Check for updates

ASME Journal of Biomechanical Engineering
Online journal at:
https://asmedigitalcollection.asme.org/biomechanical

Inferring Fine Finger Motions for Prosthetic Control: An Ultrasound-Based Approach to Real-Time Estimation of Finger Kinematics

Dean Zadok[1]
Department of Computer Science,
Technion,
Haifa 3200003, Israel
e-mail: deanzadok@cs.technion.ac.il

Oren Salzman
Department of Computer Science,
Technion,
Haifa 3200003, Israel
e-mail: osalzman@cs.technion.ac.il

Alon Wolf
Department of Mechanical Engineering,
Technion,
Haifa 3200003, Israel
e-mail: alonw@me.technion.ac.il

Alex M. Bronstein
Department of Computer Science,
Technion,
Haifa 3200003, Israel
e-mail: bron@cs.technion.ac.il

A central challenge in building robotic prostheses is the creation of a sensor-based system able to read physiological signals from the lower limb and instruct a robotic hand to perform various tasks. Existing systems typically perform discrete gestures such as pointing or grasping, by employing electromyography (EMG) or ultrasound (U.S.) technologies to analyze muscle states. While detecting finger activation has been done in the past, we are interested in detection, or inference, done in the context of fine motions that evolve over time. Examples include motions occurring when performing fine and dexterous tasks such as typing on a keyboard or playing a musical instrument. We consider this task as an important step toward higher adoption rates of robotic prostheses among arm amputees, as it has the potential to dramatically increase functionality in performing daily tasks. To this end, we present an end-to-end robotic system, which can successfully infer fine finger motions in real-time. This is achieved by modeling the hand as a robotic manipulator and using it as an intermediate representation to encode muscles' dynamics from U.S. images. We evaluated our method by collecting data from a group of subjects and demonstrating how it can be used to replay music played on the piano or text typed on a computer keyboard. To the best of our knowledge, this is the first study demonstrating these downstream tasks within an end-to-end system. [DOI: 10.1115/1.4068887]

Keywords: artificial organs and prostheses, biomechanics, physiological systems

1 Introduction

Bionic hands or robotic prostheses was sought after for decades, yet the first commercial robotic prostheses only became available in 2007 [1] while a fully functioning prosthesis enabling all range of daily tasks is still out of reach. Existing robotic prostheses are typically connected to the limb of an amputee (or to a subject with severe phalanx or palm deformations) when the subject's muscles still respond to electric potentials generated by the brain [2]. These electric potentials allow inferring different finger motions by placing a noninvasive sensor on the residual limb, having subjects attempt these motions or gestures, and then discriminating between the different types of recorded electrical signals [3,4], or learn to control a digital device online [5,6]. This sensing technology, called electromyography (EMG), allows the reproduction of grasp flexion and extension yet is unable to reproduce finer, or more gentle, motions typically involved in daily tasks, such as pressing on a keyboard.

In recent years, it was shown that replacing EMG with ultrasound (U.S.) imaging that captures the muscles' morphological state allows for better differentiation between discrete gestures or to classify full-finger flexion [7]. These images are obtained by placing a U.S. probe on the residual limb and its efficacy is based on the fact that the muscles generate different deformation patterns for different actions, and this behavior can be generalized among different subjects [7–10]. Arguably, the most popular approach to infer motions from (high-dimensional) U.S. images is via data-driven methods [8,10,11]. However, to the best of our knowledge, these methods still fall short of fully detecting dexterous and fine finger motions that the subject intended to perform. Predicting such motions, which is the focus of our work, is instrumental to the fine control of robotic hands for fine-motion tasks and, consequently, to increase functionality and adoption rates of robotic prostheses.

Specifically, the key question this research aims to answer is "*To what extent can we predict and differentiate between fine finger motions by only having access to the lower-arm muscles?*" In our context, we use the term "fine" to refer to finger motions that are gentle and require slight activation of flexion muscles [12,13]. As challenging exemplary use cases, we consider piano playing and keyboard typing. In these tasks, finger gestures required to press or type are usually finer than those required to grasp objects or fully flex

International Conference on Robotics and Automation, 2023.
[1]Corresponding author.
Manuscript received October 10, 2024; final manuscript received May 31, 2025; published online July 14, 2025. Assoc. Editor: Elizabeth Rapp van Roden.

Copyright © 2025 by ASME; reuse license CC-BY 4.0

Downloaded from asmedigitalcollection.asme.org/biomechanical/article-pdf/147/9/091005/7512464/bio-24-1272.pdf by guest on 22 September 2026

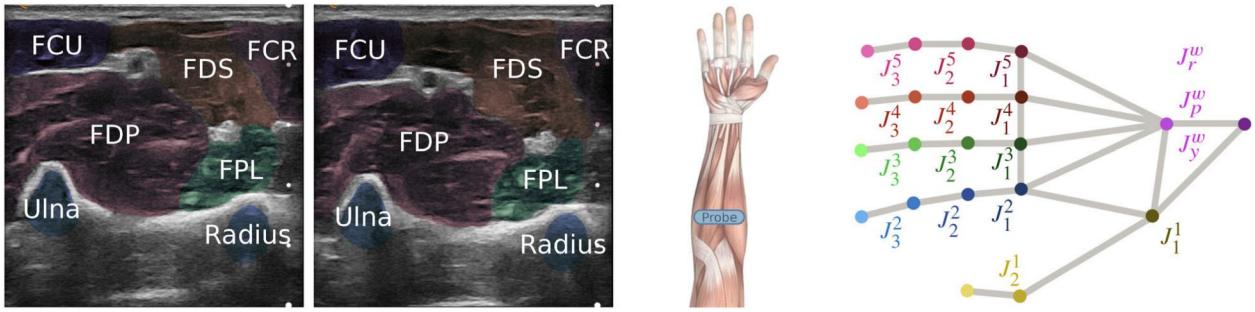


Fig. 1 (Left) Visual segmentation of two U.S. images (top of each image is the skin shell). The Ulna and radius (the two main bones) are marked in blue, and the flexion muscles are marked with different colors. All fingers are at rest in the left image while in the right image the ring finger is fully flexed. Notice that the region capturing the FDS in the right image is larger than on the left image as the FDS becomes more dominant in this area during movement. (Right) illustration of probe mounting position during data collection, based on the "transverse" location proposed by McIntosh et al. [11], and a visual illustration of a hand's configuration. The thumb (F^1) is associated with two joints $[J_1^1, J_2^1]$, the other four fingers F^i ($i \in \{2, \ldots, 5\}$) are each associated with three joints $[J_1^i, J_2^i, J_3^i]$ and three additional DOFs are associated with the wrist $[J_r^w, J_p^w, J_y^w]$. The set of joints $\{J_1^i | 1 \le i \le 5\}$, $\{J_2^i | 2 \le i \le 5\}$, and $\{J_3^i | 2 \le i \le 5\}$ correspond to the MP, PIP, and DIP joints, respectively (see Sec. 3). Figures are best viewed in color.

the fingers, generating smaller and less observable lower-arm muscle movements. To the best of our knowledge, predicting finger motions for these tasks has not been evaluated as part of an end-to-end system and we hope that our method and dataset will be used as a baseline for understanding these motions.[2]

To this end, we present an end-to-end robotic system that allows to successfully solve our research question. This success stems from the design choices we took to construct our system which we view as a major contribution of this work. These include: (i) choosing a B-mode[3] U.S. sensor to capture precise observation of the flexion muscles, (ii) using a sequence of multiple U.S. images at each step, and (iii) a learning framework designed to exploit the temporal nature of the input U.S. images. The proposed neural network (NN) architecture allows one to utilize the fact that finger pressing can be inferred by both the muscles' dynamics and an intermediate lower-dimensional representation of the palm of the hand inspired by robotic manipulators.

Our framework can be used to accurately reproduce music played or text typed, applications that were previously unattainable. This was done by collecting data from a group of subjects and successfully predicting finger motions in our two motivating applications. Finally, we demonstrate our system by connecting it to a robotic arm that replayed piano notes in real-time.

2 Related Work

Object grasping is one of many actions human hands perform daily. One reason for the low adoption rates of prosthetic arms among arm amputees is the lack of functionality in performing daily tasks [14]. To enrich the variance of gestures, advanced wearable surface-EMG (sEMG) sensors have been used (e.g., Myo armband [4,15]). The array of features provided by sEMG enabled inferring more than grasp intention, and classifying a varied set of hand gestures via machine learning and deep learning (DL) [15,16]. Additionally, recent findings suggest that DL techniques can effectively control robotic prostheses using proportional control to predict continuous degrees-of-freedom (DOFs) [17–20]. Furthermore, sEMG demonstrated promising performance for arm amputees [4,21]. Noteworthy, in this domain, subjects differ in their gestures, which puts forth the requirement of operating across a variety of muscular structures [22].

In recent years, U.S. sensing technologies emerged as a promising alternative to EMG. McIntosh et al. proposed to extract the optical flow of a U.S. stream prior to the classification [11]. Others used machine learning algorithms demonstrating gesture classification from downsampled U.S. images of the same area [7,23]. Gesture recognition was achieved with wearable single-transducer sensors [8,24,25]. In addition, researchers demonstrated that similar techniques allow for more accurate decoupling of different DOFs and use them to classify gestures [10,26]. Unsupervised techniques were also studied, showing discrimination of unlabeled gestures [9,27], which might imply the possibility to understand the muscle behavior of arm amputees without explicit labeling. Nevertheless, the feasibility of instructing a robotic hand to execute fine finger motions remains an open challenge.

For the analysis of U.S. imagery DL techniques have become increasingly dominant in tasks ranging from tissue segmentation and pathology detection [28], and image enhancement [29], replacing traditional beamforming techniques with learned beamformers for faster and better image acquisition [30,31]. In the context of learning dynamics of human organs, U.S. imaging with transducers placed under the jaw demonstrated reliable speech and pronunciation recreation from imaged tongue movements [32,33]. Vogt et al. utilized the tongue's morphological behavior to extrapolate sung pitch and phonemes [34]. Lulich et al. studied the correlation between tongue motion and the frequencies generated by clarinet playing [35]. These works suggest that U.S. provides a rich signal source describing minute muscular motion.

3 Anatomical Background

In this section, we provide the anatomical background required to understand the approach we take to predict fine finger motion. Specifically, we explain (in general terms) the relation between lower-arm muscles and the state of the wrist's and fingers' joints. The lower arm contains two main bones, the *Ulna* and the *Radius*. Surrounding these bones are different muscles that are responsible for the flexion, extension, and rotation of different joints in our palms (including all the fingers). Two annotated U.S. images of this region are presented in Fig. 1 (left), highlighting the muscles relevant to our problem as well as different muscle states corresponding to different finger motions. All images were captured by placing the probe in the location illustrated in Fig. 1 (second right).

We start by concentrating on the four fingers (Index through Little finger). There are three types of joints associated with these four fingers: the metacarpophalangeal (MP) joints, the proximal interphalangeal (PIP) joints, and the distal interphalangeal (DIP) joints (Fig. 1 rightmost). Close to the skin shell, we can find the *flexor*

[2]Note that we do not showcase our results on deformed muscles. We performed tests on healthy subjects with fully functioning hands in order to provide an accurate benchmark for the task we concentrate on. Our code and dataset are available on https://github.com/deanzadok/finemotions.

[3]Also known as Brightness mode ultrasound, the most common type of ultrasound imaging used in medical diagnostics. It produces a two-dimensional cross-sectional image of tissues and structures within the body.

Downloaded from asmedigitalcollection.asme.org/biomechanical/article-pdf/147/9/091005/7512464/bio-24-1272.pdf by guest on 22 September 2026

carpi radialis and the *flexor carpi ulnaris* that are in charge of flexion of the wrist, and the *palmaris longus* that contributes to the wrist and four fingers flexion, using tendons connecting to the MP joints. Between the flexor carpi ulnaris and the flexor carpi radialis muscles, we can identify the *flexor digitorum superficialis* (FDS) that flexes the four fingers, and, in full flexion, also contributes to the flexion of the wrist. The palmaris longus is found under the FDS and is not clearly visible given the placement of our U.S. probe but both work in concert actuating some of the joints. The FDS connects to tendons that go through the PIP and MP joints of the four fingers. Deeper into the arm, the *flexor digitorum profundus* (FDP), similar to the FDS, is in charge of finger flexion and wrist-control assistance using tendons connected to the DIP joints. Lastly, close to the FDP, is the *flexor pollicis longus*, which controls the Thumb joints. To summarize, observing the state of the FDS and FDP muscles allows (in theory) to estimate the whereabouts of all four fingers, while the flexor pollicis longus provides the knowledge required to understand Thumb movement.

4 System Design

Following the anatomical background presented, we argue that one can infer key-pressing actions by considering only lower-arm muscles. To design such a system, we require choosing (i) the type of sensor, (ii) the input it provides to the inference algorithm, and (iii) the inference algorithm itself. For our system, we chose to use a B-mode U.S. sensor capable of recording high-quality images at a high frequency, to capture muscle properties and dynamics. See Sec. 6.1 for technical details. Choosing U.S. instead of EMG relies on previous work [16] in which U.S. completely outperformed EMG in classifying gestures. Moreover, when compared to the single-dimensional signal provided by an EMG sensor, spatial features that are found in a U.S. image have more information about fingers' whereabouts and are less noisy. This will be key in the way we address the latter two design choices.

Recall that the U.S. sensor provides us with an image. However, in contrast to existing approaches [7,26,36], we chose to use a *sequence* of the latest-acquired U.S. images as the input to our inference algorithm. As we will demonstrate empirically (Sec. 6), considering a sequence of U.S. images allows to exploit the dynamics which in turn, leads to significantly better results. Given a stream of U.S. images, we chose to solve the inference problem via a NN architecture that is tailored for processing images, exploits temporal coherence, and allows a domain expert to inject domain knowledge via an intermediate representation. In our setting, this intermediate representation is an encoding of the hand, modeled as a robotic manipulator. We detail the exact architecture and specific design considerations in Sec. 6.

5 Inference Learning Framework

In this section, we describe our inference learning framework. We start (Sec. 5.1) by formally defining and modeling the inference problem and then continue (Sec. 5.2) to describe an approach where we only make use of U.S. images without the aforementioned intermediate representation. As we will see (Sec. 6), utilizing an intermediate representation during training can lead to better inference capabilities. Thus, in Sec. 5.3 we extend our first method to account for said representation. Notice that in this section, we concentrate on the high-level learning framework and the exact training and inference procedures are detailed in Sec. 6.2.

5.1 Notation and Model. In this section, we introduce the necessary notation and formally define our inference problem. We assume to be given a sequence of k U.S. images corresponding to k timestamps and wish to use them to predict which finger was pressed for each timestamp k.

Let $F^1, \ldots, F^5$ denote the five fingers with F^1 and F^5 corresponding to the Thumb and Little finger, respectively. For each finger F^i and each timestamp t_j, we associate a binary variable $p_j^i \in \{0,1\}$ such that $p_j^i = 1$ if finger F^i is pressed at timestamp t_j and $p_j^i = 0$ otherwise. For every timestamp t_j, we denote by $p_j \in \{0,1\}^5$ the binary vector indicating which fingers are pressed at t_j which we term a *pressing vector*. Namely, $p_j := \langle p_j^1, \ldots, p_j^5 \rangle$. We assume that for every timestamp t_j there is a corresponding U.S. image s_j as well as a *configuration* x_j. Here, x_j is an encoding of the fingers and wrists as a set of joint angles (see Fig. 1 rightmost).

Our training data consists of tuples $\langle \bar{s}_\ell, \bar{x}_\ell, \bar{p}_\ell \rangle$. Here, $\bar{s}_\ell := \langle s_\ell, \ldots, s_{k+\ell} \rangle$ is a sequence of k consecutive U.S. images, $\bar{x}_\ell := \langle x_\ell, \ldots, x_{k+\ell} \rangle$ is a sequence of k consecutive configurations and $\bar{p}_\ell := \langle p_\ell, \ldots, p_{k+\ell} \rangle$ is a sequence of k consecutive pressing vectors all of which start at timestamp t_ℓ. We denote by $\mathcal{S}$ and $\mathcal{X}$ the sets of all sequences of k U.S. images and configurations, respectively. For each finger F^i and each timestamp t_j we wish to predict p_j^i (i.e., if F^i was pressed at t_j). To this end, our system will output the variable $\hat{p}_j^i \in [0,1]$ estimating the probability that $p_j^i = 1$. Denote $\hat{p}_j := \langle \hat{p}_j^1, \ldots, \hat{p}_j^5 \rangle$ and $\widehat{\bar{p}}_\ell := \langle \hat{p}_\ell, \ldots, \hat{p}_{k+\ell} \rangle$. Notice that the complexity of our problem is higher than traditional classification problems, as we allow to have different fingers pressing simultaneously. As we will see, our approach will make use of an intermediate representation corresponding to the configuration at each timestamp. Thus, we will also denote $\hat{x}_j$ as the system's estimation of x_j and $\widehat{\bar{x}}_\ell := \langle \hat{x}_\ell, \ldots, \hat{x}_{k+\ell} \rangle$. In Sec. 6, we describe the experimental setting which allows us to obtain labeled training data for our two representative tasks, piano playing and keyboard typing. In the meantime, we describe our approach to solving our inference problem.

5.2 Learning Pressing Events Directly. To estimate pressing vectors, we use a hybrid architecture starting with a convolutional neural network (CNN) module [37] to extract spatial features from the U.S. images, followed by a recurrent neural network (RNN) [38,39] to aggregate these features and infer pressing vectors. Similar combinations were shown to be effective in various computer-vision tasks that require temporal coherence [40–42]. By combining these modules, we create a sequence-to-sequence model, which receives a sequence of U.S. images $\bar{s}_\ell$ as input and outputs $\widehat{\bar{p}}_\ell$ which estimate the pressing vector $\bar{p}_\ell$. We call this the multiframe (MF) model and denote it as G_θ where θ is the set of model parameters that we optimize in training, such that the trained model is used by computing $G_\theta(\bar{s}_\ell) = \widehat{\bar{p}}_\ell$ to output predicted pressing vectors $\widehat{\bar{p}}_\ell$.

5.3 Learning Pressing Events Via Hand Configurations. To estimate pressing vectors while making use of an intermediate representation of the finger's configuration, we use an auto-encoder [38,43] framework composed of two main blocks—an encoder mapping the sequence of U.S. images $\bar{s}_\ell$ to a sequence of configurations $\widehat{\bar{x}}_\ell$ and a decoder using $\widehat{\bar{x}}_\ell$ to estimate the pressing vector by outputting $\widehat{\bar{p}}_\ell$. The encoder's architecture follows the architecture proposed for the MF model (i.e., a CNN followed by an RNN) while the decoder uses an additional RNN module together with a residual connection [44] that feeds the model with the hidden features from the encoder (see Fig. 2 for visualization).

Auto-encoders allow a domain expert to inject domain knowledge via an intermediate representation (hand configurations in our setting) [45,46]. The residual connection was added to better allow the decoder to infer pressing vectors by obtaining knowledge from both U.S. images and configurations. We name this model the configuration-based multiframe (CBMF) model and denote it as $H_{\theta,\phi} = h_\phi^d \circ h_\theta^e$ where θ and ϕ are the encoder's and decoder's parameters, respectively, which we optimize in the training phase.[4]

While the exact details of how we train this model are explained in the experimental section (Sec. 6.2), we note that the training process

[4]Here the superscript "e" and "d" are used to refer to the encoder and decoder modules, respectively.

Downloaded from asmedigitalcollection.asme.org/biomechanical/article-pdf/147/9/091005/7512464/bio-24-1272.pdf by guest on 22 September 2026

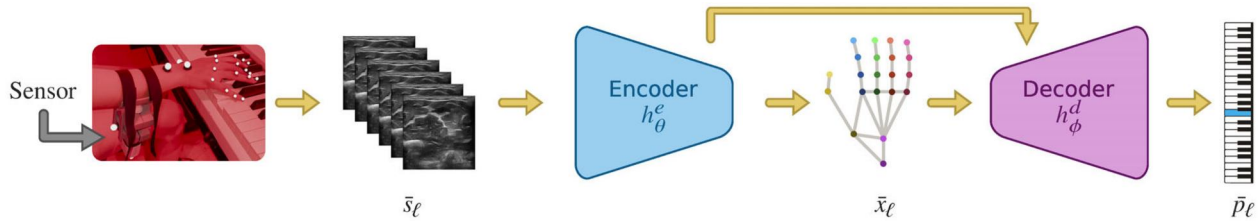


Fig. 2 A schematic flow of the proposed CBMF model (Sec. 5.3). An ultrasound sensor is placed on the lower arm while the subject is playing the piano. A continuous stream of ultrasound images $\bar{s}_{\ell\psi}$ from the sensor is fed into the neural-network encoder h^e_θ which creates a latent representation of hand skeleton configurations $\bar{x}_\ell$. The decoder h^d_ϕ receives the latent representation and outputs the vector of probabilities $\bar{p}_{\ell\psi}$ indicating the pressed keys. The entire system is trained to produce a common representation for both the skeleton tracking and key prediction tasks.

is done in two phases. In the first phase, we train only the encoder h^e_θ while in the second phase, we simultaneously train the (pretrained) encoder h^e_θ together with the (untrained) decoder h^d_θ. Splitting the learning into two phases (i.e., a single and a multimodal supervised learning task) was shown to produce better results in similar tasks [47–49]. Roughly speaking, by pretraining h^e_θ we get a "coarse" mapping between U.S. images $\bar{s}_\ell$ and predicted hand configurations $\widehat{\bar{x}}_\ell$. This allows us to "bootstrap" the network's parameters when solving the entire inference problem in our multimodal task. During real-time control, the trained model uses only a sequence of U.S. images $\bar{s}_\ell$ to compute $H_{\theta,\phi}(\bar{s}_\ell) = \langle \widehat{\bar{x}}_\ell, \widehat{\bar{p}}_\ell \rangle$ and change finger positions according to the predicted pressing vectors $\widehat{\bar{p}}_\ell$.

6 Experiments

In this section, we empirically evaluate our system and the parameters that govern its performance. This culminates in demonstrating the efficacy of our framework in performing tasks that require fine finger motions. We start (Sec. 6.1) by describing our experimental setup and data-collection methodology. We then describe our model and training details (Sec. 6.2) and finish with an analysis of the data collected, by answering (i) How accurately can we predict finger-pressing tasks (Sec. 6.3), (ii) How accurate are our intermediate robotic representations (Sec. 6.4), (iii) How good is our method in applying finger pressure (Sec. 6.5), and (iv) Demonstrating real-time applicability of our robotic system (Sec. 6.6).

6.1 Experimental Setup and Data Collection. Our setup includes a Clarius L15HD (wireless B-mode U.S. device), a Roland FP-30 (digital piano), a standard keyboard, and a Vicon motion-capture system. The U.S.'s configuration was set for the musculoskeletal task, with 10 MHz frequency generating a 5 cm × 5 cm processed image at a rate of 19–21 images per second, at a resolution of 480 × 480. It was attached using a wearable strap (see Fig. 2) to the so-called "transverse" location (see, e.g., Ref. [11]). The piano and keyboard were used to record the set of pressed notes and keys during each U.S. frame and the Vicon system was used to track the palm's joint locations. Here, each joint location is captured and used to extract the joint angles that form the hand's configuration, as described in Appendix A.

We collected data from 12 subjects; averaging at 23 years old, seven of whom are men and five of whom are women. All subjects were confirmed to be right-handed, experienced in playing piano, and without any neurological disorders or deformity in the relevant hand area. At the beginning of both tasks, subjects were instructed to put their hands in a stationary state in which each of their fingers must be on the same note/key for the entire session. For piano playing, subjects were asked to play melodies that only require five distinguished notes and can be played using one hand only. For keyboard typing, subjects were asked to randomly type on each of the five keys their fingers are stationed over, while not using more than one key at the same time. For each task, subjects were asked to perform 2–3 sessions, with approximately 90 s for each session, and rest between sessions. The process was approved by the institution's Ethics Committee. The final dataset includes 44,596 samples for piano playing and 42,143 samples for keyboard typing, each sample representing the tuple $\langle \bar{s}_\ell, \bar{x}_\ell, \bar{p}_\ell \rangle$ (see Sec. 5.1).

6.2 Model Training. Before training, both images s_i and configurations x_i were normalized to the range of [0, 1]. On feedforward, our MF model G_θ receives a sequence of $k = 8$ U.S. images $\bar{s}_\ell$, such that each image s_i is downsampled to a resolution of 224 × 224 and fed to the CNN block to extract feature maps, which are then flattened, concatenated and fed into the RNN module to output k pressing vectors. Our CBMF model $H_{\theta,\phi}$ works such that the encoder h^e_θ is fed with images similar to the MF model, and outputs k configurations, that are then fed to the decoder h^d_ϕ (an additional RNN block), along with the features from the residual connection. The decoder h^d_ϕ is designed to output k predicted configurations. The entire architecture is detailed in Appendix E. During backpropagation, for the predicted configurations, the applied loss function for $N = 32$ samples in a batch is the mean squared error on all configurations, i.e., $\mathcal{L}_{\text{MSE}} = 1/(N \cdot k) \sum_{i \in N \cdot k} ||\hat{x}_i - x_i||_2^2$. A configuration x_i is a vector of 17 angles, as depicted in Fig. 1 (rightmost). For the predicted pressing vectors, the applied loss objective for $N = 32$ samples in a batch, such that each sample is k vectors of $M = 5$ probabilities, is the binary cross entropy for all probabilities, i.e., $\mathcal{L}_{\text{BCE}} = 1/(N \cdot k \cdot M) \sum_{i \in M} \sum_{j \in N \cdot k} p^i_j \log \hat{p}^i_j + (1 - p^i_j) \log(1 - \hat{p}^i_j)$. To train the MF model, we apply the loss function $\mathcal{L}_{\text{BCE}}$, and to train the CBMF model, we apply the combined loss function $\lambda \mathcal{L}_{\text{MSE}} + \mathcal{L}_{\text{BCE}}$, where λ is chosen using a trial-and-error procedure (see Appendix C). All training sessions were executed for 20 epochs, to avoid overfitting. Adam [50] was used as the optimization method, with a learning rate of 0.001.

6.3 Downstream Tasks Evaluation. In this section, we consider the performance on our downstream finger-pressing tasks. We start by testing how accurately can we predict finger-pressing tasks and show the efficacy of our CBMF model compared to alternative methods. We then continue to concentrate on our CBMF model and evaluate its robustness by evaluating how robustly our method predicts finger pressing for different subjects and how robust our method is for different fingers. The experiments were evaluated using five-fold validation: for each task, all sessions of all subjects were divided into five folds, and trained such that the ith fold was set aside for testing and not used in training. This ensures that the test set contains unseen intervals of motion, in contrast to what has been considered in previous research [51,52].

How accurately can we predict finger-pressing tasks? We evaluated the inference capabilities of the models presented in Sec. 5 (MF and CBMF) on the piano-playing and keyboard-typing tasks by plotting the accuracy, recall, precision, and $F1$ (Fig. 3) as well as the test loss during training (Figs. 4(*a*) and 4(*b*)). For baselines, we added the single-frame (SF) model, where we simply take our MF model and replace the RNN with a multilayer perceptron (MLP) to predict a single pressing vector from a single image. To compare our methods with the wide literature, we have

Downloaded from asmedigitalcollection.asme.org/biomechanical/article-pdf/147/9/091005/7512464/bio-24-1272.pdf by guest on 22 September 2026

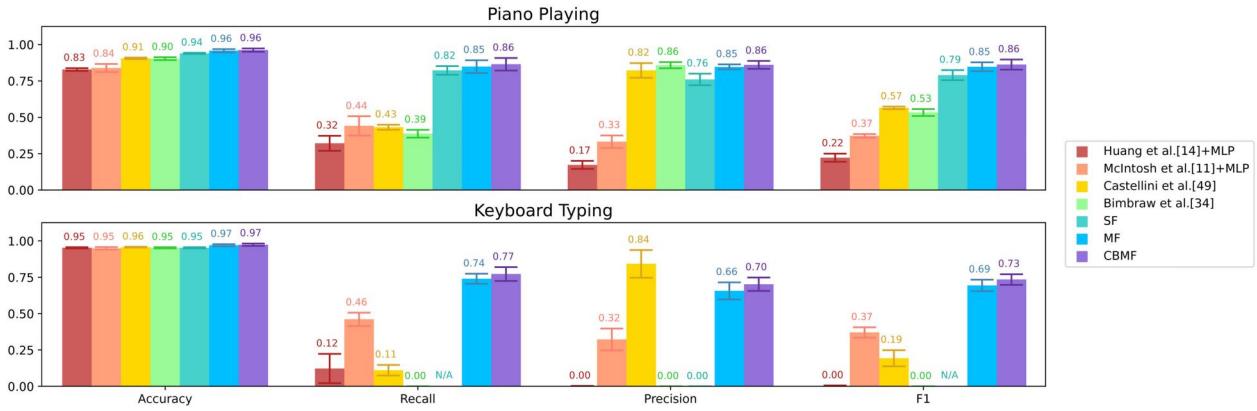


Fig. 3 Evaluation of the MF and CBMF methods (Secs. 5.2 and 5.3) as well as the five baselines (Sec. 6.3) for piano-playing and keyboard-typing tasks. The first four baselines are different state-of-the-art algorithms. Implementation details for baseline algorithms are available in Appendix B. Accuracy, recall, precision, and *F*1 were averaged over all test samples in five folds. The height of each bar represents its mean value, while the horizontal segment represents the standard deviation range.

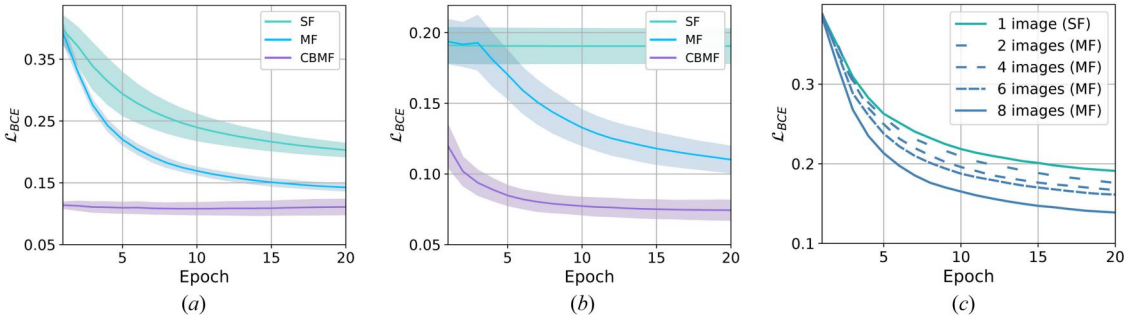


Fig. 4 Comparison of the optimization process using $\mathcal{L}_{BCE}$ as a function of the epoch number for the piano-playing (*a*) and keyboard-typing (*b*) tasks. (*c*) Training loss as a function of epoch number for different sizes of U.S. sequences as inputs.

implemented and evaluated different state-of-the-art algorithms that were designed for different ultrasound-based systems in recent years. Previous research showed that ultrasound features are linearly correlated with finger positions when performing similar tasks [51,52]. Such approach serves as a baseline to evaluate the potential of nonlinearity in neural networks. In addition, we implemented leading gesture classification algorithms [11,16] by concatenating an MLP module to their output, to allow detection of pressing vectors. Lastly, we implemented previously developed neural network architecture designed to estimate finger joints based on a single U.S. image [36], to compare it with our design choices in our SF model. Implementation details are in Appendix B. We also tested common computer-vision models, such as the VGG16 [26,53] model (pretrained on ImageNet [54], or not), and C3D [55], which

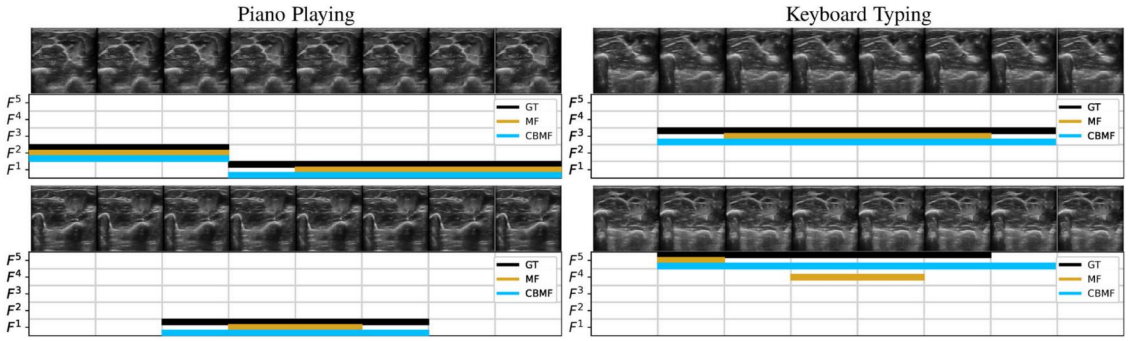


Fig. 5 Samples from executing the MF and CBMF models for both piano-playing and keyboard-typing tasks. In each rollout, the upper row represents finger F^5 (little finger) and the lower row represents finger F^1 (thumb). Ground-truth labels are marked in black. The differences between both models are expressed in consistency and correctly deciding when to press or leave a key. Notice from the U.S. images how hard is it for a human eye to understand subtle morphological changes during such gentle movements.

Downloaded from asmedigitalcollection.asme.org/biomechanical/article-pdf/147/9/091005/7512464/bio-24-1272.pdf by guest on 22 September 2026

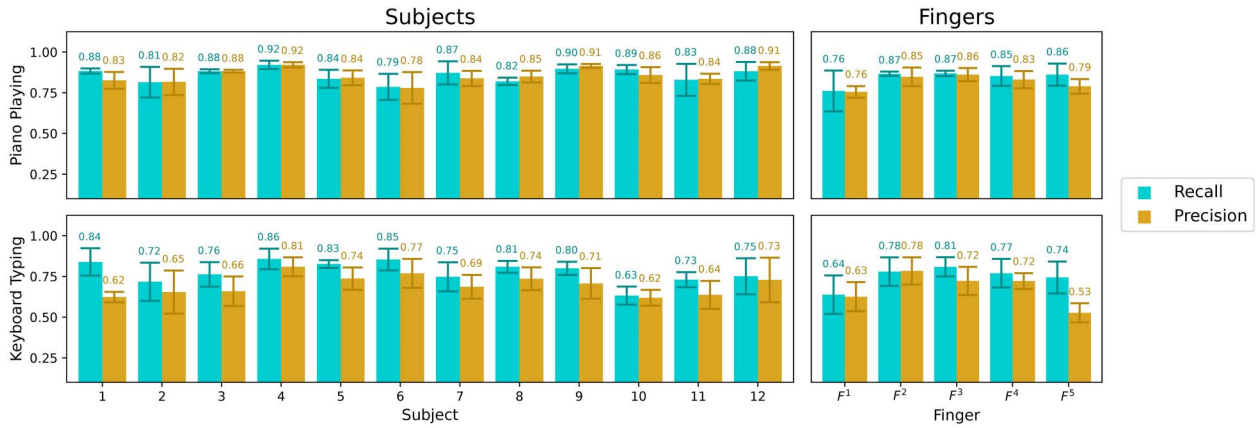


Fig. 6 (Left) Recall and precision for the CBMF model for all 12 subjects. Vertical segments denote one standard deviation. (Right) Recall and precision separated for all five fingers, showing the expected performance of each finger.

were shown to perform favorably in video classification. However, the models did not converge in any of the tasks and therefore were excluded from evaluation.

The overall performance for piano playing is significantly higher than for keyboard typing. We associate this difference with differences in pressing duration and body postures: The maximum note offset on our piano and our keyboard is 10.45 mm and 2.95 mm, respectively. This implies longer muscle flexing when playing the piano, which leads to more dominant motions in U.S. images. Additionally, playing the piano involves different body postures which often lead to more dominant motions in U.S. images.

Moving on to compare the different models, the SF model and [36] cannot recreate the finer task of keyboard typing, but are able to learn to play the piano. This backs up our conjecture that a sequence of multiple frames is required for parameterized models to infer fine finger motions. Evidently, the MF model outperforms the five baselines, while the CBMF model outperforms all (see Fig. 5 for visualization). This is especially noticeable in the harder keyboard-typing task (see recall and precision), which backs up our conjecture that our intermediate representation allows us to better infer fine finger motions, by reducing false positive and false negative predictions. Interestingly, the method proposed in Ref. [52], achieved high precision despite low recall on both tasks, resulting in safe, but a very weak detection of pressing events. It is important to mention that, unlike deep learning models that do not require optimization during inference to extract features from the input signal, this method requires linear regression to optimize features for each U.S. image, making it infeasible for real-time deployment.

In addition, we examined piano playing with different sizes of sequences ($k \in \{1, 2, 4, 6, 8\}$), showing that convergence on the test set improves gradually the more we increase k. Larger sequences may have been beneficial, however, we did not use more than eight consecutive images due to GPU memory limitations.

How robustly can our method predict finger pressing for different subjects? We plot the different metrics for each subject individually to see how the CBMF model varies across different subjects (Fig. 6). Overall, the results for piano playing are better than for keyboard typing, and those who performed better on piano playing, performed better on keyboard typing, but most importantly, the method works for all subjects, showing us the potential for large-scale adoption of these algorithms.

How robust is our method across different fingers? We computed the recall and precision values for each of the fingers individually to understand how each of the fingers affects the quality of the solution (Fig. 6). On both tasks, the model performs better on fingers F^2, F^3, and F^4, rather than on fingers F^1 and F^5. A possible explanation is that operating the three middle fingers causes larger movement in the muscles that are found in the U.S. images when compared to the Thumb and Little finger. As illustrated in Fig. 1 (left), the muscle responsible for Thumb movement is significantly less dominant than the rest of the muscles, which might explain why Thumb movement is the least understandable gesture.

6.4 Evaluation of Model's Intermediate Representation of a Robotic Hand. Recall that our CBMF model uses an intermediate representation corresponding to the configuration at each timestamp (Sec. 6.3). Here, we would like to evaluate the accuracy of the predicted representations. Specifically, we explore whether the CBMF model can predict not only the key pressing event but also the exact hand configuration that led to the predicted event. To this end, we measured the error in predicting hand configurations for each of the 12 subjects (Fig. 7), as well as the absolute error (in radians) for each of the individual 17 joints (Fig. 8).

Additionally, we plotted the error of the predicted fingertip locations by calculating their Euclidean distance from the labeled fingertips positions. This was done by taking the lengths of the fingers' links for each subject and calculating the relative positions of the fingertips for both predictions and ground truth (Fig. 8). The results demonstrate that the more the task requires longer finger movements, the higher the average error, especially for the more dominant joints, such as fingers F^2, F^3, and F^4. Interestingly, the most dominant DOFs near the wrist ($J_1^1, J_r^w, J_p^w, J_y^w$) achieve a smaller

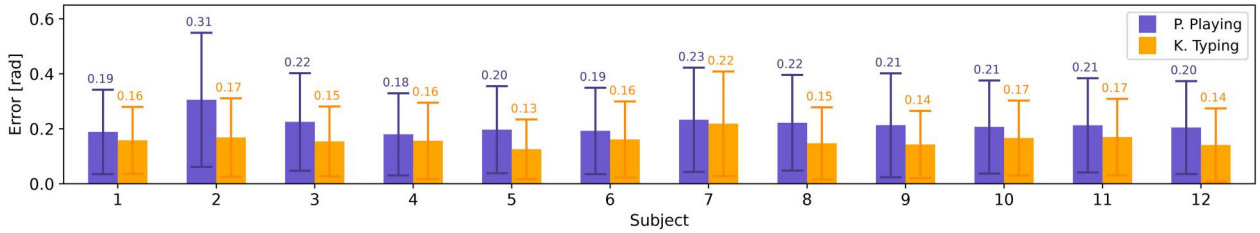


Fig. 7 Prediction of hand configurations for each of the 12 subjects using our CBMF model. Results are the average error in radians of the 17 joint angles. Vertical segments denote one standard deviation.

Downloaded from asmedigitalcollection.asme.org/biomechanical/article-pdf/147/9/091005/7512464/bio-24-1272.pdf by guest on 22 September 2026

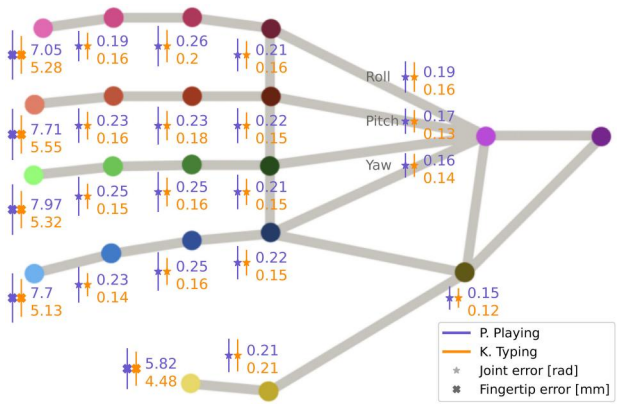


Fig. 8 Visualization of the averaged error for the 17 joint angles, along with the average errors for the fingertips in millimeters, using our CBMF model. Vertical segments denote one standard deviation.

error on average, which could be related to the large region captured by the corresponding muscles in the U.S. images.

To visually understand the accuracy of the intermediate representation, we consider two rollouts (Fig. 9) where different fingers are pressed ($F3$ and $F4$, respectively). For each one, we depict the joint angles (in radians) of the pressed finger as well as the hand configuration (both the ground truth as well as the estimated configuration obtained by the model) for two different timestamps. Note that while there is a (arguably) small error in the joint values, the estimated hand configuration is a fairly accurate representation of the ground truth. We expect that the fact that we are able to accurately estimate hand configurations will be useful in other tasks that require an explicit hand model (e.g., sign language) but this is left for future work.

It is important to mention that there is a tradeoff between optimizing the model for computing hand configurations and computing key-pressing probabilities (see Appendix C). As a result, we reached significantly smaller errors in predicting joint values when training the encoder part of the CBMF model for longer epochs, implying that the model can overfit finger motions perfectly. However, we empirically discovered that for the second phase of training the CBMF model to converge as expected, it is better to bootstrap the model with parameters from an earlier step.

6.5 Learning to Apply Finger Pressure. Important to the continuation of this research is the potential of our method to be modified to answer different tasks, especially tasks that involve applying finger pressure. Among these is the need to approximate the force of the finger while pressing a note, as in professional musical instruments. Unlike previous papers that concentrated on extracting force from a single U.S. image [56,57], we are aiming to prepare the ground for robotic prostheses that can perform note pressing while changing the force of the finger pressing.

As for musical instruments, musical instrument digital interface (MIDI) [58] is a technical standard used to digitally describe musical compositions. Technically, a MIDI message is composed of the note's pitch, if it was pressed or released, and a 7-bit "velocity" (a number ranging from 0 to 127) that is used to indicate how forcefully the note was played. To demonstrate the model's ability to learn new tasks, we show that it can be easily adapted to predict MIDI velocities while a subject is playing the piano. To this end, we modified the proposed MF model by replacing the last layer with a linear layer that aggregates the temporal features of the previous (second-to-last) layer and outputs a single-valued MIDI velocity. Then, the model was trained in a similar fashion to the original MF model. Results, presented in Fig. 10, show the error as the percentage of the maximal error (i.e., $|127 - 0|$) in predicting MIDI values. Despite the variability across subjects' performance in piano playing (implying a wide range of MIDI values), the majority of the subjects achieved less than 10% out-of-sample error. It is worth noting that force estimation may be obtained from our original models, but here, we are interested in showcasing the ability to adapt to new tasks, which in our setting is learning a function that can directly generate the desired MIDI result.

6.6 Real-Time Deployment. To demonstrate the applicability of our method to deal with data in real-time, we built a real-time system with our CBMF model and connected it to a robotic hand (see Fig. 11 and our repository), which continuously replayed notes played on the piano in real-time. Importantly, our method works without any image preprocessing and fits the online setting. This is in contrast to existing methods for which subjects were instructed to perform gestures according to predefined instructions [7], or separate between gesture categories before feature extraction [11]. We provide the technical details of the implementation in Appendix D.

During real-time inference, the U.S. sensor is not placed in the exact location for which training data was obtained (also known as sensor reattachment). To this end, we define an "enrollment" as a period of 2–3 recording sessions, in which the U.S. sensor was not removed from the forearm. We used a single subject to demonstrate the number of enrollments needed in training in order to infer from an unseen enrollment. Specifically, we randomly sampled the number of enrollments used in training and then requested the subject to play the piano after wearing the sensor back on the arm.

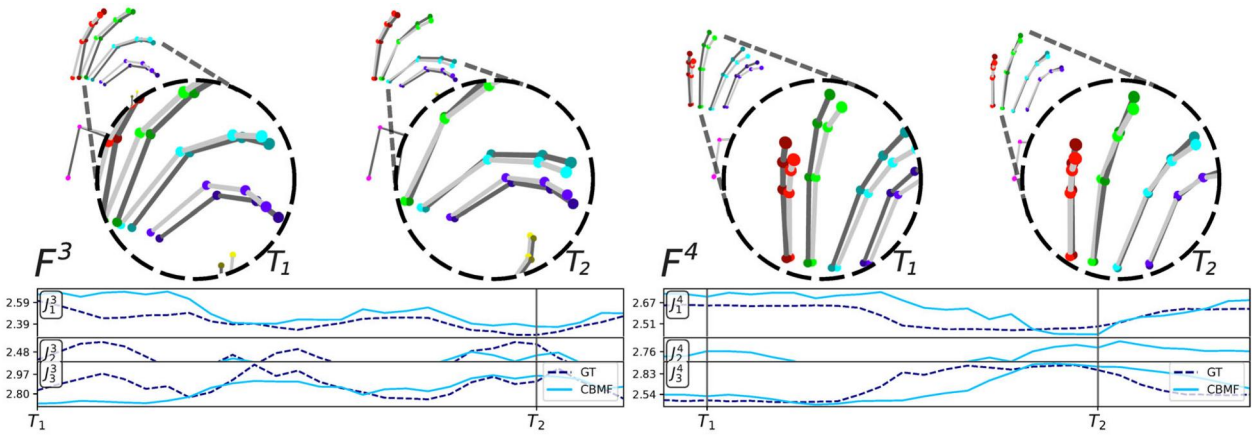


Fig. 9 (Top) Samples of hand configurations taken from the intermediate representation of the CBMF model (dark gray) as well as the ground truth (light gray). In each rollout, the three graphs represent the joint angles in radians of the relevant finger that is enlarged in the top two illustrations (i.e., F^3 and F^4 in left and right, respectively). This is done both for our model as well as for the ground truth.

Downloaded from asmedigitalcollection.asme.org/biomechanical/article-pdf/147/9/091005/7512464/bio-24-1272.pdf by guest on 22 September 2026

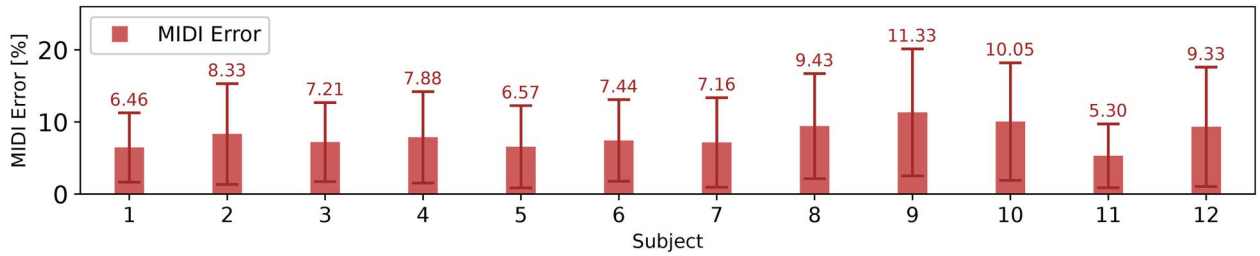


Fig. 10 The average error in predicting MIDI velocities for all 12 subjects. The error is reported as the percentage of the maximal error. Results are averaged using five-fold validation, with horizontal segments denoting one standard deviation.

We plot (Fig. 12(*a*)) the four metrics (Accuracy, Precision, Recall, and $F1$) as a function of the number of training enrollments. We can observe that (i) as the number of enrollments increases in the training session, our inference capabilities increase and (ii) the number of enrollments required in the training session is relatively moderate. In other words, our method provides a real-time system adaptable to sensor replacement, without large data requirements.

Figure 12(*b*) shows how the methods described in Sec. 6.3 behave under the same conditions. We focus on measuring $F1$, since it is composed of both recall and precision and allows us to count false positive and false negative predictions. We can see that not only does our CBMF model outperform any other method but it is the fastest method to achieve significant performance for smaller amounts of data. For example, to achieve $F1 = 0.6$, the CBMF model needs only 2–3 enrollments, while any other method requires at least twice the data to obtain the same performance.

7 Discussion and Future Work

In this paper, we explored the task of predicting and differentiating between fine finger motions by only having access to a stream of lower-arm U.S. images. Our findings demonstrate that incorporating temporal coherence and a kinematic hand representation using our CBMF model doubles the performance on the finer task of keyboard typing compared to the five baselines (see $F1$ values in Fig. 3). Furthermore, we show the versatility of our approach by demonstrating accurate estimation of finger kinematics (Sec. 6.4) and adapting the model to estimate finger pressure (Sec. 6.5). We believe that the inference capabilities we present will pave the way toward enabling fully operational prostheses for those in need.

7.1 Limitations. While U.S. probes offer better performance in detecting hand gestures [16], its use faces several limitations compared to EMG-based sensors. Notably, the delay in image acquisition, from radio frequency signal transmission to image retrieval, can impair the accurate detection of rapid finger motions, such as fast keyboard typing. Additionally, U.S. imaging consumes more energy compared to EMG, which may result in a reduced battery life for devices that use this technology. Addressing these concerns is important for the successful development of U.S.-based robotic hands. Importantly, the study did not address the challenge of generalizing results to previously unseen subjects, as this would require incorporating prior knowledge about muscle structure.

7.2 Future Work. Future directions that we are exploring include (i) restoring fine finger motions for people with muscular deformations such as different types of amputations or trauma conditions, where we cannot obtain accurate labels of intended motions. This will potentially require different data-driven solutions to learn continuous and unlabeled finger motions for nonable-bodied subjects. (ii) Recent advancements in feedback sensing for robotic

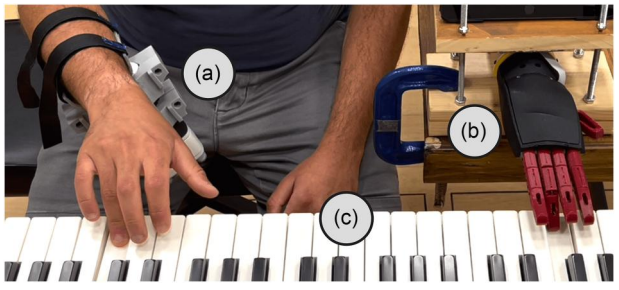


Fig. 11 Our real-time system implementation. (*a*) The subject wears the ultrasound probe, and the probe sends U.S. images to the algorithm. The algorithm computes predictions and sends instructing signals to (*b*) the robotic hand that flexes fingers to (*c*) play the piano.

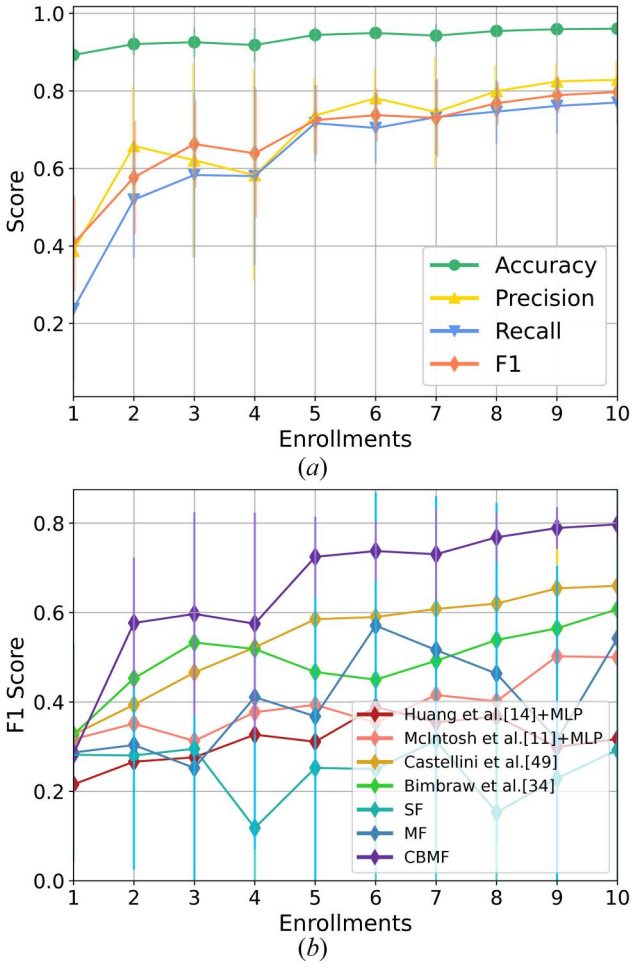


Fig. 12 Piano-playing evaluation given a certain number of training enrollments. At each step, we randomly add a new enrollment to the training set and evaluate an unseen enrollment. (*a*) Behavior of the CBMF model for the four metrics. (*b*) $F1$ values for each of the methods described in Sec. 6.3. Each point is an average over ten random experiments.

Downloaded from asmedigitalcollection.asme.org/biomechanical/article-pdf/147/9/091005/7512464/bio-24-1272.pdf by guest on 22 September 2026

prostheses allow amputees to sense objects they are engaging with [59]. Sensor feedback can also be used to endow our inference algorithm with additional input such as surface type or texture which can then be used to improve decision making. (iii) Commercially available robotic prostheses are often equipped with manual options that allow changing the system's behavior or parameters according to the desired task (e.g., grasping, pinching, etc.). We conjecture that our internal representation (possibly augmented) can assist with providing additional daily gestures, improve control of fingers' operation, and replace traditional approaches of manually deciding between operated gestures.

Acknowledgment

We thank Haifa3D, a nonprofit organization providing self-made 3d-printed solutions, for their support.

Funding Data

- The Technion Autonomous Systems Program (TASP).
- The European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (Grant No. 863839); Funder ID: 10.13039/501100000781.
- The Israeli Ministry of Science and Technology (MOST) Grant Nos. 3-16079 and 3-17385; Funder ID: 10.13039/501100006245.
- The United States-Israel Binational Science Foundation (BSF) Grant Nos. 2019703 and 2021643; Funder ID: 10.13039/100006221.

Data Availability Statement

The datasets generated and supporting the findings of this article are obtainable at the below link.[5]

Appendix A: Joints Computation

The set of joint angles forming a configuration is acquired by placing markers on the relevant joints and using a motion-capture system to obtain the $[x, y, z]$ location of each marker (see Fig. 13). Then, each joint angle is computed using the markers' locations as we detail next. Hereafter, we compute vectors as the point-wise subtraction of the marker locations. For example, given two 3D points (that can be considered as two 3D vectors with respect to some reference frame) f_{42} and f_{43}, we denote the vector from f_{42} to f_{43} as follows:

$$\mathbf{v}_{f_{42}\to f_{43}} = f_{43} - f_{42} \tag{A1}$$

Next, we describe how we compute the angle between two vectors. Consider, for example, the vector from f_{42} to f_{43} and the vector from f_{42} to f_{41}, we compute the joint angle J_2^5 as follows:

$$J_2^5 = \arctan 2(||\mathbf{v}_{f_{42}\to f_{41}} \times \mathbf{v}_{f_{42}\to f_{43}}||_2 \; \mathbf{v}_{f_{42}\to f_{41}} \cdot \mathbf{v}_{f_{42}\to f_{43}}) \tag{A2}$$

This allows us to compute joint angles. We start with the four fingers $F^2, \ldots, F^5$ and will use finger F^5 (Little finger), for example. The three joints that are required are J_1^5, J_2^5, and J_3^5. Hence, we will compute J_1^5 and J_3^5 as follows (fingers F^1, F^2, and F^3 are calculated in a similar fashion):

$$\begin{aligned} J_1^5 &= \arctan 2(||\mathbf{v}_{f_{41}\to t_5} \times \mathbf{v}_{f_{41}\to f_{42}}||_2 \; \mathbf{v}_{f_{41}\to t_5} \cdot \mathbf{v}_{f_{41}\to f_{42}}) \\ J_3^5 &= \arctan 2(||\mathbf{v}_{f_{42}\to f_{41}} \times \mathbf{v}_{f_{42}\to f_{43}}||_2 \; \mathbf{v}_{f_{42}\to f_{41}} \cdot \mathbf{v}_{f_{42}\to f_{43}}) \end{aligned} \tag{A3}$$

[5] https://github.com/deanzadok/finemotions

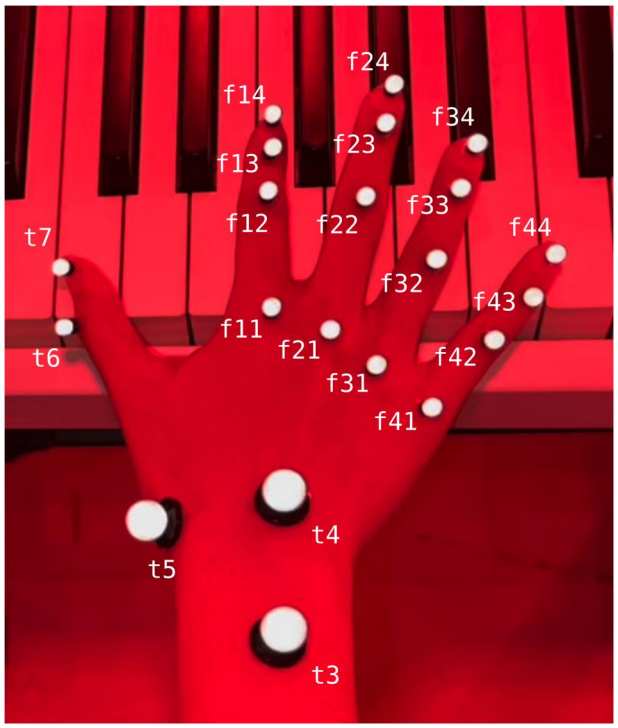


Fig. 13 An image showing the position of each marker on the subject's hand

As for the Thumb, we compute two joint angles; J_1^1 and J_2^1. Similar to Eq. (A3), we compute J_2^1 as follows:

$$J_2^1 = \arctan 2(||\mathbf{v}_{t_6\to t_7} \times \mathbf{v}_{t_6\to t_5}||_2 \; \mathbf{v}_{t_6\to t_7} \cdot \mathbf{v}_{t_6\to t_5}) \tag{A4}$$

where joint J_1^1 is acquired by computing the angle between $\mathbf{v}_{t_5\to t_6}$ and the normal vector to the plane approximating the palm's plane, by taking the three points f_{11}, f_{41}, and t_4. Denoting this normal as $\hat{\mathbf{n}}_p$, joint J_1^1 is computed as follows:

$$J_1^1 = \arctan 2(||\mathbf{v}_{t_5\to t_6} \times \hat{\mathbf{n}}_p||_2 \; \mathbf{v}_{t_5\to t_6} \cdot \hat{\mathbf{n}}_p) \tag{A5}$$

This is done differently in order to compute the angle that corresponds to the pressing motion of the Thumb. As for the three DOFs of the wrist, we have J_r^w, J_p^w, and J_y^w for roll, pitch, and yaw, respectively. As for pitch, we compute the angle between $\hat{\mathbf{n}}_p$ and $\mathbf{v}_{t_4\to t_3}$. For yaw, we compute the angle between $\mathbf{v}_{f_{41}\to f_{11}}$ and $\mathbf{v}_{t_4\to t_3}$, and for roll, we compute the angle between $\hat{\mathbf{n}}_p$ and a vector obtained from two additional points located on the elbow.

Appendix B: Baselines Implementation

As presented in Sec. 6.3, we implemented five baselines to asses our models in solving the two downstream tasks. The evaluation was using the same five-fold validation over the same datasets. We started by evaluating the importance of temporal coherence by defining the SF model. Here, we take our MF model and replace the RNN with an MLP, such that the CNN module is the same as in the MF and CBMF models and the addition is two linear layers. On inference, the model receives a single 224×224 U.S. image and predicts a single pressing vector.

In addition, we compared our work to four additional papers that succeeded in inferring finger motions from U.S. images. Importantly, all four papers were missing code repositories and therefore were implemented according to the explained text. The methods

Downloaded from asmedigitalcollection.asme.org/biomechanical/article-pdf/147/9/091005/7512464/bio-24-1272.pdf by guest on 22 September 2026

presented by Huang and Liu [16] and McIntosh et al. [11] were originally designed to classify full recorded sessions in an offline setting, and since these algorithms were evaluated on different gestures with different sensors, we modified the input to receive the last acquired eight images as in our models. Second, these algorithms were designed to discriminate between gestures and here we allow multiple fingers pressing at the same time. Therefore, we set the number of available classes to $2^5 = 32$ (i.e., a class corresponds to a subset of fingers pressed).

Huang and Liu [16] proposed to sample five columns of pixels from the image, imitating the feature structure of A-mode transducers. These columns are then separated into segments, from which they extracted two coefficients using linear fitting. Then, these features are concatenated across the temporal axis and fed to the classifier. Here, the authors used sequences of 36 images, but since we did not observe any improvement, the results are presented for sequences of eight images. Next, McIntosh et al. [11] proposed to extract the optical flow from adjacent frames, perform average pooling on patches of size 20×20, and accumulate them across the temporal axis. In the same way, we calculated these features for a sequence of eight U.S. images and fed them to the classifier. For both baselines, we evaluated all classifiers that were proposed by the authors, and in both cases, MLP outperformed all other classifiers.

In addition, we implemented the algorithm presented by Castellini et al. [51,52]. They proposed to compute linear regression on rounded patches to estimate the gray value given pixel coordinates, and taking the resulting coefficients as features. We chose their best solution of 180 patches (i.e., $180 \times 3 = 540$ features for image). The radius of the patches was decided using trial and error, and ridge regression was implemented for each finger separately as proposed in the paper.

Lastly, we implemented the neural network architecture presented by Bimbraw et al. [36] to infer finger positions from a single U.S. image. The architecture was implemented according to the explained in the paper, and was used to evaluate our backbone by changing the output to predict a pressing vector, and thus compare it to our SF model. Evidently, the SF model outperformed, leaving the same backbone to be used by our MF and CBMF models.

Appendix C: Losses Weighting

We empirically chose the value of λ to minimize the objective function $\lambda\mathcal{L}_{\text{MSE}} + \mathcal{L}_{\text{BCE}}$. Here, we executed training sessions to obtain results for $\lambda \in \{2^i | i \in [0, \ldots, 6]\}$ and chose the result yielding the lowest $\mathcal{L}_{\text{BCE}}$, without harming the optimization of $\mathcal{L}_{\text{MSE}}$. Results, presented in Fig. 14, show that by taking $\lambda = 4$, we benefit from using both modalities. The same experiment with $\lambda = 0$ (i.e., only using $\mathcal{L}_{\text{BCE}}$) resulted in $\mathcal{L}_{\text{BCE}} = 0.093$ and $\mathcal{L}_{\text{MSE}} = 2.582$, and therefore was ruled out from the evaluation.

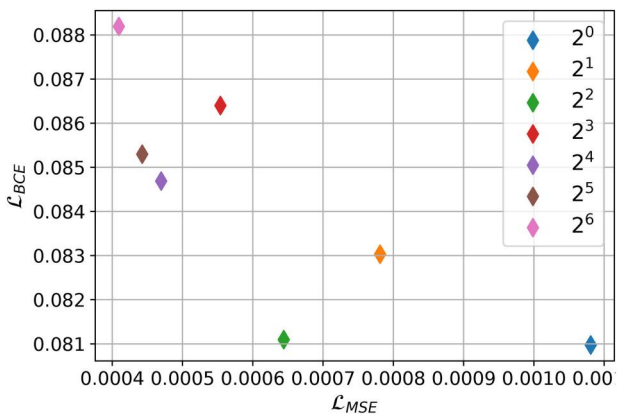


Fig. 14 $\mathcal{L}_{\text{BCE}}$ **and** $\mathcal{L}_{\text{MSE}}$ **for different values of** λ**. On average, results are showing that using lower values of** λ **yields lower values of** $\mathcal{L}_{\text{BCE}}$ **and higher values of** λ **yields lower values of** $\mathcal{L}_{\text{MSE}}$ **and higher values of** $\mathcal{L}_{\text{BCE}}$**.**

Appendix D: Real-Time Implementation

We built a setup for online real-time inference to demonstrate piano playing on demand. The code is written in C++ and includes multithreading to deal with interfaces to the U.S. sensor, the robotic hand, and model inference simultaneously. For optimized inference, our CBMF model was divided into two software modules—the encoder's convolution block and the remaining recurrent blocks including the residual connection. On inference, the incoming images from the U.S. sensor are fed to the first module to extract features, while an additional thread, simultaneously feeds the sequence of the last eight extracted features to the second module and stores the predictions as instructions that are sent to the robotic hand. The model was optimized using the ONNX Runtime library, the entire model weighs 745 MB and the inference is executed at 60 Hz on a powerful GPU (Nvidia RTX 2080Ti) or 29 Hz on a low-budget GPU (Nvidia Quadro P620).

Appendix E: Neural Network Architecture

Tables 1 and 2 show the network architectures for the MF and CBMF models, respectively. The tables are divided into blocks, such that the first block represents the CNN module and the others represent the RNN modules. The input shape for both models is $[224 \times 224 \times 8]$, corresponding to eight consecutive grayscale U.S. images. The shape of the intermediate representation is $[8 \times 17]$, corresponding to a trajectory of eight robotic configurations, each of size 17. The output shape is $[8 \times 5]$, denoting eight pressing vectors, each containing five probabilities for the five fingers.

The CNN module is based on the encoding part of UNet [60], due to the high success of this architecture when used in medical imaging tasks. This block is used to extract 512 3×3 feature maps from each of the eight images, such that the same parameters are shared across all images. These are flattened, concatenated, and fed into the recurrent block. In the CBMF model, the trajectory, along with 8×128 features are taken from the output of the second GRU layer, concatenated, and fed as an input to the decoder.

Table 1 Neural-network architecture of our MF model G_θ

Layer	Act.	Out. shape	Parameters
Conv2D	ReLU	$224 \times 224 \times 32$	320
Conv2D	ReLU	$224 \times 224 \times 32$	9248
MaxPool2D (2×2)	—	$112 \times 112 \times 32$	—
Conv2D	ReLU	$112 \times 112 \times 64$	18.50k
Conv2D	ReLU	$112 \times 112 \times 64$	36.93k
MaxPool2D (2×2)	—	$56 \times 56 \times 64$	—
Conv2D	ReLU	$56 \times 56 \times 128$	73.86k
Conv2D	ReLU	$56 \times 56 \times 128$	147.58k
MaxPool2D (2×2)	—	$28 \times 28 \times 128$	—
Conv2D	ReLU	$28 \times 28 \times 256$	295.17k
Conv2D	ReLU	$28 \times 28 \times 256$	590.08k
Dropout ($p = 0.3$)	—	$28 \times 28 \times 256$	—
MaxPool2D (2×2)	—	$14 \times 14 \times 256$	—
Conv2D	ReLU	$14 \times 14 \times 512$	1.18 M
Conv2D	ReLU	$14 \times 14 \times 512$	2.36 M
Dropout ($p = 0.3$)	—	$14 \times 14 \times 512$	—
MaxPool2D (4×4)	—	$3 \times 3 \times 512$	—
Flatten	—	4608	—
Concatenate ($\times 8$)	—	8×4608	—
GRU	ReLU	8×1024	17.31 M
GRU	ReLU	8×128	443.14 K
GRU	Linear	8×5	2025
Total			22.93 M

Downloaded from asmedigitalcollection.asme.org/biomechanical/article-pdf/147/9/091005/7512464/bio-24-1272.pdf by guest on 22 September 2026

Table 2 Neural-network architecture of our CBMF model $H_{\theta,\phi}$

Layer	Act.	Out. shape	Parameters
Conv2D	ReLU	224 × 224 × 32	320
Conv2D	ReLU	224 × 224 × 32	9248
MaxPool2D (2 × 2)	—	112 × 112 × 32	—
Conv2D	ReLU	112 × 112 × 64	18.50k
Conv2D	ReLU	112 × 112 × 64	36.93k
MaxPool2D (2 × 2)	—	56 × 56 × 64	—
Conv2D	ReLU	56 × 56 × 128	73.86k
Conv2D	ReLU	56 × 56 × 128	147.58k
MaxPool2D (2 × 2)	—	28 × 28 × 128	—
Conv2D	ReLU	28 × 28 × 256	295.17k
Conv2D	ReLU	28 × 28 × 256	590.08k
Dropout ($p = 0.3$)	—	28 × 28 × 256	—
MaxPool2D (2 × 2)	—	14 × 14 × 256	—
Conv2D	ReLU	14 × 14 × 512	1.18 M
Conv2D	ReLU	14 × 14 × 512	2.36 M
Dropout ($p = 0.3$)	—	14 × 14 × 512	—
MaxPool2D (4 × 4)	—	3 × 3 × 512	—
Flatten	—	4608	—
Concatenate (× 8)	—	8 × 4608	—
GRU	ReLU	8 × 1024	17.31 M
GRU	ReLU	8 × 128	443.14 K
GRU	Linear	8 × 17	7497
Merge	—	8 × 145	—
GRU	ReLU	8 × 256	309.50 K
GRU	ReLU	8 × 128	148.22 K
GRU	Linear	8 × 5	2025
Total			22.93 M

References

[1] Connolly, C., 2008, "Prosthetic Hands From Touch Bionics," Ind. Rob., **35**(4), pp. 290–293.

[2] Herberts, P., 1969, "Myoelectric Signals in Control of Prostheses: Studies on Arm Amputees and Normal Individuals," Acta Orthop., **40**(sup124), pp. 1–83.

[3] George, J. A., Davis, T. S., Brinton, M. R., and Clark, G. A., 2020, "Intuitive Neuromyoelectric Control of a Dexterous Bionic Arm Using a Modified Kalman Filter," J. Neurosci. Methods, **330**, p. 108462.

[4] Said, S., Sheikh, M., Al-Rashidi, F., Lakys, Y., Beyrouthy, T., and Nait-Ali, A., 2019, "A Customizable Wearable Robust 3d Printed Bionic Arm: Muscle Controlled," Proceedings of the 2019 3rd International Conference on Bio-Engineering for Smart Technologies (BioSMART), Paris, France, Apr. 24–26, pp. 1–6.

[5] Armiger, R. S., and Vogelstein, R. J., 2008, "Air-Guitar Hero: A Real-Time Video Game Interface for Training and Evaluation of Dexterous Upper-Extremity Neuroprosthetic Control Algorithms," Proceedings of the 2008 IEEE Biomedical Circuits and Systems Conference, Baltimore, MD, Nov. 20–22, pp. 121–124.

[6] Vásconez, J. P., López, L. I. B., Caraguay, Á. L. V., and Benalcázar, M. E., 2022, "Hand Gesture Recognition Using EMG-IMU Signals and Deep Q-Networks," Sensors, **22**(24), p. 9613.

[7] Dhawan, A. S., Mukherjee, B., Patwardhan, S., Akhlaghi, N., Diao, G., Levay, G., Holley, R., Joiner, W. M., Harris-Love, M., and Sikdar, S., 2019, "Proprioceptive Sonomyographic Control: A Novel Method for Intuitive and Proportional Control of Multiple Degrees-of-Freedom for Individuals with Upper Extremity Limb Loss," Scientific Rep., **9**(1), p. 9499.

[8] Yan, J., Yang, X., Sun, X., Chen, Z., and Liu, H., 2019, "A Lightweight Ultrasound Probe for Wearable Human–Machine Interfaces," IEEE Sens. J., **19**(14), pp. 5895–5903.

[9] Yang, X., Zhou, D., Zhou, Y., Huang, Y., and Liu, H., 2019, "Towards Zero Re-Training for Long-Term Hand Gesture Recognition Via Ultrasound Sensing," IEEE J. Biomed. Health Inf., **23**(4), pp. 1639–1646.

[10] Yang, X., Yan, J., Fang, Y., Zhou, D., and Liu, H., 2020, "Simultaneous Prediction of Wrist/Hand Motion Via Wearable Ultrasound Sensing," IEEE Trans. Neural Syst. Rehabil. Eng., **28**(4), pp. 970–977.

[11] McIntosh, J., Marzo, A., Fraser, M., and Phillips, C., 2017, "EchoFlex: Hand Gesture Recognition Using Ultrasound Imaging," Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, Denver, CO, pp. 1923–1934.

[12] Cerveri, P., De Momi, E., Lopomo, N., Baud-Bovy, G., Barros, R., and Ferrigno, G., 2007, "Finger Kinematic Modeling and Real-Time Hand Motion Estimation," Ann. Biomed. Eng., **35**(11), pp. 1989–2002.

[13] Milojević, A., Linß, S., Cojbašić, Ž., and Handroos, H., 2021, "A Novel Simple, Adaptive, and Versatile Soft-Robotic Compliant Two-Finger Gripper With an Inherently Gentle Touch," ASME J. Mech. Rob., **13**(1), p. 011015.

[14] Resnik, L., Meucci, M. R., Lieberman-Klinger, S., Fantini, C., Kelty, D. L., Disla, R., and Sasson, N., 2012, "Advanced Upper Limb Prosthetic Devices: Implications for Upper Limb Prosthetic Rehabilitation," Arch. Phys. Med. Rehabil., **93**(4), pp. 710–717.

[15] Côté-Allard, U., Fall, C. L., Drouin, A., Campeau-Lecours, A., Gosselin, C., Glette, K., Laviolette, F., and Gosselin, B., 2019, "Deep Learning for Electromyographic Hand Gesture Signal Classification by Leveraging Transfer Learning," IEEE Trans. Neural Syst. Rehab. Eng., **27**(4), pp. 760–771.

[16] Huang, Y., and Liu, H., 2016, "Performances of Surface EMG and Ultrasound Signals in Recognizing Finger Motion," Proceedings of the 2016 9th International Conference on Human System Interactions (HSI), Portsmouth, UK, pp. 117–122.

[17] Zhang, Q., Pi, T., Liu, R., and Xiong, C., 2020, "Simultaneous and Proportional Estimation of Multijoint Kinematics From EMG Signals for Myocontrol of Robotic Hands," IEEE/ASME Trans. Mechatron., **25**(4), pp. 1953–1960.

[18] Simpetru, R. C., März, M., and Vecchio, A. D., 2023, "Proportional and Simultaneous Real-Time Control of the Full Human Hand From High-Density Electromyography," IEEE Trans. Neural Syst. Rehabil. Eng., **31**, pp. 3118–3131.

[19] Li, J., Yue, S., and Pan, L., 2023, "Simultaneous and Proportional Control of Wrist and Hand Movements Based on a Neural-Driven Musculoskeletal Model," IEEE Trans. Neural Syst. Rehabil. Eng., **31**, pp. 3999–4007.

[20] Li, D., Kang, P., Yu, Y., and Shull, P. B., 2024, "Graph-Driven Simultaneous and Proportional Estimation of Wrist Angle and Grasp Force Via High-Density EMG," IEEE J. Biomed. Health Inf., **28**(5), pp. 2723–2732.

[21] Nowak, M., Vujaklija, I., Sturma, A., Castellini, C., and Farina, D., 2023, "Simultaneous and Proportional Real-Time Myocontrol of up to Three Degrees of Freedom of the Wrist and Hand," IEEE Trans. Biomed. Eng., **70**(2), pp. 459–469.

[22] Herbst, Y., Zelnik-Manor, L., and Wolf, A., 2020, "Analysis of Subject Specific Grasping Patterns," PLoS One, **15**(7), p. e0234969.

[23] Bimbraw, K., Fox, E., Weinberg, G., and Hammond, F. L., 2020, "Towards Sonomyography-Based Real-Time Control of Powered Prosthesis Grasp Synergies," Proceedings of the 2020 42nd Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Montreal, QC, Canada, July 20–24, pp. 4753–4757.

[24] Yang, X., Chen, Z., Hettiarachchi, N., Yan, J., and Liu, H., 2021, "A Wearable Ultrasound System for Sensing Muscular Morphological Deformations," IEEE Trans. Syst., Man, Cybern.: Syst., **51**(6), pp. 3370–3379.

[25] Sgambato, B. G., Hasbani, M. H., Barsakcioglu, D. Y., Ibáñez, J., Jakob, A., Fournelle, M., Tang, M., and Farina, D., 2024, "High Performance Wearable Ultrasound as a Human-Machine Interface for Wrist and Hand Kinematic Tracking," IEEE Trans. Biomed. Eng., **71**(2), pp. 484–493.

[26] Bimbraw, K., Nycz, C. J., Schueler, M. J., Zhang, Z., and Zhang, H. K., 2022, "Prediction of Metacarpophalangeal Joint Angles and Classification of Hand Configurations Based on Ultrasound Imaging of the Forearm," Proceedings of the 2022 International Conference on Robotics and Automation (ICRA), Philadelphia, PA, May 23–27, pp. 91–97.

[27] Vostrikov, S., Anderegg, M., Benini, L., and Cossettini, A., 2024, "Unsupervised Feature Extraction From Raw Data for Gesture Recognition With Wearable Ultra Low-Power Ultrasound," IEEE Trans. Ultrason., Ferroelectr., Freq. Control, **71**(7), pp. 831–841.

[28] Wang, Y., Ge, X., Ma, H., Qi, S., Zhang, G., and Yao, Y., 2021, "Deep Learning in Medical Ultrasound Image Analysis: A Review," IEEE Access, **9**, pp. 54310–54324.

[29] Vedula, S., Senouf, O., Bronstein, A. M., Michailovich, O. V., and Zibulevsky, M., 2017, "Towards CT-Quality Ultrasound Imaging Using Deep Learning," e-print arXiv:1710.06304.

[30] van Sloun, R. J. G., Cohen, R., and Eldar, Y. C., 2019, "Deep Learning in Ultrasound Imaging," Proc. IEEE, **108**(1), pp. 11–29.

[31] Vedula, S., Senouf, O., Zurakhov, G., Bronstein, A. M., Michailovich, O. V., and Zibulevsky, M., 2019, "Learning Beamforming in Ultrasound Imaging," Proceedings of the 2nd International Conference on Medical Imaging with Deep Learning, Vol. 102, pp. 493–511.

[32] Bliss, H., Bird, S., Cooper, A. P., Burton, S., and Gick, B., 2018, "*Seeing Speech: Ultrasound-Based Multimedia Resources for Pronunciation Learning in Indigenous Languages*," University of Hawaii Press, Honolulu, HI.

[33] Kimura, N., Kono, M., and Rekimoto, J., 2019, "SottoVoce: An Ultrasound Imaging-Based Silent Speech Interaction Using Deep Neural Networks," Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, Glasgow, UK, p. 146.

[34] Vogt, F., McCaig, G., Ali, M. A., and Fels, S. S., 2002, "Tongue 'n' Groove: An Ultrasound Based Music Controller," Proceedings of the 2002 Conference on New Instruments for Musical Expression (NIME-02), Dublin, Ireland, May 24–26.

[35] Lulich, S. M., Charles, S., and Lulich, B., 2017, "The Relation Between Tongue Shape and Pitch in Clarinet Playing Using Ultrasound Measurements," J. Acoust. Soc. Am., **141**(3), pp. 1759–1768.

[36] Bimbraw, K., Nycz, C. J., Schueler, M. J., Zhang, Z., and Zhang, H. K., 2023, "Simultaneous Estimation of Hand Configurations and Finger Joint Angles Using Forearm Ultrasound," IEEE Trans. Med. Rob. Bionics, **5**(1), pp. 120–132.

[37] Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., and Wang, G., 2015, "Recent Advances in Convolutional Neural Networks," Pattern Recognit., **77**, pp. 354–377.

[38] Rumelhart, D. E., Hinton, G. E., and Williams, R. J., 1985, "Learning Internal Representations by Error Propagation," Institute for Cognitive Science, University of California, San Diego, CA.

[39] Sherstinsky, A., 2018, "Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network," Phys. D: Nonlinear Phenom., **404**, p. 132306.

[40] Guo, J., Liu, Y., Yang, Q., Wang, Y., and Fang, S., 2021, "GPS-Based Citywide Traffic Congestion Forecasting Using CNN-RNN and C3D Hybrid Model," Transportmetrica A, **17**(2), pp. 190–211.

Downloaded from asmedigitalcollection.asme.org/biomechanical/article-pdf/147/9/091005/7512464/bio-24-1272.pdf by guest on 22 September 2026

[41] Lu, W., Li, J., Li, Y., Sun, A., and Wang, J., 2020, "A CNN-LSTM-Based Model to Forecast Stock Prices," Complexity, **2020**(1), pp. 1–10.
[42] Zou, Q., Jiang, H., Dai, Q., Yue, Y., Chen, L., and Wang, Q., 2020, "Robust Lane Detection From Continuous Driving Scenes Using Deep Neural Networks," IEEE Trans. Veh. Technol., **69**(1), pp. 41–54.
[43] Baldi, P., 2012, "Autoencoders, Unsupervised Learning, and Deep Architectures," Proceedings of the ICML Workshop on Unsupervised and Transfer Learning. JMLR Workshop and Conference Proceedings, Bellevue, WA, pp. 37–50.
[44] He, K., Zhang, X., Ren, S., and Sun, J., 2016, "Deep Residual Learning for Image Recognition," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, June 27–30, pp. 770–778.
[45] Hu, J., Guo, X., Chen, J., Liang, G., Deng, F., and Lam, T. L., 2021, "A Two-Stage Unsupervised Approach for Low Light Image Enhancement," IEEE Rob. Autom. Lett., **6**(4), pp. 8363–8370.
[46] Yang, Y., Li, B., Li, P., and Liu, Q., 2019, "A Two-Stage Clustering Based 3D Visual Saliency Model for Dynamic Scenarios," IEEE Trans. Multimedia, **21**(4), pp. 809–820.
[47] Singh, A., Liu, H., Zhou, G., Yu, A., Rhinehart, N., and Levine, S., 2021, "Parrot: Data-Driven Behavioral Priors for Reinforcement Learning," e-print arXiv:2011.10024.
[48] Tan, C., Sun, F., Fang, B., Kong, T., and Zhang, W., 2019, "Autoencoder-Based Transfer Learning in Brain–Computer Interface for Rehabilitation Robot," Int. J. Adv. Rob. Syst., **16**(2), p. 1729881419840860.
[49] Zadok, D., McDuff, D., and Kapoor, A., 2021, "Modeling Affect-Based Intrinsic Rewards for Exploration and Learning," Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, May 30–June 5, pp. 13078–13084.
[50] Kingma, D. P., and Ba, J., 2015, "Adam: A Method for Stochastic Optimization," e-print arXiv:1412.6980.
[51] Castellini, C., and Passig, G., 2011, "Ultrasound Image Features of the Wrist Are Linearly Related to Finger Positions," Proceedings of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, San Francisco, CA, Sept. 25–30, pp. 2108–2114.
[52] Castellini, C., Hertkorn, K., Sagardia, M., González, D. S., and Nowak, M., 2014, "A Virtual Piano-Playing Environment for Rehabilitation Based Upon Ultrasound Imaging," Proceedings of the 5th IEEE RAS/EMBS International Conference on Biomedical Robotics and Biomechatronics, Sao Paulo, Brazil, Aug. 12–15, pp. 548–554.
[53] Simonyan, K., and Zisserman, A., 2015, "Very Deep Convolutional Networks for Large-Scale Image Recognition," e-print arXiv:1409.1556.
[54] Deng, J., Dong, W., Socher, R., Li, L., Li, K., and Fei-Fei, L., 2009, "ImageNet: A Large-Scale Hierarchical Image Database," Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, June 20–25, pp. 248–255.
[55] Tran, D., Bourdev, L. D., Fergus, R., Torresani, L., and Paluri, M., 2015, "Learning Spatiotemporal Features With 3D Convolutional Networks," Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, Dec. 7–13, pp. 4489–4497.
[56] Zhou, Y., Zeng, J., Yang, Y., and Liu, H., 2019, "Finger Position and Force Simultaneous Prediction Using A-Mode Ultrasound," Proceedings of the 2019 IEEE 9th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER), Suzhou, China, July 29–Aug. 2, pp. 242–246.
[57] Kamatham, A. T., Alzamani, M., Dockum, A., Sikdar, S., and Mukherjee, B., 2024, "SonoMyoNet: A Convolutional Neural Network for Predicting Isometric Force From Highly Sparse Ultrasound Images," IEEE Trans. Hum. Mach. Syst., **54**(3), pp. 317–324.
[58] Caldwell, D., 1995, "Standard MIDI File Format," The Hong Kong Polytechnic University, Hong Kong.
[59] Hari, A. M., and Rajan, L., 2021, "Advanced Materials and Technologies for Touch Sensing in Prosthetic Limbs," IEEE Trans. NanoBiosci., **20**(3), pp. 256–270.
[60] Ronneberger, O., Fischer, P., and Brox, T., 2015, "U-Net: Convolutional Networks for Biomedical Image Segmentation," Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, Oct. 5–9, pp. 234–241.

Downloaded from asmedigitalcollection.asme.org/biomechanical/article-pdf/147/9/091005/7512464/bio-24-1272.pdf by guest on 22 September 2026