FPVRGame: Deep Learning for Hand Pose Recognition in Real-Time Using Low-End HMD

Eder de Oliveira[1][0000−0002−6437−4883], Esteban Walter Gonzalez Clua[1], Cristina Nader Vasconcelos[1], Bruno Augusto Dorta Marques[1][0000−0003−4999−9436], Daniela Gorski Trevisan[1], and Luciana Cardoso de Castro Salgado[1]

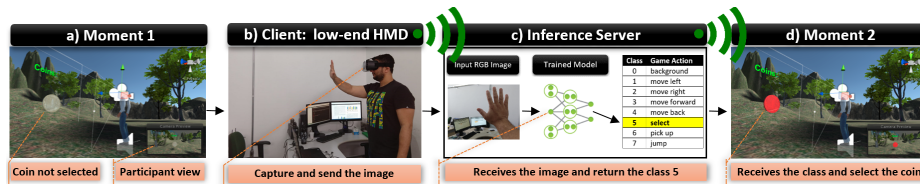Fluminense Federal University, Niteri-RJ, Brazil
eder_oliveira@id.uff.br

Fig. 1. The FPVRGame is a virtual reality environment developed to simulate a first-person view game which allows the usage of the bare hand to control a character. This figure shows the action of selecting a coin and the process involved: capture of the image, the inference of the convolutional neural network and the label send for the execution of the action in the game.

Abstract. Head Mounted Display (HMD) became a popular device, drastically increasing the usage of Virtual, Mixed, and Augmented Reality. While the systems' visual resources are accurate and immersive, precise interfaces require depth cameras or special joysticks, requiring either complex devices or not following the natural body expression. This work presents an approach for the usage of bare hands to control an immersive game from an egocentric perspective and built from a proposed case study methodology. We used a DenseNet Convolutional Neural Network (CNN) architecture to perform the recognition in real-time, from both indoor and outdoor environments, not requiring any image segmentation process. Our research also generated a vocabulary, considering users' preferences, seeking a set of natural and comfortable hand poses and evaluated users' satisfaction and performance for an entertainment setup. Our recognition model achieved an accuracy of 97.89%. The user's studies show that our method outperforms the classical controllers in regards to natural interactions. We demonstrate our results using commercial low-end HMD's and compare our solution with state-of-the-art methods.

Keywords: Hand Poses Recognition · Convolutional Neural Network · Deep Learning · Virtual Reality · User Interfaces.

1   Introduction

Head-Mounted Displays (HMDs) are becoming popular and accessible, leveraging Virtual, Mixed and Augmented Reality applications to a new level of consumption. A considerable amount of these market is strongly attached to low-end devices, based on smartphones as displays and computing hardware, enhancing the possibility of users interacting with virtual worlds anytime, anywhere, whether to watch a movie, work or play games [14].

High-end devices, such as Oculus Rift or HTC Vive, provide sophisticated interfaces controllers and tracking systems, allowing powerful and complex interactions with the virtual environment [16]. Due to the lack of these components, mobile-based systems must be projected to be more straightforward and in many cases less immersive solutions.

Visual immersion achieved by the HMDs can generate high interaction expectation among the users. It is common to observe, at interaction time, that the user makes undesired body and hand gestures, moved by a natural body instinct [22].

This body interaction cannot be implemented with regular HMD joysticks and controllers. Thus, research has been conducted to offer a more natural engagement to users. For instance, several body and hand gestures recognition solutions are being presented in the last years, some of them using very different approaches than traditional joysticks: heart rate monitors [28], Coulomb friction model [9], acoustic resonance analysis [32] and even clothing that restricts joint movements [1].

The usage of bare hands is what seems to be the most natural and immersive solution, and some researchers are working on this [24, 26]. In this sense, precise and comfortable solutions still require some dedicated hardware, such as depth cameras, structured light-based systems, and even Inertial measurement unit (IMU) based hardware.

In this work, we present an interaction solution based on bare hand interactions, which perform the real-time recognition with 98% accuracy and can be executed indoor and outdoor using ordinary cameras as input devices through low-end platforms, such as smartphones. Our solution is constrained for egocentric point of view and for a specific set of hand poses. It can be easily Incorporated in any application and presents good performance, suitable for low-end VR devices.

The hand pose recognition is a Machine Learning problem modeled as a pattern recognition task. Given that the state-of-the-art algorithms for pattern recognition in images are based on Convolutional Neural Networks (CNNs) [25], we have chosen to use this approach in our work. We adopted three CNNs based architectures (GoogLeNet, Resnet, and DenseNet), with which we conducted several training sessions using two different datasets until we get to the model used in our solution. One of the data sets was created specifically for this work, containing approximately 59,000 Red-Green-Blue (RGB) images of the hand in an egocentric vision [18, 30, 33].

To simulate a First-Person Vision (FPV) navigation system, we developed the FPVRGame (Figure 1), a Virtual Reality (VR) environment. The FPVRGame design and evaluation process involved three empirical studies. In Study One we specified a preliminary vocabulary containing the hand poses considered more intuitive to represent the actions of a character from an FPV perspective. In Study Two we evaluated the hand poses existing in the preliminary vocabulary and built a new and more comfortable

vocabulary, capturing the images required to create the dataset. Finally, in Study Three we validated our results using a low-end HMD and a simple VR environment.

The main contributions can be summarized as:

1. Implementation of a CNN-based method for recognition of user's hand poses captured from an FPV navigation system perspective in any environment (indoor and outdoor) and without any background or lighting constraints;
2. Creation of an open dataset with approximately 59,000 images of hand poses from a FPV navigation system perspective;
3. Generation and assessment of a hand pose vocabulary by using the Wizard of Oz method;
4. Empirical evaluation of user's experience and performance, using the proposed hand posture recognition in comparison to the main interfaces available for HMDs, in both low and high-end systems;

In addition, while we attempted to solve FPV systems, our solution can be trivially extended to any other interaction paradigm, depending only on providing a new image dataset.

The remainder of this paper is organized as follows: Section 2 describes the related works. Section 3 presents our CNN based solution for hand posture recognition and describes our Dataset, which we defined as public. Section 4 presents the FPVRGame design and evaluation process, i.e, the hand poses vocabulary construction process; the comparison between the accuracy achieved by our method with other interfaces. Finally, Section 6, present the conclusions of our work.

2 Related Work

Although there are several precise and functional interface devices for HMD's, we claim that the usage of bare hands is the most natural, intuitive and immersive [24].

Among several works, Son and Choi [27] proposed a hand pose detection approach that is capable for classifications based on raw RGB images. Their method recognizes three distinct hand poses employing a faster R-CNN, capable of identifying the region of interest and classifying one of the three possible poses. The dataset for training the network requires additional annotation for the palm position and fingertip. Their method aims to estimate the bounding box of the hands and identify which hand is in the camera field of view (left or right). In our paper, we consider that the game controller should work similarly for both hands, allowing left-handed and right-handed users to share the same experiences.

Hand gestures are classified in static and dynamic gestures [24]. Different datasets containing images in egocentric vision are available, such as [17, 18, 30, 33]. However, most of them contain dynamic gestures. Dynamic gestures recognition usually exploits spatiotemporal features extracted from video sequences. The evaluation of this type of method usually employs a sliding window of 16 frames or more [4], which introduces a significant input lag corresponding to the time between the 16 frames and the gesture recognition. Our work recognizes static gestures through a CNN, classifying hand poses from a single frame and allowing real-time recognition without the before-mentioned

input lag penalty. Furthermore, tests performed in an egocentric dataset [33] shows that our method has a competitive accuracy when taking account gestures similar to our hand poses vocabulary. We created a specific dataset with a limited number of poses but with higher accuracy.

Depth cameras (RGB-D) have the capacity of delivering depth information for each pixel, making possible the use of different techniques for geometry reconstruction and estimation of inverse kinematics bones positioning [24]. In indoor and controlled environments the depth cameras perform very well and are being vastly used. However, depth sensors can generate noisy depth maps, presenting some limitations: restricted field of view and range, near-infrared interference (such as light solar) and non-Lambertian reflections, and thus cannot acquire accurate measurements in outdoor environments [23]. These issues become more critical when the cameras are not fixed.

Yousefi et al. [31] presented a gesture-based interaction system for immersive systems. Their solution makes use of the smartphone camera to recognize the gesture performed by the user's hands. The recognition process is based on matching a camera image with an image in a gesture dataset. The gesture dataset contains images of a user's hand performing one of the 4 available gestures. The images were recorded for both left and right hands under different rotations and a chroma key screen was employed to remove the background pixels. The construction of the dataset is labor-intensive, requiring the manual annotation of 19 joint points for each image in the dataset. At runtime, a preprocessing step is necessary to ensure that only the relevant data is fed to the gesture recognition system. This stage consists in segmenting the hand from the background and crop the image in the region of interest. The gesture recognition system performs a similarity analysis based on L1 and L2 norms to match the camera image with one of the dataset images. A selective search strategy based on the previous camera frame is used to reduce the search domain and efficiently recognize the gesture in real time.

The previous method requires extensive labor-intensive adjustments through the manual annotation of 19 joint points for each sample on the dataset creation process. Furthermore, the segmentation process requires manual adjustments based on the user's environment. As opposed to their approach, our method employs a dataset creation process that automatically annotates the images while the user is experimenting with the application. Furthermore, our recognition system works on raw input images and does not require background extraction, chroma key, and lighting adjustments.

3   Hand Pose Recognition Solution

The hand pose recognition is the process of classifying poses of the user's hands in a given input image [24]. We use the recognized pose to perform an action in an interactive game. Since we are using the player's hands like a game controller, the process must be robust to recognize the player's hands in multiple scenarios and different environments. It is important to have a consistent result in the hand recognition since a wrong classification would result in an involuntary movement in the game, potentially harming the user's experience.

The hand pose recognition is a Machine Learning problem modeled as a pattern recognition task. Given that the state-of-the-art algorithms for pattern recognition in images are based on Convolutional Neural Networks [25], we choose to use this approach in our work.

Convolutional Neural Networks are learning algorithms that require a two-step process. A compute-intensive training step executed once, and a fast inference step, performed in the application runtime [24, 26]. Considering that our method aims to be executed in a mobile environment, the hand pose recognition must be executed in interactive time even on low spec mobile devices. This requirement makes the Deep Neural Network a suitable approach for our purposes.

The input of our method is an RGB image containing the user's hands (Figure 4). The output is a probability distribution of the k possible classes. The classes are composed of the specified vocabulary described in Section 4, and an additional Background Class, that represents the absence of the user's hands in the input image.

We adopted three off-the-shelf CNN architectures for the hand pose recognition: GoogLeNet [29], Resnet 50 [8], and DenseNet [10].

3.1 Datasets

We use two datasets in the CNNs training.

Dataset 1 (DS1): We use a pre-training dataset, containing 1,233,067 samples, taken from publicly available sign language datasets [15]. A sample in the dataset consists of a tuple (image, label) where the image portrays an interpreter performing a sign language gesture. Even though this dataset does not contain the correct label for our recognition system, the images of the dataset are employed to precondition the CNN to recognize features related to human hands under different poses.

Dataset 2 (DS2): We created a second dataset specially tailored for our recognition system. The dataset consists of 58,868 samples captured in an indoor environment, comprising images of fifteen people (eleven men and four women). To improve our detection for both right-handed and left-handed users, we applied a mirror transformation in the images. The images were manually annotated with one of the seven classes that represent the possible actions of the game, or a class representing the background (Figure 4).

A preprocessing step in both datasets ensures that the images have the same dimensions (256 x 256 pixels). A bicubic transformation was performed to resize images to adequate dimensions.

3.2 CNNs Training Results

Training a CNN from scratch requires a significant amount of labeled data; therefore, we trained three CNNs through a process known as fine-tuning. We loaded a pre-trained model with weights adjusted to the ImageNet dataset [6]; then we fine-tuned our network to the DS2 dataset. Figure 2a shows a summary of the results obtained during the training.

The ResNet architecture obtained a not satisfactory result, with a mean accuracy of 85.46% (test) and 79.25% (validation), with a mean error of 0.854%. Aiming to improve

Fig. 2. a) Best results achieved during the training of CNNs, using the two data sets DS1 and DS2. b) DenseNet's confusion matrix. c) GoogleNet's confusion matrix.

	GoogleNet (DS1 and DS2)	DenseNet - 121 (DS2)	GoogleNet (DS2)	DenseNet - 121 (DS1 and DS2)	Resnet - 50 (DS1 and DS2)	Resnet - 50 (DS2)
Fold - 1	98,50	97,80	98,60	96,25	89,00	73,34
Fold - 2	98,28	98,93	99,33	99,04	96,38	89,02
Fold - 3	97,13	97,17	94,46	97,09	91,41	83,78
Fold - 4	99,90	99,86	99,82	99,99	99,92	98,07
Fold - 5	96,44	95,68	95,13	93,76	88,44	83,07
Mean validation	96,34	97,19	96,27	95,73	89,78	79,25
Mean test	98,05	97,89	97,47	97,23	93,03	85,46
Mean loss	0,144	0,096	0,162	0,182	0,406	0,854

b) DenseNet's confusion matrix

	0	1	2	3	4	5	6	7
0	96,6%	0,7%	0,6%	1,5%	0,3%	0,2%	0,0%	0,2%
1	0,6%	98,7%	0,0%	0,1%	0,5%	0,0%	0,0%	0,1%
2	0,7%	0,1%	98,6%	0,0%	0,4%	0,0%	0,0%	0,2%
3	2,0%	0,0%	0,0%	96,3%	0,1%	0,0%	0,0%	1,5%
4	0,1%	0,3%	0,2%	0,2%	98,5%	0,0%	0,1%	0,7%
5	0,0%	0,0%	0,0%	0,2%	0,0%	99,7%	0,1%	0,0%
6	0,0%	0,1%	0,1%	0,2%	0,3%	0,3%	99,0%	0,1%
7	0,1%	0,4%	0,1%	2,8%	0,3%	0,0%	0,0%	96,3%

c) GoogleNet's confusion matrix

	0	1	2	3	4	5	6	7
0	98,9%	0,2%	0,2%	0,5%	0,1%	0,0%	0,0%	0,0%
1	0,9%	97,5%	0,4%	0,1%	0,7%	0,0%	0,0%	0,4%
2	1,1%	0,0%	97,4%	0,1%	0,6%	0,0%	0,1%	0,6%
3	0,8%	0,0%	0,0%	98,0%	0,2%	0,0%	0,0%	1,0%
4	0,2%	0,6%	0,7%	1,2%	97,0%	0,0%	0,1%	0,3%
5	0,1%	0,0%	0,0%	0,2%	0,0%	99,7%	0,1%	0,0%
6	0,0%	0,1%	0,0%	0,3%	0,1%	0,1%	99,2%	0,2%
7	0,2%	0,3%	0,0%	5,2%	0,0%	0,0%	0,1%	94,2%

the accuracy of the classification, we performed a second experiment that exploits the features learned from the DS1 dataset.

The second experiment consists in, first, fine-tuning the CNN from the ImageNet dataset to the DS1 dataset, then fine-tuning the resulting model to the DS2 dataset. The result of this experiment for the ResNet architecture results in a mean accuracy of 93.03% (test) and 89,79% (validation) with a mean error of 0.406%. When compared to the first approach, we obtained a significant improvement in the mean accuracy of 8.85% (test) and 13.28% (validation) with a mean error of 0.406%.

While the ResNet highly benefit from the second approach, the GoogleNet and DenseNet obtained only a slight change in the mean accuracy when compared to the first experiment. The GoogLeNet test accuracy improved by a small margin (from 97.47% to 98.05%) while the DenseNet present a small decrease in test accuracy (from 97.89% to 97.23%).

Overall, the GoogLeNet obtained the highest mean accuracy of 98.05% on tests while the DenseNet, trained with the first approach, achieved the lowest mean error on the validation and a better mean accuracy distribution across the different classes. Furthermore, the training process was facilitated due to the usage of the first approach that does not require the finetuning to the DS1 dataset. The mean accuracy across multiple classes can be observed in the confusion matrix depicted in Figure 2b and c. In our hand pose recognition system, we choose to use the DenseNet implementation due to less associated error across multiple classes and the relative uncomplicated single training process.

For validation purposes, we tested our trained CNN against a public available ego-centric benchmark dataset, EgoGesture [33]. The dataset contains 2,081 RGB-D videos, 24,161 gesture samples totaling 2,953,224 frames. There are 83 classes of gestures, mainly focused on interaction with wearable devices. Because it is a data set different from ours, we have chosen a subset of gestures that are similar to the poses of our vocabulary. The mapping between the EgoGesture classes and our vocabulary classes is shown in the Table 1. Our GloogleNet model, even though have never been trained with any of the images in the EgoCentric dataset achieved an accuracy of 64.3%. This result is superior to the mean average accuracy of 62.5% in the VGG16 model presented

by Cao et. al [4]. On one hand, we could improve our accuracy results by considering spatiotemporal strategies like appending an Long Short-Term Memory (LSTM) network to the output of our last fully connected layer, on the other hand, this introduction would increase the input lag in our application, thus making the model inadequate for VR applications.

Table 1. Gesture mapping our vocabulary of hand poses to EgoGesture [33] gestures.

Our Vocabulary		EgoGesture	
Class	Gesture	Class	Gesture
1 move left		66 thumb toward left	
2 move right		65 thumb toward right	
3 move forward		83 move fingers forward	
4 move back		67 thumbs backward	
5 select		29 number 5	

Most of the errors associated with our model are the misclassification of gestures as background (class 0), as shown in the confusion matrix depicted in Figure 3a. This error is associated with the different nature of our training dataset and the tested dataset (video sequences in the EgoGesture vs single frames in our dataset). Frames at the two extremes (beginning and ending) of a video sequence in the EgoGesture dataset contains no identifiable gestures, for example, partially visible hands or the very beginning/ending of a gesture. To test this hypothesis, we tested our model by dropping the few beginning and ending frames of the video sequences. With 4 and 8 frames dropped, the model achieved a notable higher mean accuracy of 76.17% and 78.81%. The improvement in the model's accuracy and the confusion matrix (Figure 3b and c) confirm our hypothesis.
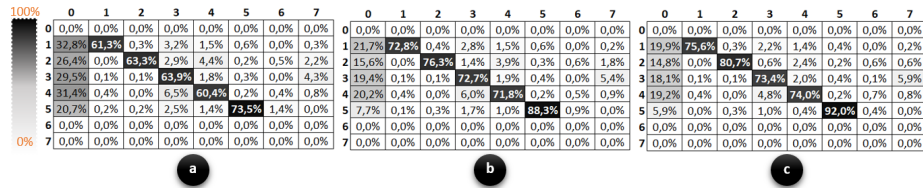
a) 0 frames dropped

	0	1	2	3	4	5	6	7
0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
1	32,8%	61,3%	0,3%	3,2%	1,5%	0,6%	0,0%	0,3%
2	26,4%	0,0%	63,3%	2,9%	4,4%	0,2%	0,5%	2,2%
3	29,5%	0,1%	0,1%	63,9%	1,8%	0,3%	0,0%	4,3%
4	31,4%	0,4%	0,0%	6,5%	60,4%	0,2%	0,4%	0,8%
5	20,7%	0,2%	0,2%	2,5%	1,4%	73,5%	1,4%	0,0%
6	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
7	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%

b) 4 frames dropped

	0	1	2	3	4	5	6	7
0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
1	21,7%	72,8%	0,4%	2,8%	1,5%	0,6%	0,0%	0,2%
2	15,6%	0,0%	76,3%	1,4%	3,9%	0,3%	0,6%	1,8%
3	19,4%	0,1%	0,1%	72,7%	1,9%	0,4%	0,0%	5,4%
4	20,2%	0,4%	0,0%	6,0%	71,8%	0,2%	0,5%	0,9%
5	7,7%	0,1%	0,3%	1,7%	1,0%	88,3%	0,9%	0,0%
6	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
7	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%

c) 8 frames dropped

	0	1	2	3	4	5	6	7
0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
1	19,9%	75,6%	0,3%	2,2%	1,4%	0,4%	0,0%	0,2%
2	14,8%	0,0%	80,7%	0,6%	2,4%	0,2%	0,6%	0,6%
3	18,1%	0,1%	0,1%	73,4%	2,0%	0,4%	0,1%	5,9%
4	19,2%	0,4%	0,1%	4,8%	74,0%	0,2%	0,7%	0,8%
5	5,9%	0,0%	0,3%	1,0%	0,4%	92,0%	0,4%	0,0%
6	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
7	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%

Fig. 3. Confusion Matrix: a) 0 frames dropped, b) 4 frames dropped, c) 8 frames dropped. As for classes 6 and 7 we did not find similar gestures. Class 0 does not make its inference because it does not appear in the label of the base EgoCentric.

The CNN training was executed on a DGX-1 machine with the following specification: Intel Xeon E5-2698 v4 2.2Ghz, 512 GB DDR 4, 8 x NVIDIA P100 Graphics Processing Unit (GPU). All the networks are trained using 4 GPUs adopting Stochastic Gradient Descent (SGD) as our solver. We applied 5-fold cross-validation to our model, splitting the dataset into 5 distinct folds [7]. We run the tests for 30 epochs with batch size 96. The learning rate is set initially to 0.01 with the exponential decay (gamma=0.95).

3.3   Inference Server Implementation

The recognition system is based on a client/server system. The FPVRGame, running on a smartphone Moto X4, act as a client that captures the HMD camera image and send them to the inference server application through a TCP/IP protocol. The server feeds the CNN with the received image and carries the recognized hand pose identification back to the FPVRGame. The inference server application (Figure 1c) was implemented with Python 3.6 using the Caffe framework [11] within an Intel® Core™ i7-7700HQ CPU @ 2.80GHz, 16GB RAM and NVIDIA GeForce® GTX 1050 Ti machine and running the DenseNet model performs the inference with an average of 28 milliseconds. Thus, the inference process can run in real-time (35 fps) on any modern GPU enabled devices. Alternatively, it is possible to use our CNN model with third-party inference engines such as NVIDIA TensorRT [20], Clipper [5], and DeepDetect [12].

4   FPVRGame - Design and Evaluation Process

FPVRGame is a VR environment developed to simulate an FPV game. Its primary objective is to navigate through the scenario and collect as many coins as possible. For this, it captures the image of the players hand and forwards the images to the Inference Server, see Figure 1. As the camera is the single input device, the player has to position his hand inside the camera field. This limitation may require considerable physical effort, and if the hand pose is not the most suitable, may cause pain.
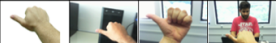
Fig. 4. Vocabulary of hand poses used for the player.

Figure 4 shows the vocabulary provided by FPVRGame, built based on two empirical studies, that offer a set of intuitive and comfortable hand poses. Study One included identifying users' preference for the most appropriate hand poses. This study was

attended by 173 people, being 105 males and 68 females, aged between 18 and 39 years (M = 25, SD = 4.06), 92.5% and 7.5% left-handers. What resulted in preliminary vocabulary. In Study Two by using the Wizard of Oz method [13], we evaluate the existing hand poses in the preliminary vocabulary and construct a new, more comfortable vocabulary, capturing the images needed to create the data set. This study was attended by 15 people, 11 males, and 4 females, aged 18 to 43 years (M = 26.46, SD = 6.94), all right-handed.
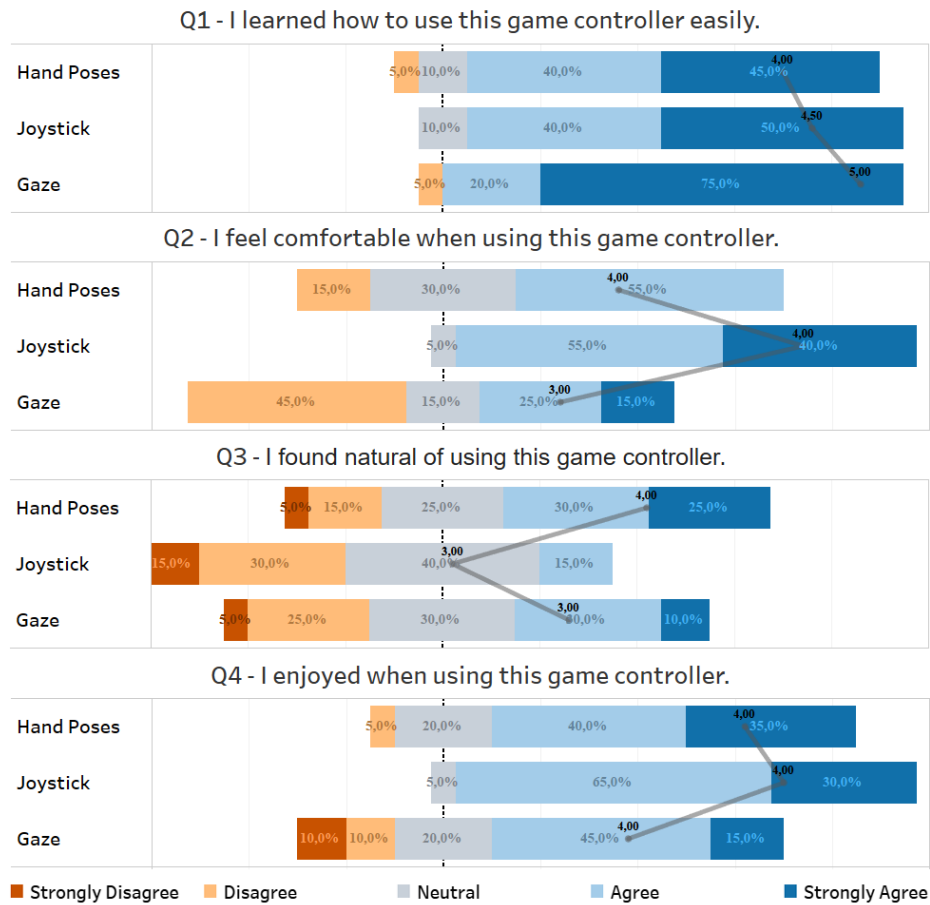
Fig. 5. Perceptions of the participants considering all game controllers.

Study Three is aimed at assessing the participant's experience and performance when using our hand poses recognition solution. The primary measures used in this study were the participant's feelings (Easy to learn, Comfort, Natural and Enjoyment) when using

different game controllers (joystick, gaze and hand poses) to control a virtual character in the FPVRGame.

Figure 5 shows a summary of the results for questions Q1 - Easy to learn, Q2 - Comfort, Q3 - Natural, and Q4 - Enjoyment, for each game controller. All statistical analyses were performed using IBM SPSS [1] with ($\alpha = 0.05$).

For the item "Q1 - Easy to learn," we find none significant difference between the game controllers (Friedman $X^2_{(2)} = 4.480$, $p = 0.106$) .

For the item "Q2 - Comfort," there was a significant difference between gaze and joystick controllers (Friedman $X^2_{(2)} = 16.033$, $p < 0.001$). The joystick achieved the highest percentage of positive feelings among the controllers (95%), is significantly higher than the gaze (40%).

For the item "Q3 - Natural," there was a significant difference between the joystick and hand pose controllers (Friedman $X^2_{(2)} = 12.400$, $p = 0.002$). The hand pose achieved the highest percentage of positive feelings among the controllers (55%), is significantly higher than the joystick (15%).

For the item "Q4 - Enjoyment", there was no significant difference between the game controllers (Friedman $X^2_{(2)} = 8.041$, $p = 0.018$, with the multiple comparisons tests with p value adjusted (Gaze-Hand Pose $p = 0.207$), (Gaze-Joystick $p = 0.144$), (Hand Pose-Joystick $p = 1$)).
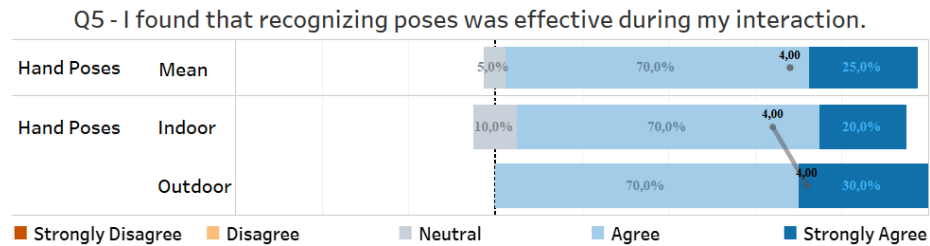

Fig. 6. Participants' perception of effectiveness while using the hand pose controller, first row: all participants, second row: participants separated per group (indoor, outdoor).

For the evaluation of "Effectiveness" perception (Figure 6) the majority of the participants pointed out positive feelings regarding the effectiveness of the hand pose recognition with only 5% of "Neutral" responses. In addition, we found no significant difference between the participants who performed the study in different environments (Mann-Whitney $U = 41.500$, $p = 0.423$).

Concerning the performance of participants using the hand pose controller in indoor and outdoor environments we find none significant difference (t-test $t_{(18)} = 0.631$, $p = 0.536$). This result is in agreement with the evaluation of effectiveness perception.

We also recorded the number of coins collected by participants using each game controller (see Figure 7). The Shapiro-Wilk test shows that the data follows a normal
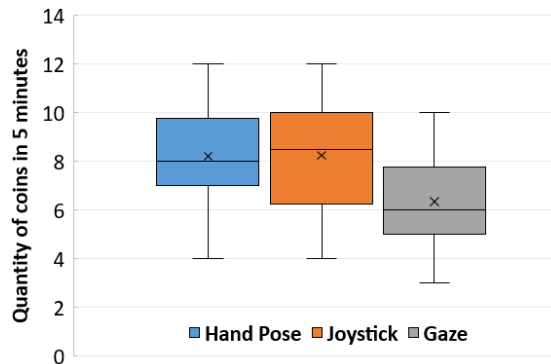
---

[1] https://www.ibm.com/analytics/spss-statistics-software.

Fig. 7. Quantity of coins per game controller: Hand Pose (M = 8.2, SD = 0.46), Joystick (M = 8.25, SD = 0.51) and Gaze (M = 6.35, SD = 0.43).

distribution (Hand Pose: $W = 0.970$, $p = 0.760$, Joystick: $W = 0.958$, $p = 0.499$ and Gaze: $W = 0.952$, $p = 0.396$). Thus, a One-way ANOVA with repeated measures ($\alpha = 0.05$) with *posthoc* and correction of Bonferroni was used ($F_{(2,38)} = 8.218$, $p = 0.001$). We note that the quantity of coins collected while using the hand pose controller was significantly higher when compared to the gaze controller. However, it was not different from the performance achieved while using the joystick.

One limitation of FPVRGame is to use a single input device. The interaction occurs only when the user places his hand within the angle of the camera, which can cause some physical discomfort. Therefore, we evaluated the participants' comfort when using hand pose and surprisingly, such aspect was not a problem for the participants to enjoy and to achieve excellent performance.

5 Discussion

Natural User Interfaces (NUIs) is not a trivial definition [19]. Bill Buxton [3] argues that NUI exploits skills that were acquired through a lifetime of living, which minimizes the cognitive load and therefore minimizes the distraction. He also states that NUIs should always be designed with the use of context in mind. Bowman et al. [2] questioned naturalism in 3D interface saying that high levels of naturalism can enhance performance and the overall user experience, but moderately natural 3D UIs can be unfamiliar and reduce performance. Traditional, less natural, interaction styles can provide excellent performance, but result in lower levels of presence, engagement, and fun.

Dealing with this trade-off between naturalism versus performance is still a challenge, and few efforts have been reported about how to explore the design space in order to find the appropriate and natural interaction for a specific context of use. Different from Son et al. [27] and Yousefi et al. [31], our work addresses such issue by using a user-centered design approach to build a hand pose vocabulary for 3D user interaction in an egocentric vision scenario. The Wizard-of-Oz technique used in Study Two shown to be adequate

for validating the preliminary vocabulary achieved by Study One and contributing for a good final user's experience with the FPVRGame, as was discussed in Study Three.

In addition, the simulation with the wizard allowed appropriate conditions for recording the images and generating the data set. Before using the simulation we tried to ask the volunteers to perform some hand poses to be captured and used in the CNNs training, using the same method of capturing the databases cited above in related works [17,30,33]. However, in practice, the results were not good, and we assume that it was due to the robotic and not natural movements made by the volunteers without causing oscillations in the poses.

Even with a dataset composed only of images collected indoor, no need for background extraction, chroma key, and lighting adjustments [31] or data augmentation [27], our CNN model was able to generalize the recognition of hand poses, and it works appropriately for both indoors and outdoors environments. Besides that, we observed that the performance of the hand pose interaction with the FPVRGame presented satisfactory results, similar to the joystick in the number of collected coins (Figure 7).

6   Conclusion

Our FPVRGame proposal allows the use of bare hands as a control for VR and Head Mounted Display scenarios, especially for low-end VR devices. Our scenario is focused on egocentric vision and presents a set of natural and comfortable hand poses, considering users' preferences. We developed a public dataset [21], which allows extensions and inclusions of new hand poses. The FPVRGame demonstrates a hand pose interaction solution, based on deep learning, that using a trained CNN model capable of recognizing hand poses, captured at indoors or outdoors environments and without any illumination or background constraint. We achieved an average accuracy of 97.89%, which allowed smooth and comfortable human interaction through different usage scenarios. We demonstrate our results using commercial low-end HMD's and compare our solution with traditional interaction devices.

References

1. Al Maimani, A., Roudaut, A.: Frozen suit: Designing a changeable stiffness suit and its application to haptic games. In: Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems. pp. 2440–2448. CHI '17, ACM, New York, NY, USA (2017). https://doi.org/10.1145/3025453.3025655, http://doi.acm.org/10.1145/3025453.3025655

2. Bowman, D.A., McMahan, R.P., Ragan, E.D.: Questioning naturalism in 3d user interfaces. Communications of the ACM **55**(9), 78–88 (2012)

3. Buxton, B.: Sketching user experiences: getting the design right and the right design. Morgan kaufmann (2010)

4. Cao, C., Zhang, Y., Wu, Y., Lu, H., Cheng, J.: Egocentric gesture recognition using recurrent 3d convolutional neural networks with spatiotemporal transformer modules. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 3763–3771 (2017)

5. Crankshaw, D., Wang, X., Zhou, G., Franklin, M.J., Gonzalez, J.E., Stoica, I.: Clipper: A low-latency online prediction serving system. In: NSDI. pp. 613–627 (2017)

6. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. pp. 248–255. Ieee (2009)

7. Fushiki, T.: Estimation of prediction error by using k-fold cross-validation. Statistics and Computing **21**(2), 137–146 (2011)

8. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)

9. Höll, M., Oberweger, M., Arth, C., Lepetit, V.: Efficient physics-based implementation for realistic hand-object interaction in virtual reality. In: 2018 IEEE Conference on Virtual Reality and 3D User Interfaces (2018)

10. Huang, G., Liu, Z., Weinberger, K.Q., van der Maaten, L.: Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. vol. 1, p. 3 (2017)

11. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: Convolutional architecture for fast feature embedding. arXiv preprint arXiv:1408.5093 (2014)

12. JoliBrain: Deep detect (2018), https://deepdetect.com, accessed: 2018-09-20

13. Kelley, J.F.: An empirical methodology for writing user-friendly natural language computer applications. In: Proceedings of the SIGCHI conference on Human Factors in Computing Systems. pp. 193–196. ACM (1983)

14. Knierim, P., Schwind, V., Feit, A.M., Nieuwenhuizen, F., Henze, N.: Physical keyboards in virtual reality: Analysis of typing performance and effects of avatar hands. In: Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems. pp. 345:1–345:9. CHI '18, ACM, New York, NY, USA (2018). https://doi.org/10.1145/3173574.3173919, http://doi.acm.org/10.1145/3173574.3173919

15. Koller, O., Ney, H., Bowden, R.: Deep hand: How to train a cnn on 1 million hand images when your data is continuous and weakly labelled. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 3793–3802. Las Vegas, NV, USA (Jun 2016)

16. Lee, S., Park, K., Lee, J., Kim, K.: User study of vr basic controller and data glove as hand gesture inputs in vr games. In: 2017 International Symposium on Ubiquitous Virtual Reality (ISUVR). pp. 1–3 (June 2017). https://doi.org/10.1109/ISUVR.2017.16

17. Li, Y., Ye, Z., Rehg, J.M.: Delving into egocentric actions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 287–295 (2015)

18. Molchanov, P., Yang, X., Gupta, S., Kim, K., Tyree, S., Kautz, J.: Online detection and classification of dynamic hand gestures with recurrent 3d convolutional neural network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4207–4215 (2016)

19. Mortensen, D.: Natural user interfaces–what are they and how do you design user interfaces that feel natural. The Interaction Design Foundation (2017)

20. NVIDIA: NVIDIA tensorrt (2018), https://developer.nvidia.com/tensorrt, accessed: 2018-09-20

21. Oliveira, E.: Dataset from egocentrics images for hand poses recognition (2018), https://goo.gl/EEbtcP, accessed: 2018-09-10

22. Piumsomboon, T., Clark, A., Billinghurst, M., Cockburn, A.: User-defined gestures for augmented reality. In: IFIP Conference on Human-Computer Interaction. pp. 282–299. Springer (2013)

23. Proença, P.F., Gao, Y.: Splode: Semi-probabilistic point and line odometry with depth estimation from rgb-d camera motion. In: Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on. pp. 1594–1601. IEEE (2017)

24. Rautaray, S.S., Agrawal, A.: Vision based hand gesture recognition for human computer interaction: a survey. Artificial Intelligence Review **43**(1), 1–54 (Jan 2015). https://doi.org/10.1007/s10462-012-9356-9, https://doi.org/10.1007/s10462-012-9356-9

25. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision (IJCV) **115**(3), 211–252 (2015). https://doi.org/10.1007/s11263-015-0816-y

26. Sagayam, K.M., Hemanth, D.J.: Hand posture and gesture recognition techniques for virtual reality applications: a survey. Virtual Reality **21**(2), 91–107 (Jun 2017). https://doi.org/10.1007/s10055-016-0301-0, https://doi.org/10.1007/s10055-016-0301-0

27. Son, Y.J., Choi, O.: Image-based hand pose classification using faster r-cnn. In: Control, Automation and Systems (ICCAS), 2017 17th International Conference on. pp. 1569–1573. IEEE (2017)

28. Sra, M., Xu, X., Maes, P.: Breathvr: Leveraging breathing as a directly controlled interface for virtual reality games. In: Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems. pp. 340:1–340:12. CHI '18, ACM, New York, NY, USA (2018). https://doi.org/10.1145/3173574.3173914, http://doi.acm.org/10.1145/3173574.3173914

29. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1–9 (2015)

30. Tewari, A., Grandidier, F., Taetz, B., Stricker, D.: Adding model constraints to cnn for top view hand pose recognition in range images. In: ICPRAM. pp. 170–177 (2016)

31. Yousefi, S., Kidane, M., Delgado, Y., Chana, J., Reski, N.: 3d gesture-based interaction for immersive experience in mobile vr. In: Pattern Recognition (ICPR), 2016 23rd International Conference on. pp. 2121–2126. IEEE (2016)

32. Zhang, C., Xue, Q., Waghmare, A., Meng, R., Jain, S., Han, Y., Li, X., Cunefare, K., Ploetz, T., Starner, T., Inan, O., Abowd, G.D.: Fingerping: Recognizing fine-grained hand poses using active acoustic on-body sensing. In: Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems. pp. 437:1–437:10. CHI '18, ACM, New York, NY, USA (2018). https://doi.org/10.1145/3173574.3174011, http://doi.acm.org/10.1145/3173574.3174011

33. Zhang, Y., Cao, C., Cheng, J., Lu, H.: Egogesture: A new dataset and benchmark for egocentric hand gesture recognition. IEEE Transactions on Multimedia **20**(5), 1038–1050 (2018)