Deep Learning Approach to Modeling a Sensorimotor Skill of Periodontal Probing

Project Report for ECE-GY 9123 Deep Learning Course

GitHub Repository: https://github.com/vbabushkin/ECE-GY-9123_DEEP_LEARNING

Vahan Babushkin

Abstract

A sensorimotor skill can be defined as a sequence of continuous motions produced by the human body in response to external stimuli. Periodontal probing is an example of a sensorimotor skill, that involves the periodic insertion of a periodontal probe between the tooth and gingiva with the purpose of diagnosing a periodontal disease (periodontitis). The goal of the project is to study the professional sensorimotor skill of periodontal probing from the Deep Learning perspective using the dental professional's recordings collected with the VR dental simulation system. In this project, we develop CNN and LSTM models that characterize the sensorimotor skill of periodontal probing by detecting which region of the tooth is probed. We also investigate how far the proposed models are capable of identifying the periodontal pockets. Finally, we discuss the possible challenges and provide a roadmap for future work.

Index Terms

Deep Learning, Artificial Neural Networks, Machine Learning, Haptics and Haptic Interfaces, Sensorimotor Learning, Virtual Reality and Interfaces

I. Project Description

THE focus of the project is creating a model of the sensorimotor skill of periodontal probing. The periodontal probing task consists of measuring the depth of the gingival sulcus (space between the tooth and surrounding tissues) with a special dental tool called a probe, to detect the periodontal pockets – where deterioration of gingiva occurred due to the development of periodontal disease (inflammation of gingival sulcus caused by bacterial infection). The probing procedure requires an accurate orientation and rotation of the probe to ensure its tip reaches the bottom of the gingival sulcus following the shape of the root of the tooth. The dental specialist estimates the depth of the pocket by looking at the circumferential markings on the probe. In the case of the healthy gingiva, the depth does not exceed 3 mm, deeper depths indicate the presence of a periodontal pocket. Thus the periodontal probing task is a complex sensorimotor skill that requires the proper positioning and rotation of the probe, and exertion of appropriate forces.

For this project, we consider the sensorimotor skill of periodontal probing from the Deep Learning perspective by developing a simple Convolutional Neural Network (CNN) and Long-Short Term Memory (LSTM) Network that are capable of determining the probing region of the tooth and detecting periodontal pockets in the probed region from the time series data of probe position, velocity, orientation, and exerted forces. The proposed approach could be used to design Virtual Reality simulators for dental training, capable to detect which region of the tooth has been probed, whether or not it contains pockets, measure their depth, and evaluate the performance of the dental student. In the future, the Deep Learning models can be used to evaluate the proficiency level of the user and if necessary, adapt to the user's skills and provide guidance. Another perspective direction is developing models that generate the skill path from the professional's recordings and evaluate how similar is the generated skill to the actual skill of a professional.

II. Previous Research

The Machine Learning algorithms for sensorimotor skill modeling are widely used in imitation learning for learning the skill trajectory. For example, the skill can be encoded at trajectory level

as a set of transition probabilities using the Hidden Markov Models (HMMs) and then replayed by the stochastic algorithm [1]. The probabilistic models consider the task as a whole which allows learning the common features of sensorimotor skill associated with the given task from multiple demonstrations [2]. The probabilistic models are also capable of evaluating the proficiency of the expert by estimating the number of repeated states and similarity of the transitions between different states – an expert performs the demonstration the way to ensure the variations in each state and keep the transitions between states as similar to each other as possible [3].

There are several applications of Artificial Neural Networks (ANN) based algorithms for sensorimotor skill modeling in imitation learning. The ANN learns a skill representation from the examples provided by a human instructor (expert). ANN utilizes incremental learning to extend the acquired skill to some variations of the same task [4]. This feature allows to embed ANNs in the artificial sensorimotor skill communication systems for performing simple sensorimotor tasks online, e.g. audio-visual tracking – the system, after learning from demonstration continues updating weights of the embedded network without further supervision utilizing auditory signals and information about color, luminance, and motion, thus increasing the recognition of the object [5].

The earliest studies referring to the implementation of ANNs in imitation learning focus on deploying the neural networks to construct the skill representation from the training data, e.g. Radial-Basis Function Networks (RBFs) [6] [4]. As universal approximators, RBFs are able to learn from the examples and support incremental learning due to their ability to utilize time-delays for processing spatio-temporal data [4]. The effectiveness of the RBF-based approach was demonstrated with tasks such as peg-into-hole insertion and opening a door [4].

The Deep Learning Networks have been used for the assessment of surgical skills by mapping multivariate motion data into three proficiency levels (novice, intermediate, expert) [7]. The developed deep model of surgical skill allows to directly process the multivariate time series without manual feature engineering. In this project, we adopted a similar concept of a deep model that learns the sensorimotor skill representation from the raw data. We consider the application of a proposed model for learning the sensorimotor skill of periodontal probing. To our best knowledge, there are no published studies describing applications of Machine Learning/Deep Learning for dental skill modeling. The first attempt to characterize the periodontal probing skill by detecting which region of the tooth has been probed from the haptics data has been done in [8] (available online). The SVM model trained on 11 recordings and tested on 12-th demonstrated relatively high accuracy in detecting the probed region (around 0.85) and also was able to identify the presence of the pockets (with recall around 0.65 and precision of 0.8). For this project, we replicate these results with the Deep Learning models (CNN, LSTM), compare these two approaches, discuss possible challenges and propose the direction of future work.

III. Experimental setup and dataset

The data collection is performed using the Haptodont periodontal simulation VR system which generates a bi-manual 3 DoF force feedback to simulate haptic interaction with the dental probe and mirror instruments displayed by Geomagic Touch device, a finger support mechanism to simulate fulcrum (Novint Falcon™), and a VR headset (Oculus Rift). A dental professional (dental professor from NYU college of dentistry) performed the periodontal probing task on a mandibular low jaw 3D model with rendered gingiva and teeth 32-28 on the lower right buccal, and 17-21 on the lower left lingual side of the mouth. A Microsoft speech recognition system (Windows Desktop Speech) was integrated into the VR interface, accepting two commands – "START" and "STOP" to mark the beginning and the end of the current stage of the probing task.

The raw data consists of 102 features recorded with the 500 Hz sampling rate and precision of 6 digits. However, most of the features are either repeated (e.g. global and local features are generally the same since the Haptic device is stable) or they are auxiliary (e.g. used to reconstruct and replay

the probe trajectory in case if the model was rotated). After preprocessing there are the following 24 features used in the model that act as skill descriptors:

- Device Local Position (x_{loc}, y_{loc}, z_{loc}),
- Gripper Angle (degrees),
- Gripper Angular Velocity (rad/sec),
- Device Local Linear Velocity (v_x^{loc}, v_y^{loc}, v_z^{loc}),
- Device Local Angular Velocity (ω_x^{loc}, ω_y^{loc}, ω_z^{loc}),
- Gripper Force,
- Device Local Force (F_x^{loc}, F_y^{loc}, F_z^{loc}),
- Local Rotation Matrix:
  ($R_{0,0}^{loc}$, $R_{1,0}^{loc}$,$R_{2,0}^{loc}$, $R_{0,1}^{loc}$, $R_{1,1}^{loc}$, $R_{2,1}^{loc}$, $R_{0,2}^{loc}$, $R_{1,2}^{loc}$, $R_{2,2}^{loc}$).

There are also columns for time (in sec) and a skill label (0 – no skill path, 1 – skill path recorded). This set of features can be reduced by using univariate feature selection methods, such as Mutual Information (MI) which is capable of capturing different kinds of dependencies between features and targets, and is invariant to data transformation [9].

The data were pre-processed to split each labeled region of interest into probing regions around each tooth: 3 on the lower-left lingual side: the mesial lingual (ML), lingual (L), distal lingual (DL), and 3 on the lower-right buccal side of the jaw: mesial buccal (MB), buccal (B), and distal buccal (DB). The unsupervised learning algorithm (k-means) was applied to the positional coordinates of the tooltip and then the boundaries of the extracted probing regions were corrected in consultation with dental professors from NYU College of Dentistry and UoT School of Dentistry. There are three labels per tooth resulting in 30 distinct classes – 15 for buccal and 15 for lingual (5 teeth $\times$ 3 regions of probing). The naming of categorical labels includes the tooth number and the probing region, e.g. 32_DB meaning the distal buccal region of tooth #32. Usually, there are around 10000 timesteps per each region of probing in one recording.

IV. Deep Learning Approaches

Since the periodontal probing skill path is a set of positional, rotational, and force components measured at the given timepoint, the problem of classifying the probing regions or determining the presence of pockets can be regarded as time series classification. The deep learning approach to time series classification creates a mapping between the input space to the probability distribution over the class labels [10]. In this project due to time and space constraints, we decided to focus on two main approaches – CNN and LSTM.

1) Convolutional Neural Networks (CNNs): The ability of CNNs [11] to automatically extract the features by creating hierarchies of abstractions of the data with progressively increasing complexity will be useful for understanding the skill, i.e. to determine which features constitute the skill. The problem of determining which region is probed can be reduced to the problem of time series analysis with CNN by sliding the window of length w over the timecourse of the path for each region [12] [7]. In this case, the training data containing N timepoints each of f features will be split into $d = \lfloor N/w \rfloor$ instances of size $w \times f$ that can be treated as independent inputs, i.e. the tensor of size $d \times w \times f$ tensor will be fed to the input layer. The 1D CNN [10] might also be a suitable choice for this purpose. In the case of 1D CNN, the approach would be similar to 2D CNN but the input instances will be flattened to a vector of length $1 \times wf$ (a tensor of size $d \times 1 \times wf$, where f is the number of features).

2) Long-Short Term Memory Networks (LSTMs): Unlike the CNNs that don't take the temporal dynamics into account, LSTMs [13] [15] store the information from the past in the internal memory and are better suited for the detection of temporal dependencies. The inputs for LSTM will constitute the aggregated timesteps (e.g. n timepoints each of $f = 24$ features can be split on $d = \lfloor n/b \rfloor$ batches of size $b \times f$, where b is the batch size). Similarly to the CNN, the output of the last LSTM layer can

be processed by the dense layer with softmax activation to produce a vector of class probabilities for each probed region (or sigmoid for determining if there is a pocket or not). A trial-and-error process is adopted to determine the optimal number of layers in the deep learning network – starting with the simple case of one layer and gradually increasing the architecture complexity.

V. Data Preprocessing

1) Sliding window approach: The sliding window approach to data preprocessing for CNN is depicted in Fig. 1. We slide a window of size w over a timecourse of features and extract $w \times f$ matrices, where f is the number of features. To make the newly-created matrices processable with Conv2D layers in PyTorch an additional singleton dimension was added to convert them into 3D tensor of size $w \times f \times 1$. Each window is assigned a probing region label. For example, on Fig. 1 the windows at the beginning of the path extract $w \times f$ that correspond to the buccal and medial buccal regions of tooth 32, while the windows at the end of the path correspond to the medial-buccal region of tooth 28. At this stage of

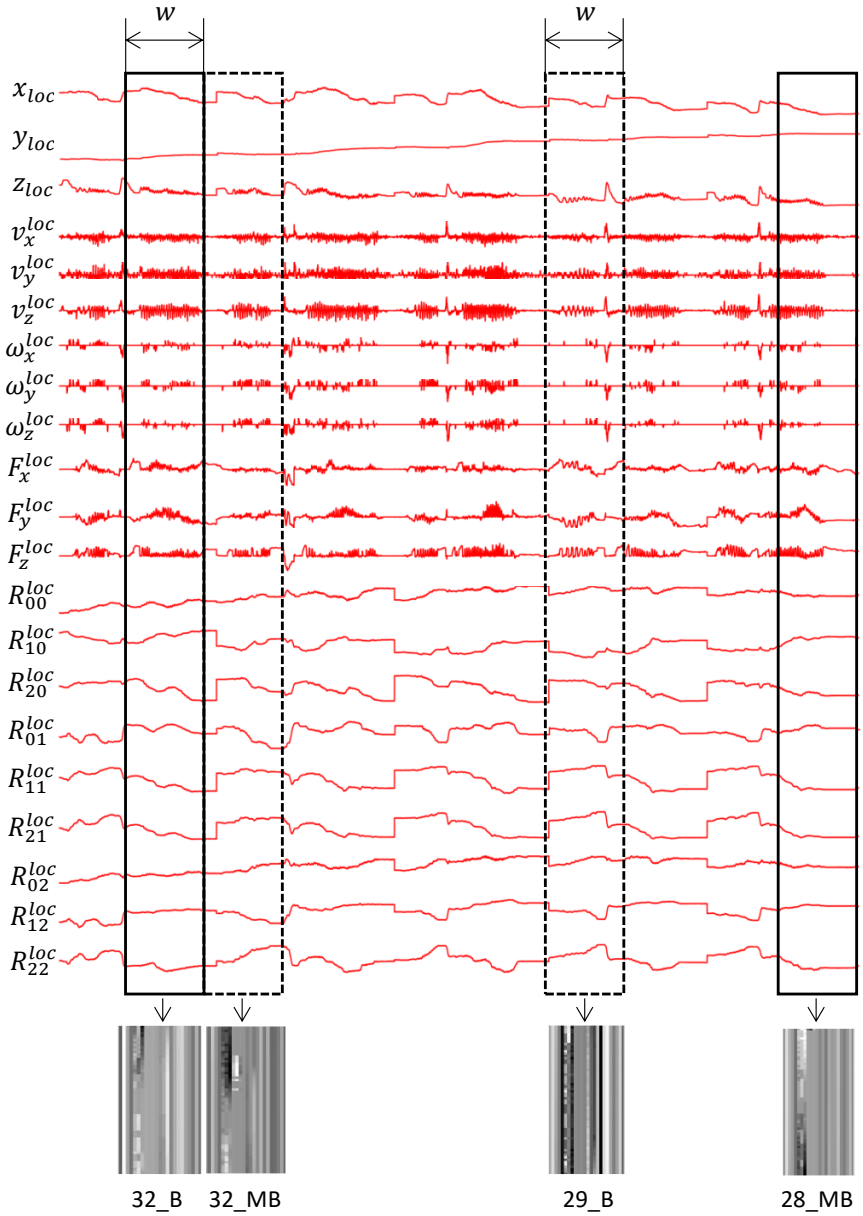


Fig. 1: Converting the time-series data into $w \times f \times 1$ tensors by sliding the window over the features. Here w is the size of the window and f is the number of selected features.

the project, we consider non-overlapping windows, however, in future work it would be also interesting to investigate how the size of the overlap between adjacent windows affects the performance of the model.

Conducting experiments with different windows sizes we noticed that the overall test accuracy grows for small window sizes and then decays slowly when the size of the window increases. The accuracy of the current model first increases from 0.6 to 0.85 with the increase of window sizes from $w = 2$ to $w = 6$ and platoes for window sizes from 6 to 14 with best testing accuracy achieved at $w = 8$. Starting from $w = 14$ the accuracy gradually decreases and drops below 0.5 for w values around 150. Therefore, we decided to use the window size of $w = 8$ in this study.

2) Univariate Feature Selection with Mutual Information: The univariate statistical tests allow to determine the level of correlation between the features and the labels. This study uses the Mutual Information (MI) metrics in feature significance analysis since MI can capture any kind of dependencies between variables and targets, including nonlinear relationships and it is invariant to the data transformations [9]. MI also estimates how much the knowledge about the value of one variable affects the uncertainty on the other [14], which gives an idea about the relevance of a feature subset to the targets [9]. The MI is defined as:

$$I(x,y) = \sum_{i=1}^{n}\sum_{j=1}^{n} p\left(x_i, y_j\right) \cdot \left(\frac{p\left(x_i, y_j\right)}{p\left(x_i\right) p\left(x_j\right)}\right), \tag{1}$$

where $p\left(x_i\right)$ is a mass probability of a discrete random variable x: $p\left(x_i\right) = \Pr\left\{x = x_i\right\}, x_i \in x$ [9].

The MI is 0 when the random variables are statistically independent. The mutual information between a set of m features and the class variable Y can be computed as:

$$I\left(\left\{x_1, \ldots, x_m\right\}, Y\right) = \sum_{k=1}^{m} \sum_{\substack{S \subseteq \{x_1,\ldots,x_m\}, \\ |S|=k}} I\left(\left[S \cup Y\right]\right), \tag{2}$$

where $I\left(\left[S \cup Y\right]\right) = I\left(s_1, s_2, \ldots s_k, Y\right)$.

The top ten significant features, detected by the Mutual information applied to all 12 recordings for lingual and buccal sites are summarized in Table I. For the purpose of this study the following 4 significant features have been selected: $x_{loc}, y_{loc}, R^{loc}_{1,1}, R^{loc}_{2,0}$.

TABLE I: Top 10 features according to the Mutual Information scores.

Feature	MI significance score	
	Lingual	Buccal
y_{loc}	2.529802	2.568962
$R^{loc}_{2,0}$	1.870411	1.728601
x_{loc}	1.860232	1.918211
$R^{loc}_{1,1}$	1.703054	1.647311
$R^{loc}_{2,1}$	1.683456	1.700431
$R^{loc}_{1,0}$	1.662256	1.580518
$R^{loc}_{2,2}$	1.598940	1.586183
$R^{loc}_{0,0}$	1.537663	1.578130
z_{loc}	1.494291	1.598704
$R^{loc}_{1,2}$	1.474145	1.536801

VI. Deep Learning Networks Architecture

1) CNN: We consider a simple CNN model that contains two 2-D convolutional layers each with 3×3 kernels, both with unit stride and padding of 1 added to both sides of the input (see Fig. 2). The input of the network is $w \times f \times 1$ "grayscale image" – a tensor obtained with sliding window and the output is the vector of activations which max value corresponds to the class of the probed region. The

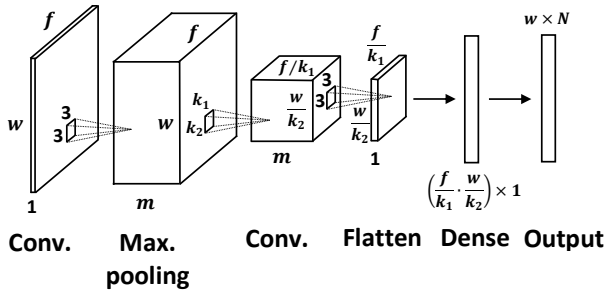


Fig. 2: The architecture of the proposed CNN model. w is the window size, f – number of selected features, m – number of channels after the first convolution layer, k_1 and k_2 – the kernel dimensions of the max pooling layer and N – number of classes ($N = 15$).

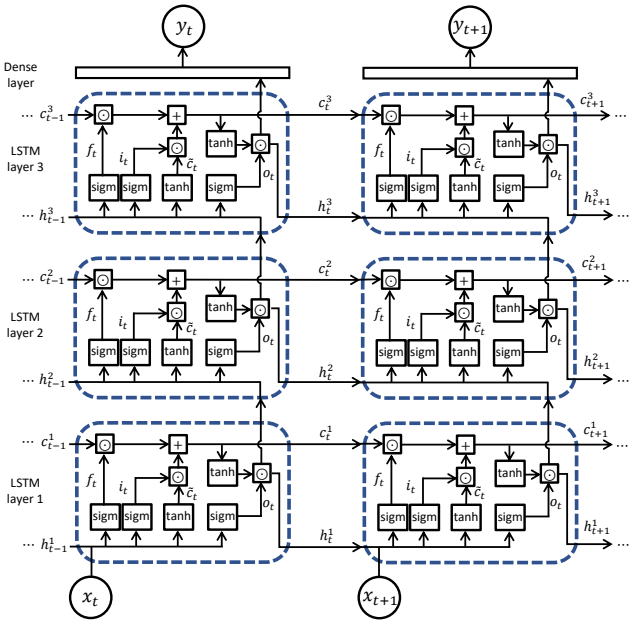


Fig. 3: The architecture of proposed LSTM model. It consists of 3 LSTM units and a Dense layer. f_t, i_t, o_t are the outputs from forget, input and output gates correspondingly. For more information about LSTM networks please refer to [13] [15].

output of the first convolutional layer is set to $m = 4$ channels, i.e. it extracts 4 feature maps with 4 kernels. The first convolutional layer is followed by a 2-D max pooling layer with $k_1 \times k_2$ kernel. The second convolutional layer outputs one channel only. After the second convolutional layer, there is a connected dense layer that classifies the extracted features into corresponding regions of probing. To accelerate the training process, the inputs to the max pooling and the dense layers are standardized with batch normalization and then activated with the Rectified Linear Unit (ReLu) function. We also use the Cross Entropy loss as a loss function and Adam optimizer for model training.

The proposed architecture is independent of window size w. However, the choice of the 2-D max pooling layer's kernel depends on the number of selected features f. After the max pooling the input to the next 2-D convolutional layer is a tensor of size $\frac{f}{k_1} \times \frac{w}{k_2} \times m$, where m is the number of output channels of the first 2-D convolutional layer. Since the Dense layer accepts inputs equal to the window size w it is important to make sure that the outputs of the second 2-D convolutional layer after flattening are scaled well with the inputs to the Dense layer, i.e. $\frac{f}{k_1} \cdot \frac{w}{k_2} = w$. Therefore, k_1 and k_2 are chosen the way so their product is equal to the number of features: $k_1 \cdot k_2 = f$. For example, in the case of 4 significant features, used in this study $k_1 = k_2 = 2$.

2) LSTM: The LSTM deep learning network consists of three stacked recurrent LSTM layers and a final Dense layer with softmax activation to produce a vector of class probabilities for each probed region. Each LSTM layer has 256 dimensions. We split the n samples into $d = \lfloor n/b \rfloor$ batches of size $b \times f$, where b is the batch size (we use $b = 128$ for LSTM). Also, we did not perform the feature selection and were using all 24 features. For each fold, we split the dataset into training, validation, and testing subsets, and wrap them with data loaders. The cyclic learning rate scheduler approach [16] with cosine function was adopted to vary the learning rate between $\eta = 10^{-3}$ and $\eta = 10^{-5}$. It allows to increase the model's classification accuracy and convergence speed. For both architectures, we use PyTorch implementation of Cross Entropy as a loss function, which allows to skip the softmax activation after the last dense layer.

VII. Results

1) Probing Region Prediction: To evaluate the performance of the proposed CNN and LSTM models we use haptic data recorded by the dental professional with the help of the Haptodont dental simulator. We performed 12-fold cross validation on the collected 12 independent recordings, by training the model on the 11 recordings and testing it on the 12th at each iteration. The 90% of 11 recordings were used to train the model and the remaining 10% to validate it during training. We use 100 epochs to train the CNN model and 20 epochs to train LSTM (since for LSTM it takes about 10 epochs to achieve training accuracy of $> 95\%$ with validation accuracy fluctuating around 90%, while for CNN it takes at least 70 epochs to achieve the same performance). It could be explained by the effect of cyclic scheduler implemented in LSTM, which leads to faster convergence of the gradients, but it requires further investigation. In terms of training time, the LSTM model takes longer than of CNN (almost 100 sec. per each epoch vs. < 1 sec.). This difference in speeds can be attributed to the architecture of LSTM that processes information sequentially, i.e. the subsequent steps depend on previous ones. In contrast, computations in the CNN can be handled in parallel, e.g. at the same time the same filter can be applied to different parts of the image.

The accuracy, precision, recall, and F1-scores after 12-fold cross validation of the model on the dataset with the selected top four significant features are plotted on Fig. 4. For the CNN model, the average accuracy for the lower-right buccal side is 0.84, with precision and recall of 0.85 and 0.83 correspondingly, while for the lower-left lingual the average accuracy is 0.85 with the precision of 0.86 and recall of 0.84. The LSTM model achieves an average accuracy, precision, and recall of 0.89 for the lower-right buccal side and accuracy of 0.85, the precision of 0.86, and recall of 0.85 for the lower-left lingual. Interestingly that based on the performance metrics the CNN model predicts the lower-left lingual side slightly better than the LSTM model. The opposite is true for the lower-right buccal side. However, the difference is not very significant ($\sim 5\%$). In overall, the performance of the LSTM model is slightly better than that of the CNN. The confusion matrices averaged after 12 folds for CNN and LSTM models are presented in Fig. 5 and Fig. 6. Notice that the accuracy of the CNN model is almost the same as of the SVM model used in [8], however, the CNN model leads to less variance in predictions and allows more accurately infer the lingual site probing regions (compare Fig. 4 and Fig. 2 in [8]). Similarly, the accuracy of the LSTM model is higher than the accuracy of SVM model in [8].

2) Periodontal Pocket Detection: Pocket detection has also been attempted. There are four pockets on a low jaw 3D model with rendered teeth and gingiva: two on the lower right buccal and two on lower

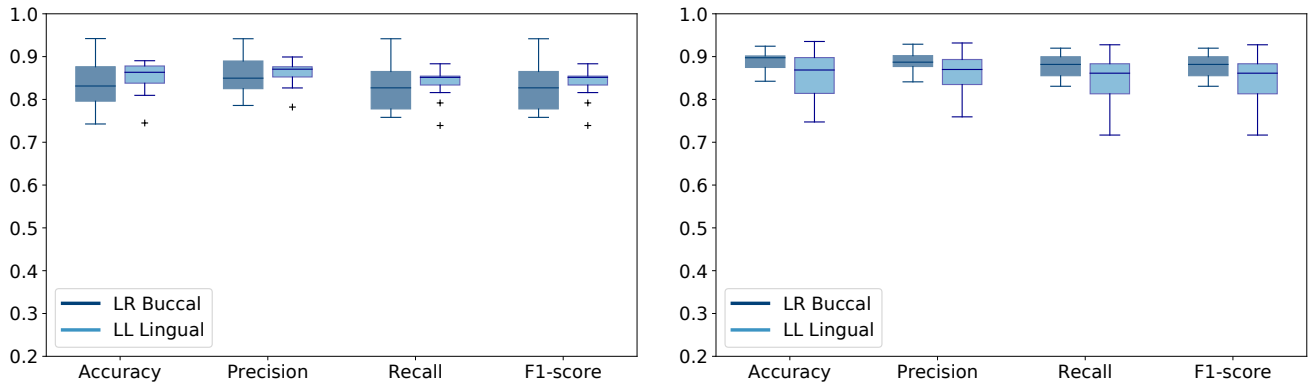
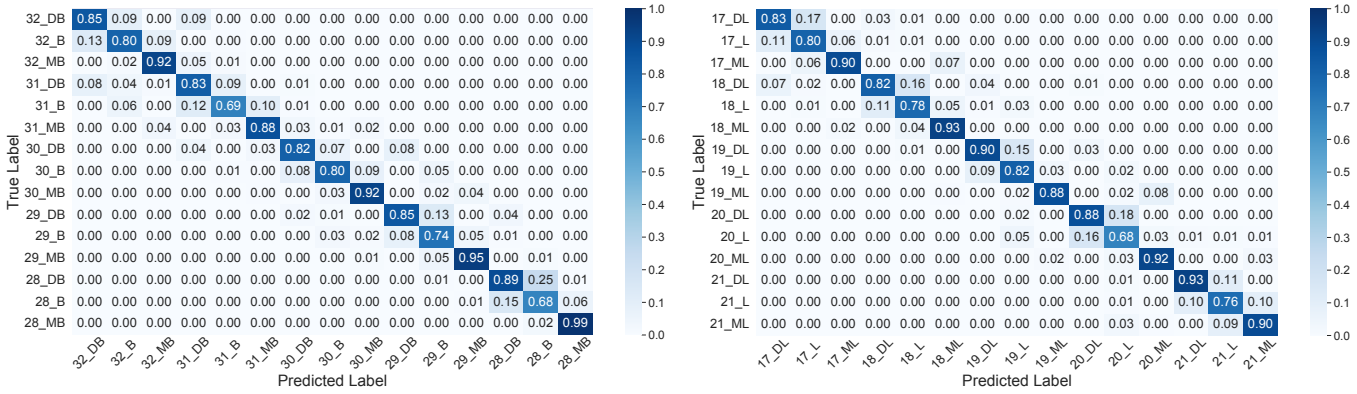
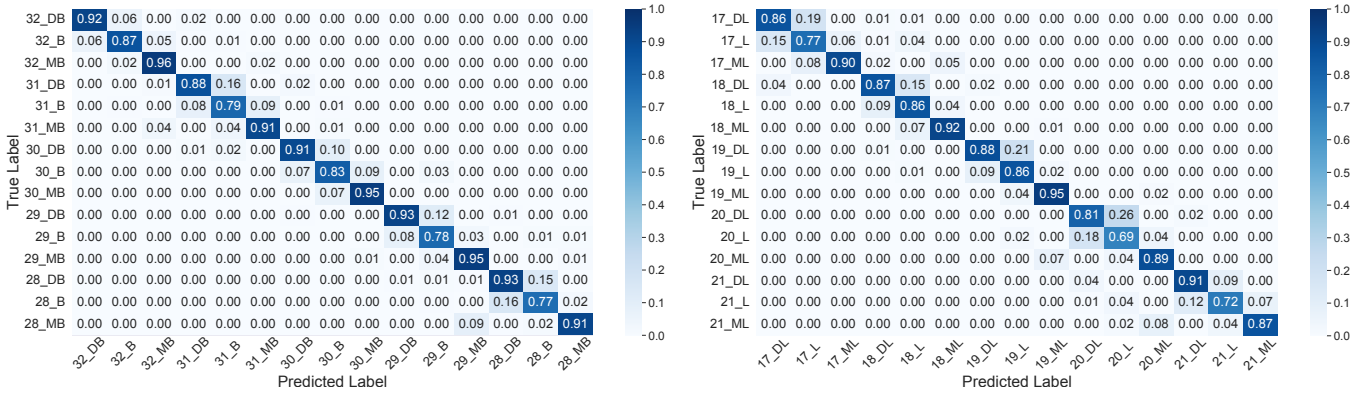


Fig. 4: Average accuracy, precision, recall and F1-score for a.) the CNN model with 4 significant features and $w = 8$, b.) LSTM model.



a.) b.)

Fig. 5: Normalized confusion matrices for the CNN model with 4 significant features and window of length 8 after 12-fold cross-validation. a.) lower right buccal, b.) lower left lingual.



a.) b.)

Fig. 6: Normalized confusion matrices for the LSTM model with 24 features after 12-fold cross-validation. a.) lower right buccal, b.) lower left lingual.

left lingual sides (Fig. 7). The data is highly imbalanced – out of the 30 regions, there are only four regions containing pockets. The problem converges to a binary classification task – to detect from the haptic data if there is a pocket or not on the periodontal probing path. The feature selection with mutual information determined that x, y, z coordinates and $R_{2,2}, R_{3,1}, R_{3,3}$ components are the top significant features. We implement the same CNN and LSTM architectures used for the prediction of probing region for the periodontal pockets detection. A similar split on training, validation and testing sets was adopted. Due to high class imbalance it makes no sense to consider accuracy as a performance measure, instead, we focus on precision and recall values averaged over 12 folds. For the CNN model, the evaluation with 12-fold cross-validation revealed that the average recall and precision values for class 1 for the lower-left lingual are 0.42 and 0.5, correspondingly. For lower-right buccal CNN achieves average precision of 0.42 and an average recall of 0.5. Using the whole set of 24 features does not improve the prediction of the CNN model. The low recall value can be attributed to the heavy imbalance in the dataset, and this issue can be addressed by providing enough samples that represent the pocket in the future.

At the same time, the application of the LSTM model for pocket detection gives quite encouraging results. We reused the same model architecture for probing region detection with all 24 features. The confusion matrices are presented on Fig. 8. The averaged over 12 folds precision and recall values were 0.93 and 0.92 for the lower-left lingual, and 0.92 and 0.97 for the lower-right buccal correspondingly. These results indicate that in some sense, the LSTM networks are capable of handling the high class imbalance in the periodontal probing data. The superior performance of the LSTM network on pocket

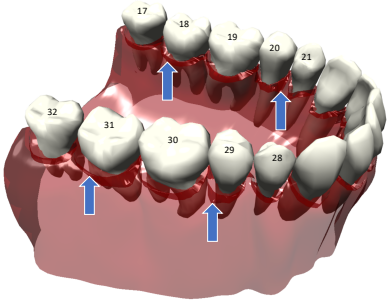
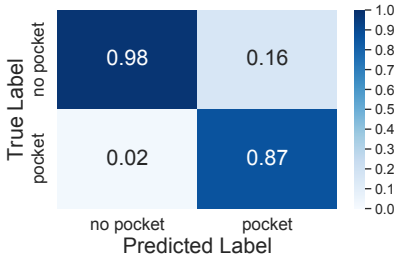


Fig. 7: Periodontal pockets locations (see also Table 1 in [8]).

a.)



b.)

Fig. 8: Normalized confusion matrices for LSTM model for pocket detection after validating over 12-folds: a.) lower right buccal, b.) lower left lingual.

prediction might be explained by their ability to learn the sequential information from the trajectory path. Indeed, the depth of the gingival sulcus around the pocket increases gradually, which affects the probing patterns such as tooltip coordinates of the probe and the exerted forces. In contrast to SVM and CNN, the LSTM network is capable to store and analyze past information, and therefore to detect gradual changes in the haptics data. For future work, it might be interesting to explore this ability of LSTM models and determine the parameters leading to the most accurate and fast pocket detection.

VIII. Conclusion and Future Work

The main goal of this project is to study the applications of Deep Learning networks to sensorimotor skill modeling. We evaluated the performance of two major classes of Deep Learning networks, namely, CNN and LSTM for the modeling of the periodontal probing task. The intention was to develop a model that is capable to determine which region of the tooth is probed and to detect the periodontal pockets from the professional's skill path, which includes positional coordinates, components of rotation matrices, the linear/angular velocities of the device tooltip, and the exerted forces.

The results demonstrate that the Deep Learning approaches can be used to model the skill of periodontal probing, in particular, detecting which region of the tooth is being probed. The LSTM model performs slightly better than CNN for the prediction of the probed region and far outperforms CNN when it comes to periodontal pocket detection. The high performance of the LSTM model for the pocket detection can be attributed to the ability of recurrent networks to "remember" the previous information and therefore, to infer the gradual increase of the depth of gingival sulcus in the regions close to the pocket, that might signal the network about the presence of the pocket. The pocket depths could be among the factors that might affect the performance of the LSTM network – the deeper the pocket the more gradual the "slopes" on both sides and the higher the detection accuracy.

Application of Deep Learning Networks for determining the region of probing and pocket detection opens an opportunity to integrate multimodal data, such as haptic, auditory, visual, and EMG recordings for creating a complete model of the sensorimotor skill for periodontal probing. The complete model must be capable of not only determining which region is probed but whether the student angulates the probe properly and measures the depth of the pocket correctly, i.e. to evaluate the student's performance, adjust to the student's level and provide proper training and guidance.

There are several directions that can be taken for future work. For example, it would be interesting to examine the feature maps extracted by the hidden layers and to determine which features are important for the periodontal probing skill. This knowledge might shed the light on the internal mechanisms of the sensorimotor skill and understand how the skill of a professional is different from the skill of a novice. Another direction is to explore the application of generative models, which trained on professional's recordings can produce an artificial skill path indistinguishable from the expert's. For instance, the Generative Adversarial Networks (GANs) potentially can be used for learning the data distribution

from the professional's recordings which will help to better understand the data and to generate the new samples. GANs can also be used for directly extracting a policy from the data and thus learning the skill for future reproduction [17].

From the practical point of view, it is important to investigate the ability of LSTM models to detect periodontal pockets. The main critique to the exceptional performance of the LSTM network for the pocket detection is that with the current data it is hard to say if the model learns the skill patterns associated with the pockets. Indeed, since the pocket locations are the same for all 12 recordings it would be difficult to ascertain whether the model indeed learns to detect the presence of the pocket from the features or just memorizes the pocket location from the path coordinates x, y, z. Therefore, in the future, it is important to ensure the variability of pocket depth and locations in the data by placing the pockets in different regions during the training stage and testing the network on a low jaw 3D model with a totally different periodontal pocket configuration.

The challenges faced while working with the CNN model are mostly related to the proper preprocessing of the data for CNN inputs. For PyTorch, it is essential to add a singleton dimension to the input tensor to mimic the RGB channels of the image. CNN also requires large volumes of data to be trained, for example, the current model used 11 recordings for training. The decrease in the number of training recordings drops the accuracy of the model. In the case of LSTM model, there is a trade-off between the training time, complexity, and accuracy of the network – the simple architectures require less time to train but slow to converge. More complex architectures require significant time for training. Despite the challenges the Deep Learning Networks provide a fast and more reliable way for detecting which region is currently probed from the haptics data recorded by an expert.

REFERENCES

[1] T. Cederborg, Ming Li, A. Baranes, and P. Oudeyer, "Incremental local online gaussian mixture regression for imitation learning of multiple tasks," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 267–274.

[2] T. Sato, Y. Genda, H. Kubotera, T. Mori, and T. Harada, "Robot imitation of human motion based on qualitative description from multiple measurement of human and environmental data," 11 2003, pp. 2377 – 2384 vol.3.

[3] Y. C. Zhao, A. Al-Yacoub, Y. M. Goh, L. Justham, N. Lohse, and M. R. Jackson, "Human skill capture: A hidden markov model of force and torque data in peg-in-a-hole assembly process," in *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2016, pp. 000 655–000 660.

[4] M. Kaiser and R. Dillmann, "Building elementary robot skills from human demonstration," in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 3, 1996, pp. 2700–2705 vol.3.

[5] D. D. Lee and H. S. Seung, "Learning in intelligent embedded systems," in *Workshop on Embedded Systems (Workshop on Embedded Systems)*. Cambridge, MA: USENIX Association, Mar. 1999. [Online]. Available: https://www.usenix.org/conference/workshop-embedded-systems/learning-intelligent-embedded-systems

[6] T. Poggio and F. Girosi, "Networks for approximation and learning," *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1481–1497, 1990.

[7] Z. Wang and A. M. Fey, "Deep learning with convolutional neural network for objective skill evaluation in robot-assisted surgery," *International Journal of Computer Assisted Radiology and Surgery*, vol. 13, pp. 1959–1970, 2018.

[8] V. Babushkin, M. H. Jamil, D. L. Sefo, P. M. Loomer, and M. Eid, "Modeling a sensorimotor skill of periodontal probing," 2021, (in preparation). [Online]. Available: https://drive.google.com/file/d/1j2pTw8RzR-pCmduCp80PIMtpJO6lkWCl/view?usp=sharing

[9] J. Vergara and P. Estevez, "A review of feature selection methods based on mutual information," *Neural Computing and Applications*, vol. 24, 01 2014.

[10] H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, "Deep learning for time series classification: a review," *Data Mining and Knowledge Discovery*, vol. 33, 07 2019.

[11] Y. LeCun and Y. Bengio, "The handbook of brain theory and neural networks. chapter convolutional networks for images, speech, and time series," *MIT Press, Cambridge, MA, USA*, vol. 218, pp. 255–258, 1998.

[12] F. Li, K. Shirahama, M. A. Nisar, L. Köping, and M. Grzegorzek, "Comparison of feature learning methods for human activity recognition using wearable sensors," *Sensors (Basel, Switzerland)*, vol. 18, 2018.

[13] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, pp. 1735–80, 12 1997.

[14] M. Beraha, A. M. Metelli, M. Papini, A. Tirinzoni, and M. Restelli, "Feature selection via mutual information: New theoretical insights," in *2019 International Joint Conference on Neural Networks (IJCNN)*, 2019, pp. 1–9.

[15] J. Castro, P. Achanccaray Diaz, I. Sanches, L. Cue La Rosa, P. Nigri Happ, and R. Feitosa, "Evaluation of recurrent neural networks for crop recognition from multitemporal remote sensing images," 11 2017.

[16] L. N. Smith, "Cyclical learning rates for training neural networks," in *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2017, pp. 464–472.

[17] J. Ho and S. Ermon, "Generative adversarial imitation learning," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, ser. NIPS'16. Red Hook, NY, USA: Curran Associates Inc., 2016, p. 4572–4580.