Development of a virtual hand model moving synchronously to a computer vision tracked hand based on the classification of the tracked hand movements

Aaron Neu Gil

***Abstract*—Modern prosthetic hands can significantly improve the quality of life after losing a hand. Nevertheless, the rejection rate is unexpectedly high due to non-intuitive controls and a delayed rehabilitation start. Therefore, this work presents a Mixed Reality application for the HoloLens 2 to provide a training tool to control a virtual hand model based on electromyography signals. The overall project is structured in two parts, comprising the proposal and implementation of a flexible electromyography wristband interpreting the muscle activity based on a neural network and the development of an application providing a virtual hand model controlled by the wristband. The presented work only focuses on the design and implementation of the application, including the establishment of a dataset for visual gesture recognition, the training of a feed-forward neural network used to classify gestures visually, and the design of a virtual prosthetic hand attached to the residual limb. Furthermore, the integration of the electromyography wristband is described, including the collection of individual electromyography data based on which the neural network is trained. The feed-forward neural network for visual gesture recognition was trained on 1585 samples, achieving an accuracy of 99% to classify four gestures: pinch, spread, fist and thumb up. By cross-validating the model during runtime by performing each gesture ten times in a randomised order and manually defining the actual label, the accuracy dropped to 59%. However, the average delay of 423 ms between the classified gestures was not accounted for. Future work should focus on improving the current system, including a comparison of different classification models and features to improve visual gesture recognition. Additionally, the application can be improved by modifying the user interface and integrating a training process to further help patients get accustomed to prosthetic hands.**

***Index Terms*—Human-Computer Interaction, Machine Learning, Gesture Recognition, Mixed Reality**

I. Introduction

THE human hand can perform various tasks and gestures, an essential aspect of daily living. In 2005, 1.5 million upper extremity amputations were caused by multiple factors such as trauma, cancer, disease progression, or congenital malfunctions [1], [2]. The loss of a hand can affect autonomy and social acceptability. Therefore, prosthetic hands can improve functionality, aesthetics, and social interaction. Modern prosthetic hands allow complex gestures, increasing life standards by enabling the patient to do more daily tasks [3]. Despite the mechanical effort to build a perfect prosthetic hand, the rejection rate is significantly high, which is partly due to the non-intuitive control and unsuccessful therapy based on lack of user training and a late start of rehabilitation therapy [3], [4].

A. Neu Gil is a Student in Medical Technologies, MCI, Innsbruck, Austria, e-mail: na5626@mci4me.at.

Research has been conducted to improve the intuitive control of prosthetic hands using surface electrodes, interpreting electrical signals of the residual muscles [3]. Electromyography is challenging due to individual anatomic characteristics, lower activation signals in residual muscles, crosstalk between other proximal muscles, and the complexity of contraction combinations [1], [3], [5]. Deep Learning has gained importance in recent years and is a helpful tool for dealing with the different challenges of interpreting electromyography signals [5]. Another challenge is unsuccessful rehabilitation therapy, partly due to the waiting time for a prosthetic fit and the lack of training and patient motivation [4], [6]. Virtual, artificial, and mixed reality can be considered to improve training, diagnosis, and treatment in clinical decisions [6]. Providing a virtual prosthesis attached to the residual limb controlled via electromyography, rehabilitation can start immediately without additional costs and increase the motivation of the individual subject for repetitive tasks [4], [6].

This paper is published in addition to a Master's thesis, presenting briefly the developed mixed reality application containing a virtual prosthetic hand attached to the residual limb controlled through an EMG wristband. The application provides a feature to create an individualised EMG dataset, which may increase the performance of the gesture recognition model. This is done by instructing the user to perform the gestures in random order while the electromyography signals are recorded through the EMG wristband. The presented model is based on the assumption that the subject only has one amputated hand to classify the gestures of the healthy hand visually and provide a label for the EMG signals. The assumption is that signal quality increases by simultaneous muscle activation on both sides during a gesture in addition to having visual feedback. The visual gesture recognition is based on skeleton data recorded using existing software. However, the application also provides the user with visual feedback on the performed gesture to act as a training tool. This allows the patient to start rehabilitation without additional costs immediately.

II. State of the art and Fundamentals

This Chapter provides an overview of the basics and current advancements in upper limb prostheses, including amputation, prosthetic control and the limitations of current technologies. In addition, potential solutions and upcoming technologies to improve intuitive prosthetic hand control are presented.

A. Upper Limb Amputation

Upper extremity amputations occur due to different circumstances, including trauma, cancer, disease progression or congenital malformations [2]. The estimation of the global number of amputations is challenging to estimate [7]. Therefore, Ziegler-Graham et al. [8] estimated the number of amputations in the United States to 1.5 million in 2005 by calculating the number of persons discharged from the hospital after amputation. Different literature ([2], [7], [8]) assumed an increasing number of amputations based on the diabetes epidemic, and higher life expectations, which increases the risk of amputation. As of the 1.5 million persons with limb loss, approximately 25% suffered from upper limb loss [7]. Different surgical procedures to amputate the upper limb are named depending on the level at which they are performed. The primary consideration on which section the upper limb is amputated is the preservation of joints and the residual limb's length due to the increased capability to interact with the environment [2]. The most commonly used surgical procedure for upper limb amputations is transcarpal, accounting for 61% of the cases [9], [10]. However, despite the location of the amputation, target muscle reinnervation is applied to all upper extremity amputation patients who could use a myoelectric prosthesis [5]. This technique redirects nerves controlling a specific muscle to innervate accessory muscles. The aim is to improve surface electromyography signal (sEMG) recording and allow intuitive prosthesis control [5]. The procedure is commonly done during amputation but is also possible during revision surgeries [2].

B. Prosthetic and Orthotic Market

Losing an upper limb can severely impact one's daily life and social acceptance. However, modern prosthetics can improve functionality, aesthetics, and social interaction [3]. Some significant competitors providing upper limb prosthetics are Blatchford Inc, Össur, Ottobock, Bauerfeind AG, WillowWood Global LLC, and Fillaur LLC [11]. Upper limb prosthetics are classified into five categories: prosthetic wrists, prosthetic elbow, prosthetic shoulder, prosthetic arm, and terminal devices [12]. The market of upper limb prosthetics in the United States was approximately one billion USD in 2018 and is expected to increase to 2.3 billion USD in 2025 [12]. A modern prosthetic hand costs 15 000 – 100 000 USD with additional customisation costs [3]. These high costs result primarily from imitating the human hand. The human hand allow 21 degrees of freedom and is composed of 29 skeletal muscles, 27 bones and 15 joints and is, therefore, able to perform a wide range of different tasks and gestures [1]. The muscle fibres performing the movement are activated through electrical signals, which initiate the contraction [13]. However, precise muscle control is needed. The human body achieves this control by feedback through a sensor array measuring the muscle length, pressure and temperature [3].

C. Prosthetic Hands

Much progress has been made in prosthetic hands. The development goes from mechanical hooks to modern prosthetic hands, which offer more degrees of freedom and therefore, the user can solve more complex tasks [3]. Nowadays, there are different types of prosthetic hands, which Pylatiuk and Döderlein [14] divided into three different categories. One category is cosmetic prostheses, which provide esthetical and psychological support and have no specific functionality. The second category is kinematic prostheses, which offer additional functional support and can be controlled through mechanical signals of the body. The last category is myoelectric prostheses, which also provide esthetical, psychological, and functional support. Myoelectric prostheses are controlled through the body's electrical signals and are powered by lithium-ion batteries, providing a usage period of approximately twelve hours depending on the prosthesis [5]. A modern myoelectric prosthetic hand can flex and extend a finger individually and rotate the thump and wrist. Nevertheless, only 30% of all existing devices on the market have more than 5 degrees of freedom, and from those, only 10% use sEMG control due to low accuracy, sensitivity and specificity of the control [1], [3].

D. Prosthetic Hand Control

Despite the mechanical effort to build and control the perfect prosthetic hand, modern technologies do not solve the biological sensors and intelligent control [15]. However, much research has been done, and more complex actions can be provided to the user [15]. Currently, the control of a prosthetic hand with the pre-processed sEMG signals is non-intuitive and requires training of the user [5]. Atzori and Müller [5] described the commonly used control strategy applied on the most available prosthetic hands to improve robustness and accuracy. The sequential control strategy defines a specific signal to trigger a predefined movement [5]. The activation signal can activate a single or a combination of more muscles, and the signals can be individualised to improve the user's acceptance [5]. However, to overcome the problem of intuitive control to increase prosthetics acceptance, much research has been done in pattern and gesture recognition based on multidimensional EMG signal recordings [3]. At this moment, the importance of Machine Learning as a classification tool has increased in recent years, providing the possibility to learn and perform tasks from provided electromyography recordings [16]. Nevertheless, machine learning is limited in processing EMG signals due to inconsistency, noise, abstraction and high dimensionality of the recorded EMG signals [16]. Therefore, Deep Learning, a branch of machine learning, is used for more complex tasks. The architecture consists of a hierarchical model of deep layers to extract feature information in multiple representative layers [16].

Arteaga et al. [1] compared five standard supervised methods: Linear Discriminant Analysis (LDA), Artificial Neural Networks (ANN), Hidden Markov Model (HMM), Support Vector Machine (SVM) and K-Nearest Neighbour (k-NN) based on the accuracy of recognising the proper movement via EMG signals. The accuracy of the methods is between 80% and 99% depending on the quantity of the input channels and defined gestures. Arteaga et al. [1] and Kumar et al. [3] described LDA as predominant and commonly used with additional thresholding. However, the feature selection and signal acquisition method can impact the classification accuracy [3]. Additionally challenging is the processing of real-time data. Thresholding to define the period of the contraction of the measured muscle and resolving the redundancy using reduction techniques like Principal Component Analysis (PCA) can reduce the quantity of data and increase the quality [5].

E. Visual Gesture Recognition

Gesture recognition based on Deep Learning is restricted to predetermined hand movements [5]. In [17] hand gestures were classified into two groups: dynamic and static. Static describes gestures with a stable shape, whereas dynamic describes the movement of the gestures. Castro et al. [18] proposed five static hand gestures: neutral position, pinch, tripod pinch with index, middle finger, thumb, closed hand, and opened hand, where all fingers and the thumb are stretched. Instead of interpreting EMG signals to recognise a gesture, visual gesture recognition is based on recording the hand shape and interpreting the gestures via different cameras [19]. Nowadays, cost-effective camera-vision-based sensor technologies are used for gesture recognition in clinical operations, sign language, robot control, virtual environments, home automation, and gaming [17]. Therefore, different hardware can be used, such as RGB, time of flight, thermal, or night vision cameras [17].

In [17] the methods currently employed in hand tracking using computer vision were classified. The first approach is colour-based recognition of skin colour by detecting matching pixels or region-based skin detection. Colour-based recognition is a popular facial and hand segmentation method for applications like person movement tracking, video observation, gesture identification, or degraded photograph recovery. Another approach is appearance-based recognition, which compares extracted features from the input image with modelled features imitating a visual appearance. Comparing the image to the model is based on haar-like features describing the posture pattern, histogram techniques, or edge detection techniques like Canny, Sobel, or Prewitt operators. The third approach described by Oudah et al. [17] is motion-based recognition. The region of interest is extracted from a series of image frames by defining the centre point of the hand and extracting features throughout the series of images, which are compared to a matching model. The depth-based recognition method is based on 3D geometric information about the hand. The advantage is that lightning, shade, or colour does not influence performance. However, costs, size, and availability are limiting factors. The last approach is skeleton-based recognition based on the geometric attributes of the hand to reduce complex features. The most commonly used geometric features are the joint orientation, space between joints, joint location, and the degree of angle between joints [17].

Oudah et al. [17] and Hu et al. [19] defined the flexibility and diversity of each person to perform the same gesture differently as the main challenge of visual gesture recognition. Furthermore, different skin colours, lighting, background, and other factors like speed impact the performance. However, much progress has been made in real-time hand tracking, which improved the acquisition of skeleton data [20]. Commercial hardware like LeapMotion [21], Intel Realsense Camera [22], and smart glasses like the HoloLens [23] provide tools to acquire skeleton data [20].

To interpret the acquired skeleton data, most hand gesture recognition methods are dependent on the selected features, which are explicit or implicit [19], [20], [24]. Explicit features are hand-crafted, and implicit features are extracted from machine learning algorithms. Chen et al. [20] described different explicit features based on the skeleton information of the hand. The first step is defining the hand skeleton via Fisher vector representations. A hand direction and wrist orientation histogram can then describe the hand movement. The second set of features to recognise a gesture is based on Handwriting-Inspired Features (HIF3D). The third approach is based on processing the trajectory of the hand, like smoothing or scaling. Despite the used method, the extracted features from the image are compared to model data through distance metrics. However, nowadays, Deep Neural Networks extract the implicit features for gesture recognition [19], [24]. Bhushan et al. [25] described popular classification techniques for gesture recognition like Naïve Bayes, K-Nearest Neighbours (KNN), random forest, XGBoost, Support vector classifier (SVC), logistic regression, Stochastic Gradient Descent Classifier (SGDC), and Convolution Neural Networks (CNN). In [17], [19] and [26], other popular methods like Feed-forward Neural Network (FNN), Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM) and Support Vector Machine (SVM) were presented to classify gestures.

F. Virtual, Augmented and Mixed Reality

Visual gesture recognition is also an essential topic in Human-Computer Interaction, especially in Virtual (VR), Augmented (AR) and Mixed Reality (MR) summarised as Extended Reality (XR) technologies [6], [19]. The most known methodology is VR, which completely immerses the user in a different environment by deceiving the senses of the user with a Head Mounted Display (HMD) or headset [6]. Instead, AR enhances the real world by overlaying digital elements [6]. In MR, the virtual and real components are combined and blended to allow the user to simultaneously manipulate and experience real and virtual environments [6]. In the healthcare industry, the significance of XR technologies has grown exponentially [6]. These technologies have improved education, training, diagnostics, and clinical decision-making through Head-Mounted Displays [6].

Palumbo's publication [6] summarised various healthcare applications using Optical See-Through Head-Mounted Displays. These applications include surgical navigation, which provides users with information to navigate the operation room, access medical data, and improve the positioning of real elements. Other applications utilise virtual avatars to enhance patient interaction and improve mental health. Additionally, XR technology can facilitate medical education, training, teaching, telemonitoring, and teleconsulting by visualising context and enabling interaction. In rehabilitation, XR can provide interactive digital training for daily activities while analysing and recording data. A study conducted by Held et al. [27] investigated the training effect with and without XR real-time feedback with stroke patients analysing the gait pattern. Based on real-time feedback through XR, it was concluded that patients demonstrated a positive gait adaptation during overground walking.

III. Requirements and Methods

Instead of stroke patients, this work focuses on upper limb amputees. This Chapter describes the requirements based on literature to develop a user-friendly Mixed Reality application to increase the acceptance rate of modern prosthetic hands. The concept of the application is presented, and finally, the evaluation methods are presented.

A. Requirments

Brack and Amalu [12] estimated the actual rejection rate of prosthetics to 60%. To understand the low acceptance rate, Kumar et al. [3] analysed the requirements and expectations of users. However, only the relevant requirements are summarised further in this Chapter. Users criticised the accuracy of classifying the correct command due to the self-placement of the electrodes and the lack of measuring stability. However, increasing the accuracy leads to decreased performance. Therefore, Kumar et al. [3] recommended a maximal response time of 150 to 250 ms. The lack of real-time feedback was also criticised, which can be beneficial to improve the ability of the user to control a device. Another requirement is an intuitive control and command structure. Kumar et al. [3] proposed using pattern recognition to provide individual, appropriate functionality and suitable control methods. The last relevant requirement is user training, a common reason for rejection due to a late start of posttraumatic intervention and waiting time for a prosthetic fit [3], [4]. Melero et al. [4] recommended XR applications to increase the patient's motivation and start rehabilitation earlier.

B. Hardware

For this project, the Optical See-Through Head-Mounted Display (OST-HMD) HoloLens 2, developed and manufactured by Microsoft, is used to deploy a mixed-reality application. It is a pair of smart MR glasses that mix virtual and real elements. The HoloLens 2 was released in 2019 and provides different interaction tools, including hand tracking. Compared to the predecessor model, HoloLens, released in 2016, the HoloLens 2 has an enhanced field of view, reduced weight and improved battery life [6].

C. Software

The application is developed using Unity, a cross-platform game engine providing features and tools for 2D and 3D effects. The scripts used in Unity are written in C# and edited in Visual Studio, an Integrated Development Environment (IDE) providing features and debugging tools supporting the workflow for Unity projects. Anaconda Navigator and Spyder are used for Python script editing. Anaconda Navigator is a graphical user interface (GUI) to manage Python environments and tools. Spyder is an IDE to edit Python scripts, providing features, tools, and library integration. To develop and test the application on the HoloLens 2, Holographic Remoting provided by Microsoft is used. It allows streaming the Mixed Reality application wireless from the development environment to the HoloLens 2, enabling real-time interaction and testing.

D. Project Plan

Kumar et al. [3] suggested training and testing each device individually to improve accuracy, adaptiveness and acceptability due to the variety of EMG signals of each user. Therefore, this work describes the development of an application for the HoloLens 2 to provide the user with a virtual prosthetic hand attached to the residual limb. For simplification, the user is assumed to have at least one healthy hand, which can be tracked. The virtual prosthesis is controlled through an EMG wristband by interpreting the muscle activity. The purpose is to provide the user with visual feedback and, therefore, able to start rehabilitation as soon as possible. Moreover, the application allows users to create a personalized dataset of labelled EMG signals to train the neural network interpreting the muscle activity. Therefore, the user is assumed to have a healthy hand which can be visually tracked. Additionally, the gesture of the tracked hand is visually classified and then assigned to the EMG signals acquired through the wristband. The other hand represents the prosthesis and performs the classified gesture for visual feedback. The idea behind training a neural network based on individual data is to improve gesture recognition through EMG signals. The application is structured into four main features: The virtual hand model, the hand tracking and acquisition of the skeleton data, the visual gesture recognition, and lastly, the virtual prosthetic hand control, as further described in the following.

E. Application Concept

This Chapter describes the application concept regarding the structure and logic of the single scenes and components.

1) Scene 1: Start Menu

The first scene (Figure 4a) is loaded in the beginning and displays the accuracy of the FNN model through the `TextMeshPro` element. It is part of a Mixed Reality Toolkit (MRTK) canvas with an additional menu with four buttons. The first button loads the second scene (Chapter III-E2) to record data of different gestures. The second button loads the scene described in III-E4, where the prosthetic hand is controlled via the classified EMG signals. The third button toggles the Handedness, which is by default set right, defining the tracked hand. A singleton class containing the variable is needed, providing a global access point throughout different scenes. The last button loads scene three (Chapter III-E3), handling the visual gesture classification. The menu and the text dynamically follow the movement of the head of the user by applying the camera pose to the canvas pose with a defined distance. The last element is the `GUILayout` button created in an editor class by overwriting the `OnInspectorGUI` method. This allows the user to add a button in the Inspector window that triggers the creation and training of the feed-forward neural network (FNN) model discussed in Chapter III-F.

2) Scene 2: Record Data

The second scene is to create and record a skeleton-based dataset of predefined gestures. The scene consists of an MRTK Canvas with a button to exit the scene and load the first scene (Chapter III-E1). The button follows the movement of the head by applying the camera pose to the canvas pose at a certain distance. The scene contains a right and a left-hand model projected on the corresponding tracked hand for visual feedback. Additionally, four different hand gestures are projected in front of the user. By reaching out to the corresponding gesture, the skeleton data is recorded.

3) Scene 3: Classify Gestures

The scene described in this chapter also contains an MRTK Canvas dynamically following the user's head motion. The canvas includes a button similar to Chapter III-E2 to exit and load the first scene (Chapter III-E1). The `TextMeshPro` element displays the current classified gesture. The recognition of performed gestures through skeleton data is based on an FNN, explained further in Chapter III-F. As before in Chapter III-E2 the scene also contains a right and a left-hand model projected on the corresponding tracked hand for visual feedback. However, this applies only to one hand. The other hand is a prosthesis, performing movements based on the classified gesture. The Handedness can be acquired by accessing the singleton class.

4) Scene 4: EMG Control

Similar to scene III-E3, an MRTK Canvas dynamically follows the user's head motion containing an exit button, loading scene one III-E1. It also includes a right and a left-hand model of whom one is controlled as a prosthetic hand, and the other is projected on the actual tracked hand. The assignment of which hand is the prosthesis happens through the singleton instance.

F. Gesture Recognition

The gesture recognition through skeleton data is based on a feed-forward neural network implemented in Python. The data describes the position $\mathbf{p}_j(t)$ and rotation $\mathbf{r}_j(t)$ of each joint in the world space at time t. However, the hand pose is independent of its global position. Therefore, the global coordinates are converted into relative coordinates (Equation 1 and Equation 2) of the wrist $\mathbf{p}_0(t)$ and $\mathbf{r}_0(t)$, allowing pose normalisation and hand sign alignment, increasing robustness and efficiency. Additionally, the coordinates are normalised by dividing each value by the extracted absolute maximum value of the data.

$$\mathbf{p}'_j(t) = \mathbf{p}_j(t) - \mathbf{p}_0(t) \tag{1}$$

$$\mathbf{r}'_j(t) = \mathbf{r}_j(t) - \mathbf{r}_0(t) \tag{2}$$

The FNN architecture is shown in Figure 1, consisting of fully connected layers. As the input Layer, the FNN expects the pre-processed skeleton data in two branches, one for the relative joint position and one for the relative joint rotation. The input is reshaped from a three-dimensional to a one-dimensional input tensor in the second layer. The next step merges both input branches in a concatenate layer to a combined tensor. The following Hidden Layers consist of weighted connections and activation functions to reduce the tensor, resulting in the number of predefined gestures of the output layer.

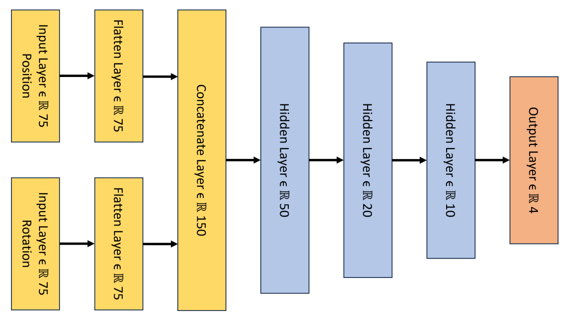


Fig. 1: Architecture of the feed-forward neural network

G. Interface

Scene 4 (Chapter III-E4) allows controlling the prosthetic hand through the EMG wristband. Therefore, an interface (Figure 2) is developed, with three instances: a microcontroller (ESP32) representing the EMG wristband to acquire the EMG signals, a Computer (PC) to handle the EMG signals and classify the current gesture and the HoloLens as the last instance, with the application described in this work.

The communication is based on the TCP client (green) and server (dark blue). The ESP32 creates a Local Network (light blue) with an included WLAN module. The PC opens a TCP Server Socket on a predefined port (orange) and waits for a client to connect. The HoloLens establishes a wireless TCP connection to the PC and creates a Network stream for data reception. The arrows visualise the connection establishment. However, the data stream is binary. The HoloLens continuously waits for data from the PC. The interface between ESP32 and PC is through a wire visualised as a continuous arrow. The EMG signals from the ESP32 are received on the PC, processed, and classified. The resulting label is then sent to the HoloLens, which triggers the prosthetic hand's movement.

H. Evaluation

The evaluation of this work is based on the performance measurement of the feed-forward neural network and the application in general, described further in this chapter.

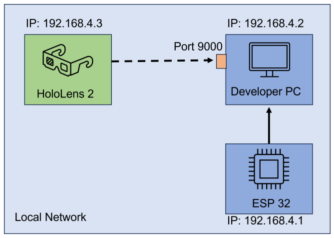


Fig. 2: Concept of Interface to EMG Classification

1) *Neural Network*

A confusion matrix is calculated to evaluate the neural network, comparing the classified label with the actual label based on testing data. Therefore, splitting the ground truth dataset into training and testing data is necessary. The training data is used to train, and the testing data is used to evaluate the FNN model. By splitting the data randomly, the model is assessed with independent data, ensuring reliable results.

When the predicted gesture is classified as positive and corresponds to the ground truth label, it is called True Positive (TP). In the case that the predicted gesture label is classified as positive but does not resemble the ground truth label, it is defined as False Positive (FP). If the predicted label is classified as negative, but the condition is positive, it is called False Negative (FN). The last combination is called True Negative (TN) and is defined if the predicted gesture classification is labelled negative and the condition is also negative.

To evaluate the model, the accuracy and the F1 score are calculated. The model's accuracy is determined using Equation 3, measuring the model's overall performance as the ratio of correctly classified instances to the total number of instances. Equation 4 calculates the F1 score, which is the harmonic mean of precision and sensitivity. The value of the F1 score is from 0 to 1, where 1 is the best possible value.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

$$F1Score = \frac{2TP}{2TP + FP + FN} \tag{4}$$

2) *Application Performance*

The Mixed Reality Documentation [28] offers criteria and testing methods for evaluating the quality of applications. The relevant criteria are summarised in this section with the methods to evaluate the quality of the presented application.

One of the essential criteria for evaluating application quality is the frame rate, which plays a crucial role in providing hologram stability and user comfort. The optimal frame rate achievable is 60 fps, measured in the Windows Device Portal. Stabilisation planes, distance to spatial anchors, and tracking can also influence hologram stability. In this application, hand tracking may also have an impact. The user can move their head from side to side to check if the hologram shows any unexpected movements or if the frame rate drops.

Another criterion is the hologram's position on natural surfaces, which is not a high priority for this application. However, it should be verified that the projected hands are not placed above or below the tracked hand to avoid user inconvenience. The viewing zone of comfort can also be evaluated by checking the content distances. According to the Mixed Reality Documentation [28], the content distance should be between 1.25 and 5 m, with exceptions like stationary content to prevent visual discomfort or fatigue of the eyes.

IV. RESULTS

In this Chapter, the results of this work are described and evaluated. The first part evaluates visual gesture recognition. The second part, the integration with the EMG wristband and the application evaluation, is described.

A. *Visual Gesture Recognition*

The app includes a feature for recording skeleton data, as outlined in Chapter III-E2. The used dataset consits of 23659 samples from a single user recorded with a sample rate corresponding to the HoloLens frame rate, which averages 60 frames per second (fps). However, the samples are summarised as an average of 15 frames, reducing the dataset to 1585 samples from a single user. To train and evaluate the FNN described in Chapter III-F, the recorded data is randomly split into 75% training and 25% testing data. Additionally, cross-validation is applied to achieve constant and best results. Therefore, the model is trained on different subsets of the data and evaluated to find the best-performing model represented in a confusion matrix.

A classification report (Table I), which includes precision, recall, F1-score, accuracy, and sample sizes, is extracted from the confusion matrix. Based on the provided data, an overall accuracy of 99% was achieved.

TABLE I: Visual Gesture Classification Classification Report

Index	Gesture	Precision	Recall	F1-Score	Sample Size
0	Fist	1.00	0.99	1.00	393
1	Pinch	0.99	1.00	0.99	460
2	Spread	0.99	0.99	0.99	395
3	Thumb up	1.00	0.99	1.00	337
	accuracy			0.99	1585

This result is cross-evaluated by a real-time classification. Hereby, the user performs each gesture ten times for approximately two seconds in random order. Every 250 ms, the current skeleton data is classified, and the label is recorded in a txt file with the current time. Whenever the probability is beneath 0.7, the gesture label is set to none. Additionally, the whole process is recorded via camera to define the actual label manually. In Figure 3, the target gesture label in red and the classified gesture in blue are visualised over time, represented by the recorded frames. Based on the comparison between the detected and the actual labels, an accuracy of 0.59 can be calculated. However, this simplified evaluation includes every sample and does not consider the movement between gesture changes. Therefore, the mean delay of 423 ms in recognising the true gesture is approximated.

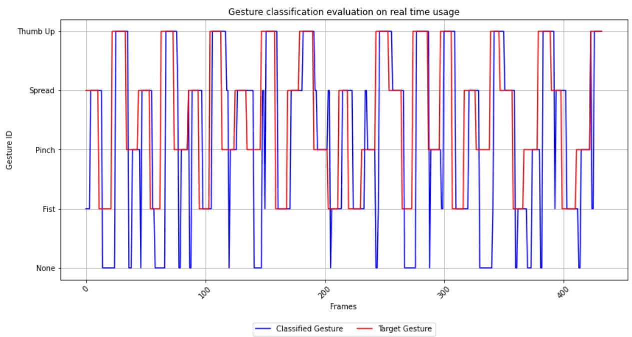
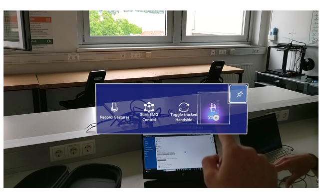


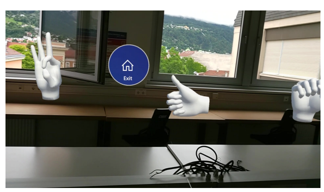
Fig. 3: Real Time Gesture Classification actual gesture vs classified gesture

B. Application

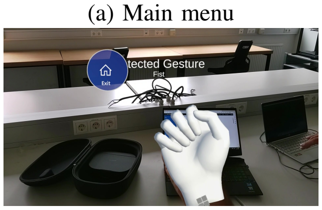
The application was successfully exported to the HoloLens and integrated with the EMG wristband. The different scenes are shown in Figure 4. The first scene (Figure 4a) shows the buttons to handle the different application options. Figure 4b shows three of four projected gestures. By reaching out to one projection, the recording of this gesture starts, and all projections are disabled. In Figure 4c, the visual gesture classification is performed, and the detected gesture is visualised as text. The EMG wristband is also worn, recording the muscle activity synchronously while the application records the performed gestures. Figure 4d shows scene four (Chapter III-E4), where the EMG wristband controls the virtual model.



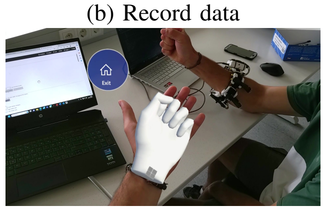
(a) Main menu



(b) Record data



(c) Visual gesture recognition



(d) EMG control

Fig. 4: Scenes of the application on the HoloLens 2

The application was evaluated by the criteria described in Chapter III-H2. The framerate measured in the Windows Device Portal did not show significant changes in the fps rate while moving the head from side to side. The model's position was also accurate, as shown in Figure 4. According to the Mixed Reality documentation, the UI elements were placed sufficiently far during development to prevent visual discomfort or fatigue of the eyes.

V. Conclusion

The developed application for the HoloLens 2 was successfully implemented and exported to a standalone Mixed Reality application and combined with an EMG wristband interpreting the muscle activity to predict the movement to perform with the prosthetic hand.

The application is structured in four different scenes. The first is the main menu to load the other scenes, each with an exit button leading back to the first scene. Additionally, the user can configure the handedness of the prosthetic hand and the visually tracked hand in the main menu. This information is needed for the scenes, handling the visual gesture recognition and the scene controlling the prosthetic hand through the EMG wristband to define the prosthetic hand. The visual gesture recognition is based on skeleton data interpreted by a feed-forward neural network. Another scene provides a projection of each gesture to acquire the training data for visual gesture recognition. By reaching out, the recording of the skeleton data starts labelling it with the corresponding projection gesture. By moving back, the recording stops. The neural network is trained by calculating each joint's average of 15 frames to reduce possible errors. In total, 1585 samples were recorded for the four gestures, resulting in an accuracy of 99%.

While performing each gesture ten times for approximately two seconds, only an accuracy of 59% could be achieved. However, the time between the change of the gesture was not considered, and additionally, the reference labels were defined manually, increasing the human error factor. The scene classifying the gesture visually during runtime additionally safes the label, probability and current time in a txt file to combine it with the EMG signals and create a training data set for interpreting the gesture based on the muscle activity. Therefore, the ESP32 recording the EMG signals receives a start command from the HoloLens via a TCP connection on the local network to start recording synchronously. The purpose of creating a new training data set of the neural networks is to improve performance and accuracy by assigning each patient their own trained neural network. The last scene, handling the prosthetic hand control based on the EMG wristband, attaches a virtual hand to the wrist joint of the user. To perform a gesture, the EMG signals of six sensors are sent from the EMG wristband to a PC via a USB cable, where the muscle activity is interpreted in 250 ms window. The classification is based on an artificial neural network with an accuracy of 97%, which decreases to 90% while testing the performance during runtime. After classification, the label is sent via an established TCP connection between the HoloLens and the PC on a local network provided by the ESP32. The new label of the gesture triggers the movement of the virtual prosthetic hand. The speed of performing the gesture can be configured, depending on the user's preferences. The other hand is projected similarly to the other scenes on the actual hand following the movements.

This work demonstrates a possible way to provide a training tool to train the intuitive control of a prosthetic hand without a prosthetic fit. Moreover, the application provides tools to adjust and create own data for gesture recognition based on skeleton data and EMG data. This data can then be used to train individual neural networks for each patient to increase acceptance and improve intuitive prosthetic control. However, the results are inaccurate, and the benefit is not yet validated. The accuracy could improve with the current system by a different evaluation process due to the human factor and not considering the movement duration between the different gestures. By introducing a non-gesture label in the target labels before and after every change by an average delay of 423 ms, the accuracy increases to 74%. Nevertheless, the gesture label is classified mostly correctly in the two-second window the user is asked to perform the movement. Therefore, filtering the labels and including a non-gesture classifier for the time between the gestures can significantly increase the accuracy of the current system.

Future work could include dynamic gesture recognition and improve visual gesture recognition. Therefore, different neural networks or features like the space between joints or orientation between the joints could be compared. Additionally, the integration of the EMG wristband could be improved by reducing the PC as an interface instance. Furthermore, the recorded labels of the visual gesture recognition and the EMG data are currently combined manually and synchronised. By automatisation of the process, the user comfort and applicability can be improved. Another application improvement could be the gamification of the training tool or adjusting the UI and further developing the application as a training tool by integrating a training plan.

References

[1] M. Arteaga, J. Castiblanco, I. Mondragon, J. Colorado, and C. Alvarado-Rojas, "Emg-driven hand model based on the classification of individual finger movements," *Biomedical Signal Processing and Control*, vol. 58, 2020.

[2] P. Maduri and H. Akhondi, "Upper limb amputation," *StatPearls - NCBI Bookshelf*, 2023.

[3] D. K. Kumar, B. Jelfs, X. Sui, and S. P. Arjunan, "Prosthetic hand control: A multidisciplinary review to identify strengths, shortcomings, and the future," *Biomedical Signal Processing and Control*, vol. 53, 2019.

[4] M. Melero *et al.*, "Upbeat: Augmented reality-guided dancing for prosthetic rehabilitation of upper limb amputees," *Journal of healthcare engineering*, vol. 2019, 2019.

[5] H. Atzori, M. Müller, "Control capabilities of myoelectric robotic prostheses by hand amputees: A scientific research and market overview," *Frontiers in systems neuroscience*, vol. 9, 2015.

[6] A. Palumbo, "Microsoft hololens 2 in medical and healthcare context: State of the art and future prospects," *Sensors*, vol. 22, no. 20, 2022.

[7] L. Wheaton, "Neurorehabilitation in upper limb amputation: understanding how neurophysiological changes can affect functional rehabilitation," *Journal of neuroengineering and rehabilitation*, vol. 14, no. 1, 2017.

[8] K. Ziegler-Graham, E. MacKenzie, P. Ephraim, T. Travison, and R. Brookmeyer, "Estimating the prevalence of limb loss in the united states: 2005 to 2050," *Archives of Physical Medicine and Rehabilitation*, vol. 89, no. 3, pp. 422–429, 2008.

[9] F. Cordella *et al.*, "Literature review on needs of upper limb prosthesis users," *Frontiers in Neuroscience*, vol. 10, no. 209, 2016.

[10] J. J. Bartoletta, J. S. Israel, and P. C. Rhee, "Transradial amputation with pedicled pronator quadratus interposition and advanced neuroma-prevention techniques," *The Journal of hand surgery*, vol. 46, no. 12, 2021.

[11] "Prosthetics and orthotics market - growth, trends, and forecasts (2023 - 2028)," Mordor Intelligence, Hyderabad, India, 2023, (accessed: 29.09.2023). [Online]. Available: https://www.mordorintelligence.com/industry-reports/prosthetics-and-orthotics

[12] R. Brack and E. Amalu, "A review of technology, materials and r&d challenges of upper limb prosthesis for improved user suitability," *Journal of Orthopaedics*, vol. 23, pp. 88–96, 2021.

[13] T. Tuncer, S. Dogan, and A. Subasi, "Surface emg signal classification using ternary pattern and discrete wavelet transform based feature extraction for hand movement recognition," *Biomedical Signal Processing and Control*, vol. 58, 2020.

[14] C. Pylatiuk and L. Döderlein, "Bionische" armprothesen. stand der forschung und entwicklung," *Der Orthopade*, vol. 35, no. 11, 2006.

[15] C. Pylatiuk and L. Döderlein, "Die kunsthand. ein uberblick über die prothetische versorgung der hand," *Der Orthopade*, vol. 32, no. 5, pp. 406–412, 2003.

[16] W. Li, P. Shi, and H. Yu, "Gesture recognition using surface electromyography and deep learning for prostheses hand: State-of-the-art, challenges, and future," *Frontiers in neuroscience*, vol. 15, 2021.

[17] M. Oudah, A. Al-Naji, and J. Chahl, "Hand gesture recognition based on computer vision: A review of techniques," *Journal of Imaging*, vol. 6, no. 8, 2020.

[18] M. C. F. Castro, S. P. Arjunan, and D. K. Kumar, "Selection of suitable hand gestures for reliable myoelectric human computer interface," *Biomedical engineering online*, vol. 14, 2015.

[19] Z. Hu, Y. Hu, J. Liu, B. Wu, D. Han, and T. Kurfess, "3d separable convolutional neural network for dynamic hand gesture recognition," *Neurocomputing*, vol. 318, pp. 151–161, 2018.

[20] X. Chen, G. Wang, H. Guo, C. Zhang, H. Wang, and L. Zhang, "Mfa-net: Motion feature augmented network for dynamic hand gesture recognition from skeletal data," *sensors*, vol. 19, no. 2, 2018.

[21] "ultraleap homepage," ultraleap, California, United States, 2023, (accessed: 29.09.2023). [Online]. Available: https://www.ultraleap.com/

[22] "intel realsense homepage," intel Realsense, 2023, (accessed: 29.09.2023). [Online]. Available: https://www.intelrealsense.com/

[23] "microsoft homepage," microsoft, 2023, (accessed: 29.09.2023). [Online]. Available: https://www.microsoft.com/

[24] G. Devineau, W. Xi, F. Moutarde, and J. Yang, "Deep learning for hand gesture recognition on skeletal data," in *13th IEEE Conference on Automatic Face and Gesture Recognition*, Xi'An, China, 2018.

[25] S. Bhushan, M. Alshehri, I. Keshta, A. K. Chakraverti, J. Rajpurohit, and Abugabah, "An experimental analysis of various machine learning algorithms for hand gesture recognition," *sensors*, vol. 11, no. 6, 2022.

[26] E. Yavsan and A. Uçar, "Gesture imitation and recognition using kinect sensor and extreme learning machines," *Measurment*, vol. 94, pp. 852–861, 2016.

[27] J. Held *et al.*, "Augmented reality-based rehabilitation of gait impairments: Case report," *JMIR Mhealth Uhealth*, vol. 8, no. 5, 2020.

[28] "Mrtk documentation," 2023, (accessed: 29.09.2023). [Online]. Available: https://learn.microsoft.com/en-us/windows/mixed-reality/mrtk-unity/mrtk2/?view=mrtkunity-2022-05