A Wearable Smart Glove and Its Application of Pose and Gesture Detection to Sign Language Classification

Joseph DelPreto*[1], Josie Hughes*[2], Matteo D'Aria[3], Marco de Fazio[3], and Daniela Rus[1]

***Abstract*—Advances in soft sensors coupled with machine learning are enabling increasingly capable wearable systems. Since hand motion in particular can convey useful information for developing intuitive interfaces, glove-based systems can have a significant impact on many application areas. A key remaining challenge for wearables is to capture, process, and analyze data from the high-degree-of-freedom hand in real time.**

We propose using a commercially available conductive knit to create an unobtrusive network of resistive sensors that spans all hand joints, coupling this with an accelerometer, and deploying machine learning on a low-profile microcontroller to process and classify data. This yields a self-contained wearable device with rich sensing capabilities for hand pose and orientation, low fabrication time, and embedded activity prediction.

To demonstrate its capabilities, we use it to detect static poses and dynamic gestures from American Sign Language (ASL). By pre-training a long short-term memory (LSTM) neural network and using tools to deploy it in an embedded context, the glove and an ST microcontroller can classify 12 ASL letters and 12 ASL words in real time. Using a leave-one-experiment-out cross validation methodology, networks successfully classify 96.3% of segmented examples and generate correct rolling predictions during 92.8% of real-time streaming trials.

***Index Terms*—Wearable Robotics; Gesture, Posture, and Facial Expressions; Soft Sensors and Actuators; Embedded Systems.**

I. Introduction

WEARABLE devices have many applications ranging from health monitoring and analysis to virtual and augmented reality. These devices must be robust, have a small form factor, and provide significant information content such as postures or motion. This is particularly challenging for wearable devices focused on the hand and wrist, due to the many joints and degrees of freedom that must be captured simultaneously. This also raises a secondary challenge of interpretation and analysis, requiring methods to classify or understand this complex sensory data in real time.

Due to the many application domains for smart gloves, there have been a variety of innovative approaches to capturing postural or force information. These include research and commercial devices based on strain, capacitance, and piezoresistance. For example, stretchable electronics can provide high-resolution strain sensing [1], and small soft capacitive sensors can detect micro-gestures from small postural changes [2]. Combining such techniques with learning pipelines can yield capable systems; neural networks can accurately reconstruct hand poses based on capacitive sensing [3], or identify grasp types and modalities from a high-resolution knitted piezoresistive glove [4]. Multiple modalities can also aid hand gesture classification, such as by combining muscle activity with pressure-sensitive arrays [5] or with soft electronics that measure strain and pressure [6]. Commercial gloves such as the Manus system [7] can also provide high-fidelity pose information for virtual reality or other applications, although they are typically expensive and require intricate electronics.

Such gloves showcase promising sensory technologies and learning architectures, but various challenges remain. These include scalable and accessible fabrication, and the ability to classify data online with small wearable microcontrollers.

Taking a step towards these goals, this paper presents the smart multi-modal glove system shown in Fig. 1. It features on-board classification, and is based on a commercially available glove. We sensorize its strain-sensitive conductive knit to form 16 resistive sensors spanning the hand's degrees of freedom. We combine this pose information with an accelerometer on the back of the hand. A small STM32 microcontroller processes signals and evaluates a neural network in real time.

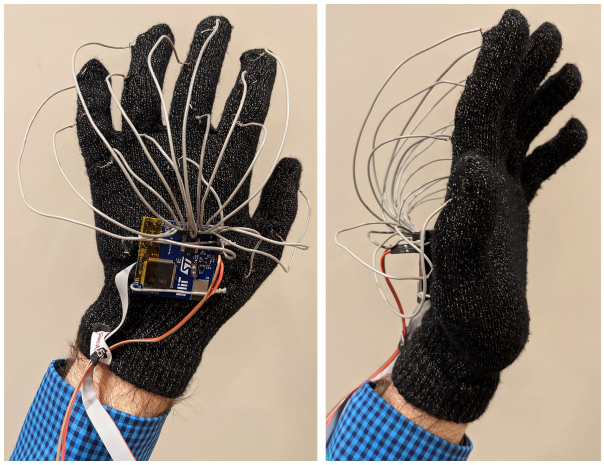

Fig. 1: The self-contained wearable system is based on a commercially available conductive knit glove. Wires are added to the back to create strain sensors. The board contains an accelerometer, and performs all signal processing, real-time neural network evaluations, and communications.

Manuscript received: 02/24/22; Revised 05/23/22; Accepted 06/18/22.

This paper was recommended for publication by Editor Jee-Hwan Ryu upon evaluation of the Associate Editor and Reviewers' comments.

This work was supported by STMicroelectronics and the Gwangju Institute of Science and Technology (GIST).

[1] MIT Distributed Robotics Lab, Cambridge, MA, USA
delpreto@csail.mit.edu, rus@csail.mit.edu

[2] EPFL CREATE Lab, Lausanne, Switzerland
josie.hughes@epfl.ch

[3] STMicroelectronics, Burlington, MA, USA
matteo.daria@st.com, marco.de-fazio@st.com

* These authors contributed equally to this work.

Videos are available at https://people.csail.mit.edu/delpreto/smart-glove

Digital Object Identifier (DOI): see top of this page.

To demonstrate the capabilities of the glove, we perform rolling time-series classification of 24 letters and words from American Sign Language (ASL) [8]. This vocabulary has been explored for wearable systems by many past research endeavors, such as [9]–[11]. It offers impactful applications such as improving communication between people or inspiring alternative human-computer interfaces, especially involving deaf participants or challenging auditory environments. The vocabulary also includes both static poses and dynamic motions. Past work has often focused on only one of these dimensions or has not been implementable on microcontrollers, which limits the wearability and practicality.

The current work builds on past research and develops an embedded learning system that leverages both strain and acceleration sensing to perform real-time pose and gesture classification. In particular, its key contributions include:

- Sensorizing a commercially available strain-sensitive glove for ease of fabrication and adding an accelerometer, to capture both hand pose and gesture information;
- Developing a neural network pipeline to detect time-series events, which is pre-trained and then embedded on a microcontroller to run in real time;
- Preliminary experiments using a vocabulary of 24 ASL words and letters, including static and dynamic gestures;
- Classification performance results evaluated on unseen sessions of wearing the glove, using offline segmented examples or online rolling predictions.

The remainder of this paper first focuses on sensorization and electronics. It then describes the learning pipeline including the experimental protocol, network, and training. Results then investigate rolling gesture classification. The paper concludes with a discussion and future directions.

II. Related Work

With advances in soft or compliant strain sensors, there has been significant growth in the availability and focus on wearable devices [12]. This includes the development of commercial solutions [13]. Due to the use of hands for manipulation, communication, and more, there is a strong focus on developing smart gloves to detect hand poses and interactions [14]. Many previous approaches focus on applying soft or compliant sensors to a glove to allow the deformation of each degree of freedom of the hand to be measured. This includes capacitive [15], resistive [16], ionic [17], or even bi-modal sensing approaches [18]. Although these have shown significant potential for pose reconstruction and motion detection, there are remaining challenges regarding scalability of the number of sensors and fabrication.

Knitted sensorized gloves are particularly attractive since they do not require post-processing or the further addition of sensors to the structure. This has the potential to make the fabrication process rapid and repeatable, and also for the number of sensors or sensory inputs to be high. Previous work has demonstrated the capabilities of custom knit-based strain sensors [19], identifying how fabric and knit parameters can change the properties of the sensor [20]. Advances in digital knitting systems have enabled many customized sensorized gloves. This includes systems that allow selecting specific yarns to design sensor characteristics and regions on the knitted surface of the glove [21]. Such flexibility to customize knitted regions or patterns has been shown to enable a variety of applications [22]. Additional approaches include processing yarns to allow the formation of sensors through knitting, such as electrospun fibers [23] or highly conductive polymers [24].

While digital knitting can leverage custom materials and form factors, using "off-the-shelf" gloves can provide an even lower barrier for fabrication and utility. The current work builds on an initial exploration of adapting a commercial glove to create a mesh network of resistive sensors [16]. Compared to this past study, the current work increases the capabilities by adding on-board signal processing and machine learning, enhances the ability to capture dynamic motions, simplifies fabrication by using fewer sensors, and explores a new application domain. The presented device can be a stand-alone wearable system rather than one that streams sensor data to a laptop. It also omits adding pressure-sensing infrastructure in favor of a more streamlined glove based on the commercial knit. It adds an accelerometer to capture orientations and motions relative to the world, rather than only sensing in a hand-centric frame. Finally, a new learning pipeline and experimental paradigm demonstrate applicability to real-time gesture and pose detection.

III. Fabrication and Electronics

The deployed system can be considered as three main aspects: the sensorized resistive glove and accelerometer, custom electronics with ST technologies for data acquisition and classification, and a neural network running on the microcontroller. This section focuses on sensors and electronics, while the next sections discuss the learning pipeline.

A. Resistive Knit

This work uses a commercially available knitted conductive glove designed to work with capacitive touch screens, namely the Original Sport glove by Agloves [25]. Due to the silver threads within the knit, the glove has a resistance of approximately $5\,\Omega/\mathrm{cm}$ that varies when a strain is applied. Due to the knit pattern, the material can undergo approximately 70% strain without permanent deformation. Adding electrode connections to this "off-the-shelf" glove enables rapid creation.

As explored in [16], the sensing properties of the knit can be characterized by mounting a section of the material in an Instron machine and cyclically applying 0-50% strain. Since the knit pattern is directional, the resistance response curve depends on the strain direction as shown in Fig. 2. It exhibits high sensitivity to strain *along* the knit, and low sensitivity to strain *across* the knit. This is suitable for detecting common hand deformations, since the interphalangeal finger joints cause bending along the knit. Additionally, since the glove is fully conductive, self-collisions between parts of the hand will cause significant resistance changes regardless of bending direction. The response along the knit exhibits high repeatability between testing cycles as shown by the shading in Fig. 2, which indicates that resistance varies consistently

This article has been accepted for publication in IEEE Robotics and Automation Letters. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/LRA.2022.3191232

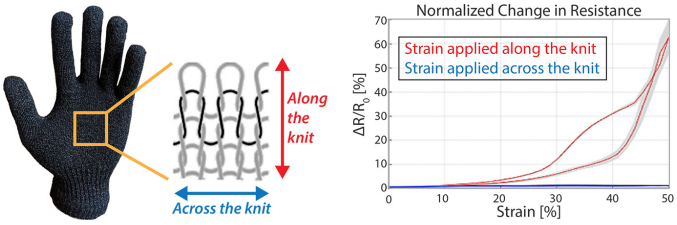

Fig. 2: The response of the glove material was characterized using 80 cycles of straining either along or across the knit. Adapted from [16].

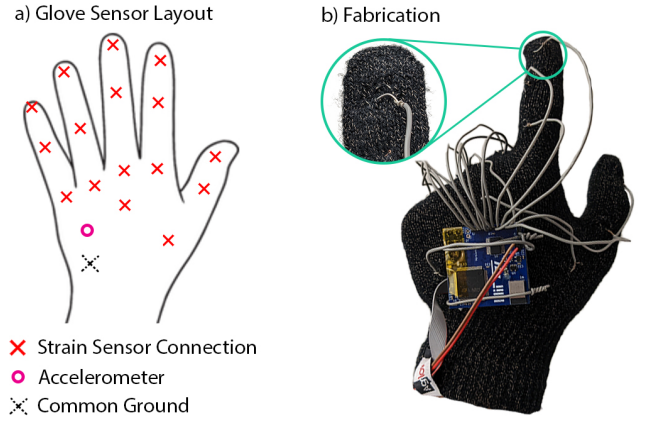

Fig. 3: Wires are connected to 17 points on the back of the hand to form strain sensors on each finger segment and the hand, including a common ground. An accelerometer is also on the PCB on the back of the hand. (a) depicts the locations schematically, while (b) shows the connection method.

during stretching. However, it also exhibits high hysteresis; this is likely due to the nonlinear behavior of the knitted fibers. The average response and recovery times were found to be 0.35 s and 0.8 s, respectively; this is sufficient for the gesture detection currently explored, but may hinder the use of the glove in applications that require faster responses.

B. Glove Sensorization

To enable hand pose identification, we use this conductive glove to form strain sensors spanning all joints of the hand. A connection is made by simply weaving approximately 2 cm of stripped wire through the knit in a rough loop as shown in Fig. 3. Measuring the resistance between two such connections forms a strain sensor due to the knit's strain-sensitive response. In contrast to [16] which considered all pairs of read-out points, the current work simplifies processing while still detecting motions of all joints by only considering the resistance between each of 16 read-out points and a common ground. Connections are placed on each finger segment to maximize the information content.

Wires are attached to the microcontroller with sufficient slack to not hinder finger motion. In the future, they could also be hidden and protected by a non-conductive glove layer.

C. Electronics Design

To achieve low-noise readings from the strain sensors in a compact form that can be easily worn on the hand, a custom electronics board was designed. It incorporates the strain sensor reads-outs, an ST microcontroller, and a 3-axis accelerometer. At the heart of the board is an STM32H7 microcontroller, which performs data acquisition, signal processing, and real-time neural network evaluation.

The readout for the 16 strain sensors has been designed to maximize stability and sampling frequency. The strain sensors in the glove are connected between a constant current source and ground. The voltage drop generated by this current depends on the sensor resistance. It is amplified, adjusted for offset, and acquired by the STM32H7 analog-to-digital converter (ADC). Two digital-to-analog converters (DACs) of the STM32H7 are used to control the analog front-end that regulates the amount of current generated and the amount of offset removed from the measurement.

To use the same circuitry for all sensors, two 16-channel switches iteratively connect to each sensor; this multiplexing thus performs sequential acquisition. This circuit, together with the STM32H7, allows for the current and offset voltage to be specified for each single sensor within fixed boundaries. This can maximize the resolution of measurements for a range of sensors which may not be homogeneous, may have different sensitivities, or may have different base resistances.

The 3-axis accelerometer connects to the STM32H7 via I^2C.

All data acquisition, signal processing, and computation is performed on-board. The board also contains a Bluetooth module, although the current tests used a USB connection to report classification results and for power. Simply activating the wireless transmission option and adding a small battery allows the device to be untethered and self-contained.

IV. Gesture Vocabulary

The chosen vocabulary consists of poses and gestures representing 24 letters, words, or phrases of ASL. This highlights the capabilities of the sensorized glove to detect both static and dynamic gestures, and demonstrates a potential application that could make human interactions more natural by translating ASL into text or speech in real time.

Fig. 4 illustrates the 11 static poses and 13 dynamic gestures. ASL naturally showcases the necessity of detecting both hand pose and motion. Certain pairs of vocabulary entries, such as *I* and *J* or *A* and *Sorry*, have the same pose but different dynamics. Other sets, such as *Eat*, *Home*, and *Thank You*, have subtle differences in hand poses, orientations, or motion directions. Some gestures such as *Please* or *Yes* are periodic motions that could have varying numbers of repetitions. Most of the letters are static poses without motion. Altogether, the chosen vocabulary thus probes the system's ability to combine pose and motion information for multi-class gesture detection.

V. Training Data Collection

To train a classifier for this task and explore its robustness, we performed 7 training experiments. These spanned multiple days, and the glove was always removed between experiments. Between same-day sessions, it was removed for an average of 1.4 hours. This allows us to explore the robustness of the classifier across episodes of use, including effects such as the glove being worn or stretched differently on the hand, general

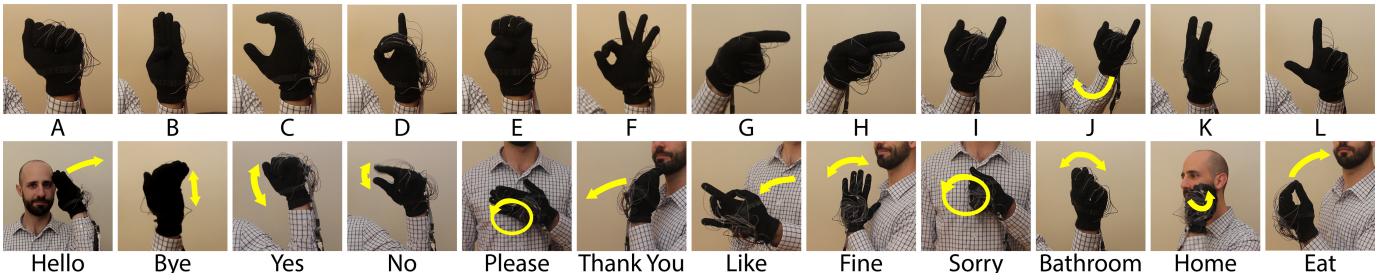

Fig. 4: A vocabulary of 24 ASL letters and words was selected, which requires identifying a range of poses and dynamic motions. This yields an informative corpus for evaluating the combination of strain-based pose information and accelerometer-based motion information featured by the embedded glove system.

Fig. 5: The pipeline processes collected training data to create a neural network that can classify segmented examples or streaming real-time data.

wear-and-tear, and varying hand temperatures or moisture levels. The current study involved a single ASL novice; this is sufficient to demonstrate feasibility of the wearable system, but future work should expand the subject pool. Each experiment recorded 10 trials of each letter or word, with blocks recorded sequentially. This provides 70 total examples of each gesture.

Data was recorded continuously throughout the experiments. A 2.5 s cue was presented for each trial, which started shortly after the correct starting position was achieved. A rest pose was assumed once the cue ended. The status of the gesture cue was recorded along with each data sample, creating square waves of ground truth labels. Raw ADC values were recorded from the 16 strain channels and the 3 acceleration channels via USB serial at 100 Hz.

This data was then processed and used for training and evaluation as described by Fig. 5 and the following sections.

VI. Data Processing and Feature Extraction

To create training data for the classifier, recorded data is segmented into labeled examples, conditioned, and transformed into feature vectors. Data augmentation is also used to improve the robustness and accuracy of the trained network, especially when applied to real-time streaming data.

A. Segmentation into Labeled Examples

Based on the stream of ground truth labels, a 2 s window centered around each trial is extracted. This yields positive gesture examples with 200 timesteps each. In addition, examples of not making any gesture are important for the network to learn, and such unstructured activity may be highly variable. Examples are extracted as 2 s windows in three possibly overlapping locations: ending 1 second before a trial label starts, starting 1 second after the label ends, and centered between successive trials. Overall, each 10-trial block of a word/letter yields 10 positive examples and 29 baseline examples.

B. Data Augmentation

Although the network will be trained on extracted labeled examples, it will ultimately be evaluated in a streaming fashion by classifying rolling windows of real-time data. In addition, dynamic gestures may be performed at different speeds. To help address this, data augmentation is used to improve the network's robustness to the timing of gestures.

Time-shifted synthetic examples aim to encourage the network to accommodate examples that are not perfectly centered in its classification window. For each window of a positive example, 3 new windows are defined that are shifted earlier and 3 new windows are defined that are shifted later. Each shift is a random duration between 50 ms and 400 ms. The same procedure is applied to each baseline window as well, but with only 1 shifted window in each direction.

Time-scaled synthetic examples aim to improve robustness to varying gesture speeds. Compressing and dilating time is used to represent faster or slower gestures; the timestamps from the entire experiment are scaled by a chosen factor, and then the data is resampled with linear interpolation to restore a 100 Hz sampling rate. The 200 samples centered around the original centered sample of a window is then used as the new synthetic example. This procedure is performed 6 times for each window of a positive example: 3 times that scale time by a random factor between 0.8 and 0.95, and 3 times that use a random factor between 1.05 and 1.2. Note that the entire experiment is resampled for each augmentation, since extra data on each side of the original window will be needed when speeding up the gesture. This augmentation is not performed for baseline examples, as the features are generally not time dependent. Note that the current pipeline does not address scaling of acceleration magnitudes to more completely simulate varying gesture motion speeds.

C. Feature Extraction: Smoothing and Normalization

Data within each original or augmented window is then preprocessed to generate features. To simplify the pipeline and allow the network to uncover useful characteristics, processed data is passed to the network directly instead of manually defining a reduced set of features. Both strain and accelerometer data are used, since the ablation results of Section VIII confirm that both provide valuable information.

Firstly, each strain channel is smoothed by a moving mean with a trailing window spanning 0.1 s (10 timesteps). This helps remove any high-frequency noise or outliers.

To make the classifier robust to short-term or long-term drift in the strain sensors while also avoiding calibration routines, the strain values are dynamically normalized on a rolling basis. For each 2 s window, the minimum and maximum values across all strain channels are computed, and then all values are shifted down by this minimum and scaled by this range. The new strain values in each window will thus be between 0 and 1. Jointly shifting and scaling all channels by the same factor preserves the relative magnitudes between channels. Computing these offsets and factors on a rolling basis can accommodate drifts throughout experiments or across days due to effects such as the glove's hysteresis or fit on the hand, while avoiding tuned factors or dedicated calibration periods.

For the accelerometer, recorded data is shifted and normalized according to the bounds of the ADC outputs. Constant values are used here instead of a rolling approach since both absolute and relative magnitudes embed useful information, such as motion speeds and the direction of gravity.

Feature matrices were then created from each window by concatenating all 16 strain readings with the 3 accelerometer readings. This yields a 200 x 19 matrix for each labeled example, with its values ranging between 0 and 1.

VII. Neural Network Training

To classify the time series data, we use a long short-term memory (LSTM) recurrent neural network. Since LSTMs have feedback connections to process sequences of data, they are well-suited to our task of classifying poses and motions.

The network accepts a 200 x 19 feature matrix representing a sequence of strain and accelerometer readings. It then has a single LSTM layer of size 100, a 20% dropout layer, and a dense output layer with softmax activations. The output has 25 classes: the 24 letters and words, and a baseline class representing that no gesture is being made.

This architecture was designed to be relatively lightweight to facilitate evaluation on a resource-limited microcontroller, but the size of the single layer was chosen to be large enough to discover useful characteristics in the time-series feature matrices. The dropout layer aims to reduce overfitting during training. Alternative structures can be explored in the future, but the current pipeline is demonstrated to be sufficient for an initial exploration of the glove's capabilities.

We use a leave-one-experiment-out 7-fold cross validation strategy for training and evaluation. All examples from an experiment are used as the test set, such that the network will be tested on data from an episode of wearing the glove that did not influence the training at all. Each experiment is iteratively treated as the holdout experiment, so 7 different networks are trained. Using each experiment as a test set instead of using randomized *k*-fold cross validation helps avoid data leakage between training and testing sets, since data within a session is likely correlated along such aspects as user behavior or glove properties. The selected procedure aims for a more robust evaluation by simulating performance that would be expected on a new day of using the glove without network retraining.

Each test set has 5,208 examples. This includes 3,120 positive examples that are originals, time-shifted augments, or time-scaled augments. The set also includes 2,088 original or time-shifted baseline examples. The remaining 6 datasets are then split into training and validation sets, with the validation set having the same size as the test set. This corresponds to the training set having 26,040 examples. The random split into training and validation sets is implemented to maintain the original proportions of labels.

While the above procedure includes all data augmentation examples, analogous networks were also trained on corpuses that left out time-shifted examples and/or time-scaled examples. This facilitates evaluation of how the data augmentation impacts performance. Similarly, networks were trained using only strain-based features (200 x 16 inputs) or only accelerometer-based features (200 x 3 inputs) to facilitate ablation results exploring the impact of multiple modalities.

Each network was trained for 50 epochs using a batch size of 32. The network and training process were implemented in Python 3.9 using version 2.5 of TensorFlow and Keras.

A. Embedded Deployment

The ST CUBE-AI software converts the pre-trained network to embedded code for the STM32 microcontroller. While the current network is lightweight enough to be loaded directly, the software offers options to trade off memory usage, speed, and accuracy; this can be critical if using a more resource-constrained device or a larger network. To optimize memory, input and output buffers can be allocated in the same space as activations by overwriting data once it is no longer needed; this has no expected impact on accuracy, but may decrease speed. Dense layers can also be compressed via floating-point quantization by generating lookup tables, but this may decrease accuracy. Section VIII-C explores these options.

VIII. Results and Discussion

We consider three scenarios to evaluate the system: classifying segmented examples, streaming classifications using recorded data to simulate real-time behavior, and the embedded implementation on the microcontroller. Throughout the discussion, distributions are often summarized as $\mu \pm \sigma$, where μ is the mean and σ is the standard deviation.

A. Performance on Segmented Examples

As described in Section VII, networks were trained using leave-one-experiment-out cross validation. Using data augmentation and both strain and acceleration features, training and validation set accuracies averaged $99.2\% \pm 0.2\%$ and $99.1\% \pm 0.3\%$, respectively. Each network was then evaluated on the segmented examples from its held-out experiment performed at a different time after glove removal. Accuracies on these test sets averaged $96.3\% \pm 2.1\%$, and the blue bars of Fig. 6 illustrate the individual results. Some variation is expected due to effects such as how the glove is positioned and stretched, skin conductivity, or user behavior. The relatively consistent performance across all held-out experiments is thus promising and suggests that overfitting was mitigated. Future investigations with multiple users and longer-term wear-and-tear should further explore this generalizability.

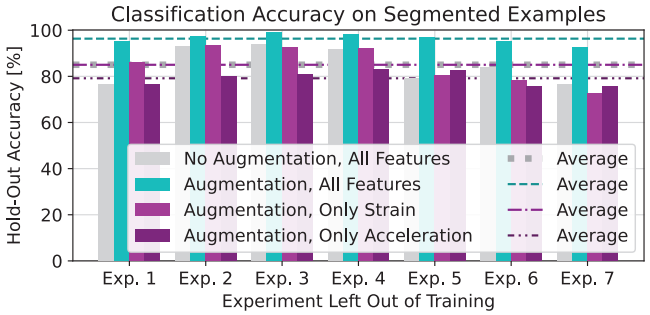

Fig. 6: Each experiment was iteratively left out of the training process, and networks were evaluated on the left-out trials. The second bar of each group uses the full pipeline; others explore removing augmentation or features.

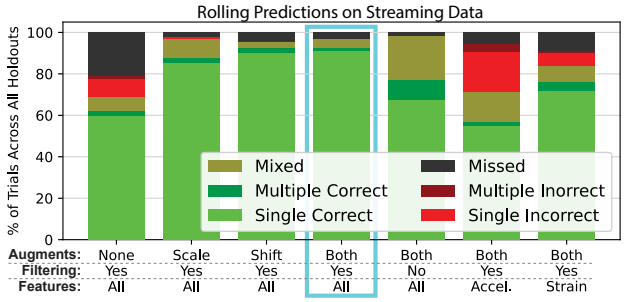

Fig. 7: Classifying a rolling buffer at every timestep simulates real-time performance, then results summarize the predictions that would have been made while each true gesture was performed. The center outlined bar represents the full selected pipeline, while the others ablate various aspects.

Fig. 6 also summarizes ablation studies that probe aspects of the presented learning pipeline. Removing data augmentation decreased test set accuracy to $85.0\% \pm 7.3\%$. Using time-shifting alone yielded $93.5\% \pm 3.3\%$, and time-scaling alone yielded $94.2\% \pm 3.2\%$. Combined with the results of the full pipeline, this suggest that both types of augmentation improved robustness and in complementary ways.

Only using strain-based features yielded test set accuracies averaging $85.0\% \pm 7.5\%$, and only using acceleration-based features yielded $79.2\% \pm 3.0\%$. This suggests that both types of features provide useful information for classifying static and dynamic gestures, and that the network successfully leveraged these complementary modalities.

B. Performance on Streaming Data

1) Simulating Real-Time Behavior: While classification of segmented examples is a good indication of performance, the networks will ultimately be used in a rolling fashion on streaming data. To simulate this using recorded data, a feature matrix is created for every timestep of the experiment at 100 Hz, using trailing 2 s windows. To mimic the microcontroller's ability to evaluate the embedded network at 5 Hz, every 20^{th} matrix is classified and a zero-order hold is applied between the results.

To help filter spurious predictions such as during a static pose at the beginning of a dynamic gesture, we maintain a trailing rolling window of four predictions (0.8 s). A filtered prediction is outputted if all of them agree.

When a user makes a gesture, the streaming predictions would ideally create a single pulse of correct labels lasting one or more timesteps. To assess this, we compare rising edges of the predicted label sequence with the sequence of ground truth cues. If a rising edge is between 1.5 s before the start of the ground truth label and 2 s after it ends, then the edge is associated with that cue. Each true gesture may thusly be matched with 0, 1, or many rising-edge predictions.

2) Aggregated Streaming Performance: Averaging across all holdout experiments, the filtered networks made a single, correct rising-edge prediction during $91.2\% \pm 8.0\%$ of the cued windows. Multiple predictions, all for the correct gesture, were made during $1.6\% \pm 2.3\%$ of the cues. There were no trials in which only incorrect predictions were made, although $4.1\% \pm 6.9\%$ of the trials had both correct and incorrect predictions. $3.1\% \pm 3.6\%$ of trials were missed altogether.

Fig. 7 illustrates these results and explores the impact of parts of the learning pipeline. The first four bars demonstrate that adding either type of data augmentation converts many trials that were missed or incorrect into trials that have a single correct prediction, and using both yields the best performance. This suggests that augmentation successfully improved network robustness in the streaming scenario, consistent with the results observed for the segmented examples.

Comparing the fourth and fifth bars indicates that filtering rolling classifications eliminated many spurious incorrect predictions and some redundant correct predictions. The filter thus successfully created smoother and more reliable results.

The final two bars summarize the performance when using only accelerometer-based or strain-based features. This decreases the percent of trials with a single, correct prediction to $55.0\% \pm 20.2\%$ or $71.7\% \pm 14.5\%$, respectively. This corroborates the conclusion that both modalities provide valuable and complementary information about the gestures.

3) Confusion Results: Fig. 8 further explores performance of the full pipeline by considering each gesture separately. The main section forms a confusion matrix focusing on the ideal case of a single rolling prediction per gesture; each cell reports the percent of trials in which the true gesture of the row yielded a single rising edge of predictions for the column gesture. The remaining possibilities are to have multiple, mixed, or missed predictions for a gesture; these are summarized by the extra four columns on the right, so each row sums to 100%. Results aggregate all 7 classifiers evaluated on their respective hold-out experiments, so each row summarizes 70 true gestures.

Results are promising for robust streaming performance, highlighting that there were no cases of a gesture only being associated with incorrect predictions. It is also interesting to note that certain gestures yielded noticeably more missed or redundant predictions. For example, single brief gestures often have more missed trials, possibly since a small portion of the 2 s window actually contains the gesture; these include *Home*, *Eat*, *Thank You*, *Like*, and *Hello*.

Certain gestures also had more instances of mixed correct and incorrect predictions. To expand on this case, Fig. 9 reports the number of times that each label was predicted during each true gesture. Note that each row no longer sums to the total number of true gestures, since varying numbers of predictions may be made for each one. However, the results

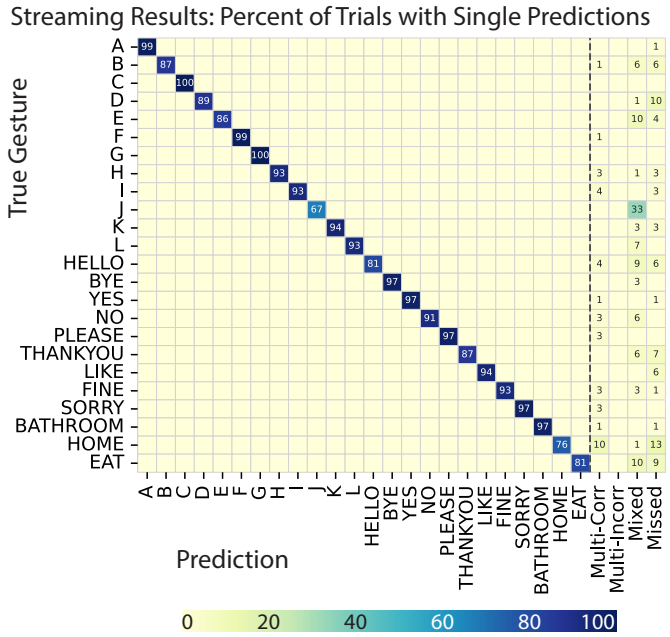

Fig. 8: The main confusion matrix summarizes when gestures yielded a single rising edge of streaming predictions. The extra four columns on the right summarize when multiple or no predictions were made. Values are percents of the 70 gesture trials represented by the row.

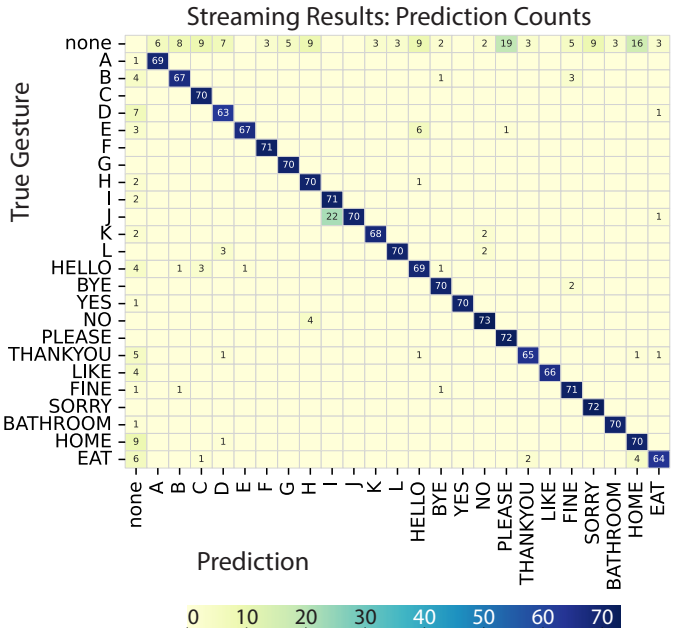

Fig. 9: The matrix presents absolute counts of streaming classifications during each true gesture. Each row spans 70 true gesture trials, but the rows may have a different sum since multiple or no predictions can be outputted during each trial. The top row represents false positives.

can provide insight into which gestures were confused by the network. For example, the matrix shows that *J* often included incorrect *I* predictions, but *I* was never incorrectly classified as *J*. As demonstrated by Fig. 4, these use the same hand pose but *I* is static while *J* involves a brief motion. Thus, the erroneous predictions were likely during the initial static phase of the *J* gesture and then followed by correct predictions, leading to the observed mixed results. This particular case also suggests that the network successfully used accelerometer data to differentiate the two gestures.

The top row indicates 124 total false positives. This is acceptable for the current application, considering that classifications were performed at 5 Hz over a total of 4.47 hours.

C. Online Results

Additional tests demonstrated feasibility of deploying the pipeline on a self-contained embedded wearable system.

1) Embedded Deployment: The smoothing, normalization, and feature extraction procedures were implemented in the real-time embedded context. The network trained while holding out experiment 7 was then converted as described in Section VII-A. Evaluating the network requires 9,622,900 multiply-accumulate operations. Optimization options were tested via CUBE-AI. To measure inference time, it deployed the network on the microcontroller and evaluated 10 random inputs with values uniformly distributed between $[0, 1]$. To measure correctness, it evaluated the network on a laptop using 10,000 random inputs and compared outputs with the original Keras model. Table I summarizes the results. Accuracy considers which class has the highest probability, while RMSE considers the probabilities directly. Note that only marginal resource impacts are observed for this network since only its input and output layers are affected. Based on these results, a network with compression factor 8 was deployed; this reduces memory without significantly impacting speed or accuracy.

TABLE I: Embedded Network Deployment Options

	Weights [KiB]	RAM [KiB]	Inference Time [ms]	Accuracy	RMSE
Original	198.54	18.07	166.82	100.00%	0.000
Optimize allocations	198.54	17.97	166.72	100.00%	0.000
Compression factor 4	192.21	18.07	167.15	99.95%	0.001
Compression factor 8	**190.05**	**18.07**	**166.83**	**99.39%**	**0.010**

2) Performance: Online results were obtained by performing each gesture 10 times and observing the streaming classifications. Prior to the experiment, the user could briefly practice each gesture while watching the output. Future evaluations can be more comprehensive, but the current study was designed to demonstrate that the embedded system can successfully implement the pipeline and validate the leave-one-experiment-out simulated streaming results.

The network made predictions at approximately 5 Hz as expected. Considering raw rolling classifications across all 240 gestures, 89.2% of trials yielded only the correct prediction. 5.8% of trials yielded mixed correct and incorrect predictions. 2.1% and 2.9% of trials had only incorrect predictions or no predictions, respectively. When filtering rolling classifications, 86.7% of trials yielded only the correct classification, and 1.7% of trials yielded mixed correct and incorrect predictions. 1.3% and 10.4% of trials had only incorrect predictions or no predictions, respectively.

This performance is comparable with the offline streaming results. Qualitatively, the network was often sensitive to small pose changes, especially regarding contact between fingers which create large resistance changes. Additionally, the 2 s classification window coupled with the previously observed response times of the knit can cause some prediction delays.

These results demonstrate that the pipeline was successfully deployed in the embedded context. Future work can investigate improving performance by expanding the training corpus, adjusting the network structure, tuning filter windows, or exploring optimal strain sensor placements.

IX. Conclusions and Future Work

This paper presents a wearable smart glove that utilizes a strain-sensitive resistive knit for postural information and an accelerometer for motion. A small custom PCB and a microcontroller read sensors, perform feature extraction, and run a pre-trained neural network. The system is used to classify sign language poses and gestures in real time.

This work demonstrates the potential of combining novel soft sensors with state-of-the-art microcontrollers and machine learning. However, future work can further characterize the capabilities, limitations, and learning pipeline design.

Future studies should expand the subject pool to evaluate robustness and generalizability, such as whether the network can be plug-and-play for new users or whether a tuning procedure should be added (either offline or online). They could also explore factors such as the user's ASL experience level, hand size, or skin conductance. Insights from cross validation may guide structural adjustments to improve robustness. Adjusting the classification windows or filtering may also reduce latency.

Exploring the capacity and trade-offs of the learning pipeline is also valuable. Adding gestures could be done with minimal impact on speed or memory by simply adjusting the softmax layer, but only until the LSTM's learning capacity saturates. Scaling the LSTM layer or adding layers scales how many operations the microcontroller must perform to evaluate the network. Network size and gesture count also impact how much training data is required. Such trade-offs between size, speed, accuracy, and training can be application-specific and nontrivial to characterize for neural networks.

Finally, augmenting the glove with additional modalities could unlock applications ranging from healthcare to sports.

This work thus takes a step towards more deployable machine learning in embedded form factors that are suitable for wearable devices. This moves closer towards the vision of ubiquitous smart wearables that could improve communication and lead to more intuitive human-machine interfaces.

References

[1] T. Sagisaka *et al.*, "High-density conformable tactile sensing glove," in *Int. Conf. Humanoid Robots*, 2011.

[2] V. Nguyen *et al.*, "Handsense: Capacitive coupling-based dynamic, micro finger gesture recognition," in *Conf. Embedded Networked Sensor Systems*, 2019.

[3] O. Glauser *et al.*, "Interactive hand pose estimation using a stretch-sensing soft glove," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 4, 2019.

[4] S. Sundaram *et al.*, "Learning the signatures of the human grasp using a scalable tactile glove," *Nature*, vol. 569, no. 7758, 2019.

[5] X. Zhang *et al.*, "Cooperative sensing and wearable computing for sequential hand gesture recognition," *IEEE Sensors Journal*, vol. 19, no. 14, 2019.

[6] W. Dong *et al.*, "Soft wrist-worn multi-functional sensor array for real-time hand gesture recognition," *IEEE Sensors Journal*, 2021.

[7] "MANUS finger and full-body tracking." (2022), [Online]. Available: https://manus-meta.com/.

[8] C. Valli and C. Lucas, *Linguistics of American sign language: An introduction*. Gallaudet Press, 2000.

[9] F. Pezzuoli *et al.*, "Recognition and classification of dynamic hand gestures by a wearable data-glove," *SN Computer Science*, vol. 2, no. 1, 2021.

[10] B. G. Lee and S. M. Lee, "Smart wearable hand device for sign language interpretation system with sensors fusion," *IEEE Sensors Journal*, vol. 18, no. 3, 2017.

[11] A. K. Singh *et al.*, "A low-cost wearable Indian sign language interpretation system," in *Int. Conf. Robotics and Automation for Humanitarian Applications*, 2016.

[12] P. Kumari *et al.*, "Increasing trend of wearables and multimodal interface for human activity monitoring: A review," *Biosensors and Bioelectronics*, vol. 90, 2017.

[13] M. Caeiro-Rodriguez *et al.*, "A systematic review of commercial smart gloves: Current status and applications," *Sensors*, vol. 21, no. 8, 2021.

[14] C. Demolder *et al.*, "Recent advances in wearable biosensing gloves and sensory feedback biosystems for enhancing rehabilitation, prostheses, healthcare, and virtual reality," *Biosensors and Bioelectronics*, 2021.

[15] J. Pan *et al.*, "A wireless multi-channel capacitive sensor system for efficient glove-based gesture recognition with AI at the edge," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67.9, 2020.

[16] J. Hughes *et al.*, "A simple, inexpensive, wearable glove with hybrid resistive-pressure sensors for computational sensing, proprioception, and task identification," *Advanced Intelligent Systems*, vol. 2.6, 2020.

[17] Y. J. Son *et al.*, "Humidity-resistive, elastic, transparent ion gel and its use in a wearable, strain-sensing device," *Materials Chemistry A*, vol. 8, no. 12, 2020.

[18] J. Pan *et al.*, "Hybrid-flexible bimodal sensing wearable glove system for complex hand gesture recognition," *ACS sensors*, vol. 6, no. 11, 2021.

[19] O. Atalay *et al.*, "Comparative study of weft-knitted strain sensors," *J. Industrial Textiles*, vol. 46.5, 2017.

[20] O. Atalay and W. Kennon, "Knitted strain sensors: Impact of design parameters on sensing properties," *Sensors*, vol. 14, no. 3, 2014.

[21] Y. Song *et al.*, "Design framework for a seamless smart glove using a digital knitting system," *Fashion and Textiles*, vol. 8, no. 1, 2021.

[22] Y. Luo *et al.*, "KnitUI: Fabricating interactive and sensing textiles with machine knitting," in *Conference on Human Factors in Computing Systems (CHI)*, 2021.

[23] X. Fu *et al.*, "Knitted Ti3C2Tx MXene based fiber strain sensor for human-computer interaction," *Journal of Colloid and Interface Science*, vol. 604, 2021.

[24] S. Seyedin *et al.*, "Knitted strain sensor textiles of highly conductive all-polymeric fibers," *ACS applied materials & interfaces*, vol. 7, no. 38, 2015.

[25] *Agloves*, 2022. [Online]. Available: http://agloves.com/.