IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 17 January 2024, accepted 31 January 2024, date of publication 12 February 2024, date of current version 28 February 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3365274

RESEARCH ARTICLE

A Novel Hybrid Deep Learning Architecture for Dynamic Hand Gesture Recognition

DAVID RICHARD TOM HAX, PASCAL PENAVA, (Member, IEEE), SAMIRA KRODEL, LILIYA RAZOVA, AND RICARDO BUETTNER, (Senior Member, IEEE)

Chair of Information Systems and Data Science, University of Bayreuth, 95447 Bayreuth, Germany

Corresponding author: David Richard Tom Hax (David.Hax@uni-bayreuth.de)

This work was supported by the Open Access Publishing Fund of the University of Bayreuth.

ABSTRACT Hand gestures are a form of natural communication used in human-computer interaction, however, when gestures are video-based, extraction of features for classification is complex. Current machine learning models struggle to achieve high accuracies when using videos recorded in realistic environments. In this work, we propose a hybrid architecture consisting of a recurrent neural network (RNN), including a long short-term memory layer, on top of a convolutional neural network, to recognize dynamic hand gestures recorded in realistic environments. We used a dataset of 6 dynamic hand gestures: scroll-left, scroll-right, scroll-up, scroll-down, zoom-in, and zoom-out. Our implemented inception-v3 model extracted features and provided the wrapped frame-feature map as input for the RNN, which performs the final classification. The proposed model classifies gestures with an average accuracy of 83.66%. By doing so, we intend to narrow the disparity between realistic environments and high accuracy. Finally, we compare the accuracy of our proposed dynamic hand gesture recognition model with that of the benchmark.

INDEX TERMS Human-computer interaction, hand gesture recognition, video hand gesture, dynamic hand gesture, machine learning, deep learning, convolution neural networks, CNN, recurrent neural network, RNN, long-short-term memory, LSTM, inception model, inception-v3 architecture, hybrid architecture, feature extraction.

I. INTRODUCTION

Human-Computer Interaction (HCI) describes the relationship between humans and machines. With the increase in information technology in the last decades, computer systems have been integrated into our lives and are being relied upon in many everyday life activities [1]. This development requires a more natural and accessible HCI such as communicating through gestures [2].

The application of gesture recognition affects many relevant research areas such as smart home devices [3], gamification [4], education [5], driving, and even lie detection [6].

Gestures describe body movements transmitting and exchanging important information with the environment. Nonverbal communication plays an important role in human interactions as it is a natural and intuitive way to convey messages [6], [7]. In comparison to face, fingers, arms, and body, hand gestures are the most widely used in human-to-human interaction, therefore, easily applicable in communicating with computers [8].

Research on gesture recognition derives from glove-based devices, which the user had to put on and as a result, the sensors detected hand activity. However, this method is considered inconvenient and unsuitable for daily life [9]. A more suitable approach is vision-based detection. Gestures are captured by a video camera which creates a sequence of images [10], [11]. The use of deep learning algorithms, especially convolutional networks (CNN), contributes to the high performance of hand gesture recognition and demonstrates high accuracy [1], [12]. This method provides end-users with the opportunity to naturally interact with computers without requiring special equipment. However,

The associate editor coordinating the review of this manuscript and approving it for publication was Orazio Gambino.

vision-based gesture recognition encounters a range of challenges: real-time transmission of data, non-detection of unusual gestures, and, consequently, lack of response from the machine when faced with an unusual gesture [7].

Abdullahi and Chamnongthai [13], [14] showed a recognition technology that requires minimal external equipment and recognizes hand gestures even in realistic environments. In both projects, they have developed novel methods to enable American Sign Language (ASL) detection using a body-worn Leap Motion Controller (LMC) that records skeletal videos of the hands [13], [14]. With their novel architecture, they have made it possible on the one hand to classify dynamic hand signs well and also to include noise and errors in hand motion tracking [13]. And on the other hand, they have also developed an innovative method to distinguish similar hand signs [14]. However, the authors were forced to generate their own data, as extensive public datasets on 3D skeletal information about the hand and where the subject is in motion hardly exist [13], [14]. The data shown and constructed by the authors, reveals that on the one hand the environment is realistic, but on the other hand, the focus of the camera is on the hands rather than on the whole body of the subjects.

Overall and most importantly, realistic environments require training with datasets consisting of images taken in natural conditions. In previous works, most gestures were very close to the camera enabling easier conditions for a machine to detect and classify hand gestures in ideal environments or with clear focus to the hands. However, images of real-life gestures include other objects, people in the background, and other elements [15]. Figure 1 demonstrates the works, which achieved high accuracy using datasets with perfect environments,[1] whereas, realistic environments reduce the robustness of algorithms. This presents a research gap in hand gesture recognition.

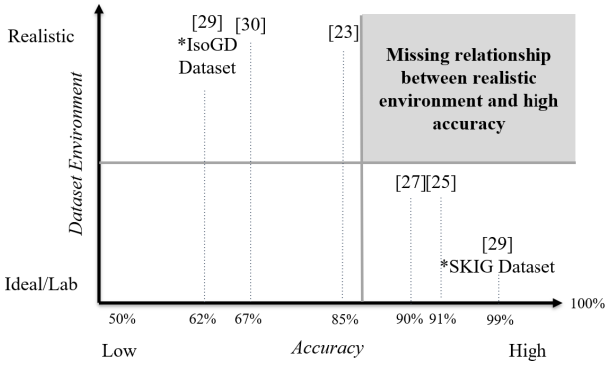

FIGURE 1. Current research gap between realistic environment and high accuracy.

The research gap also includes the need for a possibility for HCI in which subjects can simply make gestures in their natural environment with which they can interact with machines and do not have to learn a sometimes complex type of sign language. This requires an innovative method that can classify recordings of subjects' bodies in such a way that even unconscious gestures are not misrecognized. The basis for this is a new comprehensive data set that contains entire subjects performing simple gestures, without focusing on their hands.

In the following, we are aiming to discover to what extent a machine can detect and recognize hand gestures in realistic environments with the use of a deep learning model. Datasets capturing images of real-life gestures including other objects, people in the background, and other elements are required to test and train deep-learning models for real-life usage.

Concerning the research gap, our main contributions are as follows:

1) We have developed a novel hybrid deep learning architecture that is both computationally effective and quite capable of classification.

2) With an average accuracy of 83.66%, we have successfully managed to classify a new comprehensive dataset on hand gesture recognition, with competitively comparable results.

3) The architecture we show enables a system suitable for real-world use, in which users can be in realistic environments and without external equipment. As well as no need to focus on the hands for gesture recognition. And in which unintentional gestures are not misclassified and the subject's motion is taken into account.

In summary, with our novel deep learning architecture, we are creating a possible new basis for modern HCI with hand gesture recognition

This work is structured as follows: First, we provide an overview of the related work followed by a presentation of the used method, including a description of the dataset, applied approach, data pre-processing, and the evaluation of the model. After, we present the achieved results, followed by a discussion. Finally in the conclusion, we point out the limitations and describe future work.

II. RELATED WORK

A. EVOLUTION OF MODERN CONVOLUTIONAL NEURAL NETWORKS

CNNs are a specific type of deep learning architecture, that combines feature extraction and final classification into a single step. By convolution and pooling methods, they can optimize the feature extraction methods and thereby improve the final classification [16]. Under the name back-propagation algorithm, the foundation for today's CNNs was laid back in 1990, by establishing a network that could recognize handwritten digits [17]. One of the main applications of CNNs lies within image and video classification as well as object recognition, and to this day they are state-of-the-art for these applications [18]. However, the networks of that time had little to do with today's CNNs

[1]Perfect environment is defined as the following: no objects, or people in the background, only hands or upper body are visible.

and could not be used for more complex image and video classifications [19].

The first real breakthrough with more complex tasks was achieved by Krizhevsky et al. [20] with their CNN architecture AlexNet, which was the first model to win an ImageNet competition. ImageNet represents one of the most used image datasets for research. Building on this success, other architectures were able to further improve classification accuracy: VGGNet [21], GoogleNet (Inception Models) [22] and ResNet [23]. The trend of the models was to increase their depth and complexity and thus they have gained classification accuracy. The increasing complexity of the models has also increased the computational power required, which has made the models less and less practical for real-world use with less powerful computing devices [19].

However, the current trend and state-of-the-art of classic CNN architectures is trying to move back towards less complex and more light-weights models that can also be used on mobile devices or in general in real-world scenarios [18].

EXCERPT ON DEEP LEARNING FOR VIDEO CLASSIFICATION

Basically, there are five different model architectures for video classification tasks. **2-dimensional (2D) CNNs** are the most computationally inexpensive among them. They use classical CNNs to perform feature extraction for the individual frames and then use state-of-the-art classification models to classify the fused features of all frames. **3-dimensional (3D) CNNs** are more complex than 2D CNNs, and take the approach of adding another dimension to the CNN to include the movements between individual frames. Also **Two-Stream Approaches** are computationally quite expensive in that they use both a CNN for the extraction of spatial features and parallel a Recurrent Neural Network (RNN) to detect the temporal features. Both feature vectors are then fused for the final classification. **RNNs** can also be used individually for classification, and thus represent a less complex architecture compared to two-stream approaches. There are also **Hybrid Approaches**, in which CNN and RNN architectures are combined to be able to include both spatial and temporal features. Regarding RNNs, it can be seen that long short-term memory (LSTM) and gated recurrent unit (GRU) networks perform best and are therefore the two most frequently used networks [24].

Given the complexity of the models, it must be noted that the inclusion of optical flow via RNNs can lead to better classification results on the one hand, but on the other hand requires very high computing power, which makes real-world use more difficult [24].

B. GESTURE RECOGNITION WITH DEEP LEARNING

There are numerous studies aiming to recognize hand gestures with the aid of deep learning. In the following section, we will review the most notable works.

Many papers achieved excellent technical accuracy in research of both static and dynamic hand gesture recognition [25], [26], [27]. Static hand gesture recognition recognizes gestures while those poses are fixed (in the form of images) and hands do not move. On the contrary dynamic hand gesture recognition can recognize gestures in motion in the form of videos [27]. Static hand gesture recognition is often used in applications where gestures are performed slowly and the image can be recognized as a single frame, for example when interpreting sign language [25], [28]. Dynamic gestures involve a sequence of frames over time and differentiate in length due to the speed of motion. Hand motions are common for real-time recognition. This is possible with the aid of static and dynamic models for classification, whereas dynamic models have advantages over static ones due to easier processing online [29]. However, classifying dynamic hand gestures is more complex because both the hand shape, the movement as well as the direction of the hand must be taken into consideration [30]. As HCI requires real-time reaction, processing fast motions in complex environments and backgrounds, this research will focus on dynamic gestures only [31].

Two novel methods emerged exploring real-time hand gesture recognition: Temporal Segment Networks (TSN) and Temporal Shift Modules (TSM). Both tools are applied in deep learning to analyze videos. TSNs split videos into segments and analyze each segment separately so that the machine recognizes the gesture in the overall video according to the prediction of each segment [32]. However, the disadvantage is that TSN cannot build a temporal structure of the video as well as put the segments in the right order. Whereas TSMs support the neural network to capture the changing patterns over time in the video, by enabling time adjustments of different video segments. It means that TSM can arrange the video streams in the right temporal relationship [33] These methods reached an accuracy of 82.90% (TSN) and 85.10% (TSM), indicating a slightly better result with TSM than with TSN [34].

A further approach, presented by Naguri and Bunescu [35], combines traditional CNNs with LSTM networks to recognize dynamic hand gestures from video sequences. This approach reached promising results and an accuracy of 97.00%. As the gestures have been tracked by Leap Motion sensor and were based not on images or videos but on vectors, another paper using CNN and LSTM has been investigated [35].

Nguyen and Luong [36] further developed this approach on two streams of CNN architecture, allowing the system to consider both the spatial and temporal aspects of the hand gestures in videos. Hand gestures often involve complex movements and disproportions of the hand. A two-stream CNN with an LSTM can better capture these dynamics compared to a single-stream CNN, reaching an accuracy of 91.25% and improving computational time and memory resources of the ConvNet model [36].

Another useful method to evaluate video data is 3D CNNs [37]. 3D CNNs can learn spatiotemporal features directly from multiple feature maps, while 2D CNNs have to learn only from a single feature map including width and height dimensions, but lack the time dimensions [15]. Zhang and Wang [38] applied this advanced CNN method to dynamic gestures and tested it with people sitting close to the camera and showing particular gestures. This approach achieved a good accuracy of 90%. However, the authors state that it works only under the condition of having a small distance between a person and a web camera [38]. Another important challenge for dynamic hand gesture recognition is a complex background, which distracts the machine from relatively small hands and arms. This obstacle is tackled by Zhu et al. combining both above mentioned networks 3D CNN and LSTM. The key mentioned by the author is that CNN effectively extracts short-term spatiotemporal features, and passes them as input to LSTM. This one helps to distinguish the correlation between separate frames in a sequence learning long-term spatiotemporal features. Together it builds a good framework for video recognition [39].

Abdullahi et al. [13], [14] have also shown novel approaches and architectures that can be used to better recognize ASL hand gestures. Their approach is based on skeletal hand videos of LMCs. First, the authors present a new approach based on multi-stacked deep LSTMs, which allows them to be sensitive to hand dynamics compared to previous work. Also, their model allows the inclusion of noise and errors in hand motion tracking. The architecture uses a newly developed schema based on a bi-directional recurrent neural network that handles feature extraction, overcoming the problem of single LSTMs not being able to learn features as well as their susceptibility to overfitting [13]. In another work, they have developed a model architecture that addresses the existing problem of distinguishing similar hand gestures in ASL. The new architecture uses a fast fisher vector to extract the features from the video data and then pass them on to a bi-directional long-short term memory network [14]. In both works, the authors were able to show great progress in ASL gesture recognition. However, there are no widely available datasets for this type of HCI, which is why the authors had to use specially constructed data in which the focus of the videos is on the hands.

Another approach uses ultra-wide-band (UWB) radars to enable classification in more realistic environments. Skaria et al. [40] tested architectures with 3D CNNs or LSTMs for feature extraction with UWB data, while Park et al. [41] showed a novel architecture using fast-fourier transforms for classification to convert the video data from a time-domain to a frequency-domain, thus improving classification results for CNNs using UWB data. In both cases, very good results were achieved, although no comprehensive dataset was available and own experiments were carried out. UWB sensors for data acquisition are required for either application.

When showcasing hand gesture recognition methods and their accuracy, the datasets should also be taken into consideration. The size of the dataset, the variety of gestures, and the complexity of the background of the images all influence how accurate the result is. According to our comparison of datasets provided by [15], one of the benchmarks is the ChaLearn LAP Iso GD (IsoGD) dataset. This one contains 47,933 videos with 249 action types, each sample contains two values, RGB and depth. Each gesture was shot in a natural environment. To improve the robustness, objective conditions such as lighting and background were changed, and nonstandard gestures were included [15]. However, all studies testing their models using the IsoGD dataset show relatively low accuracy. The highest accuracy of 67.71% was reached by applying the ResC3D model [43].

Another benchmark is the Sheffield Kinect Gesture dataset. Including a total of 10 categories of hand gestures: circle (clockwise), triangle (anti-clockwise), up-down, right-left, wave, ''Z'', cross, come here, turn-around, and pat. However, this dataset consists of videos only with hand, whereas the whole body is not visible. [15]. Research studies using this dataset demonstrate high-accuracy results. When applying the 3D CNN and bidirectional ConvLSTM as high as 99.53%. When this method was applied to the IsoGD dataset, the accuracy lowered to 62.14% [42].

Table 1 depicts an overview of the works with used datasets (perfect/lab-like or realistic environment) and achieved accuracy.

To summarize, it is evident that datasets depicting more realistic conditions have a lower accuracy: datasets IPN Hand and IsoGD received the lowest accuracy with 62.14% [42], 67.71% [43], 82.90% and 85.10% [34]. All studies using less complex datasets achieved an accuracy above 90.00% [36], [38], [42].

Alzubaidi et al. [44] mention data scarcity as a crucial problem for deep learning models in general but also specific for video recognition. The lack of data can severely impair the quality of a video recognition algorithm for gesture recognition, for example. Sufficient video data is not available for every practical application, which could make gesture recognition more difficult to use. However, there are also state-of-the-art solutions for this challenge.

On the one hand, the possibility of artificially generating several data based on the few existing data using a Generative Adversarial Network in order to subsequently train the deep learning models is discussed [45], [46]. On the other hand, transfer learning is presented, which represents the possibility of ''further training'' another model based on an already trained deep learning model with just a little data and using the finished model for this purpose. With this approach, use cases with only a small amount of initial video data can also be realized with a separate model [47].

TABLE 1. Related Work using realistic or lab-like environment datasets and their respective accuracies.

Work	Dataset Size	Deep Learning Method	Environment		Accuracy
			Real	Lab-like	
Zhang & Wang, 2019 [38]	20BN-jester: 148,092 videos	3D CNN		X	90.00%
Nguyen & Luong, 2020 [36]	20BN-jester: 148,092 videos	Two-stream CNN & LSTM		X	91.25%
Zhang et al., 2017 [42]	SKIG: 2,160 videos	3D CNN & ConvLSTM		X	99.53%
Zhang et al., 2017 [42]	IsoGD: 47,933 videos	3D CNN & ConvLSTM	X		62.14%
Miao et al., 2017 [43]	IsoGD: 47,933 videos	ResC3D	X		67.71%
Benitez-Garcia et al., 2021 [34]	IPN Hand: 4,000 videos	TSN, TSM	X		82.90% (TSN), 85.10% (TSM)

C. HYBRID DEEP LEARNING BASICS

In a nutshell, the hybrid deep learning approach in video gesture recognition is based on the fact that a gesture cannot be recognized by a single frame, but rather that there is a temporary relationship between individual frames of a gesture [48]. Several studies have therefore already attempted to use a hybrid approach between a CNN and an RNN, in which the output of the CNN is used as input for the RNN. In video classification tasks in general [49], [50], but also especially in hand gesture recognition tasks [36], [48], [51].

While in a regular neural network, the neurons are only connected forward, in an RNN there is the possibility that neurons can also be connected to each other within a layer and also in opposite directions between layers [52]. The concept stems from the realization that the input and output of a neural network can be dependent on each other, for example in video sequences, where the classification of a frame can depend on the classification of a previous frame [53].

LSTMs are a type of RNN that allows long-term dependencies to be recognized and are therefore well suited for the classification of sequences such as videos. In contrast to regular RNNs, LSTMs have so-called memory cells, which contain the output of the previous time and, based on the new input, decide whether the information is relevant and should be kept or discarded. In this way, connections between the individual points in time can be recognized and retained [54].

As shown in figure 2, the CNN is used for feature extraction. CNNs are particularly well suited to identifying the features of image data that are relevant for classification by means of convolution and pooling, thereby forming feature maps that facilitate classification and the identification of patterns [55]. CNNs classify each image individually and do not take into account any temporal relationships, instead, they extract depth features for the classification of the images. The hybrid architecture makes use of this fact and lets the CNN output the feature maps for the individual frames, but does not yet classify the images. As shown in figure 2, the feature maps of all frames inside a time-frame are then passed to the LSTM, which then performs the temporary feature extraction and the final classification. This approach reaches better classification results, since it makes use of both deep learning architectures [36], [48], [51].

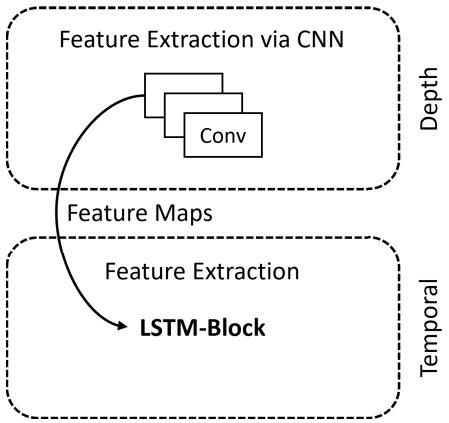

FIGURE 2. CNN-RNN basic architecture.

III. METHOD

A. HYBRID DEEP LEARNING ARCHITECTURE

In contrast to static images that represent hand signs, gestures include movement and contain spatial information. These more complex data are depicted using videos, which are at their core image frames arranged in a fixed order, also called a frame sequence. For motion recognition, this order and, thus the temporal relationship, is crucial and must be considered [15], [27], [56]. For this reason, a hybrid architecture was applied by stacking an RNN which includes an LSTM layer [54] on top of a CNN. In our case, the Inception-v3 architecture by Google Inc. [57] was used to extract the relevant features of the individual image frames and provide the wrapped frame-feature map as input for the RNN which performs the final classification. It is shown that transforming video files into 2D spatiotemporal feature maps with CNNs is a promising approach when it comes to video classification tasks [42].

This kind of hybrid architecture is known as CNN-RNN [36]. CNNs have been proven to work sufficiently when it comes to static image recognition tasks. As stated by Szegedy [57] the Inception-v3 architecture is highly

optimized to be as ''[efficient] as possible by suitably factorizing convolutions and aggressive regularization''. This efficiency is relevant in case of limited hardware specifications and confirmed our selection of the inception architecture as a feature extractor in our machine learning pipeline. The final classification layers are removed from the inception architecture as it is shown in figure 3. This reduction is necessary because the top layers are implemented to classify single images. Thus, the model provides 2048 features for every individual frame as output which acts as input vector (all frames together) for our following RNN which is referred to as the frame-feature map. This input is passed with the dimensionality (None, 10, 2048) with ''None'' being a placeholder for varying batch sizes.

For CNNs, the model architecture follows established patterns by applying a mathematical construct that typically consists of three layers in sequence: convolution, pooling, and fully connected (dense) layers. While the first two perform feature extraction, the final fully connected layer concatenates these features into a final output, in our case the final classification. A CNN is a type of deep learning model that can automatically and adaptively learn spatial information from low- to high-level patterns. The convolution task, as the name implies, is a key process in CNN. 2D convolution is a composition of mathematical operations where a specialized type of linear operation is applied on a 2D grid which is typical for image data. The linear operation that is applied to several image positions is called kernel and consist of a small grid of parameter mostly in a quadratic shape. A kernel size, which is typically 3×3, and the number of kernels that are used for one convolution, are key hyperparameters in CNNs [58]. Like conventional established CNNs the inception model keeps these mentioned patterns, but it should be added that the mapping of the extracted features is happening in the following RNN model where we have two fully connected layers.

The Inception-v3 model architecture, shown in figure 3, mainly consists of 2D convolutional layers with batch normalization and rectified linear unit (ReLU) activation output, 2D pooling layers, and the inception blocks. As pooling layers, both manifestations, average-pooling and max-pooling, are used. The inception blocks consist of parallel connected sub-pipelines mainly composed of convolutional layers. These sub-pipelines are concatenated before being passed on to the next sequential layer in the main pipeline, which is frequently another Inception-v3 block.

Since the parallelized sub-pipelines within an inception block can also be broken down into different structures, such as multiple convolutional layers with or without different pooling layers that can vary in layer depth, readers are referred to Szegedy [57] for more detailed information. The outstanding benefit associated with this parallelized structure is the simultaneous application of filters with different kernel sizes. This led not only to a big decrease in computational

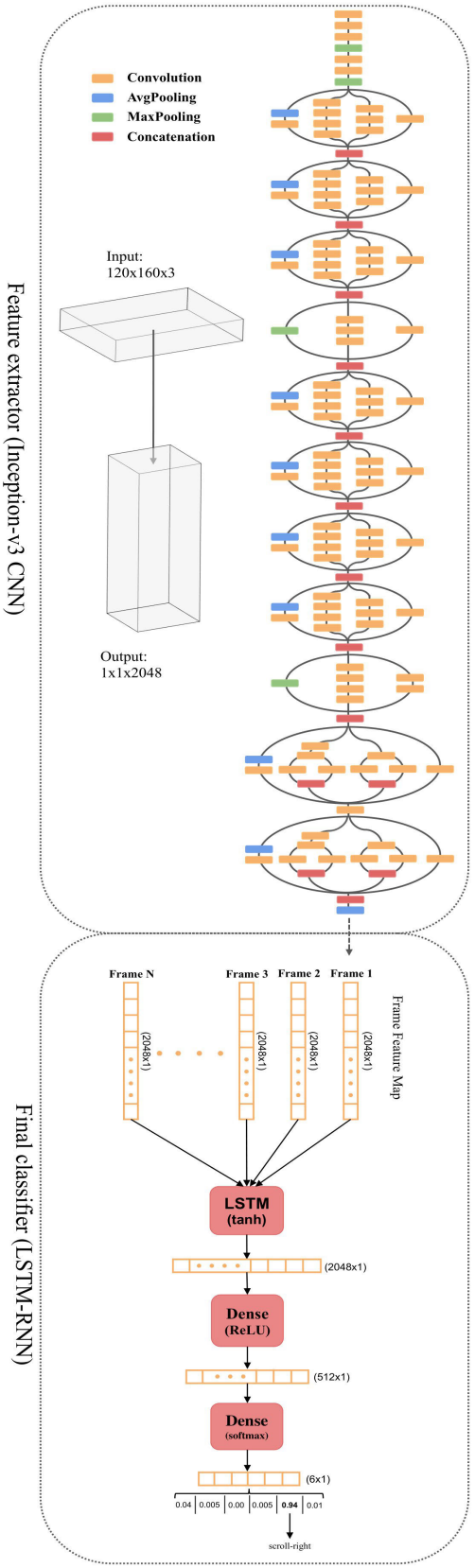

FIGURE 3. Hybrid architecture.

costs but also to the ability to focus both on coarser and finer patterns simultaneously and map them together at the end of one inception block [57]. Following the last concatenation, the input is transferred to a 2D global average pooling layer where the dimensionality is finally reduced from (2, 3, 2048) to (1, 1, 2048) and is then passed into the RNN as a wrapped frame-feature map (10 frames at once).

Our RNN essentially consists of the LSTM layer which plays a key role in incorporating the temporal context of the individual frames in terms of classification. RNN is one of the most forward ways to exploit sequences of inputs. LSTM networks represent the further development of RNN which facilitates remembering historic data [36]. With conventional ''backward propagation through time'' or ''real-time recurrent learning'' the problem arises that error signals either blow up or vanish. While the further may lead to oscillating weights, the latter may lead to insufficiently bridge long time lags. As a solution, Hochreiter and Schmidhuber introduced a novel, efficient, gradient-based method (LSTM) which improved many RNN architectures not only in speed [54].

The dropout layers prevent the model from overfitting on the training data by forcing the model to forget a part of the neural connections (in our case 50%). A dropout is embedded in the LSTM layer but also between the two fully connected dense layers. The last dense layer leads to the final classification using the softmax-activation function.

In comparison to the architectures shown in the related works and in general to the three main architectures in the field of dynamic gesture recognition, namely ''Two-Stream Networks'', ''3D CNNs'' and ''LSTMs.'' Reference [15], we show that a hybrid architecture, based on a 2D CNN and an LSTM network, as it is already used in the field of general video classification can also be used. In contrast to two-stream networks, the CNN and RNN do not run in parallel with feature vector fusion, but the two models are used sequentially and the CNN's feature vectors serve as input for the RNN [24].

B. EVALUATION

When it comes to model compilation and evaluation the focus is on the accuracy of the correct predictions. The accuracy is calculated as the division of the sum of the true-positive-instances (TP) and the true-negative-instances (TN) through the total population (TOTAL)(1).

$$Accuracy = \frac{TP + TN}{TOTAL} \tag{1}$$

Additionally, to show deeper insights into the classification errors the confusion matrix is used. Supported by this visualization we present and discuss the precision (2) and the true-positive-rate (TPR) also called Sensitivity (SN) or Recall in this context. The precision is calculated by the division of the TP through the positive predictions (POS PRED). The TPR is calculated as all TP divided through the number of actually positive instances (3).

$$Precision = \frac{TP}{POS\ PRED} \tag{2}$$

$$Sensitivity(TPR) = \frac{TP}{ACTUAL\ POS} \tag{3}$$

All the stated 6 gestures are equally relevant so we can aggregate our calculations. To ensure a more robust and representative model performance evaluation k-fold cross-validation is applied as described in III-D (Data Pre-Processing).

To further evaluate our shown method, we use another architecture to compare their classification accuracies. Since we use an approach using an RNN, we use GRUs as a reference method, as these together with LSTM networks are the most promising types of RNNs with the best accuracies in video classification tasks [24]. For the comparison, we leave the same data in the same folds as in the architecture with the LSTM network, but replace it with a GRU, resulting in a CNN-GRU architecture. We then calculate the accuracies for each fold and compare them with the initial results.

In addition to changing the RNN architectures, we also tested other state-of-the-art models for feature extraction to demonstrate the robustness and efficiency of our chosen approach. In addition to the inception model described above, we tested feature extraction using the VGG16, VGG19, ResNet50, and Xception models, which are all currently used models for feature extraction in research. The architecture already described changes to the extent that the CNN for feature extraction is replaced in comparison to the methodology described and the new architecture is tested and compared with both RNN methods. The CNN models differ in their architecture but can be exchanged flexibly in our proposed architecture. However, the feature vectors generated by the models differ in their dimensions. While the Inception, ResNet50, and Xception models output vectors with 2048 features, the VGG models generate vectors with only 512 features. Due to this fact, we had to adapt the input shape of the RNN architectures to the different feature vectors.

C. HAND GESTURE DATASET

We used the Depth_Camera_Dataset [59] to train and evaluate our machine learning model which is made up of 6 different hand gestures with a total of 662 sequences corresponding to each gesture. Each sequence contains 40 frames where one of the following actions is executed: scroll-left, scroll-right, scroll-up, scroll-down, zoom-in, and zoom-out. 10 frames of scroll-right and zoom-out are presented in figure 4, respectively. These gestures are carried out by different people at different time instances. Additionally, the entire body is in front of the camera, accomplishing natural movements to simulate natural HCI instead of an isolated, lab-like environment. Moreover, the background of the captured images is complex and realistic, sometimes including other people in the background and other factors

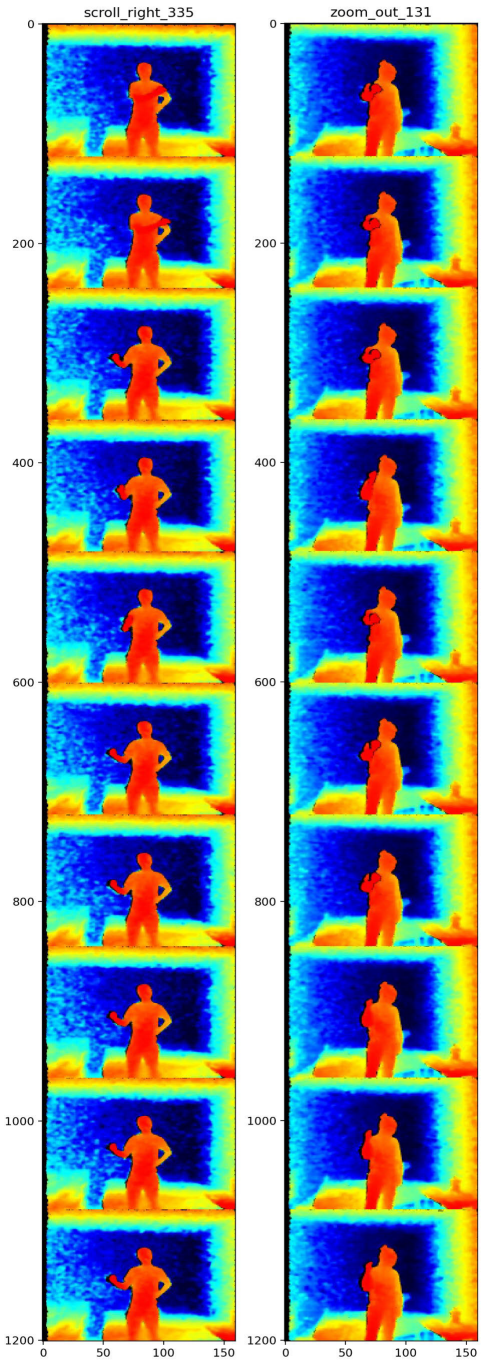

FIGURE 4. Example of RGB dynamic hand gestures: scroll-right and zoom-out.

that occur in real-world situations. This realistic environment makes the dataset stand out. As stated by the collectors of the dataset, the hand is oriented in various ways while being captured simultaneously. All the movements are captured with an Intel RealSense Depth Camera D435 which collected both RGB and depth frames at the same time. For our data-driven approach to classifying these gestures by a machine learning algorithm, we focused on the RGB variant of the dataset. Because of hardware and runtime limitations, we used 3000 sequences (500 sequences per class) and extracted only every 4th frame out of the respective sequence.

D. DATA PRE-PROCESSING

For data pre-processing, the Python libraries NumPy, Pandas, and TensorFlow are mainly used. The individual frames of a gesture sequence are loaded from a directory and stored in a Pandas DataFrame as a 4-dimensional array where the individual frames are stacked and represent one entire movement. The dimensionalities result from the height and width of the individual frames, where each of the data points consists of an RGB tuple of depth 3, which represents the respective color channel in a 0-255 encoding. The final input format is therefore (10, 120, 160, 3) because each of our frames has a resolution of 120 $\times$ 160 pixels and we focused on 10 frames per gesture. Regarding the data size and limited computing capacity only every 4th frame is used, as more frames did not increase the accuracy to an effective comprehensive.

Furthermore, a uniformly distributed sample size of data is preprocessed containing 3000 movements (500 $\times$ 6 different gestures) separated with the percentages 70%, 20%, and 10% for training, validation, and test data respectively. The dataset is shuffled before the allocation is made but it is ensured that the percentage of classes in each dataset is maintained to be representative. There is no predefined train-test split on this dataset, so we ensured a random division in the respective sets using the Python pseudorandom module random. K-fold cross-validation is applied by dividing the 3000 movements into 5 stratified folds where the percentage of samples for each class is maintained through every train/test split across all folds here as well. This ensures a more robust model performance evaluation since the effects of a biasing nonrepresentative train/test split are averaged out by evaluating the different folds. It is shown that batching the individual instances for training and testing does not yield to increased accuracy, so we provided the instances unbatched in our final model fit.

IV. RESULTS

The end-to-end machine learning project was developed and executed in Google Colaboratory (Colab). Colab is known as a cloud service based on Jupyter Notebook and provides access to a runtime configured for deep learning and a robust GPU as well as a user-friendly interface for developing and educating purposes. Nevertheless, there are of course limitations in terms of GPU, RAM, and runtime. The GPU we applied for the model compilation was mainly the NVIDIA-A100-SXM. The functionalities and modules of the TensorFlow v2.11.0 environment were used for this purpose. The model is trained for 100 epochs with an initial learning rate of 0.001. To ensure sufficient learning the categorical-cross-entropy cost function as training- and validation loss is minimized utilizing the adam-optimizer.

During the whole pipeline, the activation function rectified linear unit (ReLU) is applied before transferring the output

into the next layer, except for the LSTM layer and the final output layer. There, the hyperbolic tangent (TANH) and the softmax function are used, respectively.

The saturation of training and validation loss can be obtained in figure 8. The training- and validation accuracy also saturates as shown in figure 9 and achieved the percentages 96.43% and 85.50% after 100 epochs on our best fold. On the final evaluation of the test set an accuracy of 86.33% could be achieved. The final train-, validation-, and test-accuracies of the remaining folds are shown in table 2. By applying k-fold cross-validation with 5 stratified folds we achieved the averaged accuracies of 96.73%, 84.77%, and 83.66% for the train-, validation-, and test-sets.

The confusion matrix resulting from the test evaluation can be observed in figure 5.

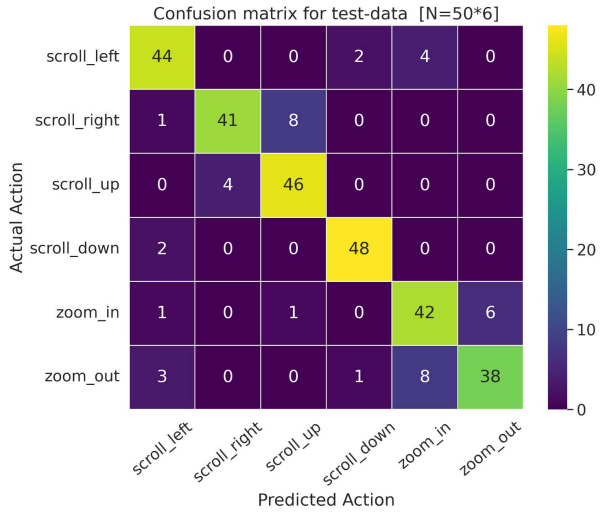

FIGURE 5. Confusion matrix of test-data fold 1.

As it can be extracted from the matrix the gesture scroll-down is the most precisely detected movement with a TPR of 96%. The zoom-out gesture is the least precisely detected movement with a TPR of 76%. Most classification confusions occur to the algorithm when distinguishing the gestures scroll-right and scroll-up as well as zoom-in and zoom-out. In 16% of the cases where the true gesture was scroll-right, our model misclassified the movement as scroll-up. In 12% of the cases, where the true gesture was zoom-in our model misclassified the movement as zoom-out. In 16% of the cases where the true gesture was zoom-in, our model misclassified the movement as zoom-out. As can be seen in figures 6 and 7 this behavior also occurs on the associated training and validation data.

The training and validation loss of other folds as well as the associated training and validation accuracy can be observed in the appendix (figures 1-4). Further confusion matrices resulting from other folds can be observed in the appendix as well (figures 5-16).

For many classification tasks, it is often important to determine how close the decision from the algorithm was.

FIGURE 6. Confusion matrix of training-data fold 1.

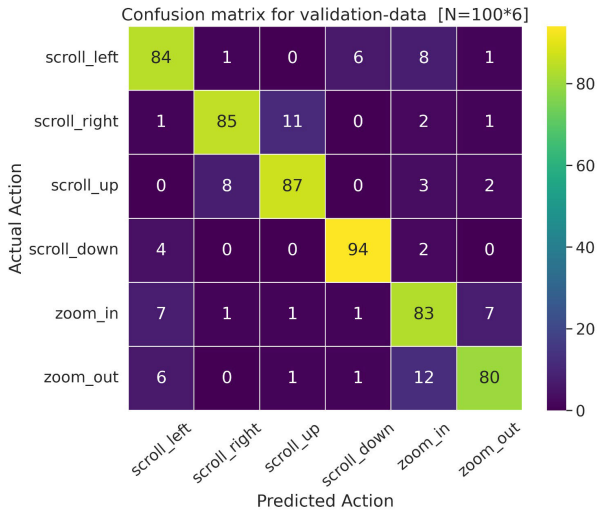

FIGURE 7. Confusion matrix of validation-data fold 1.

Because the final classification layers deliver a probability array it is often a very close decision between the Top N classified classes. For this purpose, we further evaluated how well our model predicted when the correct prediction has only the second or third-highest probability.

In addition to the results of the Inception-RNN architectures, table 2 also shows the results of the Xception-RNN architectures. The classification results of the CNN-RNN architectures with VGG16, VGG19, and ResNet50 as feature extraction models were not shown, as these did not exceed the random guess probability of 16.67% across all folds.

V. DISCUSSION

As shown in the accuracy and loss plot in figures 8, 9 our proposed model architecture successfully learns to solve our classification problem with a solid test accuracy. The amount of data is sufficient in interplay with the complexity

TABLE 2. Achieved accuracies of all datasets across all folds.

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Average
Train Accuracy (Inception-LSTM)	96.43%	97.81%	97.63%	93.14%	98.62%	96.73%
Validation Accuracy (Inception-LSTM)	85.50%	86.17%	85.33%	82.00%	84.83%	84.77%
Test Accuracy (Inception-LSTM)	86.33%	83.33%	82.00%	82.64%	84.00%	83.66%
Test Accuracy (Inception-GRU)	80.33%	80.00%	73.67%	81.00%	77.00%	78.40%
Test Accuracy (Xception-LSTM)	74.00%	68.00%	71.00%	69.67%	70.00%	70.53%
Test Accuracy (Xception-GRU)	63.00%	50.33%	57.00%	55.33%	54.33%	56.00%

FIGURE 8. Loss during training.

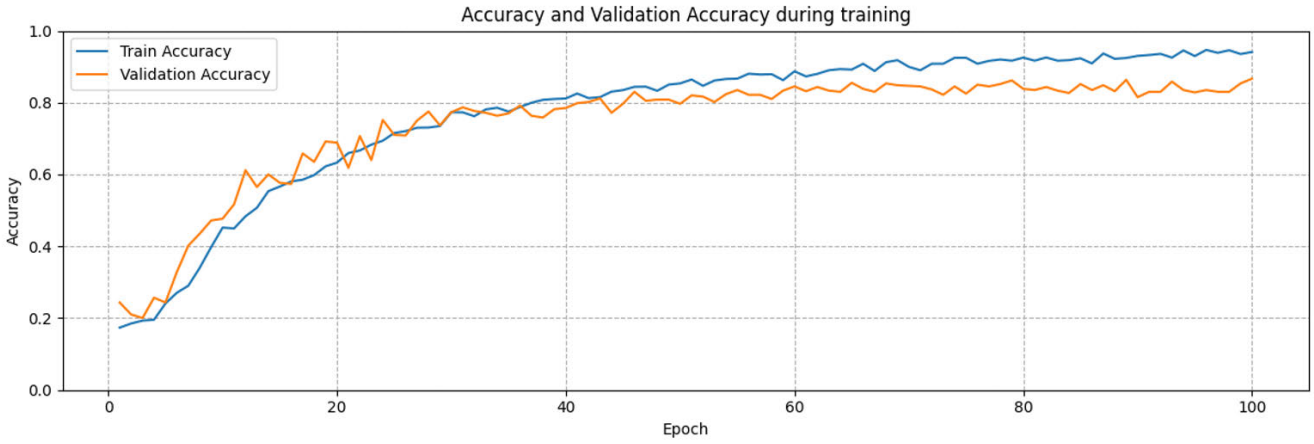

FIGURE 9. Accuracy during training.

of input properties to allow a proper generalization. This can be substantiated by the robust accuracy resulting from all train/test splits across all folds. It is shown whether an overfitting or an underfitting has occurred. The risk of overfitting the training data is also addressed with the two dropout layers at the top of the RNN.

By looking at the confusion matrices it is shown that the error properties maintain over the different sets across all folds. The most misclassifications occur when the model is distinguishing the gestures scroll-right and scroll-up as well as zoom-in and zoom-out. The former misclassification is very likely to occur because of unclean movements. In the

post-analysis of the misclassifications, it could be observed that individual probands moved their arm slightly up before or while scrolling right. This seems to confuse the algorithm and has already been manifested in the training set. But as shown in figure 5 the TPR of both gestures scroll-right and scroll-up is still above 80% so these events have a low occurrence rate. If we consider the gestures zoom-in and zoom-out the algorithm is confronted with big challenges distinguishing these actions, which is understandable because the movements are similar and only differ in the temporal direction. It could be observed that individual probands tend to perform the opposite gesture to reach the initial starting point for the gesture execution. For example, a slight zoom-in is performed before the zoom-out gesture. This is explainable because it seems natural for the human to further increase the range of motion to emphasize the executed gesture more, which in this case may imply performing a part of the opposite move in advance of the actual gesture.

Our results with the excerpt in table two demonstrate that our shown architecture provides the best accuracies among many different state-of-the-art models, demonstrating the robustness and efficiency of the approach. We hypothesise that the Inception and Xception models, as described in other research works [57], are efficient at extracting features and that the bigger number of identified features compared to the other models, might lead to better RNN classification results.

Considering the data scarcity problem mentioned in the related work section, our model and its architecture with the underlying dataset could serve as a model to realize further use cases of HCI where there is not enough video material of the individual hand gestures available. However, this possibility would have to be investigated in future research.

As shown in the theoretical background, the state-of-the-art in the field of classic CNNs is increasingly trying to develop lighter and less complex architectures, which also achieve very good classification results and can therefore be better used in real-world scenarios. The architecture we use corresponds exactly to this current trend, as we use a highly efficient model with the Inception-v3 model, which is also efficient in the use of computation resources [57]. Our quite competitively comparable results with this model suggest that the architecture we have shown can further complement the current state-of-the-art.

VI. CONCLUSION

In this work, we proposed a neural network with a hybrid architecture to classify hand gesture video sequences in a realistic environment with complex backgrounds and other factors that occur in real-world situations. This architecture consists of an RNN on top of a CNN which operates as a feature extractor providing 2D spatiotemporal feature maps as input for the RNN. We applied our proposed model on Depth_Camera_Dataset, which is a large dataset containing 6 different gestures where we achieved a sufficient final accuracy of 86.33% on our best fold but confirmed that the results are very robust through all folds with a 5-fold cross-validation. It can be observed, that our model achieves high recognition accuracies and maintains robustness against complex circumstances in the environment around the gesture, while still being low on computational costs.

With our proposed machine learning model, our goal was to address the research gap between realistic background environments and high accuracy in gesture recognition. In comparison to related works, we achieved a slight increase in the benchmark but if the averaged accuracy over all folds is considered we are 1.44% beneath the current benchmark stated in table 1. It has to be mentioned, that there are more gestures to classify in the approach of Benitez-Garcia et al. [34]. Nevertheless, we could prove with our proposed model that it is possible to classify the Depth_Camera_Dataset to a near benchmark accuracy of classifying hand gestures in a complex environment. The most misclassification occurs when the model is distinguishing the gestures scroll-right from scroll-up and zoom-in from zoom-out. The former is very likely to result from an unclean direction of the movement in advance of the actual gesture. The latter is very likely because of the similarity of the gestures performed. There are several ways to optimize our model in further iterations. With dynamic learning rate adaption and multiple training epochs, the feature extractor, as well as our RNN could be further optimized. Also, different optimizers can be used in the training and validation process.

Also, our model might be a solution to data scarcity, which is one of the problems of state-of-the-art deep learning approaches. Transfer learning, which refers to using a pre-trained model as a basis for further deep learning models to overcome the data problem, is a current trend in deep learning approaches [44]. To enable the use of gesture recognition in various different and individual HCI applications, our proposed architecture could act as a transfer learning model and address the problem of missing data in these.

A. LIMITATION AND FUTURE WORK

Although we focused on gestures in a natural, complex, and non-lab-like environment HCI in the real world often takes place in even more difficult situations. For instance, when the angle of the proband is considerably different from the frontal position or multiple people perform multiple gestures simultaneously. For these highly advanced challenges, our model may not be able to deliver sufficient results, especially when it comes to several gestures at the same time. It should also be mentioned that several more relevant hand gestures can be taken into consideration when expanding HCI to more sophisticated levels of interactions. In our approach we only focused on the RGB images, however, it makes sense to check if depth images bring better results. Moreover, it is possible to focus on different frame sequences, like including just the middle 10 or 20 frames of the 40 frame sequence or including all possible values for frame density and to estimate and evaluate the greatest cost-benefit ratio. In the case of the more sophisticated levels of interactions, more gestures can

be included, and the model could be further developed and optimized in this way.

To address the problem of performing opposite gestures to get to the starting point for gesture execution, the algorithm can be further trained to distinguish the crucial starting points of a hand gesture and the actual gesture to be classified. It may be valuable to observe the velocity of the gestures or to split the action into different segments and focus on the last/most significant one. In this case, a pre-classification could be made on which movement in the sequence is the most relevant movement to identify with the highest probability. Further research has to be done to better emphasize the micro-movements people tend to do before, while, and after the relevant gesture to address this topic. This challenge has also been mentioned and thoroughly discussed in [39].

We have tested our novel deep learning architecture in our work with a modern comprehensive dataset on dynamic hand gestures. Even though there are not many other datasets in this area with non-lab-like environments, this is a limitation, which is why we will evaluate the shown architecture with further datasets in future research and will also show further evaluation metrics.

If we take another look at the current trends in machine learning, we can see that the use of generative adversarial networks for data generation is increasing. Future research could therefore also investigate whether the relatively high error rate in the area of scroll-up and scroll-down movements could be compensated for by artificially generated additional videos for those cases. In general, with regard to the state-of-the-art in deep learning, it could be investigated whether our architecture can be used in the area of transfer learning, as mentioned earlier in our article, and whether artificially generated data is suitable for our approach.

Last but not least, we demonstrated a novel approach for the video classification task of hand gestures and were able to compare many state-of-the-art models and identify the most suitable combination. In further work, we will try to optimise this architecture by using additional datasets and applying hyper-parameter tuning and fine-tuning, for example.

ACKNOWLEDGMENT

(David Richard Tom Hax and Pascal Penava are co-first authors.)

REFERENCES

[1] G. R. S. Murthy and R. S. Jadon, "Hand gesture recognition using neural networks," in *Proc. IEEE 2nd Int. Advance Comput. Conf. (IACC)*, Patiala, India: IEEE, Feb. 2010, pp. 134–138. [Online]. Available: http://ieeexplore.ieee.org/document/5423024/

[2] R. P. Sharma and G. K. Verma, "Human computer interaction using hand gesture," *Proc. Comput. Sci.*, vol. 54, pp. 721–727, 2015. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S187705091501409X

[3] P.-J. Gonzalo and A. H.-T. Juan, "Control of home devices based on hand gestures," in *Proc. IEEE 5th Int. Conf. Consum. Electron. Berlin (ICCE-Berlin)*. Berlin, Germany: IEEE, Sep. 2015, pp. 510–514. [Online]. Available: http://ieeexplore.ieee.org/document/7391325/

[4] Y. Zhu and B. Yuan, "Real-time hand gesture recognition with Kinect for playing racing video games," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*. Beijing, China: IEEE, Jul. 2014, pp. 3240–3246. [Online]. Available: https://ieeexplore.ieee.org/document/6889481

[5] D. Yang, J.-K. Lim, and Y. Choi, "Early childhood education by hand gesture recognition using a smartphone based robot," in *Proc. 23rd IEEE Int. Symp. Robot Human Interact. Commun.* Edinburgh, U.K.: IEEE, Aug. 2014, pp. 987–992. [Online]. Available: http://ieeexplore.ieee.org/document/6926381/

[6] S. Mitra and T. Acharya, "Gesture recognition: A survey," *IEEE Trans. Syst., Man Cybern., C, Appl. Rev.*, vol. 37, no. 3, pp. 311–324, May 2007. [Online]. Available: http://ieeexplore.ieee.org/document/4154947/

[7] B. Peng and G. Qian, "Online gesture spotting from visual hull data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 6, pp. 1175–1188, Jun. 2011. [Online]. Available: http://ieeexplore.ieee.org/document/5639014/

[8] S. S. Rautaray and A. Agrawal, "Vision based hand gesture recognition for human computer interaction: A survey," *Artif. Intell. Rev.*, vol. 43, no. 1, pp. 1–54, Jan. 2015. [Online]. Available: http://link.springer.com/10.1007/s10462-012-9356-9

[9] V. I. Pavlovic, R. Sharma, and T. S. Huang, "Visual interpretation of hand gestures for human-computer interaction: A review," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 677–695, Jul. 1997. [Online]. Available: http://ieeexplore.ieee.org/document/598226/

[10] S. Sharma and S. Singh, "Vision-based hand gesture recognition using deep learning for the interpretation of sign language," *Expert Syst. Appl.*, vol. 182, Nov. 2021, Art. no. 115657. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417421010484

[11] T. Kapuscinski and K. Inglot, "Vision-based gesture modeling for signed expressions recognition," *Proc. Comput. Sci.*, vol. 207, pp. 1007–1016, 2022. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1877050922010389

[12] D. Tasmere, B. Ahmed, and S. R. Das, "Real time hand gesture recognition in depth image using CNN," *Int. J. Comput. Appl.*, vol. 174, no. 16, pp. 28–32, Jan. 2021. [Online]. Available: http://www.ijcaonline.org/archives/volume174/number16/tasmere-2021-ijca-921040.pdf

[13] S. B. Abdullahi and K. Chamnongthai, "American sign language words recognition of skeletal videos using processed video driven multi-stacked deep LSTM," *Sensors*, vol. 22, no. 4, p. 1406, Feb. 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/4/1406

[14] S. B. Abdullahi and K. Chamnongthai, "American sign language words recognition using spatio-temporal prosodic and angle features: A sequential learning approach," *IEEE Access*, vol. 10, pp. 15911–15923, 2022.

[15] Y. Shi, Y. Li, X. Fu, M. Kaibin, and M. Qiguang, "Review of dynamic gesture recognition," *Virtual Reality Intell. Hardware*, vol. 3, no. 3, pp. 183–206, Jun. 2021. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2096579621000279

[16] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.

[17] Y. LeCun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, and L. Jackel, "Handwritten digit recognition with a back-propagation network," in *Advances in Neural Information Processing Systems*, D. Touretzky, Ed., vol. 2. Burlington, MA, USA: Morgan-Kaufmann, 1989, pp. 396–404.

[18] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, "A survey of convolutional neural networks: Analysis, applications, and prospects," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 12, pp. 6999–7019, Dec. 2022.

[19] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai, and T. Chen, "Recent advances in convolutional neural networks," *Pattern Recognit.*, vol. 77, pp. 354–377, May 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0031320317304120

[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, Eds., vol. 25. Red Hook, NY, USA: Curran Associates, 2012.

[21] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

[22] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[24] A. Rehman and S. B. Belhaouari, "Deep learning for video classification: A review," *TechRxiv*, pp. 1–12, Aug. 2021.

[25] V. Adithya and R. Rajesh, "A deep convolutional neural network approach for static hand gesture recognition," *Proc. Comput. Sci.*, vol. 171, pp. 2353–2361, Jan. 2020. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1877050920312473

[26] C. Xia, A. Saito, and Y. Sugiura, "Using the virtual data-driven measurement to support the prototyping of hand gesture recognition interface with distance sensor," *Sens. Actuators A, Phys.*, vol. 338, May 2022, Art. no. 113463. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0924424722001017

[27] T. L. Dang, S. D. Tran, T. H. Nguyen, S. Kim, and N. Monet, "An improved hand gesture recognition system using keypoints and hand bounding boxes," *Array*, vol. 16, Dec. 2022, Art. no. 100251. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2590005622000844

[28] A. Kasapbaşi, A. E. A. Elbushra, A.-H. Omar, and A. Yilmaz, "DeepASLR: A CNN based human computer interface for American sign language recognition for hearing-impaired individuals," *Comput. Methods Programs Biomed. Update*, vol. 2, 2022, Art. no. 100048. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2666990021000471

[29] M. Simão, P. Neto, and O. Gibaru, "EMG-based online classification of gestures with recurrent neural networks," *Pattern Recognit. Lett.*, vol. 128, pp. 45–51, Dec. 2019. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0167865519302089

[30] S. Hussain, R. Saxena, X. Han, J. A. Khan, and H. Shin, "Hand gesture recognition using deep learning," in *Proc. Int. SoC Design Conf. (ISOCC)*, Seoul, South Korea, 2017, pp. 48–49.

[31] S. S. Rautaray and A. Agrawal, "Real time gesture recognition system for interaction in dynamic environment," *Proc. Technol.*, vol. 4, pp. 595–599, 2012. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S221201731200374X

[32] L. Wang, "Temporal segment networks for action recognition in videos," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 11, pp. 2740–2755, Nov. 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8454294/

[33] J. Lin, C. Gan, K. Wang, and S. Han, "TSM: Temporal shift module for efficient and scalable video understanding on edge devices," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 5, pp. 2760–2774, May 2022. [Online]. Available: https://ieeexplore.ieee.org/document/9219141/

[34] G. Benitez-Garcia, L. Prudente-Tixteco, L. C. Castro-Madrid, R. Toscano-Medina, J. Olivares-Mercado, G. Sanchez-Perez, and L. J. G. Villalba, "Improving real-time hand gesture recognition with semantic segmentation," *Sensors*, vol. 21, no. 2, p. 356, Jan. 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/2/356

[35] C. R. Naguri and R. C. Bunescu, "Recognition of dynamic hand gestures from 3D motion data using LSTM and CNN architectures," in *Proc. 16th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*. Cancun, Mexico: IEEE, Dec. 2017, pp. 1130–1133. [Online]. Available: http://ieeexplore.ieee.org/document/8260797/

[36] P. H. Nguyen and T. N. Luong, "Two-stream convolutional network for dynamic hand gesture recognition using convolutional long short-term memory networks," *Vietnam J. Sci. Technol.*, vol. 58, no. 4, pp. 514–523, Jul. 2020. [Online]. Available: http://vjs.ac.vn/index.php/jst/article/view/14742

[37] S. Ji, W. Xu, M. Yang, and K. Yu, "3D convolutional neural networks for human action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 221–231, Jan. 2013. [Online]. Available: http://ieeexplore.ieee.org/document/6165309/

[38] W. Zhang and J. Wang, "Dynamic hand gesture recognition based on 3D convolutional neural network models," in *Proc. IEEE 16th Int. Conf. Netw., Sens. Control (ICNSC)*, Banff, AB, Canada, May 2019, pp. 224–229. [Online]. Available: https://ieeexplore.ieee.org/document/8743159/

[39] G. Zhu, L. Zhang, P. Shen, and J. Song, "Multimodal gesture recognition using 3-D convolution and convolutional LSTM," *IEEE Access*, vol. 5, pp. 4517–4524, 2017. [Online]. Available: http://ieeexplore.ieee.org/document/7880648/

[40] S. Skaria, A. Al-Hourani, and R. J. Evans, "Deep-learning methods for hand-gesture recognition using ultra-wideband radar," *IEEE Access*, vol. 8, pp. 203580–203590, 2020.

[41] G. Park, V. K. Chandrasegar, and J. Koh, "Accuracy enhancement of hand gesture recognition using CNN," *IEEE Access*, vol. 11, pp. 26496–26501, 2023.

[42] L. Zhang, G. Zhu, P. Shen, J. Song, S. A. Shah, and M. Bennamoun, "Learning spatiotemporal features using 3DCNN and convolutional LSTM for gesture recognition," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCVW)*, Venice, Italy, Oct. 2017, pp. 3120–3128. [Online]. Available: https://ieeexplore.ieee.org/document/8265580/

[43] Q. Miao, Y. Li, W. Ouyang, Z. Ma, X. Xu, W. Shi, and X. Cao, "Multimodal gesture recognition based on the ResC3D network," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCVW)*, Venice, Italy, Oct. 2017, pp. 3047–3055. [Online]. Available: http://ieeexplore.ieee.org/document/8265571/

[44] L. Alzubaidi, J. Bai, A. Al-Sabaawi, J. Santamaría, A. S. Albahri, B. S. N. Al-Dabbagh, M. A. Fadhel, M. Manoufali, J. Zhang, A. H. Al-Timemy, Y. Duan, A. Abdullah, L. Farhan, Y. Lu, A. Gupta, F. Albu, A. Abbosh, and Y. Gu, "A survey on deep learning tools dealing with data scarcity: Definitions, challenges, solutions, tips, and applications," *J. Big Data*, vol. 10, no. 1, p. 46, Apr. 2023, doi: 10.1186/s40537-023-00727-2.

[45] V. K. Kurmi, V. Bajaj, B. N. Patro, K. S. Venkatesh, V. P. Namboodiri, and P. Jyothi, "Collaborative learning to generate audio-video jointly," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2021, pp. 4180–4184.

[46] N. Aldausari, A. Sowmya, N. Marcus, and G. Mohammadi, "Video generative adversarial networks: A review," *ACM Comput. Surveys*, vol. 55, no. 2, pp. 1–25, Jan. 2022, doi: 10.1145/3487891.

[47] X. Shen and I. Stamos, "SimCrossTrans: A simple cross-modality transfer learning for object detection with ConvNets or vision transformers," 2022, *arXiv:2203.10456*.

[48] W. Ye, J. Cheng, F. Yang, and Y. Xu, "Two-stream convolutional network for improving activity recognition using convolutional long short-term memory networks," *IEEE Access*, vol. 7, pp. 67772–67780, 2019.

[49] S. B. Abdullahi, Z. A. Bature, L. A. Gabralla, and H. Chiroma, "Lie recognition with multi-modal spatial–temporal state transition patterns based on hybrid convolutional neural network–bidirectional long short-term memory," *Brain Sci.*, vol. 13, no. 4, p. 555, Mar. 2023. [Online]. Available: https://www.mdpi.com/2076-3425/13/4/555

[50] S. B. Abdullahi, Z. A. Bature, P. Chophuk, and A. Muhammad, "Sequence-wise multimodal biometric fingerprint and finger-vein recognition network (STMFPFV-Net)," *Intell. Syst. Appl.*, vol. 19, Sep. 2023, Art. no. 200256. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2667305323000819

[51] C.-Y. Ma, M.-H. Chen, Z. Kira, and G. AlRegib, "TS-LSTM and temporal-inception: Exploiting spatiotemporal dynamics for activity recognition," *Signal Process., Image Commun.*, vol. 71, pp. 76–87, Feb. 2019, doi: 10.1016/j.image.2018.09.003.

[52] S. Grossberg, "Recurrent neural networks," *Scholarpedia*, vol. 8, no. 2, 2013, Art. no. 1888.

[53] C. L. Giles, G. M. Kuhn, and R. J. Williams, "Dynamic recurrent neural networks: Theory and applications," *IEEE Trans. Neural Netw.*, vol. 5, no. 2, pp. 153–156, Mar. 1994.

[54] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997. [Online]. Available: https://direct.mit.edu/neco/article/9/8/1735-1780/6109

[55] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *Proc. Int. Conf. Eng. Technol. (ICET)*, Antalya, Turkey, Aug. 2017, pp. 1–6.

[56] Y. Tu, X. Zhang, B. Liu, and C. Yan, "Video description with spatial–temporal attention," in *Proc. 25th ACM Int. Conf. Multimedia*. USA: ACM, Oct. 2017, pp. 1014–1022. [Online]. Available: https://dl.acm.org/doi/10.1145/3123266.3123354

[57] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 2818–2826.

[58] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, "Convolutional neural networks: An overview and application in radiology," *Insights Imag.*, vol. 9, no. 4, pp. 611–629, Aug. 2018. [Online]. Available: https://insightsimaging.springeropen.com/articles/10.1007/s13244-018-0639-9

[59] S. Jeeru, A. K. Sivapuram, D. G. León, J. Gröli, S. R. Yeduri, and L. R. Cenkeramaddi, "Depth camera based dataset of hand gestures," *Data Brief*, vol. 45, Dec. 2022, Art. no. 108659. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352340922008642

DAVID RICHARD TOM HAX received the B.A. degree in industrial engineering in Bayreuth, in 2021. He is currently pursuing the M.Sc. degree in digitalization and entrepreneurship with the University of Bayreuth, Germany. His research interests include software engineering, data analysis, and machine learning.

PASCAL PENAVA (Member, IEEE) received the B.S. degree in information systems from Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany, in 2021, and the M.S. degree in digitalization and entrepreneurship from the University of Bayreuth, Bayreuth, Germany, in 2023, where he is currently pursuing the Ph.D. degree with the Department of Information Systems and Data Science. His research interests include the development of EEG-based BCIs and machine learning based analyses of time-series data.

SAMIRA KRODEL received the B.A. degree in media and marketing management from the RFH—University of Applied Science, Cologne, Germany, in 2021. She is currently pursuing the M.Sc. degree in digitalization and entrepreneurship with the University of Bayreuth, Germany. Her research interests include the IoT, process mining, and machine learning.

LILIYA RAZOVA received the B.A. degree in international business and economics from the Schmalkalden University of Applied Sciences, in 2021. She is currently pursuing the M.Sc. degree in digitalization and entrepreneurship with the University of Bayreuth. Her current research interests include human–computer interaction, artificial intelligence, and big data.

RICARDO BUETTNER (Senior Member, IEEE) received the Dipl.-Inf. degree in computer science and the Dipl.-Wirtsch.-Ing. degree in industrial engineering and management from the Ilmenau University of Technology, Germany, the Dipl.-Kfm. degree in business administration from the University of Hagen, Germany, the Ph.D. degree in information systems from the University of Hohenheim, Germany, and the Habilitation (venia legendi) degree in information systems from the University of Trier, Germany. He is currently a Chaired Professor of information systems and data science with the University of Bayreuth, Germany. He has published over 140 peer-reviewed articles, including articles in *Electronic Markets*, *AIS Transactions on Human–Computer Interaction*, *Personality and Individual Differences*, *European Journal of Psychological Assessment*, *PLOS One*, and IEEE Access. He received 17 international best paper, the best reviewer, and the service awards and award nominations, including best paper awards by *AIS Transactions on Human–Computer Interaction*, *Electronic Markets*, and HICSS, for his work.