Post-print of: Czuszyński K., Rumiński J., Kwaśniewska A.: Gesture Recognition With the Linear Optical Sensor and Recurrent Neural Networks. IEEE SENSORS JOURNAL. Vol. 18, iss. 13 (2018), p. 5429-5438. DOI: 10.1109/JSEN.2018.2834968

Gesture Recognition with the Linear Optical Sensor and Recurrent Neural Networks

Krzysztof Czuszynski, Jacek Ruminski, *Member, IEEE*,
and Alicja Kwasniewska, *Student Member, IEEE*

***Abstract* — In this paper the optical linear sensor, a representative of low resolution sensors, was investigated in the multi-class recognition of near field hand gestures. The Recurrent Neural Network (RNN) with a GRU (Gated Recurrent Unit) memory cell was utilized as a gestures classifier. A set of 27 gestures was collected from a group of volunteers. The 27 000 sequences obtained were divided into training, validation, and test subsets. The primary research goal was to define the most appropriate model architecture in terms of the accurate recognition of each gesture. An additional aim of the research was to investigate the kind of input data, i.e., raw data, or preprocessed (feature) data, which generally produces better results. Therefore, three datasets were generated: raw data, simple features data, and high level features data (this includes information about hand poses which are already recognized). The random search method was applied to achieve hyperparameter optimization to find the best possible topology for the neural network. The analysis performed shows that selected models were characterized by a test score at a level of 96.89% for the raw data, 95.75% for simple features, and 93.38% for high level features. Results indicate that the direct use of raw data obtained from the optical linear sensor evaluated on the RNN with GRU memory cells allows for the reliable recognition of even complex gestures. Therefore, such solutions may have the potential to serve as a support to, or as an alternative to video based sensors especially for mobile devices.**

***Index Terms* — gesture recognition, human computer interaction, optical sensors, recurrent neural networks.**

I. Introduction

The gesture recognition within low complexity sensors deserves to be revisited due to the constant development of mobile devices and the increased computational capabilities of modern computers, that enable deep learning of complex models. Over the years, the variety of applications has significantly increased due to the enhanced accessibility and reduced price of devices equipped with gesture sensors. Also, improved computational capabilities has led to the utilization of larger models, which were not considered before due to the required significant training and inference time.

Focusing on basic optical sensors, the low power single LED four photodiodes (PDs) gesture sensor, with a partially open cavity package, recognizing hand swipes with an accuracy of 100%, was proposed by Kim et al. [1]. The sensor for capturing the gestures of a virtual computer mouse, based on 10 IR transceivers, was described by Tang et al. [2]. A set of gesture sensors based on several LED-photodiodes (exact number depends on the configuration) used for smart glasses, as a ring or a universal clip, handling up to 9 gestures, was investigated by Withana et al. [3]. Other input methods, which have gained a lot of popularity in the early years of the human system interaction systems, are hand glove solutions. Murakami et al. utilized a special hand glove for recognition of some examples from Japanese sign Language [4], while hand trajectories could be detected with the glove, proposed by Vamplew et al. [5]. Later, video based interfaces started to become more popular [6], [7]. Maraqa et al. also utilized a colored glove, in addition to a video signal, for hand region identification [8]. Standard RGB streams were often complemented by depth information from an additional camera like Kinect [9]–[11]. Hand gestures were also recognized from point cloud obtained from devices like Leap Motion [12][13].

Considering gesture detection methods, relatively basic sensors (in comparison to video based versions) were often satisfied with models, which do not demand high prior or preprocessing computation. Often, sensors handling a small number of clearly defined gestures (like "swipe") were relying on hardware programmable rules like Finite State Machines (FSM). In some research, FSM was applied, e.g., to recognize gestures performed with the use of a virtual mouse [2] but also to recognize hand trajectory gestures from a video (no hand pose differentiation) [14]. Although the FSM model is useful, its implementation is time consuming, and finding unequivocal rules could be tedious. Therefore, more advanced machine learning models are being considered at present for handling many complex gestures, where trajectory and/or a pose matters. Basic sensors have also benefitted from them in recent years. For example, Withana et al. included Support Vector Machine (SVM) and Bayes Network in their two stage classifier for photodiodes based gesture sensors [3]. As presented, 5 to 8 gestures (depending on the configuration of the sensor) were distinguished with an accuracy of 94.8%.

"This work has been partially supported by NCBiR, FWF, SNSF, ANR and FNR in the framework of the ERA-NET CHIST-ERA II, project eGLASSES – The interactive eyeglasses for mobile, perceptual computing and by Statutory Funds of Electronics, Telecommunications and Informatics Faculty, Gdansk University of Technology".

The authors are with the Department of Biomedical Engineering, Faculty of Electronics, Telecommunications and Informatics at Gdansk University of Technology, 80-233 Gdansk, Narutowicza 11/12 St, Poland (e-mail: krzysztof.czuszynski@pg.edu.pl, jacek.ruminski@pg.edu.pl, alicja.kwasniewska@pg.edu.pl).

Lately, more complex models have been utilized for handling a larger amount of data, e.g., video streams. Many of the recently proposed recognition systems use Hidden Markov Models (HMM). One of the first implementations of HMM, for the identification of gestures from a video stream, was accomplished by Starner and Pentland [15]. The input-output HMM were utilized for gesture recognition by Marcel et al. [16]. Stefanov and Beskow used HMM, fed with parameterized images of a person with localization of skeleton and joints from Kinect camera, for the recognition of Swedish sign language [9]. Dynamic hand gesture classification with centroid tracking and discrete HMM was used by Premaratne et al. [17]. Sign language recognition with the use of inertial motion sensing hand glove and HMM was done by Galka et al. [18]. SVM method fed with data from a Leap Motion sensor was used to classify hand poses for American sign language [12], as well as digits and letters based on circled trajectory of a hand [13]. Gesture and action recognition (trajectory and hand pose) based on processed depth images from Kinect camera were also approached through utilization of other methods, e.g., K-nearest neighbors [10] or artificial neural networks for 1 and 2 dimensional data often used as an input to a sequence handling model. Kim et al. utilized 1-D convolutional neural network (CNN) for classification of radio impulses, reflected from hand performing a gesture [19]. CNNs along with joint trajectory maps were used by Wang et al. to classify actions from video sequences [20]. Feedforward neural network model, with similarity matching for the recognition of gestures from an accelerometer based sensor, were used by Xie et al. [21].

Currently, a very common approach is to make use of special types of neural networks (Recurrent Neural Networks), which preserve their state over time steps due to memory cells. Their output relies not only on a recent input but also on a whole sequence of inputs [22]. They are often used in sequence to sequence processing tasks, e.g., language translation or speech recognition [23], but they are also good at classification. Since gestures are made up of a sequence of poses, RNNs were also utilized in the gesture recognition domain. The first experiments with Recurrent Neural Networks for gesture recognition were conducted by Murakami and Taguchi [4]. Their proposed solution allows for highly accurate (96%) recognition of 10 different Japanese sign language gestures using 16 inputs from a special hand glove. Vamplev and Adams used RNN with 3 inputs describing the position of a hand in an electronic glove [5]. They recognized 16 hand motions with a mean accuracy of 98.9%. Additionally, the recognition of gestures before completion was also checked based on checking whether a threshold value was eventually exceeded. The recognition of Arabic sign language from a video stream, with the use of a colored glove, the Elman recurrent network and a fully recurrent neural network was performed by Maraqa and Abu-Zaiter [8]. The fully recurrent network, achieved a detection accuracy of 95.11%, whereas the Elman network achieved 89.66% only, when evaluated with 30 signs. Ng and Ranganath [6] used combined outputs from RNN and HMM for a vision-based gesture recognition system. The identified hand poses and a motion vector of hands, between recent and previous sampling events, were given as inputs to independent classifiers. For each gesture (out of 14) separate RNN and HMM models were applied. Recognition of trajectory gestures with a sensor based on accelerometer and continuous time RNN (CTRNN) was applied by Bailador et al. [24]. The Jordan Recurrent Neural Network (JRNN) for gesture recognition as a sequence of hand poses was proposed in [25]. In another research, Araga et al. utilized the JRNN for recognition of gestures, performed by a hand in a glove, from video stream [25]. The multimodal approach (video streams combined with skeleton joint streams) to gesture recognition using RNN was applied by Nevarova et al. [26].

However, a single recurrent neuron is a very basic cell and has limitations, in e.g., a length of the analyzed sequence (such as the vanishing gradients problem) [27]. Yet, with the progress made in deep learning and the invention of memory cells with gated activation functions like LSTM and GRU, RNN has become resistant to such problems [28], [29]. With these, analysis of longer sequences, e.g., from sensors sampling with higher frequency, turned out to be more efficient [20], [30], [31]. Shin and Sung approached the techniques of gesture recognition based on video and accelerometer signals, both analyzed with the low complexity fixed-point RNNs, with LSTM memory cell [32]. Sign language recognition with the use of LSTM based on four joint trajectories (left/right hand and left/right elbow) obtained from Kinect camera, was investigated by Liu et al. [11]. LSTMs were also used to classify ink traces (trajectories), which were expressed as a set of features by Otte et al. [33]. The CNN-LSTM network based gesture recognition from video sequences was proposed and expanded by Tsironi et al. [7], [34]. Data from RGB and depth cameras were used by Chai et al. [35] to recognize gestures (no hand pose differentiation) in a two stream RNN with LSTM cells. Vanilla RNN, along with LSTM, and GRU were tested in the recognition of Schaeffer sign language by Oprea et al. [30]. 25 gestures were extracted from Kinect video sequences of 8 upper body joints using CNN; sliding window was applied and sequences of varied length were passed to RNN. The most effective network was a three-layered LSTM with 25 blocks on each layer. The achieved gesture recognition accuracy for LSTM and GRU based models were 93.13% and 91.07% respectively.

The goal of this paper is an analysis of the accuracy of a multi-class hand gesture recognition, method utilizing the linear optical sensor and RNN model based on GRU memory cells. Of particular interest is the impact on the detection accuracy of different data types representing a captured gesture: as raw signals from the sensor, as signals processed into features, as higher level features.

The paper is organized as follows: Section I consists of the introduction, the state of the art, and objectives of the work. Section II presents a description of the utilized gesture sensor, a gestures dataset, and methods used for recognition. Section III shows the results obtained from training and testing on differently processed datasets, followed by the discussion of achieved results. The paper is concluded in Section IV.

II. Materials and Methods

A. Linear optical gesture sensor

The research was conducted with the designed low resolution linear optical gesture sensor consisting of n_{PD}=8 aligned IR photodiodes (TSL260RD) and n_{LED}=4 IR LEDs (KP-3216F3C) (Fig. 1). The photodiodes (PD) of the sensor are 1 cm apart, whereas the LEDs were mounted with a spacing of 2 cm. The applied light collimator limits the field of view of PDs and LEDs to 60° and 120° respectively. The specific photodiode model (which for the wavelength of λ=940 nm saturates with the irradiance at the level of around 65 μW/cm^2) was selected, as it has an in-built operational amplifier and does not require additional hardware signal conditioning. The described formation of optoelectronic elements of the sensor was confirmed to have good properties in terms of hand pose recognition capabilities as presented in our previous studies [36][37]. The sensor is also equipped with a PIC24FV16KA302 microprocessor supplied with 5 V, which was sampling signals from the photodiodes and sending data via an UART interface to the PC.

The principle of operation depends on the operating mode of the sensor. In the active mode, light from pulsating LEDs is reflected by a nearby hand and produces a specific light intensity pattern on the sensor's surface, sampled spatially and temporally by distributed photodiodes. In the passive mode, LEDs are not utilized and ambient light (if present), when covered by a hand, produces a shadow intensity pattern. In this research, at the beginning of each period matched by the sampling frequency, the sensor performed sampling in both the active (LEDs on) and passive (LEDs off) mode. Subtracting the obtained patterns allows for a reduction in the impact of possible ambient light changes.

The values of light intensity, obtained during each sampling, are 8 numbers describing how much the given photodiode is illuminated with the IR light. Such array of 8 values will be referred to as a *data frame* (DF). Obtained values are in the range from 0 V (no light) to 3.8 V (PDs saturation). In this research, the LEDs were pulsating together with a frequency of 100Hz. The time set for LEDs on was 375μs.

In previous works, different formations of optoelectronic elements within a linear optical sensor structure and other geometrical, and technical parameters were already studied based on measurements and simulations. They were also referred to quantities obtained from experiments, performed on a group of volunteers [36]. The utilized touchless linear gesture sensor is dedicated for unobtrusive interaction and was designed to detect hand poses, performed within a close distance to the device (up to 5 cm) [36]. As different hand poses (finger arrangements) produce differentiated reflection/shadow patterns, the ability of the sensor to classify hand pose was evaluated. Utilizing artificial neural networks, it was able to classify three static poses (single finger, two and four joined) with the accuracy of 90.02%, when operating in the active mode. In further research, we improved the classification accuracy in the active mode to 93.34%, and obtained 98.76% for the passive mode [37]. In that research the sensor was also evaluated in differentiated ambient light conditions and at different angles to the light source. It proved to maintain high pose classification accuracy in the passive mode, when enough light was present. Additionally, the methodology for evaluating the objective condition for switching the operating mode of the sensor between active and passive, depending on the ambient light conditions, was presented. Because of the sensors features, high pose classification accuracy and possibility of operating mode adaptation was investigated in this study in the recognition of various gestures based mainly on the described hand poses.

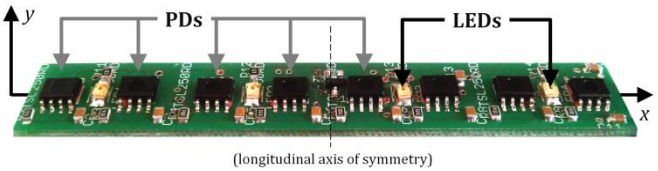

Fig. 1. Linear optical gesture sensor (optical block not presented for clarity).

B. Gestures set

In previous works, the classification of 3 compact hand poses by the linear sensor was performed [38]. The poses are: single finger, two, and four joined fingers (encoded as 1FS, 2FJ, and 4FJ respectively). Additionally, a two separated fingers pose (encoded as 2FS), investigated in [36], was also considered in this study. Such poses were selected as the linear construction of the sensor allows for the recognition of 1D patterns of reflected light, created by objects, located in its field of view. Therefore, there are only few different hand poses that could be distinguished. Three and four separated fingers were not considered due to the limited resolving power of the sensor and the physiological inconveniency of arranging finger into them in comparison to enlisted poses. The poses mentioned will form the basis of various discrete gestures (Fig. 2).

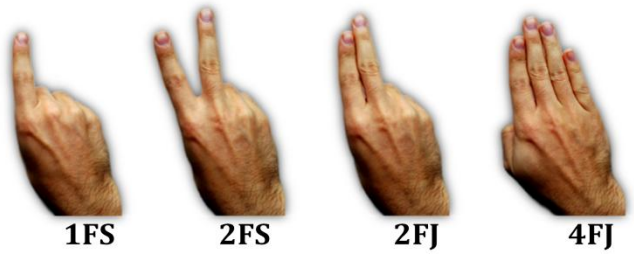

Fig. 2. Base hand poses utilized for gestures.

However, a gesture can be generally described as a sequence of poses in time. In this paper, we focus on a model supporting off-line gestures, what requires a completion of a gesture in order to provide a response to the user (when inferred within a system). This means that a decision about performed gestures may also be made based on the knowledge about the full gesture. That determines both the start and finish phase of a gesture to be precisely defined. For the linear optical sensor, a standard deviation of a data frame, sd_{DF}, was found to be a good indicator of the presence of a hand [37]. It was calculated as:

$$sd_{DF} = \sqrt{\frac{\sum_{i=1}^{nPD}(v_i - \bar{v})^2}{nPD-1}} \quad (1)$$

where v_i is the output voltage of i-th photodiode, $\bar{v}$ is the mean

TABLE I
DESCRIPTION OF GESTURES INVESTIGATED IN THE STUDY

Name	Poses involved	Localization / Path	Category of a gesture
1FS	1FS	Center	Static pose
2FS	2FS	Center	
2FJ	2FJ	Center	
2FJside	2FJ	Right / Left	
2FJhigh	2FJ	Center High	
4FJ	4FJ	Center	
1FSslow	1FS	Right / Left	Static pose and path
2FSslow	2FS	Right / Left	
2FJslow	2FJ	Right / Left	
2FJfast	2FJ	Right / Left	
4FJfast	4FJ	Right / Left	
2FJzoom	2FJ	In / Out	
4FJzoom	4FJ	In / Out	
2FJshakeVer	2FJ	Vertical shake	
2FJshakeHor	2FJ	Horizontal shake	
2FJellipse	2FJ	Ellipse	
Cut	2FS, 2FJ	Center	Dynamic pose
Dbcut	2FS, 2FJ	Center	
Chopin	4FJ waving	Center	

value of all PDs, and n_{PD} is the number of PDs in the sensor. Therefore, the process of capturing a gesture may be summed up in three precise steps:

- no pattern observed ($sd_{DF} < T_{sd}$) – sensor waits,
- object appears in front of the sensor ($sd_{DF} > T_{sd}$) – data storage / transfer begins,
- object leaves the field of view of the sensor ($sd_{DF} < T_{sd}$) – evaluation of the performed gesture starts,

where T_{sd} is the standard deviation threshold. In this study, as in our former research [38], a value of T_{sd} =0.1 V was used. This value was chosen after experiments with the use of simple models of joined fingers and after experiments with the participation of 11 volunteers (each performing multiple poses). This configuration enabled very good results of gesture recognition. However, the threshold value could be optimized in the future works.

Until step 2 ends, the sensor samples subsequent *data frames*, which, when a gesture is complete, will constitute a sequence. In terms of data formats, the sequence can be put into an array of a specified number of columns (reflecting the number of features / length of a *data frame*) and the number of rows depending on the duration of a gesture (number of sampling events - timestamps).

The gestures included in this study are discrete (system responses after activity), which is a category of the *flow* dimension of the taxonomy introduced by Wobbrock et al. [39]. The gestures were divided into three groups according to Wobbrock's *form* dimension categories. Most of the names of the gestures, considered in this study, were proposed by the authors (except the Cut gesture, which can be find e.g. in [3]).

1) Static pose

The cases from the first group are gestures, where a certain pose is inserted in front of the sensor for a short time (less than a second) and taken back. The rule is that both the pose and the position of the hand (except the phase of putting the hand inside/outside of the field of view of the sensor) remains unchanged during the gesture.

In order to demonstrate how this group can be enriched, some of the gestures were demonstrated above the center of symmetry of the sensor (x=0 cm) at a specific height, h=3 cm, above the center but further (h=5 cm), and some on the sides (h=3 cm, x=±3 cm). Gestures for which execution localization matters, are classified under a world-depended category from Wobbrock's *binding* dimension.

2) Static pose and path

For the second category gestures, the hand pose does not change, but the path varies. They include: swipe motions along the x axis (we distinguish right / left direction and slow / fast velocity) performed at h=3 cm, zoom-related actions (zoom in and zoom out) from outside the field of view of the sensor up to h=1 cm or inversely, shakes, where the hand pose oscillates rapidly about 3 times in the vertical or horizontal direction, while remaining in the field of view of the sensor; ellipse, where such trajectory is circled.

3) Dynamic pose

The third group gathers gestures, in which pose varies during the activity. The Cut gesture is a sequence of 2FS, 2FJ and 2FS, all performed during the single-hand appearance in front of the sensor. The Dbcut (double cut) is a sequence like Cut, but additionally followed by 2FJ, 2FS. They mimic the single and double click of the computer mouse. While performing the Chopin gesture, a person puts a full hand in front of the sensor and pretends to play on a virtual piano, waving fingers asynchronously above the sensor for a short time.

Taking into account the static pose position and movement direction degrees of freedom, the total number of gestures is 27. Table I summarizes the gestures included in the study. Each gesture is described by a base pose or poses required for performing it. Additionally, localization describes the position of a hand mainly in reference to the x axis (Fig. 1), when a hand does not change its position during a gesture. Otherwise, the movement of a hand is described by a path (e.g., Right: $+x$, Left: $-x$, In: $-y$, Out: $+y$, etc.). The last column of Table I presents Wobbrock's category of a gesture.

C. Experimental design

26 adult volunteers (13 females, 13 males; 34.3±10.4 yrs) were invited to participate in the experimental study consisting of repeating each gesture 20 times. Therefore, more than 500 representations of each gesture were recorded. Altogether, more than 13000 gesture sequences were registered with the sensor and successfully transmitted to a computer for further processing.

During the study, the gesture sensor was mounted in a holder on a desk (face up) in front of volunteers sitting on a chair, so they could conveniently control their movements. A sheet of cardboard with the expected trajectories of desired gestures was mounted perpendicular to a face of the sensor in order to increase the reproducibility of individual gestures performed by different participants (Fig. 3).

Pictures of a holder with a mounted sensor and a cardboard sheet along with enhanced trajectories of exemplary gestures are presented in Fig. 4.

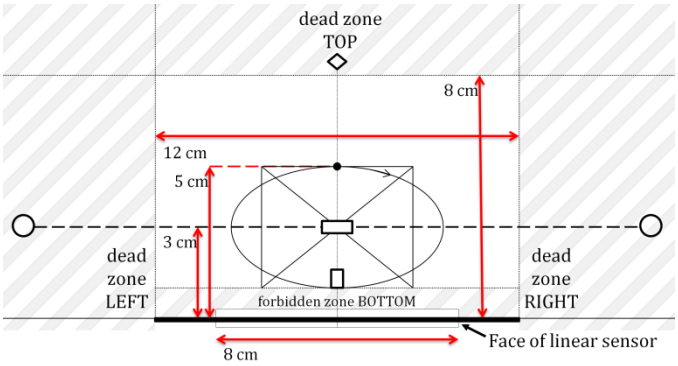

Fig. 3. Image with trajectories and positions for performing repeatable gestures that was printed on the sheet of cardboard and mounted on the holder behind the linear sensor. The face of the sensor was placed at the level of black bold horizontal line (bottom of an image).

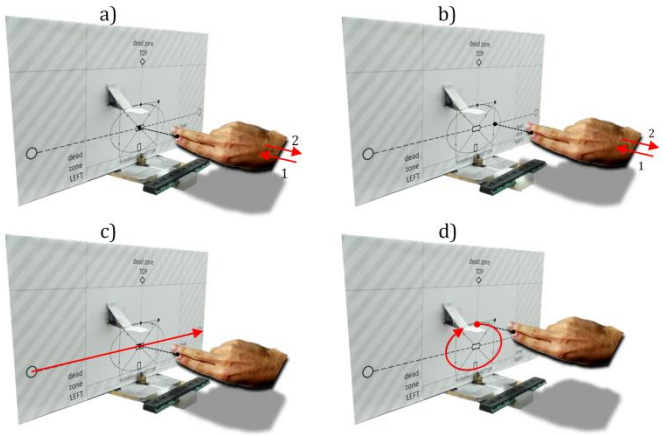

Fig. 4. Exemplary gestures. *Static pose*: 2FJ, 2FJsideR (a, b); *static pose and path*: 2FJslowR, 2FJellipse (c, d). Red arrows indicate a path for the hand to track, the dashed line shows the spot on the cardboard, at which the hand is pointing. The black part at the bottom of each image is the gesture sensor with an overlay. Note the position of the shadow produced by the hand as it indicates the top projection (important for *static pose* gestures). The hand size on the image may not reflect the exact scale in relation to the sensor.

D. Datasets

In further processing, three kinds of datasets were considered. They are composed of the same exact sequences, performed by the same volunteers, but differed in the type of data representing the gestures. Some methods, described in the Introduction section, feed a model with an implicit hand pose (e.g. by joints localization). We compare the performance of feeding the RNN with implicit and explicit information about the current hand pose.

1) Gesture as raw data

As gesture was defined as a sequence of poses timestamps; the most fundamental representation of a gesture recorded by a linear sensor would be a 2D array. This array, composed of *data frames*, contains raw values of light intensity patterns in subsequent samplings (8 columns: each with a light intensity value from one photodiode of the optical sensor). This data does not contain any interpretation of the presented hand pose, or hand position, hence the gesture recognition model has to reason it by itself. Such dataset will be referred to as *raw* data.

2) Gesture as features

Gestures can be also considered as a series of elementary poses in time and space. Therefore, for each of timestamps, in all sequences, a set of features was calculated. These are the features based on which a gesture recognition system (GRS), for pose classification, was proposed: number of peaks in DF (*PKS*); center of gravity of DF (*COG*); *mean*(DF); sd_{DF}; and number of values in DF smaller than $2sd_{DF}$ [36], [38]. In order to enable the model to track the movements of a hand with greater precision, the described set of features was supplemented with *max*(DF). Therefore, a timestamp of the second dataset consists of 6 columns and it will be referred to as *features*. This data contains explicit information about the position of a hand (*COG*, max) in relation to the sensor. Also, the model will get features that describe a pose, hence it will possibly converge faster.

3) Gesture as higher level features

The third dataset consists of just 3 features. The first feature is the recognized as a hand pose (1FS, 2FS, 2FJ or 4FJ) obtained from a GRS with a 2-layer neural network [38]. The remaining two features are COG and max to reflect the position of the hand. This dataset will be referred to as *HLfeatures*.

E. Recurrent Neural Network model

Unlike in the pose classification problem, the position or trajectory (path) of a hand in subsequent sampling cycles for gestures is also of importance. Therefore, Recurrent Neural Network model for supervised sequence classification was selected in this study as the gesture recognition method. Training and testing phases of a model were performed with the sequence to vector fashion, where a recurrent network is fed with a *k* timestamps-long sequence and only the last (*k*-th) output of the network is analyzed.

In our preliminary tests, it was confirmed that standard RNN memory cells are not suitable for the recognition of long (more than 30 timestamps), strongly varied in length sequences. On the other hand, training sessions with GRU memory cell converged faster and the computation took less time than for the more complex LSTM units at longer distances. Hence, in this study, we investigate GRU cells only. The sequences were extracted and preprocessed using Matlab software. The RNN implementation was developed using TensorFlow and Scikit-learn libraries [40] and run on NVIDIA DGX Station with 4 Tesla V100 GPUs[1].

The random search approach was applied for hyperparameters optimization of the model [41]. For each of the three datasets, 128 trial training sessions, which lasted for at most 1000 training epochs, *e*, were performed. The hyperparameters were sampled from the following sets:

- learning rate uniformly from the range <1E-7, 0.05>,
- number of layers chosen with an equal probability from the set of [1, 2, 3, 4, 5],
- number of neurons (the same for each layer) chosen with equal probability from the range <10, 100>,
- activation function of hidden layer chosen with equal probability from the set of [tanh, softsign, ReLU, ELU, SELU [42]],
- dropout output keep probability chosen with equal probability from the set of [0.5, 0.7, 0.7, 1],
- kernel initializer of the GRU cell chosen with equal probability from the set of [He initialization, Xavier

[1] sources available at: https://github.com/ChrisQlasty/RNNforGestures

initialization, None],

Dropout of 0.7 was specified twice to have a higher sampling probability than the other two options, as it was found to be effective in some research in which experiments are sensitive to hyperparameters [43].

After hyperparameters sampling and during training sessions, the following rules (e.g. early stopping) and limits were applied to make computations more efficient:

- if (n_layers == 1), doDropout = False,
- while (n_neurons·n_layers > 100), n_neurons-=3,
- if (activation function == ReLU), scale the inputs to the range of <0, 1>, <-1, 1> otherwise,
- stop training session if no progress of accuracy on the validation set was observed for 100 epochs,
- stop training session at the epoch *e* if the best observed validation accuracy was observed before epoch *e*/2 [41],
- stop training session if accuracy on the validation set was 0% across four checks in a row, as most likely an event of exploding or vanishing gradient has occurred,
- minimum of 100 and maximum of 1000 epochs were applied for training when non-zero validation accuracy was obtained.

One may observe that the total number of neurons limit leads to an investigation of shallow wide and deep narrow networks. The applied optimizer was Adam. A small batch size (256) was selected because for the cost of the longer computation time the trained model is likely to generalize better [44]. Also, batch shuffling was employed to increase the generalization. Gradient clipping with a norm equal to 1 for all of the datasets was applied as well. A dense layer with 27 neurons (number of gestures to recognize) and the softmax activation function was used as the output of the network. The evaluation step interval was 5 epochs.

F. Final gestures datasets

The symmetry of the sensor allows for the performance of data augmentations. The applicable transformations are: *mirror* (swap along *x* axis), *time reverse* (swap in time) and *mirror and time reverse*. However, they have to be carefully applied as, for swipes or zoom gestures transformation, it may be turned into an instance of the opposite class (e.g. 1FSslow right swipe instead of 1FSslow left swipe). Additionally, the ellipse gesture is liable to only one transformation (*mirror* and *time reverse* at once). Therefore, 1000 sequences per gesture were obtained. The average length of a sequence is 58.66±40.91 timestamps (with min = 4; max = 332).

During the experiments, when participants were asked to perform slow and fast swipe gestures, no specific constraints on the pace of a swipe were given (similar to real situations). Therefore, they performed gestures according to an individual sense of speed. As a result, in terms of hand movement velocity, slow gestures, performed by some participants, were comparable to fast gestures for other participants. Therefore, a sharp threshold defining the permissible duration of a slow and fast swipe was introduced. It was based on the number of timestamps producing a maximal sensitivity for two speed classes (21 timestamps). The estimation of velocity can be fully outsourced from the RNN model, so the user could set the threshold to his needs, but in this work, it was investigated how well models are able to distinguish the same gestures performed at varied time scales.

The datasets were divided into training, validation and testing sets with the ratio 0.7/0.15/0.15 utilizing stratified sampling and obtaining sets with perfectly balanced classes. The training set consists of 18900 gestures (700 instances per class), the validation set has 4050 gestures (150 instances per class), and the testing set also consists of 4050 gestures (150 instances per class).

III. Results and Discussion

We compare results for three ways of representing a gesture in the form of cumulated plots. Among 128 random trials, almost 50% of models trained with *raw* data reached at least 90% classification accuracy on the validation set. The best model obtained with *raw* data has a validation score at the level of 96.86%. Over 35% of models trained with the *features*, reached 90%, whereas the best model reached 95.95%. For the *HLfeatures*, scarcely above 30% of trials exceeded 90% of classification accuracy with a 93.78% for the most accurate model. The cumulated plots are presented in Fig. 5.

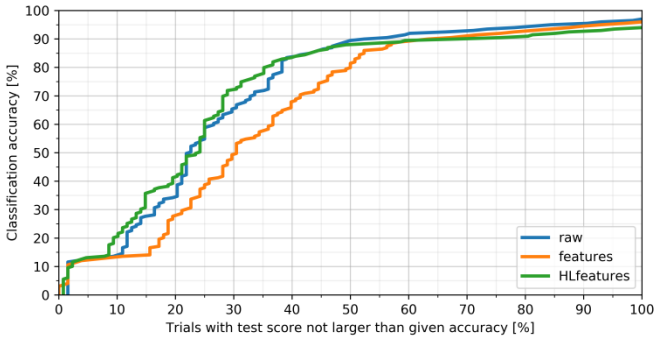

Fig. 5. Accuracy on the validation subset for models trained with three types of datasets.

The standard deviation of accuracy of the top 3 models trained on each of the datasets was less than 0.08%. Therefore, from among them, we selected models trained for the largest

TABLE II
GESTURES DETECTION ACCURACY, EVALUATED ON THE TESTING SET, FOR EACH OF THE SELECTED MODELS DEDICATED FOR THREE TYPES OF DATA (HIGHLIGHTED CELLS INDICATE DETECTION WITH THE ACCURACY ≥ 95%)

raw	*features*	*HLfeat.*	gesture	*raw*	*features*	*HLfeat.*	gesture
Detection accuracy [%]				Detection accuracy [%]			
97.33	96.67	90.67	1FS	94.67	94	90	2FJfastR
98.67	96	98	2FS	94.67	89.33	89.33	2FJfastL
94	94	88.67	2FJ	93.33	90	88	4FJfastR
96.67	99.33	100	2FJsideL	98	92.67	90.67	4FJfastL
99.33	99.33	99.33	2FJsideR	97.33	92.67	91.33	2FJzoomi
100	98.67	92	2FJhigh	99.33	94	86.67	2FJzoomo
100	98.67	93.33	4FJ	98.67	95.33	89.33	4FJzoomi
99.33	99.33	100	1FSslowR	98.67	100	99.33	4FJzoomo
98.67	98	96.67	1FSslowL	91.33	98	96	2FJshakeVer
98	96.67	96	2FSslowR	94	98	99.33	2FJshakerHor
94.67	91.33	88.67	2FSslowL	95.33	94	96.67	2FJellipse
98	95.33	90.67	2FJslowR	99.33	97.33	95.33	Cut
98	94.67	92	2FJslowL	95.33	98	96	Dbcut
				94.67	94	90	Chopin

number of epochs as the models, that will be evaluated on the testing subsets. The accuracy of detection of individual gestures from the testing subsets, by each of the selected models, is presented in Table II. The summary of the selected models containing values of sampled hyperparameters and their performance on the testing set is presented in Table III.

The error confusion matrix (ECM) for the best model, trained on *raw* data and evaluated on the testing set, which achieves 96.89% accuracy, is presented in Fig. 6. The range of color map from the image was set to emphasize the detection errors. The diagonal from the ECM was obtained by subtracting its values from 100%. Hence, the values in each row, not located on a diagonal, sum up to a value on a diagonal.

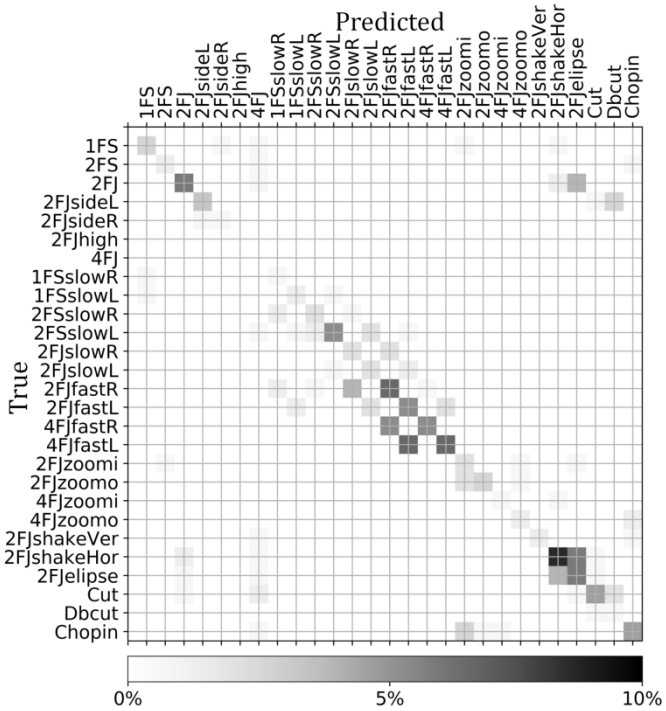

Fig. 6. Error confusion matrix of the best model recognizing gestures from raw data. The color bar indicates the percentage of error. Values on the diagonal present the sum of classification error for a given class.

From Fig. 5. one can see that statistically it is slightly easier to train RNN on the *HLfeatures* than on *raw* data and to reach a high classification accuracy (up to 80%). At this point, the number of models, trained on both of these datasets, exceeds the number of models, trained on the *features* data, by over 10 percentage points. However, as training sessions get deeper (where classification accuracy exceeds 90%), the *HLfeatures*-based model becomes less competitive. This is most likely because the utilized pose classifier (which delivers one of three inputs for these models) has its own classification error, which burdens the overall recognition ability of the RNN, although in this dataset the pose is already identified for the top RNN model. The *raw* dataset is the only one, where over half of the sampled sets of hyperparameters occurred to constitute models able to classify gestures with a recognition accuracy of over 90%. Also, the performance of the selected model trained on *raw* data surpassed the best models trained on features data. Although the *raw* model was not trained for the largest number of epochs, it is characterized by the lowest standard deviation of the classification accuracy of individual classes, 2.39% (the lower the better), and with the highest value of the observed worst performance on any of the gestures from the analyzed set, 91.33% (the higher the better).

TABLE III
SUMMARY OF THE SELECTED MODELS FOR EACH TYPE OF DATA

Dataset type:	*raw*	*features*	*HLfeatures*
n_inputs	8	6	3
n_neurons	97	48	32
n_layers	1	2	3
Learning rate	0.0027	0.0051	0.0036
Dropout keep probability	1.0	0.7	0.7
Activation function	tanh	softsign	ELU
Kernel initializer	Xavier	He	None
Best epoch	280	520	245
Validation set accuracy	96.69%	95.83%	93.75%
Test set accuracy	**96.89%**	**95.75%**	**93.38%**
Minimal test score for class	91.33%	89.33%	86.67%
Median test score for class	98%	96 %	92%
Maximal test score for class	100%	100%	100%
Stdev. of test scores for class	2.39%	2.90%	4.26%

Additionally, from the top10 models trained on the *raw* data, 7 consisted of 3 or more hidden layers. On the other hand, from the top10 models trained on *features* data only 3 models had a number of hidden layers greater or equal to 3. For *HLfeatures*, half of the top10 models had at least 3 layers. This shows that when a network was already given extracted information (features), shallow topologies were favored. By contrast, when given *raw* data, deep networks proved to have generally more competitive learning capabilities. However, the selected models do not confirm this observation. Table III also shows that selected models preferred a large number of neurons in the network, which was close or equal to the applied limit (100). Hence, models with a larger capacity may be considered in the future.

The performance of static pose gesture recognition by the selected model trained on the *HLfeatures* dataset cannot be directly compared with the performance of the hand pose classifier [38]. However, it may be observed that RNN model detected 2FJ poses with relatively similar (±1%) accuracy. By contrast, the recognition of 1FSs was smaller by 7%, while for 4FJs it was larger by around 8%. However, the model trained using the *raw* dataset, proved to be able to recognize *static poses* much better in general. Considering the *raw* dataset, detection of swipe gestures (*static pose and path*) most often failed in the recognition of proper hand poses and also in the differentiation of swipe speed (Fig. 6). The *dynamic pose* cut gesture was marginally confused with double cut gestures (Fig. 6). The ellipse gesture was frequently confused with the horizontal shake and 2FJ was confused with the ellipse (Fig. 6). This could be due to the pace of the performed gestures, therefore, additional spatiotemporal filtering should be considered in future work.

The overall performance of the applied network trained on *raw* data is at a high level (96.89%), which may be compared to the works of other authors investigating similar sensors (94.8% across the configurations [3]). However, when considering low resolution sensors, most of the methods proposed by other authors handle a much lower amount of gestures. Therefore, RNN based model with GRU memory

cells is able to keep track of both hand pose and trajectory of a hand with good accuracy.

The size of the networks of selected models for all datasets is very similar, hence, utilizing *features* or *HLfeatures* would not present any advantage considering the forward pass time during inference (and could even be worse due to the time demanded for feature calculation). The high overall performance of the network may possibly be improved by excusing the network from the task of recognizing a motion pace. Also, the impact of reduced sampling frequency on gesture classification accuracy may be verified. In future works, the size of a hand may also be taken into account in gesture recognition, which would probably decrease the classification error even more. Further research may well refer to more efficient hyperparameter searches based on one of the optimization techniques [45] or an increased number of trials.

When considering the current consumption of the optoelectronic elements of the sensor, it is estimated to be at a level of 5.05 mA (25.25 mW), which was obtained from the usage reported in [36] and taking into account the sampling frequency applied within this study (100Hz).

Gesture interfaces based on video streams from RGB cameras, e.g., Kinect or Leap Motion sensor, are very powerful but require analysis of large amount of data in comparison to devices with several sensing elements like a linear optical sensor. Smaller amount of data allows to train smaller classification models, what is an important factor, when considering the utilization of gesture sensors within mobile devices with limited battery capacity and computation capabilities.

The optical linear sensor, utilized in this study, has also already been evaluated within the frame of smart glasses (Fig. 7). In this form, it has been tested in the detection of basic discrete and continuous gestures [46]. In the future research on the human system interactions, the RNN gesture recognition models, elaborated in this study using the TensorFlow library, will be implemented within wearable devices like the eGlasses platform. In our early research, the trained models were evaluated on a Samsung Galaxy S8 smartphone in order to measure the inference time. For each of three selected models, (Table III) a sequence of adequately processed data of a mean length (60 timestamps), was fed and each test was repeated 16 times. The average time of sequence inference for *raw* data (1 layer model) was 374±14ms, for *features* (2 layers) it was 765±29ms and for *HLfeatures* (3 layers) 1049±15ms. The same tests were performed on a PC with G2130 processor and the corresponding inference times were 1 order of magnitude smaller (20±3ms, 32±7ms and 42±9ms). It indicates that some optimization of an application for mobile devices can be done to improve the responsiveness of potential gesture interface based on the linear sensor. In our preliminary studies, we have also performed a resampling of gesture sequences and an effective sampling frequency was reduced from 100Hz to 25Hz, making the sequences four times shorter. As an effect, the gesture classification accuracy of new models trained on resampled sequences has dropped to 93.35%, 95.43%, and to 89.87% for *raw*, *features* and *HLfeatures* representations respectively. However, shorter sequences were processed faster (for 15=60/4 timestamps) and the corresponding inference times on Samsung Galaxy S8 were 115±21ms, 210±6ms and 292±16ms.

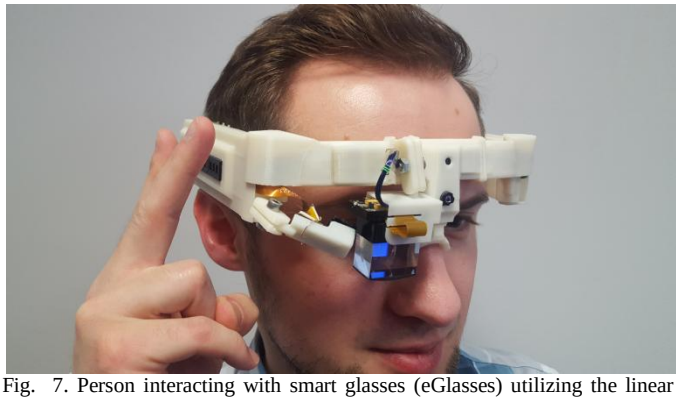

Fig. 7. Person interacting with smart glasses (eGlasses) utilizing the linear optical gesture sensor.

IV. Conclusion

In this paper, we have investigated the recognition of wide base of hand gestures (27), recorded by linear optical sensor consisting of 8 photodiodes. Three types of gesture representation, differing in the processing level (*raw*, *features*, *HLfeatures*) were considered for recurrent neural networks of different topologies. It was presented that a low resolution optical sensor, when utilized with a model of high complexity and unprocessed data (*raw*), may deliver a large amount of gestures, which are recognized with a high average accuracy of 96.89%. As previously stated, the performance may possibly be increased when taking into account the performed observations (motion speed or hand size issues). It has also been denoted, that the proposed set of gestures can be significantly enriched by many other combinations of *poses* and *paths*.

The presented gesture recognition abilities and low power consumption of the linear sensor open the possibility to consider the basic sensors as a strong, power saving support, or even as an alternative to video based interfaces for mobile, wearable devices, e.g., smart glasses and smartphones.

References

[1] J. S. Kim, S. J. Yun, and Y. S. Kim, "Low-power motion gesture sensor with a partially open cavity package," *Opt. Express*, vol. 24, no. 10, pp. 10537–10546, 2016.

[2] S. K. Tang, W. C. Tseng, W. W. Luo, K. C. Chiu, S. T. Lin, and Y. P. Liu, "Virtual Mouse: A Low Cost Proximity-Based Gestural Pointing Device," in *Human-Computer Interaction. Interaction Techniques and Environments: 14th Int. Conf., HCI International 2011*, Orlando, USA, *Proc., Part II*, 2011, pp. 491–499.

[3] A. Withana, R. Peiris, N. Samarasekara, and S. Nanayakkara, "zSense : Enabling Shallow Depth Gesture Recognition for Greater Input Expressivity on Smart Wearables," in *CHI '15 Proc. of the 33rd Annual ACM Conf. on Human Factors in Computing Systems*, Seoul, Korea, 2015, pp. 3661–3670.

[4] K. Murakami and H. Taguchi, "Gesture Recognition using Recurrent Neural Networks," in *Proc. of the SIGCHI conf. on Human factors in computing systems*, New Orleans, USA, 1991, pp. 237–242.

[5] P. Vamplew and A. Adams, "Recognition and anticipation of hand motions using a recurrent neural network," in *IEEE International Conf. on Neural Networks*, Perth, Australia, 1995, pp. 3–6.

[6] C. W. Ng and S. Ranganath, "Gesture recognition via pose classification," in *Proc. - Int. Conf. on Pattern Recognition*, Barcelona, Spain, 2000, vol. 15, no. 3, pp. 699–704.

[7] E. Tsironi, P. Barros, and S. Wermter, "Gesture Recognition with a Convolutional Long Short-Term Memory Recurrent Neural Network," in *Eur. Symp. on Artificial Neural Networks*, Bruges, Belgium, 2016, pp. 213–218.
[8] M. Maraqa and R. Abu-Zaiter, "Recognition of Arabic Sign Language (ArSL) using recurrent neural networks," in *1st Int. Conf. on the Applications of Digital Information and Web Technologies, ICADIWT 2008*, Ostrava, Czech Republic, 2008, pp. 478–481.
[9] K. Stefanov and J. Beskow, "Gesture Recognition System for Isolated Sign Language Signs," in *The 4th European and 7th Nordic Symposium on Multimodal Communication, 29-30 September 2016, University of Copenhagen, Denmark*, 2016, pp. 57–59.
[10] H. Stern, K. Smilansky, and S. Berman, "Depth Based Dual Component Dynamic Gesture Recognition," in *IPCV'13 - The 2013 Int. Conf. on Image Processing and Computer Vision*, Las Vegas, USA, 2013.
[11] T. Liu, W. Zhou, and H. Li, "Sign language recognition with long short-term memory," in *2016 IEEE Int. Conf. on Image Processing (ICIP)*, Phoenix, USA, 2016, pp. 2871–2875.
[12] C.-H. Chuan, E. Regina, and C. Guardino, "American Sign Language Recognition Usin Leap Motion Sensor," in *2014 13th Int. Conf. on Machine Learning and Applications*, Detroit, USA, 2014, pp. 541-544.
[13] Y. Chen, Z. Ding, Y. L. Chen, and X. Wu, "Rapid recognition of dynamic hand gestures using leap motion," in *2015 IEEE Int. Conf. on Information and Automation, ICIA 2015 - In conjunction with 2015 IEEE Int. Conf. on Automation and Logistics*, Lijiang, China, 2015, pp. 1419–1424.
[14] P. Hong, M. Turk, and T. S. Huang, "Constructing Finite State Machines for Fast Gesture Recognition," in *Pattern Recognition, 2000. Proc. 15th Int. Conf. on*, Barcelona, Spain, 2000, pp. 691–694.
[15] T. E. Starner and A. Pentland, "Visual Recognition of American Sign Language Using Hidden Markov Models," *MIT Cambridge Dept Of Brain And Cognitive Sciences*, pp. 189–194, 1995.
[16] S. Marcel, O. Bernier, J. E. Viallet, and D. Collobert, "Hand gesture recognition using input-output hidden Markov models," in *Automatic Face and Gesture Recognition, 2000. Proc. Fourth IEEE International Conference on*, Grenoble, France, 2000, pp. 456–461.
[17] P. Premaratne, S. Yang, P. Vial, and Z. Ifthikar, "Centroid tracking based dynamic hand gesture recognition using discrete Hidden Markov Models," *Neurocomputing*, vol. 228, no. October 2016, pp. 79–83, 2017.
[18] J. Galka, M. Masior, M. Zaborski, and K. Barczewska, "Inertial Motion Sensing Glove for Sign Language Gesture Acquisition and Recognition," *IEEE Sens. J.*, vol. 16, no. 16, pp. 6310–6316, 2016.
[19] S. Y. Kim, H. G. Han, J. W. Kim, S. Lee, and T. W. Kim, "A Hand Gesture Recognition Sensor Using Reflected Impulses," *IEEE Sens. J.*, vol. 17, no. 10, pp. 2975 – 2976, 2017.
[20] P. Wang, W. Li, C. Li, and Y. Hou, "Action Recognition Based on Joint Trajectory Maps with Convolutional Neural Networks," *https://arxiv.org/abs/1612.09401*, 2016.
[21] S. Y. Kim, H. G. Han, J. W. Kim, S. Lee, and T. W. Kim, "A Hand Gesture Recognition Sensor Using Reflected Impulses," *IEEE Sens. J.*, vol. 17, no. 10, pp. 2975 – 2976, 2017.
[22] J. Elman, "Finding structure in time* 1," *Cogn. Sci.*, vol. 14, no. 1 990, pp. 179–211, 1990.
[23] D. Bukhari, Y. Wang, and H. Wang, "Multilingual Convolutional, Long Short-Term Memory, Deep Neural Networks for Low Resource Speech Recognition," *Procedia Comput. Sci.*, vol. 107, pp. 842–847, 2017.
[24] G. Bailador, D. Roggen, G. Tröster, and G. Trivino, "Real time gesture recognition using continuous time recurrent neural networks," in *Proc. of the ICST 2nd Int Conf. on Body area networks*, Florence, Italy, 2007.
[25] Y. Araga, M. Shirabayashi, K. Kaida, and H. Hikawa, "Real time gesture recognition system using posture classifier and Jordan recurrent neural network," in *Proc. of the International Joint Conference on Neural Networks*, Brisbane, Australia, 2012, pp. 10–15.
[26] N. Neverova, C. Wolf, G. Paci, G. Sommavilla, G. W. Taylor, and F. Nebout, "A multi-scale approach to gesture detection and recognition," in *Proc. of the IEEE International Conference on Computer Vision*, Sydney, Australia, 2013, pp. 484–491.
[27] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training Recurrent Neural Networks," in *ICML'13 Proc. of the 30th Int. Conf. on Machine Learning*, Atlanta, USA, 2013, pp. 1310–1318.
[28] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[29] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling," *https://arxiv.org/abs/1412.3555*, pp. 1–9, 2014.
[30] S. Oprea, A. Garcia-Garcia, J. Garcia-Rodriguez, S. Orts-Escolano, and M. Cazorla, "A recurrent neural network based Schaeffer gesture recognition system," in *2017 Int. Joint Conf. on Neural Networks (IJCNN)*, Anchorage, USa, 2017, pp. 425–431.
[31] A. Géron, *Hands-On Machine Learning with Scikit-Learn & TensorFlow*. O'Reilly Media Inc., 2017.
[32] S. Shin and W. Sung, "Dynamic hand gesture recognition for wearable devices with low complexity recurrent neural networks," in *IEEE Int. Symp. Circ. and Sys. (ISCAS)*, Montreal, Canada, 2016, pp. 2274–2277.
[33] S. Otte, D. Krechel, M. Liwicki, and A. Dengel, "Local Feature Based Online Mode Detection with Recurrent Neural Networks," in *2012 Int. Conf. on Frontiers in Hand. Rec.*, Barti, Italy, 2012, pp. 533–537.
[34] E. Tsironi, P. Barros, C. Weber, and S. Wermter, "An analysis of Convolutional Long Short-Term Memory Recurrent Neural Networks for gesture recognition," *Neurocomputing*, vol. 268, pp. 76–86, 2017.
[35] X. Chai, Z. Liu, F. Yin, Z. Liu, and X. Chen, "Two Streams Recurrent Neural Networks for Large-Scale Continuous Gesture Recognition," in *2016 23rd Int. Conf. on Pat. Rec. (ICPR)*, Cancun, Mexico, 2016.
[36] K. Czuszynski, J. Ruminski, and J. Wtorek, "Analysis of the properties of the active linear gesture sensor," *Metrol. Meas. Syst.*, vol. 24, no. 4, pp. 617–630, 2017.
[37] K. Czuszynski, J. Ruminski, and J. Wtorek, "The passive operating mode of the linear optical gesture sensor," *Adv. Electr. Comput. Eng.*, vol. 18, no. 1, pp. 145-156, 2018.
[38] K. Czuszynski, J. Ruminski, and J. Wtorek, "Pose classification in the gesture recognition using the linear optical sensor," in *Human System Interactions (HSI), 2017 10th Int. Conf. on*, Ulsan, Korea, 2017, pp. 18–24.
[39] J. O. Wobbrock, M. R. Morris, and A. D. Wilson, "User-defined gestures for surface computing," in *Proc. of the 27th Int. Conf. on Human factors in comp. sys. - CHI 09*, Boston, USA, 2009, pp. 1083–1092.
[40] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay, "Scikit-learn: Machine Learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2012.
[41] J. Bergstra and Y. Bengio, "Random Search for Hyper-Parameter Optimization," *J. Mach. Learn. Res.*, vol. 13, pp. 281–305, 2012.
[42] G. Klambauer, T. Unterthiner, A. Mayr, and S. Hochreiter, "Self-Normalizing Neural Networks," https://arxiv.org/abs/1706.02515, 2017.
[43] T. Shen, T. Zhou, G. Long, J. Jiang, S. Pan, and C. Zhang, "DiSAN: Directional Self-Attention Network for RNN/CNN-free Language Understanding," *https://arxiv.org/abs/1709.04696*, 2017.
[44] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang, "On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima," *https://arxiv.org/abs/1609.04836*, pp. 1–16, 2016.
[45] J. Bergstra, B. Komer, C. Eliasmith, D. Yamins, and D. D. Cox, "Hyperopt: A Python library for model selection and hyperparameter optimization," *Comput. Sci. Discov.*, vol. 8, 2015
[46] K. Czuszynski, J. Ruminski, A. Bujnowski, and J. Wtorek, "Semi complex navigation with an active optical gesture sensor," in *UbiComp '16 Proc. of the 2016 ACM Int. Joint Conf. on Pervasive and Ubiquitous Computing: Adjunct*, Heidelberg, Germany, 2016, pp. 269–272.

Krzysztof Czuszyński received a M.Sc. Eng. degree in Biomedical Engineering at the Department of Electronics Telecommunications and Informatics (2012) and an M.Sc. degree in Applied Informatics in Management at the department of Management and Economics (2015), both at Gdansk University of Technology. His main research activities refer to gesture sensors for human system interactions in mobile devices, for which he employs machine learning techniques.

Jacek Rumiński received a M.Sc. degree in Electronics, a Ph.D. degree in Computer Science, and habilitation in Biocybernetics and Biomedical Engineering. He is a professor at Gdansk University of Technology. He has been either a coordinator or an investigator in about 20 projects receiving a

number of awards, including four best papers, practical innovations (7 medals and awards) and also the Andronicos G. Kantsios Award. He is the author of about 210 papers, and several patent applications and patents. Recently he was a main coordinator of the European eGlasses project focused on human-system interaction using smartglasses. His research is mainly focused on the application of machine learning in healthcare.

Alicja Kwaśniewska received a M.Sc. Eng. degree in Biomedical Engineering at the Department of Electronics Telecommunications and Informatics (2015) at Gdansk University of Technology. Her M.Sc. Thesis was developed in cooperation with the Norwegian University of Science and Technology in Trondheim, Norway. In her studies, she is primarily focused on machine learning with a particular application of image processing and convolutional neural networks for thermal images.