AirScript - Creating Documents in Air

Ayushman Dash[*‡], Amit Sahu[*‡], Rajveer Shringi[*‡], John Gamboa[§]
Muhammad Zeshan Afzal[§], Muhammad Imran Malik[†], Sheraz Ahmed[†] and Andreas Dengel[†]

[‡]Knowledge-Based Systems Group,
Department of Computer Science, University of Kaiserslautern,
P.O. Box 3049, 67653 Kaiserslautern, Germany
[§]MindGarage, University of Kaiserslautern, Germany
[†]German Research Center for AI (DFKI)
Knowledge Management Department,
Kaiserslautern, Germany

Abstract—This paper presents a novel approach, called AirScript, for creating, recognizing and visualizing documents in air. We present a novel algorithm, called 2-DifViz, that converts the hand movements in air (captured by a Myo-armband worn by a user) into a sequence of x, y coordinates on a 2D Cartesian plane, and visualizes them on a canvas. Existing sensor-based approaches either do not provide visual feedback or represent the recognized characters using prefixed templates. In contrast, AirScript stands out by giving freedom of movement to the user, as well as by providing a real-time visual feedback of the written characters, making the interaction natural. AirScript provides a recognition module to predict the content of the document created in air. To do so, we present a novel approach based on deep learning, which uses the sensor data and the visualizations created by 2-DifViz. The recognition module consists of a Convolutional Neural Network (CNN) and two Gated Recurrent Unit (GRU) Networks. The output from these three networks is fused to get the final prediction about the characters written in air. AirScript can be used in highly sophisticated environments like a smart classroom, a smart factory or a smart laboratory, where it would enable people to annotate pieces of texts wherever they want without any reference surface. We have evaluated AirScript against various well-known learning models (HMM, KNN, SVM, etc.) on the data of 12 participants. Evaluation results show that the recognition module of AirScript largely outperforms all of these models by achieving an accuracy of 91.7% in a person independent evaluation and a 96.7% accuracy in a person dependent evaluation.

I. INTRODUCTION

In the last few decades, the definition of document has been the topic of significant discussion. Some authors have restricted documents to "things that we can read" [1], while others have extended it to anything that *functions* as a source of evidence [2]. We consider a document as any resource for furnishing information evidence or proving the information authenticity[2]. However, the value of documents "cannot be fully estimated by just looking at their contents"[3] as the activities related to the documents provide key information related to them. These discussions are important because they

have implications on the strategies that can be used to generate documents in air.

During the Information Age, the media where documents are created has undergone a fast transition from traditional paper-based methods to any digital device. Documents are nowadays created in laptops, PCs and smartphones, by means of text editors and drawing tools, or alternatively generated in real-time on flat surfaces able to perform handwriting recognition. However, despite the progress, all of these methods are limited in that they restrict the region where the input is received to a given surface of reference.

Fig. 1. A smart classroom scenario that we envision, where AirScript can be used for writing in air by just wearing a Myo-armband. The images on the left show how a person writes in air (in our case, digits). The numeric label in orange represents the sequence of hand movements. While the person writes in air, the Myo-armband captures raw IMU and EMG signals from the arm and sends it to AirScript, running on a digital device. AirScript gives a realistic visual feedback to the user in real time on that device, showing what the user wrote in air. It also recognizes the written digit, giving possible suggestions for it.

In this article we introduce AirScript, a novel approach for document creation in air, whereby a sensor device is attached to the user's arm, capturing its movements. This way, we eliminate the dependency on a reference surface, overcoming a major drawback of the previous methods. Our method has the potential to enable people to annotate pieces of texts wherever they want, with complete freedom of movement. It could be

---

[*]These authors contributed equally to this work.

[2]Keynote speech by Andreas Dengel (DAS 2014): https://das2014.sciencesconf.org/resource/page/id/5

[3]Keynote speech by Koichi Kise (ICDAR 2015): http://www.iapr.org/archives/icdar2015/index.html%3Fp=372.html
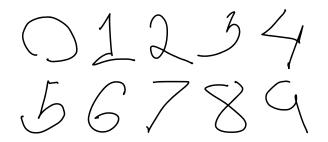
Fig. 2. Visualization of the hand movements for the digits that were drawn in air. Section IV-B describes how the hand movements were converted into a sequence of x, y coordinates, represented in the form of images.

used in highly sophisticated environments, such as a smart classroom or a smart factory. The recognized content could then be easily displayed on a board or any other canvas if the user so likes.

AirScript is composed of two modules. Its visualization module, **2-DifViz**, projects the multi-dimensional sensor data onto a 2D surface, producing a realistic visualization of the input. This visualization is then used, along with the raw sensor data, by AirScript's **recognition module** to predict the content of the hand movement. We developed our proof of concept using a Myo-armband, as described in Section IV-A. Finally, our recognition module performs Handwritten Digits Recognition in Air (HDRA). The data sensed by the Myo-armband is fed into a Gated Recurrent Unit (GRU) network. 2-DifViz produces differential features that are fed to another GRU Network, as well as used to produce visualizations that are fed to a Convolutional Neural Network (CNN). The result of the three networks is then fused to produce a prediction of the digit written in air. It was found that this fusion made the classification results more robust when testing the model with several participants. Figure 1 shows an example scenario, in which we envision that our approach could be employed. A demo video of the = is also available.[4]

To gauge the performance of AirScript, it is evaluated on HDRA data of 12 different participants. Evaluation results reveal that AirScript achieves an average accuracy of 96.7% on a person dependent evaluation, and 91.7% on a person independent evaluation. Furthermore, the visualization of hand movements produced by AirScript are also very realistic, which makes it suitable for using in real scenarios. Figure 2 shows the visualization of the hand movements of the participants while they were drawing the digits in air. This not just shows the potential of the proposed HDRA model but also makes it evident that the recognized digits can also be converted into a reproducible human readable format.

II. RELATED WORK

A lot of work has been done on extending and simplifying the process of creating documents. Examples include using gesture-based input control [3]–[6], swipe-based input methods [7], voice-based input methods [8] and even interaction

[4]https://drive.google.com/file/d/0B5xrMEupo2dPbEk4ekNVOUhMOWs

methods on imaginary surfaces [9]. Specially relevant to this work are methods that create a Virtual Reality[5] environment where the user explicitly inputs information. Because of their Virtual Reality nature, these methods are inadequate in environments where the user has to interact with real world objects, such as a smart factory or a smart class. Our approach differs from these methods in that the user remains in the real world.

Handwriting Recognition in Air (HWRA) has been performed by Amma et al. [10], using a prototype glove with an embedded IMU sensor, showing promising results. They combine a Hidden Markov Model (HMM) with a language model, and achieve a word error rate of 11% on a person independent evaluation and 3% on a person dependent evaluation. However, their method does not give any visual or haptic feedback to the user. Alternative computer vision-based approaches relying on finger tracking[6] or multi-camera 3D hand tracking have been used for HWRA [11]–[13] but face problems similar to those of the finger tracking approaches. These methods are dependent on a tracking device that has to be on the line of sight, restricting the freedom of movement of the user. To our knowledge, Deep Learning methods have not been explored for HWRA.

In opposition to HWRA, handwriting recognition on surface using Deep Learning models like Bidirectional LSTMs [14], Connectionist Temporal Classifiers [15], and Multidimensional RNNs [16] have outperformed other baseline models (*e.g.*, [17]). Similar Deep Learning models, such as Convolutional Neural Networks [18] and LSTMs [19], have also shown improved results in the domain of gesture recognition.

III. DATA ACQUISITION

To train and test the performance of AirScript, a dataset was collected from 12 right handed participants while they drew digits in air. The recording includes raw IMU and EMG signals from a Myo-armband as well as their ground truth labels. For data acquisition and visualization, a complete Graphical User Interface based solution called Pewter[7] was developed.

The Myo-armband has an Inertial Measurement Unit (IMU) that consists of a 3-dimensional accelerometer that measures the non-gravity acceleration, a 3-dimensional gyroscope that measures the angular momentum, and a magnetometer that measures orientation w.r.t. the Earth's magnetic field. A 10-dimensional vector $\mathcal{M}$ is acquired from the IMU at a sampling rate of $50Hz$. Each of these sensors acts like a function $f : T \to \mathbb{R}$ that maps a timestep to a real value. For a duration of time, they form a sequence (represented as vector) of real values corresponding to the digit written in that time.

The participants were asked to wear the device on the right arm in accordance with the Myo-armband instructions[8]. To avoid fatigue and priming, the data for each digit was collected

[5]For example, https://www.tiltbrush.com/
[6]For example, https://www.leapmotion.com/
[7]https://github.com/sigvoiced/pewter
[8]https://s3.amazonaws.com/thalmicdownloads/information+guide/important-information-guide-v03.pdf

in three phases. In Phase-I and Phase-II, 3 iterations of every digit were conducted. Phase-III consisted of 4 iterations.

Due to unusual vibrations in the Myo-armband during the data collection process, 30 data samples were visualized using Pewter and removed manually from the dataset. Hence, the final dataset contained 1270 samples in total.

IV. AirScript: The Presented Approach

To generate documents in air we propose a two phase process that breaks down the problem into two different tasks:

1) **Phase-I (2-DifViz)**: the hand movements are converted into a realistic visualization of the digit written in air.
2) **Phase-II (HDRA)**: handwritten digits in air are recognized using a fused classifier.

Figure 3 shows the complete workflow of AirScript, in which the raw IMU data from the Myo-armband is processed using 2-DifViz and a signal standardization pipeline. The processed data is then fed to a recognition module consisting of three classifiers. Each of these classifiers provides a list of ranked results. These are then fused and a final prediction is generated. Because the mistakes committed by the three classifiers are generally different, this fusion was found to improve the robustness of the model.

A. Myo-Armband

The Myo-armband is an unobtrusive sensor device easily available and integrable to several platforms through its off-the-shelf SDKs. Its Inertial Measurement Unit (IMU) senses the orientation, acceleration and angular velocity of the arm at any given moment. Additionally, the arm's muscle activity is captured by 8 Electromyography (EMG) pods embedded in the device.

Let $\mathcal{D} = \{d_i \mid i = 1, ..., n\}$ represent our HDRA dataset, where n is the number of data instances in $\mathcal{D}$ (in our case 1270). Each data instance $d_i = (\mathcal{M}_i, \mathcal{E}_i, L_i)$ is a tuple consisting of a time-series $\mathcal{M}_i$ representing the IMU sensor data, a time-series $\mathcal{E}_i$ representing the EMG sensor data, and a class label $L_i \in \{0, ..., 9\}$. For our models we use only the IMU sensor data. Every $\mathcal{M}_i = \{\mathcal{M}_i^{(1)}, \mathcal{M}_i^{(2)}, ..., \mathcal{M}_i^{(\tau_i)}\}$ is a time-series consisting of τ_i time-steps (*i.e.*, $|\mathcal{M}_i| = \tau_i$), and each element $\mathcal{M}_i^{(t)} = \begin{bmatrix} \mathbf{a}_i^t & \mathbf{g}_i^t & \mathbf{q}_i^t \end{bmatrix}$. The vectors $\mathbf{a}_i^t \in \mathbb{R}^3$ and $\mathbf{g}_i^t \in \mathbb{R}^3$ are the 3 axes of the accelerometer and the gyroscope, respectively. Similarly, $\mathbf{q}_i^t \in \mathbb{R}^4$ denotes a quaternion representing the orientation.

B. 2D Differential Visualization (2-DifViz)

To generate realistic and reproducible visualizations of the handwritten digits in air, we developed a method called 2-Dimensional Differential Visualization (**2-DifViz**). We use the set of steps below[9] to get coordinate sequences C_i, henceforth referred to as *2-DifViz features*. These sequences are then plotted on a 2D canvas and are interpolated to smoothen the curve and make the visualizations continuous.

[9]This pipeline was built upon the mouse controller application (http://developerblog.myo.com/build-your-own-mouse-control-with-myo/) developed by Thalmic Labs (https://www.thalmic.com/).

- **STEP-1 (Rotate Frame of Reference)**: $\mathbf{g}_i^t$ holds the angular velocity of the user's arm in degrees per second in the three dimensions: dx (*pitch*), dy (*yaw*) and dz (*roll*). These values are in the frame of reference of the arm on which the Myo-armband is worn. We assume that the digits were written on an imaginary canvas in air, to which we refer as "the world frame of reference", and hence we rotate $\mathbf{g}_i^t$ and bring it to "the world frame of reference". Let $(\mathbf{q}_i^t)^{-1}$ denote the inverse of $\mathbf{q}_i^t$. The rotated vector is therefore:

$$\hat{\mathbf{g}}_i^t = \mathbf{q}_i^t \star \mathbf{g}_i^t \star (\mathbf{q}_i^t)^{-1} \tag{1}$$

where $\mathbf{g}_i^t$ is reinterpreted as a quaternion whose real coordinate is 0, $\hat{\mathbf{g}}_i^t$ is the rotated gyroscope vector, and $\star$ denotes the Hamilton product.

- **STEP-2 (Extract Pitch and Yaw)**: we construct the vector $g_i^t = (dx_i^t, dy_i^t)$ from $\hat{\mathbf{g}}_i^t$, where each dx denotes the *pitch* and each dy denotes the *yaw* in a given time-step. We ignore the *roll* as we are concerned with mapping the hand movements to a 2-dimensional vector sequence that we can visualize as realistic digits.

- **STEP-3 (Determine the Gain)**: a gain factor K is calculated, that maps the arm movements to pixels on the imaginary canvas. K is determined by hyperparameters like *sensitivity*, *acceleration* and *pixel density* and acts as a scaling factor for g_i^t.

- **STEP-4 (Calculate Sequence of Differentials)**: g_i^t is multiplied with K and a frame duration F to scale the hand movements on a 2-dimensional canvas and smoothen the transitions on it.

$$\tilde{g}_i^t = g_i^t \times K \times F \tag{2}$$

Here $\tilde{g}_i = \{\tilde{g}_i^1, \tilde{g}_i^2, ..., \tilde{g}_i^{\tau_i}\}$ is a sequence of 2-dimensional vectors that contains the number of pixels to move on the imaginary canvas at every time-step. Since $\tilde{g}_i^t$ consists of pixels, dx and dy are converted into integers.

- **STEP-5 (Create Coordinate Sequences)**: a sequence of 2-dimensional coordinates $C_i^t = \{(x_i^1, y_i^1), ..., (x_i^{\tau_i}, y_i^{\tau_i})\}$ is created from $\tilde{g}_i^t$, where x and y are coordinates on the horizontal and vertical axis of a 2-dimensional Cartesian plane, respectively, and $|C_i| = \tau_i + 1$. To create $C_i^{(t)}$ we start by setting $C_i^1 = (0, 0)$. Then, $\forall t \in [1, ..., \tau_i]$ and $\tilde{g}_i^t = (dx_i^t, dy_i^t)$ we set $C_i^{t+1} = (x_i^t + dx_i^t, y_i^t + dy_i^t)$.

Figure 2 shows visualizations of handwritten digits in air using 2-DifViz. This canvas is then stored as an SVG or PNG file. These visualizations were used to create a set $I = \{I_i \mid i = 1, ..., n\}$ of images, where n is the number of instances in the dataset.

C. GRU Networks

Since Recurrent Neural Networks (RNN) using LSTM have shown state-of-the-art performance for handwriting recognition [14]–[16], we chose to use a similar architecture with a variant of LSTM called Gated Recurrent Units (GRU) for HDRA. RNNs with GRU have fewer parameters than LSTM and their performance is at par with LSTM Networks [20].
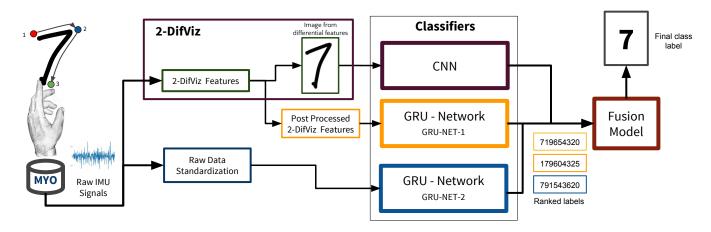
Fig. 3. The architecture of the proposed HDRA fusion model. The colored dots on the top-left show the sequence of hand movements. The red dot represents the starting point, the blue dot represents an intermediate point and the green dot represents the end of the hand movement. A sequence of x, y coordinates representing the user's hand movements is extracted from the raw IMU signals using 2-DifViz. These features are projected onto a 2D plane, forming the digit the user drew, and fed to a Convolutional Neural Network (CNN). They are also post-processed and fed to a Gated Recurrent Unit (GRU) Network. Additionally, the raw signals are standardized and fed to a separate GRU Network. Finally, the output of the three classifiers is fused to produce the final prediction. 2-DifViz is the Visualization module of AirScript (Phase-I), and the CNN, GRU-NET-1 and GRU-NET-2 compose its recognition module (Phase-II).

RNNs are able to learn from sequential data and incorporate contextual information, making them a best fit for the HDRA task.

GRUs use gating units as follows [20]:

$$h_t^j = (1 - z_t^j) h_{t-1}^j + z_t^j \tilde{h}_t^j \tag{3}$$

$$z_t^j = \sigma \left(W_z x_t + U_z h_{t-1} \right)^j \tag{4}$$

$$\tilde{h}_t^j = \tanh \left(W x_t + r_t \odot (U h_{t-1}) \right)^j \tag{5}$$

$$r_t^j = \sigma \left(W_r x_t + U_r h_{t-1} \right)^j \tag{6}$$

The activation h_t^j (Equation (3)) of a GRU at time t is a linear interpolation between the previous activation h_{t-1}^j and the candidate activation $\tilde{h}_t^j$, computed by Equation (5), where r_t is a set of reset gates and $\odot$ is an element-wise multiplication. The *update gate* z_t^j decides how much the GRU updates its content, according to Equation (4). Finally, the reset gate r_t^j controls how much of the previously computed state to *forget* and is computed by Equation (6).

The formation of digit in air is dependent on the past as well as the future context and needs to be classified only after the whole digit has been formed. Therefore, we further extended the GRU by combining it with a Bidirectional RNN [21] resulting in a Bidirectional GRU, known to give better results than unidirectional RNNs [22]. Two GRU networks were trained using the following architectures:

1) BGRU Network (GRU-NET-1): The 2-DifViz features (*i.e.,*, C_i) are post-processed in the following way:

- **STEP-1 (Smoothing)**: to remove noisy points, each C_i^t was smoothed by averaging adjacent points using the following equation [23]:

$$\hat{C}_i^t = \left(\frac{x_i^{t-2} + \cdots + x_i^{t+2}}{5}, \frac{y_i^{t-2} + \cdots + y_i^{t+2}}{5} \right)$$

- **STEP-2 (Redundancy Removal)**: points too close to each other can convey noise in the direction of movement of a stroke [23]. The following equation was used to calculate a threshold Δ, and adjacent points with $\Delta \leq 5$ were removed:

$$\Delta = \sqrt{(x_i^t - x_i^{t-1})^2 + (y_i^t - y_i^{t-1})^2}$$

- **STEP-3 (Standard Scaling)**: C_i^t is whitened by scaling its values by its mean and standard deviation.

$$\hat{C}_i^t = \frac{C_i^t - \mu_c}{\sigma_c} \tag{7}$$

Where μ_c and σ_c are the mean and standard deviation of C_i, respectively. This helps in removing the offset of the sequences by making $\mu = 0$ and normalizes the fluctuation by making $\sigma = 1$.

- **STEP-4 (Interpolation)**: all coordinate sequences C_i are linearly interpolated to the same length $|\hat{C}_i| = 100$

The newly generated $\hat{C}_i$ are used to train a 1-layer BGRU network with 32 output units with a sigmoid activation function. A softmax output layer was used with 10 units for the 10 digits. We trained the network using Stochastic Gradient Descent (SGD) over the training data for 150 epochs with a categorical crossentropy loss and the Adam optimizer with a learning rate of 0.001, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$ and decay of 10^{-6}.

2) BGRU Network (GRU-NET-2): All the raw $\mathcal{M}_i$ are standardized in the following way:

- **STEP-1 (Absolute Scaling)**: all $\mathcal{M}_i^t$ are scaled to a range of $[-1, 1]$.
- **STEP-2 (Resampling)**: a set $T = \{\tau_i \mid 1, ..., n\}$ is defined such that $n = |\mathcal{D}|$ and τ_i is the number of time-steps in $\mathcal{M}_i$. Let t_{max} be the maximum of all values in

T. For each d_i, the time-series $\mathcal{M}_i$ is resampled such that $|\mathcal{M}_i| = t_{max}$. This is done to normalize the length of the sequences and remove jitter.

We trained a 1-layer BGRU on the standardized $\mathcal{M}_i$. The number of units, activation, output layer, loss and optimizer used were the same as the GRU-NET-1.

D. Convolutional Neural Network

Convolutional Neural Networks (CNN) have shown state-of-the-art results in classifying images [24]–[26]. Since we were able to convert the handwritten digits in air into realistic visualizations (images), we could reduce the HDRA problem to an image recognition problem. CNNs have outperformed other existing models for the MNIST digit recognition task [24] so we chose to use CNNs for HDRA using 2-DifVis. We used a transfer learning approach [27] for re-training a pre-trained CaffeNet[10] [28] on the image set I. The output layer of the CaffeNet was replaced with a softmax layer with 10 units and the network was re-trained using the Adam optimizer with a learning rate of 0.0001, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$ using a categorical cross entropy loss.

E. Fusion Model

By using GRU-NET-1 and GRU-NET-2 we were able to capture the temporal information from the preprocessed co-ordinate sequences and IMU data. With the CNN we could acquire a spatial representation of images generated by 2-DifViz. This motivated us to extend our model and fuse these three modalities to capture the spatio-temporal representation of the raw IMU data. To do so, we fused the ranked results using Borda Count, whereby we extracted the ranked class labels from the three classifiers and decided a final class label for the input. Since the modalities are independent of each other, the sources of errors are independent too. This makes our fusion model robust and a best fit for the HDRA task.

V. EVALUATION

We evaluated our method using a person dependent test and a person independent test, as described in the sections below. To benchmark the components of the fusion model we compared their accuracies with a Hidden Markov Model (HMM), a Support Vector Machine (SVM), a Naive Bayes (NB) classifier and a K-Nearest Neighbor (KNN) classifier.

A. Person Dependent Test

To perform a person dependent test, the data from a single participant is used to train and evaluate the model. We split the person's data into 5 stratified folds. Each fold consists of a training set and a test set. We repeat this process for 10 randomly selected participants. We train and evaluate all our models on all the folds from each of the selected participants. The average accuracy on the 5 folds for each participant is recorded and a mean of these averages is calculated, as well as their standard deviation. Table I shows the results for all the evaluated models.

[10]https://github.com/BVLC/caffe/tree/master/models/bvlc-reference-caffenet

Classifier	Avg. Mean Accuracy (%)	Std. deviation
HMM	75	22.9
KNN	77.5	16.9
NB	75.3	21.3
SVM	81.67	13.9
CNN	95.1	5.9
GRU-NET-1	84.4	13.4
GRU-NET-2	88.7	10.6
Fusion Model	**96.7**	**0.02**

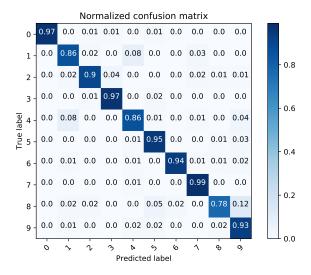| Classifier	Mean Accuracy (%)	St. Deviation
HMM	15.8	5.8
KNN	13.6	8.8
NB	31.7	13.6
SVM	19.7	13.8
CNN	84.6	11.2
GRU-NET-1	67.4	10
GRU-NET-2	87.6	10.4
Fusion Model	**91.7**	**0.06**

Fig. 4. Confusion matrix for HDRA using the fusion model.

B. Person Independent Test

We use this test to evaluate the robustness of a classifier, independently of any specific person. In this setting, we withhold the data of a randomly selected participant to be used as a test set, while the rest of the data is used as a training set. This is repeated for 10 different participants and the average accuracy is recorded. Table II shows the average accuracy of all the evaluated models.

C. Analysis

1) Performance Analysis: In both the person dependent and person independent tests, the fusion model outperformed all the baseline models with a substantial margin. The CNN,

GRU-NET-1 and GRU-NET-2 were the top 3 classifiers considering the average accuracy. Thus, we chose to use them in our fusion model, which resulted in an improvement in accuracy. Even though the CNN, GRU-NET-1 and GRU-NET-2 had an accuracy of more than 80%, they had a standard deviation of more than 10. However, fusing them together resulted in a drop in standard deviation, making the fusion model much more robust.

We observed that the results were better in the person dependent test than the person independent test. This behavior can be attributed to each person having different speed and style of writing the same digits.

2) Digit Analysis: We analyzed the confusion matrices of the fusion model and its three components individually. For doing so, we relied on the 2-DifViz visualizations of the hand movements of the participants. This analysis helped us in improving the classifiers and observing the subtle problems in digit recognition. The confusion matrix of the fusion model is shown in Fig. 4. The confusion matrices of the different classifiers allowed us to have an insight on what errors they were committing. For instance, digits 1 and 2, as well as 1 and 4 were confused in GRU-NET-2, possibly because of similar initial hand motion. Similarly, the CNN tended to confuse digits with loops (like 6, 8 and 9). Fusing the models together allowed us to overcome these difficulties and achieve a higher robustness.

VI. CONCLUSION AND FUTURE WORK

We introduce a novel approach called AirScript for creating documents in air using a Myo-armband. For doing so, we split the problem into a visualization task (2-DifViz) and a handwriting recognition task (HDRA). We show a proof of concept for the latter by proposing a classifier fusion model which achieves a recognition rate of 91.7% on a person independent evaluation, and 96.7% on a person dependent evaluation. For the visualization task we introduce a new method called 2-DifViz, which converts the hand movements into realistic visualizations on a 2D canvas of the digits written in air, that can be stored in an SVG or PNG format. This shows the potential use of AirScript in many application areas such as, smart factories, smart offices, smart class rooms, virtual reality games and even augmented reality environments. We envision AirScript to be used in a smart classroom environment using augmented reality, where people can scribble anything as air notes and visualize these notes in the form of handwriting, thus giving the process of creating a new definition. AirScript uses a Myo-armband, our method is mobile and easy to integrate on multiple platforms, while still providing the user with freedom of movement.

We plan to extend AirScript by adding a handwriting recognition model using Sequence to Sequence models or LSTM Networks with Connectionist Temporal Classifiers, along with a language model. We plan to use Deep Generative Models for improving the visualizations of handwriting in air and making them more realistic.

REFERENCES

[1] K. Macdonald, "Using documents," *Researching social life*, p. 194, 2001.

[2] M. Buckland, "What is a digital document," *Document numérique*, vol. 2, no. 2, pp. 221–230, 1998.

[3] S. Kumar and J. Segen, "Gesture-based input interface system with shadow detection," Sep. 23 2003, uS Patent 6,624,833.

[4] E. Tamaki, T. Miyaki, and J. Rekimoto, "Brainy hand: an ear-worn hand gesture interaction device," in *CHI'09 Extended Abstracts on Human Factors in Computing Systems*. ACM, 2009, pp. 4255–4260.

[5] P. Mistry, P. Maes, and L. Chang, "Wuw-wear ur world: a wearable gestural interface," in *CHI'09 extended abstracts on Human factors in computing systems*. ACM, 2009, pp. 4111–4116.

[6] J. Tong, J. Zhou, L. Liu, Z. Pan, and H. Yan, "Scanning 3d full human bodies using kinects," *IEEE transactions on visualization and computer graphics*, vol. 18, no. 4, pp. 643–650, 2012.

[7] W. C. Westerman, H. Lamiraux, and M. E. Dreisbach, "Swipe gestures for touch screen keyboards," Nov. 15 2011, uS Patent 8,059,101.

[8] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates *et al.*, "Deep speech: Scaling up end-to-end speech recognition," *arXiv preprint arXiv:1412.5567*, 2014.

[9] S. Gustafson, D. Bierwirth, and P. Baudisch, "Imaginary interfaces: spatial interaction with empty hands and without visual feedback," in *Proceedings of the 23nd annual ACM symposium on User interface software and technology*. ACM, 2010, pp. 3–12.

[10] C. Amma, M. Georgi, and T. Schultz, "Airwriting: a wearable handwriting recognition system," *Personal and ubiquitous computing*, vol. 18, no. 1, pp. 191–203, 2014.

[11] A. Schick, D. Morlock, C. Amma, T. Schultz, and R. Stiefelhagen, "Vision-based handwriting recognition for unrestricted text input in mid-air," in *Proceedings of the 14th ACM international conference on Multimodal interaction*. ACM, 2012, pp. 217–220.

[12] M. Chen, G. AlRegib, and B.-H. Juang, "Air-writing recognitionpart i: Modeling and recognition of characters, words, and connecting motions," *IEEE Transactions on Human-Machine Systems*, vol. 46, no. 3, pp. 403–413, 2016.

[13] S. Vikram, L. Li, and S. Russell, "Handwriting and gestures in the air, recognizing on the fly," in *Proceedings of the CHI*, vol. 13, 2013, pp. 1179–1184.

[14] M. Liwicki, A. Graves, H. Bunke, and J. Schmidhuber, "A novel approach to on-line handwriting recognition based on bidirectional long short-term memory networks," in *Proc. 9th Int. Conf. on Document Analysis and Recognition*, vol. 1, 2007, pp. 367–371.

[15] A. Graves, M. Liwicki, S. Fernández, R. Bertolami, H. Bunke, and J. Schmidhuber, "A novel connectionist system for unconstrained handwriting recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, no. 5, pp. 855–868, 2009.

[16] A. Graves and J. Schmidhuber, "Offline handwriting recognition with multidimensional recurrent neural networks," in *Advances in neural information processing systems*, 2009, pp. 545–552.

[17] M. Liwicki and H. Bunke, "Hmm-based on-line recognition of handwritten whiteboard notes," in *Tenth international workshop on frontiers in handwriting recognition*. Suvisoft, 2006.

[18] F. J. Ordóñez and D. Roggen, "Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition," *Sensors*, vol. 16, no. 1, p. 115, 2016.

[19] D. Arsićc, L. Roalter, M. Wöllmer, F. Eyben, B. Schuller, M. Kaiser, M. Kranz, and G. Rigoll, "3d gesture recognition applying long short-term memory and contextual knowledge in a cave," in *Proceedings of the 1st ACM international workshop on Multimodal pervasive video analysis*. ACM, 2010, pp. 33–36.

[20] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[21] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[22] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional lstm and other neural network architectures," *Neural Networks*, vol. 18, no. 5, pp. 602–610, 2005.

[23] W.-L. Jiang, Z.-X. Sun, B. Yuan, W.-T. Zheng, and W.-H. Xu, "User-independent online handwritten digit recognition," in *Machine Learning and Cybernetics, 2006 International Conference on*. IEEE, 2006, pp. 3359–3364.

[24] L. Wan, M. Zeiler, S. Zhang, Y. L. Cun, and R. Fergus, "Regularization of neural networks using dropconnect," in *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, 2013, pp. 1058–1066.

[25] B. Graham, "Fractional max-pooling," *arXiv preprint arXiv:1412.6071*, 2014.

[26] J.-R. Chang and Y.-S. Chen, "Batch-normalized maxout network in network," *arXiv preprint arXiv:1511.02583*, 2015.

[27] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[28] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the 22nd ACM international conference on Multimedia*. ACM, 2014, pp. 675–678.