Towards data-driven stroke rehabilitation via wearable sensors and deep learning

Aakash Kaku[*] ARK576@NYU.EDU
Center for Data Science
New York University

Avinash Parnandi[*] AVINASH.PARNANDI@NYULANGONE.ORG
Department of Neurology
New York University School of Medicine

Anita Venkatesan ANITA.VENKATESAN@NYULANGONE.ORG
Department of Neurology
New York University School of Medicine

Natasha Pandit NGP238@NYU.EDU
Department of Neurology
New York University School of Medicine

Heidi Schambra[†] HEIDI.SCHAMBRA@NYULANGONE.ORG
Department of Neurology
New York University School of Medicine

Carlos Fernandez-Granda[†] CFGRANDA@CIMS.NYU.EDU
Center for Data Science
Courant Institute of Mathematical Sciences
New York University

Abstract

Recovery after stroke is often incomplete, but rehabilitation training may potentiate recovery by engaging endogenous neuroplasticity. In preclinical models of stroke, high doses of rehabilitation training are required to restore functional movement to the affected limbs of animals. In humans, however, the necessary dose of training to potentiate recovery is not known. This ignorance stems from the lack of objective, pragmatic approaches for measuring training doses in rehabilitation activities. Here, to develop a measurement approach, we took the critical first step of automatically identifying functional primitives, the basic building block of activities. Forty-eight individuals with chronic stroke performed a variety of rehabilitation activities while wearing inertial measurement units (IMUs) to capture upper body motion. Primitives were identified by human labelers, who labeled

[*] Equal contribution
[†] Joint corresponding/last authors.

and segmented the associated IMU data. We performed automatic classification of these primitives using machine learning. We designed a convolutional neural network model that outperformed existing methods. The model includes an initial module to compute separate embeddings of different physical quantities in the sensor data. In addition, it replaces batch normalization (which performs normalization based on statistics computed from the training data) with instance normalization (which uses statistics computed from the test data). This increases robustness to possible distributional shifts when applying the method to new patients. With this approach, we attained an average classification accuracy of 70%. Thus, using a combination of IMU-based motion capture and deep learning, we were able to identify primitives automatically. This approach builds towards objectively-measured rehabilitation training, enabling the identification and counting of functional primitives that accrues to a training dose.

1. Introduction

Stroke is the leading cause of disability in the United States, affecting nearly 1 million individuals annually and costing the US an estimated $240 billion (Go et al., 2014; Ovbiagele et al., 2013). Almost two-thirds of stroke patients have significant motor impairment in their upper extremities (UE), which limits their performance of activities of daily living (ADLs) like feeding, bathing, grooming, and dressing. Rehabilitation training, incorporating the repeated practice of ADLs, is the primary clinical intervention to reduce UE impairment. However, rehabilitation is increasingly believed to have a marginal impact on recovery because of its low numbers of functional repetitions, or training dose (Krakauer et al., 2012). In animals models, UE recovery is substantially improved by high-dose functional training delivered early after stroke (Murata et al., 2008; Jeffers et al., 2018). In humans, the optimal training dose to improve recovery is unknown, because no quantitative dose-response studies have been undertaken in the early weeks after stroke. The resulting vacuum of clinical guidelines has perpetuated the delivery of low and variable training doses (Lang et al., 2009).

A major reason for this failure is the absence of precise and pragmatic tools to measure training dose. Most rehabilitation studies use time-in-therapy to approximate dose (Lohse et al., 2018). Although one may intuit that more scheduled time equals more training repetitions, a linear relationship does not hold. In a seminal study observing standard rehabilitation practice, investigators found that the number of trained movements varied widely across clinicians and sessions (Lang et al., 2009), underscoring the imprecision of using time-in-therapy as a proxy for dose. Another approach for measuring dose is manual tallying, where a human observer identifies and counts motions of interest. Because functional motions are fluid and fast, they are difficult to disambiguate in real time. Video recordings aid scrutiny, but analysis is prohibitively time-intensive: in our experience, one minute of videotaped motion requires one hour of analysis by trained coders. This laboriousness makes manual tallying impractical for clinical or research deployment.

A third approach for measuring training dose is pairing motion capture technology with machine learning. Wearable devices such as inertial measurement units (IMUs) generate kinematic data about UE motions. Investigators decide on motions of interest (classes) that they wish to detect. Using a supervised approach, machine learning models can be

trained to recognize classes of motions from their kinematic signatures (Parnandi et al., 2019). Once these motions are detected, they can be tallied to a dose.

Recent studies using this approach have sought to classify functional motion (e.g. tying shoelaces) and nonfunctional motion (e.g. arm swinging during walking) (McLeod et al., 2016; Bochniewicz et al., 2017; Leuenberger et al., 2017). In one, chronic stroke patients performed loosely-structured activities while wearing an IMU on their paretic wrist (Bochniewicz et al., 2017). From the IMU recordings, a random-forest model distinguished functional from nonfunctional motion with 70% accuracy. The resulting unit of measure was time spent in functional motion. While the classification performance of this approach is good, the resulting metric is nearly as problematic as measuring time-in-therapy: for example, did more time in functional motion correspond to the performance of more motions, or did it simply take longer to perform the same motions? What kinds of functional motions were made? Without knowing motion content, it is challenging to identify the relationship between repetitions and recovery, or to replicate a successful rehabilitation intervention.

In this work, we sought to address these limitations by taking the first step towards measuring rehabilitation dose. To unpack the motion content of rehabilitation, we focus on functional primitives, single motions or minimal-motions that serve a single purpose (Schambra et al., 2019). There are five classes of functional primitives: reach (motion to contact an object), transport (motion to convey an object), reposition (motion into proximity of an object), stabilize (minimal-motion to keep an object still), and idle (minimal-motion to stand at the ready). Rehabilitation activities can be successfully broken down into these constituent primitives, indicating that primitives are a useful unit of measure (Schambra et al., 2019). As a unit of measure, primitives thus provide motion content information that would inform a dose-response inquiry and the replication of an intervention. We further focus on primitives for three reasons. First, because primitives are a single motion event with a surprisingly consistent phenotype, even in stroke patients (Schambra et al., 2019), automated identification is facilitated. Second, because some stroke patients are unable to fully complete activities, primitives can provide a more nuanced picture of performance. Third, because primitives may be neurally hard-wired (Graziano, 2016; Ramanathan et al., 2006), measuring their execution may enable us to more precisely track central nervous system reorganization after stroke.

To develop an approach that identifies and counts functional primitives in a practical, automated manner, we paired sensor-based motion capture with supervised machine learning. We used an array of inertial measurement units (IMUs) on the upper body to generate richly characterized motion data. We had stroke patients perform a battery of rehabilitation activities, which generated a large sample of primitives with varying characteristics (e.g. speed, duration, extent, location in space). Once the motion data was labeled, we trained various machine learning models to classify primitives. We report our steps for identifying the best-performing algorithm and for optimizing its classification performance. Our approach is illustrated in Figure 1.

Generalizable Insights about Machine Learning in the Context of Healthcare In this work, we performed a systematic comparison of machine learning methods for the task of functional-primitive identification, and propose a model that outperforms existing methodology. Our results suggest several insights that have the potential to generalize

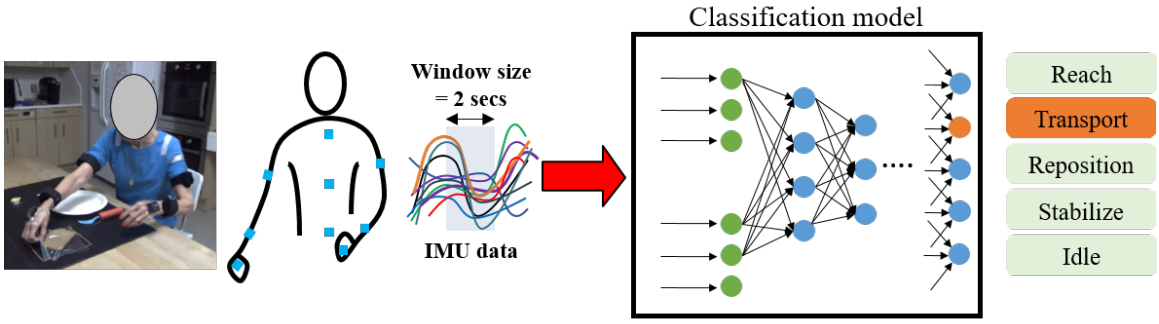

Figure 1: Diagram of the proposed approach for identification of functional primitives. Stroke patients perform a battery of rehabilitation activities while wearing IMU sensors. Machine-learning models are trained to classify the functional primitives from the sensor data.

to other healthcare applications (especially those involving wearable sensors). First, deep learning methods that directly process multivariate time series of sensor data seem to be significantly more effective than techniques based on handcrafted statistical features. Second, in order to combine data that represents different physical quantities, it may be helpful to map them to a common representation space incorporating an initial module that produces a separate embedding for each quantity. Third, adaptive feature-normalization techniques, such as instance normalization, may increase the robustness of convolutional neural networks to shifts in the distribution of the data, which can occur when the models are applied to new patients. Adaptive normalization uses statistics computed on the test data, in contrast to batch normalization, which uses statistics computed on the training data.

2. Related Work

To the best of our knowledge, only one previous study has used machine learning to identify functional primitives from IMU sensor data (Guerra et al., 2017). The authors used hidden Markov models to learn a latent representation of the sensor data, which was then used to perform classification via logistic regression. They acquired their data based on a few highly structured tasks, primarily consisting of moving objects to/from horizontal and vertical targets. Although this approach is useful to develop proof-of-concept methods, it does not reflect many of the challenges of real-world scenarios where unstructured tasks generate more varied and complex motions. In the present work, we gather data from real-world rehabilitation activities. Modeling functional primitives in this setting requires more complex models such as deep neural networks. In addition, the previous work was based on a small number of mostly mildly impaired patients; the present work increases the sample size 8-fold and captures a wider range of impairment.

Activity recognition using data gathered with wearable sensors is an active area of research in machine learning. However, it is important to emphasize that recognizing activities does not address the problem of measuring rehabilitation dose. Activities are prolonged sequences of motions that achieve several goals (Schambra et al., 2019). Problematically, activities are not standardized: their motion content varies by individual, culture, and en-

vironment (Fisher et al., 1992; Teresi et al., 1989). For example, the motions undertaken to perform a cooking activity differ if the meal is breakfast or dinner, or Japanese or German. This variable motion content not only challenges the automated recognition of activities, it also limits the identification of a dose-response relationship and the reproducibility of interventions.

Although activity recognition does not serve dose quantitation, prior studies in this area offer computational directions for classifying patterns of motion. Initially, methodology was mostly based on statistical features processed with techniques such as random forests or fully-connected neural networks (e.g. Elvira et al. (2014); Kwapisz et al. (2011a)). More recently, deep learning methods have been applied to perform activity recognition without precomputing statistical features. Specifically, Wang et al. (2017) showed that a ResNet-style convolutional architecture outperformed traditional non-deep learning methods as well as fully convolutional networks on several activity-recognition datasets (Kwapisz et al., 2011b; Thammasat, 2013; Joshua and Varghese, 2014). Cui et al. (2016) demonstrated that a simple convolutional model performed well when trained on data sampled at multiple scales. Karungaru (2015), Oukrich et al. (2018) and Murad and Pyun (2017) successfully used recurrent networks like Long Short Term Memory (LSTM) and Bi-LSTM for activity recognition. However, Ha et al. (2015) found that convolutional neural networks may outperform recurrent networks for some tasks. Given these conflicting results, in this work we sought to determine the necessity of using statistical features and the performance of recurrent versus convolutional networks for classification of functional primitives.

3. Cohort

3.1. Cohort Selection

We collected motion data from 48 stroke patients in an inpatient rehabilitation setting. Individuals were included if they were ≥ 18 years old, had premorbid right-handed dominance, and had unilateral weakness from either ischemic or hemorrhagic stroke. Individuals were excluded if they had traumatic brain injury; any musculoskeletal or non-stroke neurological condition that interferes with the assessment of motor function; contracture at the shoulder, elbow, or wrist; moderate upper extremity dysmetria or truncal ataxia; visuospatial neglect; apraxia; global inattention; or legal blindness. Table 1 describes the demographic and clinical characteristics of the patients.

3.2. Data Acquisition and Labelling

The data were gathered while the patients performed activities of daily living that are commonly trained during stroke rehabilitation. The activities included: washing the face, applying deodorant, combing the hair, donning and doffing glasses, preparing and eating a slice of bread, pouring and drinking a cup of water, brushing teeth, and moving an object on horizontal and vertical target array. See Section A for a detailed description. The patients performed five repetitions of each activity.

Upper extremity motion was recorded using nine IMUs (Noraxon) attached to the upper body, specifically to the cervical vertebra C7, the thoracic vertebra T12, the pelvis, and both arms, forearms, and hands. Each IMU samples linear acceleration, angular velocity,

	Training set	Test set 1	Test set 2
n	33	8	7
Age (in years)	56.3 (21.3-84.3)	60.9 (42.6-84.3)	58.3 (41.1-74.4)
Gender	18 F : 15 M	4 F : 4 M	4 F: 3 M
Time since stroke (in years)	6.5 (0.3-38.4)	3.1 (0.4-5.7)	3.16 (1.1-6.4)
Paretic side (Left : Right)	18 L : 15 R	4 L : 4 R	3 L : 4 R
Stroke type (Ischemic : Hemorrhagic)	30 I : 3 H	8 I : 0 H	2 I : 5 H
Fugl-Meyer Assessment score	48.1 (26-65)	49.4 (27-63)	15.3 (8-23)

Table 1: Demographic and clinical characteristics of the patients in the cohort. Mean and ranges in parentheses are shown. The cohort is divided into a training set and a test set (Test set 1) of mildly and moderately-impaired patients, and a test set of severely-impaired patients (Test set 2). There is no overlap of patients between the training and test sets.

and magnetic heading at 100 Hz. These data are then converted to 9 sensor-centric unit quaternions, representing the rotation of each sensor on its own axes, using coordinate transformation matrices. In addition, proprietary software (Myomotion, Noraxon) generates 22 anatomical angle values using a rigid-body skeletal model scaled to the patient's height and UE segment lengths. See Section B for a detailed description of these angles. This results in a 76-dimensional vector containing the linear acceleration, quaternion, and joint-angle information. As additional features, we included the time elapsed from the start of the activity in seconds and the paretic side of the patient (left or right) encoded in a one-hot vector. This increases the dimension of the feature vector to 78. Each entry (except the one indicating the paretic side) was mean-centered and normalized separately for each task repetition in order to remove spurious offsets introduced during sensor calibration.

In order to label the data, motion was synchronously captured using two cameras (1088 x 704, 60 frames per second; Ninox, Noraxon) placed orthogonally < 2 m from the patient. Trained observers watched the videos to identify and label functional primitives in the video, which simultaneously labeled primitives in the IMU data.

3.3. Evaluation Protocol

An important consideration when evaluating methodology for classification of functional primitives is the level of impairment of the patients. Impairment level was assessed using the upper extremity Fugl-Meyer Assessment (FMA), where a higher score indicates less impairment (the maximum score is 66) (Fugl-Meyer et al., 1975). We separated the patients into three levels of impairment according to their FMA score: mild (FMA 53-65), moderate (FMA 26-52), and severe (FMA 0-25) (Woodbury et al., 2013). In order to evaluate our methodology, we assigned the patients to a training set containing 33 mildly and moderately

patients, a test set containing 8 mildly and moderately impaired patients (Test set 1)[1], and an additional test set containing 7 severely impaired patients (Test set 2). Table 1 describes the characteristics of these datasets. Our first goal was to test the methods on patients with a similar impairment level as those used for training. Our second goal was to evaluate the generalizability of the trained model to patients with worse impairment. To avoid any selection bias or data leakage, the training and test sets were constructed before training any models, and model selection was carried out via cross-validation based exclusively on the training set (see Section 5 for more details).

4. Methodology

Our goal in this work was to design a machine learning model for the identification of functional primitives from sensor data. We framed this as a classification problem, where the input to the model was a window of the multidimensional time series obtained from the IMU sensors (see Section 3.2 for a detailed description), and the output was an estimate of the primitive corresponding to the center of the window. Note, however, that a significant portion of the window could contain motion corresponding to other functional primitives. The duration of the window was set to 2 seconds in order to provide sufficient context to the model, i.e. from the time steps flanking the center of the window (shorter windows yielded inferior results in preliminary experiments). In this section, we describe two key modifications to standard convolutional neural network architectures: learned embeddings that map each sensor to a common representation, and adaptive normalization of the network features. These modifications yield a model that outperformed existing techniques for primitive identification, as demonstrated by the results reported in Section 6.

4.1. Learning Embeddings for Diverse Inputs

Each layer in a convolutional neural network (CNN) computes local linear combinations of outputs of the previous layer, weighted by the coefficients of several convolutional filters. As a result, when we apply a CNN to the multivariate time series representing the IMU data, the different entries in the time series are combined at the second layer. This may be problematic because each entry represents very different kinematic information, such as accelerations, quaternions, and joint angles. To address this issue, we mapped each entry separately to a common representation space. The mapping was implemented using multiple embedding modules consisting of several convolutional layers. Each embedding module processes one of the entries in the time series. The embeddings were then concatenated and fed to a CNN. The embedding modules were optimized jointly with the CNN. Figure 2 shows a diagram of our proposed approach. A related previous work by Yao et al. (2017) proposed computing embeddings in the frequency domain.

1. The 41 mildly and moderately patients were separated into eight subgroups, balancing for impairment level and their paretic side (left or right). One patient in each group was randomly assigned to the test set. The remaining patients were assigned to the training set.

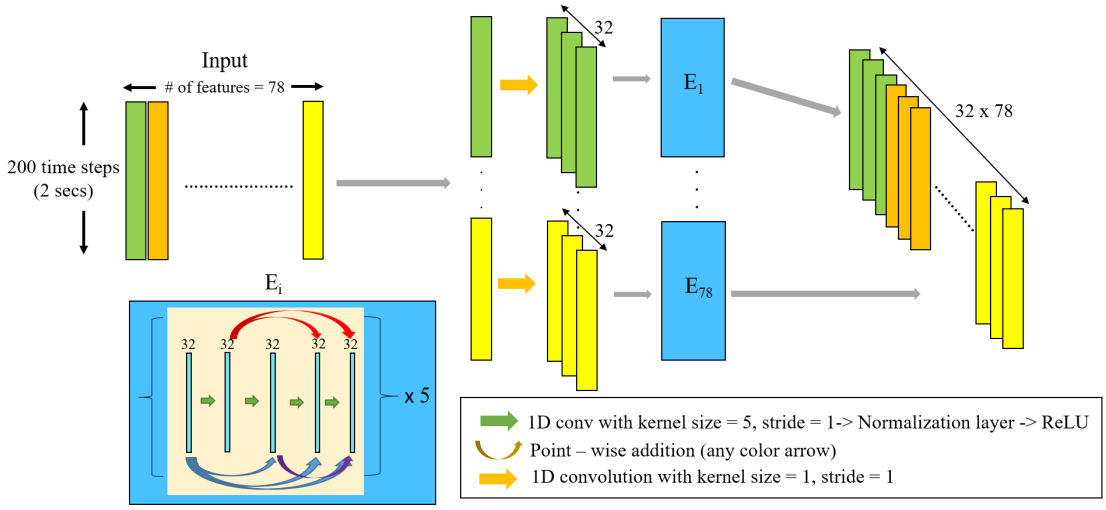

Figure 2: Diagram of the proposed approach to process multivariate time series where each entry may represent a different physical quantities. The ith entry is fed to an embedding module (denoted by E_i) consisting of several blocks of convolutional layers with DenseNet-like connections (Huang et al., 2017). The weights of each module are not shared, so that they can be calibrated to adapt to the corresponding physical quantity.

4.2. Robust Generalization via Adaptive Feature Normalization

In order to develop models that can be deployed in realistic rehabilitation settings, it is critical to ensure that they generalize accurately *to new patients* not present in the training set. This is challenging due to varying impairment levels and movement idiosyncrasies, which may produce systematic differences between the training data and the data from new patients. Achieving robustness to systematic shifts between the training and test data is a fundamental challenge in modern machine learning, particularly in healthcare applications. In the case of CNNs, recent work by Kaku et al. (2020) suggests that batch normalization may be particularly sensitive to such shifts.

Batch normalization has become a standard element in CNNs because it provides stability to different initializations and learning rates (Ioffe and Szegedy, 2015). It consists of two operations applied at the end of each layer. First, the features corresponding to each convolutional filter in the layer are centered and normalized using an approximation to their mean and standard deviation. Second, the resulting normalized features are scaled and shifted using two learned parameters per filter (a scaling factor and a shift). When the CNN is being trained, the estimates of the mean and standard deviation are obtained by averaging over each batch of examples. Simultaneously, estimates of the population mean and standard deviation of each filter are computed via running averages. The population statistics are used to perform normalization at test time. However, if the distributions of the training and test data differ, then these statistics may not center and normalize the data adequately, as demonstrated by Kaku et al. (2020).

Following Kaku et al. (2020), we applied CNN models to perform identification of functional primitives by replacing batch normalization with instance normalization, a normalization technique originally proposed to promote style invariance in image-style transfer by Ulyanov et al. (2016). In instance normalization, the features for each convolutional filter are centered and normalized using means and standard deviations that are computed over each individual example both at training and test time. This avoids the mismatch of training and test statistics that may occur with batch normalization.

5. Computational Experiments

The goal of our computational experiments was to compare the performance of different machine learning methods for identification of functional primitives, and to test our proposed approach. Inspired by the existing literature on movement identification from sensor data, we applied techniques based on statistical features (random forests and fully-connected neural networks), convolutional neural networks, and recurrent neural networks. As explained in Section 4, we framed primitive identification as a classification problem, where time-series windows were assigned to five different classes. We carried out model selection and hyperparameter optimization via cross-validation exclusively on the training set of 33 patients (see Section 3.3). To this end, we performed four different random splits. Each split contained 24 or 25 patients for training, and 9 or 8 patients for validation. Each patient appears in exactly one validation set. During validation, the models were compared using average classification accuracy[2] across the four splits. Section C reports the validation results. For each of the methods described below, we selected the hyperparameters achieving the highest cross-validation accuracy. Then, fixing those hyperparameters, we evaluated an ensemble of the models corresponding to the different splits on the two test sets. The ensemble was computed by averaging the estimated probabilities produced by each model (this resulted in a small improvement in accuracy with respect to the validation results for all methods).

In the remainder of this section we describe the hyperparameters of the different machine-learning methods in more detail. All neural-network models were trained using the Adam optimizer (Kingma and Ba, 2014) with starting learning rate of $1.25\ 10^{-4}$, which was divided by two every 20 epochs for the fully-connected and convolutional networks, and every 10 epochs for the recurrent networks. Training was terminated via early stopping based on the validation accuracy.

Random forest: The input to the random forest models was a set of five statistics computed over each dimension of the 78-dimensional windows: mean, maximum, minimum, standard deviation, and root mean square. These features capture useful information for movement identification, such as the energy of the motion and the variations within the window (Kwapisz et al., 2011a; Guerra et al., 2017). We used the scikit-learn random forest implementation (Pedregosa et al., 2011).

Hyperparameters: Minimum number of examples required to split each internal node, and minimum number of samples required to be at a leaf. The selected values were 2 and 1 respectively.

Fully-connected neural network: The input to the fully-connected neural network was the same set of five statistics as for the random forest.

2. To be clear, if c out of a n windows are classified correctly, the classification accuracy equals c/n.

Mildly / Moderately-impaired patients (Test set 1)						
Method	Random forest	FCNN	CNN	LSTM	Proposed	Ensemble
Balanced accuracy	52.98	58.04	64.01	66.58	69.21	**70.11**
Severely-impaired patients (Test set 2)						
Method	Random forest	FCNN	CNN	LSTM	Proposed	Ensemble
Balanced accuracy	32.95	36.60	38.22	41.76	43.50	**44.39**

Table 2: Balanced accuracy on Test set 1 and Test set 2 of the machine-learning models described in Section 5. FCNN denotes fully connected neural network. The ensemble is a combination of the proposed model and the LSTM, where the output probabilities were averaged.

Hyperparameters: Number of layers, number of neurons per layer, and dropout rate. The selected values were 8, 900, and 0.5 respectively.

Recurrent neural network: We used one of the most popular recurrent architectures, Long Short Term Memory (LSTM). Preliminary experiments with a Bi-LSTM architecture yielded inferior performance. The LSTM received the windows of the multivariate time series directly as an input.

Hyperparameters: Dimensionality of hidden units. The selected value was 4000.

Convolutional neural network (CNN): As in the case of the LSTM, we used CNNs to process the time-series window directly. We chose two architectures with skip connections similar to the ResNet (Wang et al., 2017) and the DenseNet (Huang et al., 2017). Preliminary experiments with an AlexNet-style architecture (Le Guennec et al., 2016) yielded worse performance. In order to evaluate the effect of input embeddings and adaptive feature normalization (see Section 4), we performed an ablation analysis where we trained the four possible combinations of these design choices for each model (with/without input embedding, with batch normalization/instance normalization). The depth of all networks was set to 44 layers. The architectures are described in detail in Section D.

6. Results

Table 2 shows the results of the different machine learning methods on the two test sets. The results correspond to the representative of each method that achieved the best cross-validation accuracy, as described in Section 5. To account for the different frequencies of each primitive in the data, we report balanced accuracy, defined as the average of the classification accuracies for each primitive[3]. The results without taking into account primitive frequency are very similar, see Table 9. Our main conclusion is that identification of functional primitives from IMU-sensor data via machine learning is possible: the deep learning methods achieved between 64% and 70% balanced accuracy on mildly-moderately impaired patients

3. The accuracy for each primitive is defined as c_i/n_i, $i = 1, 2, \ldots, 5$, where n_i is the number of windows associated with the ith primitive, and c_i denotes how many were classified correctly. The balanced accuracy is the average of the accuracies corresponding to the five primitives.

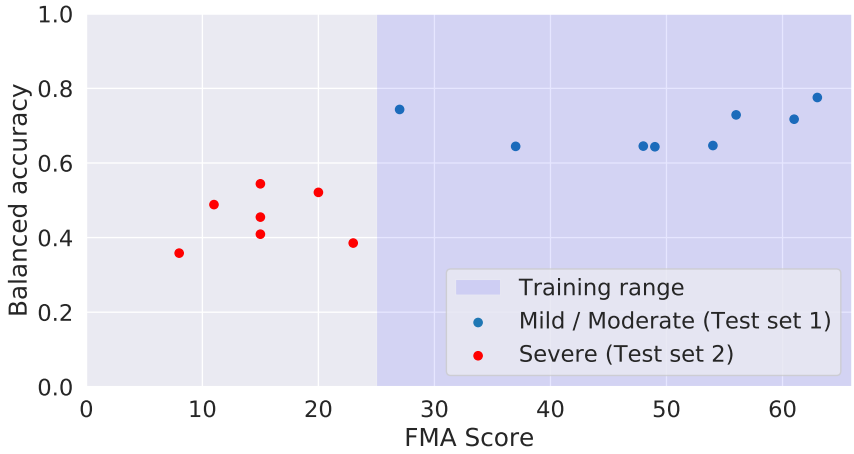

Figure 3: Balanced accuracy of the ensemble model applied to the patients in Test set 1 and Test set 2, plotted as a function of their impairment level (quantified by FMA score). The range of impairment levels of the patients in the training set is indicated by the dark-colored background. The balanced accuracy was above 60% for all mildly/moderately impaired patients, and above 50% for three out of the seven severely impaired patients (for comparison, random assignment yields 20% accuracy).

that do not appear in the training or validation data (Test set 1). On severely impaired patients (Test set 2) the average balanced accuracy was lower, between 38% and 44%. Decreased performance is expected because the training data only includes mildly and moderately impaired patients. Severely impaired patients not only had altered motion characteristics relative to less impaired patients, but their motions were also ascertained using modified activities adapted to their impairment level, as described in Section A. We used linear regression model to examine the effect of biological variables on model performance. We found no significant effects of gender, race, ethnicity, age, or impairment score on classification performance in the test set 1. Figure 3 shows the results of the ensemble model on the individual patients in the two test sets. The balanced accuracy was above 60% for all mildly-moderately impaired patients, and above 50% for three out of the seven severely impaired patients (for comparison, random assignment yields 20% accuracy).

Deep learning methods, which process the multivariate time series directly, systematically outperformed the techniques based on statistical features. Among the baseline deep learning methods, the recurrent network (LSTM) produced better results than the convolutional network. The best results overall were achieved by our proposed model, a convolutional network that incorporates input embeddings and instance normalization, as described in Section 4. An ensemble of this network and the LSTM, computed by averaging their outputs, produced a slight improvement. In Table 3 we show the results of an ablation analysis evaluating the individual contributions of input embeddings and instance normalization for two different convolutional architectures. For both architectures, the input embeddings and the adaptive normalization independently increased accuracy by 2-3%. When combined, the increase was 4-5%. The same trend was observed during validation on the different cross-validation folds, as shown in Table 15.

Architecture	ResNet		DenseNet	
Normalization	BN	IN	BN	IN
Input embedding	66.57	69.21	65.78	68.11
No input embedding	63.50	66.12	64.01	66.66

Table 3: Ablation analysis evaluating the individual contributions of input embeddings and instance normalization for two different convolutional architectures, described in more detail in Section D. The entries indicate the balanced accuracy of the different models on Test set 1 (mildly and moderately-impaired patients). BN denotes batch normalization, and IN denotes instance normalization. For both architectures the input embeddings and the adaptive normalization independently increased accuracy by 2-3%. When combined, the increase was 4-5%.

Figure 4 shows the confusion matrices of several of the methods on Test set 1. The different models had similar error patterns, indicating that some primitive pairs are inherently more difficult to distinguish. Some of these errors, e.g. between reach and transport or idle and stabilize, may result from the lack of grasp information in the data. The proposed model had the highest accuracy for all primitives except idle. The ensemble model, combining the proposed model with the LSTM, improved accuracy on the idle and transport primitives, but also decreased it slightly for reach and reposition. Figure 5 displays the probability estimates generated by the ensemble model applied to Test set 1 in the form of letter-value plots or Boxen plots (Hofmann et al., 2017). These plots show the quantiles of the probabilities, the middle line corresponds to the median. At least half of the probabilities assigned to the correct primitive (green Boxen plots) are above 0.6. In contrast, the vast majority of the probabilities corresponding to other primitives (red Boxen plots) are less than 0.6. This suggests that the probability estimate produced by the model is informative about its accuracy.

Recall that we frame primitive identification as a classification problem, where the input is a 2-second window of sensor data and the label is the primitive associated with the center of the window, which we dub the *ground-truth* primitive. A significant fraction of the window may contain different primitives, which is a potential source of errors. The machine learning models may be fooled by the other primitives and fail to detect the ground-truth primitive. Figure 6 shows the composition of windows in Test set 1, separated depending on whether they were classified correctly or incorrectly. Incorrectly classified windows tend to contain a smaller fraction of the ground-truth primitive, but the difference between the histograms (compare a and b) is not very pronounced. This suggests that the ensemble model was relatively robust to the presence of additional primitives. In fact, more than two thirds of the windows that were correctly classified contained additional primitives.

7. Discussion

This study demonstrates that deep learning can be used to identify functional primitives from IMU sensor data, which is an important step towards developing quantitative approaches for measuring stroke rehabilitation. It also suggests that input embeddings and

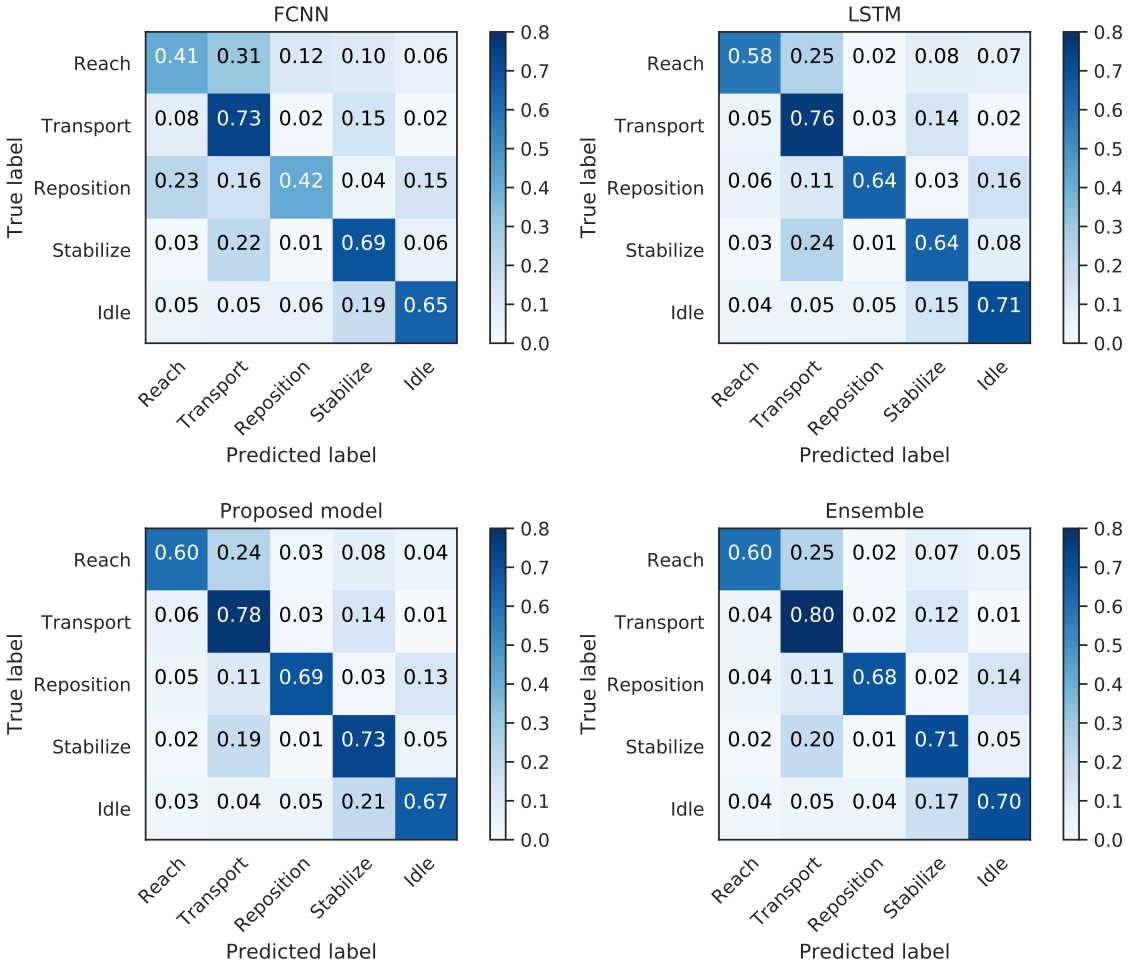

Figure 4: Confusion matrices of the fully connected neural network (FCNN), the LSTM, the proposed model, and an ensemble of the proposed model and the LSTM on Test set 1 (mildly and moderately impaired patients). Each entry indicates what fraction of windows labeled with *True label* were assigned to *Predicted label* by each model. The models had similar error patterns, indicating that some primitive pairs are inherently more difficult to distinguish (e.g. reach and transport, idle and stabilize, stabilize and transport).

adaptive feature normalization may contribute to address two challenges arising in many healthcare applications of machine learning: processing data containing different physical quantities, and ensuring robustness to distributional shifts during inference.

The classification performance of our approach exceeds random chance (20%) and is comparable to the accuracy (70%) of an approach that dichotomizes motion into time spent in functional versus nonfunctional motion (Bochniewicz et al., 2017). Importantly, our approach identifies the content of functional motion, i.e. functional primitives, that will serve as the basis for detailed rehabilitation measurement. Still, we anticipate that additional gains in classification performance can be made. We observed that the models had difficulty distinguishing reaches from transports, and idles from stabilizations. These prim-

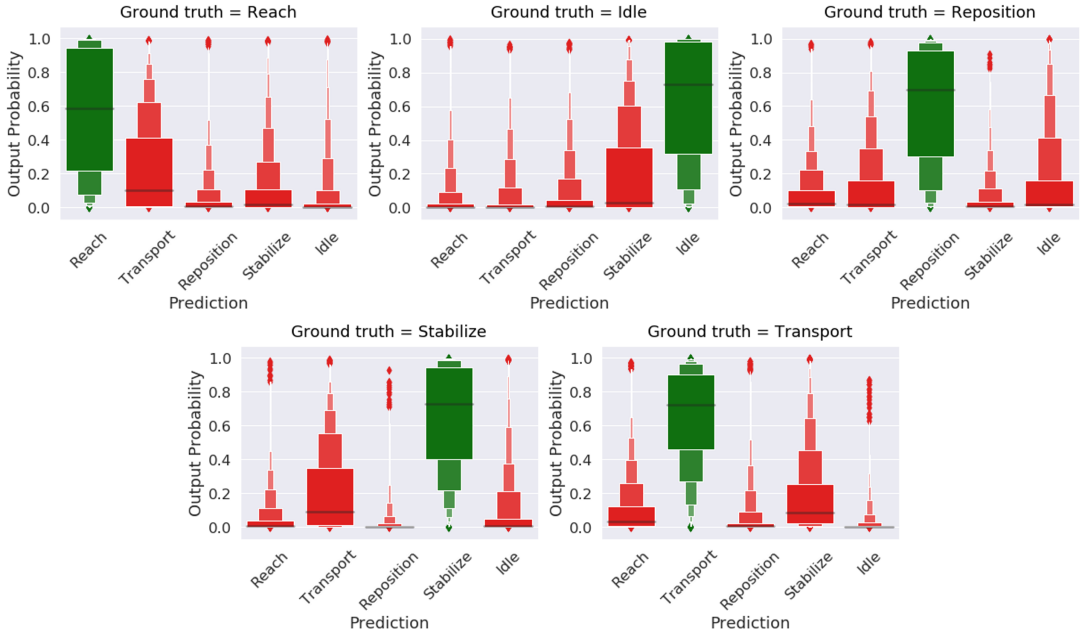

Figure 5: Letter-value plots or Boxen plots (Hofmann et al., 2017) of the probability estimates generated by the ensemble model for the different primitives. The Boxen plots corresponding to the ground-truth primitive are colored in green, the rest are colored in red. At least half of the probabilities assigned to the ground-truth primitive are above 0.6. In contrast, the vast majority of the probabilities corresponding to other primitives are less than 0.6. This suggests that the probability estimate produced by the model is informative about its accuracy.

itives differ by the presence and timing of grasp (Schambra et al., 2019), indicating that our current IMU array does not communicate this level of detail. However, affixing additional IMUs to paretic fingers would hinder hand function and further limit the practical utility of the approach. A recently developed computer vision model may offer a solution: it can extract finger position from video recordings (Cao et al., 2018). This new capability could enable us to use our existing video dataset to retrieve information about grasp. Future work will test whether combining kinematic information from IMUs and cameras can effectively boost classification accuracy.

Our study has some limitations to be considered. We studied only right-dominant patients balanced for right and left paresis. This step was necessary to simplify classification. Hand dominance may have a differential influence on the preferential roles of the UEs and their kinematic signatures (Przybyla et al., 2012). As the majority of humans are right-dominant, the proposed approach would be applicable to most patients. In the future, the inclusion of left-dominant patients for training and testing would enable us to build a more universal tool.

Another limitation of our approach is that classification performance deteriorates significantly for severely impaired patients, which means that it cannot be safely generalized to this cohort. This observation opens up two avenues for future work. First, gathering larger

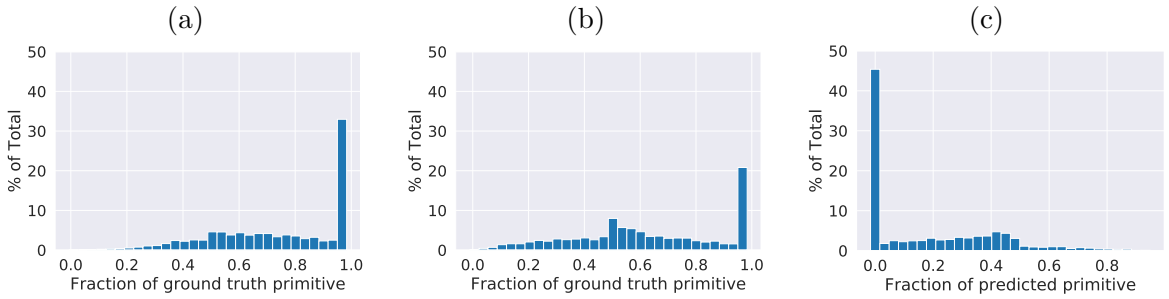

Figure 6: Histograms of the composition of the windows in Test set 1, examining the relationship between the classification results and the labels of the surrounding time steps. The ground-truth primitive of a window is defined as the label associated with the center of the window. The predicted primitive is the estimate produced by the ensemble model. (a) Percentage of time steps in each window associated with the ground-truth primitive for correctly-classified instances. (b) Percentage of time steps in the window associated with the ground-truth primitive for incorrectly-classified instances. (c) Percentage of time steps in each window associated with the predicted primitive for incorrectly-classified instances. Incorrectly-classified windows tend to contain a smaller fraction of the ground-truth primitive, but the difference between the histograms (compare a and b) is not very pronounced. This suggests that the ensemble model was relatively robust to the presence of additional primitives. In fact, more than two thirds of the windows that were correctly classified contained additional primitives. Among windows that were classified incorrectly, 45% of them did not contain the predicted primitive (c).

datasets, we will be able to include more severely impaired patients to train our models. Second, we will aim to develop machine-learning methodology capable of generalizing more robustly to different levels of impairment.

Finally, we performed primitive identification at a high time granularity (time steps of 10 ms). Future work will focus on converting these predictions to a sequence of estimated primitives. This may be expected to enable the next step in our approach, which is to automatically count primitives after they have been successfully recognized.

In summary, we present an approach that combines the kinematic data from IMUs with optimized deep learning models to identify functional primitives that constitute rehabilitation activities. We envision that once classification performance is maximized for an array of impairment, the trained model can be deployed to a clinical setting. There, patients instrumented with IMUs will undergo rehabilitation, generating unlabeled kinematic data. Using these data, the trained model will extract primitive content and count. This approach is expected to provide an objective means of quantitating the training dose of stroke rehabilitation. This measurement ability opens up a path for critical dose-response research and informed delivery of dosed rehabilitation, vital for improving recovery outcomes in stroke patients.

Acknowledgments

We would also like to thank the volunteers who contributed to label the dataset: Ronak Trivedi, Adisa Velovic, Sanya Rastogi, Candace Cameron, Sirajul Islam, Bria Bartsch,

Courtney Nilson, Vivian Zhang, Nicole Rezak, Christopher Yoon, Sindhu Avuthu, and Tiffany Rivera. We thank Dawn Nilsen, OT EdD for expert advice on the testing battery, and Audre Wirtanen for early assistance with the testing setup and data collection. This work was supported by an AHA postdoctoral fellowship 19AMTG35210398 (AP), NIH grants R01 LM013316 (AK, CFG, HMS) and K02 NS104207 (HMS), NSF NRT-HDR Award 1922658 (CFG) and by the Moore-Sloan Data Science Environment at NYU (AK).

References

Elaine M Bochniewicz, Geoff Emmer, Adam McLeod, Jessica Barth, Alexander W Dromerick, and Peter Lum. Measuring functional arm movement after stroke using a single wrist-worn sensor and machine learning. *Journal of Stroke and Cerebrovascular Diseases*, 26(12):2880–2887, 2017.

Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Openpose: realtime multi-person 2d pose estimation using part affinity fields. *arXiv preprint arXiv:1812.08008*, 2018.

Zhicheng Cui, Wenlin Chen, and Yixin Chen. Multi-scale convolutional neural networks for time series classification. *arXiv preprint arXiv:1603.06995*, 2016.

Victor Elvira, Alfredo Nazabal-Renteria, and Antonio Artes-Rodriguez. A novel feature extraction technique for human activity recognition. *2014 IEEE Workshop on Statistical Signal Processing (SSP)*, 2014. doi: 10.1109/ssp.2014.6884604.

Anne G Fisher, Yihfen Liu, Craig A Velozo, and Ay Woan Pan. Cross-cultural assessment of process skills. *American Journal of Occupational Therapy*, 46(10):876–885, 1992.

Axel R Fugl-Meyer, L Jääskö, Ingegerd Leyman, Sigyn Olsson, and Solveig Steglind. The post-stroke hemiplegic patient. 1. a method for evaluation of physical performance. *Scandinavian journal of rehabilitation medicine*, 7(1):13–31, 1975.

Alan S Go, Dariush Mozaffarian, Véronique L Roger, Emelia J Benjamin, Jarett D Berry, Michael J Blaha, Shifan Dai, Earl S Ford, Caroline S Fox, Sheila Franco, et al. Executive summary: heart disease and stroke statistics2014 update: a report from the american heart association. *Circulation*, 129(3):399–410, 2014.

Michael SA Graziano. Ethological action maps: a paradigm shift for the motor cortex. *Trends in cognitive sciences*, 20(2):121–132, 2016.

Jorge Guerra, Jasim Uddin, Dawn Nilsen, James Mclnerney, Ammarah Fadoo, Isirame B Omofuma, Shatif Hughes, Sunil Agrawal, Peter Allen, and Heidi M Schambra. Capture, learning, and classification of upper extremity movement primitives in healthy controls and stroke patients. In *2017 International Conference on Rehabilitation Robotics (ICORR)*, pages 547–554. IEEE, 2017.

Sojeong Ha, Jeong-Min Yun, and Seungjin Choi. Multi-modal convolutional neural networks for activity recognition. In *2015 IEEE International conference on systems, man, and cybernetics*, pages 3017–3022. IEEE, 2015.

Heike Hofmann, Hadley Wickham, and Karen Kafadar. Value plots: Boxplots for large data. *Journal of Computational and Graphical Statistics*, 26(3):469–477, 2017.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

Matthew Strider Jeffers, Sudhir Karthikeyan, Mariana Gomez-Smith, Sarah Gasinzigwa, Jannis Achenbach, Astrid Feiten, and Dale Corbett. Does stroke rehabilitation really matter? part b: an algorithm for prescribing an effective intensity of rehabilitation. *Neurorehabilitation and neural repair*, 32(1):73–83, 2018.

Liju Joshua and Koshy Varghese. Automated recognition of construction labour activity using accelerometers in field situations. *International Journal of Productivity and Performance Management*, 63(7):841862, Feb 2014. doi: 10.1108/ijppm-05-2013-0099.

Aakash Kaku, Sreyas Mohan, Avinash Parnandi, Heidi Schambra, and Carlos Fernandez-Granda. Be like water: Robustness to extraneous variables via adaptive feature normalization. *arXiv preprint arXiv:2002.04019*, 2020.

Stephen Karungaru. Human action recognition using wearable sensors and neural networks. *2015 10th Asian Control Conference (ASCC)*, 2015. doi: 10.1109/ascc.2015.7244580.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

John W Krakauer, S Thomas Carmichael, Dale Corbett, and George F Wittenberg. Getting neurorehabilitation right: what can be learned from animal models? *Neurorehabilitation and neural repair*, 26(8):923–931, 2012.

Jennifer R Kwapisz, Gary M Weiss, and Samuel A Moore. Activity recognition using cell phone accelerometers. *ACM SigKDD Explorations Newsletter*, 12(2):74–82, 2011a.

Jennifer R. Kwapisz, Gary M. Weiss, and Samuel A. Moore. Activity recognition using cell phone accelerometers. *ACM SIGKDD Explorations Newsletter*, 12(2):74, 2011b. doi: 10.1145/1964897.1964918.

Catherine E Lang, Jillian R MacDonald, Darcy S Reisman, Lara Boyd, Teresa Jacobson Kimberley, Sheila M Schindler-Ivens, T George Hornby, Sandy A Ross, and Patricia L Scheets. Observation of amounts of movement practice provided during stroke rehabilitation. *Archives of physical medicine and rehabilitation*, 90(10):1692–1698, 2009.

Arthur Le Guennec, Simon Malinowski, and Romain Tavenard. Data augmentation for time series classification using convolutional neural networks. In *ECML-PKDD Workshop on Advanced Analytics and Learning on Temporal Data*, 2016.

Kaspar Leuenberger, Roman Gonzenbach, Susanne Wachter, Andreas Luft, and Roger Gassert. A method to qualitatively assess arm use in stroke survivors in the home environment. *Medical & biological engineering & computing*, 55(1):141–150, 2017.

Keith R Lohse, Anupriya Pathania, Rebecca Wegman, Lara A Boyd, and Catherine E Lang. On the reporting of experimental and control therapies in stroke rehabilitation trials: a systematic review. *Archives of physical medicine and rehabilitation*, 99(7):1424–1432, 2018.

Adam McLeod, Elaine M Bochniewicz, Peter S Lum, Rahsaan J Holley, Geoff Emmer, and Alexander W Dromerick. Using wearable sensors and machine learning models to separate functional upper extremity use from walking-associated arm movements. *Archives of physical medicine and rehabilitation*, 97(2):224–231, 2016.

Abdulmajid Murad and Jae-Young Pyun. Deep recurrent neural networks for human activity recognition. *Sensors*, 17(11):2556, 2017.

Yumi Murata, Noriyuki Higo, Takao Oishi, Akiko Yamashita, Keiji Matsuda, Motoharu Hayashi, and Shigeru Yamane. Effects of motor training on the recovery of manual dexterity after primary motor cortex lesion in macaque monkeys. *Journal of neurophysiology*, 99(2):773–786, 2008.

Nadia Oukrich, El Bouazzaoui Cherraqi, and Abdelilah Maach. Human daily activity recognition using neural networks and ontology-based activity representation. *Innovations in Smart Cities and Applications Lecture Notes in Networks and Systems*, page 622633, 2018. doi: 10.1007/978-3-319-74500-8_57.

Bruce Ovbiagele, Larry B Goldstein, Randall T Higashida, Virginia J Howard, S Claiborne Johnston, Olga A Khavjou, Daniel T Lackland, Judith H Lichtman, Stephanie Mohl, Ralph L Sacco, et al. Forecasting the future of stroke in the united states: a policy statement from the american heart association and american stroke association. *Stroke*, 44(8):2361–2375, 2013.

Avinash Parnandi, Jasim Uddin, Dawn M Nilsen, and Heidi M Schambra. The pragmatic classification of upper extremity motion in neurological patients: a primer. *Frontiers in Neurology*, 10:996, 2019.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.

Andrzej Przybyla, David C Good, and Robert L Sainburg. Dynamic dominance varies with handedness: reduced interlimb asymmetries in left-handers. *Experimental brain research*, 216(3):419–431, 2012.

Dhakshin Ramanathan, James M Conner, and Mark H Tuszynski. A form of motor cortical plasticity that correlates with recovery of function after brain injury. *Proceedings of the National Academy of Sciences*, 103(30):11370–11375, 2006.

Heidi M Schambra, Avinash R Parnandi, Natasha G Pandit, Jasim Uddin, Audre Wirtanen, and Dawn M Nilsen. A taxonomy of functional upper extremity motion. *Frontiers in neurology*, 10:857, 2019.

Jeanne A Teresi, Peter S Cross, and Robert R Golden. Some applications of latent trait analysis to the measurement of adl. *Journal of gerontology*, 44(5):S196–S204, 1989.

Ekachai Thammasat. The statistical recognition of walking, jogging, and running using smartphone accelerometers. *The 6th 2013 Biomedical Engineering International Conference*, 2013. doi: 10.1109/bmeicon.2013.6687689.

Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016.

Zhiguang Wang, Weizhong Yan, and Tim Oates. Time series classification from scratch with deep neural networks: A strong baseline. In *2017 International joint conference on neural networks (IJCNN)*, pages 1578–1585. IEEE, 2017.

Michelle L Woodbury, Craig A Velozo, Lorie G Richards, and Pamela W Duncan. Rasch analysis staging methodology to classify upper extremity movement impairment after stroke. *Archives of physical medicine and rehabilitation*, 94(8):1527–1533, 2013.

Shuochao Yao, Shaohan Hu, Yiran Zhao, Aston Zhang, and Tarek Abdelzaher. Deepsense: A unified deep learning framework for time-series mobile sensing data processing. In *Proceedings of the 26th International Conference on World Wide Web*, pages 351–360, 2017.

Appendix A. Description of the Rehabilitation Activities

Tables 4 and 5 describe the activities performed by the mildly and moderately impaired stroke patients in the cohort. Tables 6 and 7 describe the activities performed by the severely impaired patients assigned to Test set 2.

Appendix B. Description of the Joint Angles

As described in Section 3.2, the sensor measurements are used to compute 22 anatomical angle values using a rigid-body skeletal model scaled to the patient's height and segment lengths. Table 8 describes these joint angles in detail.

Appendix C. Additional Results

Table 9 shows the performance of the different machine-learning models on Test sets 1 and 2 measured using accuracy instead of balanced accuracy. We also provide the cross-validation results on the individual splits, and the average validation accuracy, in Tables 11, 12 and 13 for the fully-connected neural network models, in Table 14 for the LSTM, and in Table 15 for the convolutional models.

Activity	Workspace	Target object(s)	Instructions
Washing face	Sink with a small tub in it and two folded washcloths on either side of the countertop, 30 cm from edge closest to patient	Washcloths, faucet handle	Fill tub with water, dip washcloth on the right side into water, wring it, wiping each side of their face with wet washcloth, place it back on countertop. Use washcloth on the left side to dry face, place it back on countertop
Applying deodorant	Tabletop with deodorant placed at midline, 25 cm from edge closest to patient	Deodorant	Remove cap, twist base a few times, apply deodorant, replace cap, untwist the base, put deodorant on table
Hair combing	Tabletop with comb placed at midline, 25 cm from edge closest to patient	Comb	Pick up comb and comb both sides of head
Don/doffing glasses	Tabletop with glasses placed at midline, 25 cm from edge closest to patient	Glasses	Wear glasses, return hands to table, remove glasses and place on table
Eating	Table top with a standard-size paper plate (at midline, 2 cm from edge), utensils (3 cm from edge, 5 cm from either side of plate), a baggie with a slice of bread (25 cm from edge, 23 cm left of midline), and a margarine packet (32 cm from edge, 17 cm right of midline)	Fork, knife, re-sealable sandwich baggie, slice of bread, single-serve margarine container	Remove bread from plastic bag and put it on plate, open margarine pack and spread it on bread, cut bread into four pieces, cut off and eat a small bite-sized piece

Table 4: Description of the activities performed by the mildly and moderately impaired patients in the cohort (1/2).

Activity	Workspace	Target object(s)	Instructions
Drinking	Tabletop with water bottle and paper cup 18 cm to the left and right of midline, 25 cm from edge closest to patient	Water bottle (12 oz), paper cup	Open water bottle, pour water into cup, take a sip of water, place cup on table, and replace cap on bottle
Tooth brushing	Sink with toothpaste and toothbrush on either side of the countertop, 30 cm from edge closest to patient	Travel-sized toothpaste, toothbrush with built-up foam grip, faucet handle	Wet toothbrush, apply toothpaste to toothbrush, replace cap on toothpaste tube, brush teeth, rinse toothbrush and mouth, place toothbrush back on countertop
Moving object on a horizontal surface	Horizontal circular array (48.5 cm diameter) of 8 targets (5 cm diameter)	Toilet paper roll	Move the roll between the center and each outer target, resting between each motion and at the end
Moving object on/off a Shelf	Shelf with two levels (33 cm and 53 cm) with 3 targets on both levels (22.5 cm, 45 cm, and 67.5 cm away from the left-most edge)	Toilet paper roll	Move the roll between the center target and each target on the shelf, resting between each motion and at the end

Table 5: Description of the activities performed by the mildly and moderately impaired patients in the cohort (2/2).

Activity	Workspace	Target object(s)	Instructions *	
			Proximal >Distal	Proximal <Distal
Washing face	Sink with a small tub in it and two folded washcloths on either side of the countertop, 30 cm from edge closest to patient	Washcloths, faucet handle	Reach to touch faucet knob. Place washcloth in paretic hand and bring to both sides of face.	Open and close faucet. Lift washcloth from basin and wring it out.
Applying deodorant	Tabletop with deodorant placed at midline, 25 cm from edge closest to patient	Deodorant	Reach to touch deodorant. Place deodorant in paretic hand and bring to opposite armpit.	Lift deodorant for 3 seconds. From the horizontal position, rotate deodorant upright and return to original position.
Hair combing	Tabletop with comb placed at midline, 25 cm from edge closest to patient	Comb	Reach to touch comb. Place comb in paretic hand and bring to both sides of head.	Lift comb for 3 seconds.
Don/doffing glasses	Tabletop with glasses placed at midline, 25 cm from edge closest to patient	Glasses	Reach to touch glasses.	Lift glasses for 3 seconds.
Eating	Table top with a standard-size paper plate (at midline, 2 cm from edge), utensils (3 cm from edge, 5 cm from either side of plate), a baggie with a slice of bread (25 cm from edge, 23 cm left of midline), and a margarine packet (32 cm from edge, 17 cm right of midline)	Fork, knife, re-sealable sandwich baggie, slice of bread, single-serve margarine container	Reach to touch each item separately on paretic side. Place fork in paretic hand and bring fork to mouth.	Lift each object on paretic side for 3 seconds.

Table 6: Description of the activities performed by the severely impaired patients in the cohort (1/2). * Instructions for the severely impaired patients were given based on the UE segment with greater preserved function. Proximal > distal indicates better strength in the proximal (i.e. deltoid, biceps, triceps) than distal (i.e. hand) UE, which was typically paralyzed in these patients. The initial UE position was generally at the edge of the table/counter closest to the patient. Distal > proximal had the opposite distribution of strength. The initial UE position was adjacent to the target object. All testing were done on the paretic UE.

Activity	Workspace	Target object(s)	Instructions*	
			Proximal >Distal	Proximal <Distal
Drinking	Tabletop with water bottle and paper cup 18 cm to the left and right of midline, 25 cm from edge closest to patient	Water bottle (12 oz), paper cup	Reach to touch object on paretic side. Reach across to touch object on non-paretic side.	Starting from upright position, lay object on paretic side horizontally, release, and return to upright. Perform same series of actions on the object on the non-paretic side.
Tooth brushing	Sink with toothpaste and toothbrush on either side of the countertop, 30 cm from edge closest to patient	Travel-sized toothpaste on left, toothbrush with built-up foam grip on right, faucet handle	Reach to touch object on paretic side. Place toothbrush in paretic hand and bring it to mouth.	Lift object on paretic side for 3 seconds.
Moving object on a horizontal surface	Horizontal circular array (48.5 cm diameter) of 8 targets (5 cm diameter)	Toilet paper roll (200 g) or can (200 g)	Investigator will assess if toilet paper roll can be grasped, or aluminum can if not. If grasp is possible, move roll/can between the center and each outer target, resting before and after each motion. If grasp is not possible, the toilet paper roll will be moved around the target array by the investigator and the patient will reach to touch it at each location.	Investigator will assess if toilet paper roll can be grasped, or aluminum can if not. If grasp is not possible, the toilet paper roll will be moved around the target array by the investigator and the patient will reach to touch it at each location.
Moving object on/off a Shelf	Shelf with two levels (33 cm and 53 cm) with 3 targets on both levels (22.5 cm, 45 cm, and 67.5 cm away from the left-most edge)	Toilet paper roll (200 g) or can (200 g)	Investigator will assess if toilet paper roll can be grasped, or aluminum can if not. If grasp is possible, move roll/can between the center and each outer target, resting before and after each motion. If grasp is not possible, the toilet paper roll will be moved around the target array by the investigator and the patient will reach to touch it at each location.	Investigator will assess if toilet paper roll can be grasped, or aluminum can if not. If grasp is not possible, the toilet paper roll will be moved around the target array by the investigator and the patient will reach to touch it at each location.

Table 7: Description of the activities performed by the severely impaired patients in the cohort (2/2).

Joint/segment	Anatomical angle
Shoulder	Shoulder flexion/extension Shoulder internal/external rotation Shoulder ad-/abduction Shoulder total flexion‡
Elbow	Elbow flexion/extension
Wrist	Wrist flexion/extension Forearm pronation/supination Wrist radial/ulnar deviation
Thorax	Thoracic* flexion/extension Thoracic* axial rotation Thoracic* lateral flexion/extension
Lumbar	Lumbar† flexion/extension Lumbar† axial rotation Lumbar† lateral flexion/extension

Table 8: List of anatomical angles. The system uses a rigid-body skeletal model to convert the IMU measurements into joint and segment angles. ‡ Shoulder total flexion is a combination of shoulder flexion/extension and shoulder ad-/abduction. *Thoracic angles are computed between the cervical vertebra and the thoracic vertebra. †Lumbar angles are computed between the thoracic vertebra and pelvis.

Method	Random forest	FCNN	CNN	LSTM	Proposed	Ensemble
Test set 1	59.66	62.43	64.98	68.21	70.67	71.87
Test set 2	33.79	39.62	43.11	48.78	44.44	48.36

Table 9: Accuracy on Test set 1 and Test set 2 of the machine-learning models described in Section 5. FCNN denotes fully connected neural network. The ensemble is a combination of the proposed model and the LSTM, where the output probabilities were averaged.

L	S	Fold 1	Fold 2	Fold 3	Fold 4	Average
1	2	56.23	54.77	57.97	57.21	**56.55**
1	3000	56.31	54.73	58.05	57.06	56.54
1	1500	56.27	54.80	58.08	56.99	56.53
1	120	56.22	54.58	58.07	57.06	56.48
1	700	56.29	54.72	57.92	56.95	56.47
1	300	56.36	54.58	57.86	56.92	56.43
1	50	56.19	54.56	57.92	56.82	56.37
5	120	56.27	54.57	57.80	56.82	56.37
5	300	56.23	54.62	57.73	56.80	56.34
1	20	56.06	54.48	57.93	56.88	56.34

Table 10: Validation accuracies for the random-forest models. We report the top 10 performing models. L denotes the minimum number of samples required to be at a leaf and S denotes the minimum number of examples required to split each internal node. We experimented with the following values: L = {1,5,20,50,150,400,800,1500}, S = {2,20,50,120,300,700,1500,3000}.

# of layers	Dim. of hidden units	Fold 1	Fold 2	Fold 3	Fold 4	Average
4	300	56.86	55.47	56.85	55.49	56.17
4	600	55.44	56.73	58.23	55.79	56.55
4	900	56.02	56.89	58.13	54.83	56.47
8	300	56.52	56.22	58.28	54.4	56.36
8	**600**	**56.54**	**56.43**	**58**	**56.01**	**56.75**
8	900	57.05	56.27	57.31	55.68	56.58
12	300	55.27	56.16	55.83	55.45	55.68
12	600	56.23	56.51	57.68	54.47	56.22
12	900	55.9	56.68	58	55.57	56.54

Table 11: Validation accuracy for fully-connected neural network models with different number of layers, different dimensions of hidden units, and no dropout.

# of layers	Dim. of hidden units	Fold 1	Fold 2	Fold 3	Fold 4	Average
4	300	58.01	58.12	58.83	56.57	57.88
4	600	56.78	58.06	59.07	55.80	57.43
4	900	57.70	58.21	58.37	56.64	57.73
8	300	57.70	57.64	58.47	56.32	57.53
8	**600**	**57.41**	**58.62**	**59.22**	**56.67**	**57.98**
8	900	57.06	58.22	59.14	56.34	57.69
12	300	57.43	57.49	58.78	55.70	57.35
12	600	57.29	57.01	59.67	56.60	57.64
12	900	57.40	57.97	58.90	55.68	57.49

Table 12: Validation accuracy for fully-connected neural network models with different number of layers, and different dimensions of hidden units. The dropout rate was set to 0.2 for all the models.

# of layers	Dim. of hidden units	Fold 1	Fold 2	Fold 3	Fold 4	Average
4	300	58.20	58.19	59.62	57.16	58.29
4	600	59.39	58.86	55.96	56.34	57.64
4	900	58.43	58.57	60.34	57.25	58.65
8	300	57.08	56.60	58.31	56.28	57.07
8	600	58.14	58.59	60.08	57.22	58.51
8	**900**	**58.08**	**58.86**	**60.33**	**57.38**	**58.66**
12	300	57.11	57.10	57.95	55.63	56.95
12	600	57.81	58.14	58.91	56.24	57.78
12	900	58.09	57.92	59.85	57.06	58.23

Table 13: Validation accuracy for fully-connected neural network models with different number of layers, and different dimensions of hidden units. The dropout rate was set to 0.5 for all the models.

Dim. of hidden units	Fold 1	Fold 2	Fold 3	Fold 4	Average
400	57.4	61.78	61.68	59.43	60.07
1200	59.05	61.77	62.32	61.64	61.20
2000	61.57	62.85	64.38	60.95	62.44
2800	61.27	63.21	64.11	61.74	62.58
3600	59.79	62.07	64.18	63.46	62.38
4000	**60.68**	**63.28**	**65.71**	**64.1**	**63.44**
4500	60.81	63.55	65.19	62.76	63.08

Table 14: Validation accuracies for the LSTM models.

DenseNet-style convolutional model						
	Normalization	Fold 1	Fold 2	Fold 3	Fold 4	Average
Input embedding	IN	**64.82**	**65.91**	**69.14**	**66.65**	**66.63**
Input embedding	BN	62.34	65.72	66.07	63.98	64.53
No input embedding	IN	62.71	63.64	65.69	61.19	63.30
No input embedding	BN	57.95	60.97	63.47	58.39	60.19
ResNet-style convolutional model						
	Normalization	Fold 1	Fold 2	Fold 3	Fold 4	Average
Input embedding	IN	**65.59**	**68.94**	**69.45**	**67.09**	**67.76**
Input embedding	BN	62.49	65.57	65.65	64.57	64.57
No input embedding	IN	61.57	59.15	62.21	63.12	61.51
No input embedding	BN	58.75	61.22	61.90	58.47	60.09

Table 15: Validation accuracies for DenseNet-style convolutional models (IN = Instance normalization, BN = Batch normalization)

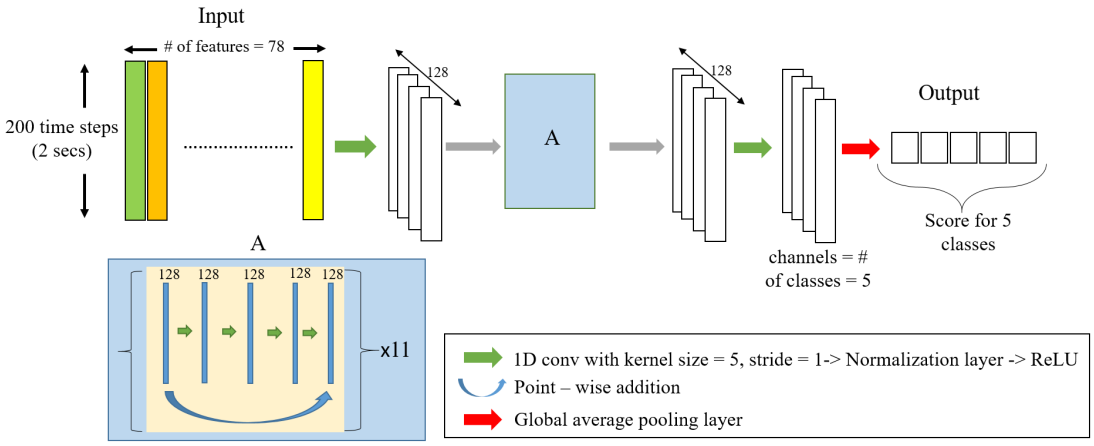

Figure 7: Diagram of the ResNet-style convolutional network

Appendix D. Convolutional Architectures

Figures 7 and 9 provide a detailed description of the baseline convolutional neural networks used for our experiments. Figure 8, and 10 show the modified architectures, which incorporate the input-embedding module described in Section 4.1.

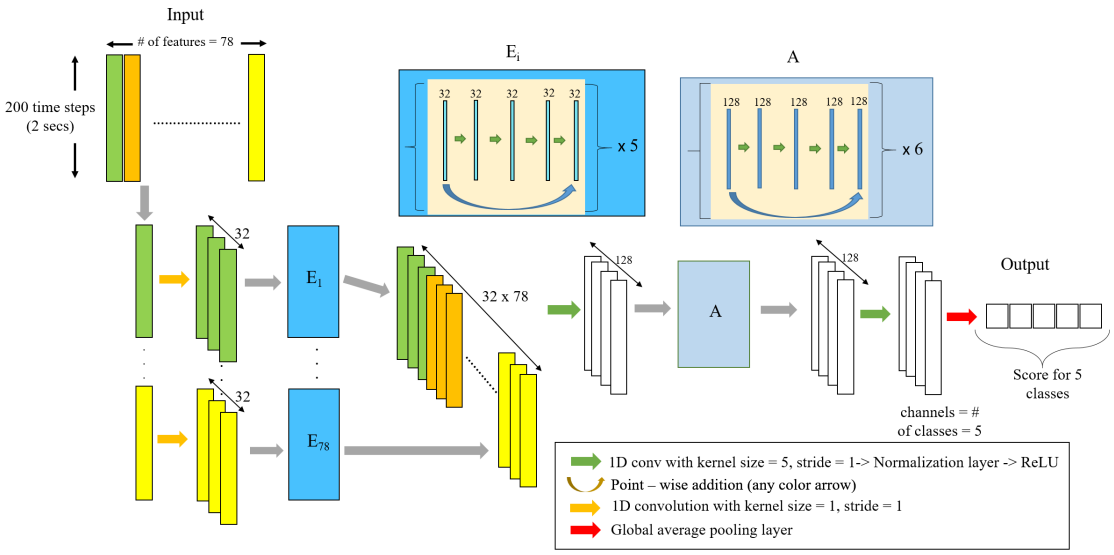

Figure 8: Diagram of the ResNet-style convolutional network incorporating the input-embedding module described in Section 4.1.

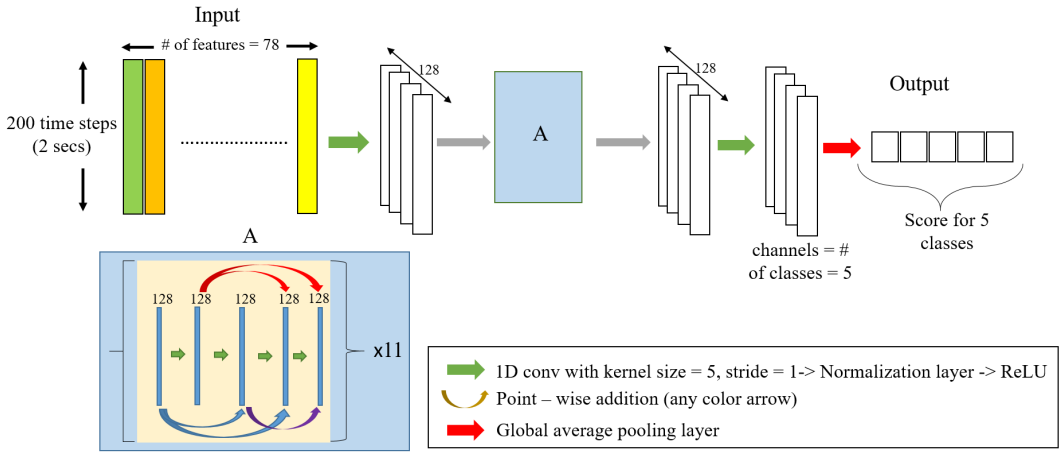

Figure 9: Diagram of the DenseNet-style convolutional network

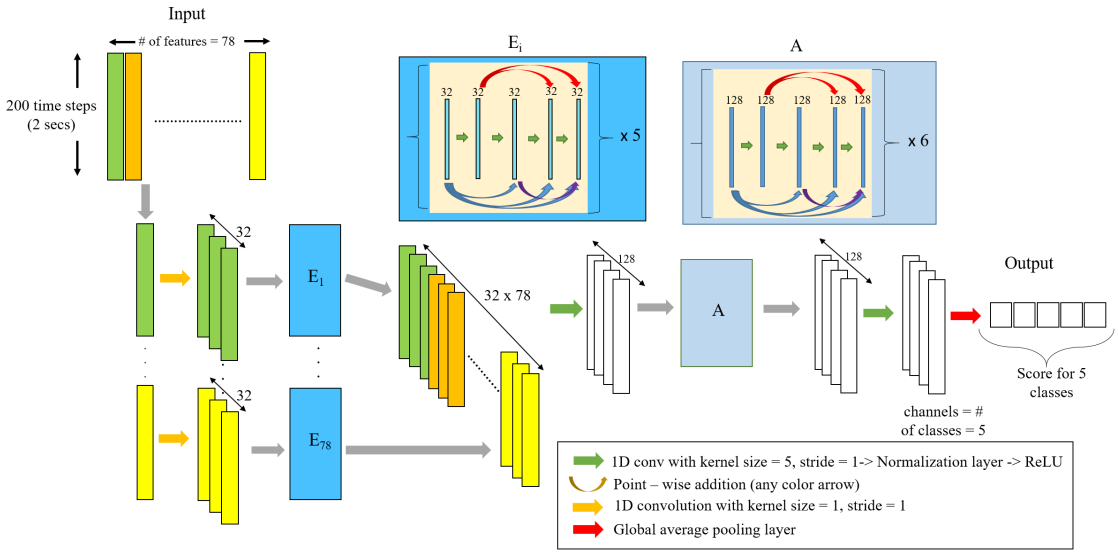

Figure 10: Diagram of the DenseNet-style convolutional network incorporating the input-embedding module described in Section 4.1.