Simultaneous Estimation of Hand Configurations and Finger Joint Angles Using Forearm Ultrasound

Keshav Bimbraw, Graduate Student Member, IEEE, Christopher J. Nycz, Matthew Schueler, Ziming Zhang, and Haichong K. Zhang, Member, IEEE

Abstract—With the advancement in computing and robotics, it is necessary to develop fluent and intuitive methods for interacting with digital systems, augmented/virtual reality (AR/VR) interfaces, and physical robotic systems. Hand motion recognition is widely used to enable these interactions. Hand configuration classification and metacarpophalangeal (MCP) joint angle detection is important for a comprehensive reconstruction of hand motion. Surface electromyography (sEMG) and other technologies have been used for the detection of hand motions. Forearm ultrasound images provide a musculoskeletal visualization that can be used to understand hand motion. Recent work has shown that these ultrasound images can be classified using machine learning to estimate discrete hand configurations. Estimating both hand configuration and MCP joint angles based on forearm ultrasound has not been addressed in the literature. In this paper, we propose a convolutional neural network (CNN) based deep learning pipeline for predicting the MCP joint angles. The results for the hand configuration classification were compared by using different machine learning algorithms. Support vector classifier with different kernels, multi-layer perceptron, and the proposed CNN have been used to classify the ultrasound images into 11 hand configurations based on activities of daily living. Forearm ultrasound images were acquired from 6 subjects instructed to move their hands according to predefined hand configurations. Motion capture data was acquired to get the finger angles corresponding to the hand movements at different speeds (0.5 Hz, 1 Hz, & 2 Hz) for the index, middle, ring, and pinky fingers. Average classification accuracy of 82.7 ± 9.7% for the proposed CNN and over 80% for SVC for different kernels was observed on a subset of the dataset. An average RMSE of 7.35°±1.3° was obtained between the predicted and the true MCP joint angles. A low latency (6.25 - 9.1 Hz) pipeline has been proposed for estimating both MCP joint angles and hand configuration aimed at real-time control of human-machine interfaces.

Index Terms—AI-enabled robotics, gesture, posture, and facial expressions, wearable robotics, design and development of robots for human-robot interaction, human-machine interfaces and robotics applications, new technologies and methodologies in medical robotics, wearable sensor systems, user-centered design and applications.

I. INTRODUCTION

Smart and intuitive upper limb interaction with physical and non-physical worlds (AR/VR) has an emerging interest in the human-computer interaction research community and has numerous interfacing and control applications [1], [2], [3], [4], [5], [6]. Upper limb motor dexterity in humans is possible with a highly advanced neuromuscular machinery developed over millions of years of evolution from primitive primates with prehensile appendages to the upper limbs. Human upper limbs have the maximum proportion of motor-sensory innervation in the human body and are used to interact with digital systems, augmented/virtual reality (AR/VR) interfaces, and physical robotic systems [7]. While carrying out these activities of daily living (ADLs), the primary joint for the fingers is the metacarpophalangeal (MCP) joint and it plays a major role in manipulation tasks [2]. It is thus important to estimate hand configurations and MCP joint angles to facilitate this interaction.

Various technologies have been used for hand configuration estimation and interpreting hand movements through sensing regions on the hand or the forearm [8]. These include surface electromyography (sEMG) [9], force myography (FMG) [10], vision-based approaches [11], resistive hand gloves [12], depth information-based approaches [13] and WiFi sensing [14]. However, these state-of-the-art methods have limitations. Placing sensors on the fingers, such as bend sensors and motion sensors can limit mobility and usage of the hands to their full extent for the individuals wearing such sensors [15]. Vision-based methods for hand state and orientation detection are often sensitive to poor resolution, frame rate, drastic illumination conditions, changing weather conditions and occlusions [16], [17].

Biological signals from the forearm can be used as an alternative to understanding hand movement without impeding the user and preventing difficult and awkward hand
movements [18], sEMG has been used for the recognition of hand motion and can facilitate human-robot-environment integration with the help of an intelligent robot perception system [19]. It has been researched for its use as a control input to prosthetics and powered orthotics [20]. Various combinations of sensors with sEMG have also been explored for upper and lower limb motion prediction [21], [22], [23], [24]. Araki et al. used sEMG to measure the middle finger metacarpophalangeal and proximal inter-phalangeal joint angles and used it to control a robotic finger [25]. Shrirao et al. used sEMG to predict index finger metacarpophalangeal joint angle while performing flexion-extension rotation of the index finger [26]. Wang et al. used sEMG to predict finger joint angles during different grasps [27]. Deep Learning has also been used for facilitating sEMG-based human computer interaction [28]. Tosin et al. provide a current review of sEMG-based upper limb movement classifiers [29] and Palumbo et al. provide a review of biopotential signal monitoring in rehabilitation [30]. Several commercial solutions have been created based on sEMG data processing for hand motion prediction in the context of prosthetics including Coapt Gen2 and Myo Plus Ottobock. However, these commercial systems have not been widely accepted to replace the traditional direct control [31]. sEMG is sensitive to muscle fatigue because of long-term muscle movements and works best for fast movements of the hand which can inhibit human-like control of digital and virtual interfaces, and prosthetics/orthotics [19], [26]. Muscle fatigue produces a decrease in the mean frequency and has a variable effect on the sEMG signal amplitude [32]. These biological signal limitations in addition to signal contamination due to motion artifact, and electromagnetic, and environmental interference might introduce variation in time, frequency, and statistical properties of the signal and are an impediment to easy adoption [29].

A. Related Works

While sEMG can be used to give an estimate of the muscle activation for hand and finger movements, ultrasound imaging of the forearm can give a 2-dimensional visualization of the musculoskeletal structure of the forearm. Ultrasound imaging of the forearm, or Sonomyography, has been explored as an alternative sensing modality that has been shown to be capable of identifying different hand gestures and finger movements from ultrasound data with a combination of image processing and classification algorithms [33], [34], [35], [36]. Sonomyography has also been used to classify several distinct hand configurations and controlling robotic mechanisms [33]. Akhlagi et al. demonstrated the classification of 15 hand motions with an average classification accuracy of 91% using Nearest Neighbor classification algorithms [37].

Yang et al. described the estimation of simultaneous and distinct wrist pronation-supination and hand opening-closing based on ultrasound imaging using subclass discriminant analysis (SDA) and principal component analysis [38]. They were able to achieve a wrist rotation classification accuracy of 99.2% and a hand gesture classification of 92.8%. McIntosh et al. classified 10 hand gestures with an average accuracy percentage of over 99% based on small ultrasound data sizes using support vector machines (SVMs) and multi-layer perceptrons (MLPs) [39]. Huang et al. compared the classification performance based on ultrasound and sEMG data for 14 finger motions [40]. They treated their ultrasound data as A-mode ultrasound signals and did not consider the spatial features which could be acquired from the full B-mode ultrasound images since according to them it was redundant.

B. Contributions

While discrete hand configuration classification using forearm ultrasound images has been implemented previously, there remains to be a need for the prediction of the finger joint angles continuously for effectively controlling systems and environments. This can be facilitated by using deep learning based algorithms and techniques to estimate hand configurations relevant to ADLs, and MCP joint angles while attaining them. Since hand gestures are a primary way to convey information because of a high degree of differentiation, flexibility, and efficiency of information transmission for human-machine interfacing, it is important to estimate them for effective human-machine interaction [12]. This is especially useful for developing technology as a communication medium for people who cannot communicate verbally [8]. There is also an active research community focused on robotics applications related to the development of rehabilitation tools and robots which can assist with upper limb rehabilitation [41], [42]. This work can be beneficial to these domains because of a direct estimation from a physiological visualization of the musculoskeletal structure of the forearm. This paper discusses a pipeline for estimating these hand configurations and MCP joint angles. A convolutional neural network (CNN) has been proposed for estimating MCP joint angles. The hand classification accuracy results have been compared by using different machine learning algorithms. Results for hand configuration classification using a support vector classifier (SVC) with different kernels, a multi-layer perceptron (MLP), and the proposed CNN have been discussed. Results from simultaneous hand configuration and angle prediction pipeline have also been described.

II. METHODS

A low-latency solution to demonstrate the performance of the combined hand configuration classification and finger angle prediction pipeline is presented. SVC, MLP, and the proposed CNN were used for hand configuration classification, and the same CNN architecture was used for continuously predicting MCP joint angles based on the forearm ultrasound.

A. Combined System

The combined system demonstration pipeline uses pre-trained models to predict both hand configurations and MCP joint angles based on forearm ultrasound images. First, a forearm ultrasound image is loaded. Then, a pre-trained model classifies the data into one of the several hand configuration classes. Based on this classification, another pre-trained model corresponding to the hand configuration is loaded. This model makes a prediction based on the loaded ultrasound image
frame. The index, middle, ring, and pinky finger MCP angle prediction plot is then displayed. This is done repeatedly for a given number of ultrasound data frames which can be specified by the user to produce the output for the combined pipeline shown in Fig. 1.

B. Hand Configuration Classification

For hand configuration classification based on ultrasound images from the forearm, SVC with different kernels, a multi-layer perceptron (MLP) network, and a convolutional neural network (CNN) were used. The SVC models were trained for three different kernels, namely linear (SVC-Lin), radial basis function (SVC-RBF), and third-degree polynomial (SVC-Ply). For each subject and speed, 5.6 seconds of training data and 2.4 seconds of test data (70% train-test split on 8 seconds of data) was acquired per hand configuration. For the 11 configurations, this lead to 88 seconds of data per speed per subject being used for training and testing the models. Different classifiers were used on full image sizes (636 x 256 pixels corresponding to 44.52 mm depth and 50 mm width forearm ultrasound images) and down-sampled image sizes (169 x 64 pixels, full images down-sampled by a factor of 4). The downsampling was done as a way to validate the performance of the classifiers at a lower resolution.

1) Support Vector Classifier: Support vector machines (SVMs) are a class of supervised machine learning algorithms that are used for image classification and regression problems. A support vector classifier (SVC) constructs hyperplanes in high-dimensional spaces, wherein a good separation is achieved by the hyperplane that has the largest distance to the nearest training-data point of any class. For ultrasound forearm image classification to predict hand configurations, SVC and other supervised machine learning algorithms have been explored in the past [33], [39]. Two-dimensional ultrasound images are the input to the algorithm, and the hand configuration class is the corresponding ground truth. After training the SVM model, ultrasound images not used for the training can be used to make a prediction of the corresponding hand configuration.

2) Multi Layer Perceptron: An MLP is a fully connected feedforward neural network. A 5-layer MLP was used to classify the 11 hand configurations based on forearm ultrasound.

3) Convolutional Neural Network: The CNN architecture used for the classification is the same which was used for MCP joint angle estimation and is explained in Section II-C. The network can be visualized in Fig. 2.

The 2D images were first flattened into a vector and a Dense layer with a dimensionality of 128 and the ReLU activation function, the three successive layers with dimensions 64, 32, and 16, with the same activation as the first hidden layer. The final layer had 11 elements and the Softmax activation function to make the prediction.

Fig. 1. The combined pipeline for hand configuration classification and MCP joint angle prediction: (a) An example of the input ultrasound image, (b) the data processing pipeline, and (c) an example of the output prediction plot. The crosses are the true values and the inverted triangles are the predicted values.
C. MCP Joint Angle Estimation

MCP joint angle prediction was implemented using a CNN. CNNs are a class of deep learning algorithms popularly used for two-dimensional image data regression-based prediction. The CNN architecture that was used in the paper is based on a modified form of VGG16 [43]. The proposed CNN network has 5 convolutional layers each followed by batch normalization and a max pooling layer. This is different from the original VGG16 since it uses two and three cascaded convolutional layers for each of the 5 different convolutional sections of the network. Fig. 2 shows the proposed CNN model.

The conv2d layer creates a convolution kernel that is convolved with the input layer to produce a tensor of outputs. The output $Z_1$ after the convolution operation ($*$) is defined in (1).

$$ Z_1 = X * f $$

where $X$ is the input image and $f$ is the filter or the convolution kernel. For the proposed model, each conv2d layer has a kernel size of 3 x 3 pixels, with ReLU activation and padding with zeros evenly around the input. The ReLU activation function is defined in [44]. The output $A_1$ after the activation function $relu$ is applied is defined in (2).

$$ A_1 = relu(Z_1) $$

The batch normalization layer normalizes its output using the mean and standard deviation of the current batch of inputs from the output of the previous layer $A_1$. The batch normalization is defined in [45]. The output $B_1$ after applying the batch normalization function $bn$ to $A_1$ is defined in (3).

$$ B_1 = bn(A_1) $$

The max pooling layer downsamples the input along its spatial dimensions by taking the maximum value over an input window of size 2 x 2 pixels for each channel of the input. The window is shifted by strides of 1 pixel along each dimension. Max pooling operation is defined in [46]. The output $M_1$ after applying the max pooling function $mp$ to $B_1$ is defined in (4).

$$ M_1 = mp(B_1) $$

This is repeated 5 times, with the successive convolution operations $Z_2$, $Z_3$, $Z_4$, $Z_5$, successive ReLU activation function outputs $A_2$, $A_3$, $A_4$, $A_5$, successive batch normalization function outputs $B_2$, $B_3$, $B_4$, $B_5$, and successive max pooling function outputs $M_2$, $M_3$, $M_4$, $M_5$. Therefore, the output $O_1$ after all of these operations is defined in (5). Each conv2d ($*$)-relu-bn-mp pair is enclosed in curly brackets.

$$ O_1 = [M_5(B_5(A_5(Z_5(M_4(A_4(Z_4(M_3(B_3(\ldots A_3(Z_3(M_2(B_2(A_2(Z_2(M_1))))))\))))))))))) $$ (5)

The flatten function $fl$ reshapes the output of the previous max pooling layer $O_1$ and reduces its dimension to the form of $(1xn)$ where $n$ is the product of the dimensions of the previous layer. The output $FL$ after applying the flatten function $fl$ is defined in (6).

$$ FL = fl(O_1) $$ (6)

The first dense layer is a deeply connected 16-unit neural network layer with ReLU activation. The output $D_1$ of the densely connected neural network layer is defined in (7).

$$ D_1 = relu(FL \cdot guf) $$ (7)

where $guf$ is the weights matrix created by the layer initialized by the Glorot/Xavier uniform initializer [47]. The ($) denotes the dot product of the $FL$ and the $guf$. This is followed by batch normalization function $bn$ with the output $B_6$ defined in (8).

$$ B_6 = bn(D_1) $$ (8)

This is followed by the dropout function $drop$ which randomly sets input units to 0 with a frequency of rate at each step during training time, which helps prevent overfitting [48]. The output $Dr_1$ is defined in (9).

$$ Dr_1 = drop(B_6) $$ (9)

The final dense layer (the output $Output$) is a deeply connected neural network layer with ReLU activation that generates the final predictions. This is defined in (10).

$$ Output = relu(Dr_1 \cdot guf) $$ (10)

After obtaining $O_1$, the $Output$ is defined in (11).

$$ Output = relu\{drop(bn(relu(fl(O_1) \cdot guf)))\} \cdot guf $$ (11)

The $Output$ layer has 11 units for hand configuration classification and 4 units for MCP joint angle estimation. For training, forward propagation and backward propagation train an optimized model. For inference, forward propagation generates the output prediction. During forward propagation, the weights and filters are randomly initialized and are used as model parameters. During backward propagation, the model parameters are updated over successive epochs to reduce the loss and improve the accuracy.

III. EXPERIMENTAL IMPLEMENTATION

This section describes the data acquisition, hardware preparation, software used, and experimental testing protocol.

A. Human Subjects and IRB Approval

The study was approved by the institutional research ethics committee at the Worcester Polytechnic Institute (No. IRB-21-0452), and written informed consent was given by the subjects prior to all sessions. Ultrasound data was captured from the right forearm of 6 subjects (1 female, 5 males; Age: 22.83 ± 3.14 years; Height: 173.55 ± 11.02 cm; diameter of the forearm around the point where the probe is placed: 20.43 ± 2.74 cm). Table I lists the age, sex, height, and forearm diameter at the ultrasound probe location for the 6 subjects enrolled in the study.
### TABLE I
SUBJECT INFORMATION

| Age (yr) | Sex (M/F) | TS 1 | TS 2 | TS 3 | TS 4 | TS 5 | TS 6 |
|---------|-----------|------|------|------|------|------|------|
|         |           | 25   | 20   | 24   | 23   | 24   | 21   |
| Height (cm) |          | 157.5| 170.2| 172.7| 180.3| 167.6| 193.0|
| Forearm Dia. (cm) |     | 22.9 | 18.4 | 18.4 | 19.1 | 18.4 | 25.4 |

Fig. 3. (a) Location of markers and vectors on the hand, and (b) the definition for MCP angle estimation.

B. Instrumentation

Velcro straps on both sides of the custom-designed 3D-printed probe casing were used to strap the ultrasound probe on the forearm. The subject’s arm rests on a rest table which was further secured with another external Velcro extender to the table after verification of the ultrasound images. Fig. 5 shows a subject’s right arm strapped to the rest table.

1) Hardware Preparation: Three-dimensional positional data of specified points on their index, middle, ring, and pinky fingers (Fig. 3(a)) was acquired along with the ultrasound data, with the ultrasound probe mounted in a transverse position on the forearm (Fig. 5). This data was acquired at three different speeds of hand movement. The experimental setup uses a Vantage 128 Verasonics ultrasound data acquisition system (Verasonics, WA, USA) and a Vicon Nexus motion capture system (Vicon Motion Systems Ltd., U.K.). The average estimated error linked to the MCP joint angle ground-truth estimation using the Vicon Nexus motion capture system has been reported to be less than the acceptable 5° except for the pinky MCP joint angle standard error of measurement which is less than 6° [49].

An L12/5 50 mm linear array ultrasound probe was used with the Verasonics Vantage 128/128 research ultrasound system. A MATLAB script was used to set sequence objects for the Verasonics data acquisition hardware to display and record ultrasound images. A trigger output signal was sent to an Arduino board from the Verasonics system as a synchronizing signal. The ultrasound data frame acquisition rate was set at 25 Hz. For 56 seconds of data acquisition per session, 1400 frames were recorded. Each data frame was 636 x 256 pixels, corresponding to a 44.52 mm depth of the ultrasound image and a 50 mm width of the imaging region. A 10-camera Vicon Vantage motion capture system with Lock+ 64-channel ADC
(Analog to Digital Converter) was used to obtain the three-dimension (3-D) positional information for different motion capture markers attached to the finger. Because of a sample rate mismatch between the trigger from the Verasonics ultrasound system and the Vicon motion capture system, an Arduino Uno was used as a bridge to record trigger signals utilizing Arduino’s interrupt service routine and generate a pulse that could be read by the Vicon analog to digital converter.

2) Audio Signal Generation: Three audio waveforms were designed for simulating three different speeds for hand movements using Audacity, an open-source digital audio workstation. 0.1-second Sawtooth waves at 440 Hz and 392 Hz were used to alert the subjects to alternate between the rest and motion hand states. The frequencies of the rest/motion switching were set as 2 Hz for fast, 1 Hz for medium, and 0.5 Hz for slow speeds. These speeds were set so that in addition to the standard 1 Hz signal, data at speeds more and less than 1 Hz by a factor arbitrarily chosen as 2 could be quantitatively acquired.

3) Computation Hardware and Software: NVIDIA GeForce RTX 2070 SUPER was used to develop and debug the deep learning models. AMD Ryzen 7 2700X Eight-Core Processor was used to develop and debug the classical machine learning models. The system had 31.91 GB of available RAM. The code was executed in Python 3.7. TensorFlow Keras API was used for deep learning models like our custom CNN and MLP [50]. Scikit-learn library was used for classical machine learning based classification models [51]. The scikit-learn function that was used to implement SVC used C-Support Vector Classification (C-SVC) implementation based on libsvm [51]. The multiclass support is handled according to a one-vs-one scheme, and the 11 classes are divided into 55 binary classification datasets. Each of these 55 binary classification models predicts one class and the model with the most predictions is predicted by the one-vs-one scheme.

C. Experimental Testing Protocol

11 hand configurations relevant to the activities of daily living (ADLs) were chosen for this project [52]. The hand configurations are shown in Fig. 4. The hand configurations: C1 (IndFlex): Index finger flexion; C2 (MidFlex): Middle finger flexion; C3 (RinFlex): Ring finger flexion; C4 (PinFlex): Pinky finger flexion; C5 (IndPinch): Index finger in contact with the thumb; C6 (IndMidPinch): Index & Middle fingers in contact with the thumb; C7 (IndMidRinPinch): Index, Middle & Ring fingers in contact with the thumb; C8 (AllPinch): All fingertips touching; C9 (MidRinPinch): Middle & Ring fingers in contact with the thumb; C10 (Fist); C11 (Hook): Movement is restrained to just the interphalangeal joint movement with the MCP joint angle remaining constant; Open: All fingers extended. The Open (all fingers extended) was considered as the rest state for all the 11 hand configurations and the subjects alternated between the open hand and selected the hand configuration. The subject was seated, and 16 motion capture markers were attached to the ends of the metacarpal and proximal phalanx for each of the index, middle, ring, and pinky fingers on their right hand as seen in Fig. 3 (a).

This marker configuration allowed us to calculate the MCP joint angle while ensuring free movement of the hand for the desired hand movements. The MCP joint angle was measured by taking the inverse cosine of the two vectors formed by each metacarpal-proximal phalanx pair. Ultrasound gel was applied to the subject’s forearm and the imaging surface on the ultrasound probe. The ultrasound probe was then strapped to the
subject’s right forearm. Then, the subject’s forearm was fastened to a rest table. The subject then performed a trial run for the 11 hand configurations. Auditory beeps were used to indicate the desired rate of hand opening and closing. There was a rest of 1 minute between each data acquisition session and 2 minutes between each speed transition. 56 seconds of data were acquired for each data acquisition session. Per subject, there were 33 data acquisition sessions, for 11 different hand movements at 3 different speeds each.

IV. DATA ANALYSIS

Post-beamformed envelop detected ultrasound data were preprocessed before training the models by normalizing, log-compressing, and reshaping to highlight relevant muscle features. For the 6 subjects, data for 11 hand configurations, each at 3 speeds was acquired, leading to a total of 198 sessions of data acquisition. 56 seconds of data were acquired per subject, and hand configuration, and speed, totaling 11088 seconds (6 * 11 * 3 * 56 seconds) of data used for analysis.

A. Data and Model Parameters

For the classification of hand configurations, small portions of each hand configuration for a specific subject and speed were used to validate the classification performance. For estimating the finger angles during the hand configuration motions, full data sizes for a specific subject, speed, and hand configuration were used to validate the regression performance.

1) Classification Parameters: For each subject for each hand configuration and speed, 8 seconds of data were considered. These files were then split with a train-test split of 70% to get independent testing and training sets. 5.6 seconds of training and 2.4 seconds of test data per subject, hand configuration, and speed were used for generating the classification results. This data was then combined across the hand configurations for each subject and speed (88 seconds of data per subject and speed) leading to 61.6 seconds of training and 26.4 seconds of test data per subject and speed.

2) Regression Parameters: The CNNs were trained for each hand configuration and speed on 56 seconds of data for every subject. Adam optimizer function was used as the gradient descent method. A learning rate of 1e-3 was set for the Adam optimizer [53]. The learning rate decay was set to 1e-3/200. Mean Absolute Error was used as the loss function. It computes the mean of the absolute difference between true and predicted values. Each data file was split with a train-test split of 70%. While training, the validation split was defined as 0.1. The number of epochs was set to 50.

B. Angle Estimation for Ground Truth

Motion capture markers were placed on the subject’s hand at the base and the head of the index, middle, ring, and pinky proximal phalanges and the metacarpals. The marker locations on a subject’s right hand can be seen in Fig. 3(a). Data was not acquired from the thumb because there is an extrinsic flexor for the interphalangeal joint in the forearm, but the more relevant motions of the thumb come from the carpometacarpal (CMC) joint and MCP joint. The motions of the CMC and MCP joints for the thumb depend on the intrinsic muscles that are not imaged. Hence, including the thumb angle, data will be the subject of future research.

For the index metacarpal, the points can be defined as $M1$ and $M2$, with the vector $\overrightarrow{M1M2}$, shown in red in Fig. 3(a). For the index proximal phalanx, the points can be defined as $P1$ and $P2$, with the vector $\overrightarrow{P1P2}$, shown in blue in Fig. 3(a). The angle between the $\overrightarrow{M1M2}$ and $\overrightarrow{P1P2}$ is 180° measured counterclockwise. As seen in Fig. 3(b), when the proximal phalanx flexes by 30° from its initial state ($\overrightarrow{P1P2}$), the angle between the $\overrightarrow{M1M2}$ and $\overrightarrow{P1P2′}$ is 210°. Similarly, when the proximal phalanx flexes by 30° and 60° from its initial state ($\overrightarrow{P1P2′}$), the angles between the $\overrightarrow{M1M2} − \overrightarrow{P1P2′}$, and $\overrightarrow{M1M2} − \overrightarrow{P1P2′′}$ pairs are 240° and 270°, respectively. The MCP joint angle was measured by taking the inverse cosine of the two vectors formed by each metacarpal-proximal phalanx pair.

C. Quantification Metrics

Accuracy percentage $Acc$ was used for evaluating the classification performance. It is defined in (12).

$$Acc = \frac{TP + TN}{N} \times 100$$ (12)

where, $TP$ is the number of True Positives, $TN$ is the number of True Negatives, and $N$ is the Total Sample Size. $TP$ is an outcome where the model correctly predicts the positive class. $TN$ is an outcome where the model correctly predicts the negative class. The classification error percentage is obtained by subtracting $Acc$ from 100.

The Root Mean Squared Error ($RMSE$) values were used to quantify the regression results, defined in (13).

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2}$$ (13)

where, $N$ is the Total Sample Size, $i$ is the integer value ranging from 1 to the total number of samples, $y_i$ is the test data value for the sample $i$, and $\hat{y}_i$ is the value predicted by the deep learning algorithm at the sample $i$.

For quantifying both $Acc$ and $RMSE$, Arithmetic Mean ($\mu$) and Standard Deviation ($\sigma$) were used. Arithmetic mean is defined in (14).

$$\mu = \frac{1}{N} \sum_{i=1}^{N} a_i$$ (14)

where, $\mu$ is the arithmetic mean, $N$ is the number of values, and $a_i$ is the value from a data set of values, which for this paper can be $Acc$ values or $RMSE$ values. Standard deviation ($\sigma$) is defined in (15).

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (x_i - \mu)^2}$$ (15)

where, $\sigma$ is the standard deviation, $N$ is the number of values, and $x_i$ is the value from a data set of values, and $\mu$ is the mean of the data calculated in (14).
TABLE II

| Metric | SVC-Lin | SVC-RBF | SVC-Ply | MLP  | CNN  |
|--------|---------|---------|---------|------|------|
| Acc    | 80.6    | 80.2    | 81.5    | 72.6 | 82.7 |
| σ      | 10.7    | 10.8    | 10.9    | 9.3  | 9.7  |

V. RESULTS

Here, the hand configuration classification results using SVC, MLP, and the proposed CNN, and MCP joint angle estimation results using the proposed CNN are described. A low latency MCP joint angle prediction and hand configuration classification pipeline aimed at real-time control of interfaces is also described. The results based on data obtained from subject number 1 are shown in Fig. 7.

A. Hand Configuration Classification

To prove that the classifier can classify the 11 classes described in Fig. 4 based on the ultrasound images from the forearm, different machine learning models were trained for each subject for all the speeds. Fig. 7(a) shows the confusion matrix using the CNN for slow-speed data from Subject 1.

1) Hand Configuration Classification Over Different Machine Learning Algorithms: The highest classification accuracy was found for the CNN. Among the different SVC kernels, the polynomial kernel performed the best classification accuracy versus the linear and the radial basis function kernels whose performance was approximately similar. MLP had the least classification accuracy. The results are summarised in Table II.

2) Hand Configuration Classification Over Different Speeds: For the five different models described in Section V-A1, it was observed that the accuracy percentage was the highest for the Slow speed (0.5 Hz) and the lowest for the Fast speed (2 Hz). The consistency of this trend across the different algorithms is indicative of the gradation between the different speeds, with the fast speed images having more variability across subjects than the medium speed images, and the medium speed images having more variability than the slow speed images, which have less variability per subject across subjects among all the speeds, thereby leading to better performance. Like the previous section, the CNN performed the best with the slow speed accuracy of 90.5 ± 7.8%, medium speed accuracy of 82.0 ± 7.5%, and fast speed accuracy of 75.7 ± 7.3%

3) Hand Configuration Classification Over Down-Sampled Image Sizes: For images down-sampled by a factor of 4, the highest classification accuracy was found for the SVC with the Polynomial Kernel. This was followed by the SVC with the linear and radial basis function kernels. Among the MLP and CNN, MLP delivered a better performance. The results are summarised in Table III.

Like 5A.2, it was observed that the accuracy percentage was the highest for the Slow speed (0.5 Hz) and the lowest for the Fast speed (2 Hz). The SVC with polynomial kernel performed the best with the fast speed accuracy of 89.3 ±
Table III

| Metric | SVC-Lin | SVC-RBF | SVC-Ply | MLP | CNN |
|--------|---------|---------|---------|-----|-----|
| Acc    | 80.6    | 80.1    | 81.8    | 79.6| 79.3|
| σ      | 4.0     | 3.8     | 4.1     | 4.1 | 3.7 |

Fig. 8. Accuracy percentages for different speeds for the five different models for full images.

4.1%, medium speed accuracy of 81.0 ± 11.4%, and fast speed accuracy of 75.0 ± 11.0%. This was followed by SVC with the linear kernel at 88.8 ± 3.8% for slow speed, 80.0 ± 11.1% for medium speed, and 73.0 ± 9.1% for fast speed. SVC with RBF kernel had the lowest performance among the three SVC kernels at 88.5 ± 3.5% for slow speed, 79.7 ± 11.5% for medium speed, and 72.0 ± 8.3% for fast speed. and the radial basis function kernel (89.3 ± 3.6% for slow speed, 79.5 ± 11.0% for medium speed, 71.8 ± 8.0% for fast speed) whose performance was approximately similar. The MLP and CNN performance were comparable, at 86.2 ± 6.2% for slow speed, 79.0 ± 12.8% for medium speed, and 73.5 ± 7.1% for fast speed. The overall RMSE value averaged over all the hand configurations, three different speeds, and all the subjects was found to be 7.35 ± 1.3°. These results show that by using the CNN architecture described in the paper, MCP joint angles can be predicted for different hand configurations based on ultrasound images from the forearm.

4) Classification Performance Comparison Over Full and Down-Sampled Image Sizes: Among all the different algorithms, the highest classification percentage was obtained for CNN with full image data size at 82.7 ± 4.1%. There was only a slight difference in accuracy percentage for SVC with linear, RBF, and polynomial kernels. For SVC with polynomial kernel and MLP, the results for down-sampled data were better than the full image data. But, these results were still worse than the CNN with full image sizes. Because the CNN results on full image sizes outperformed all other configurations, the Angle Estimation analysis was performed on the full image sizes using CNN described in Section II-C. The comparison between full image sizes and down-sampled images is shown in Fig. 10.

B. Angle Estimation

Using the CNN described in Section II-C, CNN models were trained for each hand configuration, for each speed and subject (total of 6 * 11 * 3 = 198 models). Utilizing the CNN, a good degree of visual correspondence between the test data and the model predictions on individual trained models was obtained. This is shown for four second test data in Fig. 7(b) through Fig. 7(h). RMSE values were obtained from each of the hand configurations for each speed, for every subject. For the results, the full range of hand motion ranges from 0° - 100°. The range is set from 0°-20° to highlight the averaged results and their gradation over the subjects and hand configurations. For models with successful convergence, the overall RMSE value averaged over all the hand configurations, three different speeds, and all the subjects was found to be 7.35 ± 1.3°. These results show that by using the CNN architecture described in the paper, MCP joint angles can be predicted for different hand configurations based on ultrasound images from the forearm.

1) Estimation Results for Different Hand Configurations:

To ensure the viability of the proposed approach for different hand configurations, RMSE values were calculated for the predictions averaged over all the subjects for all the slow, medium, and fast speeds for each hand configuration. Table IV shows the plot of RMSE results for different hand
TABLE IV
AVERAGED RMSE FOR DIFFERENT HAND CONFIGURATIONS

| Metric | C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | C9 | C10 | C11 |
|--------|----|----|----|----|----|----|----|----|----|-----|-----|
| RMSE   | 7.7| 8.0| 8.8| 9.7| 6.1| 6.7| 6.7| 9.1| 11.4| 4.5 |
| σ      | 2.1| 2.8| 2.5| 4.7| 1.4| 1.5| 2.4| 1.5| 3.7| 4.9 | 1.3 |

TABLE V
AVERAGED RMSE FOR DIFFERENT SUBJECTS

| Metric | S1 | S2 | S3 | S4 | S5 | S6 |
|--------|----|----|----|----|----|----|
| RMSE   | 6.1| 8.6| 9.2| 5.6| 7.9| 6.7|
| σ      | 2.3| 3.9| 3.3| 1.7| 3.1| 1.7|

Fig. 11. RMSE results for subjects at different speeds averaged over the different configurations.

3.2) Repeatability Over Subjects: To ensure that the results are repeatable over the subjects, RMSE values were calculated for the predictions averaged over all the slow, medium, and fast speeds, and the hand configuration for each subject. Table I lists the age, sex, height, and forearm diameter at the ultrasound probe location for the 6 subjects enrolled in the study. Table V shows the plot of RMSE results for different subjects. The lowest average RMSE value was obtained from subject 4 and the largest average RMSE value was obtained from subject 2. The lowest standard deviation was obtained from subject 4 and the largest average standard deviation was obtained from subject 2.

3.3) Estimation Results Over Different Speeds: Fig. 11 shows the RMSE results for subjects at different speeds averaged over the hand configurations. For subjects 2, 5, and 6, there is a trend of increasing RMSE as the hand movement speed increases from Slow (0.5 Hz) to Fast (2 Hz). For subjects 1, 3, and 4, the RMSE is higher for Slow speed (0.5 Hz) than for Medium speed (1 Hz). For all subjects except Subject 4, the RMSE is higher for Fast speed (2 Hz) than for Slow Speed (0.5 Hz). On averaging the speed over all subjects, it is observed that the average RMSE is the least for Slow speed at 6.67° ± 2.58°. The average RMSE for Medium speed is 7.25° ± 3.32°. The average RMSE is the highest for Fast speed at 8.36° ± 4.11°. The standard deviation follows a similar trend and increases from Slow speed to Fast speed.

C. Combined Prediction Pipeline

For the combined prediction pipeline SVC-Lin (SVC with a Linear Kernel) models trained to classify between 11 different hand configurations for a single speed and subject and 11 CNN models trained for prediction of MCP joint angles were used. In the combined loop, first, the hand configuration state prediction is done using the saved SVC model. Then, based on the SVC prediction, the CNN model is chosen from a dictionary of saved models. For each frame of data, the SVC prediction takes 0.1 - 0.15 seconds. CNN prediction takes 0.004 - 0.007 seconds. The total time ranges from 0.11 - 0.16 seconds for angle and state prediction for one frame of ultrasound information, leading to 6.25 - 9.1 Hz of data processing capability using the combined prediction pipeline. The pipeline is described in Fig. 1.

VI. DISCUSSION AND FUTURE WORK

The experiments and analysis show that both hand configuration classification and finger angle estimation can be obtained using one frame of ultrasound data using different classification and regression models. While ultrasound hand configuration classification is well researched, in this work, it is shown that with the experimental setup described in Section III, 82.7 ± 9.7% classification accuracy with the CNN proposed in the paper and 81.5 ± 10.9% classification accuracy using an SVC with a polynomial kernel was obtained. Due to computational limitations, classifiers that were based on ultrasound datasets larger than the ones used in the paper for classification were not used. It would be worthwhile to check out the classifier performance for bigger datasets and to see how deep learning based classification performs on ultrasound data in comparison with the traditional SVC. For real-time control applications, it would be necessary to evaluate the classification in an on-the-fly training and evaluation paradigm. Specifically, similar to the 4-state discretized classifier demonstrated in [33]. The data was acquired in a way that motion artifacts were minimized, and the subjects were supervised in a way that they would be consistent with their movements for a particular hand configuration movement set which led to accuracy percentages over 80% using both CNN and SVC with different kernels. It would be interesting to see the results in a system that is deployed for real-time use.

Several methods could be used for estimating hand movements such as a motion capture system [54], marker-less deep learning based hand movement detection sensors [55], wearable gloves [56], etc. The motion capture system was used because of its ease of synchronization with the ultrasound data acquisition system and its wide use in literature for estimating finger joint angles [57]. During the data acquisition, because
of erratic hand movements and motion capture marker occlusion, there were issues with getting proper ground truth for some data acquisition sessions. 5.6% of the total files couldn’t be used because of this issue. An instrumented glove like the Cyberglove can be used as a future improvement to avoid issues related to markers not being visible to the camera in the motion capture system [56]. Additionally, the proposed CNN model did not converge for 26.2% of the files. Because of these issues, the MCP joint angle estimation results were obtained for 68.2% of the total data files that were initially acquired. The model convergence issue for MCP joint angle estimation can be attributed to using one CNN model for all the subsets of the data over which the analysis was performed. Aggressive hyperparameter and model parameter optimization, along with increasing the model complexity for each of the subsets for which the model did not converge can help with obtaining convergence for more data. However, the updated parameters and hyperparameters might not work for the subsets for which the current model converges and thus not be generalizable over a more significant portion of the data, unlike the current model. There is ongoing research in the deep learning community on understanding generalization and its trade-off with model complexity [58]. Because of the good quality ultrasound data corresponding to the hand configurations, 100% of the ultrasound data could be used for the classification. There were no issues with model convergence and the data labeling for the classification problem unlike the issues encountered with the MCP joint angle estimation problem.

As can be seen in Table IV, configurations C4 and C10 have the biggest RMSE values. This is because both these states (PinFlex and Fist) have significant pinky finger movement. Pinky finger movements were not consistent because it’s challenging to have repeatable movements with the pinky finger. It has been previously shown that the standard error of measurement among the MCP joint for the index, middle, ring, and pinky fingers is highest for the pinky finger [49]. This is reflected in our results since the hand movements primarily comprising the pinky finger MCP joint movement have the highest RMSE. For configuration C11 (Hook), the RMSE value was the minimum because the MCP angle values are smaller than other hand configurations. It can be seen that for the hand configurations involving lesser movements such as C5 (IndPinch) and C11 (Hook), the RMSE is lesser than the hand configurations involving more movement such as C4 (PinFlex) and C10 (Fist). The synergistic movements of multiple fingers in the hands and the corresponding effect on the RMSE needs to be investigated for a model that can be used to predict MCP joint angles regardless of the hand configuration. This would be a step in the direction of generalizability across hand configurations.

For the combined prediction pipeline, most of the time is taken by SVC prediction since the current implementation is not optimized on GPU and it uses the system CPU which is much slower. Exploring more GPU-supported deep neural network based classification algorithms can potentially improve the rate of simultaneous prediction from the current 6.25 - 9.1 Hz. In the current paper, different deep learning models were trained for each angle estimation data. It would be worthwhile to explore deep learning algorithms which can generalize the MCP joint angle prediction for different hand configurations as well as for different subjects. In the direction of portability, wearability, and technology translation, a system pipeline like the one demonstrated in [33], along with the low-latency deep learning based angle estimation system proposed in [59] and in this paper can be adapted for smaller and wearable ultrasound systems like the one demonstrated in [60] and other related works. The authors are excited about a future low-power (as the ones described in [61]), small form factor (like an Apple Watch), wearable (wearable hand gesture recognition interfaces surveyed in [62]), and portable device that utilizes forearm ultrasound data to estimate hand movements. This can be beneficial as a technology for human-machine interfacing and for deducing signals from the underlying musculature to develop rehabilitation tools [63].

VII. CONCLUSION

In this paper, the results for MCP joint angle estimation and hand state prediction based on forearm ultrasound are discussed. The proposed Convolutional Neural Network was able to get an average RMSE of 7.35° over 6 subjects, 3 speeds, and 11 hand configurations. Despite memory limitations, promising classification results were obtained on a subset of the entire dataset, with the proposed CNN leading to an average classification accuracy of 82.7%, and SVC with linear, RBF, and polynomial kernels leading to an average classification accuracy over 80%. A combined pipeline for angle and hand state estimation has also been described, which for the current setup can generate predictions as fast as 6.25 - 9.1 Hz per frame of ultrasound data. These results are encouraging, and they show the capability of ultrasound in understanding muscle movements. This work can inspire research in the promising domain of utilizing forearm ultrasound for predicting both continuous and discrete hand movements which can be useful for intuitive and adaptable control of physical robots and non-physical digital and AR/VR interfaces.

ACKNOWLEDGMENT

The authors would like to thank the members of the Medical Robotics Department at WPI for their input and help with the successful completion of the project. The authors are grateful to Tess Meier for her constructive criticism of the manuscript.

REFERENCES

[1] A. Tarasenko, M. Oganesyan, D. Ivaskyvych, S. Tukaiev, D. Toleukhanov, and N. Vysokov, “Artificial intelligence, brains, and beyond: Imperial college London neurotechnology symposium, 2020,” Bioelectricity, vol. 2, no. 3, pp. 310–313, 2020.

[2] T. Vuletic, A. Duffy, L. Hay, C. McTeague, G. Campbell, and M. Grealy, “Systematic literature review of hand gestures used in human computer interaction interfaces,” Int. J. Human–Comput. Stud., vol. 129, pp. 74–94, Sep. 2019.

[3] Y. H. Jung, J.-H. Kim, and J. A. Rogers, “Skin-integrated vibrohaptic interfaces for virtual and augmented reality,” Adv. Funct. Mater., vol. 31, no. 39, 2021, Art. no. 2008805.

[4] M. Zhu et al., “Haptic-feedback smart glove as a creative human-machine interface (HMI) for virtual/aligned reality applications,” Sci. Adv., vol. 6, no. 19, 2020, Art. no. eaaz8693.
[53] D. P. Kingma and I. Ba, “Adam: A method for stochastic optimization,” 2014, arXiv:1412.6980.

[54] M. Yahya, J. A. Shah, K. A. Kadir, Z. M. Yusof, S. Khan, and A. Warsi, “Motion capture sensing techniques used in human upper limb motion: A review,” Sensor Rev., vol. 39, no. 4, pp. 504–511, 2019.

[55] F. Zhang, V. Bazarevsky, A. Vakunov, A. Tkachenka, G. Sung, C.-L. Chang, and M. Grundmann, “Mediapipe hands: On-device real-time hand tracking,” 2020, arXiv:2006.10214.

[56] G. D. Kessler, L. F. Hodges, and N. Walker, “Evaluation of the cyber-glove as a whole-hand input device,” ACM Trans. Comput.–Human Interact., vol. 2, no. 4, pp. 263–283, 1995.

[57] J. Ngeo, T. Tamei, T. Shibata, M. F. Orlando, L. Behera, A. Saxena, and A. Dutta, “Control of an optimal finger exoskeleton based on continuous joint angle estimation from EMG signals,” in Proc. 35th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC), 2013, pp. 338–341.

[58] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, “Understanding deep learning (still) requires rethinking generalization,” Commun. ACM, vol. 64, no. 3, pp. 107–115, 2021.

[59] K. Bimbraw, C. J. Nycz, M. J. Schueler, Z. Zhang, and H. K. Zhang, “Prediction of metacarpophalangeal joint angles and classification of hand configurations based on ultrasound imaging of the forearm,” in Proc. IEEE Int. Conf. Robot. Autom. (ICRA), 2022, pp. 91–97.

[60] C. Wang et al., “Bioadhesive ultrasound for long-term continuous imaging of diverse organs,” Science, vol. 377, no. 6605, pp. 517–523, 2022.

[61] T.-G. La and L. H. Le, “Flexible and wearable ultrasound device for medical applications: A review on materials, structural designs, and current challenges,” Adv. Mater. Technol., vol. 7, no. 3, 2022, Art. no. 2100798.

[62] S. Jiang, P. Kang, X. Song, B. P. L. Lo, and P. B. Shull, “Emerging wearable interfaces and algorithms for hand gesture recognition: A survey,” IEEE Rev. Biomed. Eng., vol. 15, pp. 85–102, 2021.

[63] S. Patwardhan, J. Schofield, W. M. Joiner, and S. Sikdar, “Sonomyography shows feasibility as a tool to quantify joint movement at the muscle level,” in Proc. IEEE Int. Conf. Rehabil. Robot. (ICORR), 2022, pp. 1–5.

Keshav Bimbraw (Graduate Student Member, IEEE) received the B.E. degree in mechatronics engineering from Thapar University, India, and the M.S. degree in computer software and media applications from Georgia Institute of Technology, Atlanta, GA, USA. He is currently pursuing the Ph.D. degree in robotics engineering with Worcester Polytechnic Institute, Worcester, MA, USA.

Christopher J. Nycz received the B.S. degree in mechanical engineering from Clarkson University, Potsdam, NY, USA, and the M.S. and Ph.D. degrees in robotics engineering from Worcester Polytechnic Institute, Worcester, MA, USA, where he is a Research Scientist working with PracticePoint.

Matthew Schueler received the B.S. degree in robotics engineering/computer science and the M.S. degree in robotics engineering from Worcester Polytechnic Institute, Worcester, MA, USA. He works as a Firmware Engineer with Amazon Robotics, Westborough, MA, USA.

Ziming Zhang received the Ph.D. degree from Oxford Brookes University, U.K., in 2013. He is an Assistant Professor and the PI of the Vision, Intelligence, and System Laboratory, Worcester Polytechnic Institute (WPI), Worcester, MA, USA. Before joining WPI, he was a Research Scientist with Mitsubishi Electric Research Laboratories. Prior to that, he was a Research Assistant Professor with Boston University.

Haichong K. Zhang (Member, IEEE) received the B.S. and M.S. degrees in health sciences from Kyoto University, Kyoto, Japan, and the M.S. and Ph.D. degrees in computer science from Johns Hopkins University, Baltimore, MD, USA. He is an Assistant Professor and the PI of the Frontier Ultrasound Imaging and Robotic Instrumentation Laboratory (Medical FUSION Laboratory), Worcester Polytechnic Institute, Worcester, MA, USA.