A Minimal Neural Network for Reproducible Gesture Recognition on Knitted Capacitive Touch Sensors

Daniel Schwartz[0000-0002-7301-2622], Dario Salvucci[0000-0001-8812-4996], Yusuf Osmanlioglu[0000-0002-9997-9479], Richard Vallett[0000-0002-7876-4137], Liam Ostrander[0009-0008-3024-4379], Genevieve Dion[0000-0002-4873-288X], and Ali Shokoufandeh[0000-0002-3729-4490]

Drexel University, Philadelphia PA 19104, USA
{des338,dds26,yo42,rjv35,lo359,gd63,as79}@drexel.edu

Abstract. Smart textiles with embedded capacitive touch sensors (CTS) hold great promise for intuitive gesture-based human-computer interaction. However, recognizing complex gestures in real-time on resource-constrained wearable devices remains a challenge. This paper presents a novel approach using a minimalist neural network architecture tailored for efficient gesture recognition on knitted CTS. We emphasize reproducibility throughout our work by providing detailed algorithmic implementation, the influence of key parameters on result quality, and the integration of our source code into other frameworks. Our method demonstrates the ability to accurately classify a variety of single- and multi-touch gestures, including taps, swipes, and pinches, with accuracy rates exceeding 90% on both training and testing data. The proposed approach is well-suited for deployment on embedded devices and offers a significant step towards enabling natural and seamless interaction with smart textiles across diverse applications, such as healthcare, accessibility, and fashion. This work not only advances the field of smart textiles but also contributes to the broader goals of reproducibility in pattern recognition research, making it a valuable resource for further scientific exploration and validation. Code is available at `https://github.com/dsbuddy/knitted-capacitive-touch-sensor-gesture-recognition`

Keywords: Smart textiles · Capacitive touch sensors · Gesture recognition · Embedded devices · Reproducibility

1 Introduction

The field of smart textiles, or e-textiles, is transforming the landscape of wearable technology and human-computer interaction (HCI). By weaving together traditional fabrics with advanced sensors and algorithms, these textiles create a seamless interface between the user and the digital world. Within this field, capacitive touch sensors (CTS) have emerged as a particularly promising technology for gesture recognition. Their high sensitivity, low profile, and ability to detect multiple touch points make them ideal for integrating into clothing and accessories, enabling intuitive, hands-free control of various digital devices.

Gesture recognition is a crucial aspect of HCI, especially as wearable devices become more prevalent. Accurate and reliable interpretation of gestures not only offers a more natural and engaging way to interact with technology but also opens up a world of possibilities across various domains. From improving accessibility for individuals with disabilities to enhancing healthcare monitoring and even revolutionizing the fashion industry, the applications of effective gesture recognition are far-reaching.

However, mapping raw sensor data from multiple channels into meaningful gestures is a significant challenge. The complexity of this task necessitates sophisticated signal processing and machine learning algorithms. Recognizing this need, our research presents a novel approach to gesture recognition using a minimalistic neural network architecture. Designed with computational efficiency in mind, this method is specifically tailored for deployment on embedded devices with limited resources.

Our work demonstrates the reproducibility of recognizing a variety of gestures on a knitted CTS, including taps, swipes, and pinches. Initially explored within the context of drone control, this technology's reproducible potential extends far beyond, offering robust opportunities for diverse applications. From enabling intuitive control of personal devices to facilitating gesture-based communication for individuals with speech impairments, the impact of this research could be transformative. In essence, this work represents a significant step forward in the development of practical and user-friendly gesture recognition systems for smart textiles, paving the way for a future where technology seamlessly integrates with our everyday lives.

2 Related Work

The field of "smart textiles" has seen significant advancements, with capacitive touch sensing (CTS) emerging as a key technology for user interaction [10,11,12,16]. Previous research in textile touch sensing has explored various methodologies, including contact sensing, which relies on changes in inter-yarn contact, and resistive sensing, which detects touch through changes in electrical resistance[4,5]. While these approaches have been effective in certain contexts, they face limitations in flexibility and integration into wearable devices [13,1].

Our work focuses on capacitive sensing due to its high sensitivity and low profile, making it ideal for wearable applications [2,3]. Traditional capacitive sensors often use interleaved wire matrices to detect touch location, but this method is not easily adaptable to flexible textile substrates due to the complexity of connections required.

To address these challenges, we utilize a capacitive touch sensing method that leverages standard weft knitting machinery and commercially available materials. Unlike conventional methods, our approach infers touch location across a continuous conductive substrate through current flow measurements at a limited

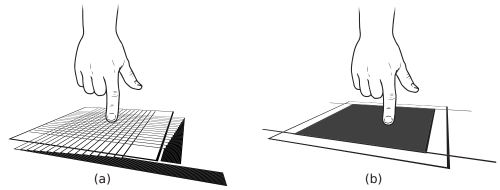
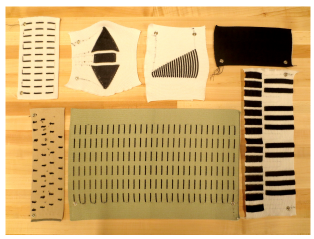
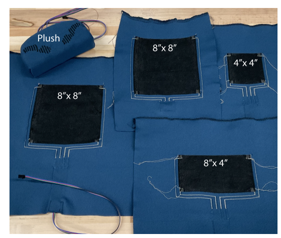


Fig. 1. Comparison of sensing substrate. (a) Capacitive sensing matrix. (b) Differential capacitive sensing.



(a) A collection of touchpads that localize touch along a serpentine linear pathway using two electrodes.



(b) A collection of touchpads that localize touch across a planar conductor using four electrodes.

Fig. 2. Examples of knitted CTS touchpads.

number of points (see Figure 1) [14]. This strategy allows for greater flexibility in substrate shape and size, making it well-suited for smart textiles. However, it requires advanced signal processing to accurately decouple touch location and induced capacitance from the data.

Furthermore, our research emphasizes reproducibility, introducing a minimalist neural network architecture for gesture recognition on knitted CTS. This architecture is designed with computational efficiency in mind, making it suitable for deployment on resource-constrained embedded devices. Unlike existing approaches that rely on complex algorithms or large datasets [14,15], our streamlined approach ensures that the models can be reproduced and validated across different platforms. By integrating this innovative neural network with our unique capacitive touch sensing method, we advance the field of smart textiles and facilitate more intuitive and seamless human-computer interaction, while also contributing to the broader goals of reproducibility in pattern recognition research.

3 Background

This paper focuses on inferring user input from a specific type of smart textile: a capacitive touch sensor (CTS) that localizes contact across a knitted textile conductor (see Figure 2) [15,14]. The CTS functions by measuring current differentials at multiple points along the textile substrate, which are then converted into voltage waveforms that encode the location and magnitude of the touch. This design simplifies the physical connections between the conductive substrate and external sensing electronics, providing greater flexibility in the overall design.

The operating principles of this CTS are grounded in the fundamental mechanics of conventional capacitive touch sensors. It employs oscillating voltage waveforms to drive current through a resistor-capacitor (RC) network. When a touch is applied, it induces additional capacitance, altering the charge and discharge rate of the baseline capacitance. This alteration is directly correlated with the touch's location and magnitude [15,14]. While this approach generates detailed data on sensor activity over time, translating this raw data into meaningful user interactions poses significant challenges. These challenges are compounded in our current planar CTS implementation, which is limited to four data channels corresponding to electrode points located at the corners of the device (see Figure 4 that illustrates the fabrication process and current flow model of this planar CTS.

Although the CTS approach has shown great potential in various recent applications (e.g., [9,8]), significant challenges remain in its practical implementation, particularly in mapping the CTS's input signals to recognizable interactive gestures such as taps, swipes, and pinches. In the following sections, we address these challenges by introducing a minimal neural network architecture designed to recognize interactive patterns in user behavior. We also demonstrate the real-time application of this architecture in a virtual drone control scenario.

The recorded data undergoes processing to extract gain attenuation or phase offset, resulting in data points that represent unique combinations of location and

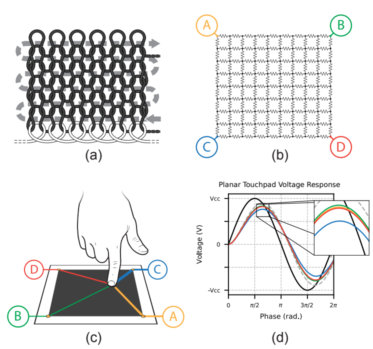


Fig. 3. CTS fabrication, circuit model, current flow, and signal measurements.

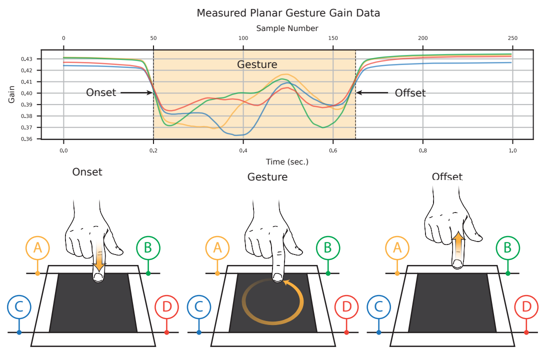


Fig. 4. Example of a gesture performed on the touchpad and the corresponding recorded data.

capacitance—except in no-touch scenarios where location cannot be determined. These data points are then analyzed as a time series to identify actions performed within a defined timing window, such as the circular swipe depicted in Figure 4, where the data is plotted over the input duration.

4 Methodology

Implementing reproducible machine learning models on wearable devices and other small electronics presents unique challenges due to limited power and storage capabilities. Traditional, complex AI models are often unsuitable for these resource-constrained environments, consuming excessive power and requiring more memory than available. This generally results in slower performance and compromised functionality, eventually leading to reproducibility issues in the absence of higher-end hardware. To overcome these limitations, we propose a shift towards minimalist architectures, which are streamlined and simplified AI models designed for efficiency. Our approach ensures reproducibility due to the minimal architectures that balance performance and efficiency, critical for reproducible research in these contexts [7]. This can result in slower performance and compromised functionality. To overcome these limitations, we propose a shift towards minimalist architectures, which are streamlined and simplified AI models designed for efficiency.

By prioritizing essential features and reducing complexity, minimalist architectures not only fit within the constraints of small devices, but also offer several advantages. Their faster data processing enables real-time decision-making, crucial for interactive applications, while their compact size frees up valuable storage space, allowing for additional features or deployment on even smaller devices. While sophisticated models like LSTMs and transformers have their merits, their complexity makes them impractical for our use case. Instead, we opt for simpler yet effective models such as fully connected networks and minimalist convolutional networks, which strike a balance between performance and efficiency. Handling complex tasks on restricted hardware resources not only enhances efficiency but also improves reproducibility, thanks to the broader accessibility of lower-end technology. This shift towards minimalist architectures represents a significant advancement in the field of machine learning, paving the way for integrating machine learning capabilities into everyday objects and ultimately enhancing their functionality and user experience.

4.1 Machine Learning Pipeline

To ensure reproducibility in developing gesture recognition for wearable technology, we designed a streamlined process to train and deploy compact AI models on an embedded device. We used the Arduino Nano 33 BLE, chosen for its low power consumption, wireless communication, and sufficient computational capabilities for on-device machine learning.

Our approach involves collecting gesture data from a knitted capacitive touch sensor and processing it using TensorFlow and Keras offline on a server. The AI model, trained to recognize specific gestures, is then compressed using TensorFlow Lite/TinyML to fit the microcontroller's resource constraints. Finally, the optimized model is integrated into the microcontroller's firmware, enabling real-time gesture interpretation on the wearable device (see Figure 5).

The key steps are as follows:

1. **Firmware Upload:** Initial firmware is uploaded to the Arduino Nano 33 BLE to enable data transmission for offline analysis.
2. **Data Collection:** The Arduino transmits a square wave signal through the smart textile. The signal is modified upon touch, and the Arduino measures the altered signal's return time to identify the touch location and gesture. This data is then stored for subsequent analysis.
3. **Model Training & Validation:** The collected data is used to train a machine learning model, which is then validated to ensure accuracy and reliability.
4. **Model Optimization:** The trained model is compressed using TensorFlow Lite/TinyML, reducing its size and complexity for deployment on the Arduino Nano 33 BLE.
5. **Firmware Integration:** The optimized model is integrated into the firmware, enabling on-device gesture recognition.
6. **Deployment:** The final firmware, incorporating the AI model, is uploaded to the Arduino, empowering the smart textile to detect and interpret gestures in real-time.

Our approach leverages minimalist architectures—streamlined AI models optimized for resource-limited devices like the Arduino Nano 33 BLE. This enables real-time responsiveness while minimizing power consumption and maintaining optimal performance.

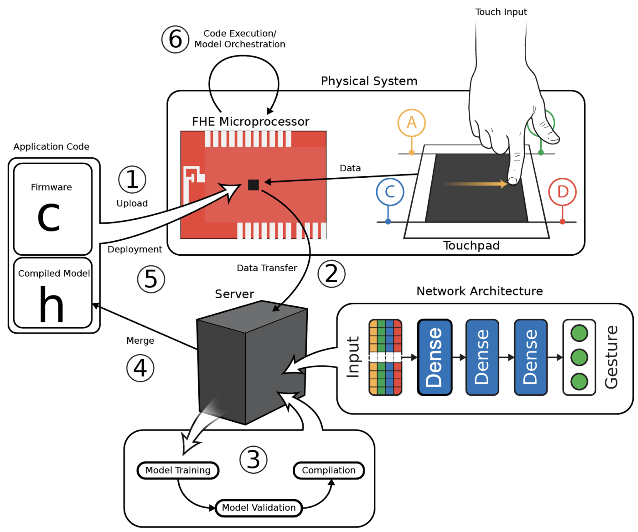


Fig. 5. Diagram of the embedded machine learning deployment pipeline. (1) The firmware application code is uploaded to the FHE microprocessor to perform data collection. (2) The data is streamed to and saved on a host PC (server). (3) The data is used to train the selected network architecture created in TensorFlow and Keras, which is validated and compiled into a header file using TensorFlow Lite/TinyML. (4) The header file is merged with the firmware. (6) The application code is reuploaded and run on the embedded hardware.

4.2 Model Architecture

The proposed model architecture is meticulously designed, with a focus on reproducibility, to address the unique challenges of classifying both single-touch tap actions and multi-touch touchpad gestures. We provide detailed algorithmic implementation and parameter settings to ensure that our approach can be easily replicated and validated by other researchers.

Single-Touch Tap Action Classification Classifying single-touch tap actions from wearable touchpad data is a unique challenge in biometric recognition and human-computer interaction. The task involves identifying subtle spatio-temporal patterns in each tap, translating raw sensor data into directional commands (e.g., north, south, east, west, and diagonals). To tackle this, we use a feedforward neural network (FNN) architecture tailored to these interactions.

The model starts with an input layer for time-series data shaped as (N, 10, 4), where N is the number of samples, 10 represents time steps, and 4 corresponds to raw sensor values from the CTS pad at each step (see Figure 6). This sequential input is flattened into a one-dimensional vector for efficient processing by the dense layers.

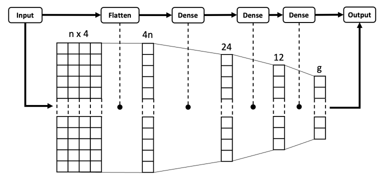


Fig. 6. Diagram of Single-Touch Tap Machine Learning Model.

Two hidden dense layers, with 64 and 32 units respectively, form the core of the FNN, extracting and transforming features using the Rectified Linear Unit (ReLU) activation function. This non-linear activation allows the model to learn complex relationships between input features and tap actions. The output layer, with 8 units and a softmax activation function, generates a probability distribution over the eight predefined classes, enabling confident classification of each tap.

Multi-Touch Touchpad Gesture Classification Recognizing signals from a wearable touchpad and classifying multi-touch gestures into eight distinct classes is significantly more complex than single-touch taps. Multi-touch gestures involve simultaneous touches and dynamic movements, requiring a model architecture that can discern the location and timing of multiple events as well as the spatial and temporal dynamics between touchpoints.

To manage this complexity, we employ a convolutional neural network (CNN) architecture (see Figure 7). The input data is structured as a sequence of 50 time steps, each with 4 features per touchpoint, representing the raw sensor values from the CTS pad. Before entering the CNN, this data is reshaped to fit the convolutional layers.

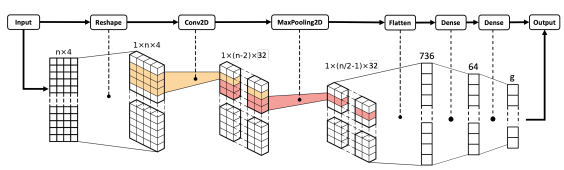


Fig. 7. Diagram of Multi-Touch Gesture Machine Learning Model.

The first convolutional layer uses 32 filters with a kernel size of (1, 4) to capture local dependencies along the temporal dimension. A 2D max-pooling layer follows, reducing spatial dimensions while retaining key features. The output is then flattened and passed through a 35% dropout layer to mitigate overfitting. A dense layer with 64 units and ReLU activation refines the features before the final classification layer, consisting of 8 units with a softmax activation function, generates a probability distribution over the gesture classes.

Classifying gestures over longer durations presents challenges due to increased variability in speed, trajectory, or pressure, introducing noise and complicating feature extraction. This variability also raises the likelihood of ambiguity between similar gestures. Additionally, defining precise gesture boundaries becomes more difficult and subjective, further complicating accurate classification over extended timeframes.

Model Training and Optimization Both the FNN and CNN models are trained using the categorical cross-entropy loss function, which is well-suited for multi-class classification problems. The Adam optimizer, an adaptive learning rate optimization algorithm, is used to efficiently update the model parameters during training. However, recognizing the potential for increased complexity and overfitting in the CNN model, a lower learning rate of 0.00005 is used for touchpad gesture classification compared to the 0.0001 learning rate used for single-touch tap action classification.

By tailoring the model architecture and hyperparameters to the specific characteristics of each gesture type, this approach aims to achieve optimal classification performance, facilitating accurate and reliable recognition of touch-based interactions in the context of biometrics and human-computer interaction.

Summary In summary, the chosen architectures—FNN for single-touch tap action classification and CNN for multi-touch gesture classification—are well-suited to the task due to their ability to capture the essential temporal and spatial dynamics of touch-based interactions. The FNN's streamlined design allows for efficient processing of sequential tap data, making it ideal for simpler, more discrete gestures. In contrast, the CNN's capability to model complex, multi-touch gestures over extended timeframes makes it highly effective for handling the intricacies of dynamic, multi-point interactions. The careful selection of architecture and hyperparameters ensures not only optimal performance but also facilitates reproducibility, making these models robust and reliable tools for advancing human-computer interaction in wearable technology.

4.3 Results and Evaluation

Our proposed models demonstrate exceptional performance in classifying both single-touch tap actions and multi-touch touchpad gestures. To enhance the reproducibility of pattern recognition research [6], we meticulously document the influence of key parameters on result quality and provide comprehensive evaluation metrics to facilitate the replication and validation of our findings by other researchers.

Single-Touch Tap Action Classification The high accuracy and F1 scores achieved in both the training and test sets, as shown in Table 1, emphasize the model's ability to effectively generalize learned patterns to unseen data, a critical factor for real-world deployment on embedded systems. The consistent performance across datasets suggests that the model is not overfitting, which reduces the risk of degraded accuracy in practical applications. Moreover, the balanced F1-scores, taking into account both precision and recall, underline the model's robustness even in scenarios where class distributions might be imbalanced. This is particularly relevant for biometric applications, where certain tap actions might be less frequent than others.

Table 1. Performance of Single-Touch Tap Action Classification

Metric	Training Set (70%)	Test Set (30%)
Accuracy	93.57%	90.37%
F1-Score	0.9668	0.9494

Multi-Touch Touchpad Gesture Classification While the accuracy on the test set for multi-touch gestures is slightly lower than that for single-touch actions, as shown in Table 2, it remains notably high, especially considering the increased complexity and variability inherent in multi-touch interactions. The F1-scores, although experiencing a minor decrease on the test set, still demonstrate a commendable balance between precision and recall, highlighting the model's ability to effectively distinguish between different gesture types, even in the presence of potential noise or ambiguity in the input data.

Table 2. Performance of Multi-Touch Touchpad Gesture Classification

Metric	Training Set (70%)	Test Set (30%)
Accuracy	92.35%	80.95%
F1-Score	0.9577	0.7955

The strong performance of both models on the test set is particularly encouraging in the context of embedded device deployment. It suggests that the proposed architectures, despite their lightweight nature, can achieve robust and accurate gesture recognition in real-time, even with the computational and memory constraints typically associated with embedded systems. This has significant implications for the development of novel human-computer interaction modalities, enabling seamless and intuitive gesture-based control of devices in a wide range of applications, from biometrics and security to consumer electronics and assistive technologies.

Furthermore, the high accuracy and F1-scores achieved by these models underscore their potential for integration into capacitive touch sensors, which are becoming increasingly prevalent in embedded devices due to their slim form factor and low power consumption. By accurately recognizing a diverse set of gestures from capacitive touch input, these models can unlock new possibilities for user interface design and enhance the overall user experience.

5 Example Application: Virtual Drone Control with Single- and Multi-Touch Taps and Gestures

To showcase the practicality and effectiveness of our proposed lightweight gesture recognition models, we developed a 3D drone flight simulator using the Unity game engine. This immersive simulator not only serves as a proof-of-concept for applying capacitive touch sensors (CTS) in real-world human-computer interaction (HCI) scenarios but also provides a platform for evaluating the performance and usability of our gesture recognition algorithms in a complex and dynamic task environment.

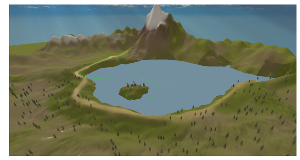
The simulator environment is designed to be both visually engaging and technically challenging. It consists of a virtual valley with a diverse terrain that includes trees, a lake, and an island (see Figure 8a). Players take control of a police drone tasked with collecting a series of gold coins scattered along a pre-defined path within a 60-second time limit. To add a layer of realism and complexity, the drone's movement is governed by a physics engine that simulates gravity, flight mechanics, and collisions with the environment. This creates a dynamic and interactive experience that demands precise and responsive control from the player.

Two distinct capacitive touch sensors serve as the primary input devices for controlling the drone, demonstrating the versatility of our gesture recognition system. The first sensor is a larger 4" x 8" multi-touch pad divided into two functional regions (see Figure 8c). The left side is dedicated to recognizing single-touch taps corresponding to cardinal directions (North, East, South, West), while the right side recognizes single-touch taps for spatial directions (up, down, left, right). Additionally, this sensor detects multi-touch tap combinations across both sides, enabling simultaneous control of multiple degrees of freedom, such as descending while moving forward ("forward & descend"). The second sensor is a smaller 2" x 4" multi-touch pad that focuses on recognizing longer-duration gestures. Specifically, it is trained to detect clockwise and counterclockwise circular motions, which are mapped to toggling the drone's engine on and off, thereby adding another dimension to the control scheme.

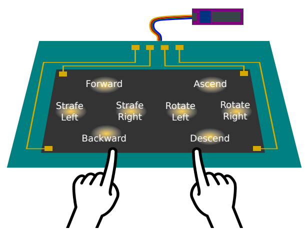
The mapping of taps and gestures to the drone's four controllable degrees of freedom (ascent/descent, forward/backward, strafing, and rotation) is designed to be intuitive and natural, promoting a seamless user experience. To assess the effectiveness of the CTS-based control interface, we also implemented traditional keyboard controls as a baseline for comparison. In user testing, participants reported that the CTS interface was easy to learn and use, offering a level of control and responsiveness comparable to the keyboard, even in the challenging task of navigating the drone along the designated path.

A video demonstrating the gameplay using the CTS sensors is available in the Supplementary Material, providing a visual illustration of the smooth and

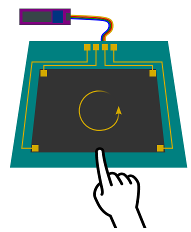
(a) 3D drone simulator game controlled by CTS sensor



(b) Bird's eye view of the terrain



(c) Multi-touch 8-direction tap controller



(d) Single-touch gesture controller

Fig. 8. Visualization of 3D drone simulator and its various control interfaces, including different types of touch controllers and a terrain view.

intuitive interaction facilitated by our gesture recognition system. This demonstration not only validates the performance of our models in a real-world application but also underscores the potential of capacitive touch sensing and gesture recognition to revolutionize human-computer interaction across diverse domains. By enabling seamless and natural control of complex systems through intuitive gestures, our technology opens new possibilities for biometric authentication, assistive technologies, gaming, and other fields where intuitive and efficient interaction is paramount.

6 Discussion

This study demonstrates the effectiveness of a minimalistic neural network architecture for real-time gesture recognition on capacitive touch sensors integrated into smart textiles. By successfully classifying both single-touch taps and complex multi-touch gestures with high accuracy (Tables 1 and 2), we have validated the potential of this approach for resource-constrained wearable devices, aligning with the increasing demand for efficient and seamless human-computer interaction. Our results highlight the model's ability to generalize from training data to unseen test data, a critical factor for real-world deployment. The balanced F1-scores across different classes indicate robustness even in scenarios with imbalanced class distributions, a common challenge in biometrics. The successful implementation of this lightweight architecture in a drone control simulation further emphasizes its practicality. The intuitive nature of the gesture-based control suggests that this approach could revolutionize HCI in various domains, including assistive technologies. Additionally, the integration with low-power capacitive touch sensors opens new avenues for unobtrusive wearable devices that seamlessly capture and interpret human intent.

While promising, this research has several limitations that we detail for the sake of ensuring reproducibility. The current gesture vocabulary is limited to a predefined set of 10 gestures, and we outline known difficult cases and potential future improvements. We provide guidance on integrating our source code into other frameworks to encourage further exploration and validation by the research community. Additionally, the system's real-time responsiveness has not been fully evaluated under varying conditions such as different fabric types, sweat levels, or movement artifacts. The potential impact of these factors on gesture recognition accuracy requires further investigation. Furthermore, the current model lacks personalization and adaptability features, meaning it does not adjust to individual users' unique gesture styles or preferences.

Moving forward, we are committed to enhancing the reproducibility of our architecture by deploying it directly onto embedded devices, exploring advanced signal-processing algorithms to improve data quality, and investigating power optimization strategies for real-world applications. We encourage further exploration and validation by the research community through the integration of our publicly available source code and detailed installation procedures. We aim to expand the gesture vocabulary to include more complex and nuanced movements, and to explore personalization techniques to adapt to individual users' gesture styles and preferences. Additionally, we plan to conduct user studies to gather feedback and assess the usability and intuitiveness of the gesture-based interface, which will guide future iterations of the system. Future research will also focus on refining the model architecture to further improve its performance on complex multi-touch gestures, potentially incorporating techniques such as recurrent neural networks or transformers to better capture temporal dependencies. By addressing these challenges and continuing to push the boundaries of gesture recognition on embedded devices, we believe this work contributes significantly to the advancement of smart textiles and their potential to revolutionize HCI in fields like biometrics, healthcare, and beyond.

Acknowledgments. Effort sponsored by the U.S. Government under Other Transaction number HQ00342190017 between NextFlex and the Government. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the U.S. Government.

References

1. Freire, R., Honnet, C., Strohmeier, P.: Second skin: An exploration of etextile stretch circuits on the body. In: Proceedings of the Eleventh International Confer-

ence on Tangible, Embedded, and Embodied Interaction. pp. 653–658 (2017)
2. Hughes, D., Profita, H., Correll, N.: Switchback: an on-body rf-based gesture input device. In: Proceedings of the 2014 ACM International Symposium on Wearable Computers. pp. 63–66 (2014)
3. Hughes, D., Profita, H., Radzihovsky, S., Correll, N.: Intelligent rf-based gesture input devices implemented using e-textiles. Sensors **17**(2), 219 (2017)
4. Inaba, M., Hoshino, Y., Nagasaka, K., Ninomiya, T., Kagami, S., Inoue, H.: A full-body tactile sensor suit using electrically conductive fabric and strings. In: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS'96. vol. 2, pp. 450–457. IEEE (1996)
5. Karrer, T., Wittenhagen, M., Lichtschlag, L., Heller, F., Borchers, J.: Pinstripe: eyes-free continuous input on interactive clothing. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. pp. 1313–1322 (2011)
6. Kerautret, B., Colom, M., Krähenbühl, A., Lopresti, D., Monasse, P., Perret, B.: Reproducible Research in Pattern Recognition: Fourth International Workshop, RRPR 2022, Montreal, Canada, August 21, 2022, Revised Selected Papers, vol. 14068. Springer Nature (2023)
7. Kerautret, B., Colom, M., Krähenbühl, A., Lopresti, D., Monasse, P., Talbot, H.: Reproducible Research in Pattern Recognition: Third International Workshop, RRPR 2021, Virtual Event, January 11, 2021, Revised Selected Papers, vol. 12636. Springer Nature (2021)
8. McDonald, D.Q., Valett, R., Saunders, L., Dion, G., Shokoufandeh, A.: Recognizing complex gestures on minimalistic knitted sensors: Toward real-world interactive systems. arXiv preprint arXiv:2303.10336 (2023)
9. McDonald, D.Q., Vallett, R., Solovey, E., Dion, G., Shokoufandeh, A.: Knitted sensors: Designs and novel approaches for real-time, real-world sensing. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies **4**(4), 1–25 (2020)
10. Olwal, A., Starner, T., Mainini, G.: E-textile microinteractions: Augmenting twist with flick, slide and grasp gestures for soft electronics. In: Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems. pp. 1–13 (2020)
11. Parzer, P., Perteneder, F., Probst, K., Rendl, C., Leong, J., Schuetz, S., Vogl, A., Schwoediauer, R., Kaltenbrunner, M., Bauer, S., et al.: Resi: A highly flexible, pressure-sensitive, imperceptible textile interface based on resistive yarns. In: Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology. pp. 745–756 (2018)
12. Shi, J., Liu, S., Zhang, L., Yang, B., Shu, L., Yang, Y., Ren, M., Wang, Y., Chen, J., Chen, W., et al.: Smart textile-integrated microelectronic systems for wearable applications. Advanced materials **32**(5), 1901958 (2020)
13. Sundholm, M., Cheng, J., Zhou, B., Sethi, A., Lukowicz, P.: Smart-mat: Recognizing and counting gym exercises with low-cost resistive pressure sensing matrix. In: Proceedings of the 2014 ACM international joint conference on pervasive and ubiquitous computing. pp. 373–382 (2014)
14. Vallett, R., McDonald, D.Q., Dion, G., Kim, Y., Shokoufandeh, A.: Toward accurate sensing with knitted fabric: Applications and technical considerations. Proceedings of the ACM on Human-Computer Interaction **4**(EICS), 1–26 (2020)
15. Vallett, R., Young, R., Knittel, C., Kim, Y., Dion, G.: Development of a carbon fiber knitted capacitive touch sensor. MRS Advances **1**(38), 2641–2651 (2016)
16. Younes, B.: Smart e-textiles: A review of their aspects and applications. Journal of Industrial Textiles **53**, 15280837231215493 (2023)