilization of capacitive-based touch panel technologies necessitates the development and refinement of sophisticated software algorithms to ensure high accuracy and precision in the computation of touch coordinates [2-4]. Existing approaches for position detection in capacitive sensor devices can be broadly classified into two main approaches: traditional analytical methods and machine learning-based methods. Traditional methods, such as centroid calculation and peak detection, are widely used due to their simplicity and reasonable accuracy in many capacitive sensor devices [2], [3]. They are simple to implement and provide good accuracy in many cases. However, they can be limited by noise, interference, and the non-linear response of the sensor, which can significantly degrade performance in compact or high-density touch panels. Machine learning-based methods, such as neural networks and support vector machines, have been increasingly used in capacitive sensor devices in recent years [5]. These methods can provide better accuracy and robustness than traditional methods, especially in cases where the sensor response is non-linear or noisy.

One of the primary challenges in developing neural network-based touch detection systems is the limited amount of data available for training, which precludes the use of deep neural networks with a large number of parameters. Such networks are prone to overfitting, resulting in decreased noise resistance and reduced recognition accuracy. Under these conditions, the use of compact models is advisable, as they can achieve high recognition accuracy using minimal datasets while maintaining computational efficiency. Additionally, capacitive sensing systems often operate in low signal-to-noise environments, where interference, parasitic capacitance, and temperature drift complicate the extraction of reliable signals. Several methodologies have been proposed to address these challenges, including filtering, amplification, noise reduction, and using machine learning [4], [6-10].

This article proposes two effective neural network architectures with a small number of weighting coefficients: a two-layer Fully Connected Neural Network (FCNN) and a neural network based on Radial Basis Function Network (RBFN). Both models are specifically designed for use in embedded devices, providing an optimal balance between recognition accuracy and computational load. Moreover, they maintain high noise immunity even when trained on limited data sets, making them suitable for a wide range of applications.

The proposed neural network models are characterized by their simplicity, efficiency, and adaptability, making them ideal for integration into embedded systems with limited hardware

Oleksandr Karpin, Vasyl Mandziy, and Andriy Luchechko are with the Department of Sensor and Semiconductor Electronics, Ivan Franko National University of Lviv, and Infineon Technologies, Ukraine.

Zinovii Liubun is with the Department of Radiophysics and Computer Technologies, Ivan Franko National University of Lviv, Ukraine.

resources. The use of these models can significantly improve the accuracy and robustness of touch detection systems, thereby enabling the development of more sophisticated and user-friendly interfaces for next-generation embedded systems.

II. RESULTS OF THE STUDY AND THEIR DISCUSSION

A. Materials and methods

Figures 1, 2 shows the two neural network architectures used in the study: a two-layer fully connected neural network and a radial basis function network.

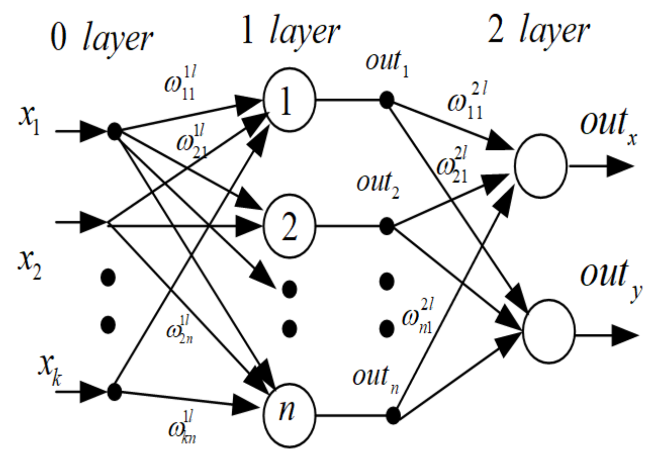


Fig. 1. Two-layer fully connected neural network

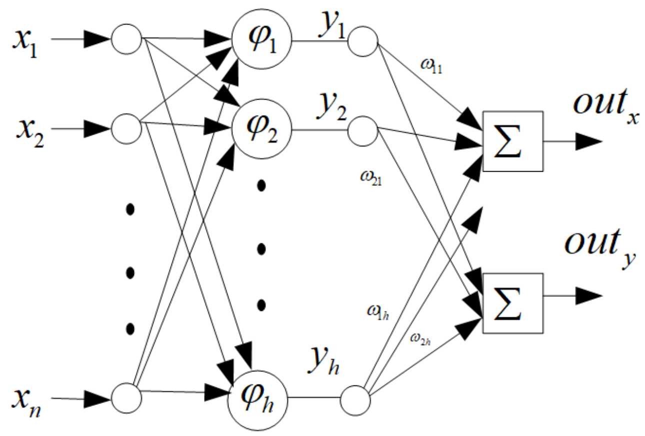


Fig. 2. Radial basis function neural network

A fully connected neural network's equations for the first and second layers are expressed as:

$$\begin{cases} h_j = f\left(b_j^{1l} + \sum_i w_{ij}^{1l} \cdot x_i\right) \\ out_j = f\left(b_j^{2l} + \sum_i w_{ij}^{2l} \cdot h_i\right) \end{cases} \qquad (1)$$

where x_i are the inputs, w_{ij}^{1l}, w_{ij}^{2l} and b_j^{1l}, and b_j^{2l} are the weights and biases of the hidden and output layers, f is a non-linear activation function (ReLu in this work).

The Gaussian radial basis function is defined as:

$$\begin{cases} y_i = e^{-\frac{(x-c_i)^2}{\sigma_i^2}} \\ out_j = \sum_i w_{ij} \cdot h_i \end{cases} \qquad (2)$$

where c_i represent the centres and σ_i represent the widths or spreads of the radial basis function.

The models were trained on a small sample of training examples obtained during real measurements of touch panel signals shown in Figure 3.

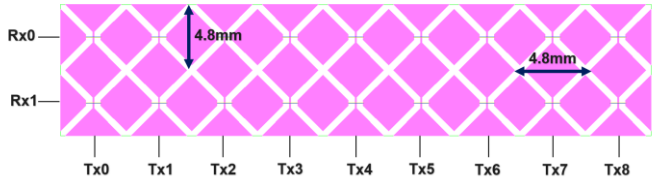


Fig. 3. Capacitive touch panel

To ensure consistency and repeatability of measurements, a custom experimental setup was developed for data collection for further training of the neural network (Figure 4).

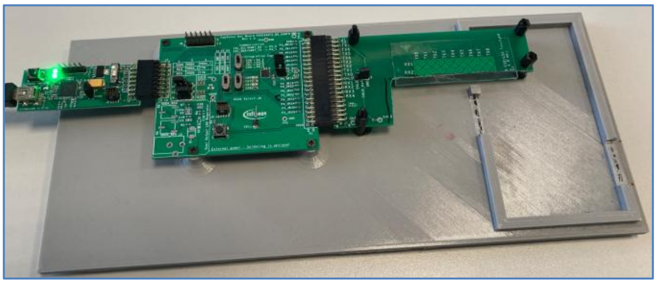


Fig. 4. Setup for data collection

Bachin T-A4 drawing robot was adapted for the automated data collection and touch panel testing, as depicted in Figure 5.

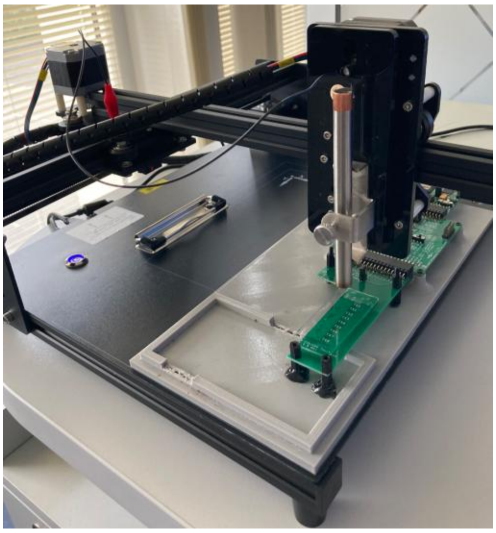


Fig. 5. Setup on robot

The schematic illustration depicted in Figure 6 delineates the interfacing protocol for interfacing the touch panel (Figure 3) with the PSOC™ 4 MCU from Infineon. This setup employs the mutual capacitance sensing technique as the primary measurement method. In this configuration, capacitance is defined as the capacitive coupling that occurs between two distinct electrodes, wherein one electrode is designated as the transmit (Tx) electrode and the other as the receive (Rx) electrode. The measurement process involves applying a signal, whose value oscillates between the supply voltage and either a ground potential (GND) or half of the supply voltage, to the Tx electrode. This oscillating signal induces a charge on the Rx electrode, the magnitude of which is directly proportional to the mutual capacitance (C_M) existing between the two electrodes. The amount of charge received by the Rx electrode is subsequently measured by the MCU, providing an indirect quantification of the mutual capacitance.

When a conductive object such as a human finger approaches the touch surface, it alters the local electric field distribution and consequently the mutual capacitance. The mutual capacitance sensing technique relies on the principle that the presence of a conductive object, in proximity to the touch panel, alters the capacitive coupling between the Tx and Rx electrodes. This alteration in capacitance is then detected and processed by the PSOC^{TM} 4 MCU, enabling the accurate determination of touch position and movement on the touch panel.

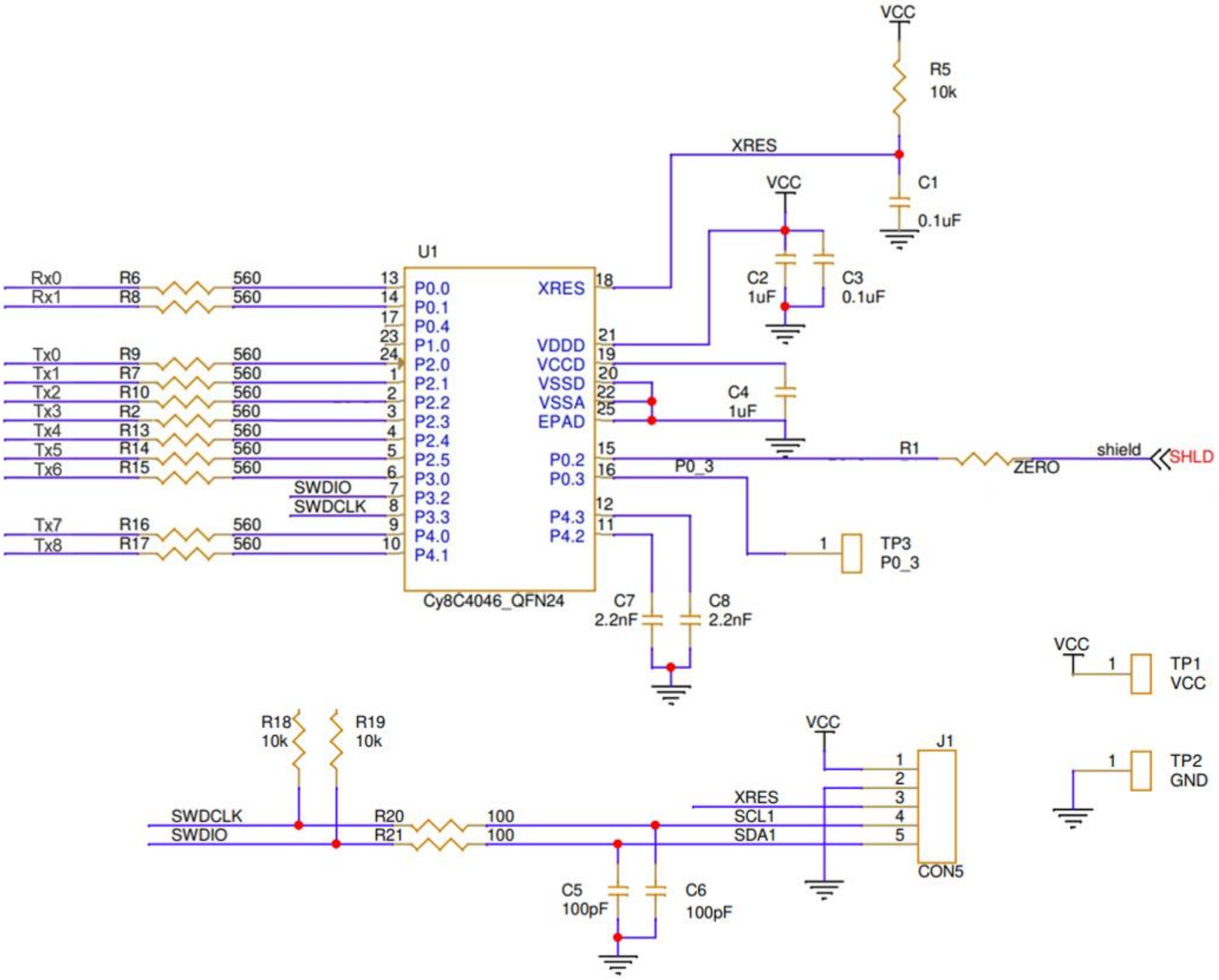


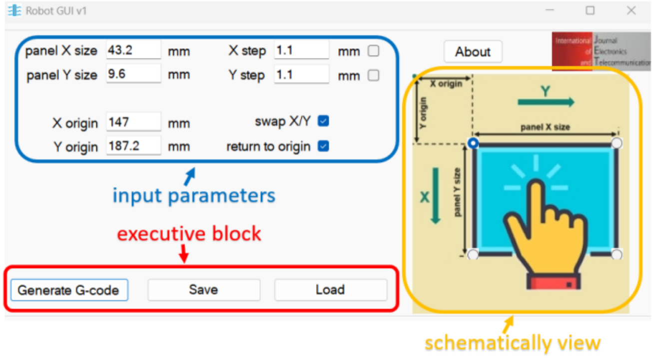
Fig. 6. Schematic for touch panel connection with PSOC^{TM} 4 MCU

To streamline the testing process, a custom C# graphical user interface (GUI) was developed with the following functionalities:

- automatically generate G-code (.nc file) to manipulate the drawing robot for the data collection and accuracy tests;
- automatically create a *.txt* file containing X and Y robot touch coordinates;
- save and load the panel configuration parameters.

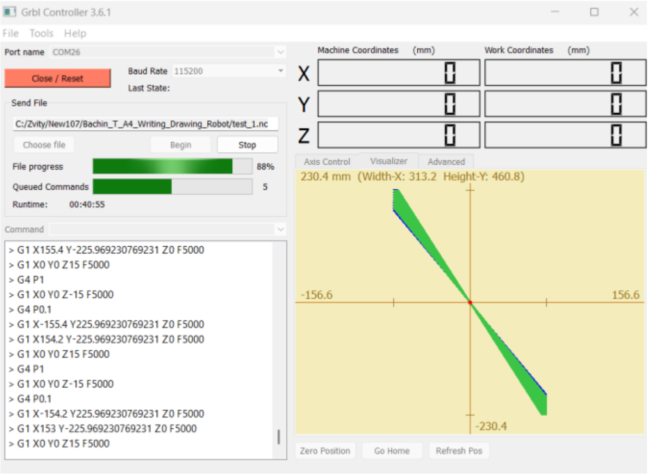
The Robot GUI (Figure 7) consists of three main blocks: input parameters, executive blocks, and a schematic view.

For executing the generated motion commands, the Grbl Controller software was used, as illustrated in Figure 8.


Robot GUI v1

panel X size 43.2 mm
X step 1.1 mm ☐
panel Y size 9.6 mm
Y step 1.1 mm ☐
X origin 147 mm
swap X/Y ☑
Y origin 187.2 mm
return to origin ☑

Fig. 7. Robot GUI interface



Port name COM26

Baud Rate 115200

Last State:

Send File

C:/Zvity/New107/Bachin_T_A4_Writing_Drawing_Robot/test_1.nc

File progress 88%

Queued Commands 5

Runtime: 00:40:55

```
> G1 X155.4 Y-225.969230769231 Z0 F5000
> G1 X0 Y0 Z15 F5000
> G4 P1
> G1 X0 Y0 Z-15 F5000
> G4 P0.1
> G1 X-155.4 Y225.969230769231 Z0 F5000
> G1 X154.2 Y-225.969230769231 Z0 F5000
> G1 X0 Y0 Z15 F5000
> G4 P1
> G1 X0 Y0 Z-15 F5000
> G4 P0.1
> G1 X-154.2 Y225.969230769231 Z0 F5000
> G1 X153 Y-225.969230769231 Z0 F5000
> G1 X0 Y0 Z15 F5000
```

Fig. 8. Grbl Controller interface for test execution

The neural network models were trained using gradient-based optimization methods, a class of algorithms that minimize the loss function and optimize the model's parameters. The limited number of training examples available required the use of data augmentation methods to artificially increase the size of the training dataset. This was achieved by adding random noises to the real-time measured signals, which were subject to a normal distribution with zero mathematical expectation and different values of the standard deviation. The addition of Gaussian noise to the training data had the effect of increasing the model's resistance to interference that may occur during the actual operation of the touch panel. This is because the model was trained to recognize patterns in the data that are robust to noise, which improved its ability to generalize to new, unseen data.

After creating the augmented dataset, it was mixed and then divided into three subsets: training, validation, and testing. The training subset was used to train the model, the validation subset was used to evaluate the model's performance during training and prevent overfitting, and the testing subset was used to evaluate the model's performance on unseen data.

The input data for the neural networks consisted of six signals from the sensors that surround the touch point, which was determined by the signal with the maximum amplitude. All input values were normalized to eliminate the influence of external factors that could change the baseline signal amplitude. It is worth noting that normalization is a common technique in machine learning used to ensure that all input features are on the same scale, thereby improving the model's performance and preventing features with large ranges from dominating the model. Most attention was paid to the cases where the touch was located on the edge of the touch panel, since this is the area where traditional analysis algorithms demonstrate the largest errors in determining the position of a touch. The edge of the touch panel presents a challenging region for touch detection algorithms, as the signal strength is often weaker and more susceptible to interference from external factors.

To address this challenge, the neural network models were designed to learn features that are robust to noise and interference, and to accurately detect touches on the edge of the panel. The models were trained on a dataset that included a large number of examples of touches on the edge of the panel, which helped to improve their performance in this region.

The signal-to-noise ratio (SNR) was estimated using the relationship:

$$SNR = \frac{A_{Signal}}{6*\sigma_{noise}} \tag{3}$$

where A_{Signal} is the signal amplitude and σ_{noise} is the standard deviation of the noise.

B. Results of the study

A comprehensive simulation study was provided to evaluate the efficacy of two distinct neural network architectures for determining the coordinates of a touch on a capacitive surface. The primary objective was to assess and compare the performance of these architectures in terms of localization accuracy and noise robustness, particularly when trained on a limited dataset.

The simulation results reveal that both proposed neural network structures exhibit high localization accuracy, even when trained on a limited amount of data. This finding suggests that the developed architectures are capable of learning complex patterns and relationships within the data, enabling them to accurately determine the coordinates of a touch on the touch surface.

TABLE I
COMPARISON OF NEURAL NETWORKS BY ACCURACY

Two-layer fully-connected network		
Number of neurons	SNR under network testing	M
8	16	0
8	8	0
8	4	< 2%
Neural network with RBF		
Number of neurons	SNR under network testing	M
18	16	0
18	8	0
18	4	< 1%

The neural network with radial basis functions (RBF) demonstrated particularly high accuracy and robustness against noise. This architecture boasts a high generalizing ability, is easily trained even on noisy data, and allows for a low error rate in a wide range of noise levels. The RBF architecture's localized activation and inherent smoothness of interpolation make it especially effective for modeling the nonlinear mapping between sensor signals and physical touch positions. In particular, when using 18 basis functions, the RBF network achieved zero instances of localization errors exceeding 1 mm at significant noise in the input data (SNR = 8). Even at a higher noise level (SNR = 4), the share of significant errors remained below 1%. These results confirm that the RBF network is highly effective in mitigating the effects of noise and achieving accurate localization.

In contrast, the two-layer fully connected neural network has a simpler implementation and requires lower computing resources. This makes it well-suited for use in embedded microprocessor systems where hardware resources and power consumption are constrained. At the same time, the network provided high accuracy for moderate noise levels (SNR ≥ 8). With a stronger noise (SNR = 4), the share of significant errors remained below 2%. This balance between computational simplicity and accuracy highlights that the two-layer fully connected network is a viable alternative for applications where computational resources are limited, such as in low-power and real-time embedded systems.

The generalized test results of the above architectures are shown in Table I. Indicator M displays the percentage of test vectors for which the modulus of an error in determining the touch coordinate exceeds 1 mm.

The two-layer fully connected network was trained on 300 “clean” training pairs and an additional 300 synthesized pairs created by adding Gaussian noise at an SNR of 16. This training dataset was designed to simulate real-world conditions, where noise is often present in the input data. Similar data sets were used to train the RBF network, but with a lower noise level (SNR = 4) to test its resilience under more challenging conditions. The obtained results suggest that these configurations are optimal for achieving the best compromise between the accuracy, noise tolerance, and implementation complexity. The two-layer fully-connected network demonstrated high accuracy and robustness to noise, while the RBF network showed exceptional noise tolerance and generalization ability.

The distribution of testing errors is illustrated in the histograms (Figure 9), which reflect the nature of the network operation under noise conditions for the worst examples from Table I.

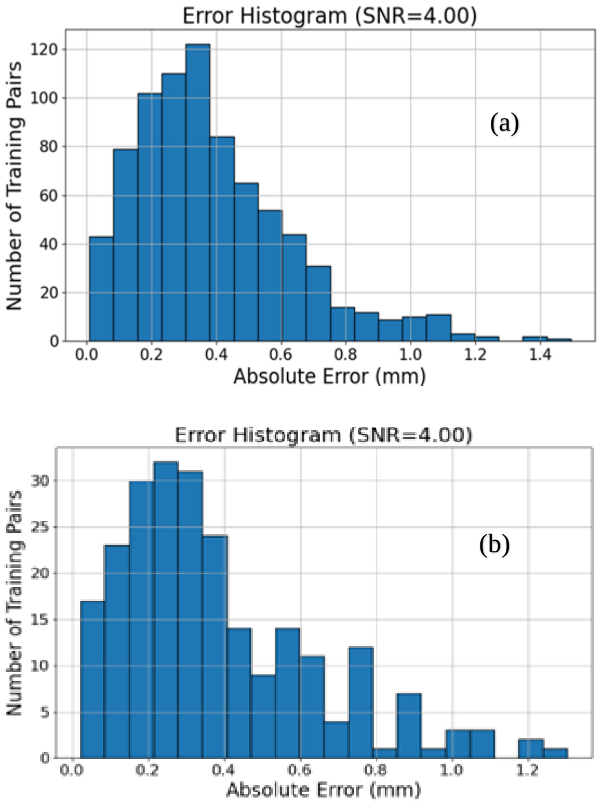


Fig. 9. Histograms of testing errors: (a) – two-layer fully connected NN, (b) – radial basis function NN

Figures 10, 11 illustrate the graphs of the testing error field for the same examples as in Figure 9. Here, the beginning of each arrow is the actual touch position (in millimeters), and the end of the arrow is the position calculated by the neural network implemented in the PSOC^{TM} 4 MCU microcontroller.

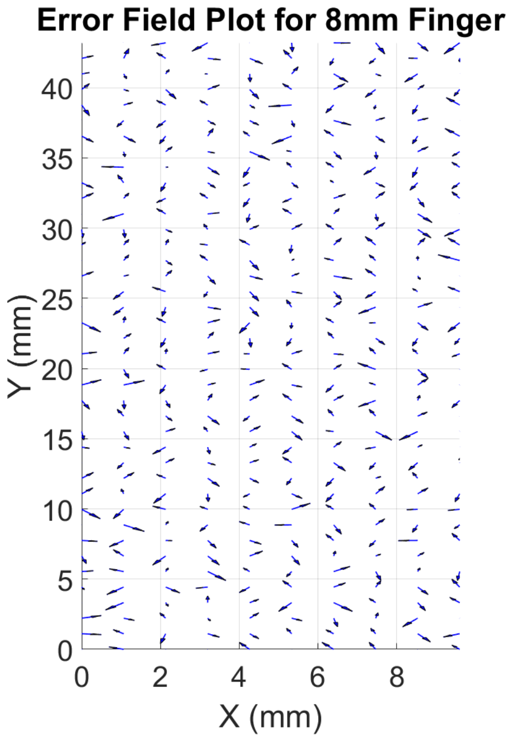


Fig.10 Error field plot for two-layer fully connected NN

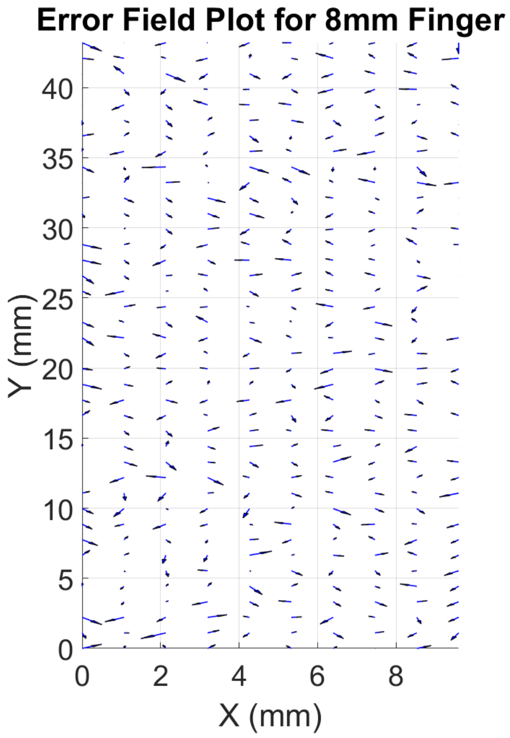
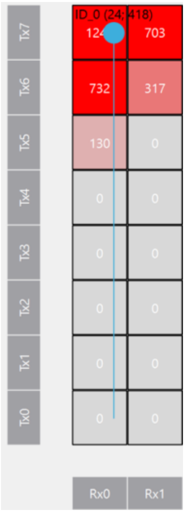


Fig. 11. Error field plot for radial basis function NN

To validate the simulation results under real operating conditions, real-time testing was performed using an actual finger interaction on the touch panel surface. The example of a vertically oriented finger swipe (from bottom to top) is shown in Figure 12, confirming the strong agreement between simulated and experimental results.



	Rx0	Rx1
Tx7	124	703
Tx6	732	317
Tx5	130	0
Tx4	0	0
Tx3	0	0
Tx2	0	0
Tx1	0	0
Tx0	0	0

Fig. 12. Example of finger trajectory when swiping vertically touch panel from bottom to top

Overall, the results of this study demonstrate the potential of compact neural network architectures to provide accurate and robust touch localization, highlighting the importance of carefully considering the trade-offs between accuracy, noise tolerance, and implementation complexity in the design of touch sensing systems.

III. CONCLUSIONS

The proposed models, comprising a two-layer fully connected neural network and a radial basis function neural network, have demonstrated high positioning accuracy even under challenging conditions, including the presence of noise and limited training data. These results confirm that both architectures are robust and effective in determining the coordinates of a touch on a touch surface, making them suitable for a wide range of applications.

The RBF network, in particular, has exhibited to be highly effective due to its exceptional generalization capability, high noise resistance, and fast training times. These characteristics make it an attractive choice for applications where accuracy and robustness are critical, and where the availability of training data is limited. The RBF network's ability to generalize well to new, unseen data also makes it suitable for applications where the touch surface is subject to variations in noise, temperature, or other environmental factors.

On the other hand, the two-layer fully connected architecture has a simpler structure and lower computing resource requirements, making it an ideal choice for implementation in microcontroller-based embedded devices. This is particularly important in applications where power consumption, memory, and computing capabilities are limited, such as in wearable devices, mobile devices, or other portable systems. The two-layer fully-connected network's simpler structure also makes it easier to train and deploy, reducing the overall development time and cost.

One of the key advantages of both networks is their low computational requirements, which makes them suitable for integration into systems with limited power, memory, and computing capabilities. This is particularly important in applications where energy efficiency is crucial, such as in battery-powered devices or other systems where power consumption must be minimized. The low computational requirements of both networks also make them suitable for real-time applications, where fast and accurate processing of touch data is critical.

CREDIT AUTHORSHIP CONTRIBUTION STATEMENT

OK: Supervision, Conceptualization, Methodology, Data curation, Formal analysis, Writing – review & editing; **ZL**: Resources, Conceptualization, Visualization, Writing – review and editing; **VM**: Investigation, Software, Formal analysis, Methodology, Writing – original draft; **AL**: Supervision, Validation, Visualization, Writing – original draft.

REFERENCES

[1] Touchscreen Controllers – Global Strategic Business Report, https://www.researchandmarkets.com/report/touchscreen-controller?srsltid=AfmBOorV3fh5ertxYveDusICbcF7otpXp-4ou7EERXXx9BJ-_TVGJLJu; October 2025

[2] Klein, H.W. et al. (2019). Advanced Capacitive Sensing for Mobile Devices. In: Makinwa, K., Baschirotto, A., Harpe, P. (eds) Low-Power Analog Techniques, Sensors for Mobile Devices, and Energy Efficient Amplifiers. Springer, Cham. https://doi.org/10.1007/978-3-319-97870-3_7

[3] Infineon Technologies CAPSENSETM Design Guide https://www.infineon.com/assets/row/public/documents/30/42/infineon-an90071-cy8cmbr3xxx-capsenser-design-guide-applicationnotes-en.pdf?fileId=8ac78c8c7cdc391c017d07236b6e4698, 2023

[4] Liubun Z., Bryk B., Mandziy V., Karpin O., Kalivoshka B., Velhosh S. Processing Sensor Signal Under Low Values of Signal to Noise Ratio // 2023 IEEE 13th International Conference on Electronics and Information Technologies (ELIT), Lviv, Ukraine, 2023, pp. 73-76, https://doi.org/10.1109/ELIT61488.2023.10310890

[5] Streli, P, Holz, C. CapContact: Super-resolution Contact Areas from Capacitive Touchscreens. CHI 21: Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems, May 2021, Article No.: 289, Pages 1–14. https://doi.org/1145/3411764.3445621

[6] Abisado M., Gerardo B., Fajardo A. Towards keystroke analysis using neural network for multi-factor authentication of learner recognition in online examination. In Manila International Conference on Trends in Engineering and Technology (2017), pp. 71–74

[7] Multi-Touch – Part 2: Filtering and Touch Detection, 2020. [Online]. Available: https://larrylisky.com/2014/03/11/multi-touch-part-2-filtering-and-touch-detection/; 2020

[8] Z. Liubun, V. Nesterenko, V. Mandziy, O. Karpin, "Implementation of digital filters using neural networks when the object movement dynamics are not known," Electronics and information technologies, Is. 19, pp. 48–57, 2022

[9] S. Haykin. Kalman Filtering and Neural Networks. First published:1 October 2001. Print ISBN:9780471369981 Online ISBN:9780471221548 https://doi.org/10.1002/0471221546

[10] Oxford-Man. Recurrent Neural Filters: Learning Independent Bayesian Filtering Steps for Time Series Prediction. Retrieved from https://www.oxford-man.ox.ac.uk/wpcontent/uploads/2020/03/Recurrent-Neural-Filters-LearningIndependent-Bayesian-Filtering-Steps-for-Time-Series-Prediction.pdf